

**CPA**



# **INTRUSION DETECTION & PREVENTION SYSTEMS**

**FULL NAME** Minhaj Shameer **MODULE ID** CP5UA980

**STUDENT ID** 21587119 **MODULE TUTOR** Dr. Haleema PK



# TABLE OF

# CONTENTS

1. INTRODUCTION TO HONEYPOTS: A PRIMER	03
2. DECODING THE MECHANICS: HOW HONEYPOTS OPERATE?	03
3. HONEYPOTS DEMYSTIFIED: UNCOVERING THEIR PURPOSE	04
4. EXPLORING HONEYPOT VARIETIES	05
5. ADVANTAGES OF DEPLOYING HONEYPOTS	06
6. DRAWBACKS OF HONEYPOTS: EXPLORING LIMITATIONS	07
7. HONEYPOT APPLICATIONS IN PRACTICAL SCENARIOS	08
8. EMULATING HONEYPOTS: A PRACTICAL APPROACH	11
9. CONCLUDING REMARKS	14
10. BIBLIOGRAPHIC DATA	15



## INTRODUCTION TO HONEYPOTS: A PRIMER

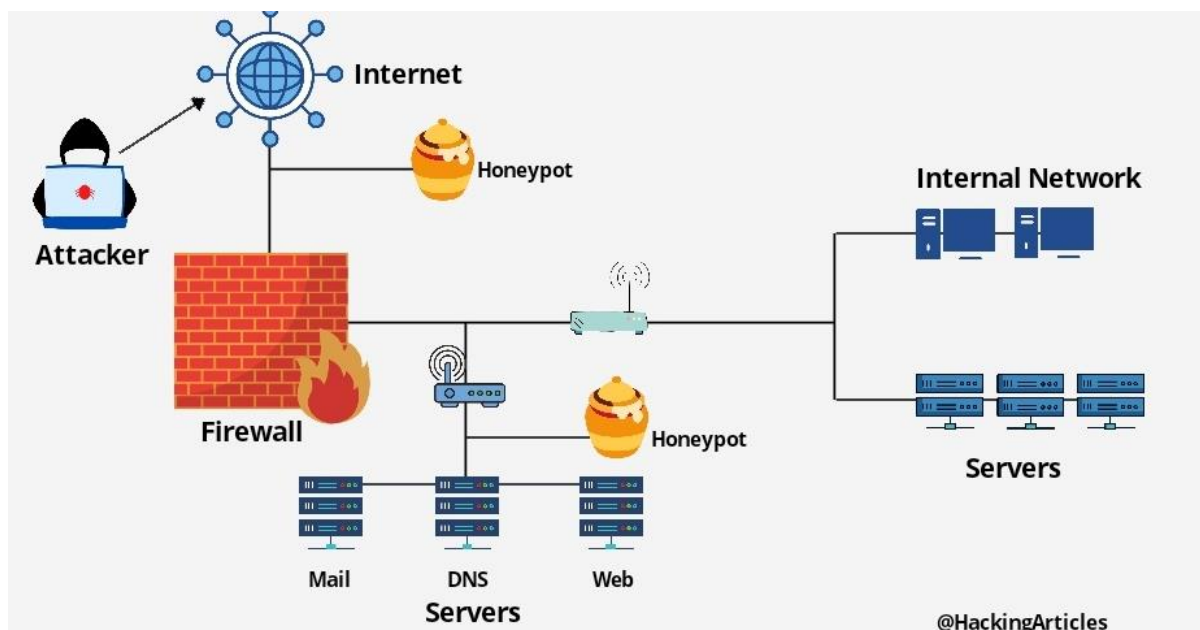
The idea of honeypots appears as an important strategic asset for enterprises looking to improve their ability to protect systems against cyber-attacks in the fast-paced quickly field of cybersecurity. A honeypot, which is an imitation system, is carefully crafted to draw in and tempt possible attackers by offering a haven in which their strategies, methods, and objectives may be watched over and examined. By strategically placing honeypots, security teams may get crucial information on the tactics and behaviors of attackers, which in turn helps them create stronger defenses.

The importance of honeypots arises from their capacity to imitate weak systems or networks, diverting attackers' focus from vital infrastructure. Organizations may take a preventive strategy for threat detection by keeping an eye on how attackers are interacting with these fictitious systems. This allows them to identify harmful behaviors and infiltration attempts early on. Additionally, the information gathered by honeypots may be used to improve threat awareness, incident response features, and current security protocols—all of which can strengthen an organization's overall security architecture. [1]

## DECODING THE MECHANICS: HOW HONEYPOTS OPERATE?

A honeypot is an advanced cybersecurity tool that imitates an actual computer system to draw in thieves by making itself seem like a desirable target. The apps and data on this spoof system are designed to look and feel like those of a real system, such as a business's payment mechanism, which is a popular target for hackers looking for credit card details. The main purpose of the honeypot is to entice intruders into a supervised area so that their activities may be observed and examined. This makes it possible for IT personnel to watch the strategies and tools that attackers employ, evaluate how well the system is defended, and spot flaws that require attention. Honeypots help organizations improve their detection of breaches and get insights into the tactics and behaviors of attackers.

