TURNING LEGAL WEBSITE into DDoS TOOL

OWASP Jakarta Tech Day Meetup
Kalpin Erlangga Silaen

OWASP
The Open Web Application Security Project
• Segala cara, tehnik, peragaan serta alat yang digunakan dalam sesi presentasi ini adalah untuk tujuan Pendidikan

• Penyalahgunaan dari sebagian atau keseluruhan cara, tehnik, peragaan, serta alat yang ditunjukkan dalam sesi presentasi ini diluar tanggung jawab instruktur/penulis.
About Me

- Graduated of Master Information Technology, Swiss German University, 2016.
- Have been experience working with Solaris, FreeBSD, RedHat Linux, Slackware, SuSe since 1998.
- More than 8 years experience for penetration test project and digital forensic investigation.
• Some legitimate websites can be used to retrieve contents from other websites
• Those legitimate websites does not have sufficient control for the respective features above
• The features of legitimate websites can be abused to launch Denial of Service (DoS) Attack toward other websites
The growth of web application and user in the Internet, number of attacks also increased in terms of size and frequency in internet such as denial of services (DoS) (Arora et al., 2011)

Source: http://www.internetlivestats.com/
OWASP
The Open Web Application Security Project
Threat Analysis social media
Facebook

Threat Analysis web online translator

Start

User put URL to translate

Request Received by web online translator

Web online translator fetch content

Translated each component

Rearrange after translation

Provide web translated content to user

Stop

Generate Unique ID

Is API's key expire?

Yes

No

Attacker Create Facebook's API

Attacker put the URL of victim

Looping

Facebook fetch the content

Stop

Is victim down?

Yes

No
METHODOLOGY (REAL)

Real World Simulation Attack

1. Start
2. Prepare victim as target
3. Prepare monitoring
4. Repeat 5 times
   - Open Legal Website
   - Put victim's URL as a target
5. Send request to legal website
6. Stop

OWASP
The Open Web Application Security Project

Diagram:
- Attackers in Las Vegas, US
- Zombie-4 Firefox+ReloadEvery
- Put victim's URL into Facebook
- N Locations
- Zombie 1..N
- Their own datacenter
- Bing's Translator Server Farm
- Google's Translator Server Farm
- Facebook Translator Server Farm
- Wordpress 1..N
- Internet
- Victim Wordpress Monitoring
- Jakarta

THE HONEYNET PROJECT
### HARDWARE SPECIFICATION

<table>
<thead>
<tr>
<th>Functions</th>
<th>CPU</th>
<th>Memory</th>
<th>HDD</th>
</tr>
</thead>
<tbody>
<tr>
<td>Victim</td>
<td>1 x E3-1230 v3 @ 3.30GHz</td>
<td>512 MB</td>
<td>20 GB</td>
</tr>
<tr>
<td>Victim’s Monitoring</td>
<td>1 x E3-1230 v3 @ 3.30GHz</td>
<td>512 MB</td>
<td>20 GB</td>
</tr>
<tr>
<td>Wordpress-1</td>
<td>1 x CPU E5-2630L v2 @ 2.40GHz</td>
<td>512 MB</td>
<td>20 GB</td>
</tr>
<tr>
<td>Wordpress-2</td>
<td>1 x E5-2660 0 @ 2.20GHz</td>
<td>1 GB</td>
<td>30 GB</td>
</tr>
<tr>
<td>Wordpress-3</td>
<td>1 x @ 3.60 GHz</td>
<td>738 MB</td>
<td>16 GB</td>
</tr>
<tr>
<td>Wordpress-4</td>
<td>2 x E3-1241 v3 @ 3.50GHz</td>
<td>512 MB</td>
<td>16 GB</td>
</tr>
<tr>
<td>Wordpress-5</td>
<td>1 x E5520 @ 2.27GHz</td>
<td>512 MB</td>
<td>20 GB</td>
</tr>
<tr>
<td>Monitoring Attacker</td>
<td>1 x E5-2630L v2 @ 2.40GHz</td>
<td>1 GB</td>
<td>30 GB</td>
</tr>
<tr>
<td>Zombie-1</td>
<td>1 x E5-2630L v2 @ 2.40GHz</td>
<td>512 MB</td>
<td>20 GB</td>
</tr>
<tr>
<td>Zombie-2</td>
<td>1 x @ 3.60 GHz</td>
<td>738 MB</td>
<td>16 GB</td>
</tr>
<tr>
<td>Zombie-3</td>
<td>1 x E5-2650 @ 2.00GHz</td>
<td>1 GB</td>
<td>30 GB</td>
</tr>
<tr>
<td>Zombie-4</td>
<td>4 x @ 2.00GHz</td>
<td>1 GB</td>
<td>60 GB</td>
</tr>
</tbody>
</table>
- The victim's server always rebooted before launch the next attack
- Testing repeat 5 times for each scenario
- Result in average
- Monitoring's server is close to the victim
• Traffic bandwidth in Kilobit per second (Kbps) and Packet per Second (PPS)
• CPU and Memory Usage in Megabyte (MB)
• MySQL per Second (MQPS)
• HTTP Response in milisecond (ms)
• Ping Time Response in milisecond (ms)
NetRange: 173.252.64.0 - 173.252.127.255
CIDR: 173.252.64.0/18
NetName: FACEBOOK-INC
NetHandle: NET-173-252-64-0-1
Parent: NET173 (NET-173-0-0-0-0)
NetType: Direct Assignment
OriginAS: AS32934

