UACEE International Journal of Advances in Computer Networks and its Security – IJCNS
Volume 3 : Issue 2 [ISSN 2250 – 3757]
Publication Date : 05 June 2013

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
E-mail: thanachon.ch@gmail.com

Yunyong Teng-amnuay
Department of Computer Engineering
Faculty of Engineering, Chulalongkorn University
Bangkok, 10330, Thailand
E-mail: Yunyong.T@chula.ac.th

***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*

The amount of research on DoS attack [4] is impressive. Slowloris [5] is a DoS attack tool that keeps many connections to the target web server open. The low-rate TCP-targeted DoS attack [6] exploits the retransmission timeout (RTO) mechanism. SYN-Flooding attack [7] sends a flood of TCP/SYN packets. However, we choose LOIC as our attack tool.

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.

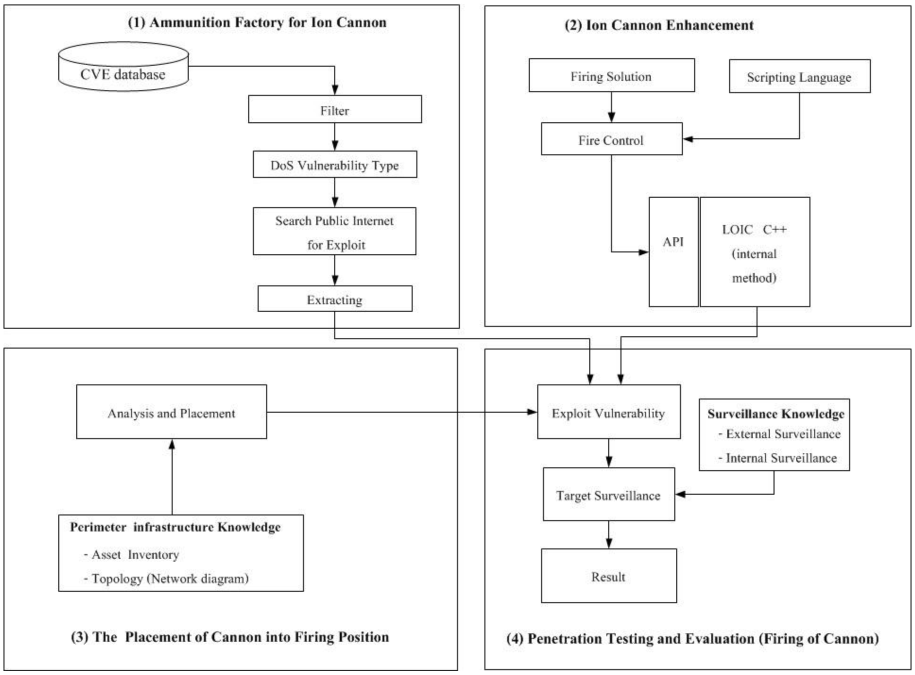

(1) Ammunition Factory for Ion Cannon
CVE database
Filter
DoS Vulnerability Type
Search Public Internet for Exploit
Extracting
(2) Ion Cannon Enhancement
Firing Solution
Scripting Language
Fire Control
API
LOIC C++ (internal method)
Analysis and Placement
Perimeter infrastructure Knowledge
- Asset Inventory
- Topology (Network diagram)
(3) The Placement of Cannon into Firing Position
Exploit Vulnerability
Surveillance Knowledge
- External Surveillance
- Internal Surveillance
Target Surveillance
Result
(4) Penetration Testing and Evaluation (Firing of Cannon)

Figure 1. Framework for DoS attack based on CVE.

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 I. SAMPLES OF CVE WITH DOS VULNERABILITY

CVE	System Services	DoS Description
CVE-2012-5533	Lighttpd	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).
CVE-2012-5329	TYPSoft FTP Server	Buffer overflow in TYPSoft FTP Server 1.1 via a long string in an APPE command.
CVE-2012-1783	Tiny HTTP Server	Tiny Server 1.1.9 and earlier via long string in a GET request.
CVE-2012-0698	tcsd daemon service	tcsd in Trousers before 0.3.10 allows remote attackers to cause a denial of service.

