

Dissertation Title: NETSTAB

A Network Perimeter Defence Penetration Testing Toolkit



Author

Ayesha Noor Arshad (175505)

Hamza Javed (183935)

Supervisor

AP Waleed Bin Shahid (IS Department)

A Dissertation

Submitted to the Department of Computer Software Engineering,
Military College of Signals,
National University of Sciences and Technology.

In partial fulfillment for the requirements
for B.E Degree in Software Engineering

July, 2020

Certificate of Corrections & Approval

Certified that work contained in this thesis titled “ *NetStab-A Perimeter Defense Penetration Testing Toolkit* ”, carried out by *Ayesha Noor Arshad and Hamza Javed* under the supervision of *AP Waleed Bin Shahid (IS Department)* for partial fulfillment of Degree of Bachelors of Electrical Engineering, in Military College of Signals, National University of Sciences and Technology, Islamabad during the academic year 2019-2020 is correct and approved. The material that has been used from other sources has been properly acknowledged/referred to.

Approved By

Supervisor

Date: _____

Declaration

No portion of work presented in this thesis has been submitted in support of another award or qualification in either this institute or anywhere else.

Plagiarism Certificate (Turnitin Report)

This thesis has been checked for Plagiarism. Turnitin report endorsed by Supervisor is attached.

Signature Of Student

(Ayesha Noor Arshad, 175505)

Signature Of Student

(Hamza Javed, 183935)

Signature Of Supervisor

Acknowledgments

Primarily, I would like to thank my Creator Almighty Allah to have directed me to complete this project with success, to have helped me when I felt it was impossible, to have given me hope when there were adversities and dead losses. Indeed without his help, I could never have accomplished even an ounce of what I have today. I extend my humble gratitude to Allah Almighty for sending those who helped me in this work.

I am exceedingly indebted to my beloved parents who raised me, gave me the nonpareil gift of education, and believed in me.

I am obliged to extend my sincerest gratitude to my supervisor, AP Waleed Bin Shahid for his assistance throughout my final year project development phase. A final year project is a bridge between theory and practical knowledge. And I can proudly say that Sir Waleed taught us how to implement theoretical knowledge in the practical field. I would also like to pay special thanks to Major Sohaib and Dr. Zaki Murtaza for their tremendous support and assistance. Each time I got stuck in something, they came up with the solution to help me during this tenure. Without their help, I wouldn't have been able to complete my project. I appreciate their guidance throughout the whole project.

Finally, I would like to express my gratitude to my team, friends, and all the individuals who have rendered valuable assistance to my study.

*Dedicated to my encouraging family and dedicated teachers whose
prodigious support and cooperation led me to this wonderful
accomplishment.*

Abstract

NetStab is a state of the art software-based toolkit that launches penetration tests on a Network Firewall to evaluate its efficiency, performance and guide the network administrator to fill up the loopholes. In the times of intricate communications converting the physical world into global villages, many organizations observe multitude network infiltrations henceforth secure communication becomes imperative to be ensured. NetStab will strengthen firewalls through its vulnerability assessment tests, therefore, preventing network infiltration, data leakage, and preventing hackers from accessing sensitive information.

The purpose of this thesis document is to proclaim constituent services our product will provide, the methodology for the product design and development, functionalities and procedures followed to deliver these functionalities, constraints respective to procedures followed, and final product results obtained at the time of project submission.

The purpose of this project revolves around the idea to establish an indigenous Firewall Penetration Testing Toolkit to confirm the efficiency of the firewall configured to secure the network. As there is a shortage of pen-test tools in the market and there isn't a tool that covers all aspects and types of firewall penetration tests, all the tools available are either not comprehensive or only offer one type of functionality required to run a penetration test. NetStab offers all these types of tests in one tool which will be easy to use and will combine all the functionality of the utility programs to ensure a comprehensive testing procedure.

Table Of Contents

Abstract	7
I. Introduction	10
II. Literature Review	12
III. Design and Development	14
3.1 User Classes and Beneficiaries of the Product	14
3.2 Prime Product Objectives	14
3.3 Deployment Diagram	15
3.4 Features	16
3.5 Environment / System: Kali Linux	16
3.6 Coding Languages	16
3.7 Modules	17
A. Locating Firewall	17
B. Discovery Scan	17
C. Port Scanning	18
D. Banner Grabbing	18
E. IP Spoofing	19
F. SSH Tunneling	19
G. Port Redirection	19
H. HTTP Tunneling	20
I. Reverse Shell Tunneling	21
J. Custom Packets Testing	21
K. Fragmented Malware Packets	21
L. ICMP Tunneling	22
M. Report Generation	22
N. UI/UX	23

IV. Project Analysis and Evaluation	26
4.1 Test Results	28
I. Reconnaissance Phase	28
II. Attack Phase	31
III. Overall Testing and Report Generation	34
4.2 Test Report	38
V. Future Work	45
VI. Conclusion	45
Appendix A - User Manual	46
I. About NetStab	46
II. Prerequisites	46
III. GUI guide	46
IV. Instructions for Use	49
V. View the Previous Report	53
Bibliography	55

I. Introduction

NetStab is a concept focused on enhancing the competence of Network Perimeter Defense devices primarily Network Firewalls. Numerous enterprises, institutions, and businesses that are connected to the internet use network firewalls to safeguard their computer networks from internal security policy transgressions and external breaches. Firewalls act as a shield against not only malicious traffic from entering the network that could cause damage or theft of the organizational resources but also monitors malicious activities from inside the network. Firewalls are the main line of defense against malicious attacks that can originate from both inside and outside the network, and yet they are not tested adequately. The prime justification for this lies in the inadequacy of dependable and comprehensive penetration testing technologies.

There are a variety of tools and techniques that can help evade a network firewall but no consolidated toolkit is present that performs the firewall penetration testing consummately. Tools like Metasploit, Nessus, Nmap, Burp Suite, Nemesis, Snort, Maltego, Libnet, and Hping3 are available in the market but none of them are comprehensive enough to perform all of the required vulnerability tests. Firewalk is an incapacitated and outdated tool. No other tool is dedicated to the network Firewall pen-testing to date. Another problem with the pen-testing procedure is that we need to trust external experts for the testing and there is no comprehensive tool to test the firewall as the classical manual approach requires resources and time.

The pen-testing procedure primarily focuses on the design and implementation of the firewall and additionally on its capability to withstand breach of protocols (**Kamara, 2003**). Testing Implementation phase first studies the firewall policy and the exploited exact same policies to

confirm if the firewall can resist the annexation (**Shwetambari, 2014**). By performing a series of tests on the network firewall's traffic filtering mechanisms and its persistence towards penetration testing the efficiency of the firewall is measured. After careful examination, the possible assailable points in a network are discovered and then these loopholes are secured. Netstab helps with the first phase of this procedure where it launches network-specific penetration testing attacks to check the persistence level and ability to secure a network.

Attackers, from inside and outside, are capable of bypassing these firewalls to perform illicit actions without getting caught. This accentuates the need to come up with a toolkit that ruthlessly gauges the efficacy of network firewalls by doing both internal and external penetration testing. There are a variety of tools and techniques that can help evade a network firewall but no consolidated toolkit is present that performs the firewall penetration testing consummately. The toolkit will launch a variety of attacks to evade the firewall from both inside and outside and generate an elucidated Audit Report at the end reflecting upon the strengths and weaknesses of the targeted network.

II. Literature Review

A network firewall is the main line of defense for computer networks against malicious attacks (**Cho Hong**), and these network perimeter devices (firewalls) are the only line of defense against the malicious attacks that can cost a significant amount of data, resources and digital assets to our organizations, businesses, and institutions (**Haeni, 1997**). Network Firewalls are described as the mainstay of digital asset security against malevolent attacks intended towards corruption, theft, and deauthentication of digital assets (**Liu, 2008**). A network firewall is primarily used to filter out restricted access to the internal network and resources of the enterprise. The network firewall logs and checks inbound and outbound communications made inside the network and from outside to the network so that it can detect if some malicious source is trying to access the network resources (**Shwetambari, 2014**). A miss configured network firewall or a bug in the development of a network firewall can create backdoors in the network that could lead to loss or disclosure of information (**Liu, 2008**). Unfortunately, a network firewall can fail in numerous scenarios if not configured properly, or if there is a loophole in the security policy (**Haeni, 1997**). These vulnerabilities are either present from the development stage or created but they can be exploited to attack an organizational computer network that the network firewall is supposed to protect. This failure of network firewall implementation and functioning can cause the loss of a fortune to these businesses.

In order to check the effectiveness of a network firewall, it needs to be tested for the possible attack scenarios called penetration testing (**Kamara, 2003**). Despite their crucial role in network perimeter defense, there are still no well-defined techniques to test firewalls but they can be

tweaked to achieve their proficiency in the protection against security breaches. Firewall security policies are also a point of weakness to be tested while testing firewalls but security policies are vendor-specific, and they need to be closely monitored as per standards (**Shwetambari, 2014**). A network firewall policy enumerates a set of rules and regulations, where these rules can be considered in the form of *(predicate)→(decision)*, where predicate defines the violation of a security protocol and decision helps administrators to block the effects administered by the predicate event (**Liu, 2008**).

III. Design and Development

This area of the artifact includes a description of constituent services and features our product will provide, the methodology for the project design and development, functionalities and procedures followed to deliver these functionalities in detail.

3.1 User Classes and Beneficiaries of the Product

There are three types of user classes in this community:

1. Network Administrators.
2. Information Security and Cyber Security Engineers.
3. Network Engineers and Network Security Analysts.

NetStab will automate the process of firewall penetration testing for Network Penetration Testers, Network Administrators, Cyber Security Analysts, Network Security Engineers, Network forensics experts, Cyber Security Critics, and Information Security Engineers to ensure impenetrable communication networks for organizations.

3.2 Prime Product Objectives

1. Gathering Network Design Details

The first phase of the NetStab pen-testing process is aimed towards obtaining Network Firewall address and scaling the network and devices present on it.

2. Reconnaissance

The second phase determines active infiltration points present in the network and it gathers information regarding the network firewall installed to secure the network. Tests like Port scanning are included to discover open or available ports and services running on them, this process enlists the possible points of infiltration and breach. Network enumeration is carried out by discovering the firewall and then banner grabbing to discover the firewall architecture.

3. Conducting Vulnerability Analysis Tests

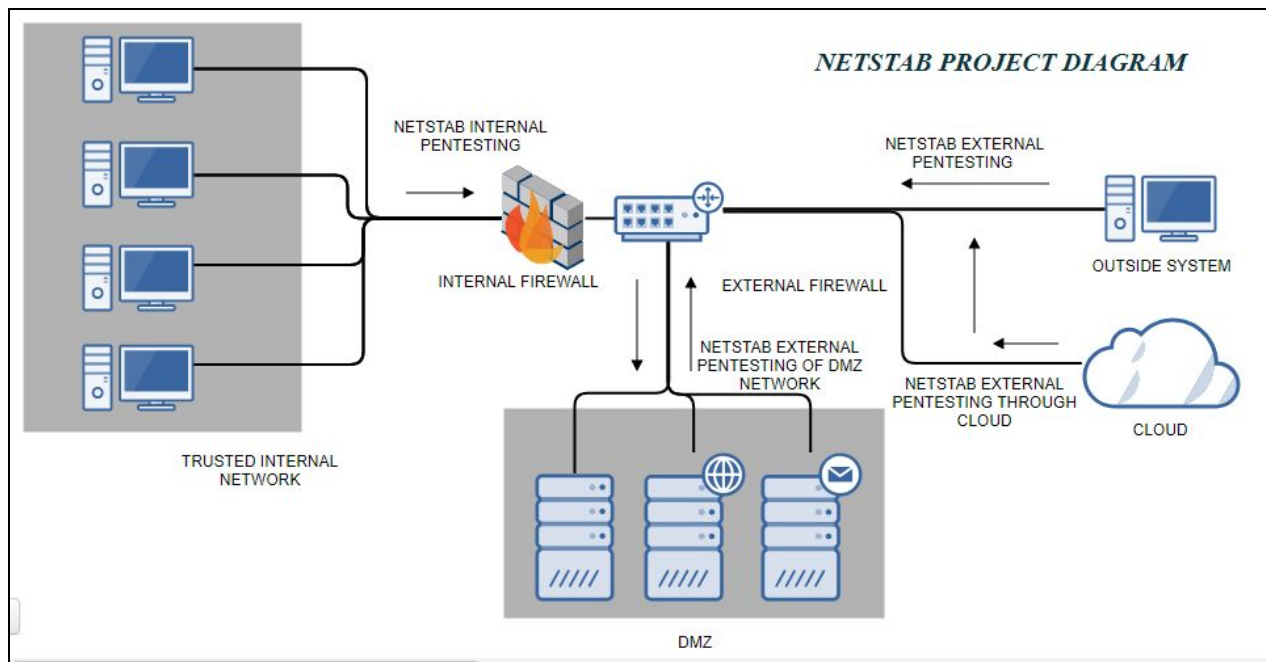
Diverse network tunneling techniques such as SSH tunneling, HTTP tunneling, ICMP tunneling, and Port redirection are used to bypass the firewall restrictions. Moreover, Custom packets generation modules are included in our novel pen-testing solution to check firewall responses to test the target firewall's ability to enforce security policies configured by network administrators.

4. Generating Audit Report

A comprehensive audit report is generated at the end of the automated pen-testing process which will show the results of all the tests run by NetStab. This report will include all the vulnerabilities found and details on all the tests run on the firewall.

3.3 Deployment Diagram

Following is the deployment diagram of NetStab included in this artifact to explain the collaboration of different participating entities in the working of this project.



3.4 Features

Architecture design for Netstab contains well-defined standalone modules. These modules are actually penetration tests designed to test the feasibility of a Firewall, and they are completely autonomous in nature. Due to low coupling, this architecture is preferred as modules are independent in working, each module (i.e automated penetration test) contributes to the resultant audit report separately with high cohesion effect kept in mind.

3.5 Environment / System: Kali Linux

3.6 Coding Languages:

1. Bash Scripting
2. Python
3. Java (GUI)

3.7 Modules

Locating Firewall

The very first module of the NetStab pen-testing toolkit is locating and confirming the existence of the network firewall. This module gets the IP address of the Firewall and confirms if there is a firewall filtering the incoming or outgoing network traffic. NetStab uses Network Mapper (Nmap) to confirm the existence of the network firewall.

