Содержание

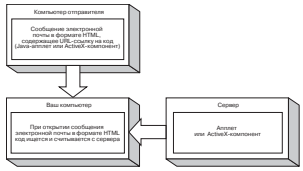
Как апплеты злоумышленника переносят вредоносный код:

Мобильный код для приложений в виде Java-апплетов, JavaScript и ActiveX-компонентов – это мощное орудие для передачи данных по Сети. Но это также и мощное орудие для распространения вредоносного кода. Вредоносные апплеты не саморазмножаются и не предназначены, например, для искажения данных, подобно вирусам. Но они часто становятся средством специфической атаки, направленной на хищение данных или приведение системы к краху.

Творческий подход к программированию:

- Осознайте возможность постороннего вмешательства в ваш код, ожидайте непредсказуемого!
- Сделайте свой код более лаконичным; часть кода, отвечающая за функциональность, должна быть минимальной.
- Проверка, проверка, проверка! Не надо полагаться только на себя и тем более замалчивать свои ошибки.

Как работает мобильный код в Java-апплетах и ActiveX-компонентах:



Мобильный код, содержащийся на Web-сервере

Инструментальные средства... Остерегайтесь ввода данных пользователем:

CGI-сценарии и программы обычно применяются, когда код разрешает пользовательский ввод, но часто данные, которые предлагает пользователь, не проверяются. Контроль информации, введенной пользователем, значительно сокращает шансы хакера на взлом вашей системы с помощью CGI-сценария.

Ознакомьтесь досконально с технологиями хакеров:

Вопрос: Что я должен делать, обнаружив «черный ход» в своем коде?
Ответ: Сначала нужно определить, действительно ли это «черный ход». Иногда некоторые части кода не имеют никаких аутентификационных данных и могут выполнять значимые операции, но тем не менее перед

их вызовом обеспечивается должная аутентификация. Если при тщательном анализе выявилось, что это «черный ход», пригласите сотрудника отдела безопасности, знающего данный язык программирования, и проведите аудит кода. Если сотрудник решит, что это «черный ход», то следует выяснить, был ли он создан преднамеренно или случайно.

Как проводить эффективную трассировку выполнения программ:

- Трассировка выполнения программы от начала до конца занимает слишком много времени.
- Вам удастся сэкономить время, если вместо этого вы займетесь непосредственно разрешением известных проблем.
- Такой подход оставляет в стороне логику обработки приложений.

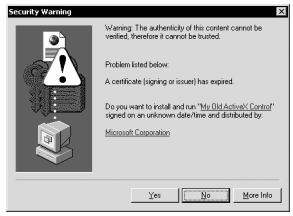
Рассмотрите все аспекты модели безопасности Java:

- Загрузчики классов.
- Проверка байт-кода.
- Менеджеры безопасности.
- Электронные цифровые подписи.
- Аутентификация с применением сертификатов.
- Подпись JAR-файлов.
- Шифрование.

Ущерб и защита... Отладка XSL:

Взаимодействие таблицы с XML-документом представляет собой довольно сложный процесс, и к сожалению, ошибки таблицы чаще всего являются криптографическими. Компания Microsoft предлагает XSL-отладчик, основанный на HTML, который предназначен для выполнения XSL. Вы сможете просмотреть и исходный код для его улучшения. Отладчик XSL доступен на сайте: http://msdn.microsoft.com/downloads/samples/internet/xml/sxl_debugger/default.asp.

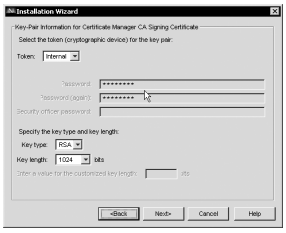
Разрешайте выполнение ActiveX-компонента, опираясь на информацию в окне предупреждения:



Как создавать безопасный ColdFusion-код:

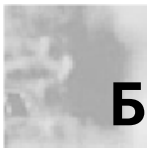
При разработке приложения в ColdFusion необходимо соблюдать осторожность с многими тегами, отвечающими за потоки данных, так как существуют различные методы атак на подобные теги. В большинстве случаев подтверждение данных, посылаемых странице, предотвратит некорректную работу с ними. В других ситуациях от взлома защитит предварительно установленный запрет некоторых тегов. Для каждого тега мы рассмотрим возможность его отключения (опция, управляемая администратором) или правильного кодирования, так как некоторые теги нельзя отключить.

