25 Techniques to Gather Threat Intel and Track Actors

Wayne Huang, PhD
Sun Huang
How we got into this...

- BS, MS from CS, NCTU, Taiwan
- 4-years mandatory military service
  - Academia Sinica
  - Cyber defense against state-sponsored actors
  - PhD from EE, National Taiwan University
  - First paper: ACM WWW 2003, Budapest

- Cofounder & CEO to Armorize (2006-2013), acquired by Proofpoint (nasdaq: PFPT)
- Proofpoint VP Engineering 2013-2018 June
- Cofounder & CEO to Xrex Inc.
Agenda

- Showcase 25 methods for gathering threat intel for over 30 real cases
- Mostly against web-based C&C servers operated by actors
- WHY: Actors lousy secops, server misconfigurations, vulnerable panel code
- HOW: pentesting, application code review
- Intelligence gathering as a security research strategy
- Conclusion
Method 1

- The story starts with us getting a whole bunch of forensics URLs from our sandboxes...

<table>
<thead>
<tr>
<th>HTTP Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>URL</td>
</tr>
<tr>
<td><a href="http://nwhelicopters.com/steve/gate.php">http://nwhelicopters.com/steve/gate.php</a></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>DNS Requests</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hostname</td>
</tr>
</tbody>
</table>

... from these URLs, our investigation starts
Method 1 – Analytics beacons

- Nurjax (Superfish shipped by notebook vendor)
- Win: Discovery of openly accessible traffic analytics
Method 1 – Analytics beacons

- **Win:** Discovery of openly accessible traffic analytics
- **Nurjax** (Superfish shipped by notebook vendor)
Method 2 – Open directories

- Cryptowall
- Win: collect tools, source code, targets, type of c2 panels in use, and unseen samples

<table>
<thead>
<tr>
<th>Index of /up</th>
<th>Spam tool</th>
<th>Outlook email harvester</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1n.rar</td>
<td>13-Jan-2016</td>
<td>114M</td>
<td></td>
</tr>
<tr>
<td>2.rar</td>
<td>13-Jan-2016</td>
<td>191M</td>
<td></td>
</tr>
<tr>
<td>3.rar</td>
<td>13-Jan-2016</td>
<td>214M</td>
<td></td>
</tr>
<tr>
<td>4.rar</td>
<td>13-Jan-2016</td>
<td>198M</td>
<td></td>
</tr>
<tr>
<td>5.rar</td>
<td>13-Jan-2016</td>
<td>169M</td>
<td></td>
</tr>
<tr>
<td>from_emails.txt</td>
<td>27-Dec-2015</td>
<td>111K</td>
<td></td>
</tr>
<tr>
<td>iscode.txt</td>
<td>28-Dec-2015</td>
<td>3.9K</td>
<td></td>
</tr>
<tr>
<td>message.txt</td>
<td>27-Dec-2015</td>
<td>575K</td>
<td></td>
</tr>
<tr>
<td>new_config.txt</td>
<td>28-Dec-2015</td>
<td>52K</td>
<td></td>
</tr>
<tr>
<td>mm.rar</td>
<td>13-Jan-2016</td>
<td>187M</td>
<td></td>
</tr>
<tr>
<td>send.txt</td>
<td>27-Dec-2015</td>
<td>22K</td>
<td></td>
</tr>
<tr>
<td>send_scripts.txt</td>
<td>28-Dec-2015</td>
<td>753K</td>
<td></td>
</tr>
<tr>
<td>sendmail.rar</td>
<td>04-Jan-2016</td>
<td>1.2M</td>
<td></td>
</tr>
<tr>
<td>sub.txt</td>
<td>27-Dec-2015</td>
<td>102</td>
<td></td>
</tr>
<tr>
<td>utils.txt</td>
<td>27-Dec-2015</td>
<td>0</td>
<td></td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Index of /outlook/reports</th>
</tr>
</thead>
</table>

