Fooling Nmap and Metasploit: Cyber Deception on Production Systems

Henry Reed, Intern
Cyber Defense Solutions Department

October 10, 2020

Approved for public release. OTR 2020-00863.

whoami

- ShellCon attendee since 2017
- Undergraduate student at California State University, Northridge majoring in Computer Science, BS
  - *Began studying cybersecurity in freshman year*
  - *Expected to graduate in Spring 2021 (woo!)*
- Intern at The Aerospace Corporation
- Areas of focus:
  - *Defensive cyber operations*
    - Research and development for defensive solutions
  - *Hack the Box, Penetration Testing with Kali Linux, Virtual Hacking Labs*
- Certs
  - *CompTIA Security+*
  - *Red Hat Certified System Administrator*
  - *GIAC Penetration Tester*

Outline

This talk assumes little background knowledge. Interrupt me if you have questions!

- What is deception?
  - *We won’t be quoting Sun Tzu, breaking years of cyber-deception-talk tradition*
- Deception in offensive cyber operations (OCO)
- Deception in defensive cyber operations (DCO)
- Counter Reconnaissance Program (CORECPRO) introduction
- CORECPRO demos
- CORECPRO development findings
- OCO: identifying deception
- Future research

Deception in Offensive Cyber Operations (OCO)

- Most proliferated use of cyber deception is in OCO
  - *Deception has been historically used in OCO; all successful cyber attacks succeed through deceiving defensive systems, this is where deception in cyber has been born*
  - *PLA attacked the Landsat-7, a USG satellite, on October 20, 2007. This attack was only discovered in July 2008.[1]*
- Still a new concept in DCO
- OCO Deception Goals
  - *Bypass intrusion prevention, anti-malware or other automated defensive software*
    - Includes heuristic and signature detection
  - *Remain undetected by DCO personnel*
  - *If client-side attack: remain undetected by end user*
- Examples
  - *Creating malicious payloads tailored to deceive defensive systems used by target*
    - Anything from ghost writing and uncommon encoding methods to Veil and custom malware
  - *Stealing credentials to act as a specific user; depending on privileges, some defenses may be bypassed*
  - *Purposely throwing alerts in one system to divert attention of the defensive teams*
    - Any action to negatively impact how a defensive team is able to allocate resources to where a real attack is occurring

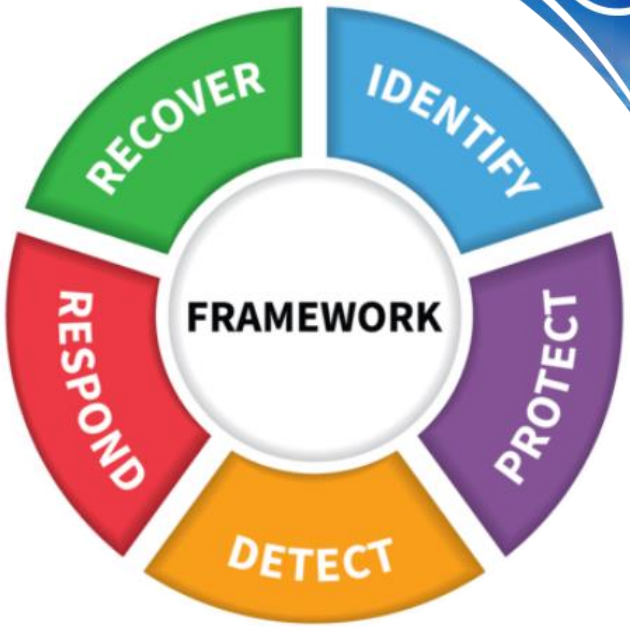
[1] P. 216. "2011 Report to Congress". U.S.-China Economic and Security Review Commission

Deception in Defensive Cyber Operations (DCO)

- The dissemination of false data to adversaries to…
  - *Produce true positives*
  - *Gain early notification of an attack*
  - *Waste the attacker’s time*
  - *Gather threat intelligence*
- Occurs during enemy:
  - *External reconnaissance (includes scanning)*
  - *Internal reconnaissance (insiders, pivoting)*
  - *Exploitation*

Deception in Defensive Cyber Operations

NIST Cybersecurity Framework 1.1



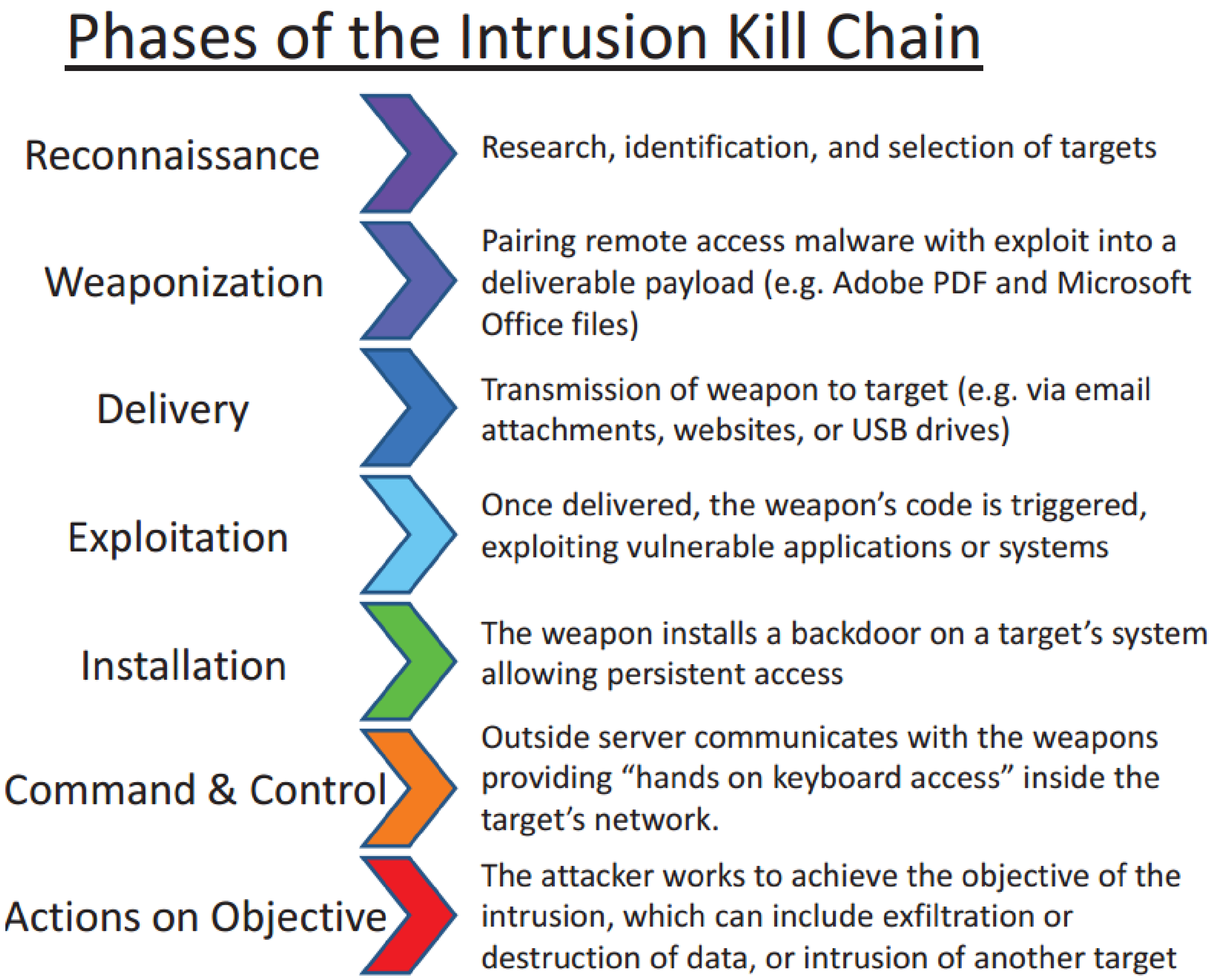
- Identify: Prerequisite for deception technology
  - *Identify critical infrastructure wherein deception technology can be placed*
- Detection: Primary goal of deception
  - *Main issue in detection using traditional methods is lowering false positive rate*
  - *Deception alerts are almost always caused by attacker activity*
  - *Deception alerts should be prioritized over all other alerts*
- Respond: Secondary goal of deception
  - *Attackers spending time on enumerating, exploiting, and exfiltrating information from deception technologies waste their time*
  - *This buys more time to respond to cyber attacks*