Выбор криптографического токена, а также типа и длины ключа:



Список типовых дефектов при программировании в Java-среде, которые не так просто обнаружить при помощи стандартных методов тестирования:

- Избыточное копирование строк – ненужные копии постоянных объектов.
- Неспособность клонировать возвращаемые объекты.
- Ненужное клонирование.
- Ручное копирование массивов.
- Ошибочное копирование или создание только частичной копии.
- Повторное тестирование нулевого результата.
- Использование `==` вместо `.equals`.
- Смешение неатомарных и атомарных операций.
- Добавление неоправданных «ловушек».
- Неудачное выполнение операций присваивания, клонирования или хэширования.



Благодарности

Мы хотели бы выразить признательность следующим людям за участие и поддержку, благодаря которым появилась эта книга.

Ричарду Кристофу (Richard Kristof) и Дункану Андерсону (Duncan Anderson) из Global Knowledge за предоставление лучших курсов по информационным технологиям и учебного оборудования, привлечение лучших преподавателей.

Ральфу Троупу (Ralpf Troupe), Ронде Сент-Джон (Rhonda St. John) и команде Callisma – их опыт в области проектирования, развертывания и поддержки глобальных корпоративных сетей просто неоценим.

Карен Кросс (Karen Cross), Лансу Тилфорду (Lance Tilford), Мегану Каннингэму (Meaghan Cunningham), Киму Уайли (Kim Wylie), Гарри Керчнеру (Harry Kirchner), Биллу Ричтеру (Bill Richter), Кевину Вотелу (Kevin Votel) и Бриттину Кларку (Brittin Clark) из Publishers Group West, охотно делившимся своими экспертными знаниями и опытом в области маркетинга.

Мэри Джинг (Mary Ging), Каролин Херд (Caroline Hird), Саймону Билу (Simon Beale), Каролин Уиллер (Caroline Wheeler), Виктории Фуллер (Victoria Fuller), Джонатану Банкеллу (Jonathan Bunkell), а также Клаусу Берану (Klaus Beran) из Harcourt International, следившим, чтобы мы не упустили важных деталей при подготовке книги.

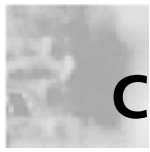
Анеку Бейтену (Anneke Baeten), Аннабел Дент (Annabel Dent) и Лаури Джайлс (Laurie Giles) из Harcourt Australia за помощь.

Дэвиду Бакленду (David Buckland), Венди Вонг (Wendi Wong), Дэниелу Ло (Daniel Loh), Мэри Чинг (Marie Chieng), Люси Чонг (Lucy Chong), Лесли Лим (Leslie Lim), Одри Гэн (Audrey Gan) и Джозефу Чан (Joseph Chan) из Transquest Publishers за веру в наши силы.

Квону Санг Джуну (Kwon Sung June) из Acorn Publishing за поддержку.

Итану Аткину (Ethan Atkin) из Cranbury International за помощь в расширении программного обеспечения Syngress.

Джо Писко (Joe Pisco), Хелен Мойер (Helen Moyer) и всем сотрудникам InterCity Press за содействие в создании книги.



Соавторы

Крис Брумс (Chris Broomes) – MCSE, MCT, MCP+I, CCNA – старший сетевой аналитик из DevonIT (www.devonitnet.com), ведущего сетевого провайдера, специализирующегося в области сетевой безопасности и VPN. Крис работает в IT-индустрии уже более 8 лет и имеет многосторонний технический опыт. Крис – основатель и президент Infinite Solutions Group (www.infinitesols.com), являющейся сетевой консалтинговой фирмой, расположенной в Лансдауне (Landsdowne), штат Пенсильвания, и специализирующейся по сетевому проектированию, системной интеграции, информационной безопасности, подготовке технической литературы и обучению. В настоящее время Крис работает над системами на основе Cisco и Netscreen VPN, имея сертификаты CCDA и CCNP.