Honeypots work by purposefully creating security holes to draw in intruders. For example, they can use insecure passwords to attract additional attackers, or they might have unsecured ports that react to port scans, a method for finding unsecured ports on a network. A honeypot's objective is to gather information about prospective threats and improve a company's capacity to identify and respond to assaults, in contrast to typical security measures that explicitly try to prevent breaches. By concentrating their attention on the most important vulnerabilities and risks, security teams may improve the efficacy and efficiency of their cybersecurity tactics. [2]



## HONEYPOTS DEMYSTIFIED: UNCOVERING THEIR PURPOSE

Honeypots are essential for improving cybersecurity since they fulfill a variety of strategic functions, including:

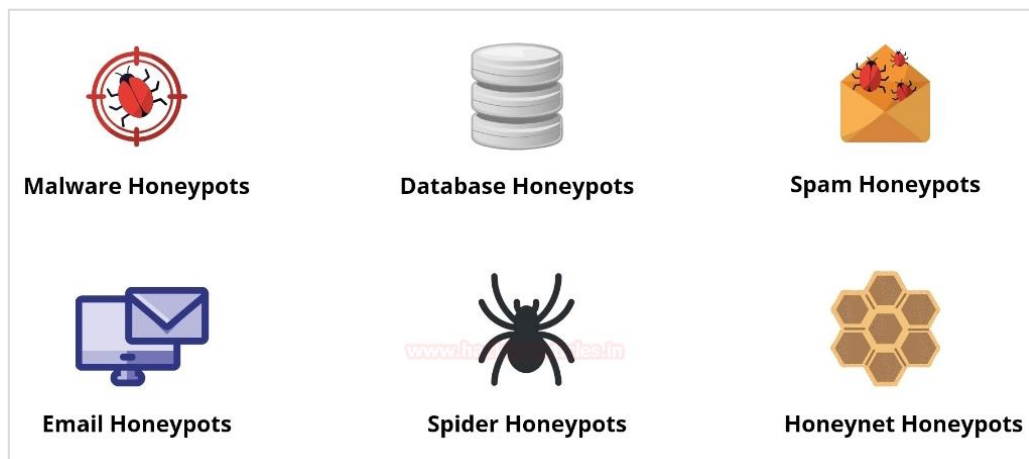
1. **Threat intelligence gathering:** By serving as bait that draws in bad actors, honeypots reveal important information about their strategies, equipment, and goals. Organizations may better recognize possible risks and create proactive defense systems by examining the behavior of these attackers.
2. **Attack Detection and Monitoring:** Such systems are made to keep an eye out for and identify a variety of assaults, such as application-level attacks, malware infections, and intrusions into networks. Honeypots aid in the creation of more advanced systems to detect and avoid intrusions by gathering comprehensive information about these instances.
3. **Research and Development:** Security researchers may analyze and comprehend the changing strategies used by hackers by using honeypots as a laboratory. By helping to create new security techniques and innovations, this study increases the resistance of platforms and networks against assaults in the near term.
4. **Legal and Law Enforcement Support:** They can serve as traps to collect data against hackers and can also be utilized for legal tasks. As part of the larger effort to combat cybercrime, law enforcement organizations may find a great use for the information gathered from these honeypots in their analyses and

trials.

5. **Strengthening Cybersecurity Posture:** Honeypots assist companies in regularly evaluating and strengthening their cybersecurity posture by giving immediate information on attacker actions and behaviors. By taking a proactive stance, vulnerabilities may be found and fixed before they are used, thus lowering the likelihood that cyberattacks would be effective.

In conclusion, honeypots are a flexible instrument in the cybersecurity toolbox that provide several advantages, including security gathering and legal support. In the continuous fight against cybercrime, their capacity to imitate tempting victims for attackers while securely observing their activities makes them a useful weapon. [3]

## EXPLORING HONEYPOT VARIETIES:



Many kinds of honeypots are intended to draw in and keep an eye on kinds of security threats:

1. **Malware Honey pots:** To draw malware assaults, these honeypots imitate software programs and APIs. They are used to find holes in APIs and, by comprehending the flaws that malware takes advantage of, to create anti-malware software.
2. **Database honeypots:** These are intentionally placed to draw in offenders who can get past firewalls, and they include fictional and susceptible datasets. They keep an eye on the kinds and quantity of database-targeting assaults, revealing weaknesses in database security.
3. **Spam Traps:** In contrast to email honeypots, spam traps are frequently connected to blacklist-maintaining organizations and Internet service providers

(ISPs). Their purpose is to apprehend spammers who buy or harvest email lists, resulting in an excessive number of unsolicited messages. Both passive and aggressive spam traps exist.

