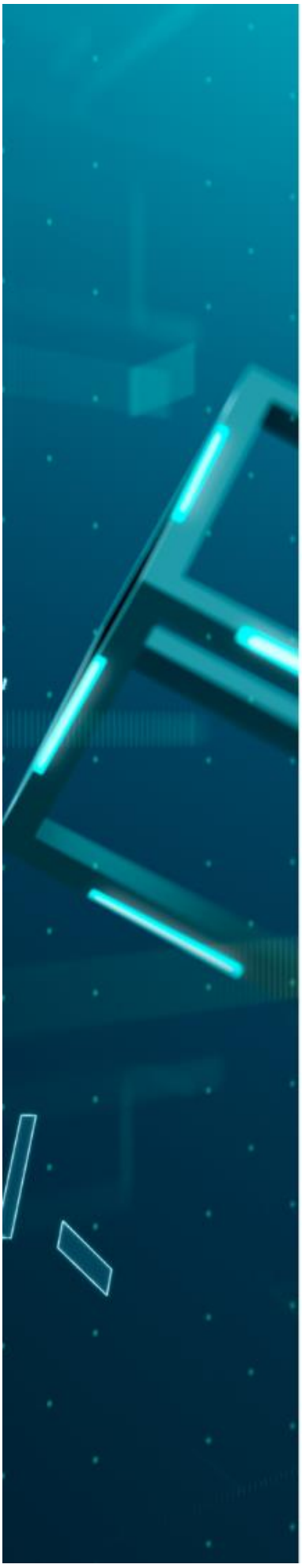
Design and implementation of honeypot focusing on session redirection

A thesis defense presentation by Samarth Desai

THESIS DEFENSE



contents

Abstract

A honeypot architecture will be implemented with other systems in the infrastructure for use. Obtaining new knowledge on the access skill of intruders allows us to improve security and protect against future attacks. In this project, the firewall redirects a session from an abnormal user to honeypots, in order to learn more advanced attacks and to respond more efficiently.

Literature review

Project background

During the last few years, many different uses of honeypots have been proposed. Some of them are deployed to waste hackers' time, others to reduce spam activity or to deceive attackers, and some to analyse hacker intrusion steps. For several years, the security community has used honeypots to analyse different techniques deployed by attackers.

Current works on this

1. How should open-source technologies be used to dynamically add or modify hacking incidents in a high-interaction honeynet system?
2. How should honeypots be made more attractive for hackers to spend more time to provide hacking evidence?

Objectives

The objectives of this experiment are:
(1) to use free and open-source technologies and methods to reduce the amount of manual intervention needed to add to or modify a honeypot system suitable for academic research, and
(2) to detect attack patterns on services and find a solution to mitigate the attacks

Samarth desai

Problem statement

Attempts by attackers to breach security systems are rising every day. Intruders use tools like Sub Seven, Nmap and Loft Crack to scan, identify, probe and penetrate an enterprise system. Firewalls are put in place to prevent such unauthorized access to the enterprise networks. However, Firewalls cannot prevent attacks coming from Intranet. An Intrusion Detection System (IDS) reviews network traffic and identifies exploits and vulnerabilities; it is able to display alerts, log events, and e-mail administrators of possible attacks. An Intrusion Prevention System on the other hand makes attempts to prevent known intrusion signatures and some unknown attacks due to the knowledge of attack behaviours in its database. However, an IDS can generate thousands of intrusions alerts every day, some of which are false positives. This makes it difficult for an IDS to detect and identify the actual threats and to protect assets. Thus, human intervention is required to investigate the attacks detected and reported by an IDS.

Definition of terms

Honeypot:
Honeypot Systems are decoy servers or systems setup to gather information about attackers who intrude into a system.

Honeynet:
A collection of honeypots forming a network.

Firewall:
A firewall is a system developed with the intention of preventing unauthorized access from entering a private network by filtering the information that comes from the public internet. This forms an information safety barrier between a private network and the public internet.

Active directory:
A database and a set of services that connect users with the network resources. It contains critical information about the environment which includes the following: what users and computers are there, and who is allowed to do what.

Dockers:
An open platform for developing, shipping and running applications. Dockers enable you to separate your applications from your infrastructure.

Intrusion Detection System (IDS):

Samarth desai

A device or software application that monitors a network or system for malicious activity.

Proposed solution

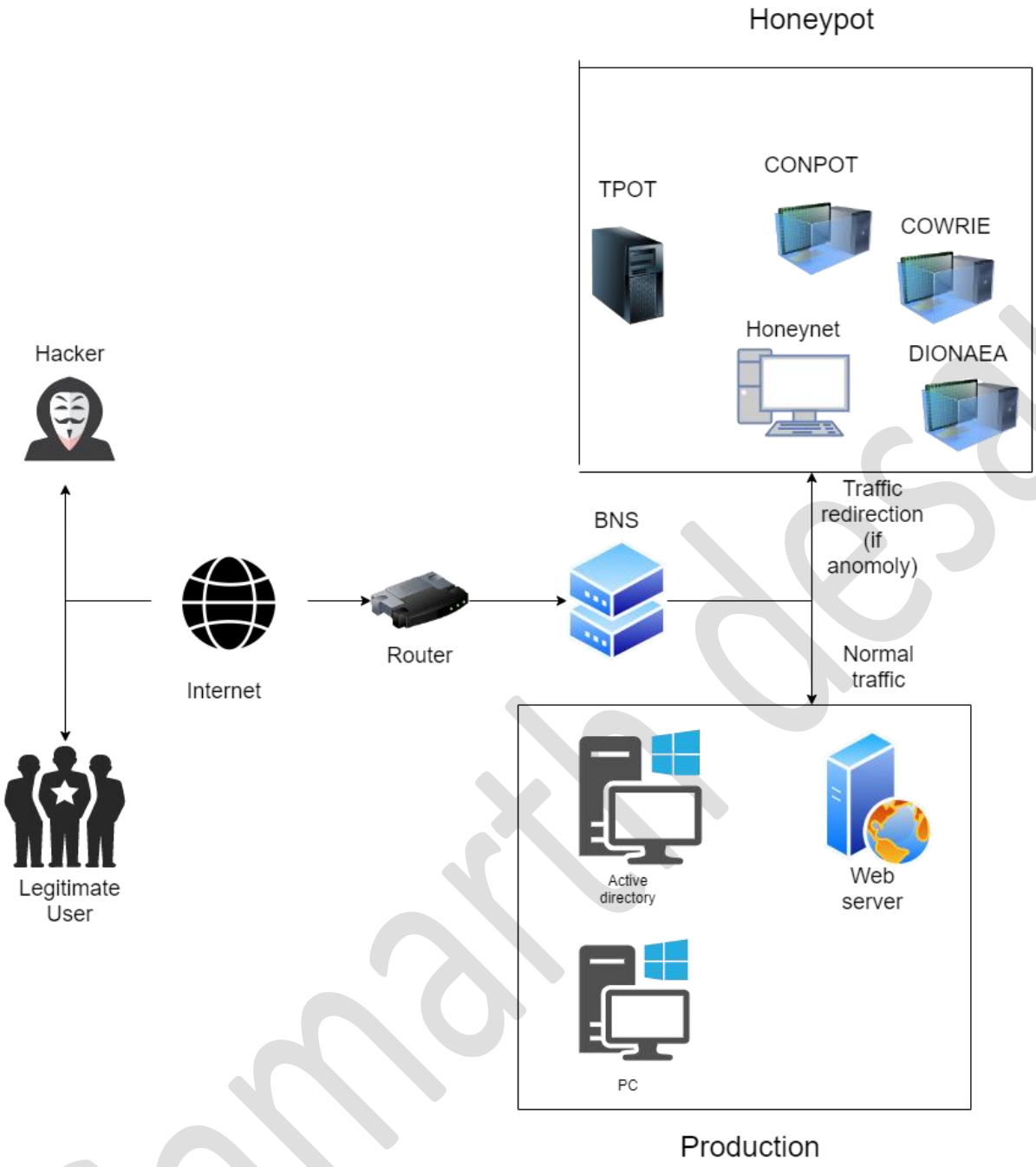
We designed a solution that is able to process different incoming attacks. We try to redirect and gather data, as well as deceive any hackers by using honeypots. This enables us to avoid attacks.

Proposed methodology

The details of the network design used to investigate attacks on our honeypots and to detect and prevent these events in real time. The approach used for this project was a mixed one, using both literature studied and laboratory experiments.

The diagram below shows the infrastructure we created. Legitimate users and intruders have a network connection, with the help of the router. BNS is an advanced honeypot router that detects anomalies. If anomalies are detected, network traffic is redirected to the honeypot environment. If no anomalies are detected, network traffic is redirected to the production environment.

Samarth desai


Honeypot
TPOT
CONPOT
COWRIE
Honeynet
DIONAEA
Hacker
Traffic redirection (if anomoly)
BNS
Internet
Router
Normal traffic
Legitimate User
Active directory
Web server
PC
Production

Smart Desal

Project phase

1. **Implementation**
   Configuring different machines in a virtual environment.
   Implementing honeypots, workstations, servers, router and IDS

2. **Performing different attacks**
   Performing different attacks like DDOS, website exploitation, exploiting AD user and using latest techniques to exploit and also try to escalate privileges.

3. **Data analysis**
   Analysis of data from the ongoing attack. Honeypot web console outputs and sequentially storing the data

4. **Post incident analysis**
   After attack incident data and analysis.
   Reviewing honeypot data and how they were helpful

5. **Results**
   Results from all steps will be compared to real life incidents and improving on the current technology

Tools and technology:

- T-pot parallax honeypots
- VMWARE Virtualization, Hacking tools
- PFSENSE firewall/router
- Active directory
- IDS

Samarth desai

Some types of cybersecurity attacks

DOS and DDOS

- A DoS involves the attacker flooding a targeted machine or resource e.g., a website with constant traffic e.g., requests, which is intended to make the services provided by a host connected to the internet inaccessible to users. This can be defended by the defender checking the resource's log, finding and blocking the IP address of the attacker. Therefore, attackers can use a distributed denial-of-service attack, where traffic is dent from many IP addresses of dumb machines.

A man in the middle attack (MitM)

- A man in the middle attack (MitM) involves an attacker secretly relaying electronic messages between the sender and recipient and intercepts them, viewing and perhaps changing them in transit. The sender and recipient believe they are communicating directly with one another.

Malware

Malware is software that performs a malicious task on a target device or network, e.g., corrupting data or taking over a system

- Viruses are usually attached to an executable file, which means the viruses may exist on a system but will not spread until the user visits an infected program.
- Trojan is a harmful piece of software that looks legitimate. Users are typically tricked into leading and executing it into their systems. Steals your data,
- Ransomware - An attack that involves encrypting data on the target system and demanding a ransom in exchange for letting the user have access to the data again. Encrypt data, money to give key, but they don't give key

Samarth desai

What is network discovery and how does it work?

Network discovery allows computers / devices to find other computers / devices on the same network. Network discovery enables network devices to connect and communicate e.g., share and find resources between the other devices connected to the same network.

IT teams utilize the following primary discovery protocols: Simple Network Management Protocol (SNMP), Link Layer Discovery Protocol (LLDP), and ping. SNMP is used for collecting and organizing data about devices on IP networks and for modifying that information to change device behaviour. LLDP allows network devices to advertise their identity and capabilities on a LAN, so when LLDP is enabled then information is transmitted from a device to the other neighbouring device in the network at regular intervals, which are then stored in the management information database of the neighbouring device. LLDP is a vendor-neutral protocol. Ping involves sending Internet Control Message Protocol (ICMP) echo request packets to a host and waiting for an ICMP echo reply. In order to test the reachability of a host, ping measures and provides packet losses, errors and some statistical summaries e.g., mean round-trip time.

"The device discovery request is given to the NWK layer by the APL layer. The network layer 3 uses the MAC layer channel scanning to discover the presence of other networks. The active scan is the preferred scan method if the device is capable of performing an active scan. Otherwise, the device will perform a passive scan."
https://www.sciencedirect.com/topics/computer-science/network-discovery

Whilst system administrators are able to create network maps due to network discovery, which allows IT teams to potentially identify points of failure before issues occur to limit costly network downtime/interruption, network discovery can also make it easier for an attacker to find resources shared between devices in the network.

Samarth desai

Our infrastructure

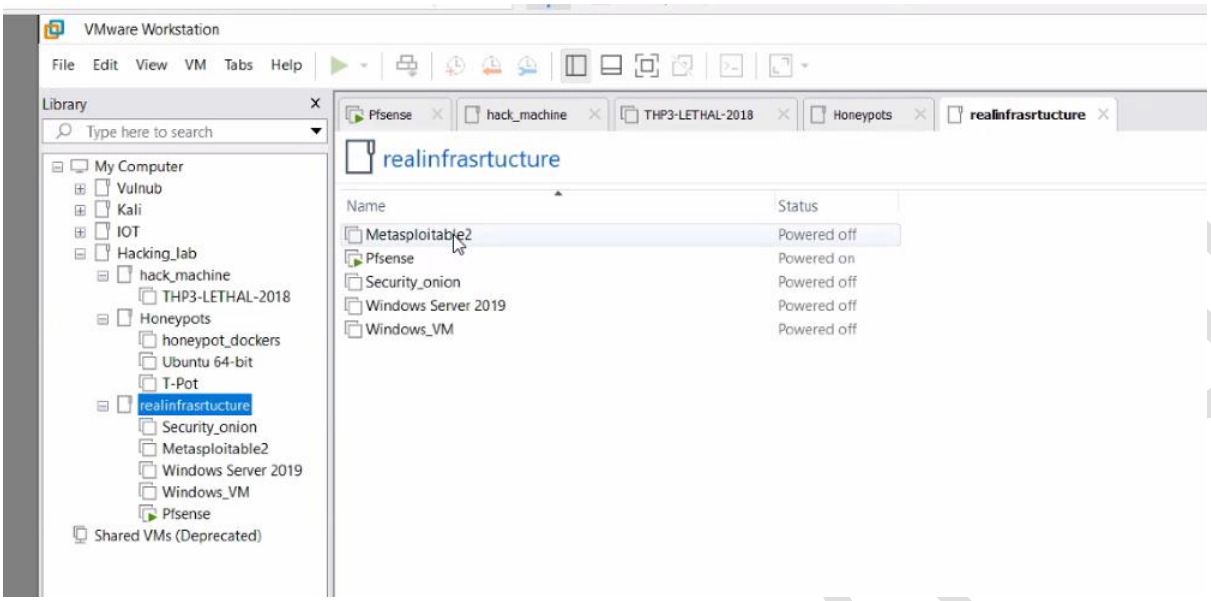


Figure 1

Figure 1 shows Production environment

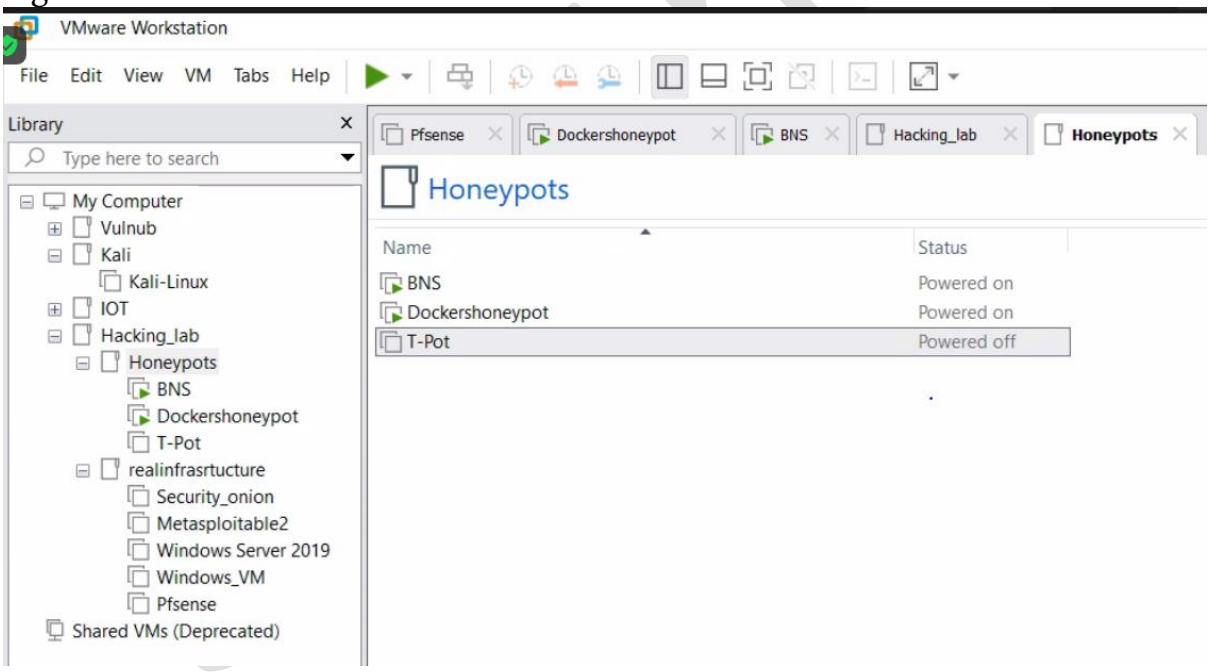


Figure 2 shows the honeypot environment

Honeypots as a Form of Deception

A honeypot may be simply defined as a system that has no legitimate purpose other than to be attacked and thereby give alerts of such activity. As has already been discussed, honeypots have been used for deception for many years. One of the problems with early honeypot technologies was the lack of the ability to scale. It often took an experienced security expert to deploy and monitor the technology, which was a full-time job in some environments, even just to monitor a few honeypots. In today's corporate environments, the older honeypot technology is simply too lab or intensive to deploy effectively. However, what was old is new again! With the advent of greater virtual technologies, container technologies such as Docker, and analytic tools such as the Elasticsearch, Logstash, Kibana (ELK) stack, what was once a full-time challenge has turned into a valuable asset of any organization's cyberdefense, which may be managed part time. As, modern honeypots are easy to deploy and manage, at scale. Even industry analysts have noted that modern deception technologies should be deployed in order to supplement the other enterprise security technologies. Honeypot technology will not replace other layers of technology, but once an attacker is inside a network, it may be your best shot at catching them. The main reason to deploy honeypots as a form of deception is to delay, disrupt, and distract the attacker in order to detect and stop them. The key attribute of honeypot technology is its low false-positive nature. By the very definition we used, honeypots should not be touched by anyone but an attacker. Therefore, when a connection is made to a honeypot, there is either a misconfigured server that needs attention, a curious user who needs attention, or an attacker who needs attention. There are no other options; therefore, honeypot technology is about as false-positive proof as you can get. In today's high-false-positive environment, the ability to deploy a low or no false-positive technology should get your attention. We'll take a closer look at the following types of honeypot technologies:

- High-interaction honeypots
- Low-interaction honeypots
- Medium-interaction honeypots
- Honeyclients
- Honeytokens

High-Interaction Honeypots

High-interaction honeypots are most often real systems that are instrumented to monitor and catch an attacker in a near real-time manner. The problem, of course, with highinteraction honeypots is that real systems may be rooted and then used by the attacker to further their attack on the hosting network or other networks. Therefore, high-interaction honeypots are risky and often avoided.

Sample Desai

Low-Interaction Honeypots

Low-interaction honeypots are at the other end of the spectrum; they are simulated services that run in some sort of emulated environment, whereby the service is simulating realistic responses. However, there is often a limit to that simulation. For example, the commands of the Telnet service may be emulated using Python or another scripting language, but not all of the commands work. If an attacker attempts to download a file with wget, for example, perhaps the command appears to work but the file is not provided to the attacker; instead, it is provided to the defender for further analysis. There are other practical limits as well in that it might not be feasible to emulate all the commands of Telnet. Therefore, if an attacker tries one of those commands and it fails, the deception could be over. We will take a look at some popular low-interaction honeypots in the coming sections and labs.

Medium-Interaction Honeypots

Medium-interaction honeypots were purposely listed after the other two, as they are a newer concept of deeply emulating services. This includes fully reproducing complex operating system processes, such as the SMB network protocol, to a degree that an attacker can run real exploits against the seemingly vulnerable service and in some cases even return a shell. This is a marked improvement over low-interaction honeypots, which would normally fail on those types of attacks. Some medium interaction honeypots actually proxy the commands to the real operating system to achieve this level of deception. Another form of medium-interaction honeypot would be a canary service, running on a real system, whose purpose is to alert the defender to attacker behaviour.