The working mechanism behind this module consists of a bash script that automates the process of sending Network Mapper probes to check if traffic faces any resistance while passing through ports. In case we discover closed or filtered ports, it indicates that the network traffic is monitored by a network firewall, and we give the verdict that the firewall is present.

Discovery Scan

After locating the network firewall the next module of NetStab pen-testing toolkit confirms whether the host is live or not in this case the host is the network firewall. This module confirms that network traffic is successfully flowing through the gateway of the firewall. This module consists of a bash script that automates the process of sending ping requests to the gateway allocated to the firewall.

Port Scanning

Port Scanning checks for open ports and services running on network ports. A total of five different Nmap port scans are incorporated in a bash script that automates these scans that check the ports using five different port scanning methods called TCP Scan, UDP Scan, Null Scan, FIN Scan, and finally Xmas Scan. Ports might be allowing different kinds of traffic so to ensure definitive test results these five different results are included in the Port Scanning module of the NetStab pen-testing toolkit.

TCP Scan sends TCP packets through all of the 65535 ports to check if ports allow TCP traffic to enter the network through these ports or not. UDP Scan sends UDP packets to these ports to check whether UDP traffic is allowed to pass through the ports into the network or not. Null Scan sends data-less packets to check if the firewall has the capability to detect these data-less packets or not. FIN Scan sends packets with FIN (finish) flag enabled, these are used to check if the firewall can detect these FIN (finish) flag enabled packets or not. Similarly, Xmas Scan is another scan used to detect open ports using specific probes.

Banner Grabbing

Banner Grabbing helps with finding out all the services and kernels running on different outlets or inlets of the firewall. It enlists the traffic acceptance protocols defined for various ports open for

communication. A Nmap Script Engine (NSE) script used for banner grabbing is triggered by a bash script in this module.

IP Spoofing

IP Spoofing checks if the firewall allows spoofed IP Packets to enter the network or not. Specially crafted hping3 packets are introduced into the network to check if the network firewall can perceive the camouflaged IP packets or not. Hping3 is the Kali Linux tool used for generating spoofed IP Packets. Spoofed IP packets are introduced to the network to check if the network firewall allows them to pass.

SSH Tunneling

Secure Shell Tunneling (SSH Tunneling) forms an alternate route or tunnel for the internet traffic to pass through the network perimeter defense. SSH tunneling is a widely used method to bypass a network firewall in order to allow malicious traffic into the organizational network and sometimes to send data of prime importance out of the network as well. An automated bash script of this module confirms if the network firewall holds the capability to stop an SSH Tunnel or not. SSH Tunnel is established using the ssh command/tool in Kali Linux and it establishes an ssh tunnel with Amazon Web Server to provide a covert route for the network traffic.

Port Redirection

Port Redirection is the process of rerouting the traffic flow through another port other than the default port, this is done in order to bypass the firewall and transit malicious traffic through an open port when the firewall is blocking the default port for the traffic. This module confirms if the network firewall holds the capability to stop Port Redirection or not. Port Redirection is done by establishing a covert route/tunnel for the network traffic to transit is established using the ssh command/tool in Kali Linux and it establishes an ssh tunnel with Amazon Web Server to provide a covert route for the network traffic.

HTTP Tunneling

HTTP Tunneling forms an alternate HTTP route or tunnel for the internet traffic to pass through the network perimeter defense. HTTP tunneling is a widely used method to bypass a network firewall in order to allow malicious traffic into the organizational network and sometimes to send data of prime importance out of the network as well. This module confirms if the network firewall holds the capability to stop an HTTP Tunnel or not. HTTP tunnel is established by an automated bash script that uses an HTTP proxy setting, the private network route/tunnel established allows HTTP network traffic to transit. This tunnel is established using the ssh command/tool in Kali Linux and it establishes an ssh tunnel with Amazon Web Server to provide a covert route for the network traffic.

Reverse Shell Tunneling

Reverse Shell Tunneling is another tunneling technique used to bypass firewall restriction but this attack requires an outside source to open communication channels and then the client from inside the networks communicates with the server and the server gets root access to the client's system. This module confirms if the network firewall allows reverse shell communication or not. This module uses python based scripts triggered by bash scripts to establish a reverse shell tunnel.

Custom Packets Testing

This module crafts custom packets using hping3 to test a variety of communications that are allowed. Communications such as TCP, UDP, PUSH flagged, Null flagged, or Urgent flagged. Since these communications can be used to send malicious traffic into the network, they are capable of generated back-doors to allow malicious traffic into the networks as well. Hping3 is the Kali Linux tool used for generating packets with different characteristics i.e. random destination packets and packets having specific flags enabled to check the persistence of the target network firewall.

Fragmented Malware Packets

Often fragmented packets are used to infect the network with malware, these fragmented malware packets can be lethal to the system. This module checks if the target network firewall allows the transfer of malicious files in the form of fragments or not. Hping3 is used to fragment the malware file wrapped in packets and sends it to the target resource within the network. If the test is successful then the resultant test verdict is appended into the NetStab report.

ICMP Tunneling

ICMP Tunneling forms an alternate ICMP route or tunnel for the internet traffic to pass through the network perimeter defense. The concept of this tunneling is to wrap up packets in the form of ping requests to deceive the firewall that it is just receiving ping requests but actually it is the restricted traffic disguised as ICMP traffic. ICMP tunneling is a very intricate method used to bypass a network firewall. ICMP tunneling is not a usual method used to open blocked websites only, as it takes time to wrap the traffic in the form of ping requests hence it is used for exceptional purposes. This module confirms if the network firewall holds the capability to stop restricted traffic disguised as ICMP requests. A python-based ICMP tunnel script controlled by a bash script is used to establish these tunnels.

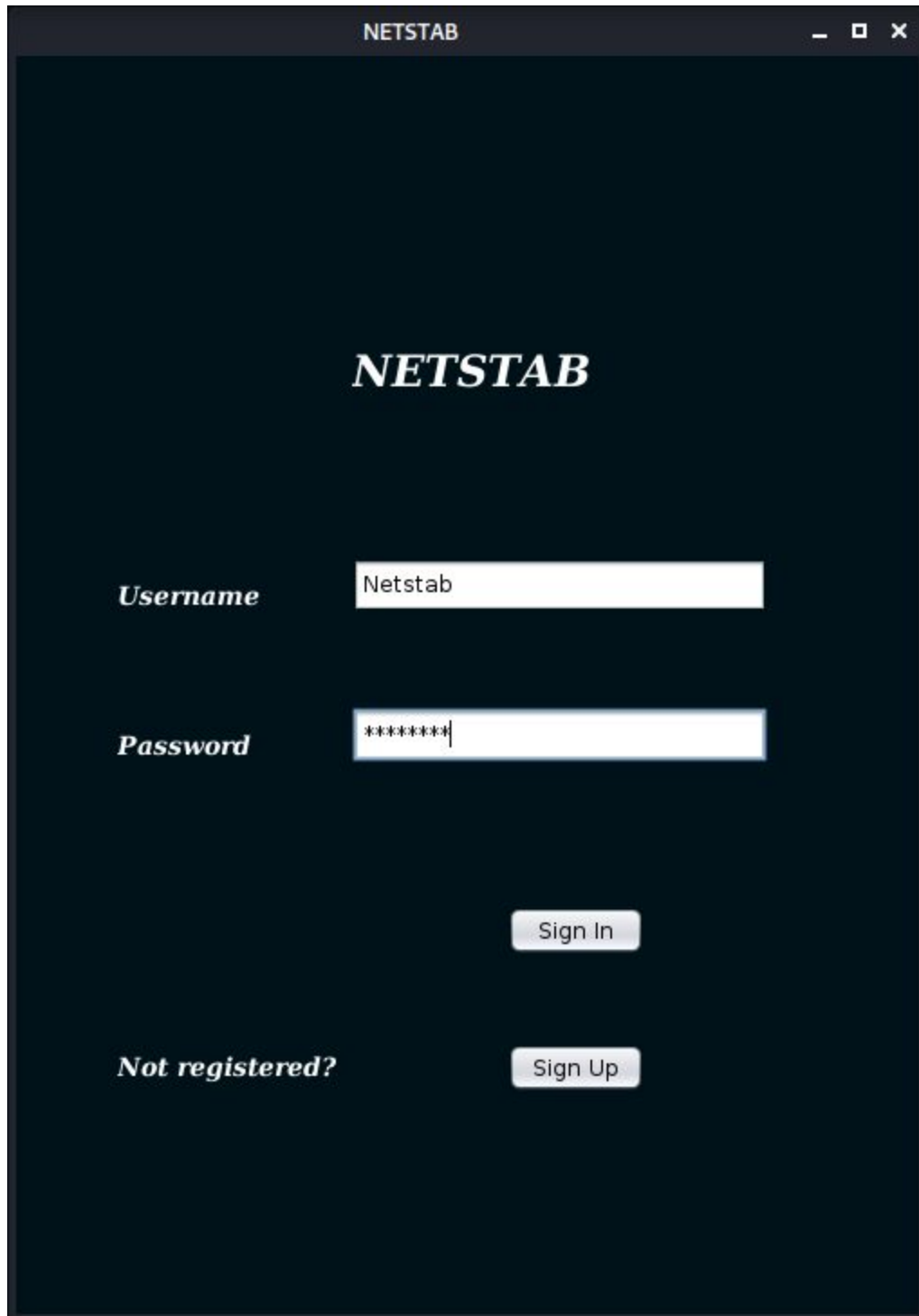
Report Generation

Report generation has two phases wherein the first phase the individual results of the tests/modules are appended to a report document file and in the second phase, this file is converted into a PDF file ready to be displayed, saved, or printed.

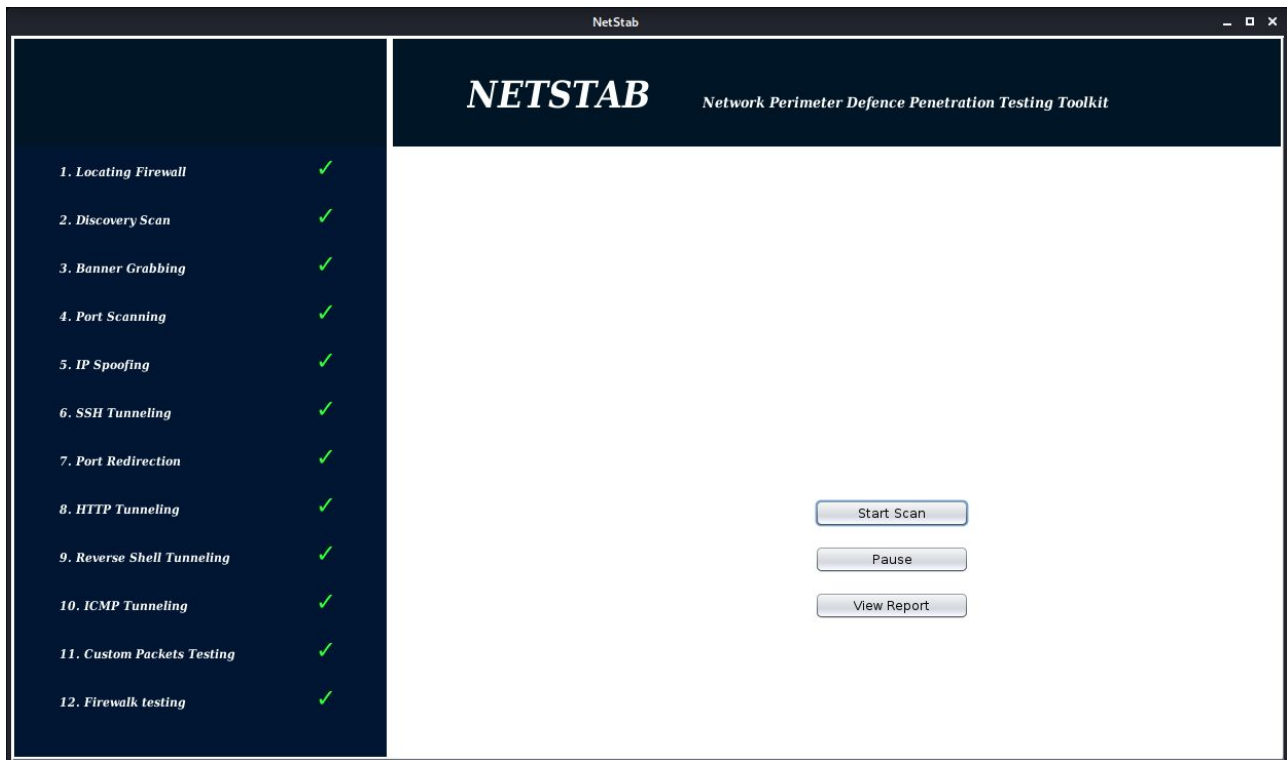
UI/UX

GUI is programmed in NetBeans Java IDE. Three windows i.e. login screen, main window, and report window are included in the GUI.