<table>
<thead>
<tr>
<th>Name</th>
<th>Last modified</th>
<th>Size</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parent Directory</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>0/</td>
<td>14-Dec-2015</td>
<td>18:40</td>
<td></td>
</tr>
<tr>
<td>Canada(CA)</td>
<td>14-Dec-2015</td>
<td>21:42</td>
<td></td>
</tr>
<tr>
<td>France(FR)</td>
<td>14-Dec-2015</td>
<td>14:07</td>
<td></td>
</tr>
<tr>
<td>Spain(ES)</td>
<td>14-Dec-2015</td>
<td>08:47</td>
<td></td>
</tr>
<tr>
<td>United Kingdom(GB)</td>
<td>15-Dec-2015</td>
<td>08:02</td>
<td></td>
</tr>
<tr>
<td>United States(US)</td>
<td>14-Dec-2015</td>
<td>16:34</td>
<td></td>
</tr>
<tr>
<td>totalstat.txt</td>
<td>17-Dec-2015</td>
<td>01:37</td>
<td>3</td>
</tr>
</tbody>
</table>
Method 2 – Open directories

- Dridex 120: targeting UK
Method 3 – Fuzzing common file names against C2

- Nurjax - stats.php
- Win: Discover C2 functionalities

<p>| | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Total Distribuido</strong>:</td>
<td>6080063</td>
</tr>
<tr>
<td><strong>Instalados Hoje</strong>:</td>
<td>401</td>
</tr>
<tr>
<td><strong>Ativos Hoje</strong>:</td>
<td>3581</td>
</tr>
<tr>
<td><strong>Total Ativo (15 dias)</strong></td>
<td>238780</td>
</tr>
<tr>
<td><strong>Total Ativo (3 dias)</strong></td>
<td>114752</td>
</tr>
<tr>
<td><strong>Total Ativo (2 dias)</strong></td>
<td>95857</td>
</tr>
</tbody>
</table>

**Ultimos 30 dias**
- Data 13/04/2016 : 401 installs
- Data 12/04/2016 : 6546 installs
- Data 11/04/2016 : 7477 installs
- Data 10/04/2016 : 6615 installs
- Data 09/04/2016 : 6805 installs
- Data 08/04/2016 : 6927 installs
- Data 07/04/2016 : 6570 installs
- Data 06/04/2016 : 5135 installs
- Data 05/04/2016 : 5403 installs
- Data 04/04/2016 : 5911 installs
- Data 03/04/2016 : 7297 installs
- Data 02/04/2016 : 7677 installs
- Data 01/04/2016 : 6597 installs
- Data 31/03/2016 : 6011 installs
Method 3 – Fuzzing common file names against C2

- UnkDownloader (targeting brazil)
- Win: C2 panel access
Method 3 – Fuzzing common file names against C2

- UnkDownloader (targeting brazil)
- Win: C2 panel access
Method 3 – Fuzzing common file names against C2

- Loki Stealer (Pony) linux.php
- Win: C2 server access via webshells
Method 4 – C2 Apache server-status

- Cryptowall
- Win: Find C2 admin login panels via the Apache server-status module