[2] P. 7 “Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1” National Institute of Standards and Technology

Deception in Defensive Cyber Operations

Lockheed Martin Corporation's Cyber Kill Chain

- Reconnaissance
  - *Deception provides false data to adversaries during their reconnaissance phase*
  - *Deception alerts defensive operators of adversarial reconnaissance*
- Exploitation
  - *Deception provides false data to exploitation frameworks used by attackers*
  - *Deception alerts cyber operations of exploitation*
- Installation
  - *Malware downloaded in the fake shell will be captured*
- Command & Control
  - *C&C will be limited to the fake instance within the deception software*
- Actions on Objective
  - *Actions will be limited to the fake instance, monitoring adversarial tactics, techniques and procedures*



Phases of the Intrusion Kill Chain

Reconnaissance: Research, identification, and selection of targets

Weaponization: Pairing remote access malware with exploit into a deliverable payload (e.g. Adobe PDF and Microsoft Office files)

Delivery: Transmission of weapon to target (e.g. via email attachments, websites, or USB drives)

Exploitation: Once delivered, the weapon's code is triggered, exploiting vulnerable applications or systems

Installation: The weapon installs a backdoor on a target's system allowing persistent access

Command & Control: Outside server communicates with the weapons providing "hands on keyboard access" inside the target's network.

Actions on Objective: The attacker works to achieve the objective of the intrusion, which can include exfiltration or destruction of data, or intrusion of another target

[3] P. 7 "A 'Kill Chain' Analysis of the 2013 Target Data Breach". US Senate Committee on Commerce, Science, and Transportation

Deception in Defensive Cyber Operations

Background

- Why did honeypots fizzle out?
- Honeypots were marketed as individual VMs to host in the network
  - *Attackers usually missed these since there were no breadcrumbs to lead to them*
    - Why go through the effort (and risk) of attacking a machine on the network if there is no evidence something interesting resides on it?
  - *Setting up a full honeypot with alerting only to have it ignored is high effort and low yield*
- This bad rep is still prevalent; most cyber engineers do not deploy honeypots [4]
- Cyber deception is very different today than it was ten years ago
- When implemented correctly, modern deception solutions provide true positive alerts that must be investigated as they are generated as a result of malicious activity
  - *These deception technologies can and should be used on production systems*
  - *This eliminates the "no one will look at the honeypot" issue*

[4] Dominguez, Andrea. "The State of Honeypots: Understanding the Use of Honey Technologies Today". SANS Institute

Deception in Defensive Cyber Operations

Why isn't DCO deception more popular?

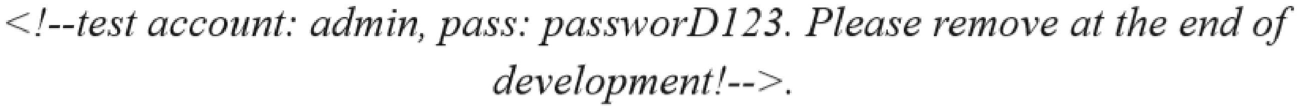
- Fear of emerging technology: "No one uses this anyway so why should we"
  - *Just because something is new doesn't mean it's bad; Snort, Splunk, and others were new at some point, too*
  - *Try out FOSS software, get demos by COTS vendors and see for yourself*
- "Once you max out your defenses, THEN you should consider deception" [5]
  - *It's easier and quicker to host a handful of FOSS deception tools than to…*
  - *Install and configure IDPS and SIEM*
    - Build a SOC
    - Hire SOC analysts
    - Pay for software, hardware, and continuous learning for analysts
    - Hire an internal red team
    - Pay for their stuff, too
  - *Deception will get you better true positives than an IDPS*
  - *Deception isn't a replacement for IDPS, SOC, etc., but it will provide alerting before you have everything else set up*
  - *Will still be useful even after you have your fully-functioning 24/7 SOC of 300 SANS-graduate analysts*
  - *Just have an intern do it*

[5] Strand, John. "Webcast: Getting Started in Cyber Deception". Black Hills Information Security.

Deception in Defensive Cyber Operations

Some existing FOSS solutions

- HoneyFiles in classified environments
  - *E.g.: Given a SECRET machine, place a fake TOP SECRET document in a visible area and see who does and does not make a report*
  - *This is a security infraction and requires reporting*
  - *Those who do not report are either insider threats or fail to comply with simple policies for handling classified information—both of which are severe and require an investigation.*
- HoneyCreds in comments or other difficult-to-access areas
  - *Monitoring attempted access to these fake accounts provides a true positive alert on adversarial activity*



<!--test account: admin, pass: passworD123. Please remove at the end of development!-->.

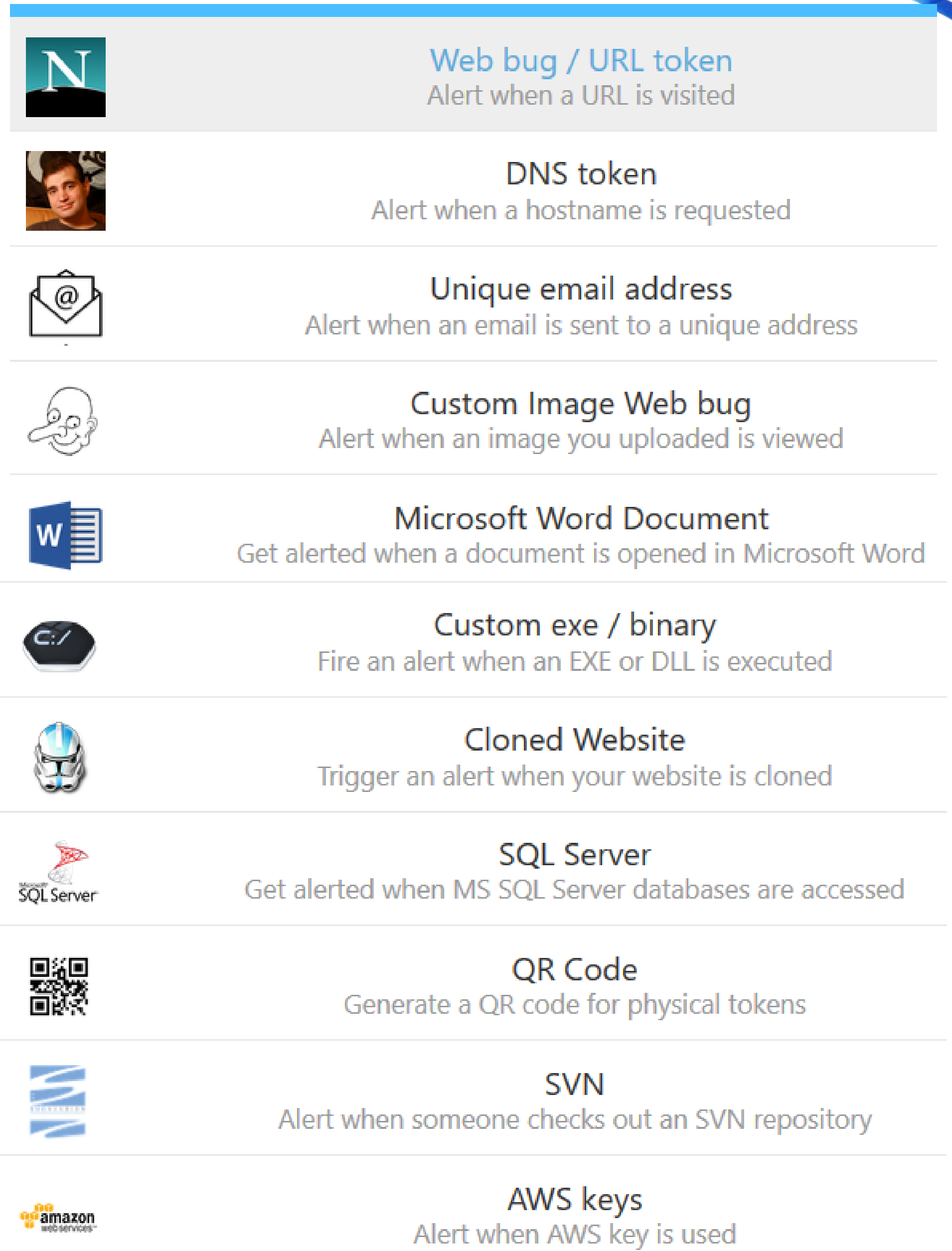
- Fake entries in robots.txt
  - *Monitor access logs for specific fake directories*
  - *Any access to those indicates that someone has actively gone through the robots.txt file and tried to access a forbidden directory—a true positive*