Honeyclients

Honeyclients are the other side of the honeypot coin. Whereas honeypots are generally services, soliciting a connection and request from an attacker, honeyclients are client applications, seeking to make connections to potentially compromised systems and extract binaries and potential malware for the purpose of analysis and defensive use of that knowledge elsewhere in the enterprise. There are web-based honeyclients and other forms of honeyclients available as well

Honeytokens

Honeytokens are any form of bait that falls outside the traditional server/client model. A common form of honeytokens is a file that contains fake data that is attractive to the attacker. When used by the attacker, this file alerts the defender to their presence. For example, imagine a file called passwords.txt that sits in the root directory of a user on a honeypot system. The file contains fake accounts and fake passwords that do not exist. However, the attacker does not know that when they try to use those accounts, and an alert is fired off in the enterprise's Security Information Event Management (SIEM) system, notifying the

samarth desai

defender to the attack. A great open-source resource for generating and tracking honeytokens is canarytokens.org

Dionaea

dionaea intention is to trap malware exploiting vulnerabilities exposed by services offered to a network, the ultimate goal is gaining a copy of the malware.
As Software is likely to have bugs, bugs in software offering network services can be exploitable, and dionaea is software offering network services, it is likely dionaea has exploitable bugs.
Of course, we try to avoid it, but if nobody would fail when trying hard, we would not need software such as dionaea.
So, in order to minimize the impact, dionaea can drop privileges, and chroot.
To be able to run certain actions which require privileges, after dionaea dropped them, dionaea creates a child process at start-up, and asks the child process to run actions which require elevated privileges. This does not guarantee anything, but it should be harder to get gain root access to the system from an unprivileged user in a chroot environment.

```
root@xx01:~# ls -l /opt/dionaea/var/dionaea/binaries/
total 67992
-rw------- 1 dionaea dionaea 5267459 Dec  4 07:15 0d7a66f4d4ba7533cb5ed132f68b53b7
-rw------- 1 dionaea dionaea 5267459 Dec  4 05:03 0e80a07bf580016d84894dcef82a8f55
-rw------- 1 dionaea dionaea 5267459 Dec  4 09:20 1893ed20136dc10af3c5bf9e9245e3f5
-rw------- 1 dionaea dionaea 5267459 Dec  4 02:15 525bbc440bf03a9602d090d2e79b4471
-rw------- 1 dionaea dionaea   79875 Dec  4 01:36 76c6b14e88cf6d0f8f650cecf2a0d27a
-rw------- 1 dionaea dionaea 5267459 Dec  4 03:14 996c2b2ca30180129c69352a3a3515e4
-rw------- 1 dionaea dionaea 5267459 Dec  4 07:14 a55b9addb2447db1882a3ae995a70151
-rw------- 1 dionaea dionaea 5267459 Dec  4 00:53 ae12bb54af31227017feffd9598a6f5e
-rw------- 1 dionaea dionaea 5267459 Dec  4 03:32 b99e6f96aefa286c73d4e8e6c2d83dda
-rw------- 1 dionaea dionaea 5267459 Dec  4 01:34 bf49a02eb6495d2cd9876a593a86b768
-rw------- 1 dionaea dionaea 5267459 Dec  4 03:14 ce494e90f5ba942a3f1c0fe557e598bf
-rw------- 1 dionaea dionaea 5267459 Dec  4 03:33 cf4f46336abeec03630297f846d17482
-rw------- 1 dionaea dionaea 5267459 Dec  4 09:05 e5840a9753ed8f90fbd7264c8db27c4b
-rw------- 1 dionaea dionaea 5267459 Dec  4 05:59 efccc5f1740e8729eeb100562a3c2076
-rw------- 1 dionaea dionaea 1009664 Dec  4 03:27 ff50a3835febac4f32700dc2eb15ddec
```

This is one of the attacks for ftp performed and as you can see the attacker is easily deceived by the honeypot

Sahil Desai

```
root@kali:~# ftp 127.0.0.1
Connected to 127.0.0.1.
220 DiskStation FTP server ready.
Name (127.0.0.1:root): foo
331 Password required for foo.
Password:
230 User logged in, proceed
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> help
Commands may be abbreviated.  Commands are:

!           dir         mdelete     qc          site
$           disconnect  mdir        sendport    size
account     exit        mget        put         status
append      form        mkdir       pwd         struct
ascii       get         mls         quit        system
bell        glob        mode        quote       sunique
binary      hash        modtime     recv        tenex
bye         help        mput        reget       tick
case        idle        newer       rstatus     trace
cd          image       nmap        rhelp       type
cdup        ipany       nlist       rename      user
chmod       ipv4        ntrans      reset       umask
close       ipv6        open        restart     verbose
cr          lcd         prompt      rmdir       ?
delete      ls          passive     runique
debug       macdef      proxy       send
```

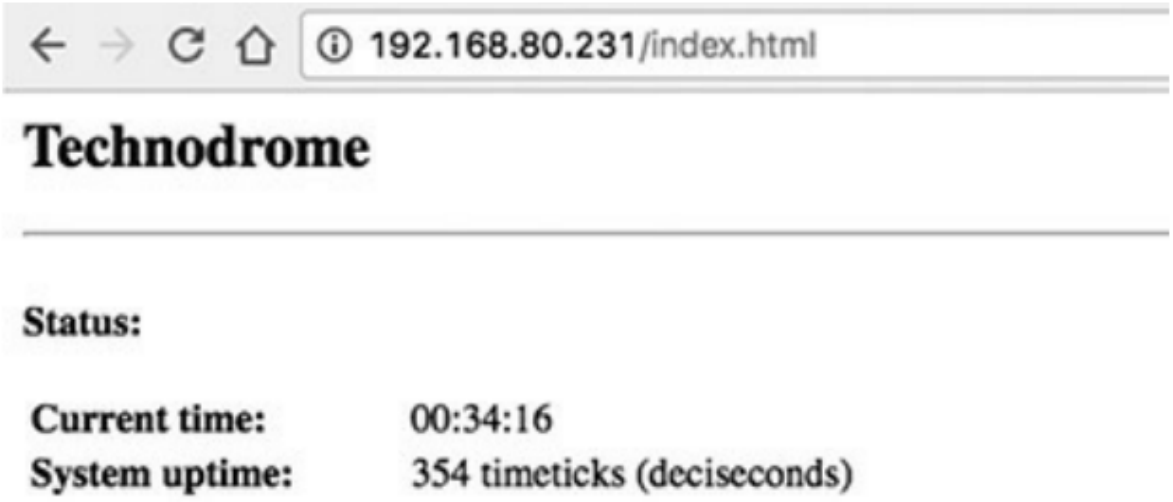
Samarth Bansal

Conpot

Conpot is an ICS honeypot with the goal to collect intelligence about the motives and methods of adversaries targeting industrial control systems.
we investigate the ConPot honeypot, which emulates an ICS/SCADA
device

```
root@kali:~# docker run -it -p 80:80 -p 102:102 -p 502:502 -p
161:161/udp -v $(pwd)/var/log/conpot:/var/log/conpot --network=bridge honeynet/conpot:latest
```

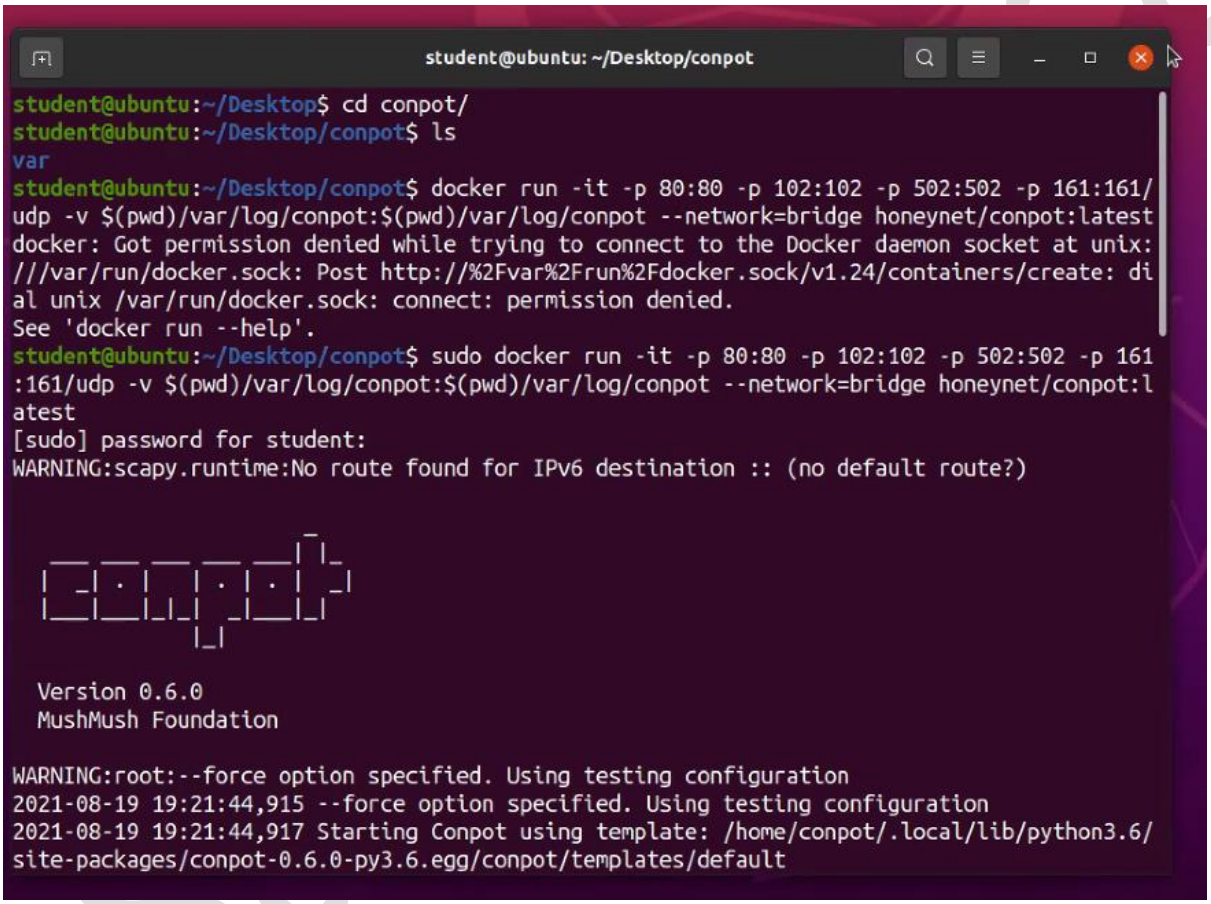
web page and view the web interface, shown next. Be sure to click Refresh a few times to see the changes.

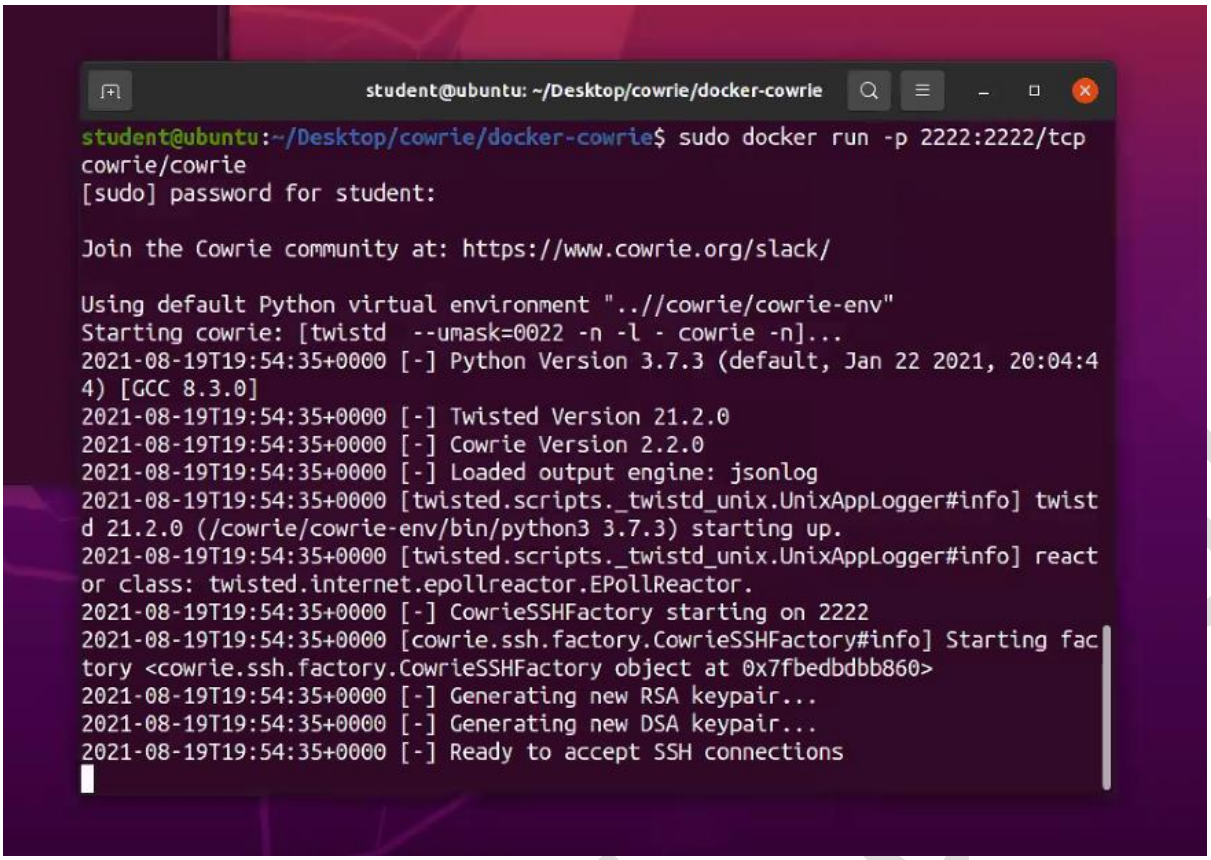



```
$ snmpwalk -c public 192.168.80.231
SNMPv2-MIB::sysDescr.0 = STRING: Siemens, SIMATIC, S7-200
SNMPv2-MIB::sysObjectID.0 = OID: SNMPv2-SMI::enterprises.20408
DISMAN-EVENT-MIB::sysUpTimeInstance = Timeticks: (415) 0:00:04.15
SNMPv2-MIB::sysContact.0 = STRING: Siemens AG
SNMPv2-MIB::sysName.0 = STRING: CP 443-1 EX40
SNMPv2-MIB::sysLocation.0 = STRING: Venus
SNMPv2-MIB::sysServices.0 = INTEGER: 72
SNMPv2-MIB::sysORLastChange.0 = Timeticks: (0) 0:00:00.00
SNMPv2-MIB::snmpInPkts.0 = Counter32: 9
SNMPv2-MIB::snmpOutPkts.0 = Counter32: 0
SNMPv2-MIB::snmpInBadVersions.0 = Counter32: 0
SNMPv2-MIB::snmpInBadCommunityNames.0 = Counter32: 0
SNMPv2-MIB::snmpInBadCommunityUses.0 = Counter32: 0
SNMPv2-MIB::snmpInASNParseErrs.0 = Counter32: 0
```

Cowrie

we pull and use the Cowrie honeypot, which, as described by the author, is a medium-interaction honeypot, capable of emulating SSH and Telnet and, most importantly, capturing each command. It is also able to replay the key sequences for an entertaining view of hacker activity. Clone the honeypot GitHub repository, and then configure, build, and run the honeypot:




student@ubuntu: ~/Desktop/cowrie/docker-cowrie
student@ubuntu:~/Desktop/cowrie/docker-cowrie$ sudo docker run -p 2222:2222/tcp cowrie/cowrie
[sudo] password for student:
Join the Cowrie community at: https://www.cowrie.org/slack/
Using default Python virtual environment "..//cowrie/cowrie-env"
Starting cowrie: [twistd --umask=0022 -n -l - cowrie -n]...
2021-08-19T19:54:35+0000 [-] Python Version 3.7.3 (default, Jan 22 2021, 20:04:44) [GCC 8.3.0]
2021-08-19T19:54:35+0000 [-] Twisted Version 21.2.0
2021-08-19T19:54:35+0000 [-] Cowrie Version 2.2.0
2021-08-19T19:54:35+0000 [-] Loaded output engine: jsonlog
2021-08-19T19:54:35+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] twistd 21.2.0 (/cowrie/cowrie-env/bin/python3 3.7.3) starting up.
2021-08-19T19:54:35+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.internet.epollreactor.EPollReactor.
2021-08-19T19:54:35+0000 [-] CowrieSSHFactory starting on 2222
2021-08-19T19:54:35+0000 [cowrie.ssh.factory.CowrieSSHFactory#info] Starting factory <cowrie.ssh.factory.CowrieSSHFactory object at 0x7fbedbdbb860>
2021-08-19T19:54:35+0000 [-] Generating new RSA keypair...
2021-08-19T19:54:35+0000 [-] Generating new DSA keypair...
2021-08-19T19:54:35+0000 [-] Ready to accept SSH connections

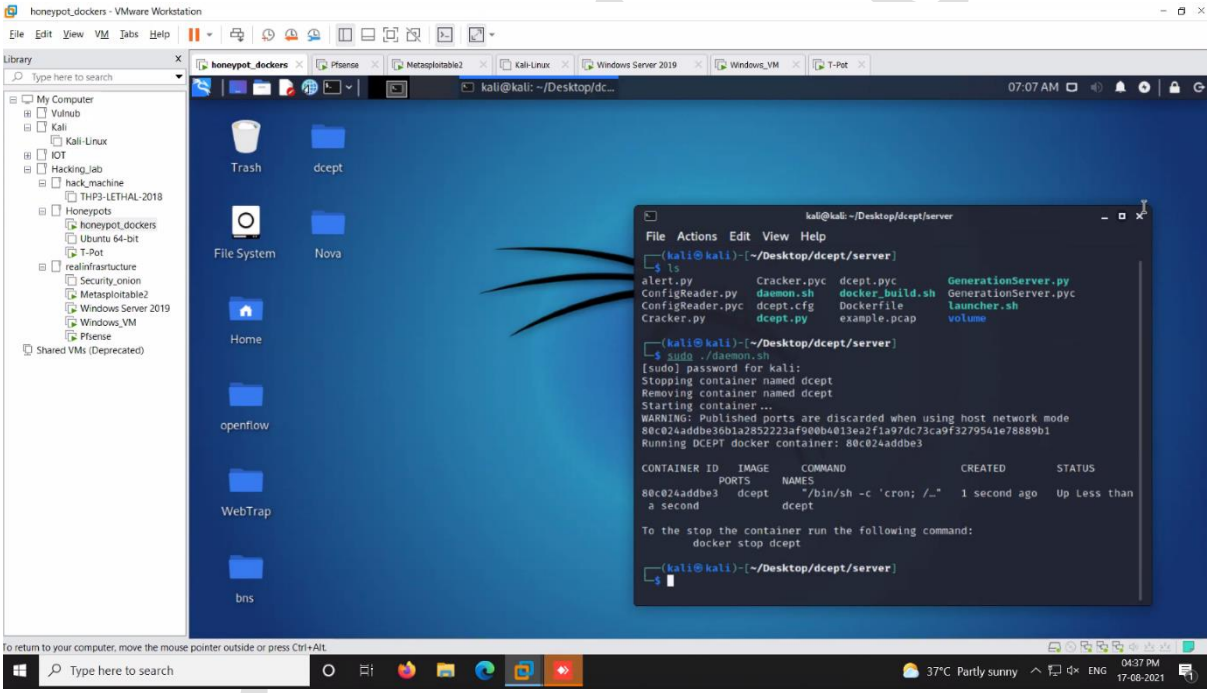
Samarth

Honeytokens DCEPT for active directory

DCEPT (Domain Controller Enticing Password Tripwire) is a honeytoken-based tripwire for Microsoft's Active Directory. Honeytokens are pieces of information intentionally littered on system so they can be discovered by an intruder. In the case of DCEPT, the honeytokens are credentials that would only be known by a someone extracting them from memory. A logon attempt using these faux credentials would mean someone was inside the network and is attempting privilege escalation to domain administrator.