<table>
<thead>
<tr>
<th>Srv</th>
<th>PID</th>
<th>Acc</th>
<th>M</th>
<th>CPU</th>
<th>SS</th>
<th>Req</th>
<th>Conn</th>
<th>Child</th>
<th>Slot</th>
<th>Client</th>
<th>Request</th>
</tr>
</thead>
<tbody>
<tr>
<td>0-1</td>
<td>24684</td>
<td>04307/49793</td>
<td>_</td>
<td>1242.38</td>
<td>6</td>
<td>4</td>
<td>0.0</td>
<td>0.28</td>
<td>24.83</td>
<td>127.0.0.1</td>
<td>localhost POST/z5mh28ar9v HTTP/1.0</td>
</tr>
<tr>
<td>1-1</td>
<td>-</td>
<td>0/45533</td>
<td>_</td>
<td>651</td>
<td>44312</td>
<td>0</td>
<td>0.0</td>
<td>0.00</td>
<td>24.05</td>
<td>::1</td>
<td>localhost OPTIONS * HTTP/1.0</td>
</tr>
<tr>
<td>2-1</td>
<td>24696</td>
<td>04323/49828</td>
<td>_</td>
<td>1144.30</td>
<td>3</td>
<td>4</td>
<td>0.0</td>
<td>0.25</td>
<td>23.90</td>
<td>127.0.0.1</td>
<td>localhost POST/9ymmqd560arn3k HTTP/1.0</td>
</tr>
<tr>
<td>3-1</td>
<td>24687</td>
<td>04314/49753</td>
<td>_</td>
<td>1238.82</td>
<td>6</td>
<td>3</td>
<td>0.0</td>
<td>0.09</td>
<td>23.92</td>
<td>127.0.0.1</td>
<td>localhost POST/18yswth97tu HTTP/1.0</td>
</tr>
<tr>
<td>4-1</td>
<td>24688</td>
<td>04331/49837</td>
<td>_</td>
<td>1200.11</td>
<td>8</td>
<td>4</td>
<td>0.0</td>
<td>0.14</td>
<td>23.97</td>
<td>127.0.0.1</td>
<td>localhost POST/7yswzrp974j HTTP/1.0</td>
</tr>
<tr>
<td>5-1</td>
<td>24689</td>
<td>04330/47746</td>
<td>_</td>
<td>1157.84</td>
<td>4</td>
<td>3</td>
<td>0.0</td>
<td>0.13</td>
<td>23.54</td>
<td>127.0.0.1</td>
<td>localhost POST/5a114arm2410 HTTP/1.0</td>
</tr>
<tr>
<td>6-1</td>
<td>24690</td>
<td>04336/49854</td>
<td>_</td>
<td>1107.58</td>
<td>5</td>
<td>254</td>
<td>0.0</td>
<td>0.24</td>
<td>24.27</td>
<td>127.0.0.1</td>
<td>localhost POST/workngwwork_a01951b4c4ea9925ae188ae71d6a45a3?p=statisti</td>
</tr>
<tr>
<td>7-1</td>
<td>-</td>
<td>0/45461</td>
<td>_</td>
<td>0.01</td>
<td>44342</td>
<td>0</td>
<td>0.0</td>
<td>0.00</td>
<td>24.10</td>
<td>::1</td>
<td>localhost OPTIONS * HTTP/1.0</td>
</tr>
<tr>
<td>8-1</td>
<td>11395</td>
<td>02133/1/50027</td>
<td>_</td>
<td>5223.45</td>
<td>1</td>
<td>3</td>
<td>0.0</td>
<td>3.23</td>
<td>24.78</td>
<td>127.0.0.1</td>
<td>localhost POST/27?pxp0n76zl HTTP/1.0</td>
</tr>
<tr>
<td>9-1</td>
<td>-</td>
<td>0/45457</td>
<td>_</td>
<td>3899.33</td>
<td>45149</td>
<td>0</td>
<td>0.0</td>
<td>0.00</td>
<td>23.49</td>
<td>::1</td>
<td>localhost OPTIONS * HTTP/1.0</td>
</tr>
<tr>
<td>10-1</td>
<td>-</td>
<td>0/45359</td>
<td>_</td>
<td>113.85</td>
<td>45102</td>
<td>0</td>
<td>0.0</td>
<td>0.00</td>
<td>23.40</td>
<td>::1</td>
<td>localhost OPTIONS * HTTP/1.0</td>
</tr>
</tbody>
</table>
| 11-1 | 24562 | 04524/49990 | W | 1275.10 | 1 | 0 | 0.0 | 0.18 | 23.95 | 127.0.0.1 | localhost POST /9am77nm9eq HTTP/1.0 }
Method 4 – C2 Apache server-status

- Loki PWS Stealer (Pony) + LOKI PLUS (Neutrino)
Method 4 – C2 Apache server-status

- Loki PWS Stealer (Pony) + LOKI PLUS (Neutrino)
Method 4 – C2 Apache server-status