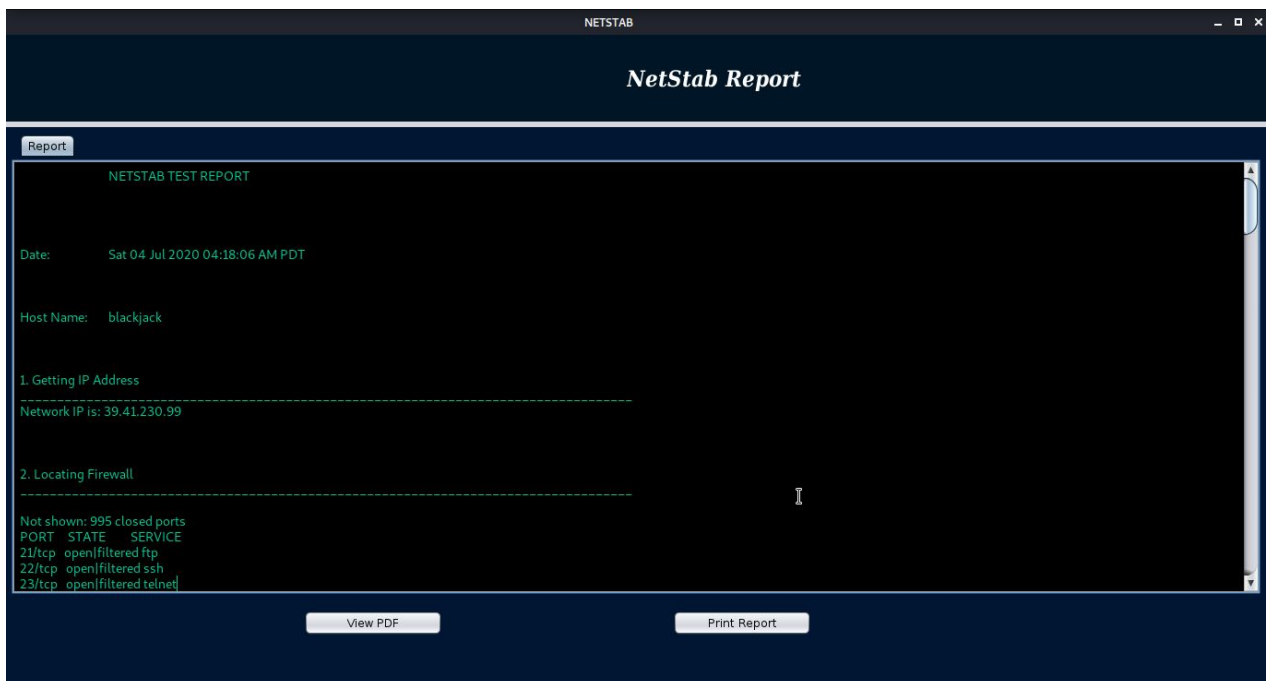
Login Screen



Main Interface



Report Window



IV. Project Analysis and Evaluation

This area of the artifact includes the findings of the test phase in detail for NetStab.

Testing System: Kali Linux

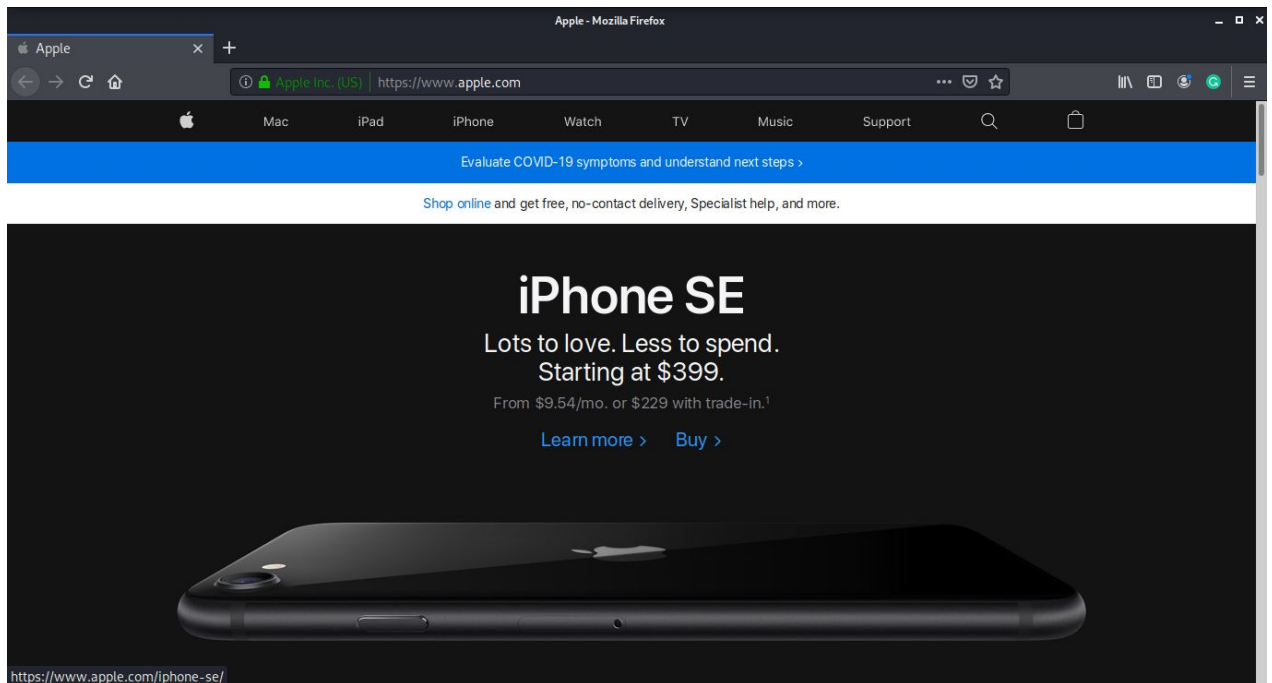
External System:

1. AWS Kali Linux Server
2. AWS Ubuntu Server

Target Firewall: UFW

Blocked Website for Testing: apple.com

1. Before blocking the website



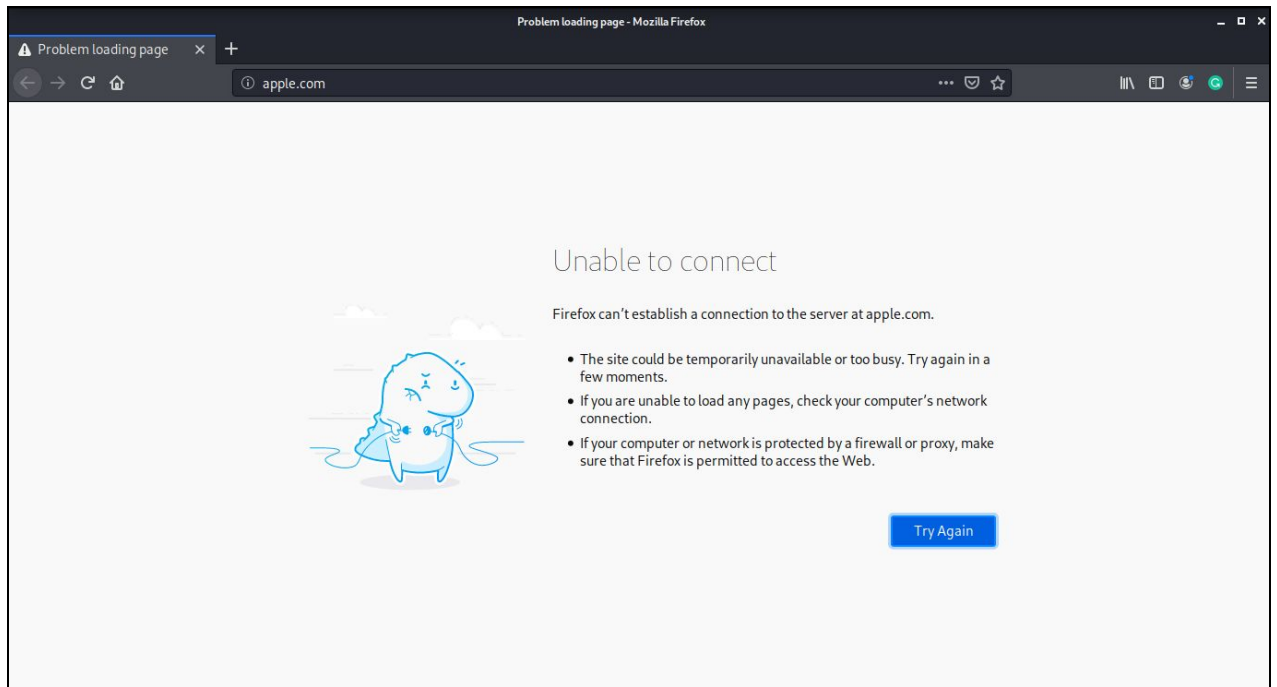
2. Blocking apple.com

```
blackjack@blackjack:~$ sudo ufw status
Status: inactive
blackjack@blackjack:~$ sudo ufw enable
Firewall is active and enabled on system startup
blackjack@blackjack:~$ sudo ufw status verbose
Status: active
Logging: on (low)
Default: deny (incoming), allow (outgoing), disabled (routed)
New profiles: skip
blackjack@blackjack:~$ host apple.com
apple.com has address 17.142.160.59
apple.com has address 17.178.96.59
apple.com has address 17.172.224.47
apple.com mail is handled by 10 nwk-aemail-lapp01.apple.com.
apple.com mail is handled by 10 nwk-aemail-lapp02.apple.com.
apple.com mail is handled by 10 nwk-aemail-lapp03.apple.com.
apple.com mail is handled by 10 ma1-aemail-dr-lapp01.apple.com.
apple.com mail is handled by 10 ma1-aemail-dr-lapp02.apple.com.
apple.com mail is handled by 10 ma1-aemail-dr-lapp03.apple.com.
blackjack@blackjack:~$
```

3. Blocked all three IPs of apple.com

```
blackjack@blackjack:~$ host apple.com
apple.com has address 17.172.224.47
apple.com has address 17.178.96.59
apple.com has address 17.142.160.59
apple.com mail is handled by 10 ma1-aemail-dr-lapp01.apple.com.
apple.com mail is handled by 10 ma1-aemail-dr-lapp02.apple.com.
apple.com mail is handled by 10 ma1-aemail-dr-lapp03.apple.com.
apple.com mail is handled by 10 nwk-aemail-lapp01.apple.com.
apple.com mail is handled by 10 nwk-aemail-lapp02.apple.com.
apple.com mail is handled by 10 nwk-aemail-lapp03.apple.com.
blackjack@blackjack:~$ sudo ufw deny from any to 17.172.224.47
Rule added
blackjack@blackjack:~$ sudo ufw deny from any to 17.178.96.59
Rule added
blackjack@blackjack:~$ sudo ufw deny from any to 17.142.160.59
Rule added
blackjack@blackjack:~$
```

4. After blocking the website on UFW



4.1 Test Results

Reconnaissance Phase

The test results of the following modules from the reconnaissance phase are mentioned below.

Initial Firewall Information Gathering

In the initial information-gathering phase NetStab gathers info about the firewall to find out possible points of vulnerabilities and appends the results in the report.

```
blackjack@blackjack:~  
blackjack@blackjack:~$ ip=$(curl ifconfig.me)  
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current  
                                 Dload  Upload   Total   Spent    Left  Speed  
100  12  100    12    0    0    26    0  --:--  --:--  --:--  26  
blackjack@blackjack:~$ echo "Network IP is: $ip"  
Network IP is: 39.41.166.27  
blackjack@blackjack:~$ sudo nmap -sX $ip | grep -w 'PORT|open|closed|filtered|MAC'  
[sudo] password for blackjack:  
Not shown: 995 closed ports  
PORT      STATE SERVICE  
21/tcp    open  filtered ftp  
22/tcp    open  filtered ssh  
23/tcp    open  filtered telnet  
80/tcp    open  filtered http  
5431/tcp  open  filtered park-agent  
blackjack@blackjack:~$ locate=$(sudo nmap -sX $ip | grep -w 'PORT|open|closed|filtered|MAC')  
blackjack@blackjack:~$ if echo "$locate" | grep -w 'closed|filtered';  
> then  
> echo "Test Verdict :Firewall Detected"  
> else  
> echo "Firewall Not Detected"  
> fi  
Not shown: 995 closed ports  
PORT      STATE SERVICE  
21/tcp    open  filtered ftp  
22/tcp    open  filtered ssh  
23/tcp    open  filtered telnet  
80/tcp    open  filtered http  
5431/tcp  open  filtered park-agent  
Test Verdict : Firewall Detected  
blackjack@blackjack:~$
```

```
/home/blackjack/NetBeansProjects/netstabgui/src/main/java/report.odt - Mousepad  
File Edit Search View Document Help  
1. Getting IP Address  
-----  
Network IP is: 39.41.230.99  
  
2. Locating Firewall  
-----  
Not shown: 995 closed ports  
PORT      STATE SERVICE  
21/tcp    open  filtered ftp  
22/tcp    open  filtered ssh  
23/tcp    open  filtered telnet  
80/tcp    open  filtered http  
5431/tcp  open  filtered park-agent  
Test Verdict :      Firewall Detected  
  
3. Discovery Scan  
-----  
39.41.230.99 : [0], 84 bytes, 41.0 ms (41.0 avg, 0% loss)  
39.41.230.99 : [1], 84 bytes, 64.1 ms (52.5 avg, 0% loss)  
39.41.230.99 : [2], 84 bytes, 86.9 ms (64.0 avg, 0% loss)  
39.41.230.99 : [3], 84 bytes, 2.03 ms (48.5 avg, 0% loss)  
39.41.230.99 : [4], 84 bytes, 2.40 ms (39.2 avg, 0% loss)  
39.41.230.99 : [5], 84 bytes, 3.92 ms (33.3 avg, 0% loss)  
39.41.230.99 : [6], 84 bytes, 2.23 ms (28.9 avg, 0% loss)  
39.41.230.99 : [7], 84 bytes, 2.46 ms (25.6 avg, 0% loss)  
Test Verdict :      Host Live
```

Banner Grabbing

```

/home/blackjack/NetBeansProjects/netstabgui/src/main/java/report.odt - Mousepad
File Edit Search View Document Help

4. Banner Grabbing
-----

Service Info:

Starting Nmap 7.80 ( https://nmap.org ) at 2020-07-04 02:28 PDT
Nmap scan report for 39.41.230.99
Host is up (0.043s latency).
Not shown: 995 closed ports
PORT      STATE SERVICE VERSION
21/tcp    open  ftp      D-Link/Comtrend DSL modem ftp firmware update
|_banner: 220 Ftp firmware update utility
22/tcp    open  ssh      Dropbear sshd 0.46 (protocol 2.0)
|_banner: SSH-2.0-dropbear_0.46
23/tcp    open  telnet   Broadcom BCM96338 DSL router telnetd
|_banner: \xFF\xFD\x01\xFF\xFD!\xFF\xFB\x01\xFF\xFB\x03
80/tcp    open  http     micro_httpd
|_banner: HTTP/1.1 400 Bad Request\x0D\x0AServer: micro_httpd\x0D\x0ACach
|_e-Contrl: no-cache\x0D\x0ADate: Mon, 03 Jan 2000 09:47:50 GMT\x0D\x0A...
5431/tcp  open  upnp     Belkin/Linksys wireless router UPnP (UPnP 1.0; BRCA00 1.0)
Service Info: OSs: Linux, Linux 2.4; Devices: broadband router, router; CPE: cpe:/o:linux:linux_kernel, cpe:/h:broadcom:bcm96338, cpe:/o:acme:micro_httpd, cpe:/o:linux:li

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 24.18 seconds

5. Port Scanning
-----

TCP port scan vulnerability report:
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
5431/tcp  open  http

NULL port scan vulnerability report:
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
80/tcp    open|filtered http
5431/tcp  open|filtered http

FIN port scan vulnerability report:
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
80/tcp    open|filtered http
5431/tcp  open|filtered park-agent

XMAS port scan vulnerability report:
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
80/tcp    open|filtered http