Джефф Форристал (Jeff Forristal) – ведущий разработчик по информационной безопасности для консалтинговой фирмы Neohapsis, основанной в Чикаго. Кроме ассистентской деятельности в оценках сетевой безопасности и проверках безопасности приложений (включая проверку исходного кода) Джефф активно занимается подготовкой еженедельного объединенного информационного бюллетеня по угрозам безопасности Security Alert Consensus, публикуемого Neohapsis, Network Computing и SANS Institute.

Дрю Симонис (Drew Simonis) – CCNA – консультант по безопасности для Fiderus Strategic Security и Privacy Services. Специалист по информационной защите с опытом работы: в области разработки руководств по безопасности, анализе произошедших атак, выявлении и предотвращении взломов, а также в области сетевого и системного администрирования. Он прекрасно знаком с протоколом TCP/IP и операционной системой UNIX (в основном AIX и Solaris) и имеет навыки практической работы с устройствами маршрутизации и коммутации. Он принимал участие в работе над несколькими масштабными Web-проектами для таких компаний, как AT&T, IBM, и ряда клиентов этих компаний. В его обязанности входило планирование и разработка проектов, связанных с банковскими онлайновыми системами, системами информационной поддержки покупателей и онлайновыми системами адаптивной гарантированной оценки, которые используются большинством национальных страховых компаний. Дрю помогает покупателям в оценке сетевой безопасности и безопасности приложений, а также ассистирует при разработках. Он является членом MENSA и обладает семью квалификационными сертификатами основных IT-компаний: IBM Certified Specialist, AIX

4.3 System Administration, AIX 4.3 Communications, Sun Microsystems Certified Solaris System Administrator, Sun Microsystems Certified Solaris Network Administrator, Checkpoint Certified System Administrator и CheckPoint Certified Security Engineer. Он проживает в Тампе (Tampa), штат Флорида.

Брайн Бэгнолл (Brian Bagnall) – Java-программист и разработчик, обладатель сертификата Sun, автор книги «Sun Certified Programmer for Java 2 Study Guide». В настоящее время является ведущим программистом в канадской компании Idle Work. Idle Work разрабатывает решения в области распределенного процессинга для крупного и среднего бизнеса, требующего высокопроизводительных вычислений. Ранее Брайн работал в IBM над заказными приложениями. Брайн – один из создателей Lejo – комплекта Java-программ для Lego Mindstorms. Брайн просил поблагодарить за поддержку его семью, и особенно его отца Херба (Herb).

Майкл Дайновиц (Michael Dinowitz) владеет CF-Talk – это объемная почтовая рассылка ColdFusion сайта www.houseoffusion.com. Он публикует и пишет статьи для Fusion Authority Weekly News Alert (www.fusionauthority.com/alert). Майкл – автор книги «Fusebox: Methodology and Techniques» издания ColdFusion и является соавтором бестселлера «ColdFusion Web Application Construction Kit». Страсть Майкла к языкам программирования очевидна всегда: исследует ли он с низших уровней функциональность ColdFusion или выступает перед аудиторией. Вне Allair есть только несколько подобных «миссионеров», посвятивших себя распространению языков и укреплению сообщества.

Джэй Д. Дайсон (Jay D. Dynson) – старший консультант по безопасности компании OneSecure, хорошо зарекомендовавшей себя как провайдер управления сервисами безопасности. Джэй также работает на полставки советником по безопасности в NASA. Помимо своей основной работы он занимается поддержкой сайта www.treachery.net и, как один из основателей, обслуживанием www.attrition.org.