- Loki PWS Stealer (Pony) + LOKI PLUS (Neutrino)
Method 5 – PHP error messages

- TROJAN Unknown Bot
- Win: Learned structure, found control panel
Method 5 – PHP error messages

- TROJAN Unknown Bot
- Win: Learned structure, found control panel

```
2016-03-24 14:22:10
```

```
Warning: copy/class_database.php): failed to open stream: Permission denied on line 25
Warning: copy/koneksi.php): failed to open stream: Permission denied
Warning: copy/getlocation.php): failed to open stream: Permission denied
Warning: copy/userstatus.php): failed to open stream: Permission denied
Warning: copy/proses.php): failed to open stream: Permission denied
Warning: copy/confirm.php): failed to open stream: Permission denied
Warning: copy/sdk.php): failed to open stream: Permission denied
Warning: copy/index.php): failed to open stream: Permission denied
Warning: copy/data.php): failed to open stream: Permission denied
Warning: copy/config.php): failed to open stream: Permission denied
Warning: copy/read.php): failed to open stream: Permission denied
Warning: copy/index.html): failed to open stream: Permission denied
```
Method 6 – Python Django debug enabled

- Asprox: Marketplace with 1400+ registered sellers
  Win: Learned everything about them
Method 6 – Python Django debug enabled

- WSO Webshells by unique domain: 3,027,423
  - gov:602+, mil:7+
- WSO Webshells by unique filename: 7,966,903
- SMTP accounts: 2,136,017
  - gov:4,000+, mil:1,574+ (Over 1,220 one military department)
- FTP accounts: 585,549
  - gov:258
- SSH-root: 1,236
- SSH-user: 50,757
  - gov:92
Methods 1-6 summary

Our exploration starts with sandbox-extracted URLs...

- Method 1 – Analytics beacons
- Method 2 – Open directories
- Method 3 – Fuzzing filenames
- Method 4 – Apache server-status
- Method 5 – PHP error messages
- Method 6 – Python Django debug enabled
Progress: now we know the C2’S file structure...

... But how can we get authenticated into the panels?
Method 7 – Weak passwords

- Blackmoo_KRBanker (Targeting Korea)
- Win: C2 panel access
Method 7 – Weak passwords

- Blackmoo_KRBanker (Targeting Korea)
- Win: C2 panel access

<table>
<thead>
<tr>
<th>MAC信息</th>
<th>操作系统</th>
<th>IP地址</th>
<th>首次安装时间</th>
<th>最后访问时间</th>
<th>版本号</th>
</tr>
</thead>
<tbody>
<tr>
<td>00-00-33-34-09</td>
<td>Windows XP</td>
<td>95.78.12.207</td>
<td>2016-03-23 9:16:57</td>
<td>2016-03-25 3:34:02</td>
<td>1.3</td>
</tr>
<tr>
<td>00-19-33-30-07-3C</td>
<td>Windows XP</td>
<td>118.1.1.245</td>
<td>2016-03-23 4:13:32</td>
<td>2016-03-25 3:58:47</td>
<td>1.3</td>
</tr>
<tr>
<td>00-64-35-07-3C</td>
<td>Windows XP</td>
<td>217.1.1.95</td>
<td>2016-03-25 2:30:00</td>
<td>2016-03-25 3:29:14</td>
<td>1.3</td>
</tr>
<tr>
<td>09-00-30-30-30</td>
<td>Windows XP</td>
<td>208.1.1.39</td>
<td>2016-03-25 3:29:33</td>
<td>2016-03-25 3:40:55</td>
<td>1.0</td>
</tr>
<tr>
<td>00-1A-30-30-07-BB</td>
<td>Windows XP</td>
<td>59.1.1.112</td>
<td>2016-03-25 3:38:34</td>
<td>2016-03-25 3:39:04</td>
<td>1.3</td>
</tr>
</tbody>
</table>
Method 7 – Weak passwords

- AZORult V2
- Win: C2 panel access

Password:
Method 7 - Weak passwords

- AZORult V2
- Win: C2 panel access
Method 8 – Hardcoded passwords / download config file