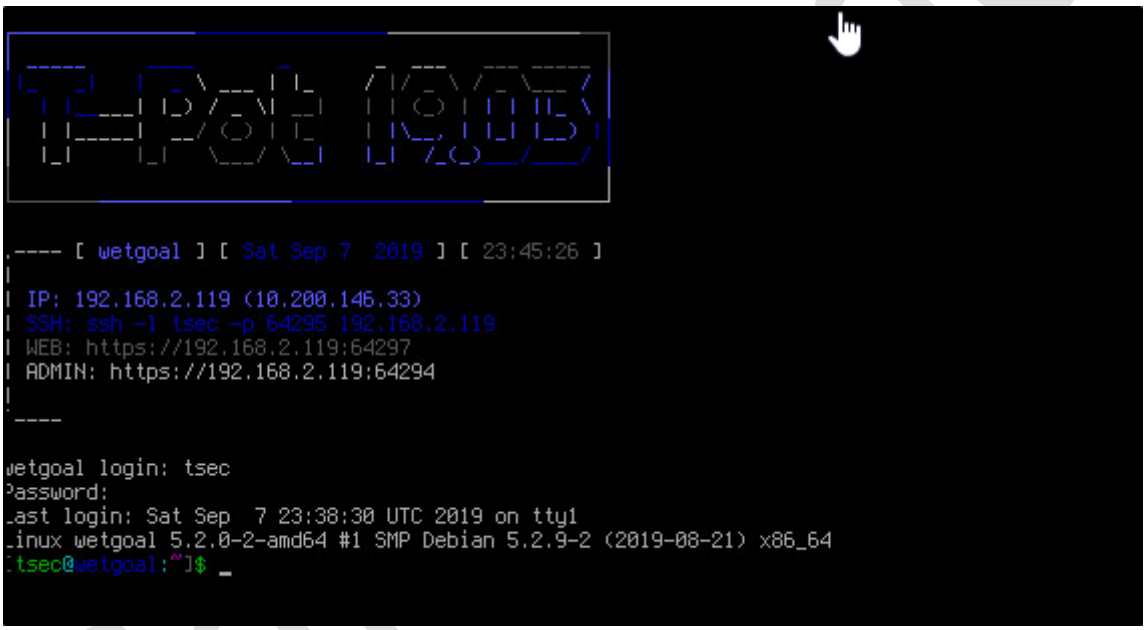
This proof of concept is being released as open source to benefit Windows system administrators. The goal of this project was to provide a free, simple, honeytoken deployment tool as well as educate administrators about the nature of these attacks. We encourage contributors to build on what we have done and welcome feedback. Has DCEPT helped your organization spot an intrusion before it was too late? We would like to hear from you.

More information about this research project can be found here:
https://www.secureworks.com/blog/dcept

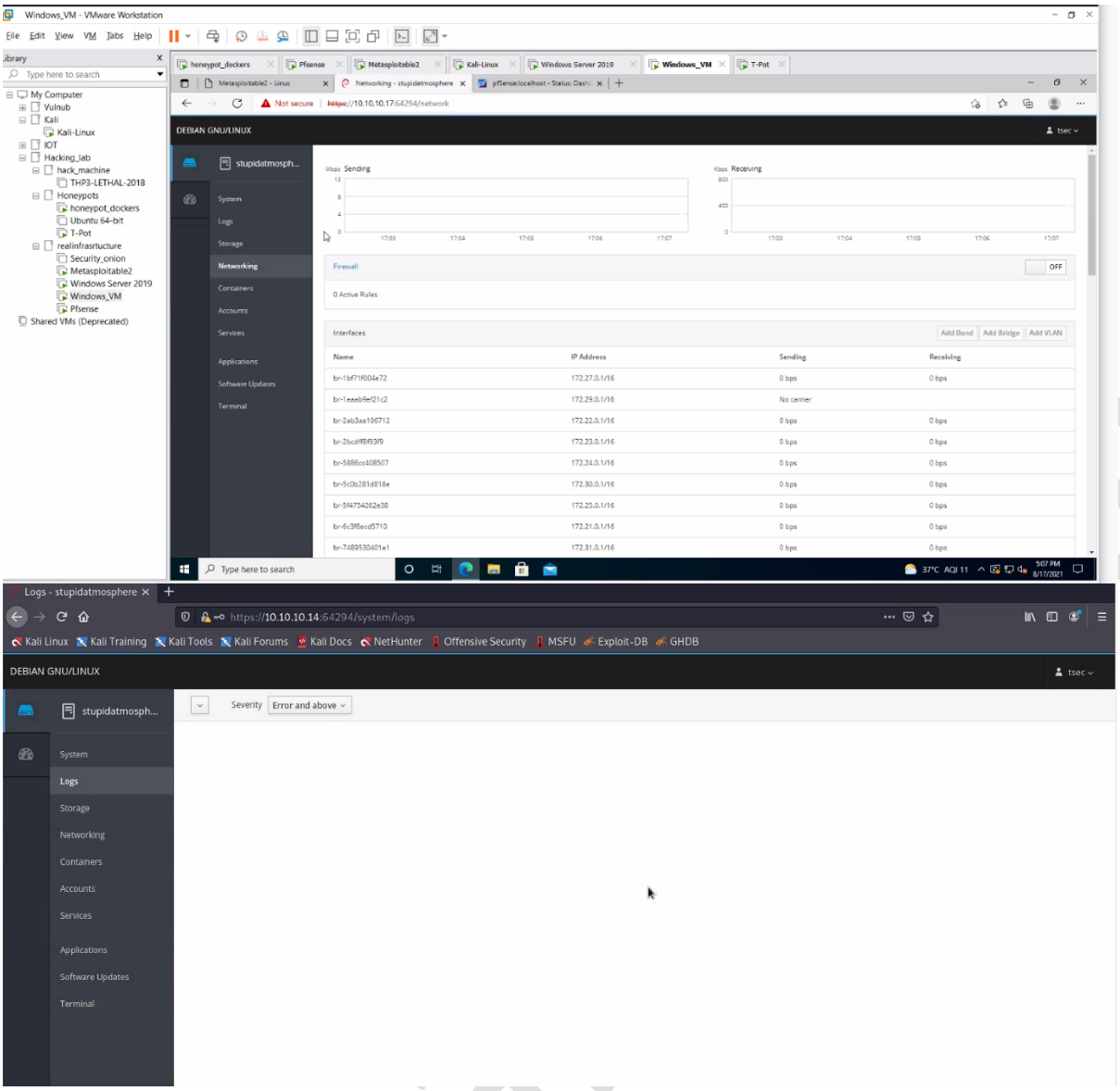


TPOT

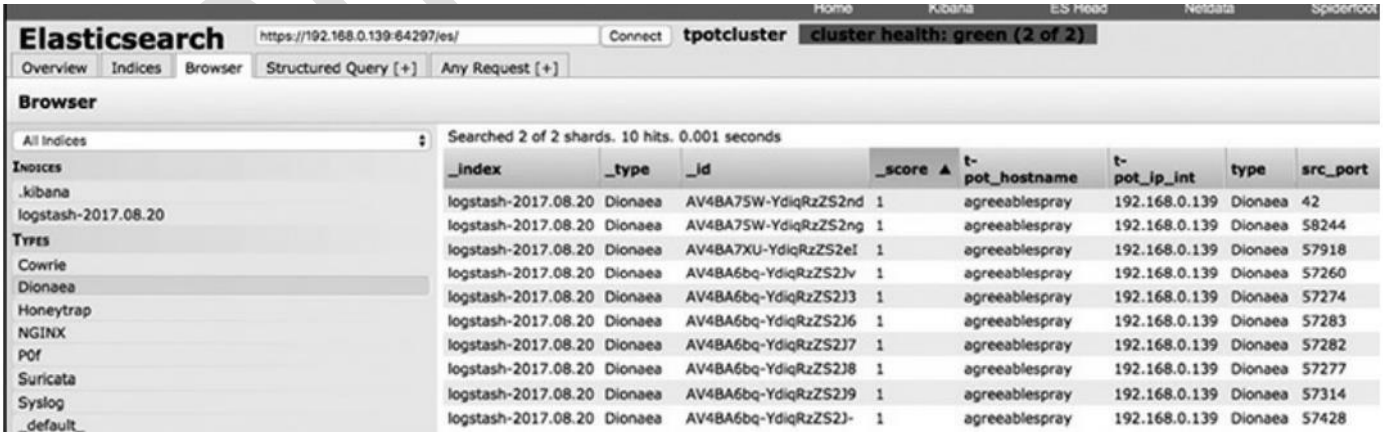
we pull it all together and download and install the T-Pot honeypot, which is an automated install of several other honeypots, including the ones we've used in previous labs. Further, T-Pot includes a user interface that's built on an Elasticsearch, Logstash, and Kibana (ELK) stack. 18 The version of T-Pot tested in this lab may be downloaded from the book's website. The latest version may be downloaded from the T-Pot GitHub (see the "For Further Reading" section). The minimum system requirements of the T-Pot honeypot are 4GB of RAM and 64GB of hard drive space for the standard honeypot (it may run with less, but these are the posted minimums). The easiest option to run the T-Pot honeypot is to download the ISO image or build your own and then mount it to a virtual CD in VMware or VirtualBox and launch the machine. The ISO is a 64-bit Ubuntu build, as shown next. Again, be sure to establish the minimum settings just given. For limited testing, you can get by with a smaller (5GB) hard drive.



From another Linux or Mac system, scan the IP with Nmap. Next, open the web interface, using the preceding IP (https://IP:64297), and select the T-Pot dashboard. You will need to place your honeypot on a public Internet connection and/or scan it to see some activity in the dashboards. However, the following screenshot shows the potential of this tool.



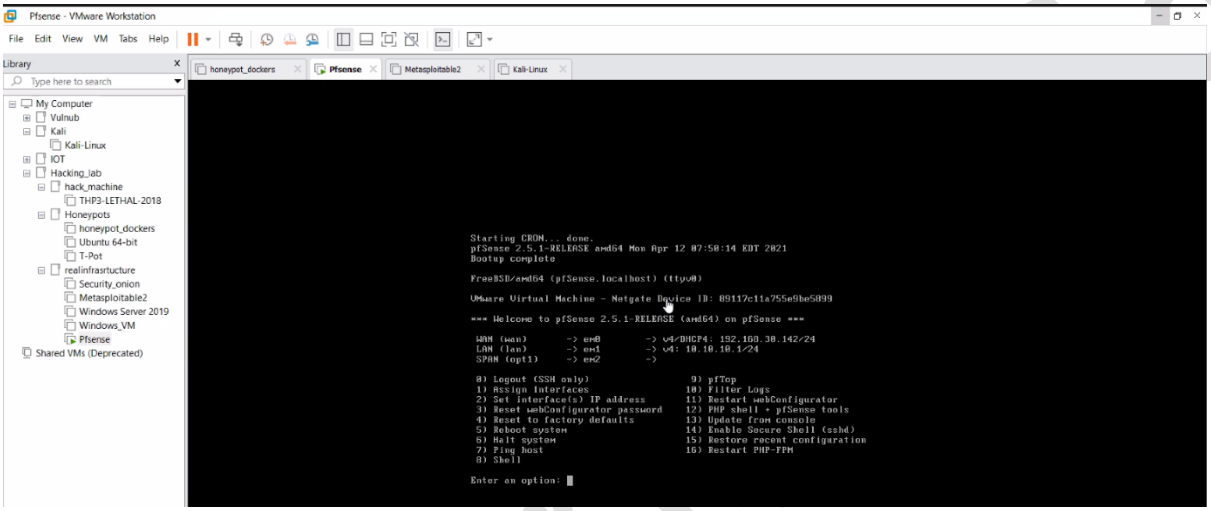
The web interface has several tools, including an Elasticsearch head (starting point for searches)

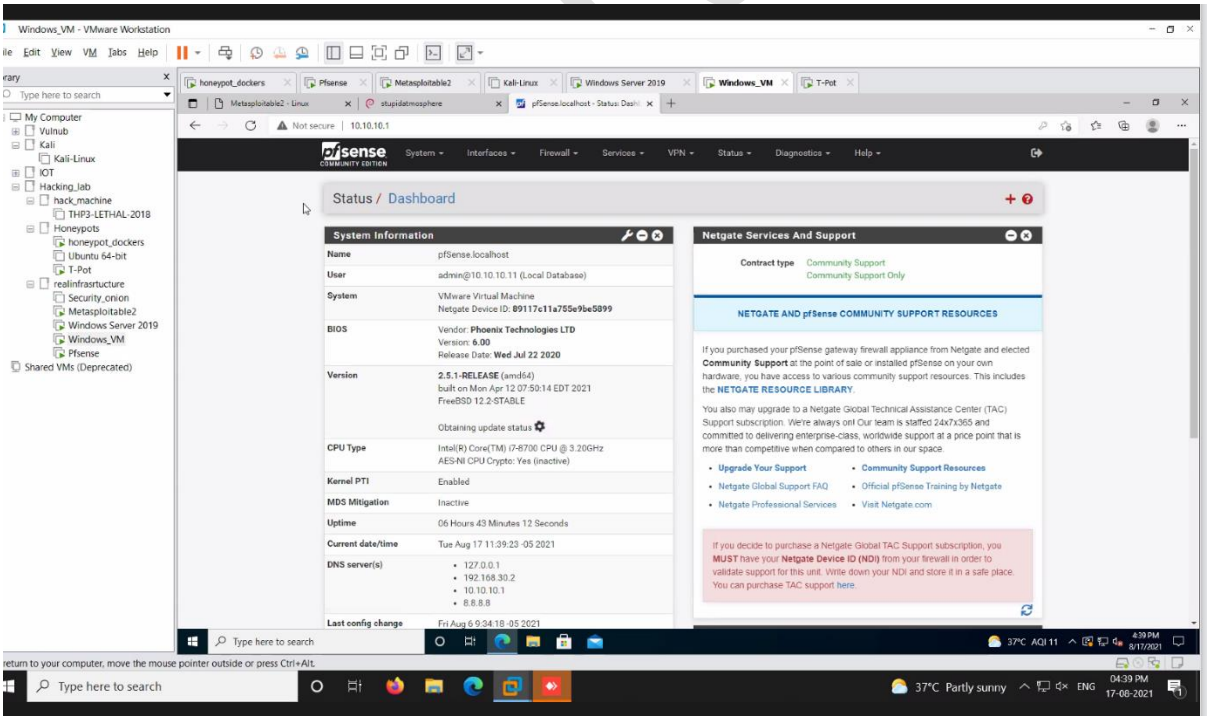


Production environment

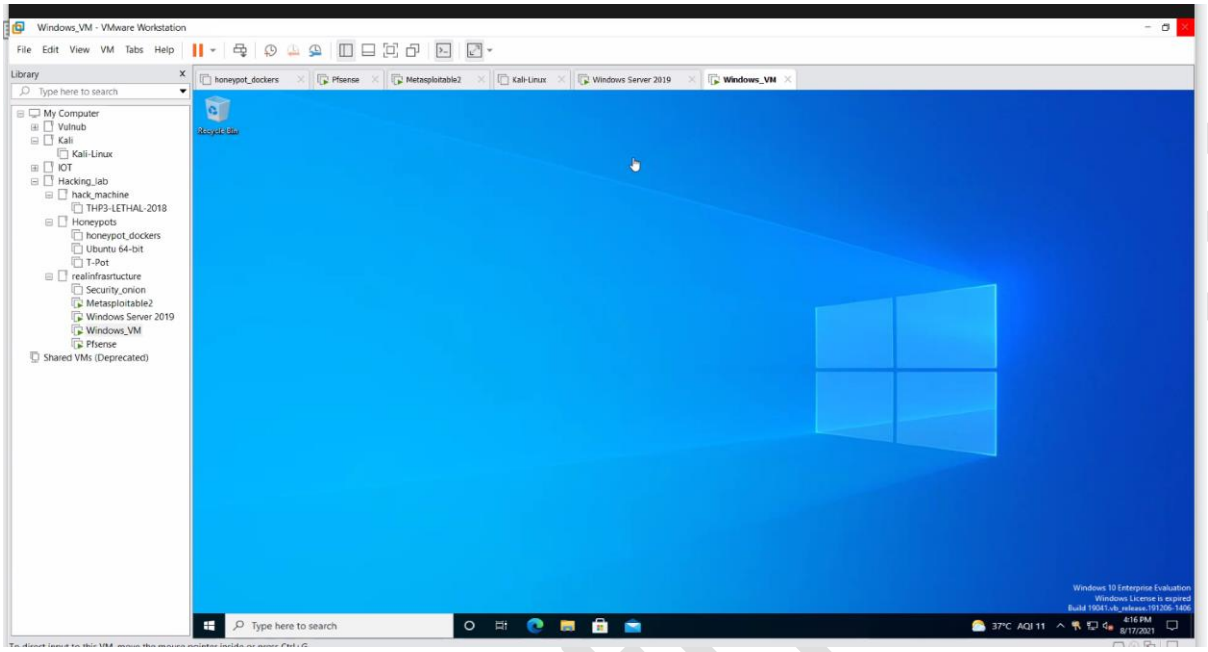
PFSENSE firewall/router

pfSense is a firewall/router computer software distribution based on FreeBSD. pfSense Community Edition (CE) is the partially open-source version while pfSense Plus has moved to a closed source model. It is installed on a physical computer or a virtual machine to make a dedicated firewall/router for a network. It can be configured and upgraded through a web-based interface, and requires no knowledge of the underlying FreeBSD system to manage





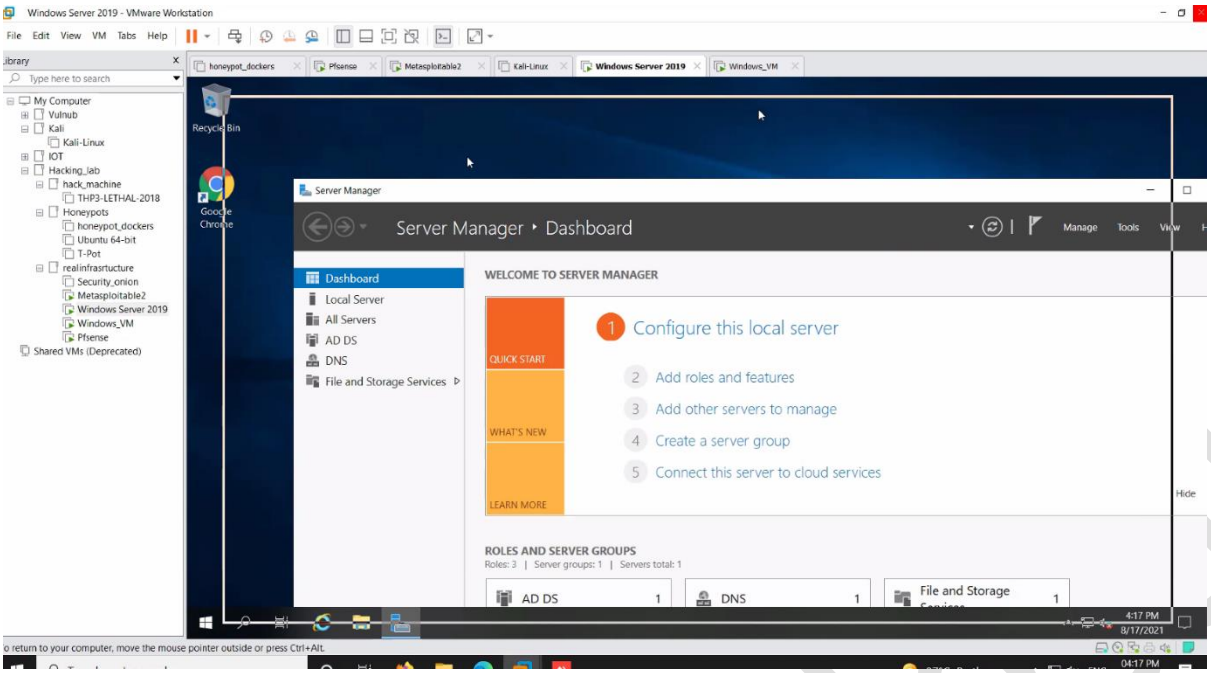
Windows mange connected to active directory