```

Port Scanning

```

/home/blackjack/NetBeansProjects/netstabgui/src/main/java/report.odt - Mousepad
File Edit Search View Document Help

5. Port Scanning
-----

TCP port scan vulnerability report:
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open  ftp
22/tcp    open  ssh
23/tcp    open  telnet
80/tcp    open  http
5431/tcp  open  park-agent

NULL port scan vulnerability report:
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
80/tcp    open|filtered http
5431/tcp  open|filtered park-agent

FIN port scan vulnerability report:
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
80/tcp    open|filtered http
5431/tcp  open|filtered park-agent

XMAS port scan vulnerability report:
Not shown: 995 closed ports
PORT      STATE SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
80/tcp    open|filtered http

```

Attack Phase

Spoofed IP Packets and Custom Packet Injection

```
File Edit Search View Document Help
/home/blackjack/NetBeansProjects/netstabgui/src/main/java/report.odt - Mousepad
*** port scan vulnerability report ***
Not shown: 995 closed ports
PORT      STATE      SERVICE
21/tcp    open|filtered ftp
22/tcp    open|filtered ssh
23/tcp    open|filtered telnet
80/tcp    open|filtered http
5431/tcp  open|filtered park-agent

6. IP Spoofing
-----
HPING 39.41.230.99 (wlan0 39.41.230.99): S set, 40 headers + 0 data bytes
Receiving Custom Spoofed IP Packets.

7. Custom Packets tests
-----

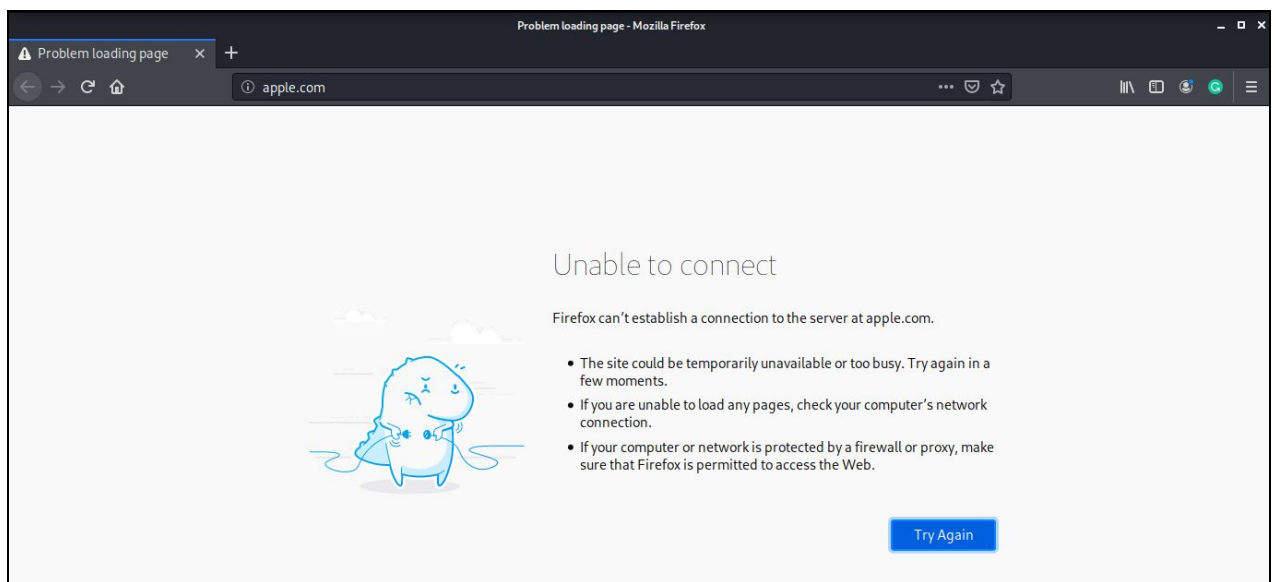
Sending Random Source Packets:
HPING 39.41.230.99 (wlan0 39.41.230.99): S set, 40 headers + 0 data bytes

Sending Packets Flaged as Urgent:
HPING 39.41.230.99 (wlan0 39.41.230.99): SU set, 40 headers + 0 data bytes
len=40 ip=39.41.230.99 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=7.7 ms

Sending Push Flag Enabled Packets:
HPING 39.41.230.99 (wlan0 39.41.230.99): SP set, 40 headers + 0 data bytes
len=40 ip=39.41.230.99 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=15.6 ms
```

SSH Tunneling

Blocked apple.com



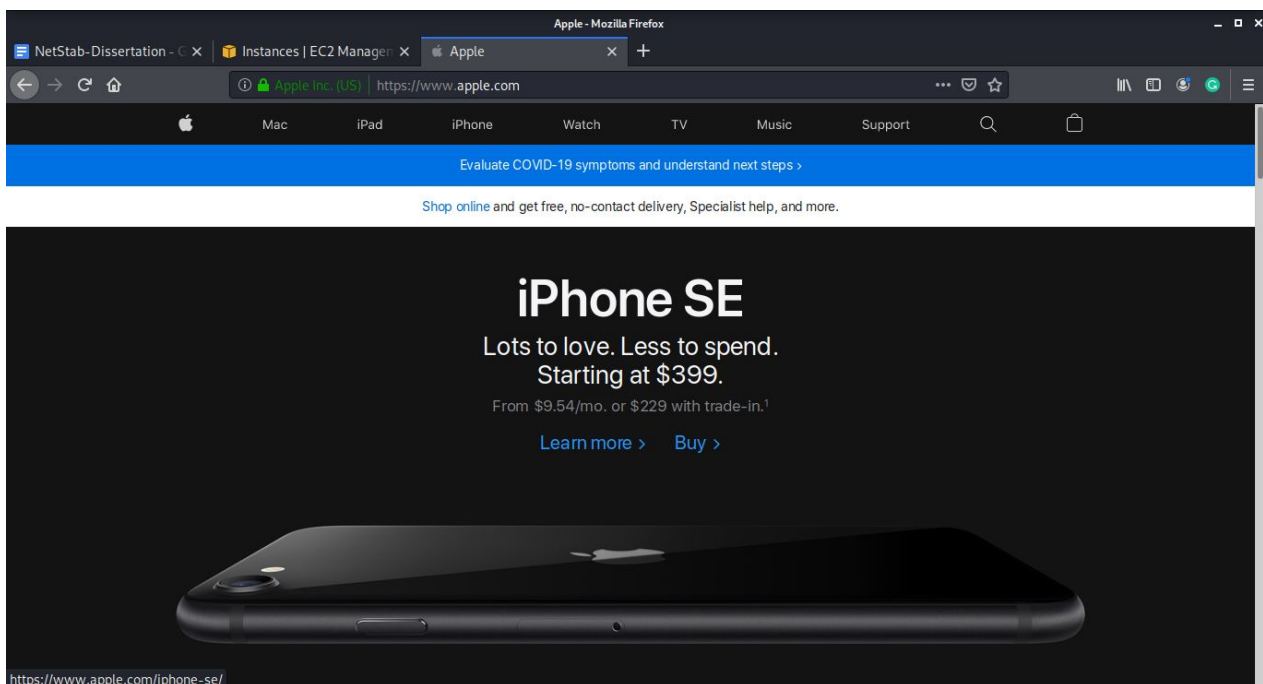
Initializing SSH Tunnel

```
ec2-user@ip-172-31-41-3:~  
blackjack@blackjack:~/NetBeansProjects/netstabgui$ ssh -i "netstab.pem" ec2-user@ec2-3-134-29-204.us-east-2.compute.amazonaws.com  
Last login: Sat Jul  4 17:22:55 2020 from 39.41.230.99  
  
┌─┴─┬─┐  
└─┬─┴─┘  Amazon Linux 2 AMI  
  
https://aws.amazon.com/amazon-linux-2/  
12 package(s) needed for security, out of 21 available  
Run "sudo yum update" to apply all updates.  
[ec2-user@ip-172-31-41-3 ~]$
```

Socks Proxy is enabled on the SSH port to establish an SSH tunnel.

```
blackjack@blackjack: ~/NetBeansProjects/netstabgui  
blackjack@blackjack:~/NetBeansProjects/netstabgui$ bash proxy_set.sh  
blackjack@blackjack:~/NetBeansProjects/netstabgui$
```

Blocked website (apple.com) is successfully accessed.



HTTP Tunneling follows the same procedure, but they only differ in setting the proxy setting where ssh tunneling and port redirection uses socks proxy and HTTP tunneling uses HTTP proxy. SSH uses ssh port as a source port for establishing the tunnel and port direction can use any possible combination of the port to form the tunnel. Although for all three of them NetStab is using dynamic port forwarding technique as it is the most reliable technique for tunneling.

ICMP Tunneling

Initializing ICMP Tunnel server

```
blackjack@blackjack: ~/NetBeansProjects/netstabgui
blackjack@blackjack:~/NetBeansProjects/netstabgui$ bash runICMP.sh
7
[sudo] password for blackjack:
bash: cannot set terminal process group (-1): Inappropriate ioctl for device
bash: no job control in this shell
ubuntu@ip-172-31-21-174:~$ #!/bin/bash/
ubuntu@ip-172-31-21-174:~$
ubuntu@ip-172-31-21-174:~$ killall icmpntunnel
icmpntunnel: no process found
ubuntu@ip-172-31-21-174:~$ cd icmpntunnel
ubuntu@ip-172-31-21-174:~/icmpntunnel$ sudo ./icmpntunnel -s -d
opened tunnel device: tun0
< /sbin/ifconfig tun0 10.0.0.1 netmask 255.255.255.0
ubuntu@ip-172-31-21-174:~/icmpntunnel$ exit
```

Initializing client side of the ICMP Tunnel

```
ubuntu@ip-172-31-21-174:~$
blackjack@blackjack:~/NetBeansProjects/netstabgui$ bash icmpclient.sh
icmpntunnel: no process found
0
[sudo] password for blackjack:
opened tunnel device: tun0
connection established.
Welcome to Ubuntu 18.04.4 LTS (GNU/Linux 5.3.0-1028-aws x86_64)

 * Documentation:  https://help.ubuntu.com
 * Management:    https://landscape.canonical.com
 * Support:        https://ubuntu.com/advantage

System information as of Sat Jul  4 17:40:31 UTC 2020

System load:  0.07          Processes:    109
Usage of /:   36.7% of 7.69GB   Users logged in:  0
Memory usage: 19%           IP address for eth0: 172.31.21.174
Swap usage:  0%             IP address for tun0: 10.0.0.1

 * "If you've been waiting for the perfect Kubernetes dev solution for
  macOS, the wait is over. Learn how to install Microk8s on macOS."
  https://www.techrepublic.com/article/how-to-install-microk8s-on-macos/

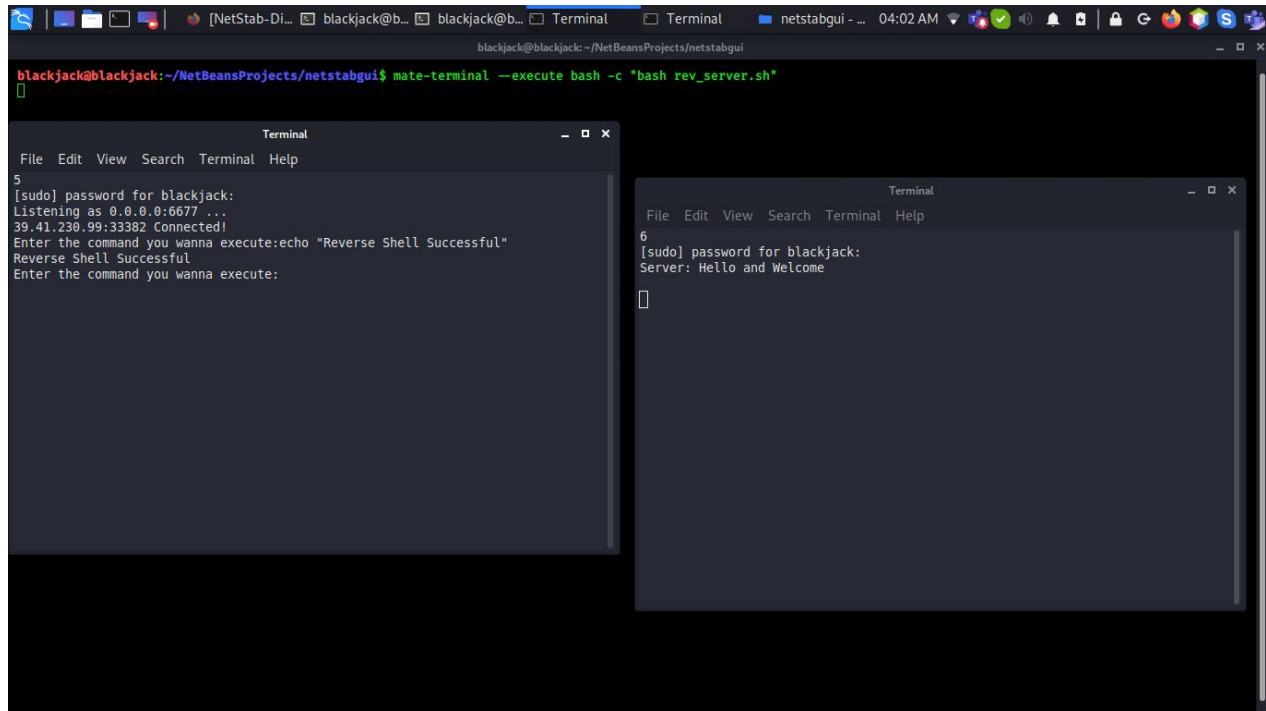
 * Canonical Livepatch is available for installation.
   - Reduce system reboots and improve kernel security. Activate at:
     https://ubuntu.com/livepatch

54 packages can be updated.
0 updates are security updates.

Last login: Thu Jul  2 11:16:27 2020 from 10.0.0.2
ubuntu@ip-172-31-21-174:~$
```

Reverse Shell

Initializing the Reverse shell and server. Commands executed to confirm if the reverse shell is established or not.