- IRC bot (not well known)
- Win: FTP root access

```plaintext
bWEtm9uZ5jb20=
Ym1AbWVpZGEtem9uZ5jb20=
SmVpFpbmUyMw==
aXaJsaXR6ZWQub3Jn
NjA==
I2J5Ng==
Z2==
Nj==
Cr: cene
```
Method 8 – Hardcoded passwords / download config file

- IRC bot (not well known)
- Win: FTP root access
Method 9 – Insufficient authentication

- Hancitor_Downloader
- Win: C2 panel access
Method 9 – Insufficient authentication

- Hancitor_Downloader
- Win: C2 panel access
Method 9 – Insufficient authentication

- Android Marcher malware
- Win: C2 panel access
Method 9 – Insufficient authentication

- Android Marcher malware
- Win: C2 panel access
Now that we’ve authenticated ourselves...

... can we expand laterally?
Method 10 – Session Fixation

- Keybase (mostly operated by Nigerian actor)
  - Also has SQL injection, file upload vulnerabilities
- Win: access others panel on the same C2 server without authentication

```php
if (!isset($_SESSION['logged_in']))
    header('Location: login.php');
exit;
```
Method 10 – Session Fixation

KeyBase

Admin Dashboard
Welcome, love to see you back.

<table>
<thead>
<tr>
<th>14</th>
<th>3058</th>
<th>1359</th>
<th>0</th>
</tr>
</thead>
<tbody>
<tr>
<td>Computers</td>
<td>Keystrokes</td>
<td>Passwords</td>
<td>Screenshots</td>
</tr>
</tbody>
</table>

Notifications from Computers

<table>
<thead>
<tr>
<th>Machine Name</th>
<th>Machine Time</th>
<th>IP Address</th>
<th>Date</th>
</tr>
</thead>
<tbody>
<tr>
<td>JMKRRK-MA</td>
<td>3:24 PM</td>
<td>20:177</td>
<td>2015-09-10 22:24:45</td>
</tr>
<tr>
<td>SHNITH</td>
<td>6:44 AM</td>
<td>41:2.58</td>
<td>2015-09-11 05:44:11</td>
</tr>
<tr>
<td>RM3H</td>
<td>2:13 PM</td>
<td>41:3.232</td>
<td>2015-09-11 10:12:57</td>
</tr>
<tr>
<td>JG3RETH</td>
<td>7:34 AM</td>
<td>70:4.81</td>
<td>2015-09-11 11:34:48</td>
</tr>
</tbody>
</table>
Method 10 – Session Fixation

- Keybase (mostly operated by Nigerian actor)
- Targeted business email compromise (BEC)
Methods 7-10 summary

Authentication and lateral expansion

- Method 7 – Weak passwords
- Method 8 – Hardcoded passwords
- Method 9 – Insufficient authentication
- Method 10 – Session fixation
Finding more vulnerabilities...

... via code review
Method 11 – Obtain source code

- Goal: obtain panel’s source code and review, learn panel structure
- Fuzz folder-based archive names
  - /bn/ -> bn.zip / bn.rar / bn.tar.gz
  - /panel/ -> panel.zip / panel.rar / panel.tar.gz
- Custom fuzzer script: collect all C2 URLs then try to fuzz
Method 12 – Cross site scripting

- ISR stealer
- Win: steal cookie and access C2 panel

```php
if ($total > 0) {
    $result = mysql_query("SELECT * FROM `logs` ORDER BY `date` ":$-_SESSION['order'] . " LIMIT ".($logspage*$-_SESSION["page"]).", ..$logspage; ");
    $i = 0;
    $fetched = mysql_num_rows($result);
    while ($row = mysql_fetch_assoc($result))
    {
        $class = ($i % 2 == 0) ? "ai" : "hi";
        echo '<tr class="'.$class.'"">
            <td style="width:5px;"><input type="checkbox" name="sel[]" value="".'.$row['id']."" />
            <td style="width:10%;">' . $row['app'] . '</td>
            <td style="width:25%;">' . $row['url'] . '</td>
            <td style="width:15%;">' . $row['username'] . '</td>
            <td style="width:13%;">' . $row['password'] . '</td>
            <td style="width:8%;">' . $row['pcname'] . '</td>
            <td style="width:12%;">' . $row['ip'] . '</td>
            <td>' . $row['date'] . '</td>
        </tr>
    }
    $i++;
```
Method 12 – Cross site scripting