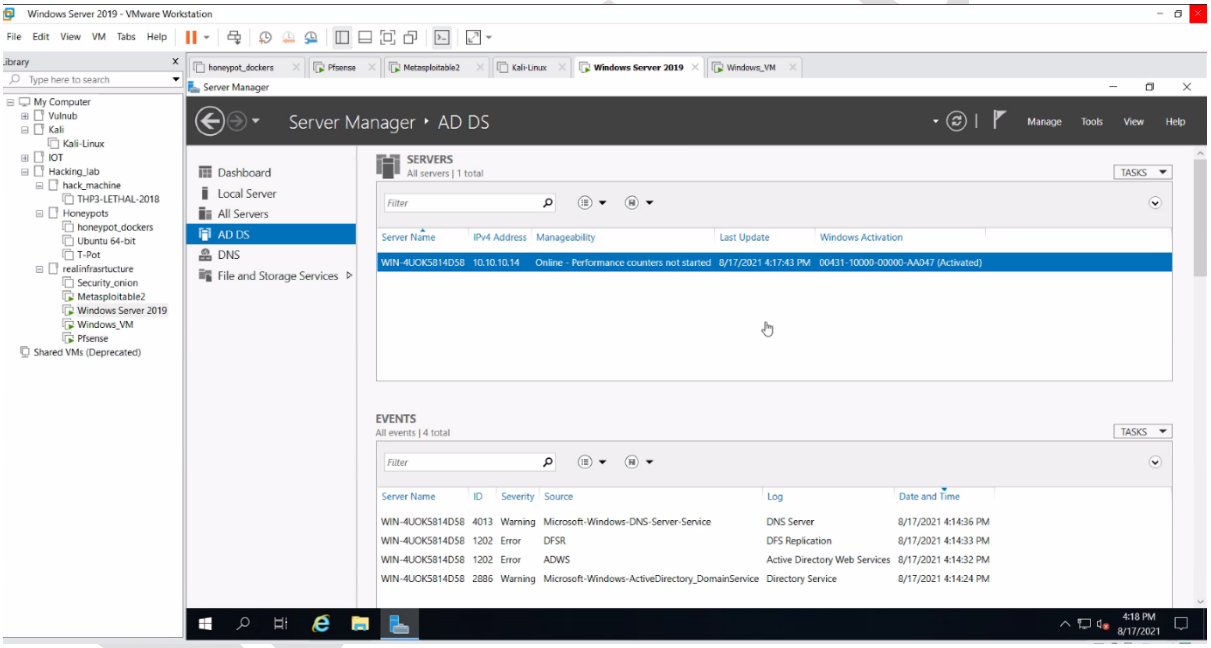
Windows virtual machine

Active directory server with domain controller

Samarth



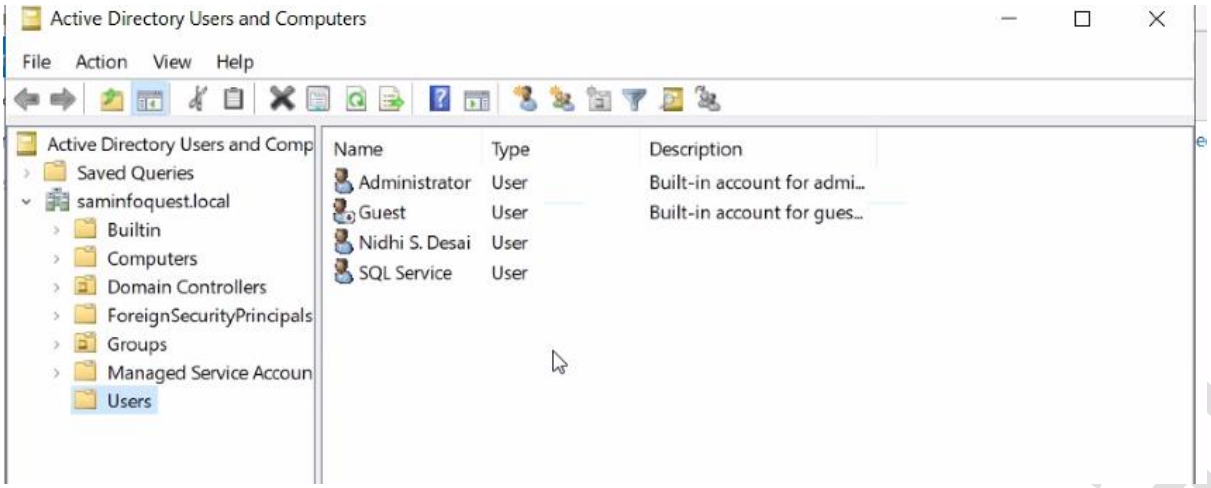
Active directory services are services provided by Microsoft and they provide authentication, security and other services for users.



File and storage services, provide cloud storage from services, providing cloud services to windows VR

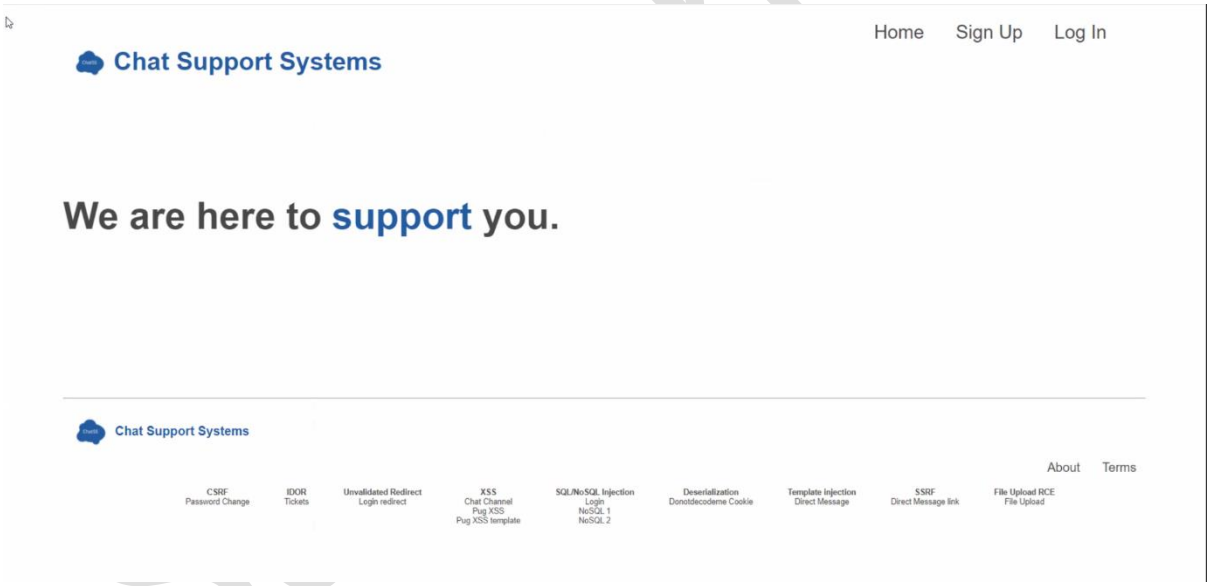
Active directory server has file shares to make to vulnerable

Active directory users:

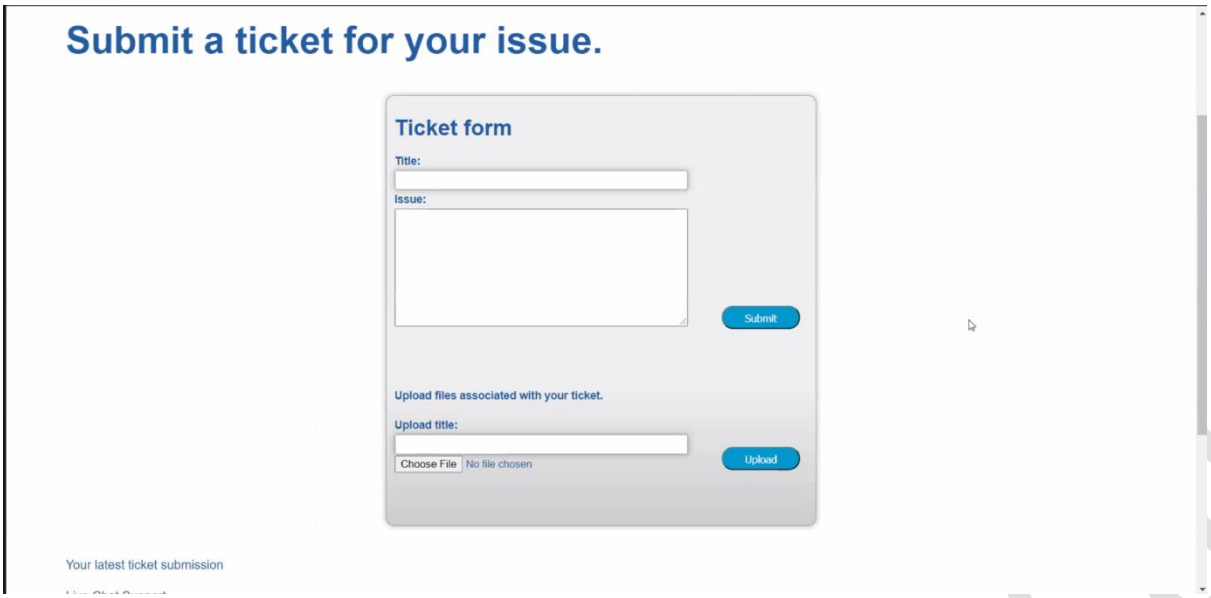


List of users connected to active directory

Web server and website



Front-page of website

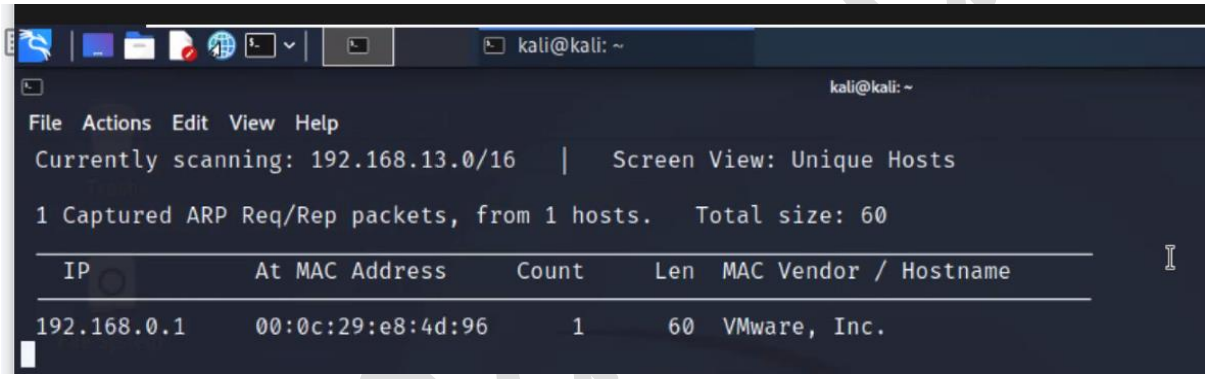


Another page of website

```
ens33     Link encap:Ethernet  HWaddr 00:50:56:2f:00:f1
          inet addr:192.168.152.128  Bcast:192.168.152.255  Mask:255.255.255.0
          inet6 addr: fe80::250:56ff:fe2f:f1/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:5 errors:0 dropped:0 overruns:0 frame:0
          TX packets:7 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:866 (866.0 B)  TX bytes:1102 (1.1 KB)

lo        Link encap:Local Loopback
          inet addr:127.0.0.1  Mask:255.0.0.0
          inet6 addr: ::1/128 Scope:Host
          UP LOOPBACK RUNNING  MTU:65536  Metric:1
          RX packets:160 errors:0 dropped:0 overruns:0 frame:0
          TX packets:160 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1
          RX bytes:11840 (11.8 KB)  TX bytes:11840 (11.8 KB)
LethalNodeJS login: [   39.894324] rc.local[1176]: Session {
[   39.894556] rc.local[1176]:   cookie:
[   39.894626] rc.local[1176]:    { path: '/',
[   39.894684] rc.local[1176]:      _expires: 2021-09-03T14:41:46.042Z,
[   39.894752] rc.local[1176]:      originalMaxAge: 60000,
[   39.894820] rc.local[1176]:      httpOnly: false } }
[  134.216258] rc.local[1176]: test
-
```

Logs of the web server

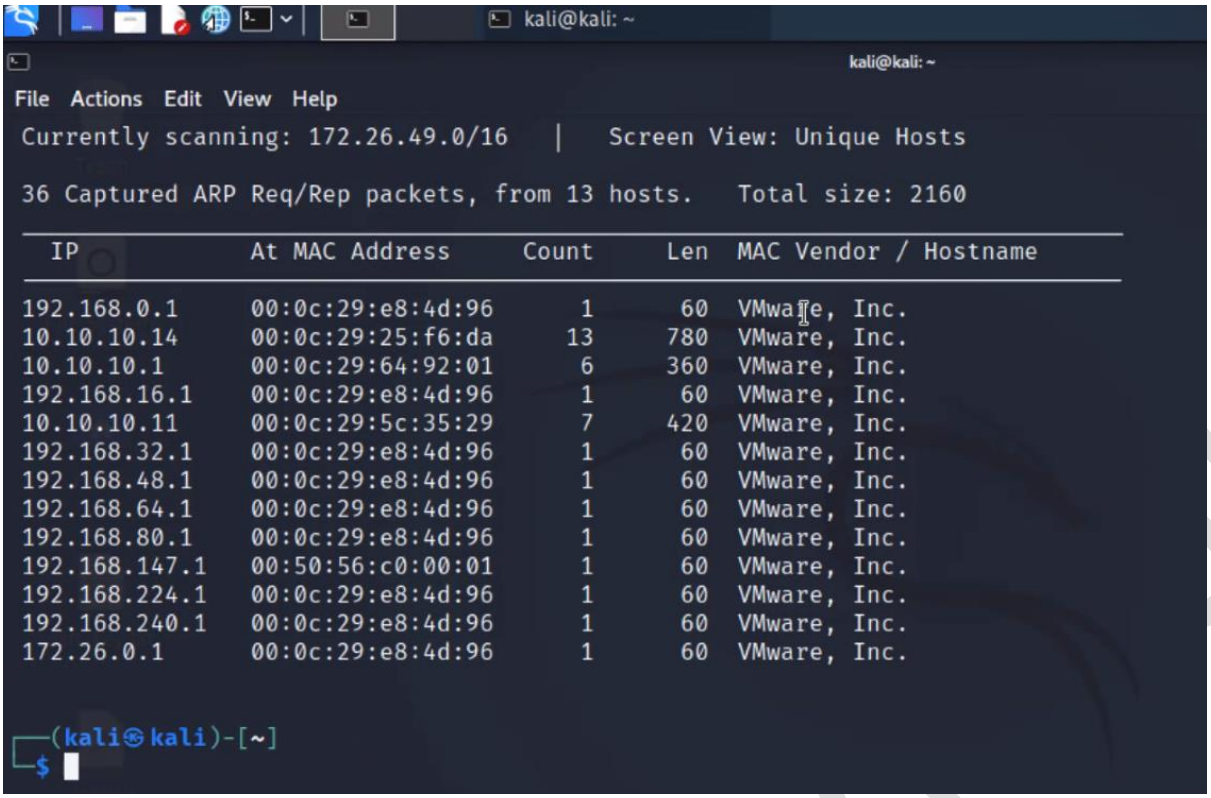
Reconnaissance on infrastructure



Obtaining addresses

Net discover discovers devices present and pings them hi and sees if they are active / live

Samadesai



Scan t-pot to see if t-pot is legitimately running

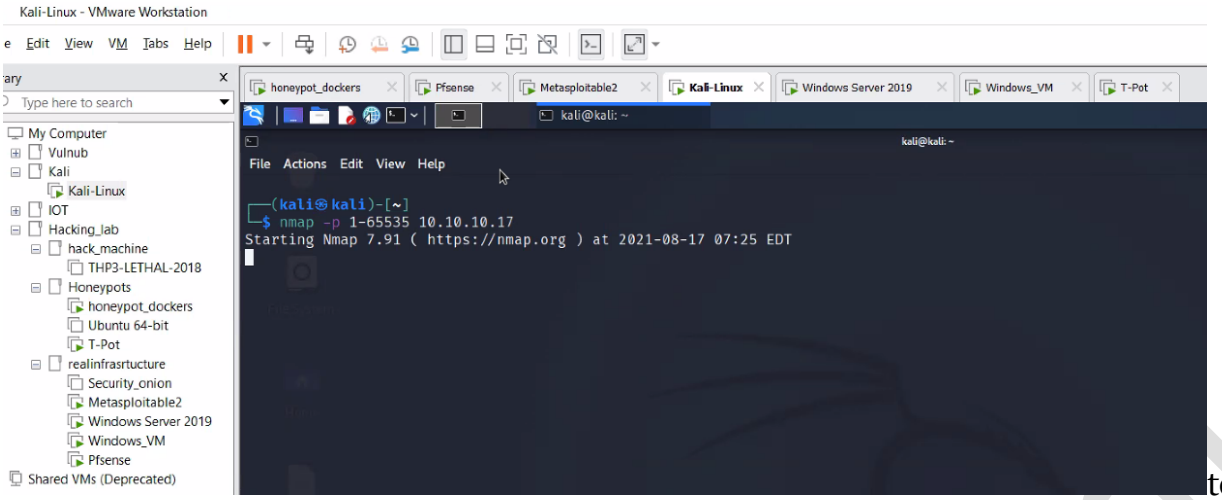
10.10.17 is IP address of t-pot, ip-range is 10. Blah. Should show 10.10.something but showing many other ip addresses shown to confuse hacker. Is not showing 10.10.17 so ip address of t-pot not detected.

Most impotant command - nmap

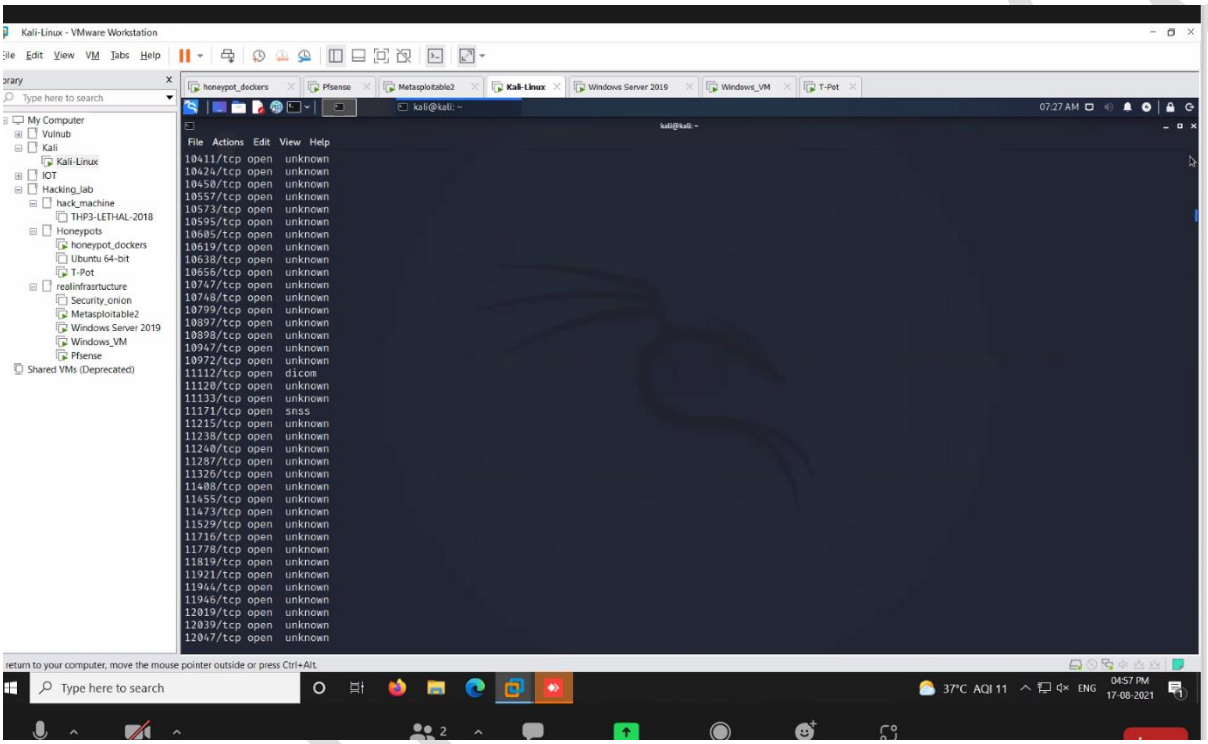
Scans ip address to check if there are open services or open ports, fetches details about them