From IP whois, we found that the IPs retrieved content from the victim is belong to Facebook
NetRange: 66.249.64.0 - 66.249.95.255
CIDR: 66.249.64.0/19
NetName: GOOGLE
NetHandle: NET-66-249-64-0-1
Parent: NET66 (NET-66-0-0-0-0)
NetType: Direct Allocation

From IP whois, we found that the IPs retrieved content from the victim is belong to Google
From IP whois, we found that the IPs retrieved content from the victim is belong to Google
| IP Address | Timestamp   | Request Method | Response Code | Referrer
|------------|-------------|----------------|---------------|--------
| 120.89.92.8 | [28/Nov/2015:07:32:45 +0700] | GET / HTTP/1.0 | 200 39587 | "WordPress/4.3.1; http://www.pprrlgroup.com; verifying pingback from 108.61.199.179"
| 120.89.92.8 | [28/Nov/2015:07:32:46 +0700] | GET / HTTP/1.0 | 200 39587 | "WordPress/3.2; http://gatiningsih.staff.ipdn.ac.id"
| 120.89.93.149 | [28/Nov/2015:07:32:46 +0700] | GET / HTTP/1.0 | 200 39587 | "WordPress/3.6.1; http://www.ppcindo.com/blog"
| 120.89.92.8 | [28/Nov/2015:07:32:47 +0700] | GET / HTTP/1.0 | 200 39587 | "WordPress/4.2.5; http://www.sembadapangan.com; verifying pingback from 119.81.1.178"
| 120.89.93.149 | [28/Nov/2015:07:32:48 +0700] | GET / HTTP/1.0 | 200 39587 | "WordPress/3.6.1; http://www.ppcindo.com/blog"
| 120.89.92.8 | [28/Nov/2015:07:32:49 +0700] | GET / HTTP/1.0 | 200 39587 | "WordPress/4.3.1; http://www.pprrlgroup.com; verifying pingback from 108.61.199.179"
| 120.89.92.8 | [28/Nov/2015:07:32:49 +0700] | GET / HTTP/1.0 | 200 39587 | "WordPress/3.2; http://gatiningsih.staff.ipdn.ac.id"
| 120.89.93.149 | [28/Nov/2015:07:32:49 +0700] | GET / HTTP/1.0 | 200 39587 | "WordPress/3.6.1; http://www.ppcindo.com/blog"

IP list above is the IP of CMS Wordpress as reflector
RESULTS (GRAPH)

Average Bandwidth in KBPS
Google Translator - Single Thread

Average Bandwidth in KBPS
Bing Translator - Single Thread

Average Bandwidth in Kbps
CMS Wordpress Ping Back - 1 Thread Simulated per second