4. **Email Honeypots:** They're specialized email addresses created to draw in and expose spammers. Since actual individuals seldom use them, it is safe to presume that any correspondence received to these domains is most certainly spam.
5. **Spider Honeypots:** Designed to entice automated crawlers, spider honeypots provide links to easily accessible webpages to capture bots that are malicious and ad-network crawlers. This kind of honeypot aids in recognizing and analyzing automated threat behavior.
6. **Honeynet Honeypots:** A honeynet is a complete network of honeypots, as opposed to an exclusive honeypot. It functions on a bigger scale and provides a thorough space for researching and keeping an eye on a variety of cyber threats. To draw in and examine the actions of cybercriminals, honeynets are capable of imitating full networks or devices.

In the context of cybersecurity, each kind of honeypot has a distinct function, ranging from drawing in and examining viruses and trash to offering an all-encompassing setting for examining a wide range of cyber threats. [2]

## ADVANTAGES OF DEPLOYING HONEYPOTS:

1. **Real Data Collection:** Honeypots gather insightful information on the tactics, methods, and procedures (TTPs) used by attackers by using data from real attacks and unauthorized activity. Understanding the changing threat landscape and enhancing security defenses need the use of this data.
2. **Reduced False Positives:** Since users cannot legitimately access honeypots, they considerably lessen the issue of false positives, which may be produced by typical cybersecurity detection systems in large quantities. By concentrating on genuine dangers, real security problems may be prioritized and addressed more skillfully.
3. **Cost-Effectiveness:** Since they solely deal with malicious activity, they don't need high-performance resources to handle a lot of network traffic, making them cost-effective. They are therefore a wise investment for businesses trying to improve their security posture without going over budget.
4. **Encryption circumvention:** Illegal behavior can be recorded by honeypots even in cases when attackers employ encryption. This feature is crucial for identifying

covert threats and making sure that secure interactions don't act as an attacker's haven.

5. **Network Penetration Testing and Reconnaissance:** Honeypots are practical instruments for these two processes. They let security teams evaluate the efficacy of protective measures in an isolated setting, uncover vulnerabilities, and watch and analyze the way intruders navigate the system by imitating genuine systems. [3]

## DRAWBACKS OF HONEYPOTS: EXPLORING LIMITATIONS

1. **Attack Risk:** Honeypots can make your surroundings more dangerous. A compromised honeypot may be used to assault, break into, or damage other systems or businesses. The intricacy and layout of the honeypot determine how risky it is. While more complicated honeypots, including those that simulate full operating systems, might present serious concerns if hacked, simpler honeypots carry less risk.
2. **Narrow Field of View:** Honeypots' capacity to keep an eye on activities other than their direct contacts is restricted. They are not aware of wider network activity unless they are explicitly targeted; they can only identify assaults that are directed at them. Because of this drawback, honeypots could overlook important security events taking place in other parts of the network.
3. **Vulnerability to Fingerprinting:** A lot of honeypots, particularly the commercial varieties, are vulnerable to fingerprinting. This happens when a hacker determines a honeypot's actual identity based on certain traits or behaviors. Attackers may be able to impersonate other systems via fingerprinting, which might cause false alerts and draw attention away from real dangers.
4. **Complexity and Sustainability Needs:** Specifically, research honeypots are difficult to set up and keep up. They need sophisticated data-gathering and processing skills, which may be labour-and resource-intensive. Although useful for learning about the tactics of attackers and creating defined plans, the logistical burden can be high.
5. **Limited Effectiveness Against Automated Attacks:** Since automated attacks sometimes target several systems at once, honeypots may not be as effective against them. Honeypots may not prevent automated scripts and tools that quickly search and exploit vulnerabilities across a variety of targets, as their purpose is to draw in and trap attackers. As a result, their ability to stop or lessen these attacks is restricted. [8]

# HONEYPOT APPLICATIONS IN PRACTICAL SCENARIOS

## Honeypot Experiment Reveals What Hackers Want from IoT Devices.

One way to understand why attackers target devices is to look at a three-year-old honeypot experiment using simulated low-interaction IoT devices of different kinds and locations.

To be more precise, the purpose of the honeypot was to produce a sufficiently diverse ecosystem and to group the data that was produced in a way that might identify the objectives of enemies.