[6]: Virilis, Nikos, et al. "Changing the game: The art of deceiving sophisticated attackers" NATO

Deception in Defensive Cyber Operations

Some existing FOSS solutions

- CanaryTokens by Thinkst
  - *Great video on this by John Strand (former SANS instructor, BHIS CEO)*
  - *Fake AWS keys*
  - *Bugging software/DLLs*
  - *Bugging webpages to combat spearphishing campaigns before they begin*
  - *Bugging word docs*
  - *Bugging directories in robots.txt*



Web bug / URL token
Alert when a URL is visited

DNS token
Alert when a hostname is requested

Unique email address
Alert when an email is sent to a unique address

Custom Image Web bug
Alert when an image you uploaded is viewed

Microsoft Word Document
Get alerted when a document is opened in Microsoft Word

Custom exe / binary
Fire an alert when an EXE or DLL is executed

Cloned Website
Trigger an alert when your website is cloned

SQL Server
Get alerted when MS SQL Server databases are accessed

QR Code
Generate a QR code for physical tokens

SVN
Alert when someone checks out an SVN repository

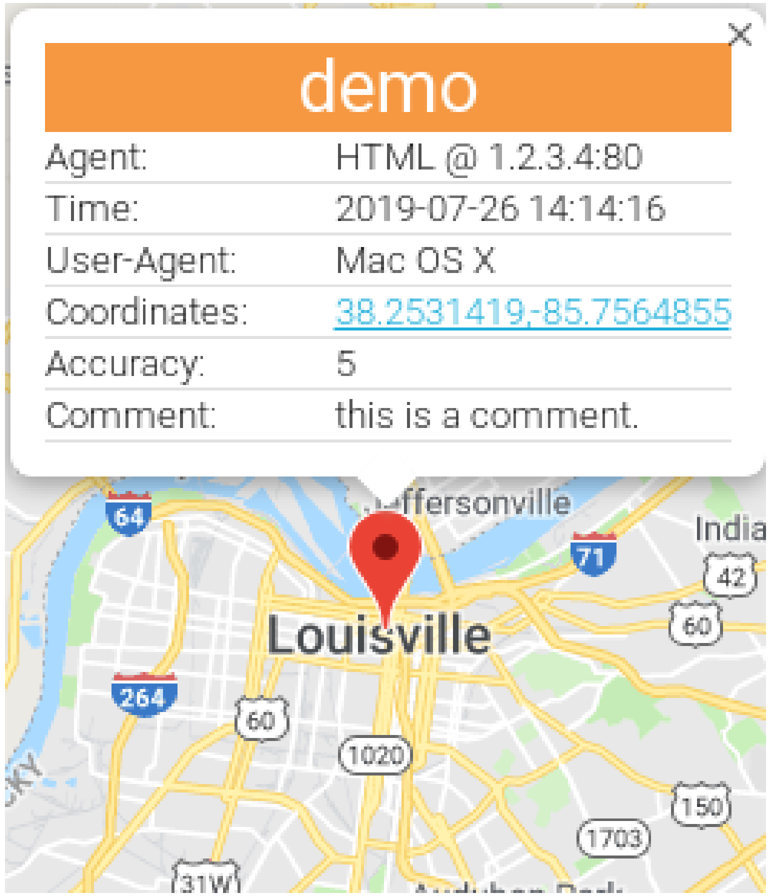
AWS keys
Alert when AWS key is used

[2] Strand, John. "Webcast: Getting Started in Cyber Deception". Black Hills Information Security.

Deception in Defensive Cyber Operations

Some existing FOSS solutions

- HoneyBadger by Black Hills Information Security
  - *Generates macros which scan for nearby Wi-Fi access points*
  - *Once adversary opens a document with those macros, the access point data is sent to you*
  - *Can use Google API to track exact location of the adversary*
  - *Accurate location within a few meters*



demo

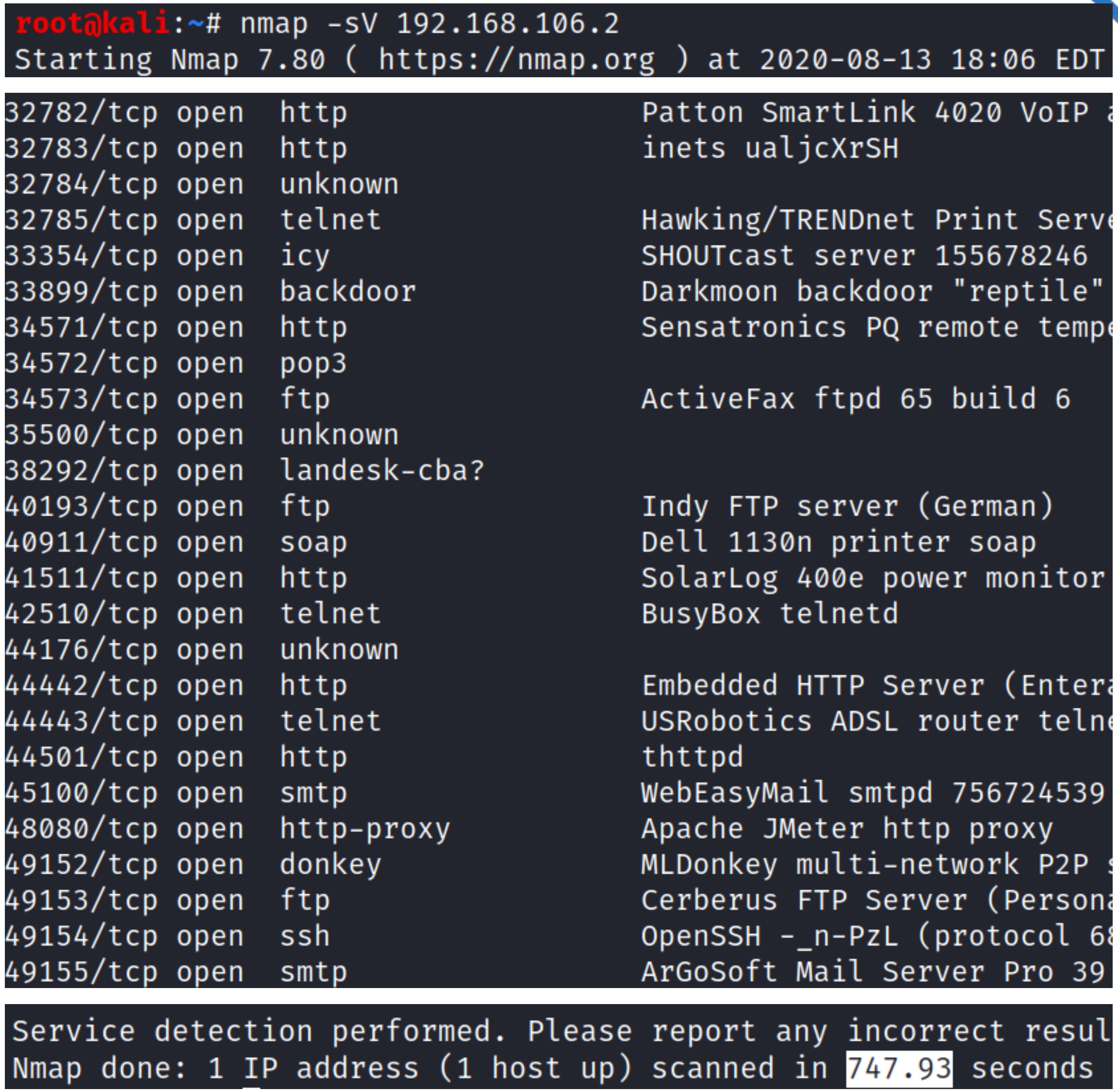
Agent:	HTML @ 1.2.3.4:80
Time:	2019-07-26 14:14:16
User-Agent:	Mac OS X
Coordinates:	38.2531419,-85.7564855
Accuracy:	5
Comment:	this is a comment.

