Санкт-Петербургский Государственный Университет

Математико-механический факультет

Кафедра системного программирования

Разработка системы для мониторинга и анализа ботнетов, распространяемых через веб приложения

Курсовая работа студентки 445 группы

Переваловой Марины Андреевны

Научный руководитель: заведующий лабораторией

РУНЦ «Информационная безопасность»

Уральский федеральный университет Зеленчук И.В.

Санкт-Петербург

2012

Оглавление

Введение

Ботнетом называют сеть компьютеров, зараженных вредоносной программой, позволяющей злоумышленникам удаленно управлять зараженными машинами без ведома пользователей. Такие программы называются **ботами**.

Управление компьютером, зараженным ботом, может быть как прямым, так и опосредованным. В случае прямого управления злоумышленник может установить связь с инфицированным компьютером и управлять им, используя команды, встроенные в тело программы-бота. В случае опосредованного управления бот сам соединяется с центром управления или другими машинами в сети, посылает запрос и выполняет полученную команду.

Ботнеты обладают мощными вычислительными ресурсами. Одна такая сеть может содержать в себе десятки, сотни тысяч, а порой даже миллионы машин, при этом подавляющее большинство зомби-машин составляют инфицированные компьютеры ничего не подозревающих пользователей. На сегодняшний день ботнеты являются одним из основных источников нелегального заработка в Интернете и они только продолжают развиваться, становясь, тем самым, все более актуальной проблемой.

Сферы использования ботнетов:

- Рассылка спама.
- DDoS атаки.
- Анонимный доступ в сеть.
- Фишинг.
- Кража конфиденциальных данных.

Для обнаружения и перехватывания управления ботнетами нет единого подхода. Технологии ботнетов многочисленны и разнообразны и, помимо этого, продолжают развиваться. Изменяются способы распространения ботнетов, техники управления ими, способы обмена информацией между узлами и многое другое. Все это рождает потребность постоянного изучения, мониторинга и анализа ботнетов.

Цель

Первой и самой необходимой задачей для анализа каких бы то ни было ботнет-технологий является, собственно, наличие материала для анализа. И хотя количество зараженных компьютеров в мировой сети огромно, для исследования и выявления современных тенденций нужна немалая собранная база изучаемого материала.

В связи с этим целью данной курсовой работы стала разработка системы по захвату и анализу ботов, распространяющихся через веб приложения.

Выбранный подход

Способов распространения ботнетов в сети Интернет много, причем новые способы изобретаются киберпреступниками постоянно. К хорошо зарекомендовавшим себя способам можно отнести распространение через спам рассылки, съемные диски и распространение через известные уязвимости общих сетевых ресурсов. Для создания системы в данной работе был выбран последний метод, как наиболее популярный и удобный способ распространения ботов, не требующий участия пользователя.

В ходе работы был произведен анализ существующих веб уязвимостей. Ниже приведены некоторые из наиболее распространенных проблем:

- **Межсайтовый скриптинг** (Cross Site Scripting или XSS). Возможность инъекции HTML-кода в уязвимую страницу. Особенность данного типа атак состоит в том, что вместо непосредственной атаки уязвимого сервера этот сервер используется в качестве средства атаки на клиента.
- **SQL-инъекция**. Популярный способ взлома сайтов и программ, работающих с базами данных, основанный на внедрении в запрос произвольного SQL-кода. В этом методе параметры, передаваемые базе данных через веб приложения, изменяются таким образом, чтобы выполняемый SQL-запрос тоже изменился. Это может дать атакующему возможность выполнить произвольный запрос к базе данных, получить возможность чтения и/или записи локальных файлов и выполнения произвольных команд на атакуемом сервере.
- **Подделка межсайтовых запросов** (Cross Site Request Forgery, CSRF). Вид атак, использующий недостатки протокола HTTP. Принцип его работы состоит в том, что когда пользователь заходит на сайт, созданный злоумышленником, от его лица тайно

отправляется некий вредоносный запрос на другой сервер, на котором этот пользователь авторизован. В отличие от межсайтового скриптинга, где используется доверие пользователя к определенному сайту, здесь используется доверие некого сайта к браузеру пользователя.

- **Расщепление HTTP-запроса** (HTTP Response Splitting). При использовании этой уязвимости злоумышленник посылает серверу специальным образом сформированный запрос, ответ на который интерпретируется жертвой атаки как два разных ответа вместо одного. При этом второй ответ полностью контролируется злоумышленником, что дает ему возможность подделать ответ сервера.
- **Выполнение произвольного кода** (Remote Code Execution). Тип уязвимостей, позволяющих атакующему выполнение произвольного кода на уязвимой системе.