Small internet-connected gadgets like cameras, lighting, doorbells, smart TVs, motion sensors, speakers, thermostats, and many more are part of the rapidly expanding IoT (Internet of Things) sector.

Over 40 billion of these devices are expected to be online by 2025, offering network access points or processing power that might be utilised for illicit cryptocurrency mining or as part of DDoS swarms. [9]

### Setting the Stage

The NIST and University of South Florida researchers assembled three parts of the honeypot ecosystem: server farms, a screening mechanism, and the infrastructure for collecting and analyzing data.

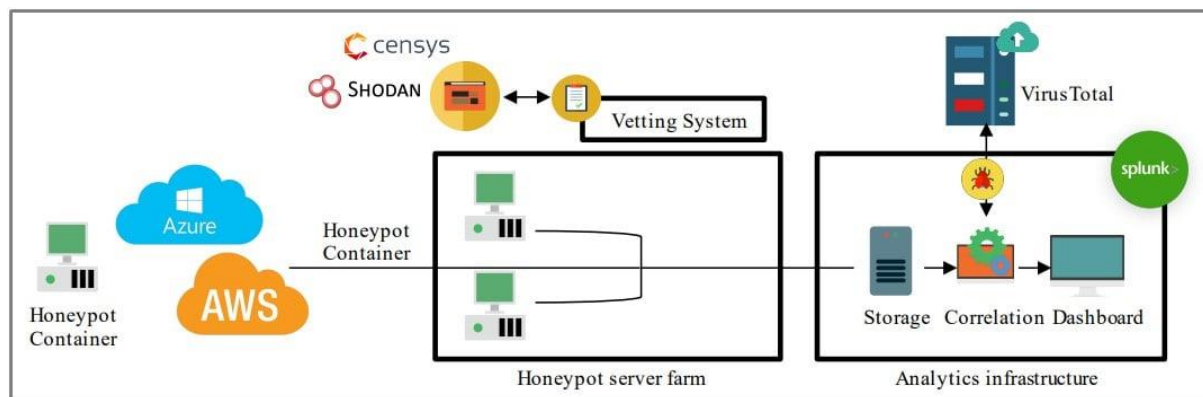
The researchers installed commercially available IoT honeypot emulators, such as Cowrie, Dionaea, KFSensor, and HoneyCamera, to establish a diversified ecosystem.

The researchers set up their instances on Censys and Shodan, two specialized search engines that locate services linked to the internet, to seem like actual devices.

These were the three primary categories of honeypots:

- i. **HoneyShell:** A Busybox Emulation
- ii. **HoneyWindowsBox:** A Windows emulator for Internet of gadgets
- iii. **HoneyCamera:** Emulating different IP cameras from D-Link, Hikvision, and other devices.





**Experiment layout.**

The fact that some honeypots were modified to react to attacker activity and attack techniques is a new aspect of this investigation.

The researchers modified the IoT defenses and configuration using the data they had gathered, and then they gathered further data reflecting the actor's reaction to these modifications. [9]

### The Findings

A staggering 22.6 million hits were made throughout the trial, the great majority of which were directed towards the HoneyShell honeypot.

Honeypot	Up Time	# of Hits
HoneyShell	12 months	17 343 412
HoneyWindowsBox	7 months	1 618 906
HoneyCamera	25 months	3 667 029

**Number of hits for each honeypot type.**

The fact that the actors' goals and methods of achieving them were probably similar explains why they all had comparable assault patterns.

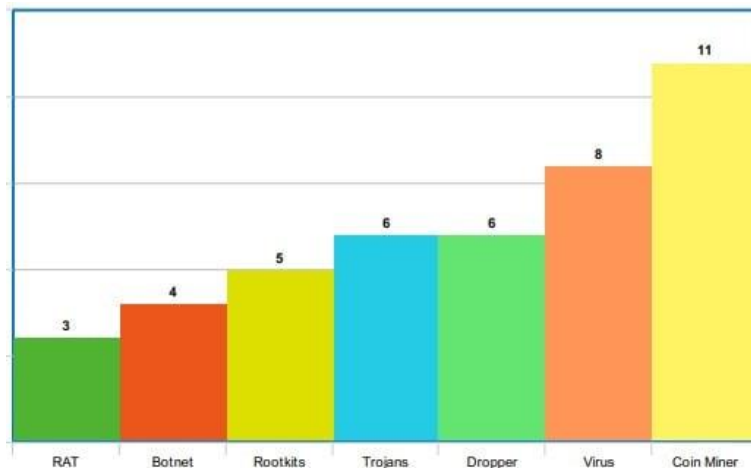
Most players, for instance, execute programs like "masscan" to look for open ports and "/etc/init.d/iptables stop" to turn off firewalls.