[7] Strand, John. "Getting Started with Tracking Hackers with HoneyBadger". Black Hills Information Security.

Deception in Defensive Cyber Operations

Some existing FOSS solutions

- Portspoof
  - *Provides false banners to Nmap version scans on every port on a machine*
  - *Renders stealth scans useless, as every port is shown as “open”*
  - *Renders version scans useless, as the adversary will need to spend a lot of time distinguishing between real and fake services*
  - *Slow down version scan to an extreme extent*
    - 12.5 minutes per host for 1000 most common ports
    - Not including any latency for over-the-internet scans
  - *Runs rootless*
  - *No native logging capability*



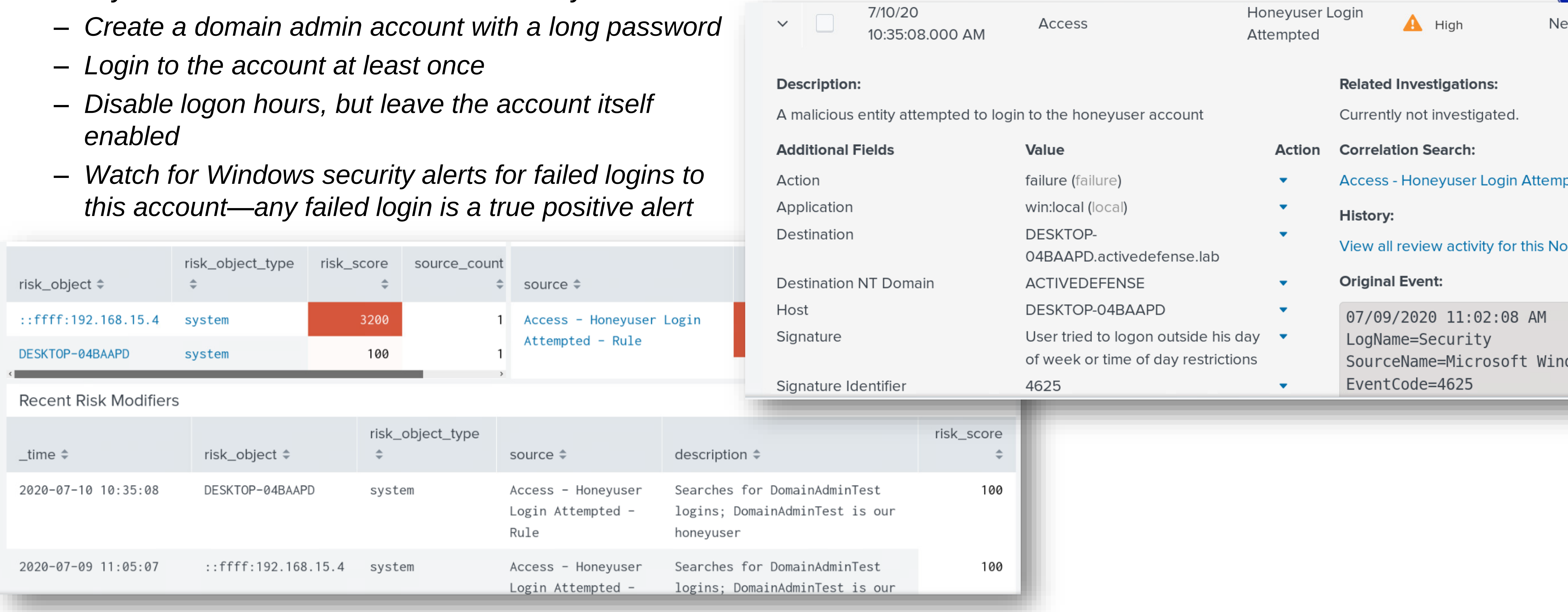
```
root@kali:~# nmap -sV 192.168.106.2
Starting Nmap 7.80 ( https://nmap.org ) at 2020-08-13 18:06 EDT

32782/tcp open  http                    Patton SmartLink 4020 VoIP a
32783/tcp open  http                    inets ualjcXrSH
32784/tcp open  unknown
32785/tcp open  telnet                  Hawking/TRENDnet Print Serve
33354/tcp open  icy                     SHOUTcast server 155678246
33899/tcp open  backdoor                Darkmoon backdoor "reptile"
34571/tcp open  http                    Sensatronics PQ remote tempe
34572/tcp open  pop3
34573/tcp open  ftp                     ActiveFax ftpd 65 build 6
35500/tcp open  unknown
38292/tcp open  landesk-cba?
40193/tcp open  ftp                     Indy FTP server (German)
40911/tcp open  soap                    Dell 1130n printer soap
41511/tcp open  http                    SolarLog 400e power monitor
42510/tcp open  telnet                  BusyBox telnetd
44176/tcp open  unknown
44442/tcp open  http                    Embedded HTTP Server (Entera
44443/tcp open  telnet                  USRobotics ADSL router telne
44501/tcp open  http                    thttpd
45100/tcp open  smtp                    WebEasyMail smtpd 756724539
48080/tcp open  http-proxy              Apache JMeter http proxy
49152/tcp open  donkey                  MLDonkey multi-network P2P s
49153/tcp open  ftp                     Cerberus FTP Server (Persona
49154/tcp open  ssh                     OpenSSH -_n-PzL (protocol 68
49155/tcp open  smtp                    ArGoSoft Mail Server Pro 39

Service detection performed. Please report any incorrect resul
Nmap done: 1 IP address (1 host up) scanned in 747.93 seconds
```

[8] Strand, John. “Active Defense & Cyber Deception - Part 3”. Black Hills Information Security.

Deception in Defensive Cyber Operations

Some existing FOSS solutions

- HoneyUsers in Windows Active Directory
  - *Create a domain admin account with a long password*
  - *Login to the account at least once*
  - *Disable logon hours, but leave the account itself enabled*
  - *Watch for Windows security alerts for failed logins to this account—any failed login is a true positive alert*



risk_object	risk_object_type	risk_score	source_count	source
::ffff:192.168.15.4	system	3200	1	Access - Honeyuser Login Attempted - Rule
DESKTOP-04BAAPD	system	100	1	

Recent Risk Modifiers

_time	risk_object	risk_object_type	source	description	risk_score
2020-07-10 10:35:08	DESKTOP-04BAAPD	system	Access - Honeyuser Login Attempted - Rule	Searches for DomainAdminTest logins; DomainAdminTest is our honeyuser	100
2020-07-09 11:05:07	::ffff:192.168.15.4	system	Access - Honeyuser Login Attempted -	Searches for DomainAdminTest logins; DomainAdminTest is our	100

7/10/20 10:35:08.000 AM | Access | Honeyuser Login Attempted | ⚠ High | Ne

Description:

A malicious entity attempted to login to the honeyuser account

Additional Fields	Value	Action
Action	failure (failure)	▾
Application	win:local (local)	▾
Destination	DESKTOP-04BAAPD.activedefense.lab	▾
Destination NT Domain	ACTIVEDEFENSE	▾
Host	DESKTOP-04BAAPD	▾
Signature	User tried to logon outside his day of week or time of day restrictions	▾
Signature Identifier	4625	▾

Related Investigations:

Currently not investigated.