Опираясь на цели данной работы, в результате была выбрана уязвимость типа Remote Code Execution, позволяющая загружать на систему произвольный вредоносный код и, тем самым, вредоносные файлы. В дальнейшем работа велась именно с веб приложениями, обладающими таким типом уязвимости.

Установка системы

Выбранное и установленное ПО

На рабочей машине был установлен веб сервер Apache HTTP Server v2 и ряд веб приложений, обладающих уязвимостью PHP Remote Code Execution ([3]-[8]). Ниже представлен список установленных решений:

- Система управления содержимым (Content Management System, CMS) e107. Версия 0.7.13.
- Веб приложение для администрирования MySQL PhpMyAdmin. Версия 3.3.7.
- ImpressPages CMS. Версия 1.0.12.
- Веб форум phpBB. Версия 2.0.15.
- WordPress CMS. Версия 3.3.1. Вдобавок к этому установлено два уязвимых плагина данной CMS:
    - WP-Syntax Plugin. Версия 0.9.8.
    - Zingiri Web Shop Plugin. Версия 2.2.3.

Конфигурация сервера

Исходящие соединения

В ходе работы все исходящие соединения на сервере были запрещены после случая взлома системы, позволяя, тем самым, принимать загружаемые на систему боты и другие вредоносные файлы, но не давая злоумышленникам использовать систему в своих целях и атаковать другие сервера.

Логирование

Лог-файл веб сайта – текстовый файл, в котором регистрируются все запросы к сайту, а также все ошибки, связанные с этими запросами. Логирование позволяет отслеживать все действия злоумышленников при попытке взлома системы, что удобно для оценки различных способов взлома, их частоты и успешности.

По умолчанию Apache ведет логирование всех запросов, сохраняя строки запросов. Этого достаточно для изучения GET запросов, используемых для получения информации с заданного ресурса. GET запрос отправляет всю информацию в заголовке запроса и его тело остается пустым. Однако, в случае POST запросов, используемых для передачи данных на сервер и, тем самым, представляющих особый интерес, для анализа отправленной злоумышленником информации необходимо сохранение тела POST запроса.

Логирование POST запросов на рабочем сервере было настроено при помощи общедоступного модуля Apache Mod_Security. Данный модель предоставляет возможности как обнаружения, так и предотвращения вторжения на веб сервер. Так как последнее контрпродуктивно целям данной работы, модуль был сконфигурирован таким образом, что защитные свойства модуля были отключены, и было оставлено исключительно логирование.