Tying to scan for honeypot using

Samarth Gupta



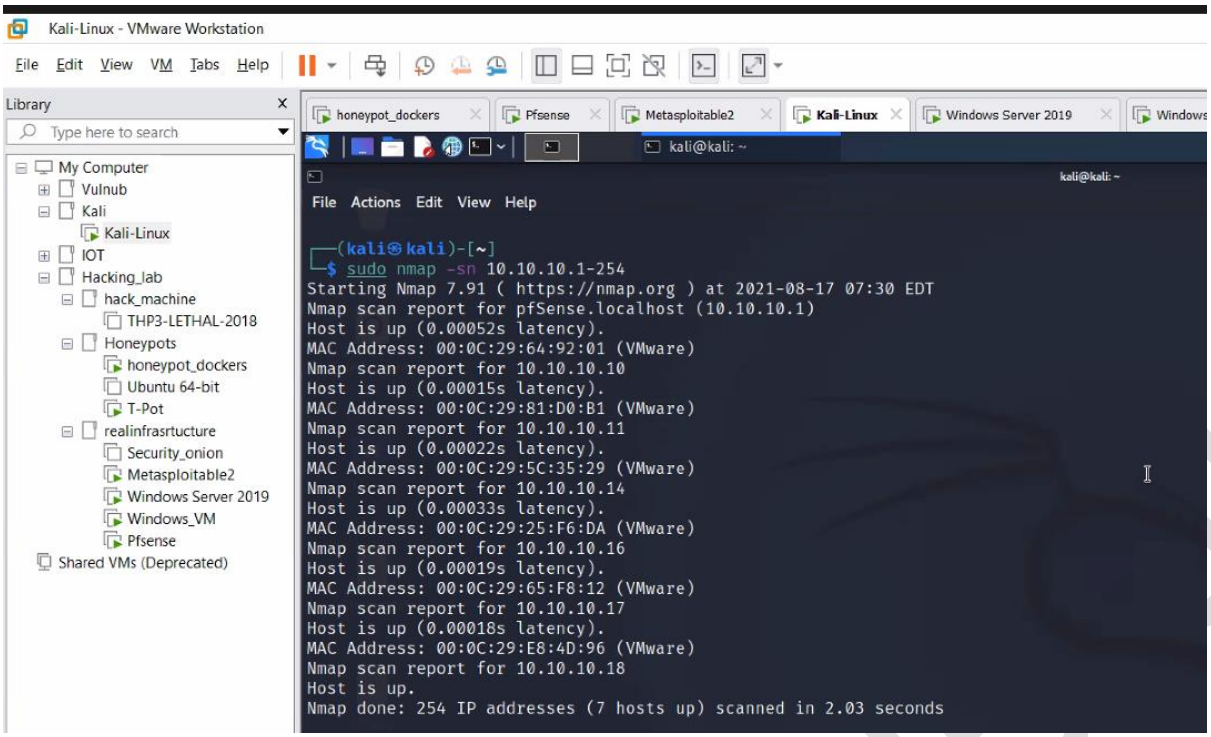
to scan for ports in this range



Put in t-pot ip
Every port is open, ports are gateway to any system e.g. http port for website, these are services which are open for attacker to attack

Sweep scan for entire network

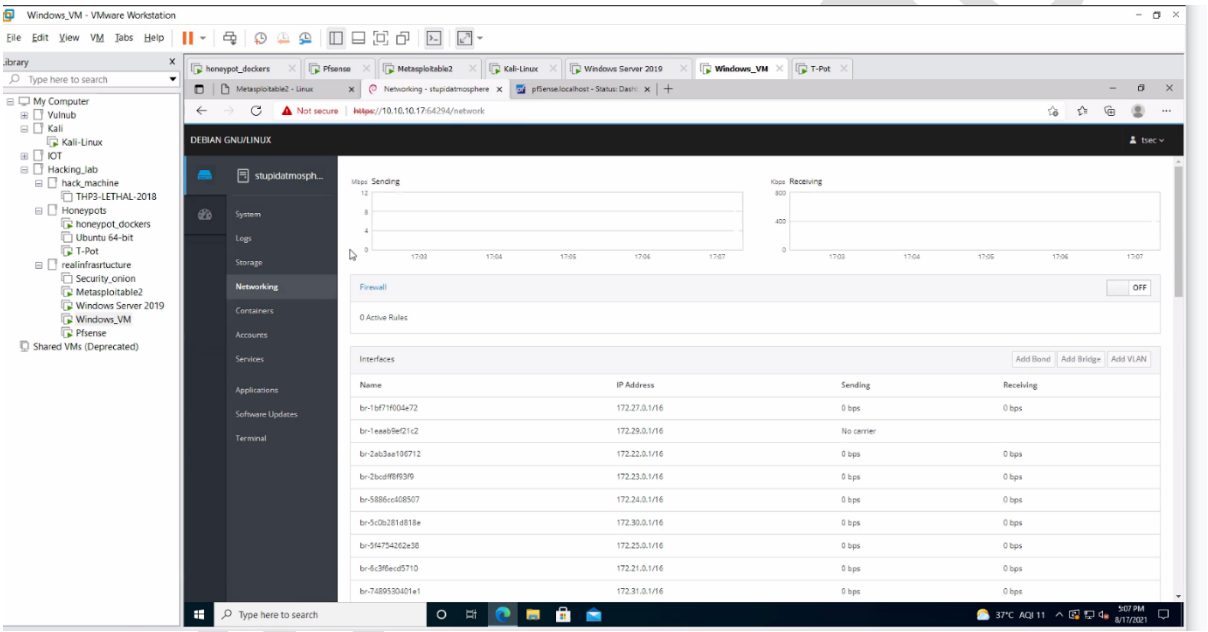
Network sweeping




```
Host is up.
Nmap done: 254 IP addresses (7 hosts up) scanned in 2.03 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -A 10.10.10.18
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-17 07:32 EDT
Nmap scan report for 10.10.10.18
Host is up (0.000085s latency).
All 1000 scanned ports on 10.10.10.18 are closed

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.58 seconds

┌──(kali㉿kali)-[~]
└─$ nmap -A 10.10.10.16
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-17 07:32 EDT
Nmap scan report for 10.10.10.16
Host is up (0.00033s latency).
All 1000 scanned ports on 10.10.10.16 are closed

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.38 seconds
```

```
┌──(kali㉿kali)-[~]
└─$ sudo nmap -O 10.10.10.14
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-17 07:35 EDT
Nmap scan report for 10.10.10.14
Host is up (0.00033s latency).
Not shown: 987 filtered ports
PORT     STATE SERVICE
53/tcp   open  domain
88/tcp   open  kerberos-sec
135/tcp  open  msrpc
139/tcp  open  netbios-ssn
389/tcp  open  ldap
445/tcp  open  microsoft-ds
464/tcp  open  kpasswd5
593/tcp  open  http-rpc-epmap
636/tcp  open  ldapssl
3268/tcp open  globalcatLDAP
3269/tcp open  globalcatLDAPssl
3389/tcp open  ms-wbt-server
5357/tcp open  wsdapi
MAC Address: 00:0C:29:25:F6:DA (VMware)
Warning: OSScan results may be unreliable because we could not find at least 1 open and 1 closed port
OS fingerprint not ideal because: Missing a closed TCP port so results incomplete
No OS matches for host
Network Distance: 1 hop

OS detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.68 seconds
```

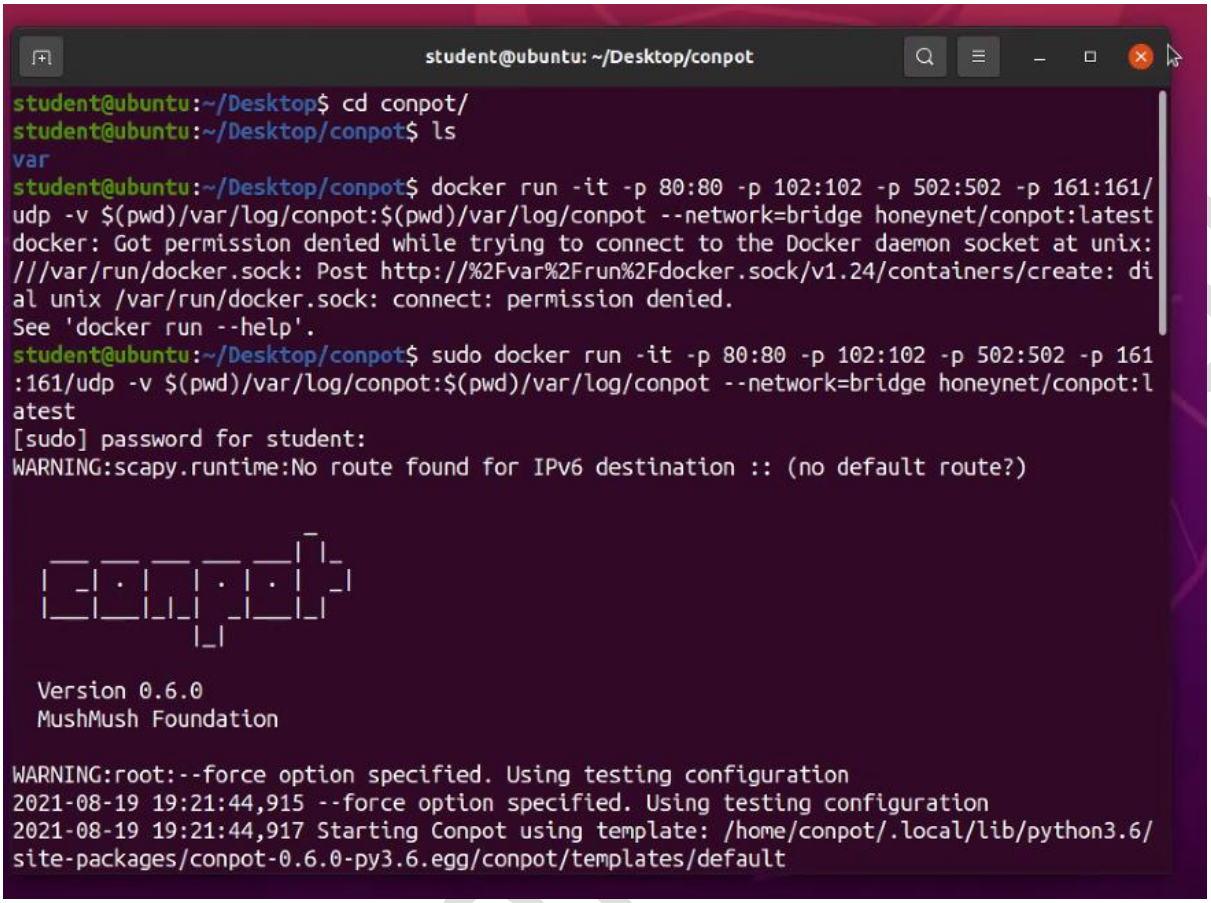
Scanning window server holding active directory

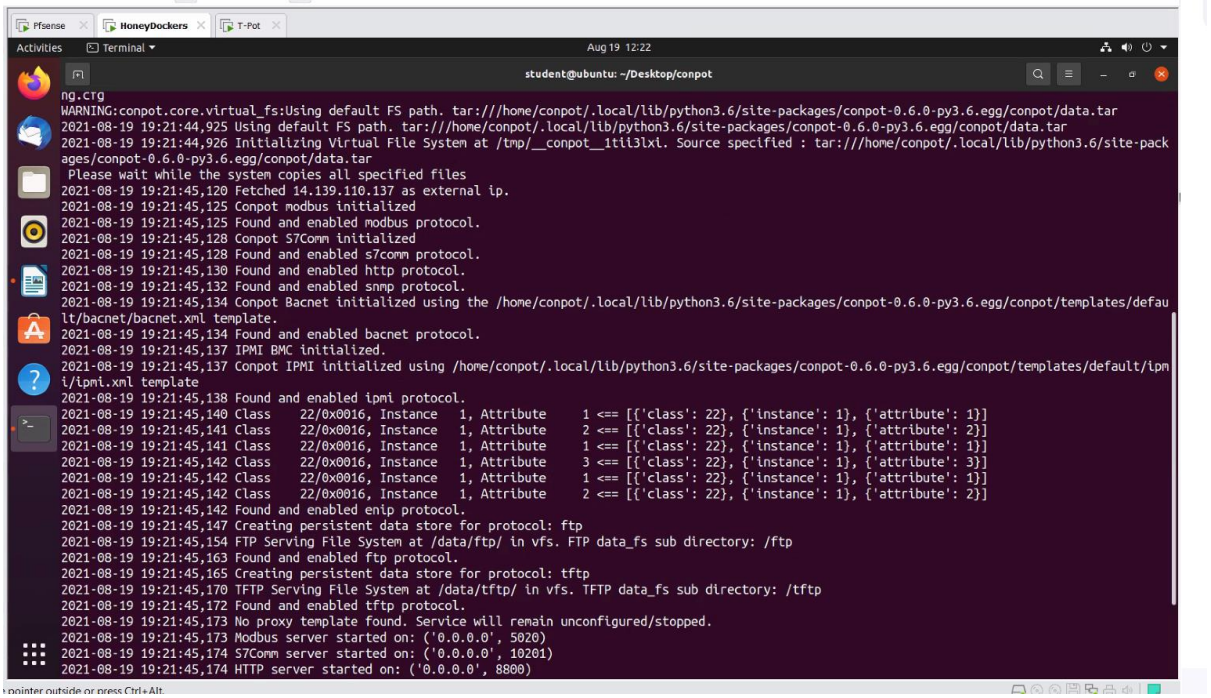


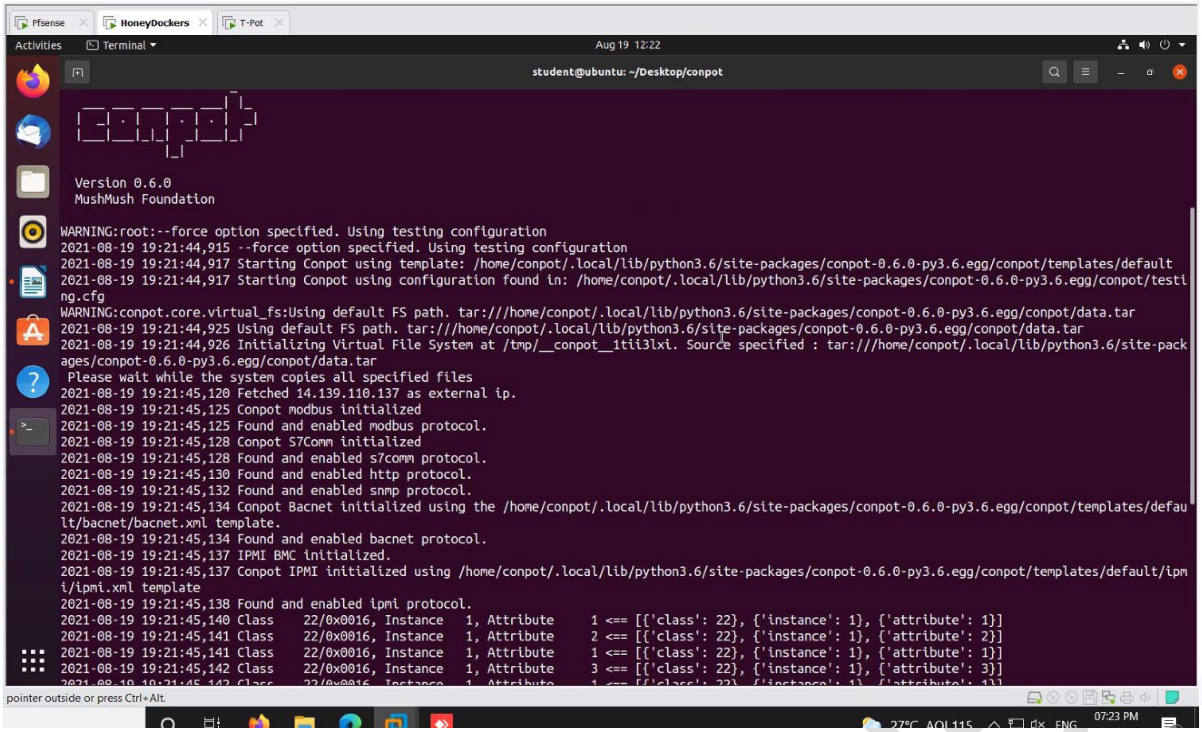
Honeypot capturing traffic caused by hacker scanning

Expoiting / attacking the honeypot:

How easy is it to attack honeypot and t-pot

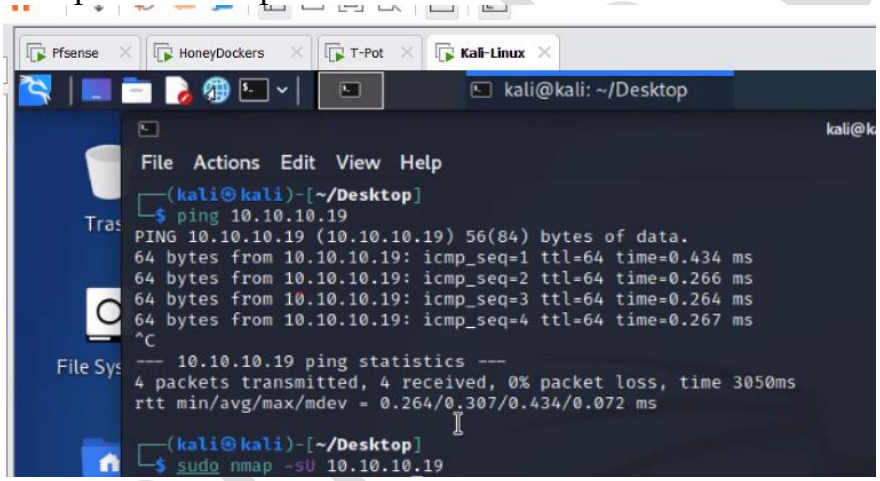






Attacking the honeypot:

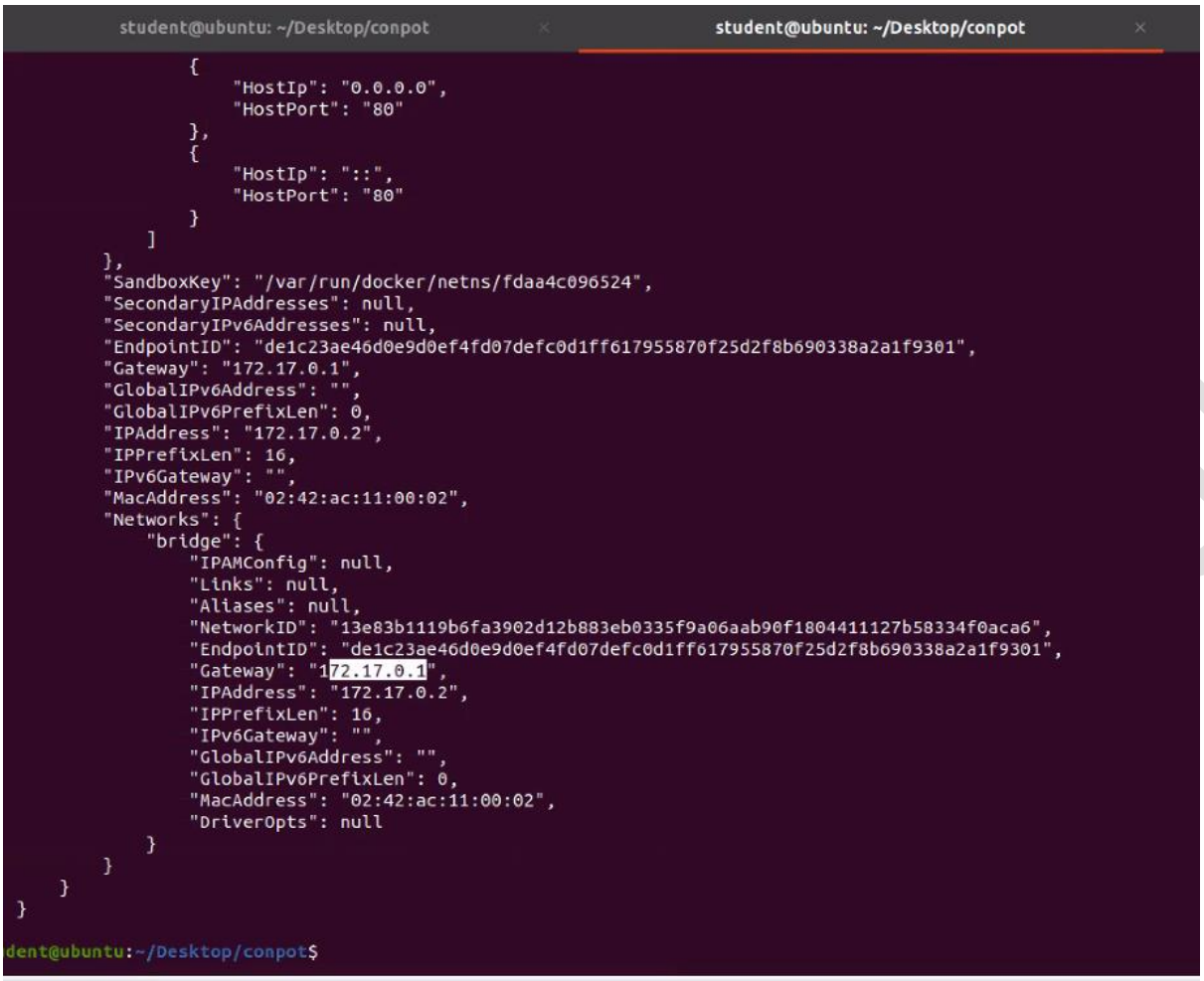
- Nmap - scan of t-pot



- Open port is snmp so snmp can be attacked - snmp organize info between devices

Trouble finding ip of honeypot

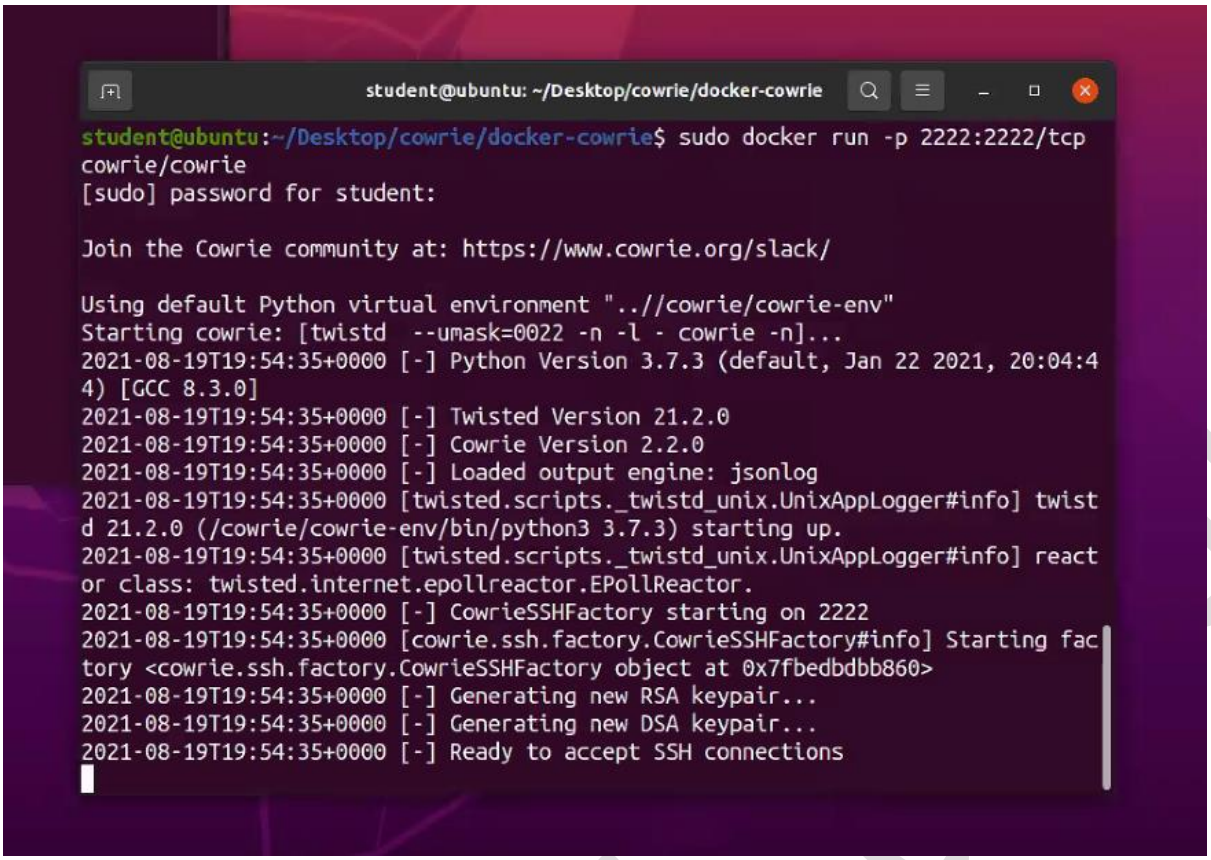
Cowrie honeypot - medium interaction honeypot capable of emulating the shell and capturing each command



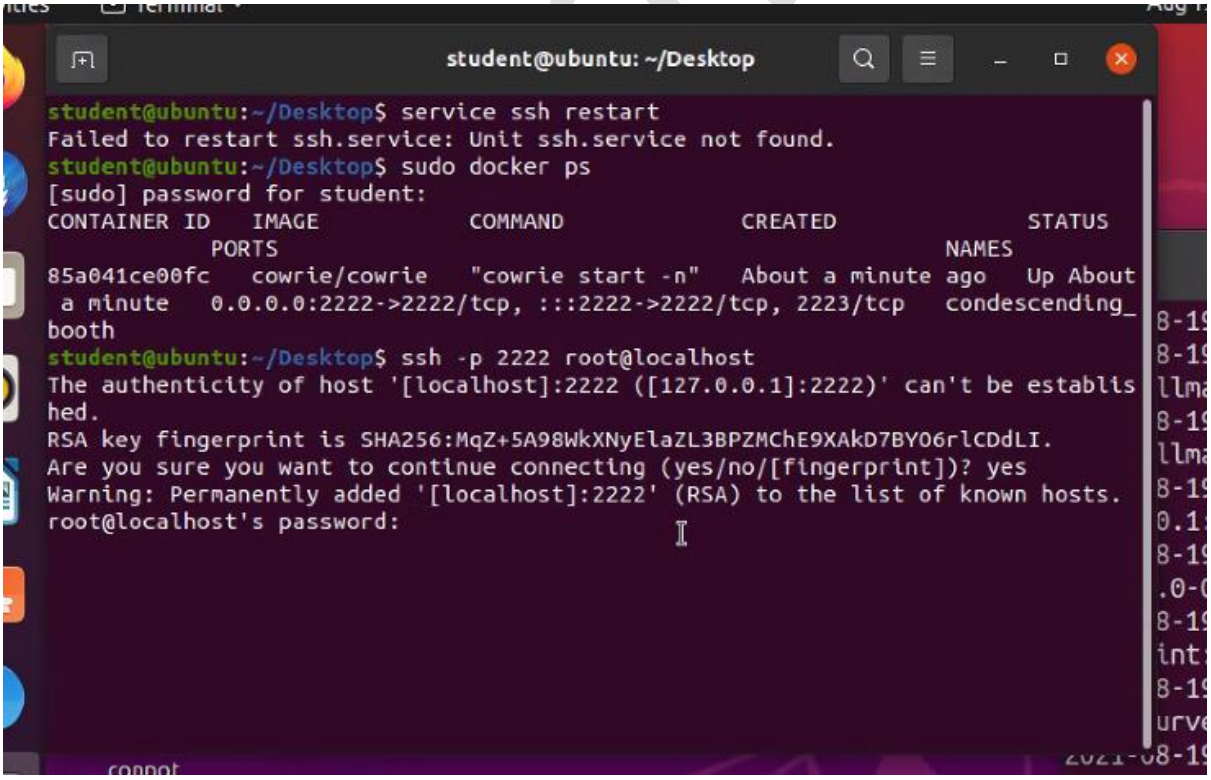
IP of t-pot

Cowrie is running:

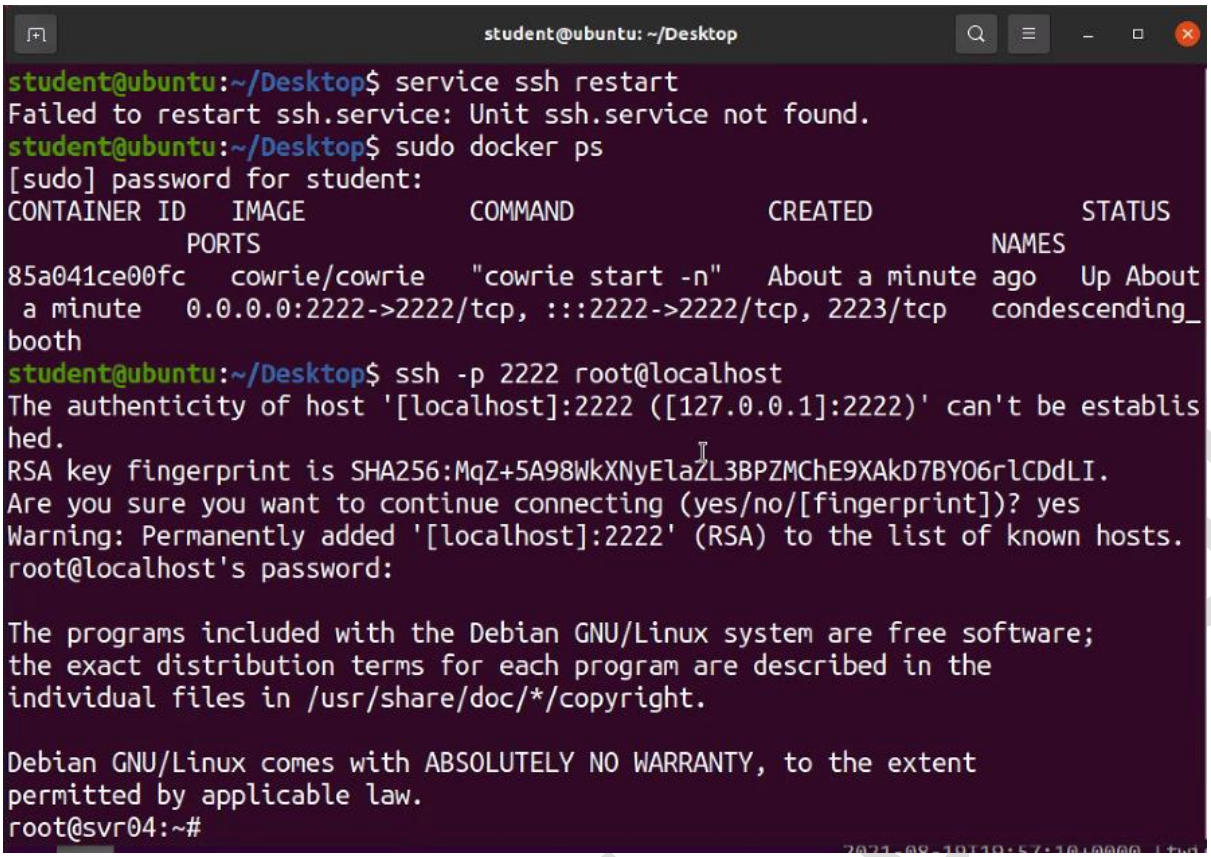
Samar

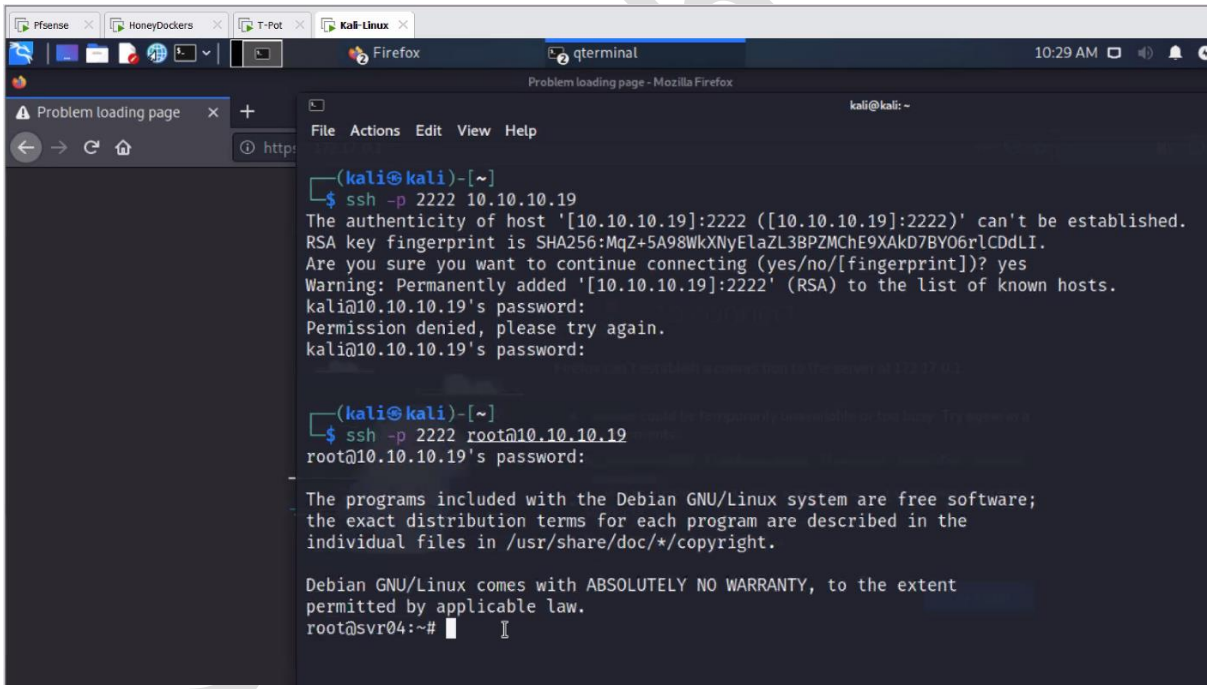


SSH (secure share service) - helps connect to terminal of other device

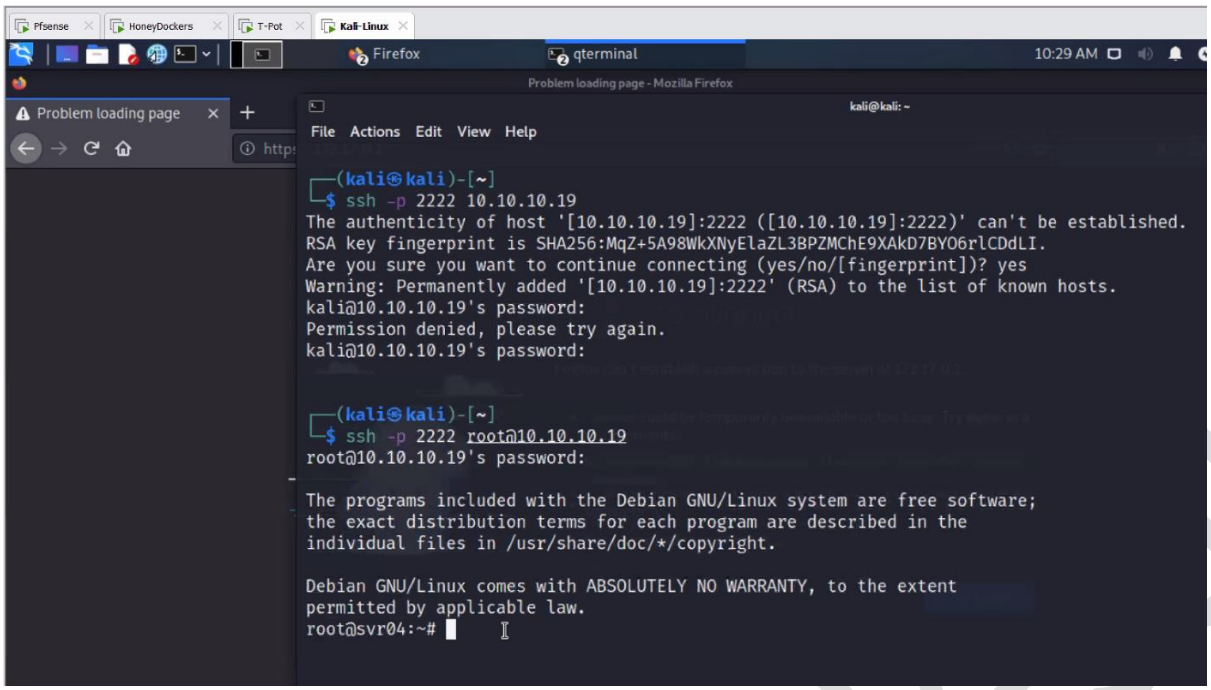


Captured the event, can see the logs showing that someone is trying to attack

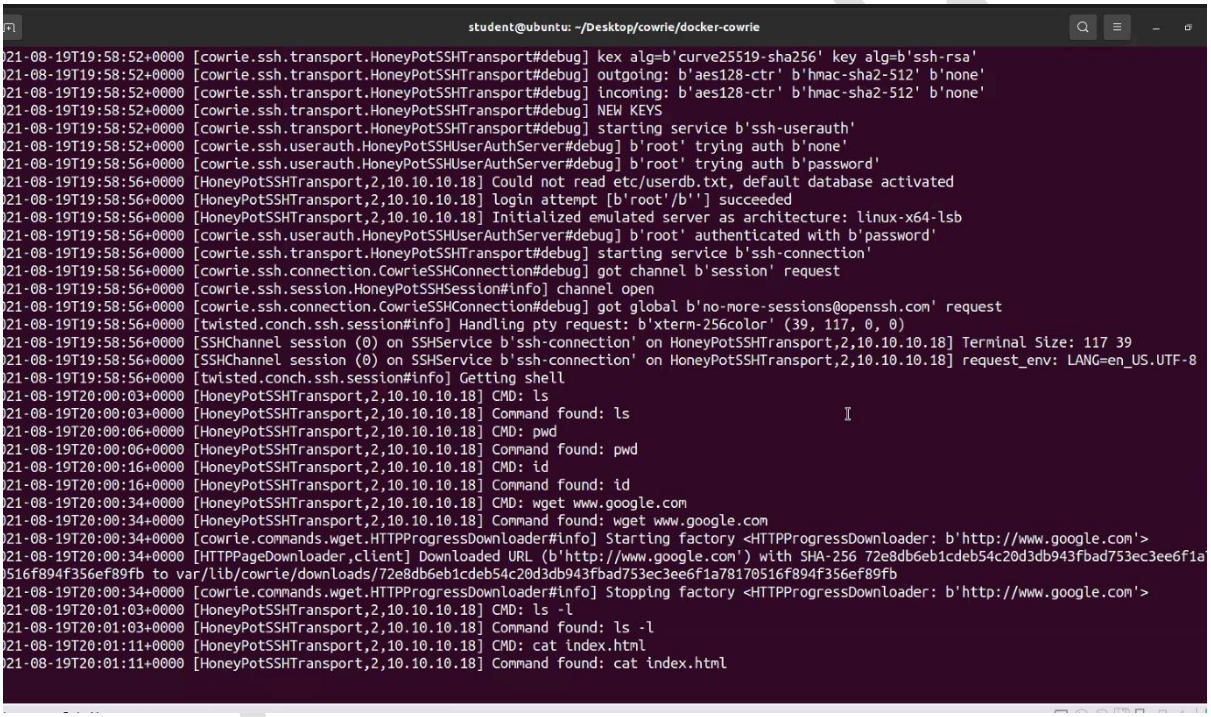


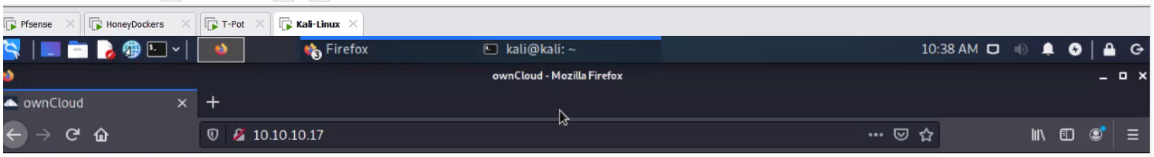


Attacker performed ssh successfully and attacker is in directory:



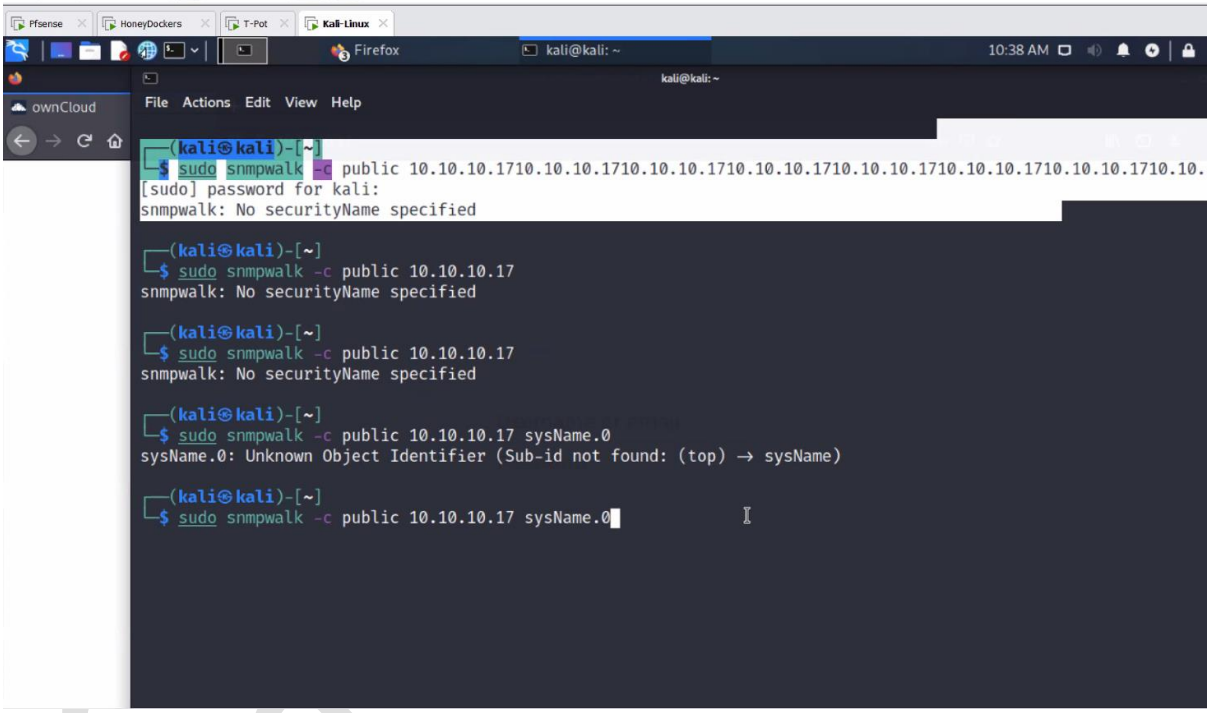
Honeypot capturing data in logs:

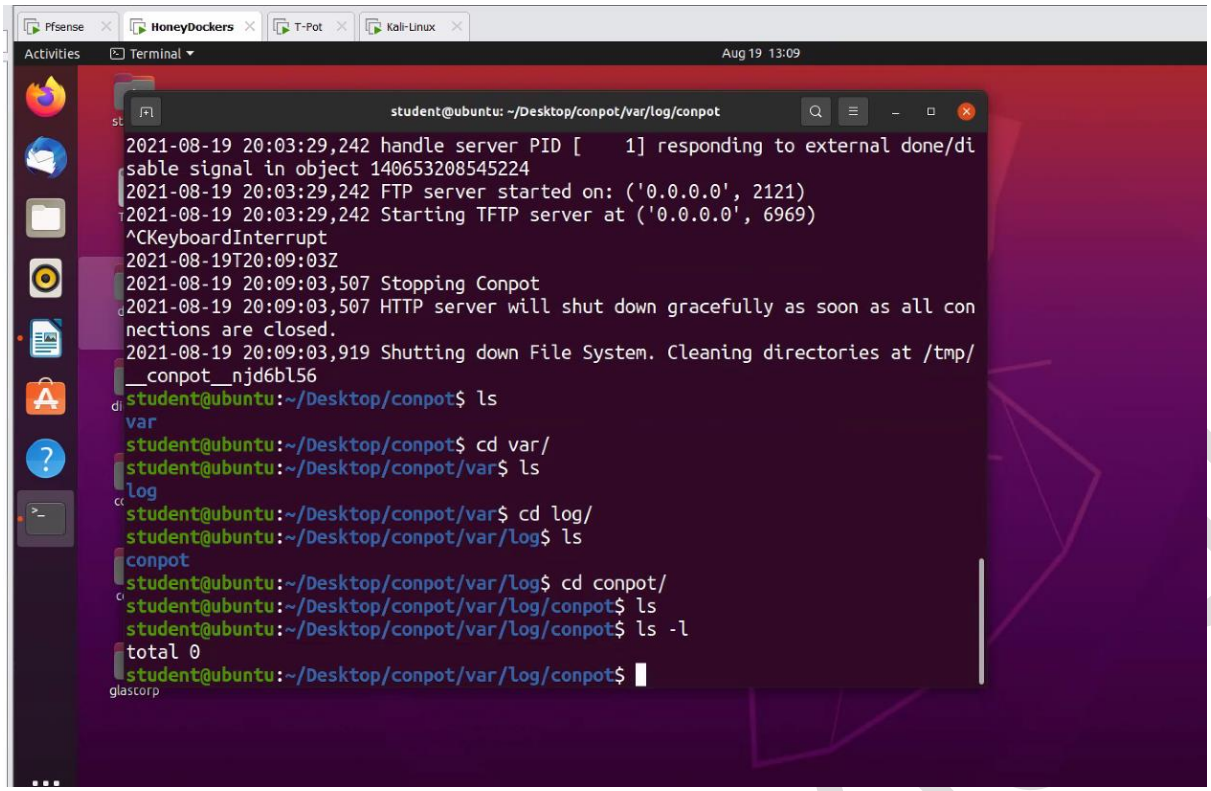




Username or email

Password





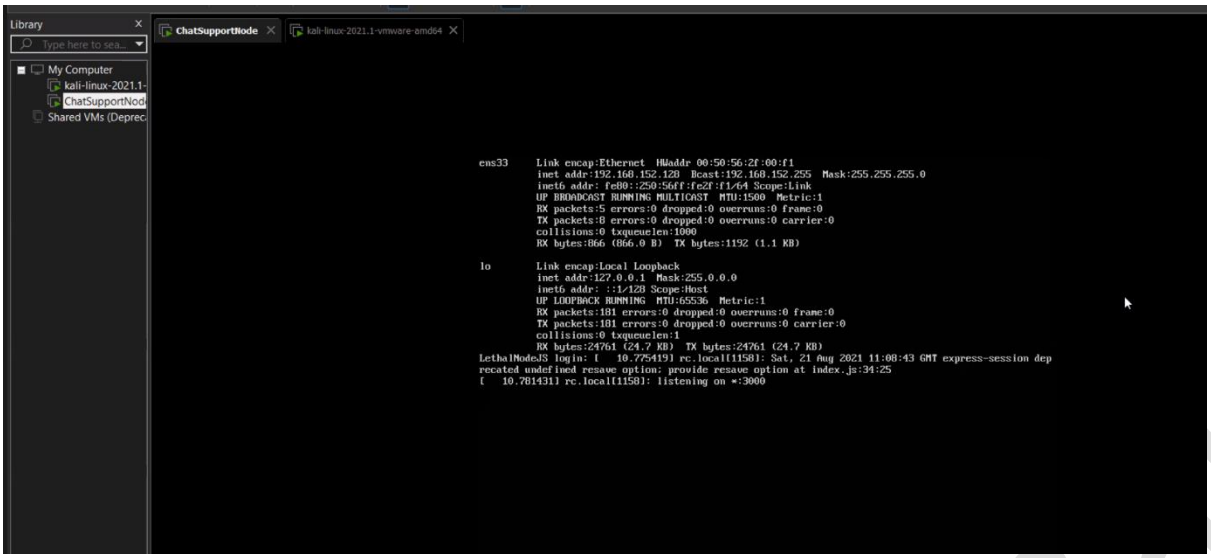
WebTrap honeypot

This project is designed to create deceptive webpages to deceive and redirect attackers away from real websites. The deceptive webpages are generated by cloning real websites, specifically their login pages. For further reading material on the tool development,

Companies keep their website running, anyone can use website. Attackers exploit this opportunity.

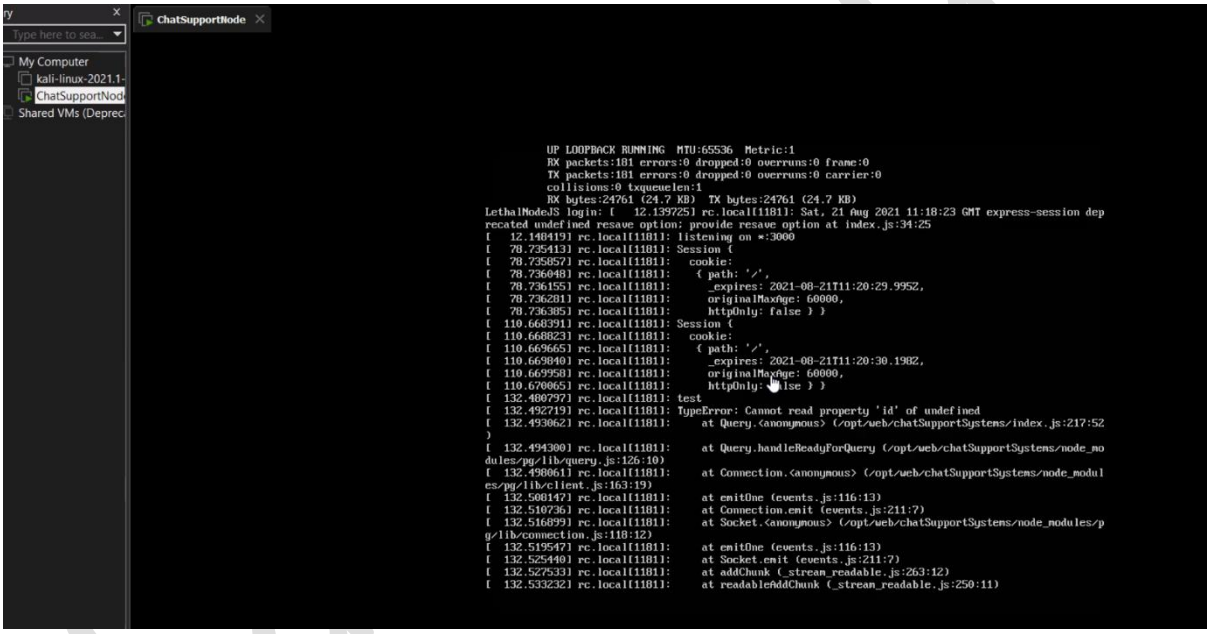
OWASP provides web vulnerability reports for free, top 10 web application security risks. Hacker uses bursuite, performs man-in-middle attack. When you go onto website communication is through URL, Intercept URL, can modify URL to try hack web server.

Samarth



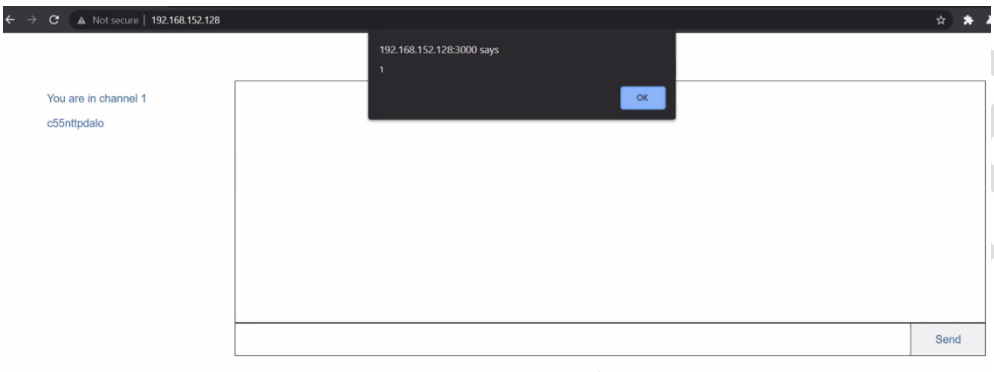
XSS dangerous, cross site scripting, causes website to go down in minutes.
This web honeypot machine is using node js, a javascript library.

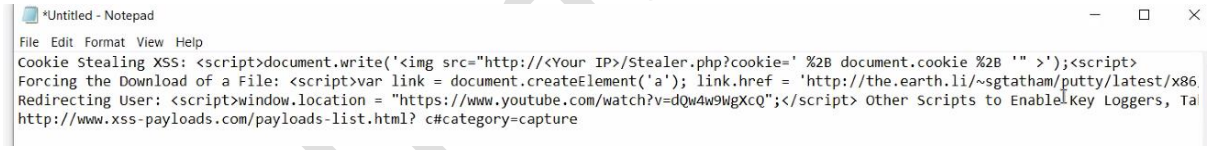
Web honeypot generating logs:



Burp intercepting request, capturing request on burpsuite:

Identifying website vulnerable to XSS:



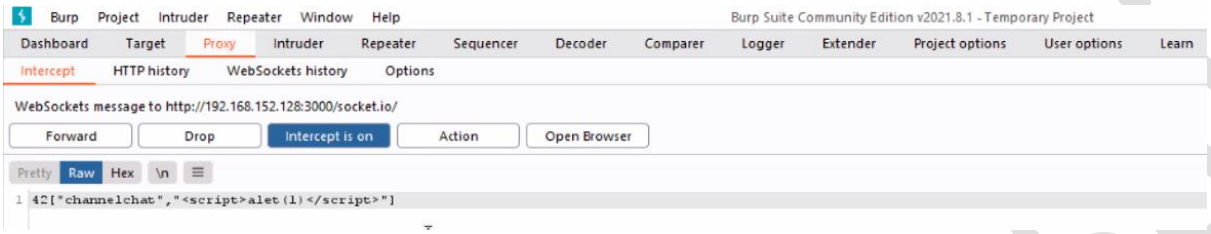


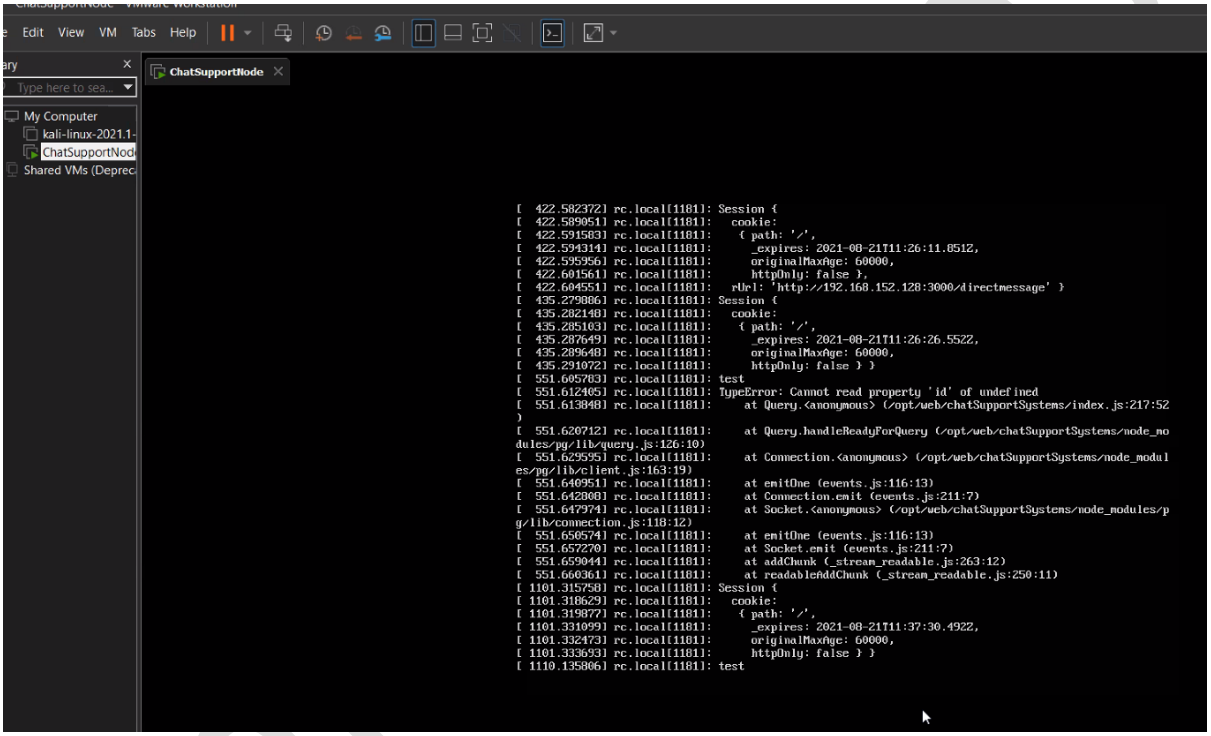
Use body tag togenerate XSS

<b onmouseover=alert('XSS')>Click Me! <svg onload=alert(1)>
<body onload="alert('XSS')">
<img src="http://test.cyberspacekittens.com" onerror=alert(document.cookie);>

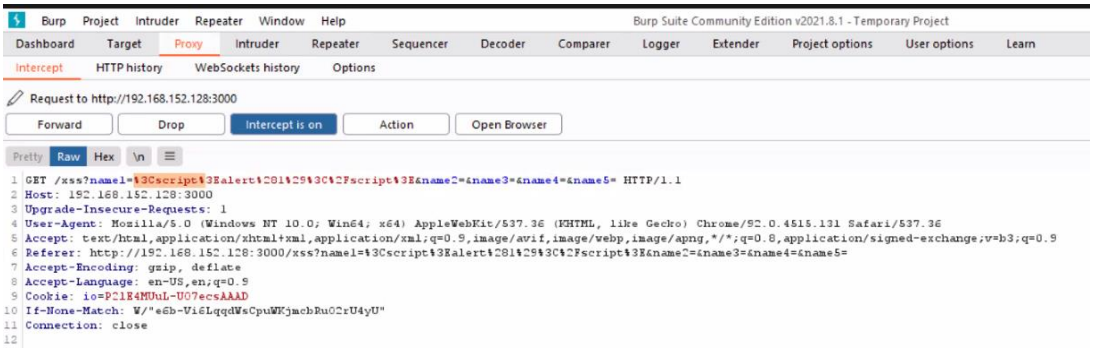
Xss hunter.com

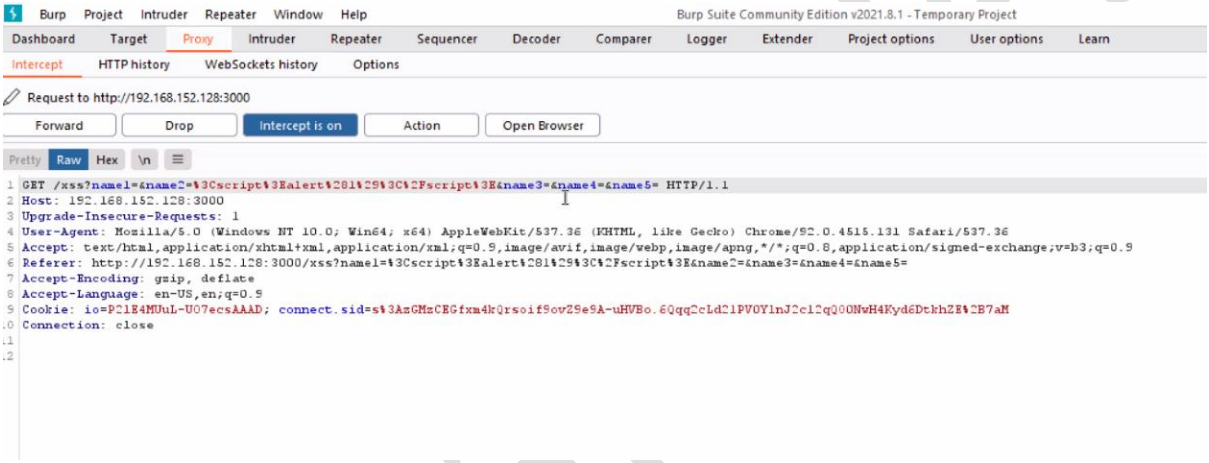
Capturing xss script on burpsuite





When you're sending a request, request in encoded and then sent to server. This encoding


Burp Project Intruder Repeater Window Help
Burp Suite Community Edition v2021.8.1 - Temporary Project
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Request to http://192.168.152.128:3000
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex \n
GET /xss?name1=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&name2=&name3=&name4=&name5= HTTP/1.1
Host: 192.168.152.128:3000
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.152.128:3000/xss?name1=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&name2=&name3=&name4=&name5=
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: io=P2lE4MUuL-U07ecsAAAD
If-None-Match: W/"e6b-Vi6LqqdWsCpuWKjmcbRu02rU4yU"
Connection: close