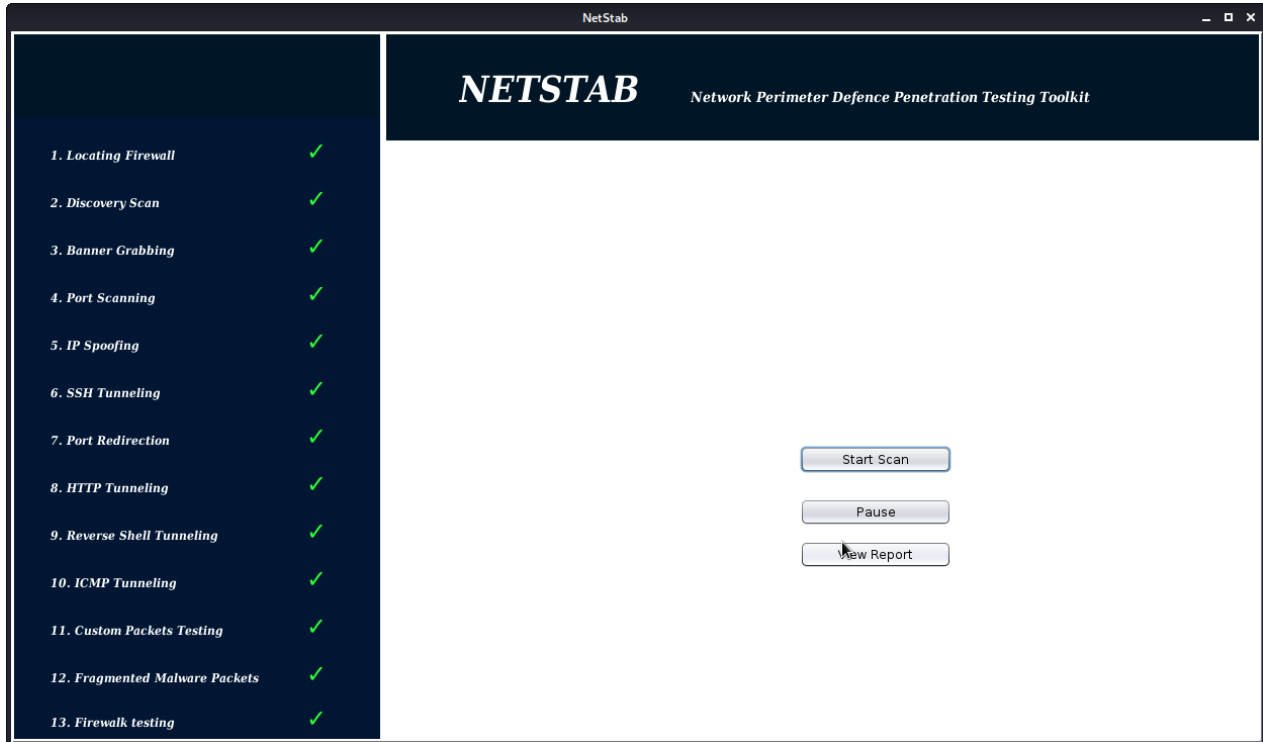


```
blackjack@blackjack:~/NetBeansProjects/netstabgui$ mate-terminal --execute bash -c "bash rev_server.sh"
5
[sudo] password for blackjack:
Listening as 0.0.0.0:6677 ...
39.41.230.99:33382 Connected!
Enter the command you wanna execute:echo "Reverse Shell Successful"
Reverse Shell Successful
Enter the command you wanna execute:

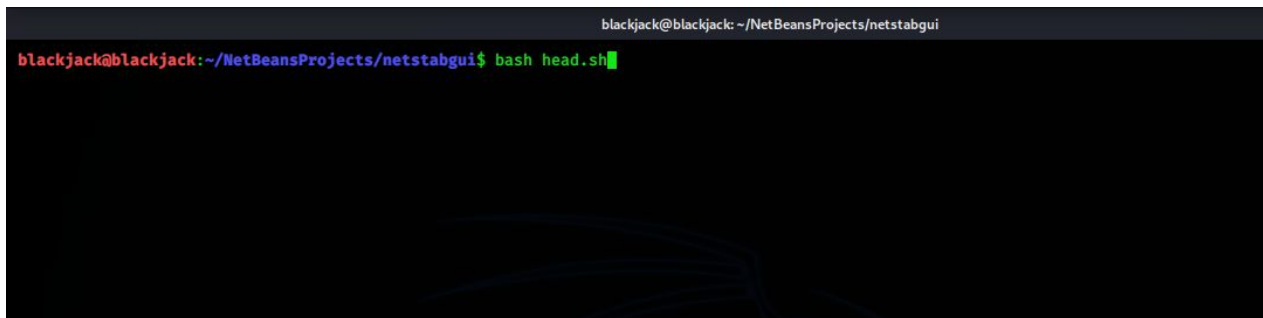
6
[sudo] password for blackjack:
Server: Hello and Welcome
```

Overall Testing and Report Generation

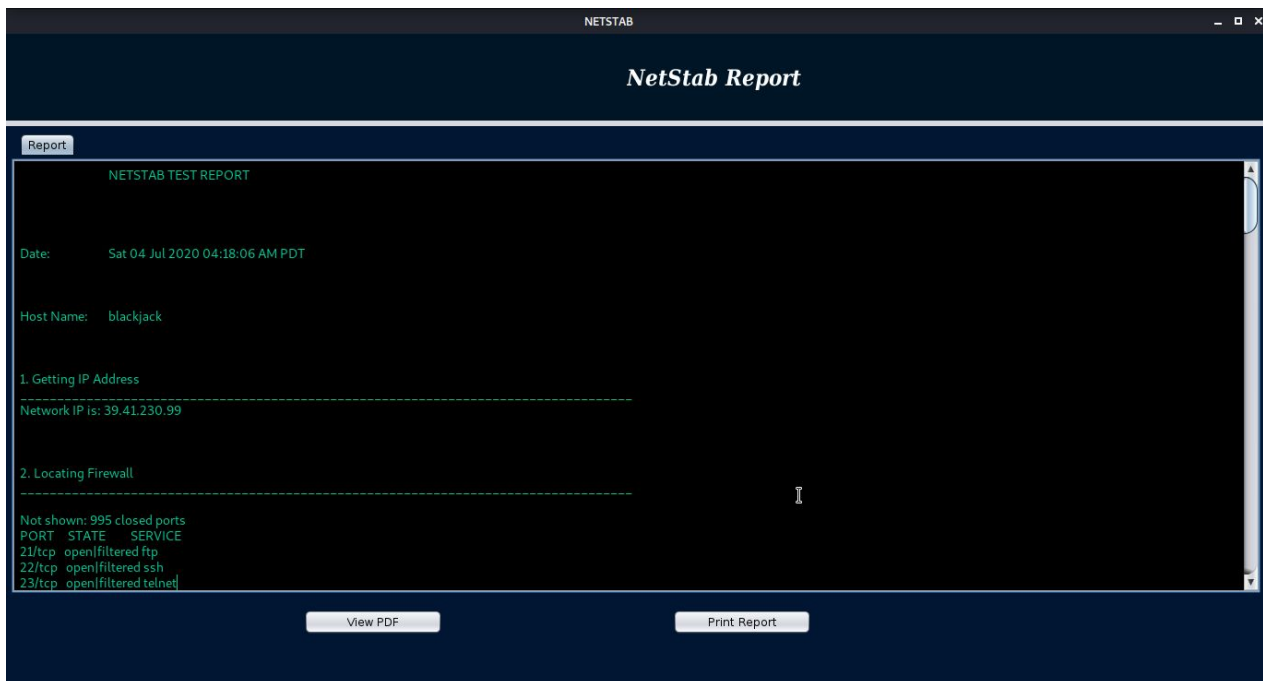
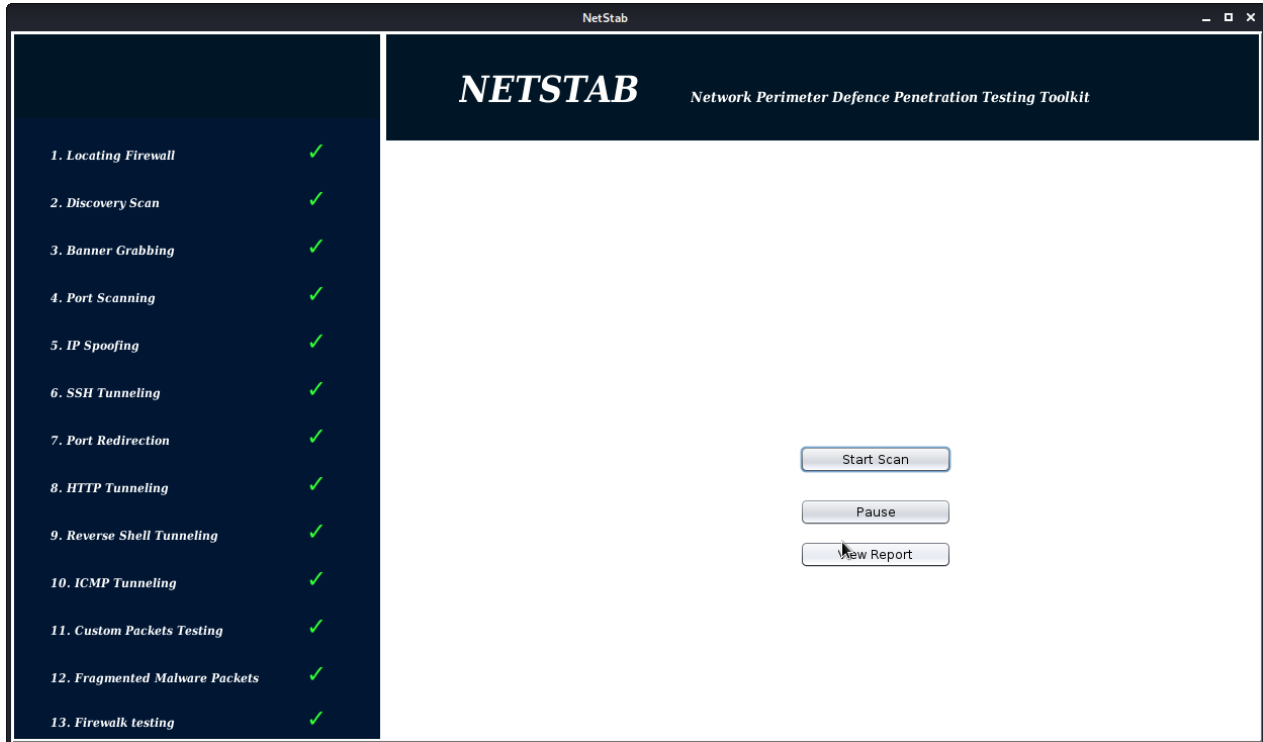
After login initializing the scan.



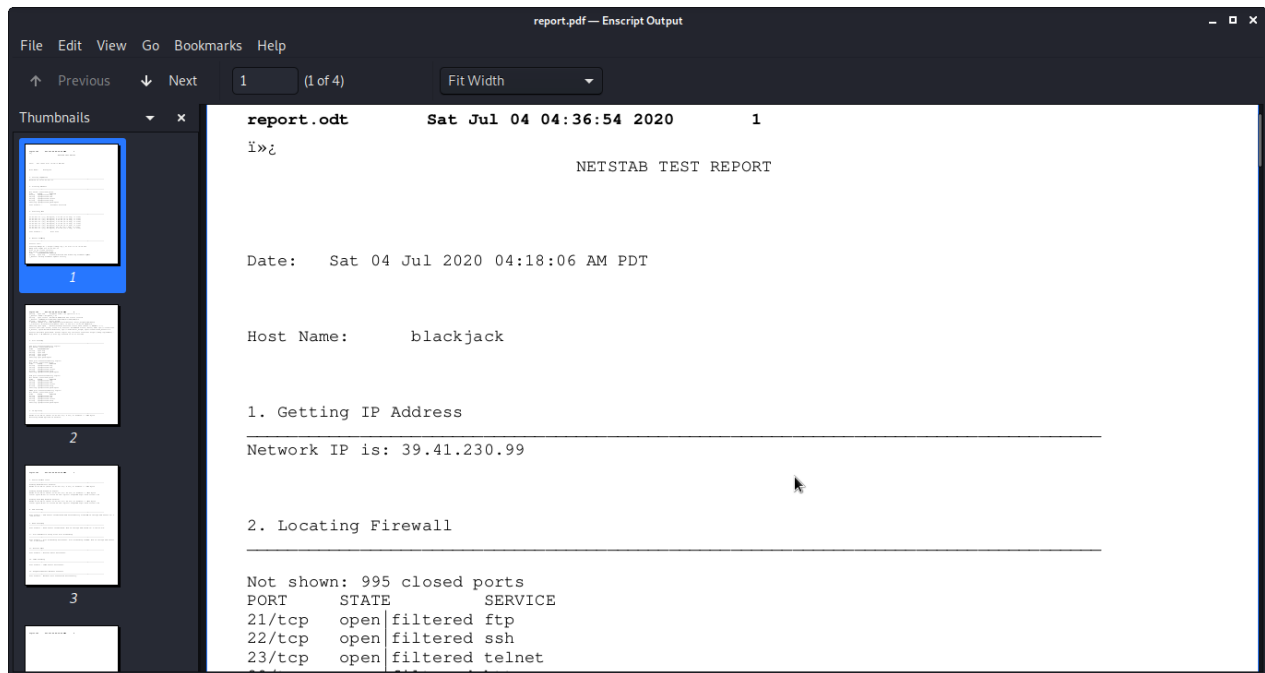
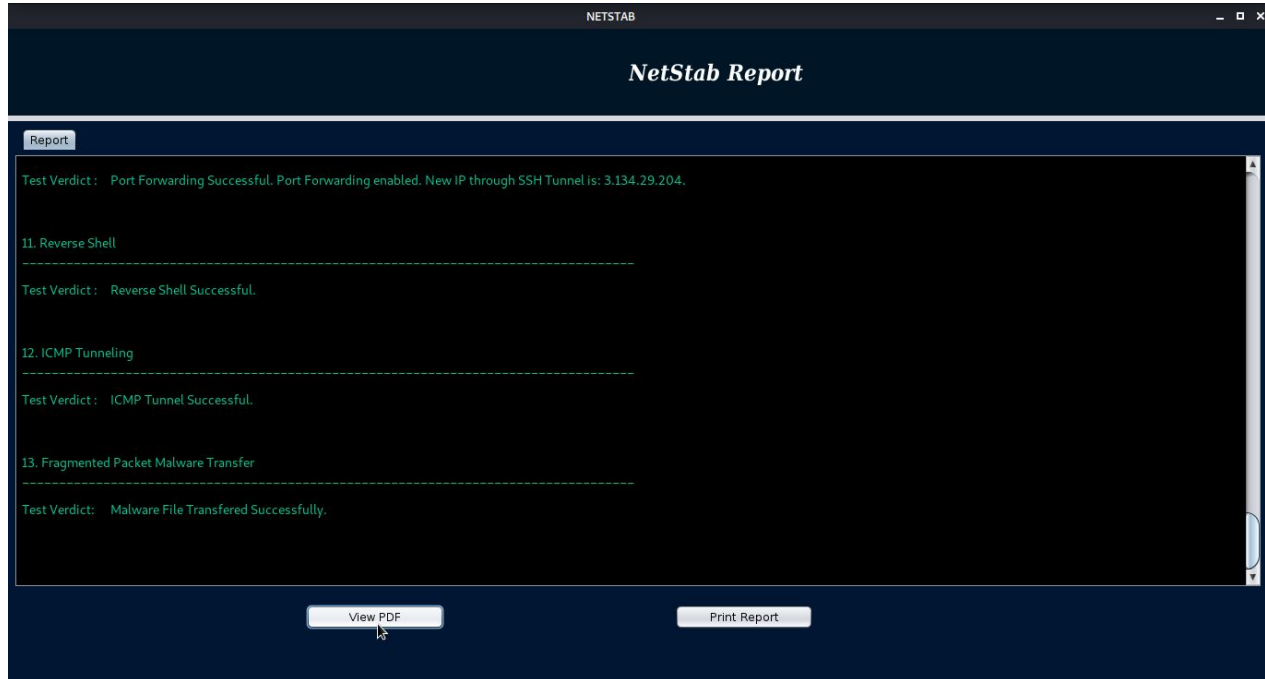
Run the head.sh (a bash script that controls all modules)