- ISR stealer
- Win: steal cookie and access C2 panel
<table>
<thead>
<tr>
<th><strong>XSS targeted experiment</strong></th>
<th>170 ISR Stealer panels on unique domain name</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Duration</strong></td>
<td>2 weeks</td>
</tr>
<tr>
<td><strong>Successful trigger</strong></td>
<td>Received 103 Cookies</td>
</tr>
<tr>
<td><strong>Successful rate</strong></td>
<td>60 %</td>
</tr>
<tr>
<td><strong>Number of victims</strong></td>
<td>66,284</td>
</tr>
<tr>
<td><strong>Actors location</strong></td>
<td>Mostly in Nigeria</td>
</tr>
</tbody>
</table>
Method 13 – Hidden backdoors

- Win: gain C2 server access
- Zeus Robot / Panther / GOZ
Method 14 – Remote command execution vulnerability

- Zeus / Citadel / ICEXI
- Win: root the C2 server
Method 14 – Remote command execution vulnerability
Method 14 – Remote command execution vulnerability

PHP Version 5.4.32

System: Linux server.cyber-node.bp2.org 2.6.32-431.29.2.el6.x86_64 #1 SMP Tue Sep 9 21:36:05 UTC 2009 x86_64
Build Date: 2010-04-05 19:18:05
Server API: CGI/FastCGI
Virtual Directory Support: disabled
Configuration File (php.ini): /usr/local/lib
Path: /home/owldon/public_html/index.php
Method 15 – SQL Injection

- Android Opfake malware
- Win: dump C2 panel’s database

```php
<?php
include ("config.php");
mysql_query ("SET NAMES 'utf8'");

$ip = $_SERVER['REMOTE_ADDR'];

$IMEI = $_GET['imei'];

$balance = $ip;

$query = mysql_query("UPDATE list SET balance='{$balance}' LIMIT 1", $db) or die(mysql_error());

$query_2 = mysql_query("SELECT * FROM list WHERE IMEI='{$IMEI}'", $db) or die(mysql_error());

$rows = array();
```
Method 15 – SQL Injection

- Android Opfake malware
- Win: dump C2 panel’s database

```
<?php
include (“config.php”);
mysql_query (“SET NAMES ‘utf8’”);
```

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 "" LIMIT 1" at line 1

```
10 $query = mysql_query("UPDATE list SET balance='$balance' LIMIT 1", $db) or die(mysql_error());
WHERE IMEI='$IMEI'
11 $query_2 = mysql_query("SELECT * FROM list WHERE IMEI='$IMEI'", $db) or die(mysql_error());
13 $rows = array();
```
Method 15 - SQL Injection

- Android Opfake malware
- Win: dump C2 panel’s database
Methods 11-15 summary

Finding vulnerabilities via code review

- Method 11 – Obtain source code
- Method 12 – Cross site scripting
- Method 13 – Hidden backdoors
- Method 14 – Remote command execution
- Method 15 – SQL injection
Having admin panel access and webshell access is GREAT...