Джо Дьюлэй (Joe Dulay) – MCSD – вице-президент по технологиям в компании IT Age Corporation. IT Age Corporation – фирма по управлению проектами и разработке программного обеспечения в Атланте (Atlanta), штат Джорджия, специализирующаяся на системах, ориентированных на поддержку потребителей и онлайновой коммерции. Обязанности Джо на данный момент включают управление отделом по информационным технологиям, работу в качестве главы технологической комиссии по архитектуре программного обеспечения, управление онлайновой коммерцией, а также совершенствование методологий и процесса разработки. Хотя большинство его обязанностей – управленческие, он все-таки остается активным участником команды по исследованиям и разработке. Джо имеет степень бакалавра компьютерных наук университета Висконсин (Wisconsin). Ранее он занимал должность старшего разработчика в компании Siemens Energy and

Automation и позицию независимого разработчика, специализирующегося на онлайновой коммерции. Джо хотел бы поблагодарить свою семью, которая всегда готова оказать ему поддержку и помощь.

Майкл Кросс (Michael Cross) – MCSE, MCPS, MCP+I, CNA – имеет сертификаты системного инженера, специалиста по продуктам фирмы Microsoft, а также сертификат администратора Novell. Майкл работает сетевым администратором, Internet-специалистом и программистом в компании Niagara Regional Police Service. Он отвечает за сетевую безопасность, администрирование, программирование приложений, а также является Web-мастером сайта компании www.nrps.com. Он консультировал и ассистировал расследования компьютерных преступлений в сети Internet и, являясь членом команды Information Technology (Информационная технология), обеспечивает поддержку базы данных, включающей более 800 пользователей.

Майкл владеет компанией KnightWare, предоставляющей услуги консультирования, программирования, сетевой разработки, Web-дизайна, тестирования и др. Он обслуживал в качестве инструктора частные колледжи и технические школы в Лондоне (London), штат Онтарио, Канада. Майкл – свободный писатель, и за последние нескольких лет он опубликовал более 25 книг и сборников. В настоящее время Майкл проживает в Санта-Катарине (St. Catharines), штат Онтарио, Канада, с невестой Дженнифер.

Эдгар Даниелян (Edgar Danielyan) – CCNA – сейчас работает на собственном предприятии. Эдгар имеет диплом юриста Британского юридического института (British Institute of Legal Executives) и является сертифицированным специалистом университета Южного Колорадо. Он работает в Армении сетевым администратором и менеджером высокоуровнего домена. Также он работал в ООН, Министерстве обороны, национальной телекоммуникационной компании, банке и был партнером адвокатской фирмы. Он владеет четырьмя языками, любит хороший чай и является членом ACM, IEEE CS, USENIX, CIPS, ISOC и IPG.

Дэвид Г. Скабру (David G. Scarbrough) – старший разработчик Американской образовательной сети (Education Network of America), в которой он является руководящим членом команды разработчиков ColdFusion. Он специализируется на создании сайтов онлайновой коммерции. Дэвид обладает сертификатом ColdFusion 4.5 Master и имеет опыт работы с HTML, JavaScript, PHP, Visual Basic, ActiveX, Flash 4.0 и MS SQL Server 7. Он также занимал должности программиста и научного сотрудника. Дэвид закончил университет (Troy State University) в Монтгомери (Montgomery), штат Алабама, с дипломом бакалавра компьютерных наук. Он живет в Смирне (Smyrna), штат Теннесси.

Технический редактор и соавтор

Джули Тракслер (Julie Traxler) – тестировщик программного обеспечения для Internet. Джули работала в компаниях DecisionOne, EXE Technologies и TV Guide, исполняя обязанности менеджера проектов, бизнес-аналитика и технического автора. Как системный аналитик и дизайнер, Джули создает процедуры QA-тестирования, формирует команды тестеров и осуществляет процесс тестирования. Разработанные ею планы включают тестирование функциональности, эргономичности, соответствия требованиям, безопасности, целостности и производительности.

Технические рецензенты

Кевин Цайсе (Kevin Ziese) – научный сотрудник компании Cisco Systems. Ранее он был старшим научным сотрудником и основателем компании Wheelgroup Corporation, которая приобретена Cisco Systems в апреле 1998 года. До основания Wheelgroup Corporation он был главой компании Advanced Countermeasures Cell.