A lot of actors also use "free -m," "lspci grep VGA," and "cat /proc/cpuinfo" to get hardware details about the device they are targeting.

It's interesting to note that almost a million searches for the username-password combination "admin /1234" revealed a misuse of credentials in Internet of Things devices.

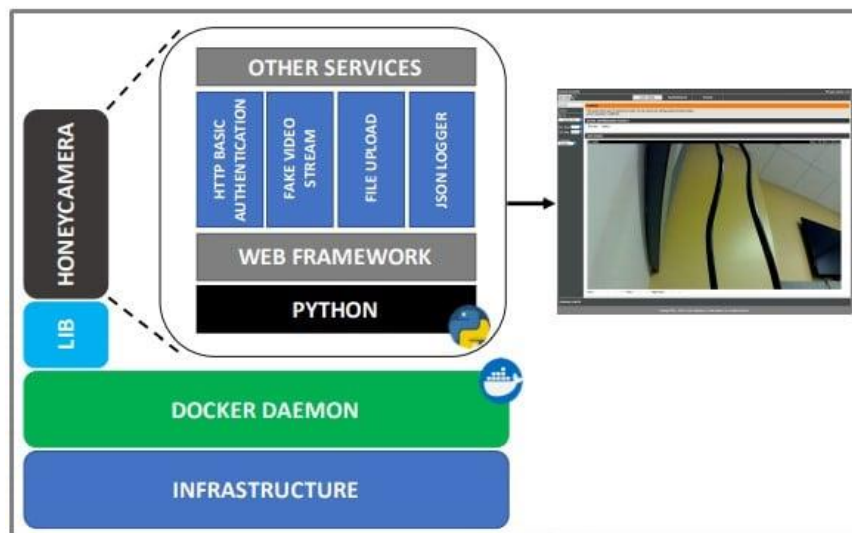
Regarding end objectives, the researchers discovered that the HoneyShell and HoneyCamera honeypots were primarily intended to attract DDoS attackers and were frequently infected with a coin miner or a Mirai derivative.

On the Windows honeypot, coin miner infections were among the more often seen threats, afterward viruses, droppers, and trojans. [9]



**Attack types targeting HoneyWindowsBox.**

When it came to the HoneyCamera, the investigators purposefully created a vulnerability to expose credentials, and they discovered that 29 perpetrators were manually making use of the weakness.



**Layout of HoneyCamera.**

"Just 314 112 (13 %) different connections have been observed with a minimum of one valid command being executed inside the honeypots," notes the study report.

"This data reveals that barely any of the cyberattacks took their next move, and the balance (87 %) mainly tried to find a suitable username/password match." [9]

### *How to Secure Your Devices*

To stop hackers from gaining control of your Internet of Things devices, adhere to these simple steps:

- Replace the default account with a strong, distinctive one (long).
- IoT devices should be connected to a different network and kept far from important resources.
- As soon as possible, be sure to install any available firmware upgrades or other security updates.
- Keep a close eye on your IoT devices and be alert for any indications of exploitation.

The most crucial thing to remember is to make sure a device is protected from unwanted remote access via a firewall or VPN if it doesn't require to be accessible to the Internet. [9]

## EMULATING HONEYPOTS: A PRACTICAL APPROACH

Choosing Cowrie as my honeypot for practical simulation, especially given its capability to log SSH connections and record sessions with detailed notes. Cowrie, an open-source project developed by Michel Oosterhof, is designed to capture brute force attacks and shell interactions, providing a comprehensive view of an attacker's activities. By offering a fake filesystem that mimics the structure of a Debian GNU/Linux system, Cowrie deceives attackers into believing they have successfully gained access to a real system. This deception allows for the collection of extensive information about an attacker's actions, which is invaluable for threat analysis and mitigation efforts. Furthermore, Cowrie's open-source nature enables customization to meet specific research or security needs, making it a versatile tool for enhancing network security through practical simulations. [10]

Below is the demonstration of the emulation:

I started my demonstration by looking for the victim machine's IP address. I accomplished this by running the `'ifconfig'` command.

```
File Actions Edit View Help
kali@kali: ~ x kali@kali: ~ x
L$ ifconfig
docker0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 172.17.0.1 netmask 255.255.0.0 broadcast 172.17.255.255
    inet6 fe80::42:26ff:fea7:c3f prefixlen 64 scopeid 0<20<link>
    ether 02:42:26:a7:0c:3f txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 5 bytes 526 (526.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

eth0: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.3.33 netmask 255.255.0.0 broadcast 192.168.255.255
    inet6 fe80::f729:ac91:b908:3ac4 prefixlen 64 scopeid 0<20<link>
    ether 08:00:27:1e:36:4a txqueuelen 1000 (Ethernet)
    RX packets 7634 bytes 748891 (731.3 KiB)
    RX errors 0 dropped 595 overruns 0 frame 0
    TX packets 318 bytes 136220 (133.0 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING> mtu 65536
    inet 127.0.0.1 netmask 255.0.0.0
    inet6 ::1 prefixlen 128 scopeid 0<10<host>
    loop txqueuelen 1000 (Local Loopback)
    RX packets 4 bytes 240 (240.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4 bytes 240 (240.0 B)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

vethd074020: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet6 fe80::344a:1fff:fecf:c6df prefixlen 64 scopeid 0<20<link>
    ether 36:4a:1f:cf:c6:df txqueuelen 0 (Ethernet)
    RX packets 0 bytes 0 (0.0 B)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 13 bytes 1182 (1.1 KiB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0
```

After running the 'ifconfig' command on the target PC, we were able to acquire the IP address specified as 192.168.3.33, as can be seen in the image.

Following that, we'll execute the command "docker run -p 2222:2222 cowrie/cowrie:latest" in the victim machine.

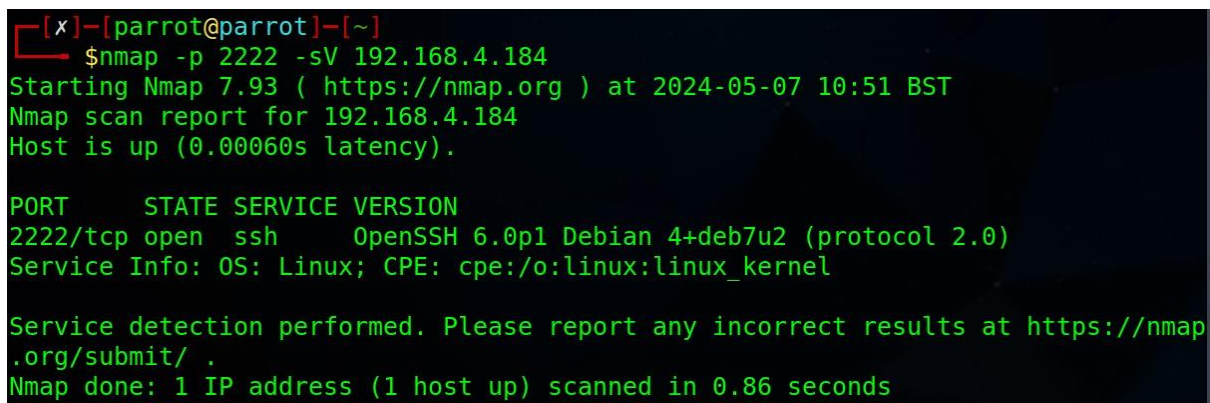
```
File Actions Edit View Help
root@kali: /home/kali x root@kali: /home/kali x kali@kali: ~ x
(root@kali)-[/home/kali]
# docker run -p 2222:2222 cowrie/cowrie:latest
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:106: CryptographyDeprecationWarning: Blowfish has been deprecated
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:110: CryptographyDeprecationWarning: CAST5 has been deprecated
  b"cast128-cbc": (algorithms.CAST5, 16, modes.CBC),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:115: CryptographyDeprecationWarning: Blowfish has been deprecated
  b"blowfish-ctr": (algorithms.Blowfish, 16, modes.CTR),
/cowrie/cowrie-env/lib/python3.11/site-packages/twisted/conch/ssh/transport.py:116: CryptographyDeprecationWarning: CAST5 has been deprecated
  b"cast128-ctr": (algorithms.CAST5, 16, modes.CTR),
2024-05-07T09:46:49+0000 [-] Python Version 3.11.2 (main, Mar 13 2023, 12:18:29) [GCC 12.2.0]
2024-05-07T09:46:49+0000 [-] Twisted Version 24.3.0
2024-05-07T09:46:49+0000 [-] Cowrie Version 2.5.0
2024-05-07T09:46:49+0000 [-] Loaded output engine: jsonlog
2024-05-07T09:46:49+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] twistd 24.3.0 (/cowrie/cowrie-env/bin/python3 3.11.2) starting up.
2024-05-07T09:46:49+0000 [twisted.scripts._twistd_unix.UnixAppLogger#info] reactor class: twisted.internet.epollreactor.EPollReactor.
2024-05-07T09:46:49+0000 [-] CowrieSSHFactory starting on 2222
```

This command initiates a Docker container utilizing the Cowrie honeypot image. Here's a breakdown:

- **docker run:** This command is utilized to start a Docker container.
- **-p 2222:2222:** This flag maps port 2222 on the host to port 2222 in the Docker container. Consequently, any traffic directed to port 2222 on the host will be directed to port 2222 in the Docker container.
- **cowrie/cowrie:latest:** This specifies the Docker image to employ for the container. In this instance, it refers to the most recent version of the Cowrie honeypot image, designed to simulate SSH and Telnet services to attract and analyze potential attackers.