Correlation Search:

Access - Honeyuser Login Attemp

History:

View all review activity for this No

Original Event:

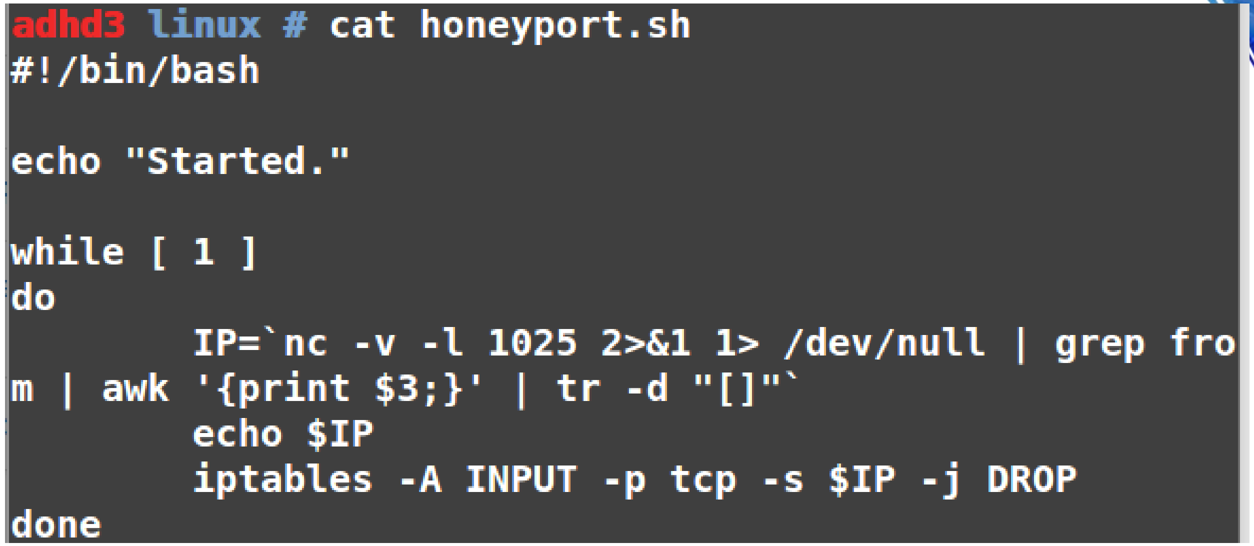
```
07/09/2020 11:02:08 AM
LogName=Security
SourceName=Microsoft Win
EventCode=4625
```

[9] Strand, John. 11:04 "Active Defense & Cyber Deception - Part 2". Black Hills Information Security.

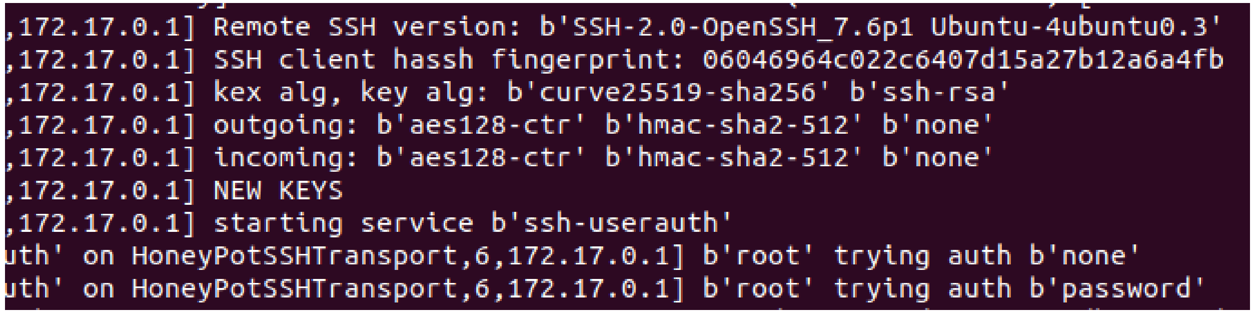
Deception in Defensive Cyber Operations

Some existing FOSS solutions

- HoneyPort scripts
  - *Upon a full handshake to a port, generate an alert and block that IP*
  - *Bash or PowerShell, usually under 50 lines of code*
  - *Benefits: Block external IPs that are trying to connect to abnormal ports. Block internal IPs, helps in case any machine has been taken over by an adversary*
  - *IP spoofing is useless against this because of the full connection requirement*
- Kippo & Cowrie
  - *Fake SSH services providing low-interactivity shells*
  - *Easy to detect once you're in the shell, though if you get shell you're already caught*
  - *Cowrie is able to forward data to a real virtual machine, which is more difficult to detect*
- All of these and more are available in ADHD; see cited video for more information



```
adhd3 linux # cat honeyport.sh
#!/bin/bash

echo "Started."

while [ 1 ]
do
        IP=`nc -v -l 1025 2>&1 1> /dev/null | grep fro
m | awk '{print $3;}' | tr -d "[]"`
        echo $IP
        iptables -A INPUT -p tcp -s $IP -j DROP
done
```



```
,172.17.0.1] Remote SSH version: b'SSH-2.0-OpenSSH_7.6p1 Ubuntu-4ubuntu0.3'
,172.17.0.1] SSH client hassh fingerprint: 06046964c022c6407d15a27b12a6a4fb
,172.17.0.1] kex alg, key alg: b'curve25519-sha256' b'ssh-rsa'
,172.17.0.1] outgoing: b'aes128-ctr' b'hmac-sha2-512' b'none'
,172.17.0.1] incoming: b'aes128-ctr' b'hmac-sha2-512' b'none'
,172.17.0.1] NEW KEYS
,172.17.0.1] starting service b'ssh-userauth'
uth' on HoneyPotSSHTransport,6,172.17.0.1] b'root' trying auth b'none'
uth' on HoneyPotSSHTransport,6,172.17.0.1] b'root' trying auth b'password'
```

[8] Strand, John. "Active Defense & Cyber Deception - Part 3". Black Hills Information Security.

Deception in Defensive Cyber Operations

Some existing COTS solutions

- Popular COTS Vendors:
  - *Acalvio ShadowPlex*
  - *Attivo Networks ThreatDefend Deception & Response Platform*
  - *Cymmetria MazeRunner*
  - *Illusive Networks Attack Detection System and Attack Intelligence System*
  - *Smokescreen IllusionBLACK*
  - *TrapX Security DeceptionGrid*
- Why?
  - *Fortune 500 entities may want to be able to use licensing and contractual obligations to shift some of the blame on the vendor; using FOSS tools puts all the blame on the fortune 500 entity itself*
  - *Some entities may not have decisionmakers who are OK with FOSS solutions*
  - *Initial configuration is usually low; upkeep is vendor's responsibility, including new features*
- You can build a majority of the capabilities through FOSS deception
- Government sector: Should be fine to rely on FOSS, instead
  - *USG generally tries to use FOSS wherever it is financially smart to do—this is definitely the case with cyber deception*

Deception in Defensive Cyber Operations

FOSS Tools Summarized

- If you were to implement all the tools mentioned, you would be able to:
  - *Detect internal and external port scans and immediately react to them*
  - *Gain attributions on your adversaries*
  - *Prevent phishing campaigns before they begin*
  - *Detect password sprays*
  - *Gather threat intelligence on the attacker before they realize they're in a fake shell*
- Missing capability: better threat intelligence
  - *Cowrie and Kippo are easy to detect once you're in the shell, unless you're running Cowrie as a sniffer between the attacker and a virtual machine*