Роберт Хансен (Robert Hansen) – компьютерный эксперт-самоучка, проживающий в Северной Калифорнии. Роберт, известный как Rsnake (сейчас как Rsenic), был активно вовлечен в хакинг с середины 1990-х годов и сегодня продолжает работать с различными группами хакеров. Роберт работал в известной баннерной рекламной компании как информационный специалист, а также в нескольких начинающих компаниях в качестве директора по операциям и руководителя по безопасности информации. Он создал несколько сайтов и организаций, связанных с безопасностью, у него брали интервью многие журналы, газеты и телевизионные шоу, такие как Forbes Online, Computer World, CNN, FOX и ABC News.

Предисловие

Эта книга обращает ваше внимание на проблему безопасности от начальной до конечной стадии разработки приложений. Дожидаться аудита или жалоб пользователей приложений – это не выход. Мы не утверждаем, что обязательно найдется способ полностью устранить риск злонамеренного нападения на ваш код, но полагаем, что, используя инструкции и рекомендации, данные в этой книге, вы существенно уменьшите и вероятность атаки, и возможный ущерб от нее.

Авторы подробно рассматривают следующие важные темы обеспечения успешной защиты Web-приложений от атак:

- В вашей организации обязательно должны быть исследованы, спланированы, разработаны и описаны меры по обеспечению безопасности. Они должен включать планы сетевой безопасности, безопасности приложений и защиты рабочих мест. Все разработчики, администраторы и группы обеспечения качества должны принимать участие в создании плана и, в конечном счете, знать свою роль в процессе обеспечения безопасности.
- Тестирование – фундаментальный компонент безопасности приложения. Тесты по безопасности должны учитывать все возможные нападения, чтобы доказать успех или несостоятельность выбранных мер безопасности. К укреплению «обороноспособности» должно быть приложено как можно больше усилий, чтобы сделать невозможным легкий взлом системы и предотвратить нападения хакеров, поскольку для этого им потребуется слишком много времени и труда.
- Разработчики должны быть готовы к постоянным изменениям и/или к совершенствованию прикладных методов и средств, которые они используют. Это важно при разработке приложений, учитывая быстрый темп изменения технологий. Часто бывают доступны новые методы и средства, которые, однако, не используются из-за недостатка осведомленности или из-за нехватки времени, связанного с текущей работой.
- Разработчики, Web-мастера и сетевые администраторы должны быть в курсе последних известных методов осуществления угроз безопасности; для этого достаточно просматривать такие сайты, как www.securityfocus.com или www.cert.org. Эти сайты предлагают не

только перечень всех текущих угроз, но также форум для разработчиков, где последние могут проконсультироваться по какой-либо проблеме безопасности: получить советы или решения по зарегистрированным в базе проблемам.

Защита должна быть многоуровневой, что гарантированно приводит к сложностям реализации атаки на всех уровнях. Что хорошо работает для одного языка программирования, может не иметь успеха для другого. Основная цель автора этой книги – пояснить разработчикам проблемы безопасности, свойственные каждой стадии программирования, и предложить правильные, надежные решения.

Глава 1 предлагает критерии классификации хакеров и их мотивов. В главе 2 рассматривается необходимость наличия у программиста творческого мышления. Объясняются недостатки разработки программного кода без полного понимания его функциональности и использования его без обеспечения безопасности. Препятствиями творческому и аналитическому мышлению могут быть: контролируемая руководством среда; интересы бизнеса, которые не способствуют реализации требований безопасности; производственный регламент; приверженность старой технологии; недостаток средств и времени при реализации проектов. В главе 3 исследуются проблемы безопасности, связанные с использованием VBScript, JavaScript, ActiveX и других форм мобильного кода, в контексте безопасности пользователя и эффективности работы приложения. При применении этих мощных технологий программирования функциональные возможности приложения, его реальная безопасность подвержены риску. В главе 4 объясняется уязвимость использования внешних программ на Web-сервере. Глава 5 представляет вниманию читателя как различные инструментальные средства и методологии, используемые хакером для успешного взлома, так и всевозможные атаки, которые могут быть предприняты взломщиком. Глава 6 описывает методы трассировки исходного кода в различных языках программирования, не свободных от потенциальной уязвимости, а также действия, которые должны быть предприняты разработчиками для устранения уязвимости кода. Главы 7, 8, 9 и 10 рассматривают различные типы рисков безопасности, которые связаны с языками Java и JavaScript, XML, ActiveX и ColdFusion. Глава 11 определяет такие понятия, как PGP, электронная подпись, сертификаты и PKI, используемые с целью формирования защиты ваших Web-приложений. В последней главе 12 перечислены методы проверки кода, обеспечивающие безопасность при разработке нового кода.