This command initialises a Cowrie honeypot installation within a Docker container and opens port 2222 on the host machine for communication.

Next, we will run the following command on Parrot OS, the system used by our attacker:  
"nmap -p 2222 -sV 192.168.3.33."



```
[x]-[parrot@parrot]-[~]
$ nmap -p 2222 -sV 192.168.4.184
Starting Nmap 7.93 ( https://nmap.org ) at 2024-05-07 10:51 BST
Nmap scan report for 192.168.4.184
Host is up (0.00060s latency).

PORT      STATE SERVICE VERSION
2222/tcp  open  ssh      OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.86 seconds
```

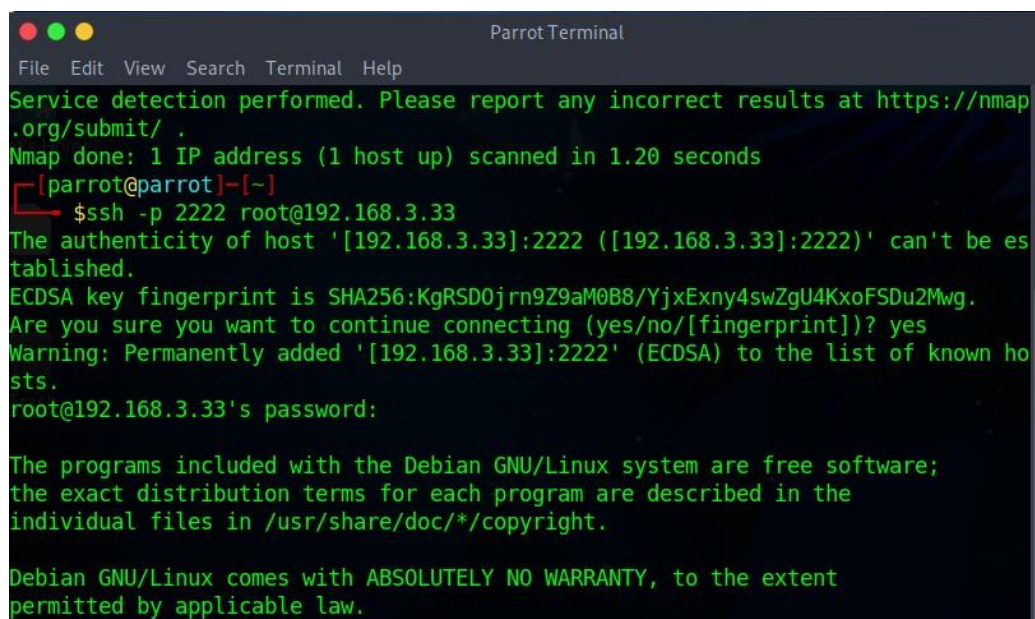
This command serves to execute a network scan using Nmap. Let's dissect its components:

- **nmap:** This robust command-line utility is tailored for network exploration and security auditing.
- **-p 2222:** This parameter is specifically designated for Nmap to scrutinize port 2222, concentrating the scan on this singular port rather than traversing all available ports.
- **-sV:** This directive is employed to activate version detection, thereby striving to discern the precise version of services operating on the designated target ports.

- 192.168.3.33: This numeric sequence denotes the IP address of the target system under investigation.

With this command, Nmap searches port 2222 on the target system (192.168.3.33) to identify the service that is using that port by determining its version.

In the final step, we execute the command “ssh -p 2222 root@192.168.3.33”.



```
Parrot Terminal
File Edit View Search Terminal Help
Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 1.20 seconds
[parrot@parrot]~$ ssh -p 2222 root@192.168.3.33
The authenticity of host '[192.168.3.33]:2222 ([192.168.3.33]:2222)' can't be established.
ECDSA key fingerprint is SHA256:KgRSD0jrn9Z9aM0B8/YjxExny4swZgU4KxoFSDu2Mwg.
Are you sure you want to continue connecting (yes/no/[fingerprint])? yes
Warning: Permanently added '[192.168.3.33]:2222' (ECDSA) to the list of known hosts.
root@192.168.3.33'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.
```

This SSH command uses port 2222 and its username of "root" to attempt to create a secure shell connection to a distant machine located at IP address 192.168.3.33.

- ssh: This command-line tool facilitates secure shell access.
- -p 2222: Here, the specified option designates the port to which the connection is directed; in this instance, port 2222.
- root@192.168.3.33: This amalgamation signifies the username "root" and the IP address of the remote server being accessed.

