Using Low Orbit Ion Cannon for Denial of Service Attack Based on CVE

Thanachon Cheepborisuttikul
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, 10330, Thailand

Yunyong Teng-amnuay
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, 10330, Thailand

Abstract—This research explores CVE as target for denial of service attack on perimeter infrastructure as part of penetration testing. Exploit of CVE susceptible to DoS attack is analysed and payload extracted to be used as bullet. We employed the Low Orbit Ion Cannon (LOIC) as our attack tool. The fire control of LOIC is enhanced to accept the payload for automated firing. System inventory of the target organization and information on the CVE are used to position the cannon and target evaluation is performed on both sides of the target.

Keywords—Denial of Service Attack; CVE; Penetration testing; LOIC;

I. Introduction
Denial of Service (DoS) attacks using high volume of traffic have become one of the major attacks on the Internet. Penetration testing [1] is increasingly significant for assessing perimeter infrastructure vulnerabilities. A major publicly available source of vulnerability reports is the Common Vulnerabilities and Exposures (CVE) hosted at MITRE [2]. This work presents a new method using CVE as targets for DoS attack employing the LOIC (Low Orbit Ion Cannon) as the attack tool.

II. Related Works
A. Vulnerability Informations
1. CVE mitre involved creating a reference list of unique vulnerability, exposure names, and mapping these to appropriate items in each tool and database. MITRE analyzes vulnerabilities and exposures identified prior to the initiative as well as newly discovered ones for possible inclusion in the CVE list.
2. CVE Details [3] is a vulnerability database web site provides unique vulnerability statistics and reports based on CVE vulnerability data from NVD.

B. DoS Attack Tools

C. Low Orbit Ion Cannon
Low Orbit Ion Cannon (LOIC) [8] is an open source network stress testing and denial-of-service attack tool [9]. LOIC was used by Anonymous during Project Chanology to attack websites belonging to the Church of Scientology [10], in the attack on the Industry Association of America’s website in October 2010 [11], and again during Operation Payback in December 2010 to attack the websites of companies and organizations that opposed WikiLeaks [12]. LOIC is attractive because it is modelled after the ion cannon used in firing beam of particles to attack the imperial star cruiser in the Star Wars movie. The affinity between the science-fiction movie and computer hacker communities became evident in 2010 due to the release of this software.

D. Field Artillery System
The main functions in field artillery system [13] are communication, command, target acquisition, control, production of firing data, fire units, specialist services, and logistic services. Firing data has to be calculated and then it is the key to direct fire. The process to produce firing data is sometimes called technical fire control which is primarily concerned with targeting and the allotment of fire units to targets. A fire unit is capable of being employed to execute a fire assigned by tactical fire controller. Artillery also refer to a system of applied scientific research relating to the design, and the ordnances.
III. Methodology

The CVE system provides a reference for publicly known information-security vulnerabilities. Thus DoS related CVE can be used to prepare a fire unit (payload or bullet) and targeting.

A. CVE with DoS Vulnerability

On CVE selection is based on the work of Yung-Yu Chang et al [14]. They extracted 15 vulnerability types, and used the DoS vulnerability type as an example to demonstrate the attack trend. A list of example CVE with DoS vulnerability is as in Table I.

<table>
<thead>
<tr>
<th>CVE</th>
<th>System Services</th>
<th>DoS Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2012-5533</td>
<td>Lighttpd</td>
<td>The http_request_split_value function in request.c in lighttpd before 1.4.32 allows remote attackers to cause a denial of service (infinite loop).</td>
</tr>
<tr>
<td>CVE-2012-5329</td>
<td>TYPSoft FTP Server</td>
<td>Buffer overflow in TYPSoft FTP Server 1.1 via a long string in an APPE command.</td>
</tr>
<tr>
<td>CVE-2012-1783</td>
<td>Tiny HTTP Server</td>
<td>Tiny Server 1.1.9 and earlier via long string in a GET request.</td>
</tr>
<tr>
<td>CVE-2012-0698</td>
<td>tsd daemon service</td>
<td>tsd in Trousers before 0.3.10 allows remote attackers to cause a denial of service.</td>
</tr>
</tbody>
</table>

B. Framework for DoS attack based on CVE

There are four parts to this work. First is the manufacturing of ammunition (bullet or payload) used in the firing of target. Second is the Ion Cannon enhancement. Third is the placement of the cannon into firing position. Fourth, the last one, is the firing of the cannon. This is depicted in Fig. 1.

IV. Manufacturing Payload

In this section, we attempt to manufacture payloads (bullets or rounds). The payload is part of the exploits for CVE with DoS vulnerability. Sifting through the exploit code allows us to locate the necessary bullets.

A. Finding Exploits

We used the search terms from information types in ontology developed by Ratsameetip et al [15] to extract exploits from the public Internet. We extracted CVE with DoS vulnerability from the list of CVE in 2012 and used keywords from the vulnerabilities to search public Internet for exploits. We obtained 52 exploits for further payload extraction.