Джули Тракслер

Глава 1

Методология хакинга

В этой главе обсуждаются следующие темы:

- **Краткая история хакинга**
- **Мотивы хакера**
- **Наиболее распространенные типы атак**
- **Распознавание угроз безопасности Web-приложений**
- **Изучение хакерских методов для защиты от взломов**

☑ **Резюме**

☑ **Конспекты**

☑ **Часто задаваемые вопросы**

Введение

Большинство читателей, вероятно, слышали об атаках на eBay, Yahoo, Amazon в феврале 2000 года, а также на другие коммерческие и некоммерческие сайты. Это были атаки типа Distributed Denial of Service (DDoS), то есть отказа в обслуживании. Произошедшее привлекло внимание IT-сообщества и прессы, различные специалисты начали активно изучать проблемы информационной безопасности. Компании занялись укреплением защиты от возможных нападений. Однако хакеры становятся все более искусными и изощренными, тем самым заставляя поднимать планку необходимой защиты на всех уровнях: от сетевого до уровня приложений.

Чтобы создать защиту, необходимо представлять себе, где и от кого можно ждать атак и какую цель преследуют злоумышленники. Настоящая книга призвана помочь разработать стратегию всесторонней и контролируемой защиты компьютерных систем и приложений от атак, целенаправленных или случайных. Специалисты, принимающие во внимание все возможные атаки, способны лучше оценить степень уязвимости систем и приложений.

Хакерами бывают как неопытные хулиганы, хвастающиеся тем, что стерли Web-сайт, так и профессиональные мошенники, которые, проникая в закрытые информационные системы, взламывая базы данных, компрометируют, угрожают и шантажируют, надеясь получить финансовую выгоду. И те и другие заслуживают общественного порицания.

Если в разговоре с человеком, знакомым с Internet, упомянуть имя Кевина Митника, то собеседник непременно поймет, о ком идет речь. Митник провел в тюрьме несколько лет за хакерские преступления и часто воспринимается общественностью как жертва обстоятельств, но для всех других хакеров он – культовый герой.

Возможно, Митник и помог хакингу оказаться в центре внимания, но он, конечно, не принимал участия в тех взломах. У многих людей сложилось представление, что хакинг – принципиально новое явление. Но как это далеко от истины! Из данной главы вы узнаете, что хакингу предшествовали различные виды программного и телефонного взлома.

В книге рассмотрены инструментальные средства для защиты от хакинга Web-приложений и предложен базовый подход, обеспечивающий безопасное управление сайтами, разработку более надежного кода, планирование защиты. Книга научит читателя «думать, как хакер», лучше защищать информационные ресурсы сайта и конфиденциальные данные, а также обеспечивать их целостность.

Терминология

Прежде всего определимся с терминологией, когда мы говорим о хакере. Для описания деятельности хакера существует множество терминов, имеющих разные толкования в зависимости от предмета обсуждения и предпочтений автора. Чтобы получить представление о том, как в информационном сообществе развивались словарь и культура хакинга, стоит взглянуть на файл жаргонизмов (http:// info.astrian.net/jargon).

