САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
Математико-механический факультет

Кафедра Системного Программирования

Попов Кирилл Владимирович

Мониторинг вредоносной активности через SSH

Курсовая работа

Научный руководитель:
ст. преподователь Зеленчук Илья

Санкт-Петербург
2013

Оглавление

Введение

SSH (англ. Secure SHell — «безопасная оболочка») [7] - сетевой протокол прикладного уровня, позволяющий производить удалённое управление операционной системой. Поэтому данный протокол используется вредоносными программами и хакерами для получения контроля над компьютером, с целью его включения в ботнет[2].

В своей работе я рассматриваю существующие средства анализа такой активности, формулирую требования необходимые для успешной работы подобных систем, а так же разрабатываю собственное OpenSource средство мониторинга SshMonitor.

Так же SshMonitor в данный момент работает на действующем honeypot'е [4]. В результате подведения промежуточных итогов были получены образцы различных программ используемых для заражения компьютера.

1. Алгоритм действий вредоносных программ

Для того, чтобы получить доступ к компьютеру, вредоносная программа или злоумышленник, как правило, пытается методом перебора угадать комбинацию логина и пароля любого из пользователей. В таблице(табл. 1) приведены наиболее часто встречающиеся комбинации логина и пароля, с которыми злоумышленники пытались авторизироваться в нашей системе.

Таблица 1: Самые популярные логины и пароли перебираемые злоумышленниками

Кол-во	Логин	Кол-во	Пароль
51741	root	1909	123456
1260	oracle	1672	password
1085	bin	1157	1234
1057	admin	957	12345
950	nagios	874	test
910	www	811	abc123
898	test	433	1

После успешной авторизации вредоносная программа пытается обнаружить, попала ли она на плохо сконфигурированную машину(поиск и использование которых является основной целью подобных программ), либо же её пустили на специально подготовленный honeypot[4]. В последнем случае программа немедленно прекращает свою работу.

Когда злоумышленник убедился в своей безопасности, происходит скачивание так называемого "exploit pack"'а - набора программ, использующих различные уязвимости в операционной системе, для повышения собственных привелегий. Это делается для того, чтобы повысить степень контроля зараженной машины и увеличить возможности по использованию её в своих целях.

Кроме "exploit pack"'а на зараженную машину загружается программа-клиент, которая по команде с коммандных серверов использует машину для одной из следующих целей:

- Заражение других компьютеров
- DDOS
- Рассылка E-mail спама
- Сбор личных данных пользователя

2. Анализ существующих honeypot'ов

2.1. Kojoney

- URL: http://kojoney.sourceforge.net/
- Язык реализации: Python/Perl
- Открытые исходные коды: Да
- Особенности: Умеет отличать людей от роботов. Это достигается за счет анализа частоты ввода символов, а так же налчия нажатий клавиши backspace.

Является одним из первых широко распространенных SSH Honeypot'ов. Реализуется как надстройка над обычным OpenSSH[6]. За счет этого многие вирусы умею определять, что попали на данный honeypot. Что подтверждается даже из демонстрационных логов, выложенных на сайте [1].

2.2. Kippo

- URL: http://code.google.com/p/kippo/
- Язык реализации: Python
- Открытые исходные коды: Да
- Особенности: Умеет эмулировать файловую систему, что, в свою очередь, минимизирует риск нанесения ущерба системе, на которой он развернут. Умеет сохранять логи в UML формате.

Разрабатывался под влияением Kojoney и во многом похож на него. Реализован по схожему принципу. Ввиду широкого распространения вирусы так же умеют детектировать его наличие на атакуемой системе.

2.3. Honeypot-ssh

- URL: http://code.google.com/p/honeypot-ssh/
- Язык реализации: C++
- Открытые исходные коды: Да
- Особенности: Умеет оповещать владельца honeypot'а через социальные сети(!) о попадании внутрь нового вируса. Умеет отличать бота от человека.

Данный honeypot использует собственную реализацию OpenSSH[6], за счет чего детектируется вирусами на атакуемой ими машине.

3. Формулирование требований для программы мониторинга

После проведенного анализа существующих решений были сформулированы следующие требования для программ осуществляющих мониторинг вредоносной активности через SSH:

- Не допустить использование исследуемого сервера в качестве узла атаки.
- Не допустить нанесение вреда серверу, на котором проводится исследование.
- Не допустить обнаружения вирусами средства мониторинга.
- Устанавливать политику логгирования отдально для каждой из группы пользователей.
- Открытый исходный код.

В связи с тем, что ни одно из рассмотренных решений не удовлетворяло полностью всем перечисленным критериям, было принято решение разработать собственную программу "SshMonitor"

4. Описание принципа работы SshMonitor

4.1. SshMonitor

Для проведения исследования была выбрана пара логин/пароль oracle/oracle. Если к OpenSSH[6] подключается пользователь с такой комбинацией логина и пароля, он помещается в chroot-окружение[3]

После этого сервером OpenSSH создается tty-абстракция[8] с которой связывается терминал только что подключивсегося пользователя. SshMonitor с помощью inotify[9] отслеживает появления новых файлов в /dev/pts/*.

После их появления происходит проверка какому пользователю они принадлежат. Если данный пользователь попадает в группу, для которой ведется мониторинг, то для только что созданного файла запускается утилита "ttylog"[10] которая записывает лог в нужную папку, сохраняя дату и время соединения.

4.2. Защита машины используемой для анализа

Чтобы помешать злоумышленникам атаковать с нас другие машины при помощи iptables[5] дропается весь исходящий трафик.

Кроме того, для предотвращения атак на сам компьютер, участвующий в исследованиях, из поставки OpenSSH убрана утилита scp. Так же, отключены такие функции OpenSSH, как туннелирование и поддержка sftp.

Заключение

В данный момент SshMonitor работает на боевом honeypot'е. Через неделю после его запуска уже был получен первый exploit pack. К настоящему времени собрано несколько вирусных программ для постоения ботнета. Параллельно с этим идет накопление статистических данных.

В дальнейшем планируется произвести более тонкую настройку chroot-окружения и Reverse engineering вирусов, которые уже к нам попали.

Список литературы

[1] Kojoney. Logs. — URL: http://kojoney.sourceforge.net/big-report.txt (online; accessed: 29.05.2013).

[2] Wikipedia. Botnet. — URL: http://en.wikipedia.org/wiki/Botnet (online; accessed: 29.05.2013).

[3] Wikipedia. Chroot. — URL: http://en.wikipedia.org/wiki/Chroot (online; accessed: 29.05.2013).

[4] Wikipedia. Honeypot. — URL: http://en.wikipedia.org/wiki/Honeypot_(computing) (online; accessed: 29.05.2013).

[5] Wikipedia. Iptables. — URL: http://en.wikipedia.org/wiki/Iptables (online; accessed: 29.05.2013).

[6] Wikipedia. OpenSSH. — URL: http://en.wikipedia.org/wiki/OpenSSH (online; accessed: 29.05.2013).

[7] Wikipedia. SSH. — URL: http://en.wikipedia.org/wiki/Secure_Shell (online; accessed: 29.05.2013).

[8] Wikipedia. TTY - абстракция. — URL: http://en.wikipedia.org/wiki/POSIX_terminal_interface (online; accessed: 29.05.2013).

[9] Wikipedia. inotify. — URL: http://en.wikipedia.org/wiki/Inotify (online; accessed: 29.05.2013).

[10] ttylog. — URL: http://search.cpan.org/~bbb/ttylog-0.83/ttylog (online; accessed: 29.05.2013).