Counter Reconnaissance Program

***CO**unter **REC**onnaissance **PRO**gram*

ACTA DEOS NUMQUAM MORTALIA FALLUNT

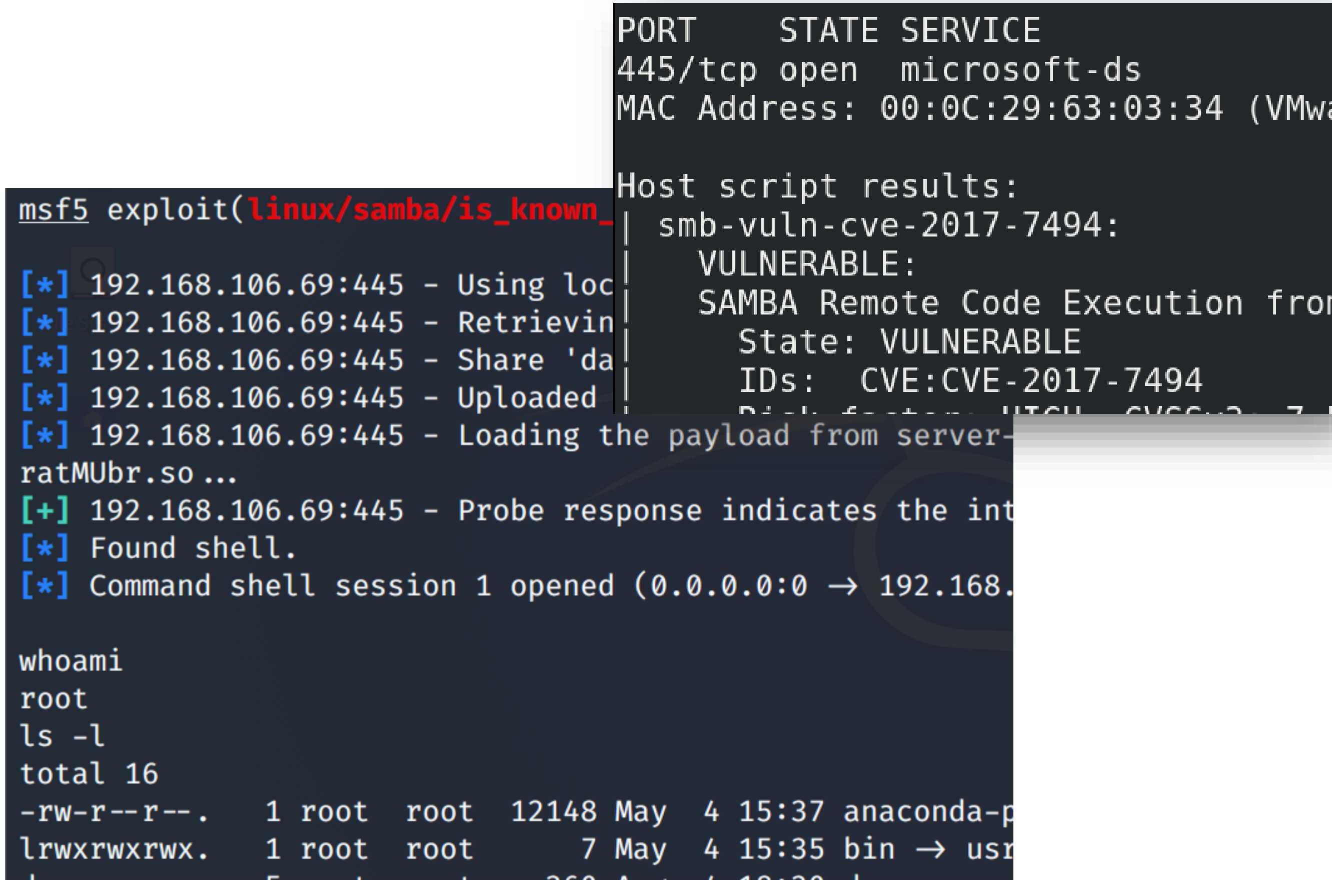
CORECPRO

- Purpose:
  - *Gain early notification of an attack*
  - *Waste the attacker’s time*
  - *Gather threat intelligence*
  - *Produce true positives*
- Design goals:
  - *Emulates vulnerable services, deceiving reconnaissance*
  - *Responds realistically to vulnerability scans*
  - *Responses are not distinguishable from genuine service traffic even upon cross-referencing with legitimate service responses in a lab*
  - *Does not interfere with services running on the production system*
  - *Reasonably secure (can be run by unprivileged user)*
  - *Logs readable by Splunk*
- Published for free on GitHub under the MIT license

Counter Reconnaissance Program

Current capabilities

- Samba 4.5.9 emulation, high interaction
  - *Emulates CVE-2017-7494, AKA SambaCry or EternalRed*
    - Remote Code Execution exploit; Metasploit module gives root shell
  - *Fools Nmap vulnerability scan, making it recognize CORECPRO as vulnerable Samba service*
  - *Fools Metasploit, making it think it has shell*
- libSSH, low interaction
  - *Emulates CVE-2018-10933*
    - Allows bypassing authentication for any user
  - *Responds to Nmap version scan (only scan available)*
  - *Notifies of Metasploit exploit attempt*



```
PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:63:03:34 (VMwa

Host script results:
| smb-vuln-cve-2017-7494:
|   VULNERABLE:
|   SAMBA Remote Code Execution fron
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-7494
```

```
msf5 exploit(linux/samba/is_known_

[*] 192.168.106.69:445 - Using loc
[*] 192.168.106.69:445 - Retrievin
[*] 192.168.106.69:445 - Share 'da
[*] 192.168.106.69:445 - Uploaded
[*] 192.168.106.69:445 - Loading the payload from server-
ratMUbr.so ...
[+] 192.168.106.69:445 - Probe response indicates the int
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 → 192.168.

whoami
root
ls -l
total 16
-rw-r--r--.   1 root  root  12148 May  4 15:37 anaconda-p
lrwxrwxrwx.   1 root  root      7 May  4 15:35 bin → usr
```

Counter Reconnaissance Program Demo

CORECPRO Development Findings

High fidelity—Interactive shell, vuln scan deception

- Development time for high-fidelity honeypot
  - *Heavily depends on the chosen protocol and developers' experience with protocol*
  - *The SMB protocol is complex and not well documented*
    - Getting SMB to work took ~3 months of full-time hours
      - *No prior experience with SMB nor networking programming*
    - Getting Docker to work took 2-3 days
- Development requirements for high-fidelity honeypot
  - *Must realistically emulate service.*
    - E.g., if the service sends timestamps, your timestamps should not be copy-pasted time stamps from the time you did your packet capture. Same applies to randomly-generated sequences, any unique user-definable values, etc.
    - Does the service have bugs? Make sure you include the same bugs in your deception technology
- What's the ROI?
  - *Instead of relying on threat intel feeds, which gather intel from breaches of other organizations, you get threat intel from adversaries attacking your organization*
    - Gain understanding of attackers' motivations; e.g., ransomware, corporate espionage
    - If they are looking to exfiltrate data, what kind of data are they looking for? If it's something specific, who knew it existed? Anyone on the team using personal resources (open Google Drives, home servers) to do their work?
      - *Consider putting beacons in your documents to see if you get pings back from non-corporate entities*

CORECPRO Development Findings

High fidelity—Interactive shell, vuln scan deception

- For whom would it be useful to build cyber deception tools?
- Fortune 500
  - *Usually scared of anything homegrown due to audit/law/corporate policy compliance*
    - Questions you might hear: If this fails to detect an adversary, how do we explain it in an audit? But is it PCI-DSS compliant? If it’s not required by [law] why do we need to spend the time or money to do it?
    - The “if this fails…” question: There is no guarantee that an adversary will fall for a trap. This lack of guarantee extends to COTS deception tools.
    - The risk remains that an advanced adversary may be able to break out of Docker
      - *Does your threat model include adversaries with the time and budget to develop these capabilities?*
      - *COTS software will have legal paperwork where you can shift some of the blame on the security vendor; homegrown software carries all the blame with it*
- FFRDCs, security vendors, and other research groups
  - *Development of production-system cyber deception is crucial*
    - Some COTS solutions are never touched because there is no reason to

OCO: Identifying and Evading Deception

- In order for an adversary to evade deception, they must gather open source intelligence first
- Search for…
  - *Lists of employees; LinkedIn is perfect for this*
  - *Resumes of aforementioned employees; usually can be found on LinkedIn profiles. If not there, search for any open directories, personal websites, blogs, leaks, etc.*
    - Use resume-specific Google Dorks, for example…
      - *Firstname Lastname Resume filetype: doc*
      - *Firstname Lastname site:docs.google.com*
      - *Firstname Lastname site:drive.google.com*
  - *Job postings from the organization*
  - *Past or removed job postings; Wayback Machine and Google Cached Pages are good for this*
  - *Employee Twitter (and other social media) accounts*
- Look for keywords…
  - *Specific deception vendor names: Attivo, Cymmetria, etc.*
  - *Words like “honeypot”, “deception”, etc.*
- Search for recordings of talks on deception (like this one!) and see who attended them

OCO: Identifying and Evading Deception

Background

Honeypot VM Evasion

- Attackers pivot based on evidence
- Many commercial tools create a large amount of fake honeypot VMs, hoping attackers would interact with them; however, attackers can simply miss these if there isn’t any evidence those machines exist
- The assumption from vendors is that attackers will scan an internal network. This is noisy and generally isn’t done by advanced adversaries.
- Given the above, it is safe to assume that technologies which rely on generating honeypot VMs that have no trail leading to them from genuine production machines will simply be missed or ignored by attackers