Согласно словарю Вебстера, слово «хакинг» (hacking) определяется как разрушительное и вредоносное действие или изощренный способ решения какой-либо задачи. Хакером может быть просто энтузиаст своего дела. В мире IT не каждый хакер имеет злые намерения, а хакинг не всегда вредит кому-то. В IT-сообществе хакеры классифицируются в соответствии со своей этикой и намерениями. Важный определяющий критерий такой классификации – обнародует ли хакер информацию об уязвимостях того или иного объекта при их обнаружении. Хакеры могут относить себя к «хакерам в белых шляпах» (символ голливудских «хороших парней» – ковбоев, то есть людей, не имеющих злого умысла) или к «хакерам в черных шляпах», проникающим в сети и системы для реализации злонамеренных планов. Однако классификация индивидуумов по этическому критерию субъективна и расплывчата; имеются так называемые «хакеры в серых шляпах», которые негодуют при их отождествлении с любой из двух вышеописанных категорий. В любом случае черта, объединяющая всех хакеров, – это их восхищение хорошей интеллектуальной задачей. Человек, совершающий взлом, используя непонятный ему код (script kiddies) или исключительно с целью проникновения в системы других пользователей (crackers), считается среди профессиональных хакеров не более чем хулиганом.

В настоящей книге под понятием «хакеры» подразумеваются люди, осуществляющие несанкционированный доступ к системам или приложениям независимо от их намерений.

Краткая история хакинга

Зачатки хакинга появились в конце 1940-х – начале 1950-х годов, когда радиолюбители-энтузиасты настраивались на полицейские или военные радиочастоты для подслушивания сведений о происходящих событий. Эти «неохакеры» были просто любознательными «информационными наркоманами», занимающимися поиском интересных данных о государственной или

военной деятельности. Что-то волнующее было в приобщении к информации, недоступной другим людям.

Хакинг и технология стали неразлучны с конца 1960-х годов, когда телефонные системы компании Bell еще были несовершенными, и хакеры обнаружили возможность бесплатного пользования телефонной связью. С развитием технологии совершенствовались и методы хакинга.

Предполагают, что термин «хакер», который употребляется в ссылках на компьютерный хакинг, впервые появился в компьютерном сообществе Массачусетского технологического института (MIT). Тогда это слово отождествлялось с одаренным и увлеченным программистом, отличающимся индивидуалистическими и бунтарскими наклонностями. Оригинально мыслящие члены одного из клубов университета продемонстрировали эту черту характера, когда отказались от программного обеспечения компании Digital Equipment Corporation, поставляемого вместе с универсальным компьютером PDP-10, и создали собственную операционную систему ITS. Многие хакеры университета были вовлечены в работы лаборатории искусственного интеллекта.

В 1960-х годах появилась первая трансконтинентальная компьютерная сеть ARPANet, благодаря которой стало возможным объединение хакеров. Теперь они могли работать одной большой группой. Эта сеть позволила хакерам обсуждать их общие цели и задачи. Были разработаны принципы хакерской культуры и опубликованы коммуникационные стандарты (файл жаргонизмов, упомянутый ранее) для сетевого сотрудничества и взаимодействия.

Хакинг телефонных систем

Имя Джон Драпер (John Draper, псевдоним Cap'n Crunch) ассоциируется с телефонным хакингом. С помощью свистка, вложенного в виде подарка в продукт детского питания, Драпер обнаружил, что звуковой тон частотой 2600 Гц позволяет ему бесплатно звонить по телефону.

В середине 1970-х Стив Возниак (Steve Vozniak) и Стив Джобс (Steve Jobs), основатели компьютерной компании Apple, сотрудничали с Драпером, который заинтересовал их своим устройством Blue Box. Данное устройство предоставляло нелегальный доступ к незанятым телефонным линиям посредством генерации определенных частот. Джобс пользовался псевдонимом Berkley Blue, а Возниак – псевдонимом Oak Toebark. Эти люди и сыграли главную роль на первой стадии развития телефонного хакинга, названного *фрикингом* (phreaking).

Драпер на самом деле разработал очень хорошую систему. Он и другие фрикеры устраивали ночные конференции, на которых обсуждали обнаруженные возможности проникновения в телефонную систему. Чтобы