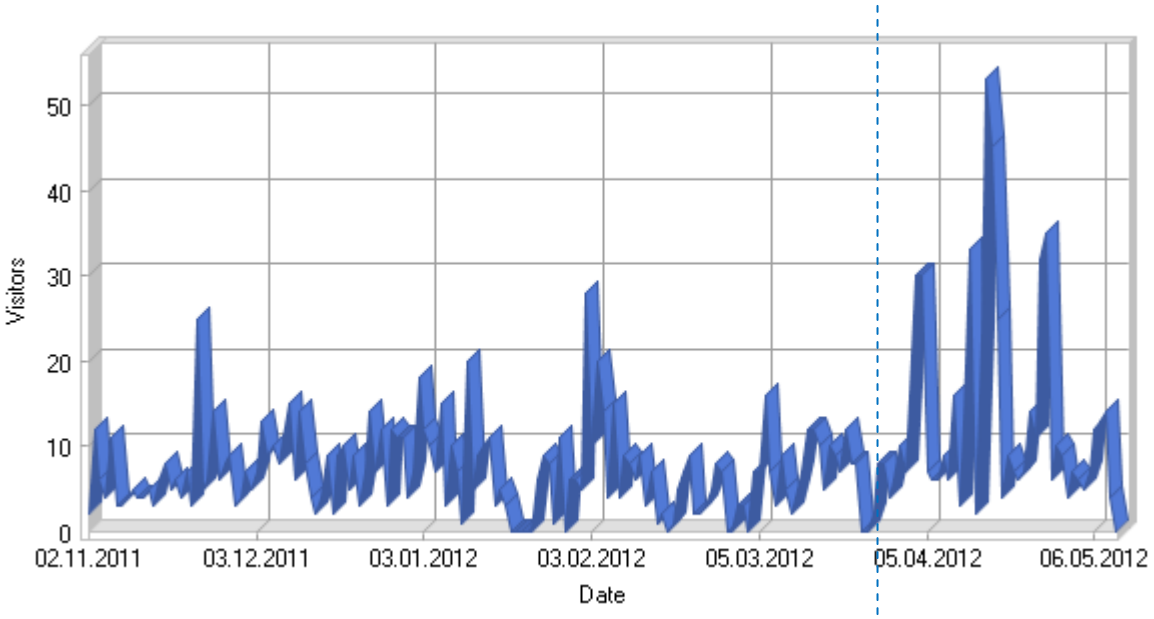
Повышение активности

Для повышения количества попыток взлома, для каждого из установленных веб сайтов было прописано свое доменное имя, после чего все они были зарегистрированы в таких популярных поисковых системах, как Google и Yandex. После этого зловредная активность на сервере возросла, что неудивительно, учитывая то, что без наличия регистрации в каких-либо поисковых системах сервер можно найти только через прямое сканирование диапазона ip-адресов.

Статистика и наблюдения

Данная работа началась в рамках летней школы «IT Summer SPb 2011». На первом этапе работы было установлено вышеупомянутое приложение e107 CMS. За месяц работы на систему было загружено около 20 ботов. По результатам исследований в ноябре 2011г. была сделана публикация на межвузовской научно-практической конференции «Актуальные проблемы организации и технологии защиты информации» СПбГУ ИТМО [9].

Работа продолжилась осенью, и система была расширена установкой остальных вышеперечисленных приложений. Интересно отметить, что, несмотря на хорошую активность e107 CMS в первые месяцы работы и успешную загрузку файлов через нее, в последнее 2-3 месяца попыток воспользоваться имеющейся уязвимостью практически не возникало. Данное наблюдение только подтверждает нестабильность и изменение способов распространения ботнетов.



(рис. 1)

На рис. 1 представлен график, отражающий обращения к серверу за весь период работы. На начало работы была установлена только одна CMS. Однако, несмотря на установку других веб приложений в процессе работы, по представленному графику можно увидеть, что это не слишком повлияло на общую посещаемость сервера. Это можно объяснить тем, что сканирование серверов на наличие уязвимостей производится, как правило, автоматически, отправлением определенных запросов на широкий диапазон ip-адресов, и не зависит от установленных веб приложений на машине. То есть большее число

уязвимых приложений влияет не на количество запросов, отправленных на сервер в целом, а на число именно успешных попыток взлома.

Так же можно отметить повышение активности после регистрации сайтов в поисковых системах (отмечено на графике пунктиром). Однако несмотря на это, активность на сервере все равно остается нестабильной и может сильно варьироваться от дня к дню.

Анализ загруженных файлов

Что касается файлов, залитых на систему, то здесь преобладали файлы двух типов. Боты, реализованные на языках PHP и Perl, и скрипты для представления удаленного доступа к зараженной машине, позволяющие злоумышленникам выполнять на ней произвольные команды. Данное наблюдение показывает, что чаще злоумышленники не загружают вредоносные файлы напрямую, а пользуются вспомогательными скриптами, предоставляющими им простоту и неограниченность загрузки файлов на зараженную систему.

Все боты, загруженные на сервер, были IRC-ориентированными, что означает, что управление ими производится на основе IRC (Internet Relay Chat). При таком подходе каждая зараженная машина соединяется с указанным в теле программы IRC-сервером, заходит на определенный канал и ждет команды от своего хозяина.

В большинстве загруженных ботов в теле программы был явно виден список доступных IRC каналов и IRC ников, используемых ботами в чате, что дает возможность подключения к IRC каналам самостоятельно и наблюдения за поведением находящихся там ботов. На рис. 2 приведен типичный кусок кода одного из ботов, демонстрирующий основные параметры подключения.

```
class pBot
{
 var $config = array("server"=>"46.249.58.41",
                     "port"=>"18498",
                     "pass"=>"",
                     "prefix"=>"{php}",
                     "maxrand"=>"5",
                     "chan"=>"#php",
                     "chan2"=>"#link",
                     "key"=>"vnc",
                     "modes"=>"+p",
                     "password"=>"h4s10",
                     "trigger"=>".",
                     "hostauth"=>"bash" // * for any hostname ( remember: /setvhost lAgi.seRius.sCan )
                     );
```

(рис. 2)