Therefore, the purpose of this operation is to establish a secure shell connection using port 2222 and the username "root" to the server at 192.168.3.33.



```
root@kali:/home/kali
2024-05-07T09:47:33+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.4.5] request_env: LC_MONETARY=en_GB.UTF-8
2024-05-07T09:47:33+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.4.5] request_env: LC_PAPER=en_GB.UTF-8
2024-05-07T09:47:33+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.4.5] request_env: LANG=en_GB.UTF-8
2024-05-07T09:47:33+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.4.5] request_env: LC_IDENTIFICATION=en_GB.UTF-8
2024-05-07T09:47:33+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.4.5] request_env: LC_TELEPHONE=en_GB.UTF-8
2024-05-07T09:47:33+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.4.5] request_env: LC_MEASUREMENT=en_GB.UTF-8
2024-05-07T09:47:33+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.4.5] request_env: LC_TIME=en_GB.UTF-8
2024-05-07T09:47:33+0000 [SSHChannel session (0) on SSHService b'ssh-connection' on HoneyPotSSHTransport,0,192.168.4.5] request_env: LC_NUMERIC=en_GB.UTF-8
2024-05-07T09:47:33+0000 [twisted.conch.ssh.session#info] Getting shell
2024-05-07T09:47:37+0000 [HoneyPotSSHTransport,0,192.168.4.5] CMD: ls
2024-05-07T09:47:37+0000 [HoneyPotSSHTransport,0,192.168.4.5] Command found: ls
2024-05-07T09:47:38+0000 [HoneyPotSSHTransport,0,192.168.4.5] CMD: cd
2024-05-07T09:47:38+0000 [HoneyPotSSHTransport,0,192.168.4.5] Command found: cd
2024-05-07T09:50:33+0000 [-] Timeout reached in HoneyPotSSHTransport
2024-05-07T09:50:33+0000 [HoneyPotSSHTransport,0,192.168.4.5] Closing TTY Log: var/lib/cowrie/tty/9aa70e055088faab8ca063ed3db3
```

Following the "getting shell" message on the victim machine, you would typically gain access to a shell prompt, indicating successful access to the remote server. At this point, you can interact with the victim machine's command line interface.

To proceed, you might want to explore the file system, gather information about the system configuration, or execute various commands to simulate the activities of a potential attacker. For example, you could use commands like `ls` to list files and directories, `cat` to view file contents, `ps` to list running processes, or `whoami` to identify the current user.

Additionally, you might consider running specific commands to simulate malicious activity or perform further reconnaissance on the victim system, depending on the objectives of your demonstration or experiment.

## CONCLUDING REMARKS

As a category of deception technology, honeypots are essential to contemporary cybersecurity tactics because they offer an extra line of protection against online attacks. By providing insights into the behaviors and techniques of attackers, they operate as a preventive step to lessen the harm caused after an attacker has gained access to a network. This data may be utilized to improve current defenses and is essential for improving network security. As honeypot technology has developed, linked networks intended to mimic real-world settings have given rise to honeynets and honeyfarms. These large-scale configurations allow attackers to be tracked across several platforms, giving a complete picture of their actions. This method helps detect threats and makes it easier to gather information that may be utilized to prosecute fraudsters. The way honeypots are being developed and incorporated into cybersecurity frameworks shows how important they are to the continuing fight against cyber threats

and how important it is to use them strategically to protect digital assets and operational integrity. [7]

## BIBLIOGRAPHIC DATA

- [1] <https://www.techtarget.com/searchsecurity/definition/honey-pot>
- [2] <https://www.crowdstrike.com/cybersecurity-101/honeypots-in-cybersecurity-explained/>
- [3] <https://usa.kaspersky.com/resource-center/threats/what-is-a-honeypot>
- [4] <https://www.rapid7.com/fundamentals/honeypots/>
- [5] <https://www.fortinet.com/resources/cyberglossary/what-is-honeypot>
- [6] <https://www.sentinelone.com/cybersecurity-101/honeypot-cyber-security/>
- [7] <https://www.xcitium.com/honeypots/>
- [8] <https://www.proofpoint.com/us/threat-reference/honeypot>
- [9] <https://www.bleepingcomputer.com/news/security/honeypot-experiment-reveals-what-hackers-want-from-iot-devices/>
- [10] <https://github.com/cowrie/cowrie>
- [11] <https://www.strongdm.com/blog/what-is-a-honeypot>
- [12] <https://www.crowdstrike.com/cybersecurity-101/honeypots-in-cybersecurity-explained/>
- [13] <https://www.linkedin.com/pulse/honeypots-cyber-security-subramaniam-sankaran>