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

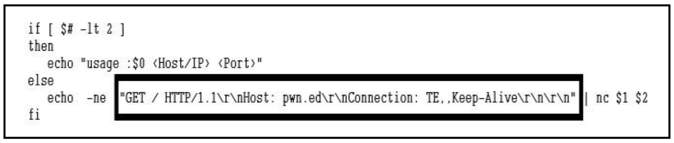
In this section, we attempt to manufacture payloads (bullets or rounds). The payload is part of the exploits for CVE with DoS vulnerability. Sifting through the exploit code allows us to locate the necessary bullets.

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.

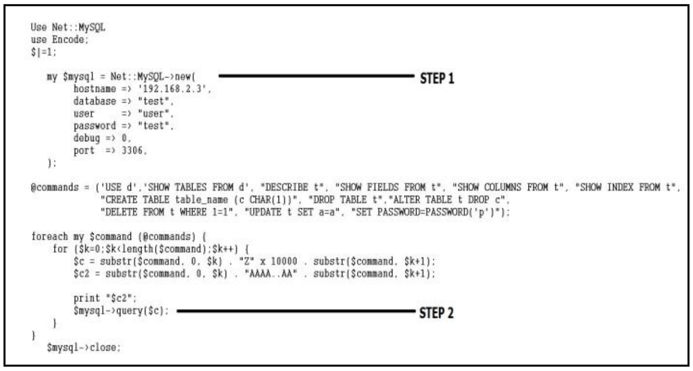
B. *Analysis of Exploits*

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.



```
if [ $# -lt 2 ]
then
   echo "usage :$0 <Host/IP> <Port>"
else
   echo  -ne "GET / HTTP/1.1\r\nHost: pwn.ed\r\nConnection: TE,,Keep-Alive\r\n\r\n" | nc $1 $2
fi
```

Figure 2. Exploit with readily extracted bullet.

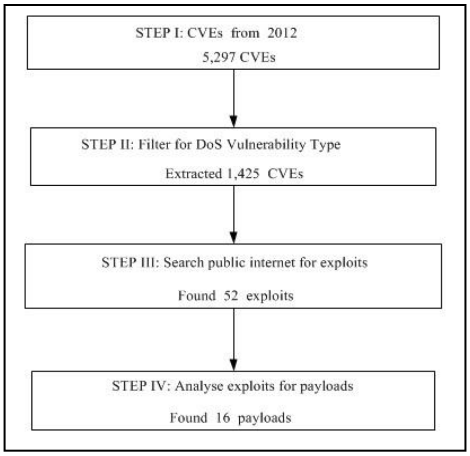


```
Use Net::MySQL
use Encode;
$|=1;

   my $mysql = Net::MySQL->new(
         hostname => '192.168.2.3',
         database => "test",
         user     => "user",
         password => "test",
         debug => 0,
         port  => 3306,
   );

@commands = ('USE d','SHOW TABLES FROM d', "DESCRIBE t", "SHOW FIELDS FROM t", "SHOW COLUMNS FROM t", "SHOW INDEX FROM t",
             "CREATE TABLE table_name (c CHAR(1))", "DROP TABLE t","ALTER TABLE t DROP c",
             "DELETE FROM t WHERE 1=1", "UPDATE t SET a=a", "SET PASSWORD=PASSWORD('p')");

foreach my $command (@commands) {
    for ($k=0;$k<length($command);$k++) {
         $c = substr($command, 0, $k) . "Z" x 10000 . substr($command, $k+1);
         $c2 = substr($command, 0, $k) . "AAAA..AA" . substr($command, $k+1);

         print "$c2";
         $mysql->query($c);
    }
}
   $mysql->close;
```

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 protolcol, 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.