Honeyport Evasion

- Honeyport principle: Nothing should interact with a fake port; if a machine connects to a fake port, that machine has been compromised
- Adversary can’t detect a honeyport without interacting with it; interaction triggers an alert
- Deception technology is not commonplace today and generally attackers are not expecting it
- Given the above, it is safe to assume most attackers will stumble upon a honeyport and trigger an alert

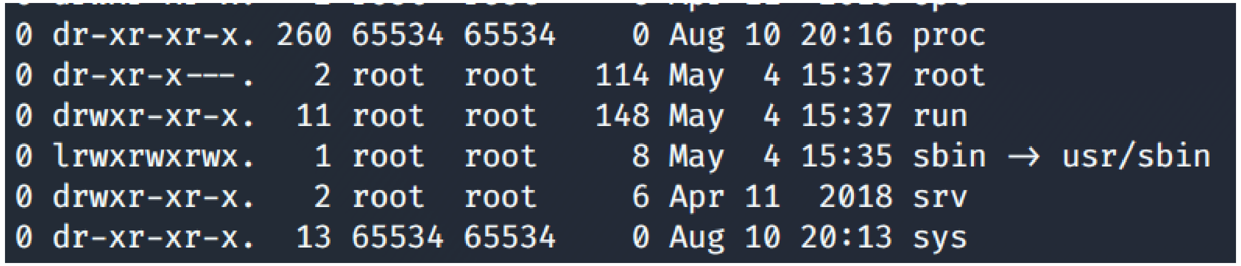
OCO: Identifying and Evading Deception

- You will need to interact with services and objects when attacking a network
  - *If you find domain admin credentials, you will need to try them*
  - *If you find an open SMB share, you’ll need to interact with it*
- Goal? Hide your TTPs while you’re attacking and move quickly
  - *Verify you are not in a Docker container, or another fake environment, before doing any data exfiltration*
  - *If the defensive operators see what you are looking for, they may be able to identify who you are*
  - *E.g.: doing data exfiltration? How did you know this data existed in the first place?*
    - If through an insider, you may be painting a target on them if you’re noisy with your searches.
    - Open directory/S3 bucket/Google Drive/etc.: Be sure you got everything from there already, it could be taken down after your attack is done; if you ever accessed this from an attributable IP, you could get caught

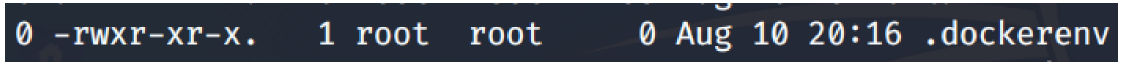
OCO: Identifying and Evading Deception

Identifying Docker environment

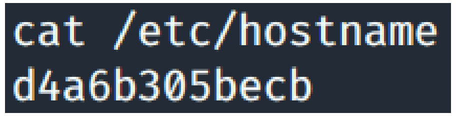
- Some telltale signs of a Docker environment:
  - */sys or /proc owned by 65534:65534*
    - Only indicative of a rootless Docker shell
    - If Docker is ran as root, these files are owned by root
  - */.dockerenv exists*
  - */etc/hostname contains a Docker container ID*
  - */etc/hosts contains a Docker container ID*
  - *Some scripts can automate Docker detection:*
    - linPEAS: https://github.com/carlospolop/privilege-escalation-awesome-scripts-suite
    - Linux Smart Enumeration: https://github.com/diego-treitos/linux-smart-enumeration
  - *Echoing to any "files" in /proc/sys/kernel gets you permission denied (rootless Docker) or "Read-only filesystem" errors (Docker ran as root)*



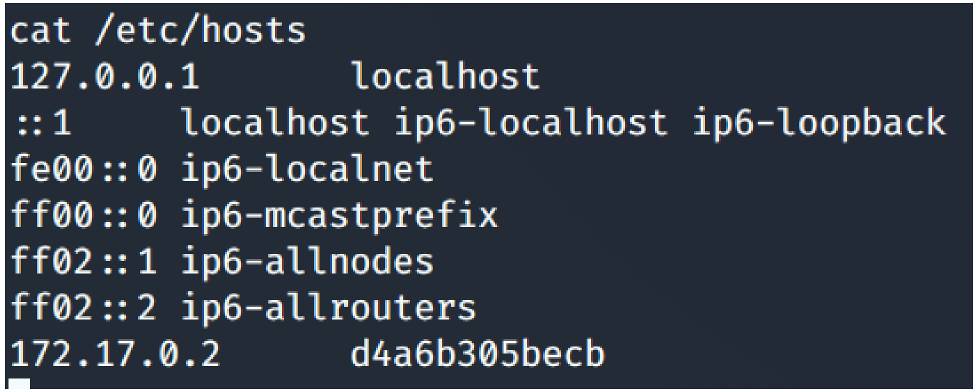
```
0 dr-xr-xr-x. 260 65534 65534    0 Aug 10 20:16 proc
0 dr-xr-x---.   2 root  root   114 May  4 15:37 root
0 drwxr-xr-x.  11 root  root   148 May  4 15:37 run
0 lrwxrwxrwx.   1 root  root     8 May  4 15:35 sbin -> usr/sbin
0 drwxr-xr-x.   2 root  root     6 Apr 11  2018 srv
0 dr-xr-xr-x.  13 65534 65534    0 Aug 10 20:13 sys
```



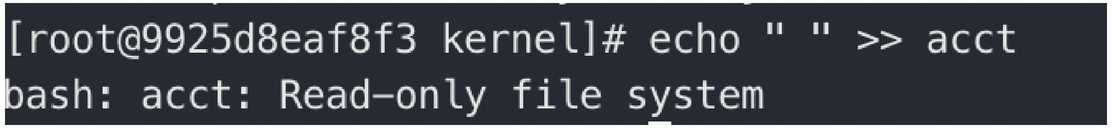
```
0 -rwxr-xr-x.   1 root  root     0 Aug 10 20:16 .dockerenv
```



```
cat /etc/hostname
d4a6b305becb
```



```
cat /etc/hosts
127.0.0.1       localhost
::1     localhost ip6-localhost ip6-loopback
fe00::0 ip6-localnet
ff00::0 ip6-mcastprefix
ff02::1 ip6-allnodes
ff02::2 ip6-allrouters
172.17.0.2      d4a6b305becb
```

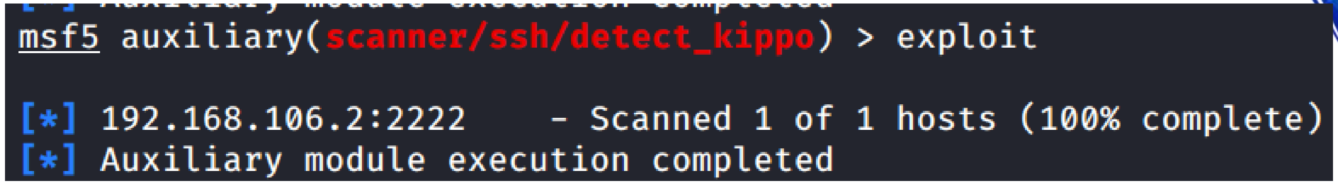


```
[root@9925d8eaf8f3 kernel]# echo " " >> acct
bash: acct: Read-only file system
```