Average Bandwidth in Kbps
CMS Wordpress Ping Back - 5 Threads Real Attack per second
<table>
<thead>
<tr>
<th>Name of Test</th>
<th>Bandwidth (in KBPS)</th>
<th>PPS</th>
<th>MQPS (Select)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>UP (Min)</td>
<td>Down (Min)</td>
<td>UP (Avg)</td>
</tr>
<tr>
<td>Normal Without Attack</td>
<td>7.44</td>
<td>6.67</td>
<td>7.51</td>
</tr>
<tr>
<td>Facebook Real Attack</td>
<td>85.85</td>
<td>20.18</td>
<td>191.67</td>
</tr>
<tr>
<td>Facebook Lab Simulation</td>
<td>91.81</td>
<td>9.02</td>
<td>1810.93</td>
</tr>
<tr>
<td>Google Real Attack</td>
<td>160.09</td>
<td>13.70</td>
<td>377.00</td>
</tr>
<tr>
<td>Google Lab Simulation</td>
<td>336.81</td>
<td>22.05</td>
<td>402.72</td>
</tr>
<tr>
<td>Bing Real Attack</td>
<td>47.04</td>
<td>12.29</td>
<td>68.02</td>
</tr>
<tr>
<td>Bing Lab Simulation</td>
<td>45.90</td>
<td>11.42</td>
<td>60.19</td>
</tr>
<tr>
<td>Wordpress Real Attack 1T1S</td>
<td>96.68</td>
<td>12.50</td>
<td>186.62</td>
</tr>
<tr>
<td>Wordpress Lab Simulation 1T1S</td>
<td>48.94</td>
<td>8.80</td>
<td>138.71</td>
</tr>
<tr>
<td>Wordpress Real Attack 1T5S</td>
<td>38.07</td>
<td>8.83</td>
<td>66.62</td>
</tr>
<tr>
<td>Wordpress Lab Simulation 1T5S</td>
<td>46.86</td>
<td>8.33</td>
<td>58.57</td>
</tr>
<tr>
<td>Wordpress Real Attack 5T1S</td>
<td>402.72</td>
<td>25.14</td>
<td>732.02</td>
</tr>
<tr>
<td>Wordpress Lab Simulation 5T1S</td>
<td>419.81</td>
<td>23.87</td>
<td>707.34</td>
</tr>
<tr>
<td><strong>Wordpress Lab Simulation 15T1S</strong></td>
<td><strong>158.86</strong></td>
<td><strong>15.35</strong></td>
<td><strong>523.09</strong></td>
</tr>
</tbody>
</table>

1T1S = 1 Thread per Second
<table>
<thead>
<tr>
<th>Name of Test</th>
<th>CPU Usage User (in %)</th>
<th>Memory Usage</th>
<th>Apps (in Mbytes)</th>
<th>Swap</th>
<th>HTTP Response (in ms)</th>
<th>Ping (in ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Min</td>
<td>Avg</td>
<td>Max</td>
<td>Min</td>
<td>Avg</td>
<td>Max</td>
</tr>
<tr>
<td>Normal Without Attack</td>
<td>1.36</td>
<td>1.37</td>
<td>1.37</td>
<td>267.66</td>
<td>268.31</td>
<td>269.46</td>
</tr>
<tr>
<td>Facebook Real Attack</td>
<td>1.42</td>
<td>1.43</td>
<td>1.46</td>
<td>177.09</td>
<td>332.65</td>
<td>398.88</td>
</tr>
<tr>
<td>Facebook Lab Simulation</td>
<td>1.41</td>
<td>1.42</td>
<td>1.43</td>
<td>247.43</td>
<td>258.42</td>
<td>269.41</td>
</tr>
<tr>
<td>Google Real Attack</td>
<td>2.23</td>
<td>3.51</td>
<td>3.73</td>
<td>314.25</td>
<td>333.66</td>
<td>338.43</td>
</tr>
<tr>
<td>Google Lab Simulation</td>
<td>3.39</td>
<td>3.76</td>
<td>3.84</td>
<td>319.02</td>
<td>328.29</td>
<td>331.18</td>
</tr>
<tr>
<td>Bing Real Attack</td>
<td>2.73</td>
<td>3.53</td>
<td>3.78</td>
<td>329.43</td>
<td>362.79</td>
<td>375.30</td>
</tr>
<tr>
<td>Bing Lab Simulation</td>
<td>2.68</td>
<td>3.07</td>
<td>3.50</td>
<td>345.79</td>
<td>371.74</td>
<td>392.90</td>
</tr>
<tr>
<td>Wordpress Real Attack 1T1S</td>
<td>6.69</td>
<td>11.17</td>
<td>12.01</td>
<td>245.08</td>
<td>248.34</td>
<td>251.12</td>
</tr>
<tr>
<td>Wordpress Lab Simulation 1T1S</td>
<td>3.04</td>
<td>6.31</td>
<td>7.12</td>
<td>269.46</td>
<td>270.90</td>
<td>272.65</td>
</tr>
<tr>
<td>Wordpress Real Attack 1T5S</td>
<td>2.40</td>
<td>3.57</td>
<td>3.68</td>
<td>268.02</td>
<td>273.27</td>
<td>275.69</td>
</tr>
<tr>
<td>Wordpress Lab Simulation 1T5S</td>
<td>2.84</td>
<td>3.26</td>
<td>3.33</td>
<td>266.76</td>
<td>271.68</td>
<td>274.20</td>
</tr>
<tr>
<td>Wordpress Real Attack 5T1S</td>
<td>16.17</td>
<td>28.48</td>
<td>31.45</td>
<td>287.48</td>
<td>319.44</td>
<td>326.28</td>
</tr>
<tr>
<td>Wordpress Lab Simulation 5T1S</td>
<td>16.70</td>
<td>27.41</td>
<td>34.71</td>
<td>284.81</td>
<td>311.80</td>
<td>318.59</td>
</tr>
<tr>
<td>Wordpress Real Attack 5T5S</td>
<td>6.25</td>
<td>11.35</td>
<td>11.96</td>
<td>273.29</td>
<td>288.34</td>
<td>299.46</td>
</tr>
<tr>
<td>Wordpress Lab Simulation 5T5S</td>
<td>8.82</td>
<td>11.51</td>
<td>11.87</td>
<td>274.10</td>
<td>282.90</td>
<td>297.46</td>
</tr>
<tr>
<td>Wordpress Lab Simulation 15T1S</td>
<td>10.79</td>
<td>51.68</td>
<td>73.29</td>
<td>323.16</td>
<td>380.25</td>
<td>423.09</td>
</tr>
</tbody>
</table>