After the test concludes, view the report.



View PDF button displays the PDF version of the report.



4.2 Test Report

NETSTAB TEST REPORT

Date: Sat 04 Jul 2020 04:18:06 AM PDT

Host Name: blackjack

1. Getting IP Address

Network IP is: 39.41.230.99

2. Locating Firewall

Not shown: 995 closed ports

PORT	STATE	SERVICE
21/tcp	open filtered	ftp
22/tcp	open filtered	ssh
23/tcp	open filtered	telnet
80/tcp	open filtered	http
5431/tcp	open filtered	park-agent

Test Verdict : Firewall Detected

3. Discovery Scan

39.41.230.99 : [0], 84 bytes, 1.97 ms (1.97 avg, 0% loss)

39.41.230.99 : [1], 84 bytes, 2.05 ms (2.01 avg, 0% loss)

39.41.230.99 : [2], 84 bytes, 2.22 ms (2.08 avg, 0% loss)

39.41.230.99 : [3], 84 bytes, 1.91 ms (2.03 avg, 0% loss)

39.41.230.99 : [4], 84 bytes, 2.03 ms (2.03 avg, 0% loss)

39.41.230.99 : [5], 84 bytes, 2.23 ms (2.06 avg, 0% loss)

39.41.230.99 : [6], 84 bytes, 120 ms (19.0 avg, 0% loss)

39.41.230.99 : [7], 84 bytes, 2.13 ms (16.9 avg, 0% loss)

Test Verdict : Host Live

4. Banner Grabbing

Service Info:

Starting Nmap 7.80 (<https://nmap.org>) at 2020-07-04 04:18 PDT

Nmap scan report for 39.41.230.99

Host is up (0.025s latency).

Not shown: 995 closed ports

PORT	STATE	SERVICE	VERSION
------	-------	---------	---------

21/tcp	open	ftp	D-Link/Comtrend DSL modem ftp firmware update
--------	------	-----	---

|_banner: 220 Ftp firmware update utility

22/tcp	open	ssh	Dropbear sshd 0.46 (protocol 2.0)
--------	------	-----	-----------------------------------

|_banner: SSH-2.0-dropbear_0.46

23/tcp	open	telnet	Broadcom BCM96338 DSL router telnetd
--------	------	--------	--------------------------------------

|_banner: \xFF\xFD\x01\xFF\xFD!\xFF\xFB\x01\xFF\xFB\x03

80/tcp	open	http	micro_httpd
--------	------	------	-------------

| banner: HTTP/1.1 400 Bad Request\x0D\x0AServer: micro_httpd\x0D\x0ACach

|_e-Control: no-cache\x0D\x0ADate: Mon, 03 Jan 2000 11:37:54 GMT\x0D\x0A...

5431/tcp	open	upnp	Belkin/Linksys wireless router UPnP (UPnP 1.0; BRCM400 1.0)
----------	------	------	---

Service Info: OSs: Linux, Linux 2.4; Devices: broadband router, router; CPE: cpe:/o:linux:linux_kernel, cpe:/h:broadcom:bcm96338, cpe:/o:acme:micro_httpd, cpe:/o:linux:linux_kernel:2.4

Service detection performed.

5. Port Scanning

TCP port scan vulnerability report:

Not shown: 995 closed ports

PORT	STATE	SERVICE
21/tcp	open	ftp
22/tcp	open	ssh
23/tcp	open	telnet
80/tcp	open	http
5431/tcp	open	park-agent

NULL port scan vulnerability report:

Not shown: 995 closed ports

PORT	STATE	SERVICE
21/tcp	open filtered	ftp
22/tcp	open filtered	ssh
23/tcp	open filtered	telnet
80/tcp	open filtered	http
5431/tcp	open filtered	park-agent

FIN port scan vulnerability report:

PORT	STATE	SERVICE
21/tcp	open filtered	ftp
22/tcp	open filtered	ssh
23/tcp	open filtered	telnet
80/tcp	open filtered	http
5431/tcp	open filtered	park-agent

XMAS port scan vulnerability report:

Not shown: 995 closed ports

PORT	STATE	SERVICE
21/tcp	open filtered	ftp
22/tcp	open filtered	ssh
23/tcp	open filtered	telnet
80/tcp	open filtered	http
5431/tcp	open filtered	park-agent

6. IP Spoofing

HPING 39.41.230.99 (wlan0 39.41.230.99): S set, 40 headers + 0 data bytes

Receiving Custom Spoofed IP Packets.

7. Custom Packets tests

Sending Random Source Packets:

HPING 39.41.230.99 (wlan0 39.41.230.99): S set, 40 headers + 0 data bytes

Sending Packets Flagged as Urgent:

HPING 39.41.230.99 (wlan0 39.41.230.99): SU set, 40 headers + 0 data bytes

len=40 ip=39.41.230.99 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=23.6 ms

Sending Push Flag Enabled Packets:

HPING 39.41.230.99 (wlan0 39.41.230.99): SP set, 40 headers + 0 data bytes

len=40 ip=39.41.230.99 ttl=64 DF id=0 sport=0 flags=RA seq=0 win=0 rtt=35.9 ms

8. SSH Tunneling

Test Verdict : SSH Tunnel established.SSH Vulnerability found. New IP through SSH Tunnel is: 3.134.29.204.

9. HTTP Tunneling

Test Verdict : HTTP Tunnel established. New IP through SSH Tunnel is: 3.134.29.204.

10. Port Redirection using Local Port Forwarding

Test Verdict : Port Forwarding Successful. Port Forwarding enabled. New IP through SSH Tunnel is: 3.134.29.204.

11. Reverse Shell

Test Verdict : Reverse Shell Successful.

12. ICMP Tunneling

Test Verdict : ICMP Tunnel Successful.

13. Fragmented Packet Malware Transfer

Test Verdict : Malware File Transferred Successfully.

V. Future Work

NetStab is a Network Perimeter defense penetration toolkit and the scope of the project is engulfed around firewall pen-testing procedures primarily. But it has the capability to have extended areas of further discovery, experimentation, and improvement.

In the foreseeable future work, NetStab can extend towards the areas of :

1. Intrusion Detection System (IDS) Pen-testing Automation

Intrusion Detection System (IDS) is another unit of Network Perimeter Defense that is responsible for alarming the network administrator in case of an intrusion in the network.

As an added functionality we can add penetration testing for IDS in NetStab to ensure if the IDS is efficient enough to detect intrusions.

2. Intrusion Prevention System (IPS) Pen-testing Automation

Intrusion Prevention System (IPS) is another layer of protection for networks and it is a part of Network Perimeter Defense. An IPS is responsible for safeguarding the network from possible intrusions, similar to a firewall. As an added functionality we can add penetration testing for IPS in NetStab to ensure if the IPS is efficient enough to detect and interrupt intrusions.

3. As an extension for Firewall packages

NetStab can be included as an extension of firewall packages that checks the firewall itself as an added measure and reflects the findings to the network administrator on a regular basis so that steps can be taken to improve the firewall efficiency.

VI. Conclusion

The objective of this project was to produce a novel solution for the network pen-testers and administrators to aid their firewall pen-testing procedure so that time and resources can be saved.

The idea was to aid the pentester's work or process of firewall penetration testing. Network Pentester or Network Administrator can use this report as a reference to deal with the loopholes in the network rather than going through the rigorous process of pentesting the network first. He can simply generate a report and start fixing the network shortcomings right away.

During this tenure, we came across many hurdles to create this tool that can make firewall penetration testing much easier and efficient by automating most of the portion of this process.

Although utility programs, tools, and libraries were still there, a huge amount of work and research was done in order to give the most efficient solution possible. After ages, long research and best possible solutions for testing and reporting were selected to be included in the modules.

A well-orchestrated methodology was contrived to detect possible vulnerability origins in a network. The purpose is to ensure that network firewalls are easy to reconfigure to provide better protection and they are easy to improve as well.

Appendix A - User Manual

About NetStab

NetStab is a state of the art software-based toolkit that launches penetration tests on a Network Firewall to evaluate its efficiency, performance and guide the network administrator to fill up the loopholes. It is an indigenous Firewall Penetration Testing Toolkit incorporating firewall evading techniques and penetration modules to confirm the efficiency of the firewall configured to secure the network. NetStab will strengthen firewalls through its vulnerability assessment tests, therefore, preventing network infiltration, data leakage, and preventing hackers from accessing sensitive information.

Prerequisites

Following are the requirements for NetStab to generate its pen-testing report:

1. A system with Kali Linux
2. Network Access
3. Bash (installed on the Kali Linux System)
4. Python3 (installed on the Kali Linux System)
5. Java (installed on the Kali Linux System)

GUI guide

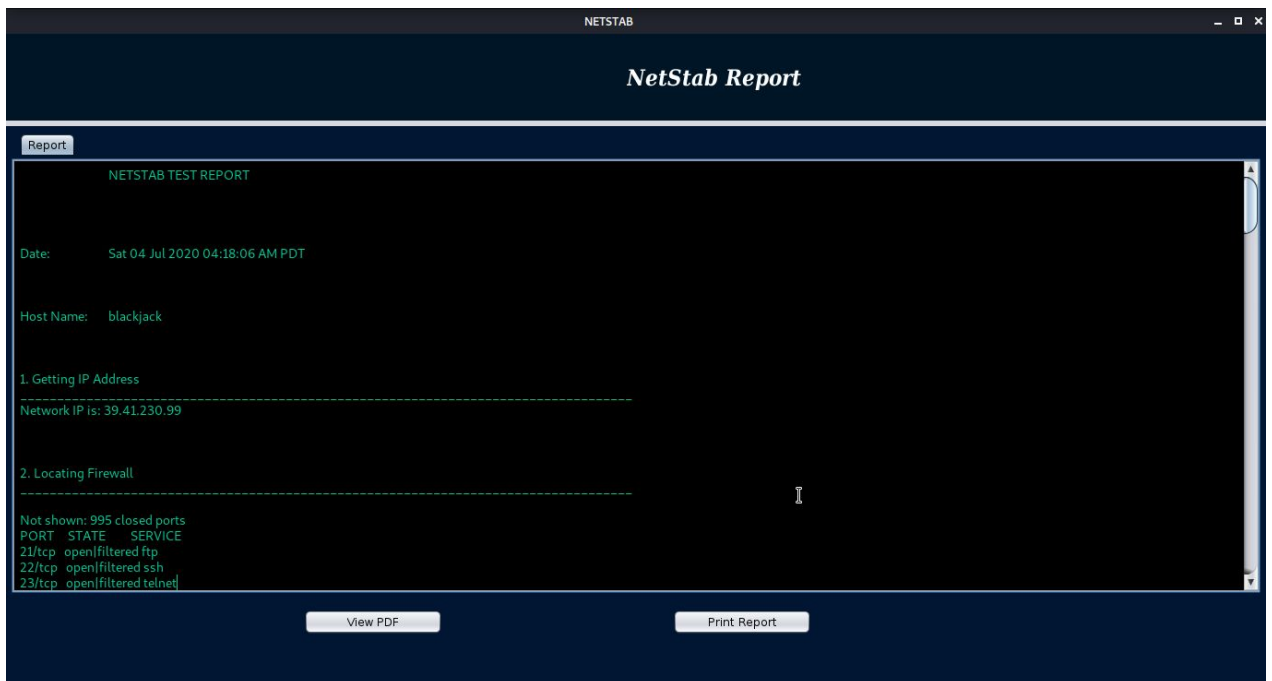
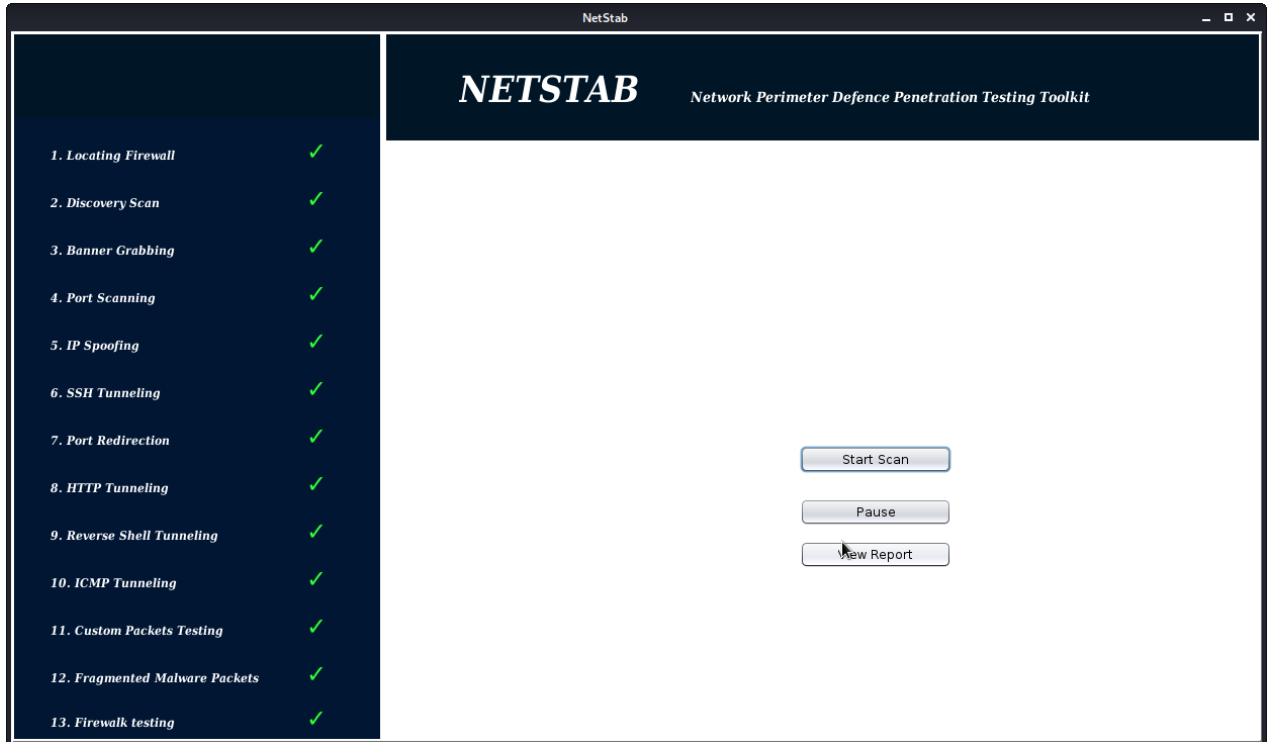
Login Window