Burp Project Intruder Repeater Window Help
Burp Suite Community Edition v2021.8.1 - Temporary Project
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Intercept HTTP history WebSockets history Options
Request to http://192.168.152.128:3000
Forward Drop Intercept is on Action Open Browser
Pretty Raw Hex \n
GET /xss?name1=&name2=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&name3=&name4=&name5= HTTP/1.1
Host: 192.168.152.128:3000
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Referer: http://192.168.152.128:3000/xss?name1=%3Cscript%3Ealert%281%29%3C%2Fscript%3E&name2=&name3=&name4=&name5=
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: io=P2lE4MUuL-U07ecsAAAD; connect.sid=s%3AzGMzCEGfxm4kQrsoif9ovZ9e9A-uHVBo.6Qqq2cLd21PV0YlnJ2c12qQ00NwH4Kyd6DtkhZE%2B7aM
Connection: close

Samar Desai

Exercise 2

Unescaped String Interpolation

```
p No results found for ! {name2}
```

name2

No results found for

Exercise 3

Escaped String Interpolation into dynamic inline Javascript

```
script.
  var user3 = #{name3};
.
.
p No results found for #{name3}
```

<script>alert(1)</script>

No results found for

Exercise 4

Exercise 5

Submit

Samarth desai

Exercise 4

Unescaped buffered code

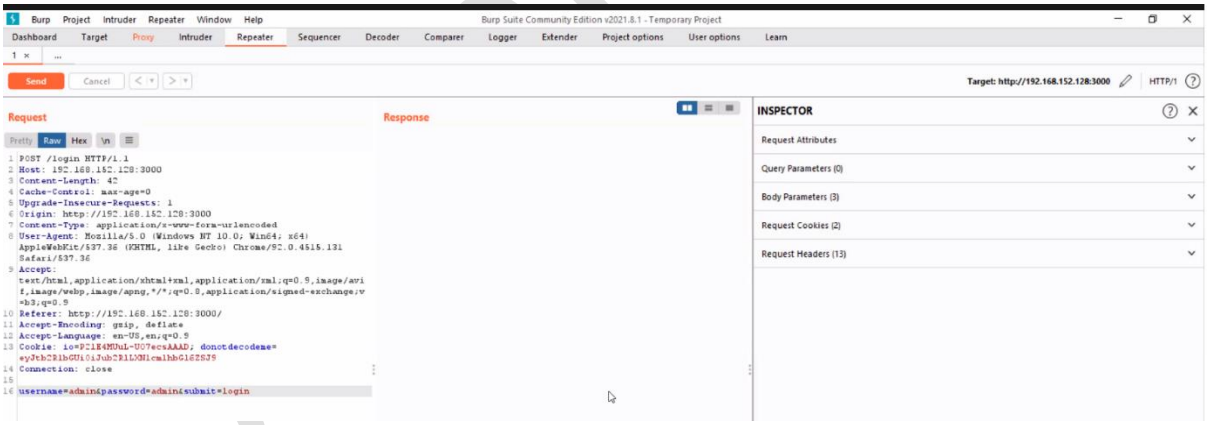
```
p!= 'No results found for '+name4
```

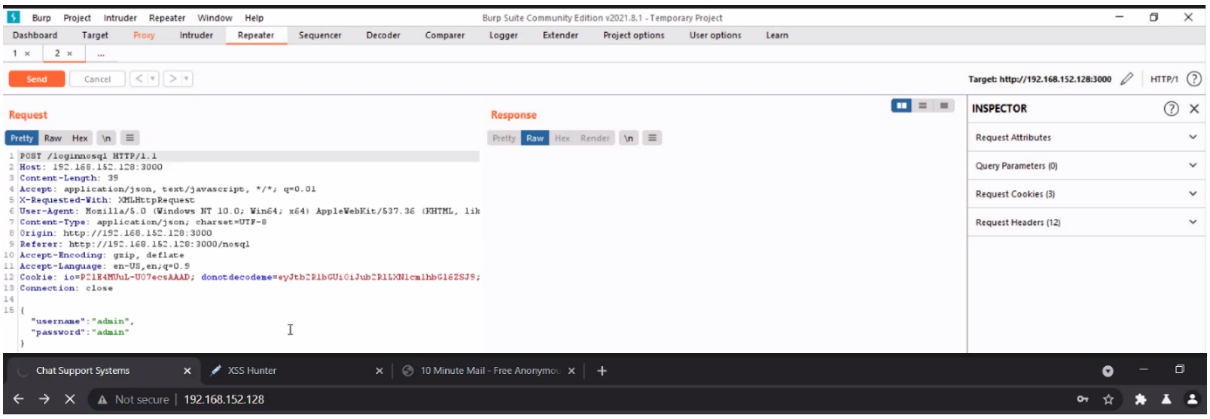
<script>alert(1)</script>

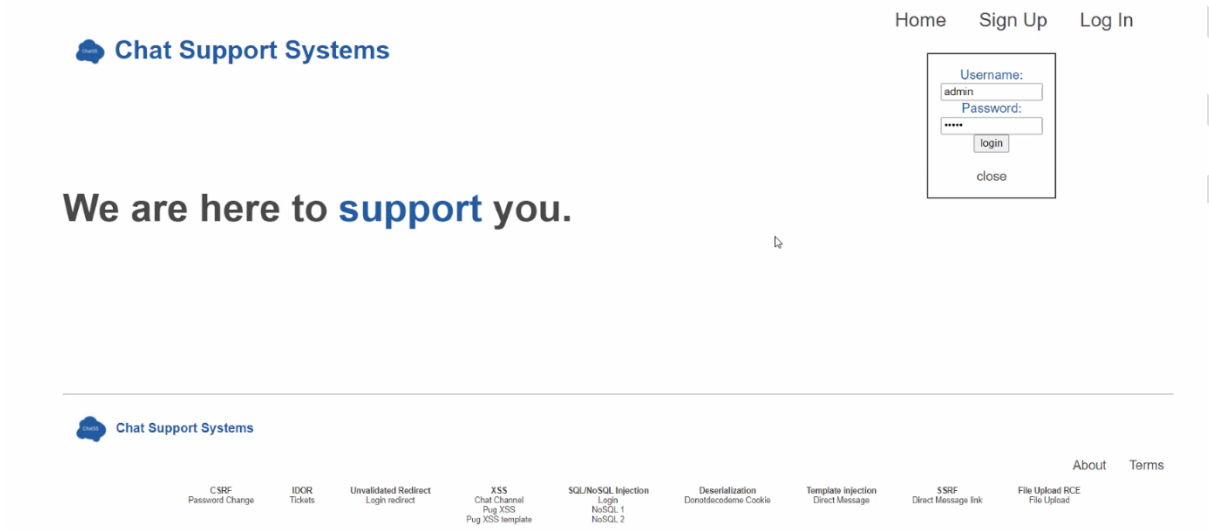
Removes tags brackets

High level of XSS

SQL injection




Burp Project Intruder Repeater Window Help
Burp Suite Community Edition v2021.8.1 - Temporary Project
Dashboard Target Proxy Intruder Repeater Sequencer Decoder Comparer Logger Extender Project options User options Learn
Send Cancel
Target: http://192.168.152.128:3000
HTTP/1
Request
Response
INSPECTOR
Request Attributes
Query Parameters (0)
Request Cookies (3)
Request Headers (12)
POST /loginnosql HTTP/1.1
Host: 192.168.152.128:3000
Content-Length: 39
Accept: application/json, text/javascript, */*; q=0.01
X-Requested-With: XMLHttpRequest
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, lik
Content-Type: application/json; charset=UTF-8
Origin: http://192.168.152.128:3000
Referer: http://192.168.152.128:3000/nosql
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: io=P2lE4MDuL-U07ecsAAAD; donotdecodeme=eyJtb2RlbGUiOiJub2RlLXNlcmlhbGl6ZSJ9;
Connection: close
{
"username":"admin",
"password":"admin"
}
Chat Support Systems
XSS Hunter
10 Minute Mail - Free Anonymo
Not secure
192.168.152.128


Home
Sign Up
Log In
Chat Support Systems
Username:
admin
Password:
login
close
We are here to support you.

Chat Support Systems
About
Terms
CSRF
Password Change
IDOR
Tickets
Unvalidated Redirect
Login redirect
XSS
Chat Channel
Pug XSS
Pug XSS template
SQL/NoSQL Injection
Login
NoSQL 1
NoSQL 2
Deserialization
Donotdecodeme Cookie
Template injection
Direct Message
SSRF
Direct Message link
File Upload RCE
File Upload

Samar

Focusing on session redirection

Bait and Switch

About The Bait & Switch Honeypot System: Violating Networks has produced a system that reacts to hostile intrusion attempts by redirecting all traffic from “bad” IPs to a honeypot that is partially mirroring your production server. Once switched, the would-be hacker is unknowingly attacking your honeypot instead of the real data while your clients and/or users are still safely accessing the real system. Life goes on, your data is safe, and as an added benefit you are learning about the bad guy. Currently the system is based on Snort, Linux's iproute2, netfilter, and custom code. We’re planning on porting it to the BSDs as soon as possible.

• External Interface: This is the public facing interface that is either the gateway or the NATing device to your production & honeypot servers. I use etho

• External IP: What is the IP of the external interface? Should *not* be on the same subnet as your honeypot/production server IP

• Production Interface: This is the internal device on your B&S box that is your production server's gateway to the rest of the world. I’m using vmnet1.

• Production IP: IP address on the Production GW Interface. Needs to be on the same subnet as your honeypot/production server IP

• Honeypot Interface: This is the internal device on your B&S box that is your honeypot's gateway to the rest of the world. I’m using vmnet2.

• Honeypot's Gateway IP: IP address on the Honeypot GW Interface. It needs to be on the same subnet as the production gw IP - which is the same subnet as your honeypot/production server IP: 192.168.2.2 is what I use.

Samarth desai

• IP of <both> Honeypot and Production Server: Your honeypot and production server should be set up with the SAME IP address. In this case, mine have 192.168.2.10 as their address.

• Length of Time (in minutes) that the mark time should be incremented: Every time snort alerts on a sig, B&S increments the length of time that that source IP will be rerouted. How much should the time be incremented for each incident?

• Length of Time DoS Protection - Max Alerts: We don't want to have someone force a source to be rerouted forever, so we limit the number of marking increments per period of time that increment the rerouting time. Enter how many alerts per period of time is too many.

• Length of Time DoS Protection: Period ((in seconds) to look for too many marks from a single IP): this is the time period referenced in the previous step

• IP DoS Protection: How many IPs per a certain amount of time is too many? We don't want someone to spoof all our clients and quickly block them all, so we limit how many new source IPs per time period we'll reroute.

• IP DoS Protection: Within what length of time should a certain number of IPs be too many? This is the time period in seconds referenced in the previous instruction.

• Fifo File Location: This is the named pipe that snort uses to talk to baitnswitch. Where do you want it and what should it be called? /tmp/bns or something should work fine.

• Log Location: We log DoS alerts from B&S. Where should they go? I use /var/log/switchcore.log

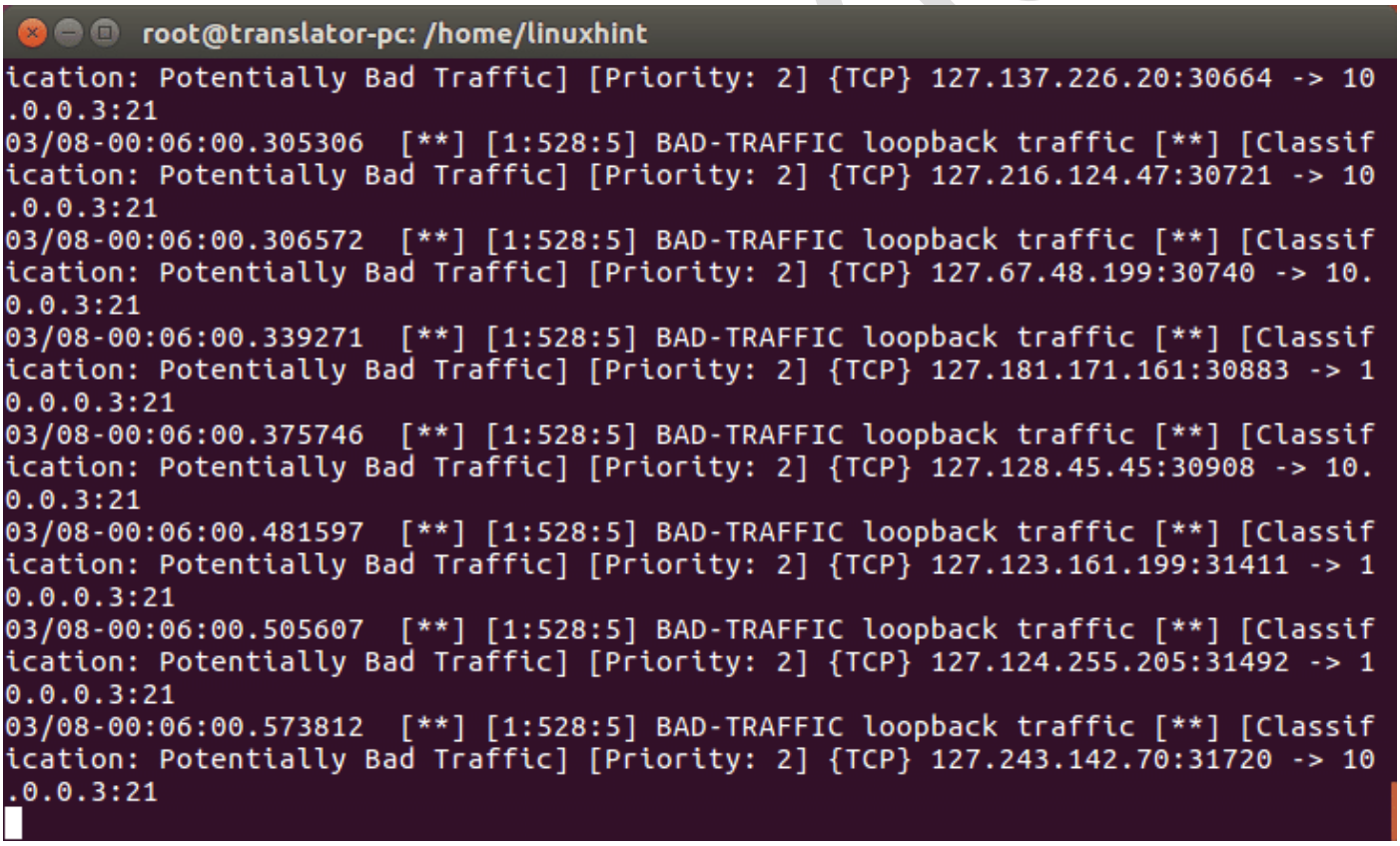
Samarth desai

• Blacklist Location: Certain IP's should always be rerouted, as they are known hostiles. Switchcore reads a list of IP's from a textfile. Where should this list be?

• Path to snort: What directory (full path to) did the snort tarball create? I use /root/dev/snort-1.9.0

Running of bait and switch

Running on VM on ubuntu and getting snort logs if it detects it we can



Results

I performed some attacks and for each type of attack I used different technique to do the attack total 10 attacks for each all as follows

DOS attack

Webapplication attacks

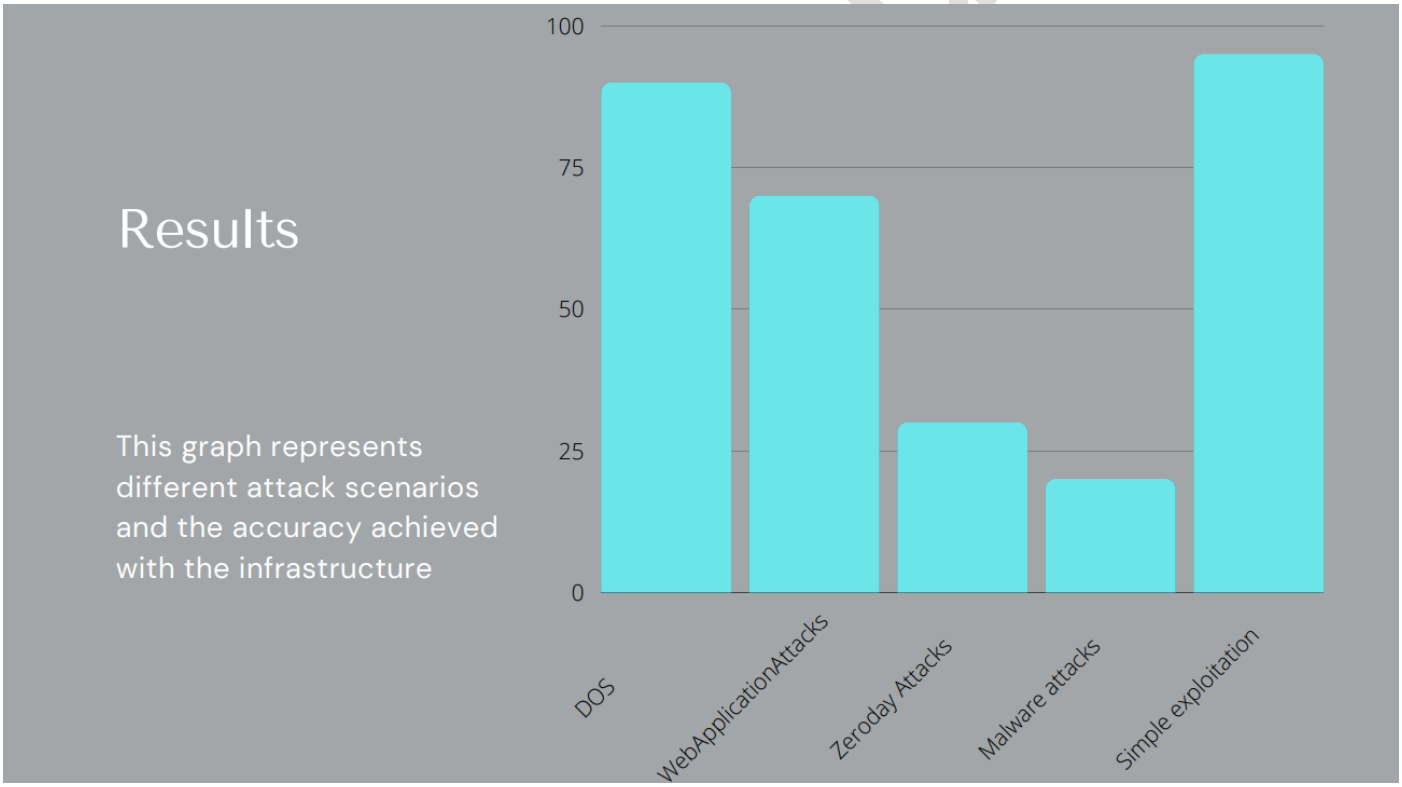
Zeroday attacks

Malware attack

Simple exploits

And with successful exploitation it was counted and for unsuccessful it was not counted

The graph shows a clear pictorial representation



Conclusion

Study highlights

Honeynet configuration with advance docker

configuration. Data collection methods using honeypot

and how effective they can be

Significant discovery

Most common attacks can be studied and some attack patterns can be determined with a deceiving machine

Interpretation of findings

Although honeypots are one of the most common ways, we can deceive attacks their applications are a wide area and can help in threat research without the risk of compromise. On the contrary honeypots and firewalls are not effective due to lack in technology and attacks can bypass them with some knowledge. Further improvement can be beneficial for this research

Implications of the research

This work can be implemented in a infrastructure to add a layer of security. Data collection can be used to analyse different attack patterns and to get information about future attacks.

Areas of improvement

This project depends on the limitations of IDS. If the IDS don't detect any abnormal activity, then the attacks cannot be avoided. Honeypots can be detected using some tools and attackers can avoid it. The accuracy of the project is not 100%. AI based threat detection IDS can be implemented on the same project. Data can be analysed with the use of AI technology and determine attack patterns

samarth desai

References

- Farouk Samu | Design and Implementation of a Real-Time Honeypot System for the Detection and Prevention of Systems Attacks (stcloudstate.edu)

- paralax/awesome-honeypots: an awesome list of honeypot resources (github.com)

- Kim M | Design and Implementation of the HoneyPot System with Focusing on the Session Redirection

- iyoung Kim, Youngsong Mun, Technical Report, "A study on intrusion responding technique using HoneyPot System,"

- Miyoung Kim, Youngsong Mun, "The Development of Honeypot System," Proceedings of the International Conference on Security

Samarth desai