Лекция 21:
Безопасность (часть 2)

Алексей Линёв
Александр Мощук
Кирилл Погорельский

somes slides are adapted from the OS course at the University of Washington

Современные проблемы безопасности

- Internet породил эпидемию атак различных типов
  - *удаленные эксплойты (remote exploits):* атакующий взламывает вашу систему
  - *черви (worms):* выполняют клонирование и распространение атакующего кода
  - *армии взломанных систем (botnets)*
  - *программы-шпионы (spyware):* программное обеспечение, которое пытается украсть ваши данные
- Основные причины возникшей ситуации
  - большинство программ содержат ошибки
  - Internet спроектирован как открытая система
    - легко запустить новый сервис, но также легко найти и атаковать жертву
  - разобраться в вопросах защищенности достаточно тяжело
    - до сих пор не предложена достаточно простая абстрактная модель безопасности или подходящая универсальная информационная служба (Universal Information Services, UIS)
    - например, что на самом деле означает иконка блокировки в Internet Explorer?

Удаленный эксплойт (Remote exploit)

- Использует уязвимости в сетевом ПО
  - например, атаку через переполнение буфера

```
int main(int argc, char *argv[]){
  char buffer[10];
  strcpy(buffer, argv[1]);
  return 0;
}
```

    - можно использовать ошибку в программе, перезаписать стек и выполнить свой код
  - например, атаку через внедрение SQL
    - набираем в форме поиска в web-магазине: `“book tipping point; SELECT * FROM CREDITCARDS”`
    - смотрим номера кредитных карт

Использование уязвимостей в сетевом ПО: черви (Using remote exploits – worms)

- Псевдокод простейшего червя

```
for (i = 0.0.0.0;  i < 255.255.255.255;  i++) {
  открываем сетевое соединение с IP-адресом “i”;
  if (succeed) {
    пытаемся использовать уязвимость "x" узла  “i”;
    if (succeed) {
      посылаем сами себя на узел-жертву и запускаем;
    }
    закрываем соединение с “i”;
  }
}
```

- Будет ли данный червь распространяться?
  - насколько быстро?

Более "хороший" червь

```
while (1) {
  открываем сетевое соединение с узлом со случайным IP-адресом;
  if (succeed) {
    пытаемся использовать уязвимость "x" узла  “i”;
    if (succeed) {
      посылаем сами себя на узел-жертву и запускаем;
    }
    закрываем соединение с “i”;
  }
}
```

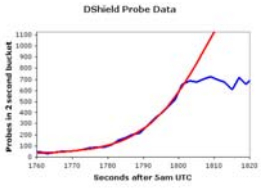
- Почему данный червь "лучше"?
- Насколько быстро он будет распространяться?
- Как сделать его еще "лучше"?

Еще "лучшие" черви…

- Сканирование локальных адресов
  - в первую очередь сканируются близлежащие узлы
    - взломанный узел 128.95.4.1
      - с вероятностью 37.5% сканируется 128.95.X.Y
      - с вероятностью 50% сканируется 128.X.Y.Z
      - с вероятностью 12.5% сканируется X.Y.Z.W
- Увеличение скорости сканирования -> более быстрое распространение
  - Code Red: примерно 5 сканирований в секунду
  - Sapphire: примерно 4000 сканирований в секунду
    - червь помещается в единственном UDP-пакете!
- Sapphire
  - количество зараженных узлов удваивается в течение 8,5 секунд
  - заражает множество из ~75000 уязвимых узлов в течение 5-10 минут!

Sapphire: история распространения

- Он распространялся слишком быстро, и это не позволило ему выжить!
  - узлам сети не наносится вред
  - но сетевые каналы сильно засоряются трафиком сканирования
  - к тому же помехи, которые различные копии создают друг другу, уменьшают скорость распространения



Удаленное использование уязвимостей – армии взломанных систем (Botnets)

- Шаг 1: взламываем удаленный компьютер
- Шаг 2: загружаем на него “botnet”-ПО
  - которое пока себя никак не проявляет
  - но позволяет атакующему управлять узлом-"зомби"
- Шаг 3: повторяем шаги 1 и 2 10,000 раз
  - собираем гигантскую армию узлов-"зомби"
- Шаг 4: управляем армией с "botnet-контроллера"
  - используем вычислительные мощности армии "зомби"
  - используем "зомби" для пересылки спама
  - используем "зомби" для организации атаки "отказ в обслуживании" (“denial of service attack”)