The image shows a screenshot of a web browser window titled "NETSTAB". The window has a dark background and contains the following elements:

- The title "NETSTAB" is centered at the top in a large, bold, white serif font.
- Below the title, there are two text input fields. The first is labeled "Username" and contains the text "netstab". The second is labeled "Password" and contains seven asterisks "*****".
- Below the password field, there is a "Sign In" button.
- Below the "Sign In" button, there is a "Not registered?" label and a "Sign Up" button.

The login window has two text fields to enter the Username and Password. A Sign in Button to sign in to the application and a Sign-up button for creating an account or register.

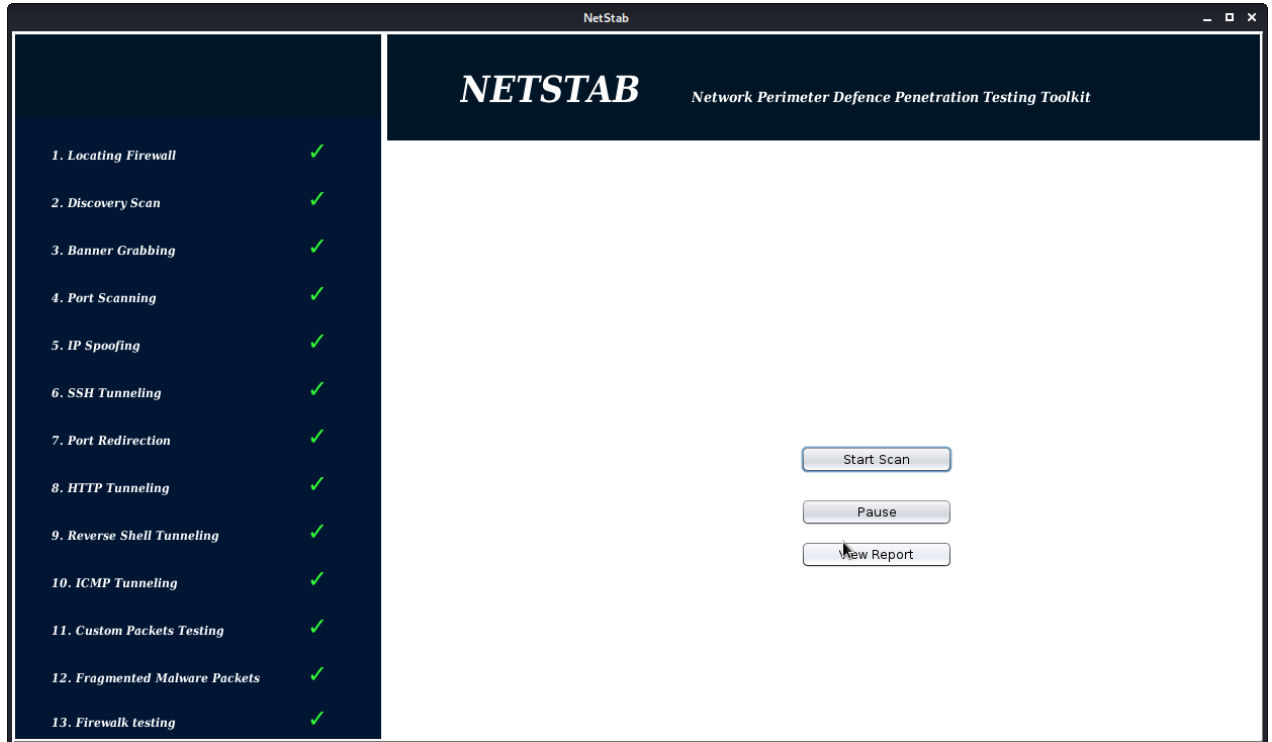


Instructions for Use

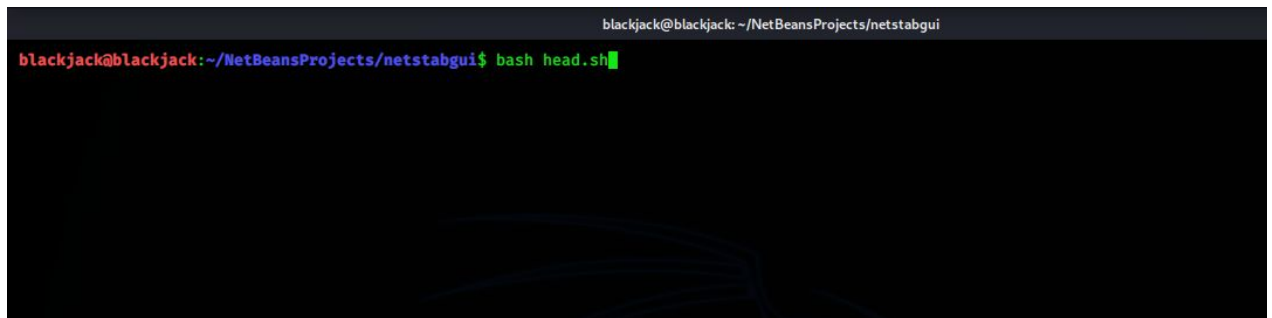
1. Run the application.
2. Enter login credentials provided at the time of purchase.



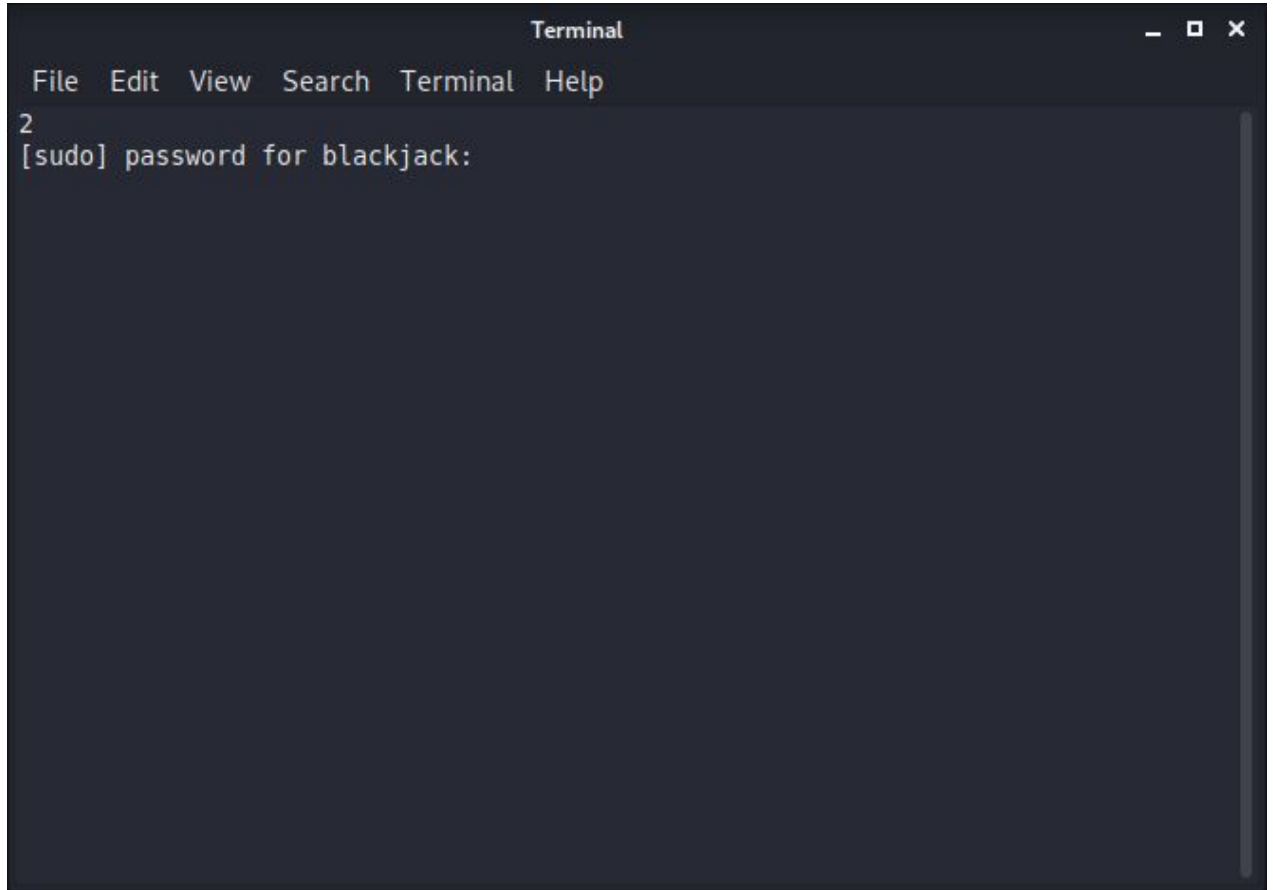
3. Click the **Start Scan** button to begin Scanning.



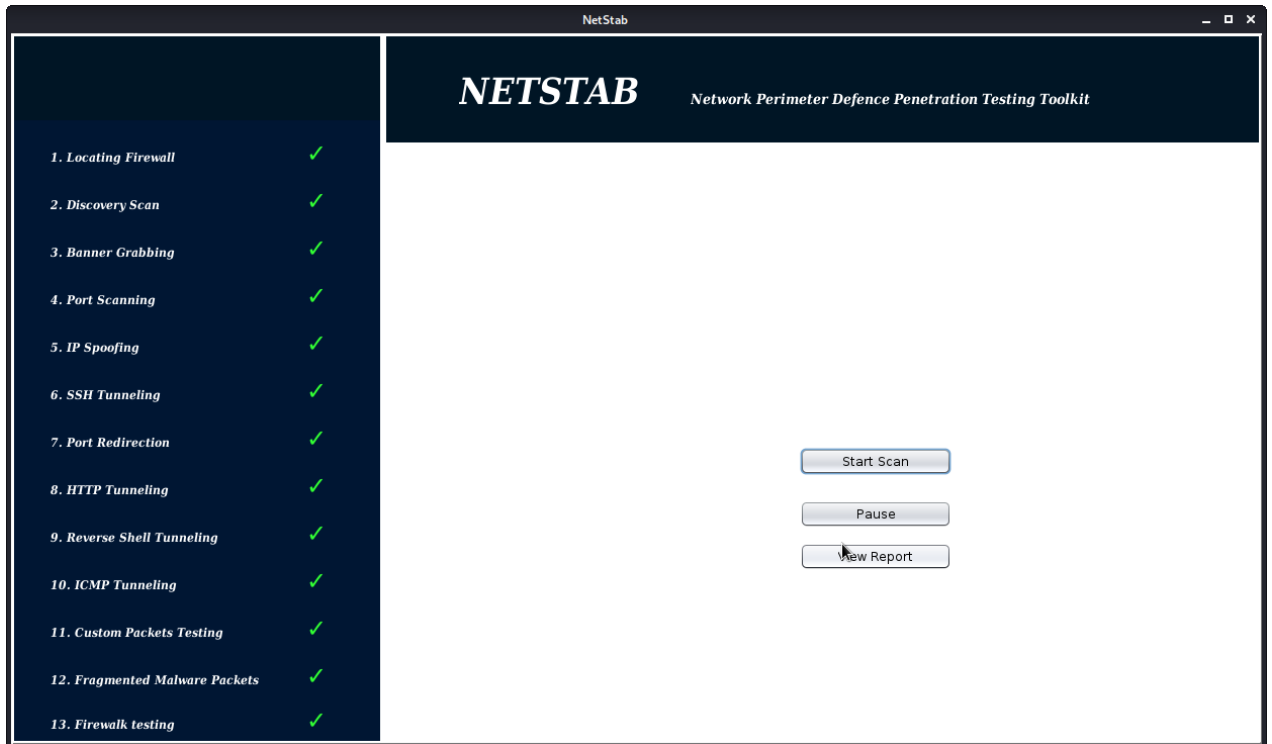
4. A terminal will open, type **head.sh** and press Enter key.