... but how about rooting the server?
Method 16 - Remote command execution

- HFS -- Vawtrak hosting payload for TinyLoader
- Win: root the C2 server
Method 17 – Shellshock (CVE-2014-6271)

- Sutra TDS – undisclosed
- Win: gain C2 server access

OS:
Linux 3.1.3 #1 SMP Mon Nov 28 00:18:51 MSK 2011 i686 i686 i386 CNU/I

path:
/var/www/.../data/www/google.com

user id:
uid=500 gid=502 groups=501,502

Environment:
SERVER_SIGNATURE="<address>Apache/2.2.23 (CentOS) Server at google.com Port 80</address>

HTTP_USER_AGENT=Mozilla/5.0 (Windows NT 5.1; rv:43.0) Gecko/20100101 Firefox/43.0
HTTP_X_FORWARDED_FOR=
SERVER_PORT=80
HTTP_HOST=google.com
Method 17 – Shellshock (CVE-2014-6271)

- Sutra TDS - undisclosed
- Win: gain C2 server access

```bash
/bin/curl -A "<(> ( ; ); /sbin/ifconfig -a" http://....
```

```
eth1    Link encap:Ethernet  HWaddr E4:...:B5  
BROADCAST     MULTICAST MTU:1500 Metric:1  
RX packets:0  errors:0  dropped:0  overruns:0  frame:0  
TX packets:0  errors:0  dropped:0  overruns:0  carrier:0  
collisions:0  txqueuelen:1000  
RX bytes:0  (<0.0 b)  TX bytes:0  (<0.0 b)  
Interrupt:17  Memory:...
```

```
lo     Link encap:Local Loopback  
inet addr:127.0.0.1  Mask:255.0.0.0  
inetc addr: ::1/128  Scope:Host  
UP LOOPBACK RUNNING MTU:16436 Metric:1  
RX packets:1  errors:0  dropped:0  overruns:0  frame:0  
TX packets:1  errors:0  dropped:0  overruns:0  carrier:0  
collisions:0  txqueuelen:0  
RX bytes:213174066  (203.2 MiB)  TX bytes:213174066  (203.2 MiB)
```
Method 18 – JAVA Unserialize Vulnerability (CVE-2015-4852)

- Android Fake-Angry
  - Oracle WebLogic Server, versions 10.3.6.0, 12.1.2.0, 12.1.3.0, 12.2.1.0 are affected
- Win: gain access C2 server
Now that we can execute arbitrary commands and access arbitrary files…

… how to very quickly grasp what’s there?
Method 19 – Webalizer/AWStat

- Northern Gold (Qbot)
- Win: Understanding a C2’s structure

<table>
<thead>
<tr>
<th>#</th>
<th>Hits</th>
<th>KBytes</th>
<th>URL</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>2142342</td>
<td>732766</td>
<td>/t</td>
<td>Qbot gate</td>
</tr>
<tr>
<td>2</td>
<td>1306853</td>
<td>716881</td>
<td>/k</td>
<td>Exploits go to sutra</td>
</tr>
<tr>
<td>3</td>
<td>240434</td>
<td>69783581</td>
<td>/v</td>
<td>Qbot exe updates</td>
</tr>
<tr>
<td>4</td>
<td>72215</td>
<td>20450287</td>
<td>/u/_qbotinj.exe</td>
<td>Qbot exe updates</td>
</tr>
<tr>
<td>5</td>
<td>12981</td>
<td>1121722</td>
<td>/w</td>
<td>Webinjests for all</td>
</tr>
<tr>
<td>6</td>
<td>12912</td>
<td>2420</td>
<td>/s</td>
<td>Session spy</td>
</tr>
<tr>
<td>7</td>
<td>5259</td>
<td>1336859</td>
<td>/u/_qbotinj.exe.pkg</td>
<td>Qbot exe updates</td>
</tr>
<tr>
<td>8</td>
<td>2010</td>
<td>6220</td>
<td>/E/J2.JS</td>
<td>Inject Iframe redirection</td>
</tr>
<tr>
<td>9</td>
<td>1825</td>
<td>1822</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>1522</td>
<td>408</td>
<td>/robots.txt</td>
<td></td>
</tr>
</tbody>
</table>
Methods 16-19 summary

Rooting the server & quickly overviewing data

- Method 16 – Remote command execution
- Method 17 – Shellshock
- Method 18 – Java unserialized vulnerability
- Method 19 – Webalizer / AWStat
Let’s try some complex techniques!
Method 20 – Path traversal