1T1S = 1 Thread per Second
RESULT (VICTIM DOWN)

The Open Web Application Security Project
The IP of Facebook, Google Translator, Bing Translator, and CMS Wordpress is shown as IP connected to the victim.

Facebook and Google translator are using several server in their side to retrieve content from the victim.

Bing translator and some Wordpress version provide the IP of whom made request.

Increasing thread or number of CMS Wordpress as reflector from 1 CMS Wordpress to 5 CMS Wordpress will make power of attack increase 3 – 5 times.

Our test with 15 thread CMS Wordpress can make the victim could not accessed due to out of memory.

From the web server log of victim, we found that all attack come from Facebook, Google translator, Bing translator, and CMS Wordpress is using HTTP-GET attack.
COUNTERMEASURE (RESULT)

```
[root@revpro ~]# for a in `seq 1 3`; do curl -D - "http://wp1.kalpin.es/xmlrpc.php" -d "<methodCall><methodName>pingback.ping</methodName><params><param><value><string>http://web.kalpin.es</string></value></param><param><value><string>http://wp1.kalpin.es/hello-world</string></value></param><params></methodCall>";
    done; date
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2015 15:45:56 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Connection: close
Content-Length: 370
Content-Type: text/xml; charset=UTF-8

<?xml version="1.0" encoding="UTF-8"?>
<methodResponse>
  <fault>
    <value>
      <struct>
        <member>
          <name>faultCode</name>
          <value><int>0</int></value>
        </member>
        <member>
          <name>faultString</name>
          <value><string></string></value>
        </member>
      </struct>
    </value>
  </fault>
</methodResponse>

[root@revpro ~]# for a in `seq 1 3`; do curl -D - "http://wp1.kalpin.es/xmlrpc.php" -d "<methodCall><methodName>pingback.ping</methodName><params><param><value><string>http://web.kalpin.es</string></value></param><param><value><string>http://wp1.kalpin.es/hello-world</string></value></param><params></methodCall>";
    done; date
HTTP/1.1 200 OK
Date: Mon, 14 Dec 2015 15:44:31 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Connection: close
Content-Length: 0
Content-Type: text/html; charset=UTF-8

HTTP/1.1 200 OK
Date: Mon, 14 Dec 2015 15:44:33 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Connection: close
Content-Type: text/html; charset=UTF-8

HTTP/1.1 200 OK
Date: Mon, 14 Dec 2015 15:44:34 GMT
Server: Apache/2.2.15 (CentOS)
X-Powered-By: PHP/5.3.3
Connection: close
Content-Type: text/html; charset=UTF-8

Mon Dec 14 22:44:34 WIB 2015
[root@revpro ~]#   
```
CONCLUSION

• We can use Facebook, Web online translator, and CMS Wordpress as our attack platform to launch DDoS attack to other sites.
• Our experiments toward provider's web application shown that those web applications send an HTTP-GET Request to the victim and attacker can loop their request by sending many HTTP-GET Request and make the victim suffer from HTTP-GET DDoS attack.
• Our countermeasure successfully prevent HTTP-GET Attack in the source by adding control into legal website's application.
• DDoS Attack against application layer such as HTTP does not need much bandwidth to make the victim unavailable to serves request.
• Use Twitter as attack platform
• Use WhatsApp as attack platform
• Use Telegram as attack platform
• Creating an automatic tool to scan any web application in the Internet to find similar problem as above.