STEP I: CVEs from 2012
5,297 CVEs
STEP II: Filter for DoS Vulnerability Type
Extracted 1,425 CVEs
STEP III: Search public internet for exploits
Found 52 exploits
STEP IV: Analyse exploits for payloads
Found 16 payloads

Figure 4. Manufacturing payload.

Of the 16 payloads we tested the firing on 3 only vulnerabilities target open source services.

V. Extending LOIC

This research employed the Low Orbit Ion Cannon (LOIQ version c++ loic 0.3 a) which was written in C++ [16] as our attack tool to simplify this work. As LOIC does not provide the API we have to extend it for automated firing and control. The list of API is as follows.

TABLE II. API EXTENDING LOIC

API	Description
setFloodParameter	Setup parameters to attack
startFlood	Start an attack
stopFlood	Stop an attack

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.

$$T = 8fsd \tag{1}$$

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 III. FIRING RESULT

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

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

[1] W. G. J. Halfond, S. R. Choudhary, and A. Orso, "Improving penetration testing through static and dynamic analysis", Software Testing Verification and Reliability vol. 21, pp. 195-214, September 2011.

[2] CVE Editorial Board, Common Vulnerabilities and Exposures: The Standard for Information Security Vulnerability Names. [cited December 25, 2011]; Available: http://cve.mitre.org.

[3] CVE Details: The Ultimate Security Vulnerability Datasource [cited January 4, 2013]; Available: http://www.cvedetails.com.

[4] W. Liu, "Research on DoS Attack and Detection Programming", 3rd international conference on Intelligent Information Technology Application, pp. 207-210, 2009.

[5] R. "RSnake" Hansen, The DoS Project's "Slowloris" Denial Of Service Attack Tool [cited June 17, 2009]; Available: http://packetstormsecurity.com/files/78491/Slowloris-Denial-Of-Service-Tool.html.

[6] P. Efstathopoulos, "Practical Study of a Defense Against Low-Rate TCP-Target DoS Attack", Internet Technology and Secured Transactions, November 2009.

[7] R. K Sahu, "A Performance Analysis of Network under SYN-Flooding Attack", 9th international conference on Wireless and Optical Communications Networks, September 2012.

[8] Praetox Technologies, An Open Source Network Stress Testing And Denial-Of-Service Attack Application. "Low Orbit Ion Cannon", [cited January 24, 2012]; Available: http://packetstormsecurity.com/files/109075/TA12-024A.txt

[9] J. Roman, B. Radek, V. Radek, and S. Libor, "Launching distributed denial of service attacks by network protocol exploitation", 2nd international conference on Applied Informatics and Computing Theory, pp. 210-216, 2011.

[10] Church Of Scientology [cited February 1, 2013]; Available: http://www.scientology.org.

[11] The Attack on The Industry Association of America's Website [cited October 2010]; Available: http://www.pcmag.com/article2/0,2817,2371784,00.asp.

[12] WikiLeaks [cited January 1, 2013]; Available: http://wikileaks.org

[13] U. S. Army Field Artillery School [cited February 17, 2013]; Available: http://sill-www.army.mil/USAFAS/.

[14] Y. Chang, P. Zavarsky, R. Ruhl, and D. Lindskog, "Trend Analysis of the CVE for Software Vulnerability Management", Privacy, Security, Risk and Trust, pp. 1290-1293, October 2011.

[15] R. Wita, N. Jiamnapanon, and Yunyong Teng-amnuay, "An Ontology for Vulnerability Lifecycle", 3rd international symposium on Intelligent Information Technology and Security Informatics, pp. 553-557, 2010.

[16] LOIC Release [cited March 3, 2011]; Available: http://loiq.sourceforge.net/.

[17] Wireshark [cited August 6, 2011]; Available: http://www.wireshark.org.