Так же в коде некоторых ботов были явственно видны возможные команды, отдаваемые центром управления, и поведение программы-бота в том случае. На рис. 3 представлен небольшой кусок такого кода.

```
case "eval":
  $eval = eval(substr(strstr($msg,$mcmd[1]),strlen($mcmd[1])));
break;
        case "sexec":
   $command = substr(strstr($msg,$mcmd[0]),strlen($mcmd[0])+1);
   $exec = shell_exec($command);
   $ret = explode("\n",$exec);
   for($i=0;$i<count($ret);$i++)
     if($ret[$i]!=NULL)
        $this->privmsg($this->config['chan']," : ".trim($ret[$i]));
break;

case "exec":
   $command = substr(strstr($msg,$mcmd[0]),strlen($mcmd[0])+1);
   $exec = exec($command);
   $ret = explode("\n",$exec);
   for($i=0;$i<count($ret);$i++)
     if($ret[$i]!=NULL)
        $this->privmsg($this->config['chan']," : ".trim($ret[$i]));
break;

case "passthru":
   $command = substr(strstr($msg,$mcmd[0]),strlen($mcmd[0])+1);
   $exec = passthru($command);
   $ret = explode("\n",$exec);
   for($i=0;$i<count($ret);$i++)
     if($ret[$i]!=NULL)
        $this->privmsg($this->config['chan']," : ".trim($ret[$i]));
break;
```

(рис. 3)

В ходе работы посредством анализа исходного кода одного из ботов были найдены уязвимости в аутентификации владельца ботнета. Аутентификация проходила только по нику в данном IRC-канале, без каких бы то ни было дополнительных проверок. Это предоставляет возможность во время отсутствия владельца в чате зайти в чат под его именем и осуществить перехват управлением ботнетом. Что, в свою очередь уже позволяет уничтожить ботнет.

Ботнет о котором идет речь состоял всего из приблизительно 30 машин, однако это доказывает возможность перехвата управления ботнетами.

Результаты и выводы

В процессе работы были выполнены:

- Обзор существующих типов уязвимостей веб приложений и выбор наиболее подходящего для данной цели
- Выбор и установка ряда веб приложений с наличием уязвимостей выбранного типа
- Конфигурирование сервера и настройка логирования всех обращений к нему
- Анализ результатов работы системы

Установленная система может быть использована как для сбора базы ботов, распространяемых через уязвимости веб приложений, и их анализа, так и для оценки характера и особенностей атак в целом.

В будущем планируется развитие и расширение системы. Ручной анализ исходных кодов ботов является утомительным, так же как и наблюдение за их поведением в IRC-каналах. Поэтому в будущем планируется разработка программы, способной автоматически извлекать необходимые данные из исходных кодов, эмулировать поведение ботов и автоматически собирать различие интересующие нас данные на многочисленных IRC-каналах. Помимо этого возможны более глубокие работы в области перехвата управления ботнетами.

Используемые источники

[1] SecureList. Ботнеты.
http://www.securelist.com/ru/analysis?pubid=204007610
[2] Sumit Siddharth, Pratiksha Doshi. Five common Web application vulnerabilities.
http://www.symantec.com/connect/articles/five-common-web-application-vulnerabilities
[3] e107 CMS v. 0.7.13. Имеющаяся уязвимость.
http://php-security.org/2010/05/19/mops-2010-035-e107-bbcode-remote-php-code-execution-vulnerability/index.html
[4] PhpMyAdmin v. 3.3.7. Имеющаяся уязвимость.
http://www.xaker.name/forvb/showthread.php?t=22371
[5] ImpressPages CMS v. 1.0.12. Имеющаяся уязвимость.
http://www.securityfocus.com/archive/1/521118
[6] phpBB v. 2.0.15. Имеющаяся уязвимость.
http://www.securiteam.com/exploits/5IP071PGBO.html
[7] WordPress Zingiri Web Shop Plugin v. 2.2.3. Имеющаяся уязвимость.http://www.securitylab.ru/vulnerability/410073.php
[8] WordPress WP-Syntax Plugin v. 0.9.8. Имеющаяся уязвимость.
http://www.securitylab.ru/vulnerability/384180.php?auth_service_id=VKontakte&auth_service_error=1&el_id=384180&sphrase_id=1160171
[9] Перевалова М.А., Колмогорцев Е.Н. Ханипоты, как средства анализа и защиты: современное состояние и опыт использования. – Межвузовская научно-практическая конференция «Актуальные проблемы организации и технологии защиты информации». СПбГУ ИТМО, 2011 г.