- MagikPOS
- Win: arbitrary file access, control panel access

```php
<?php

class settings {
    // db
    const db_hostname = "localhost";
    const db_user = "root";
    const db_password = "[REDACTED]";
    const db_name = "02kFgp";

    // account
    const umb_username = "Magic";
    const umb_password = "[REDACTED]";

    // platform

    // how many times a user can send bad authentication details
    const brute_ipban = 5;

    // login session duration in seconds
    const sessionTime = "3600";

    // folder names
    const umb_logsPath = "logs";
    const umb_updatePath = "$_updates";

    // encryption key
    const enc_key = "[REDACTED]";
}
```
Method 20 – Path traversal

- MagikPOS
- Win: arbitrary file access, control panel access
Method 21 – File upload vulnerability (unrestricted)

- Jahoo spambot
- Win: arbitrary file access
Method 21 – File upload vulnerability (unrestricted)

- Win: arbitrary file access
- Jahoo spambot
Method 21 – File upload vulnerability (unrestricted)

- Jahoo spambot
- Win: arbitrary file access
Method 22 – File upload vulnerability (Satisfy prerequisites)

- Neutrino HTTP Bot (0day)
- Win: arbitrary file access
Method 23 – File upload vulnerability via C2 communication

- Gaudox Bot (0day)
  - Hardcoded RC4 encryption key
- Win: arbitrary file access

```
.text:00408D11  mov        eax, dword_41015C
.text:00408D16  mov        dword_411464, eax
.text:00408D1B  mov        eax, dword_410160
.text:00408D20  mov        dword_411468, eax
.text:00408D25  mov        eax, dword_410164
.data:0041015C  dword_41015C  dd 4512A7E5h ; DATA XREF: sub_4071C0+179\r
.data:0041015C  dword_410160  dd 696665BDh ; sub_4071C0+18D\r
.data:00410160  dword_410164  dd 2299FA23h ; DATA XREF: sub_408790+50D\r
.data:00410164  dword_410168  dd 9A7D779h ; sub_408790+58B\r
.data:00410168  dword_410168  dd 9A7D779h ; DATA XREF: sub_408790+595\r
```

```
Method 23 – File upload vulnerability via C2 communication

- Gaudox Bot (0day)
Method 23 - File upload vulnerability via C2 communication

- Gaudox Bot (0day)

```php
if(isset($_POST["src"])) {
    $ImgBytes = pack("H*", $_POST["src"]);
    if(chmod("screenshots", 0777)) {
        $Image = fopen("screenshots/" . $ClientId . ".jpeg", "w") ;
        if($Image) {
            fwrite($Image, $ImgBytes);
            fclose($Image);
        }
    }
    chmod("screenshots", 0755);
}
```
Method 23 – File upload vulnerability via C2 communication

- Gaudox Bot (0day)
Method 23 – File upload vulnerability via C2 communication

▸ Gaudox Bot (0day)
Methods 20-23 summary

Complex techniques

- Method 20 – Path traversal
- Method 21 – File upload vuln (unrestricted)
- Method 22 – File upload vuln (satisfy prereq)
- Method 23 – File upload vuln (C2 comm)
How about the C2 server’s domain?
Method 24 – Set-cookie

- Northern Gold
- Leverage: identify the actual C2 domains behind Nginx-based proxies
Method 25

How about the C2 server’s IP?
Method 25 – PHPinfo

- Dridex 120
- Win: Pinpoint C2 IPs
  - Many actors enable PHPinfo
  - Pinpoint C2 IPs from Nginx reverse proxies
Conclusion

- These techniques help build knowledge of:
  - Who’s behind the campaign
  - Who’s being targeted
  - Actor infrastructure and tools in use
  - Unreleased malware
  - Actor operation and strategies

- Most C2 panels contain vulnerabilities
Wayne Huang,
Sun Huang
wayne@xrex.io  sun@xrex.io

Telegrams:  @huangwayne
            @huangsun

Twitter  @1nf0s3cpt