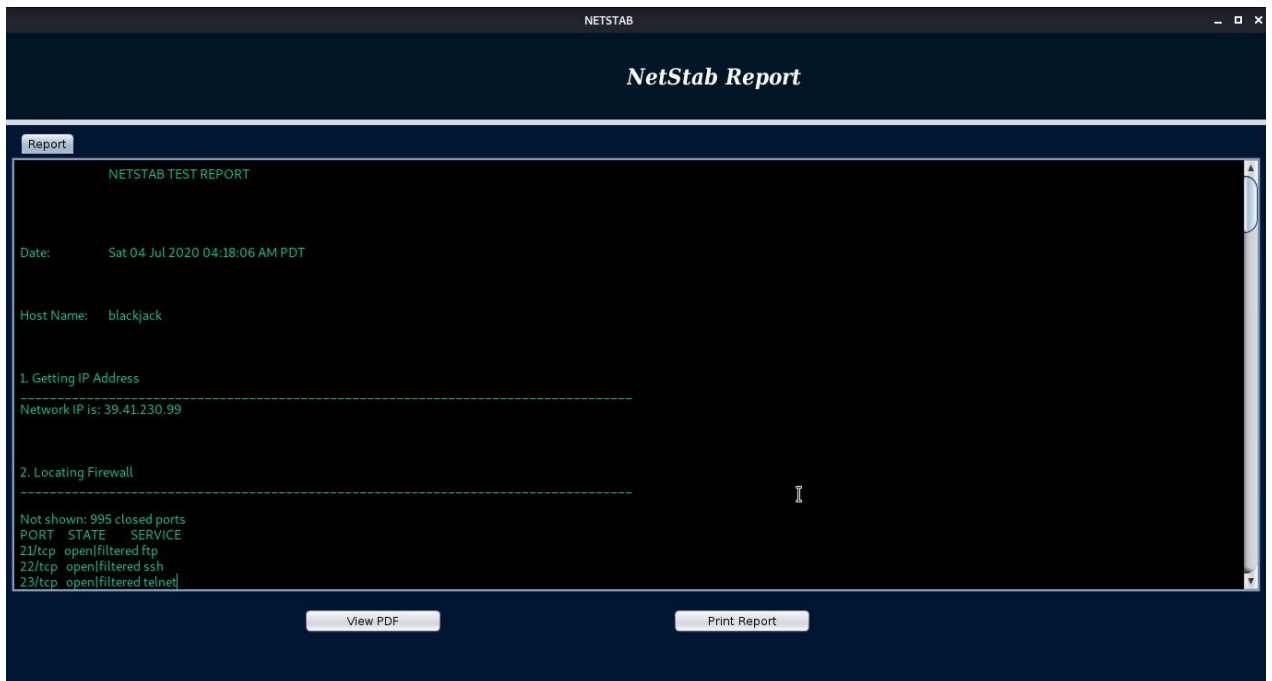
5. Subterminal windows will open, follow the serial numbers, and enter your password for the sudo account.

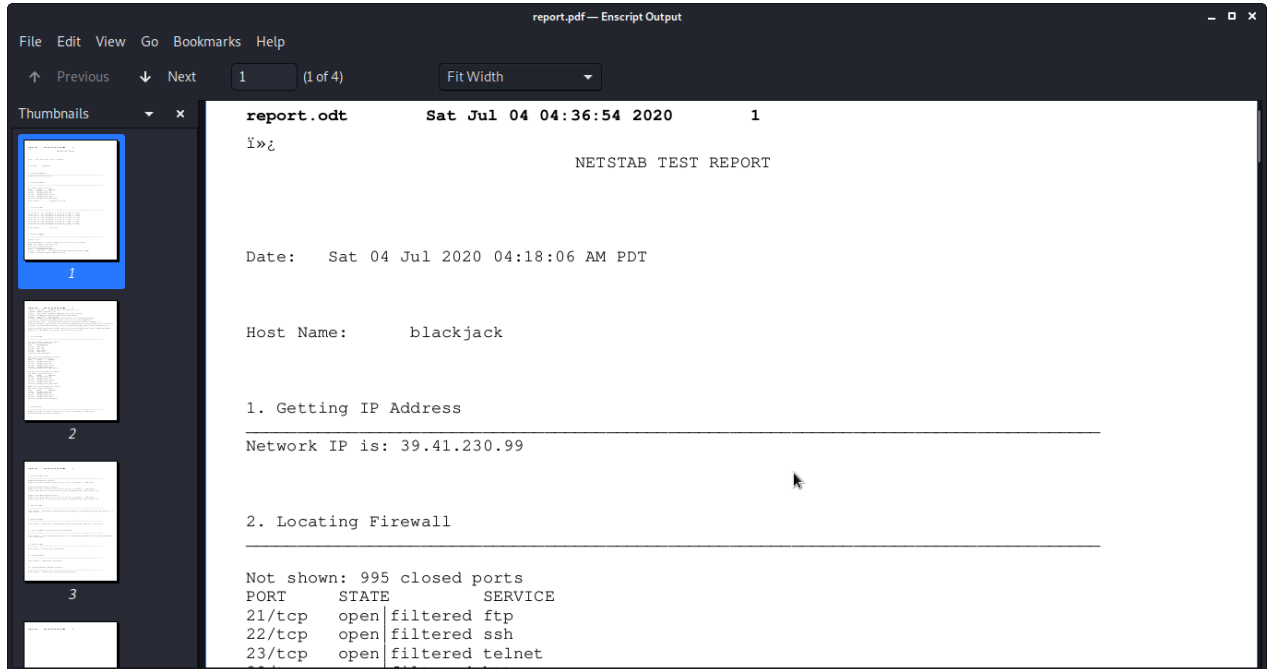


6. Once the terminal is cleared close the terminal window and click View **Report Button**.



7. Now you can examine the report in the Report Window or you can click the **View PDF** button to open the report stored in PDF form.





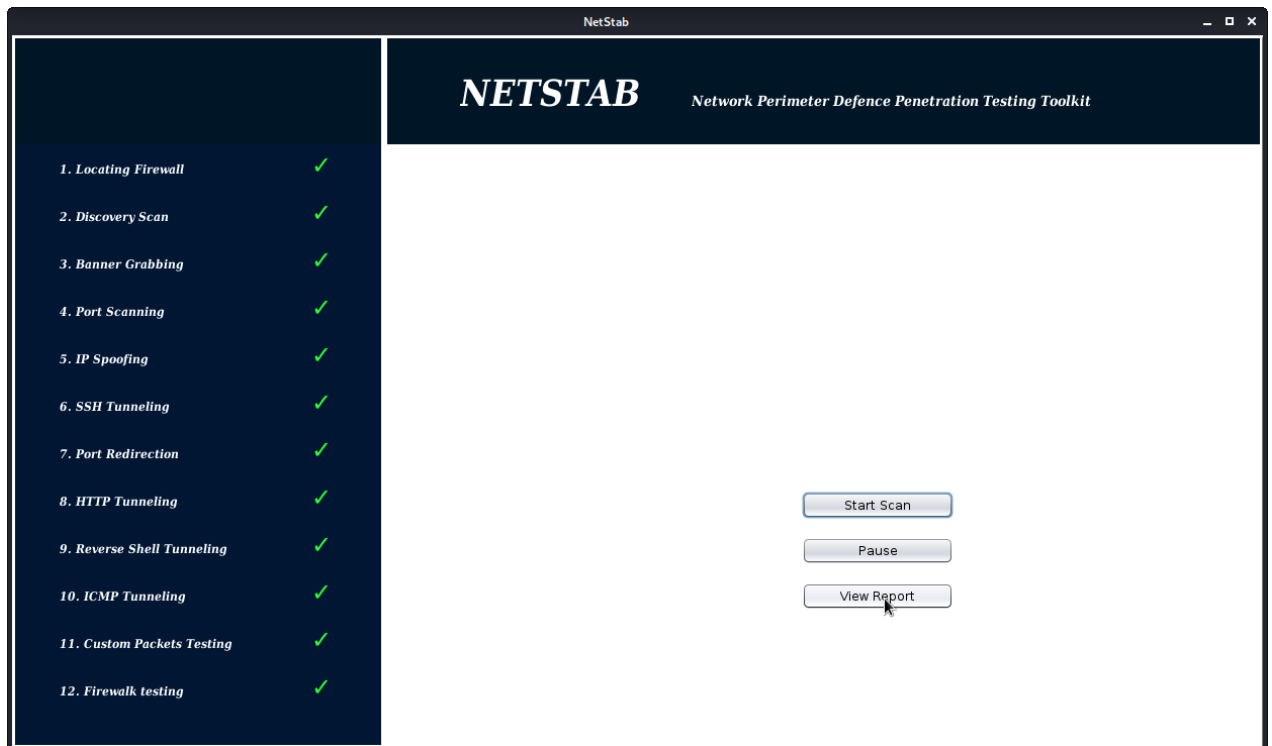
8. Click the **Print Report** button to print the report.



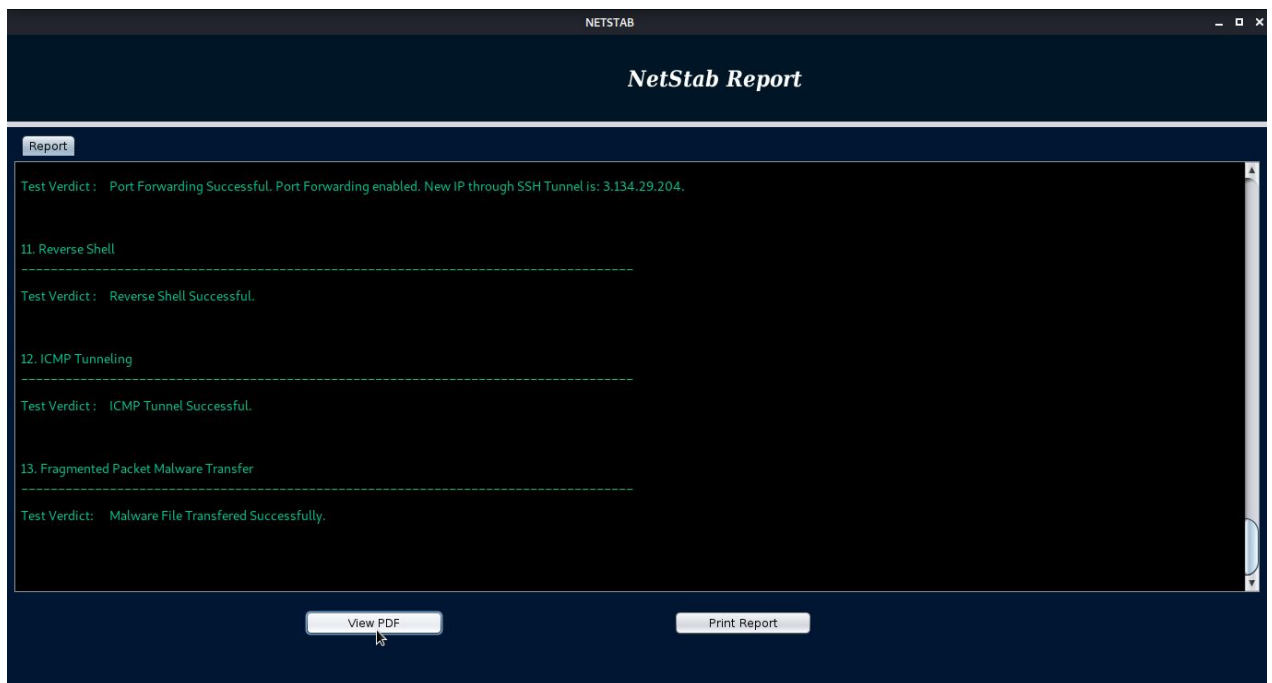
Note: Closing a single window will close all of the windows

View the Previous Report

1. Run the application.
2. Enter login credentials provided at the time of purchase.
3. Click the **Report Button** button to review the previous scan report.



4. Now you can examine the report in the Report Window or you can click the **View PDF** button to open the report stored in PDF form.



5. Click the **Print Report** button to print the report.

Bibliography

- [1]. Alex X. Liu, Member, IEEE, and Mohamed G. Gouda, Member, IEEE, (2008). "*Diverse Firewall Design.*" IEEE Transactions On Parallel And Distributed Systems.
- [2]. Brand, Murray, (2007). "*A Comprehensive Firewall Testing Methodology.*" Proceedings of The 5th Australian Information Security Management Conference 2-13.
- [3]. Cho Hong, LING, (2006). "*Internet Firewall Vulnerability Analysis Method.*" Department of Computer Science, University of Auckland.
- [4]. Cole, E., Krutz, R., Conley, J.W., (2005). "*Network Security Bible*", Wiley Publishing, Inc., Indianapolis.
- [5]. E. Schultz, (1996). "*How to perform effective firewall testing,*" Computer Security Journal, vol. 12,
- [6]. Elizabeth D. Zwicky, Simon Cooper & D. Brent Chapman, (2000). "*Building Internet Firewalls.*"
- [7]. Haeni, Reto E., (1997). "*Firewall Penetration Testing.*" The George Washington University.
- [8]. Kuang Chu, (2005). "*Network security and firewall technology*", Chongqing University Publishing House.

-
- [9]. Miss. Shwetambari G. Pundkar¹, Prof. Dr. G. R. Bamnote². (2014). "*Analysis of Firewall Technology In Computer Network Security*." International Journal of Computer Science and Mobile Computing 841-846.
- [10]. Seny Kamara, Sonia Fahmy, Eugene Schultz, Florian Kerschbaum, and Michael Frantzen, (2005). "*Analysis of Vulnerabilities in Internet Firewalls*." Center for Education and Research in Information Assurance and Security (CERIAS), Purdue University.