Отказ в обслуживании Denial of service

- Взломщик посылает внешне корректные запросы на обслуживание поставщику сервиса
- Поставщик сервиса выделяет ресурсы, необходимые для обслуживания запросов
  - сетевые порты, пространство для буферов, полосу пропускания,…
- Ресурсы истощаются, обслуживание настоящих пользователей ухудшается
- Возможно во всех случаях, когда операция формирования запроса на обслуживание значительно дешевле операции обработки запроса
  - все механизмы, основанные на запросах и подтверждениях

Пример: Phatbot

- Некоторые свойства:
  - использует полиморфное преобразование в момент заражения для обхода антивирусного ПО
  - посылает пробные e-mail для проверки возможности функционирования в качестве релея
  - может красть установочные ключи Windows
  - запускает ftp-сервер для распространения на другие узлы
  - запускает сервис перенаправления для TCP-соединений
    - ("подчищает" сетевой трафик)
  - может использовать несколько типов уязвимостей для распространения
    - (ведет себя как червь)
  - уничтожает другие черви и "botnet"-ПО для защиты своей сферы влияния
  - уничтожает процессы, в которых выполняется антивирусное ПО
  - крадет пароли к web-сайтам
  - собирает адреса e-mail для дальнейшего использования в списках рассылки спама

Шпионское ПО – Spyware

- ПО, устанавливающееся для сбора информации и передачи ее третьим лицам
  - клавиатурные шпионы (keyloggers), демонстраторы рекламы (adware), взломщики web-броузеров (browser hijackers),…
- Инсталлируются одним из следующих способов
  - внедряется в ПО, которое вы скачиваете из сети
  - скачивается "на лету"
    - web-броузер имеет уязвимости
    - web-сервер может их использовать, послав некорректный web-ответ
- Оценки
  - большая часть (50-90%) узлов, подключенных к Internet, имеют шпионское ПО
  - каждый 20-й исполняемый файл в Internet содержит шпионское ПО
  - порядка 0,5% web-страниц попытаются атаковать ваш компьютер с помощью загрузки "на лету" шпионского ПО

kingsofchaos.com

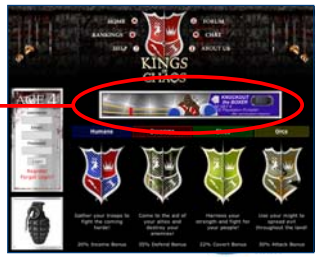
- Приятный, “безобидный” сайт для онлайновых игр
  - получает доход от рекламных сетей за показ баннеров
  - но управление содержимым рекламы отдано рекламодателям

kingsofchaos.com

- Приятный сайт для онлайновых игр
  - получает доход от рекламных сетей за показ баннеров
  - но управление содержимым рекламы отдано рекламодателям

рекламный баннер с adworldnetwork.com (зарегистрированная рекламная сеть)

javascript-код загружает HTML с сайта поставщика рекламы



Атака

- kingsofchaos.com было предоставлено следующее содержимое “рекламного блока”

```
<script type="text/javascript">document.write(‘
\u003c\u0062\u006f\u0064\u0079\u0020\u006f\u006e\u0055\u006f\
u0077\u0050\u006f\u0070\u0075\u0070\u0028\u0029\u003b\u0073\u
0068\u006f\u0077\u0048\u0069 …etc.
```

- Такая реклама
  - заваливала пользователя всплывающими окнами с рекламой
  - изменяла "домашнюю страницу" пользователя
  - использовала уязвимость IE, позволяющую устанавливать шпионское ПО

Что произошло?

- Рекламодатель оказался бывшим спамером
- Его целью было:
  - **заставить** пользователей смотреть рекламу с его сервера
  - увеличить доход, получаемый по "программам сотрудничества" при показе рекламы
    - порядок дохода – миллионы долларов
- Зачем он использовал шпионское ПО?
  - для управлением компьютером и показом рекламы даже в случае, если страница с вредоносной рекламой не открыта

Выводы…

- Обеспечение безопасности – нелегкая задача
  - фактически – противостояние, постоянно увеличивающееся в масштабе
  - мы лучше защищаемся – на нас лучше нападают
- Наши системы незащищены
  - ОС – одно из наиболее сложных человеческих творений
  - не удивительно, что у него есть недостатки!
- Современные тенденции
  - сократить ТСВ до такой степени, чтобы исключить из него ОС
  - разработать "песочницы", в которых запускать потенциально уязвимое ПО
    - ПО виртуальных машин (например, VMware)
  - программировать на более безопасных языках, чем C