B. Analysis of Exploits

We examined the exploits found in step A in details and found that payload, or bullet, can be readily extracted from some exploits as shown in Fig. 2. However, many exploits are multi-step in nature and resist the payload extraction as shown in Fig. 3. However, adding script (in this case Python) into our fire control allows us to handle multi-step attack.
Figure 2. Exploit with readily extracted bullet.

Figure 3. Exploit that is difficult to extract payload.

From the analysis, we ascertain 2 exploits that we can readily extracted payload. Another 1 exploit can give multi-step firing solution. The rest will need in depth analysis and we leave that for the future. However, many exploits attack their targets via specialized protocol, such as Stream Control Transmission Protocol (SCTP) that are not supported in LOIC and so cannot be test fired.

For CVE susceptible to DoS attack but lacks public exploit, a total of 1,373 in our analysis, detailed analysis must be made to craft bullets. This may be difficult and further research is needed. The process of manufacturing round is as shown in Fig. 4.

Table II.

<table>
<thead>
<tr>
<th>API</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>setFloodParameter</td>
<td>Setup parameters to attack</td>
</tr>
<tr>
<td>startFlood</td>
<td>Start an attack</td>
</tr>
<tr>
<td>stopFlood</td>
<td>Stop an attack</td>
</tr>
</tbody>
</table>

VI. Placement and Targeting

This phase mainly deals with identifying the network topology of the target organization and characteristic of the target systems through the use of system inventory of the organization and the database of DoS-related CVE. The inventory provides an overview of existing technologies or devices and helps in enumerating services running over the network. Information on the DoS-related CVE will help identify the port or the fingerprint of the target services. This allows us to place the LOIC near the target system and to "aim" the cannon to the appropriate IP address and port necessary for firing. Step V also helps identifying the payload for particular target. The rate of firing, or frequency (f), is the amount of packets, or bullets, per second the cannon can fire on the target. The Size (s) is the number of bytes of each bullet. If the duration of attack is d seconds, then the total (T) for particular target can be obtained using the following equation.

\[ T = 8fsd \]  

The limit of attack is thus based on the capability of the network interface of the cannon and the path capacity from the cannon to the target.

VII. Firing

From the detail about the target, firing of the cannon comprises two steps of using the information to set up the fire control script and the submission of the fire control script to the LOIC via the extended API, First is to set up the fire control script using Python with firing solution. The second step is the submission of the fire control script to order the LOIC via the API that extends LOIC via the following code snippet in Fig. 5.
import urllib
# call the LOIC API Restful as follows:
u=urllib.urlopen('http://192.168.171.138:8081/
extended-loic/ExtendedLoicApiRESTful/
setFloodParameter?param=target_ip=192.168.171.14,
cve_id=CVE-2012-5533, port=80, method=http')
data=u.read()

Figure 5. Example code calling API extended LOIC.

VIII. Surveillance

Surveillance involves monitoring the success in attacking the targeting. This was done on two fronts: external and internal surveillance. The external surveillance proposed in this paper focuses on payload packets and responses from target. We used Wireshark [17] as the packet analyzer. The internal surveillance focuses on monitoring system services and performance including the memory used and CPU usage for both before and after attack. No special tools are needed.

IX. Results

We tested the firing on 3 targets as illustrated in Table III. These targets are based on open source services but can help proof our method.

<table>
<thead>
<tr>
<th>CVE ID</th>
<th>Packet Size (Bytes)</th>
<th>Round(s)</th>
<th>Duration</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>CVE-2012-5533</td>
<td>112</td>
<td>1</td>
<td>42 msec.</td>
<td>Crash on the first HTTP request.</td>
</tr>
<tr>
<td>CVE-2012-1783</td>
<td>60</td>
<td>5</td>
<td>3.49 sec.</td>
<td>Crash.</td>
</tr>
<tr>
<td>CVE-2012-5329</td>
<td>1,500</td>
<td>8</td>
<td>4.11 sec.</td>
<td>Crash and memory corruption.</td>
</tr>
</tbody>
</table>

X. Conclusions

Our enhanced LOIC demonstrates automated firing for attack on CVE with DoS vulnerability. Although the cannon is limited in handling diverse protocols it still proves the viability of such avenue to penetration testing. With proper and intensive analysis of CVE and exploits we should be able to compile a database of payloads. The result can also help develop such tool.

In our limited analysis is based only on open source targets we can still successfully test fire 3 targets and 13 potential targets for further analysis and test firing.

XI. Future Works

This work is a proof of concept in using DoS attack tool for automated penetration testing. There are still a lot of work to be done.

- We used only 2012 CVE for our work. This should be extended to cover all DoS-based CVE and their exploits.
- CVE susceptible to DoS attack but lack public exploit should be analysed to craft the payloads.
- A more comprehensive placement and targeting should be worked out.
- Employ distributed denial-of-service (DDoS) attack tool. This includes targeting across bridges and routers and using cluster to build the cannon.
- Extend the firing to include more complex targets especially expensive and business oriented software packages.
- Extend LOIC to cover firing under extra protocols, such as the Stream Control Transmission Protocol (SCTP).

References