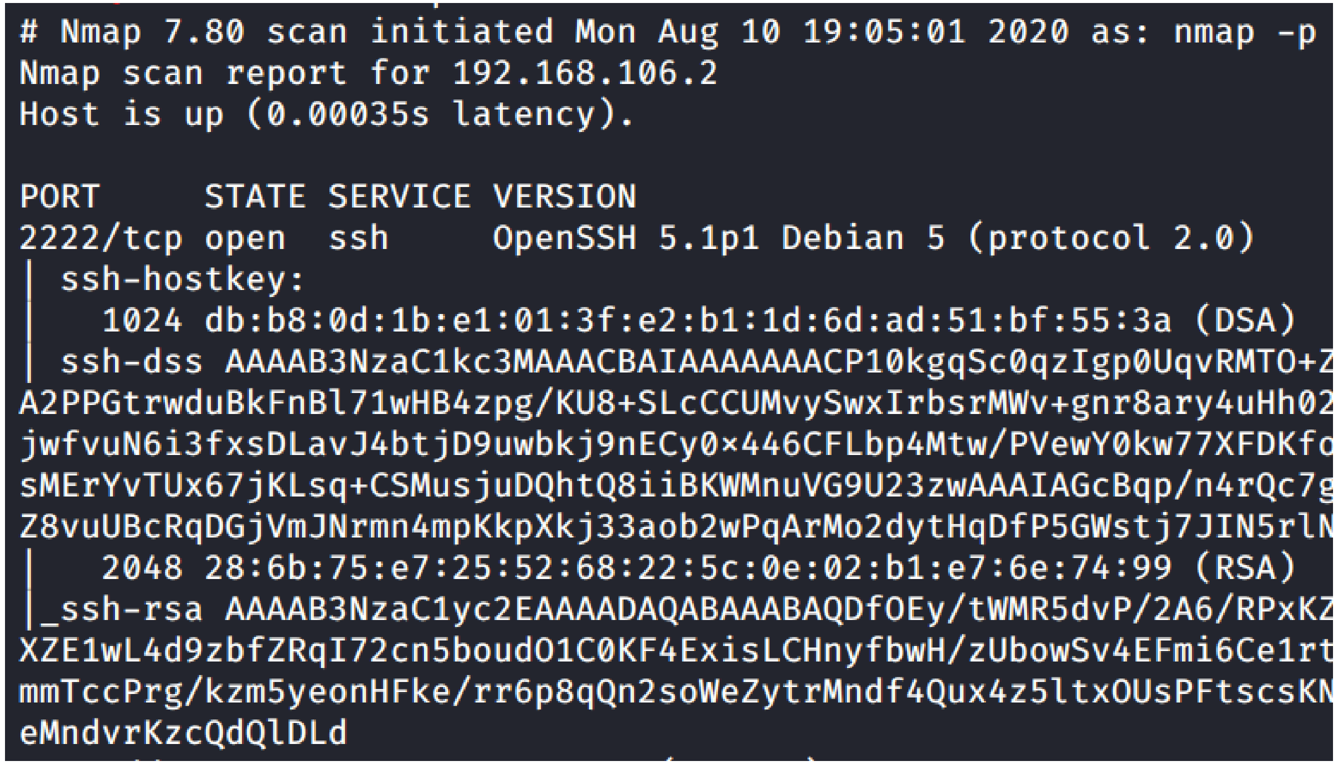
OCO: Identifying and Evading Deception

Identifying Kippo

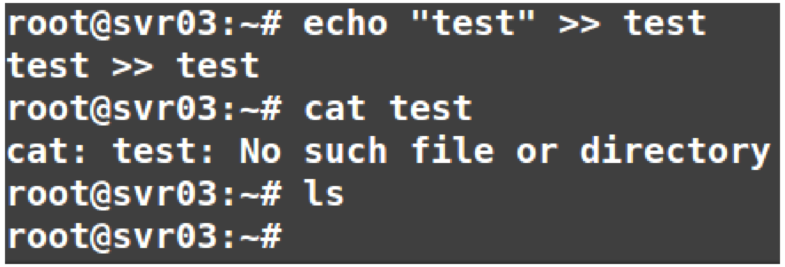
- Metasploit's Kippo detector doesn't work on the latest version of Kippo
- Regardless: Look out for the default Kippo SSH signature
  - *nmap -v -p [port] --script ssh-hostkey -sV [IP]*
  - *This could be changed by people administering Kippo*
- You also can't write to any files in Kippo. If you can't do that and you're root, you're likely in Kippo
- Immediately exit after connecting to SSH; are you in your own environment, or still stuck the SSH session? If stuck, you're in Kippo



```
msf5 auxiliary(scanner/ssh/detect_kippo) > exploit

[*] 192.168.106.2:2222    - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```



```
# Nmap 7.80 scan initiated Mon Aug 10 19:05:01 2020 as: nmap -p
Nmap scan report for 192.168.106.2
Host is up (0.00035s latency).

PORT     STATE SERVICE VERSION
2222/tcp open  ssh     OpenSSH 5.1p1 Debian 5 (protocol 2.0)
| ssh-hostkey:
|   1024 db:b8:0d:1b:e1:01:3f:e2:b1:1d:6d:ad:51:bf:55:3a (DSA)
| ssh-dss AAAAB3NzaC1kc3MAAACBAIAAAAAAACP10kgqSc0qzIgp0UqvRMTO+Z
A2PPGtrwduBkFnBl71wHB4zpg/KU8+SLcCCUMvySwxIrbsrMWv+gnr8ary4uHh02
jwfvuN6i3fxsDLavJ4btjD9uwbkj9nECy0×446CFLbp4Mtw/PVewY0kw77XFDKfo
sMErYvTUx67jKLsq+CSMusjuDQhtQ8iiBKWMnuVG9U23zwAAAIAGcBqp/n4rQc7g
Z8vuUBcRqDGjVmJNrmn4mpKkpXkj33aob2wPqArMo2dytHqDfP5GWstj7JIN5rlN
|   2048 28:6b:75:e7:25:52:68:22:5c:0e:02:b1:e7:6e:74:99 (RSA)
|_ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQDfOEy/tWMR5dvP/2A6/RPxKZ
XZE1wL4d9zbfZRqI72cn5boudO1C0KF4ExisLCHnyfbwH/zUbowSv4EFmi6Ce1rt
mmTccPrg/kzm5yeonHFke/rr6p8qQn2soWeZytrMndf4Qux4z5ltxOUsPFtscsKN
eMndvrKzcQdQlDLd
```

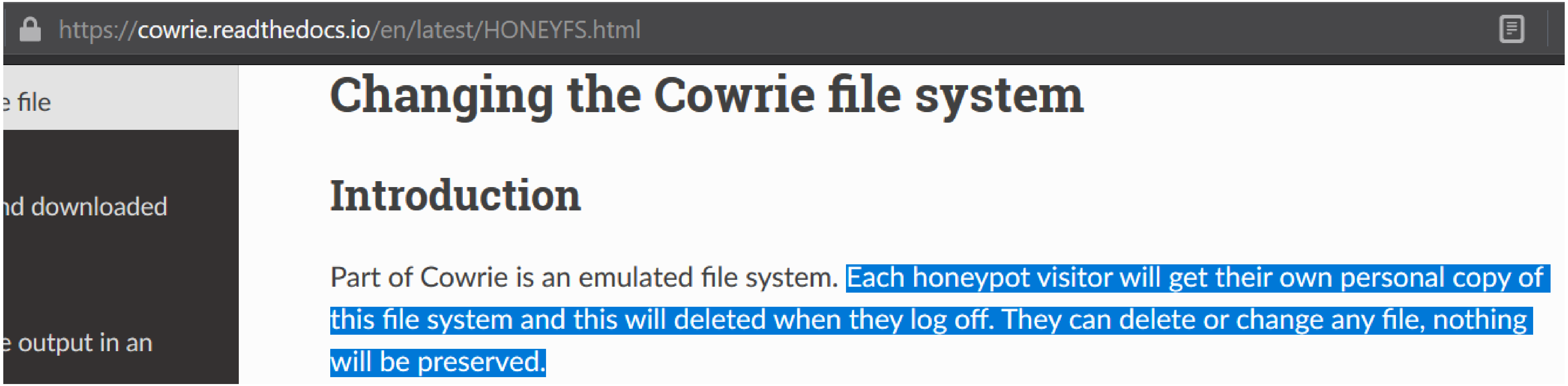


```
root@svr03:~# echo "test" >> test
test >> test
root@svr03:~# cat test
cat: test: No such file or directory
root@svr03:~# ls
root@svr03:~#
```

OCO: Identifying and Evading Deception

Identifying Cowrie

- While in Cowrie you can write to files, they're gone the second you quit your SSH session. Simply create a test file, write to it and relogin. If it doesn't exist, you're in Cowrie



Changing the Cowrie file system

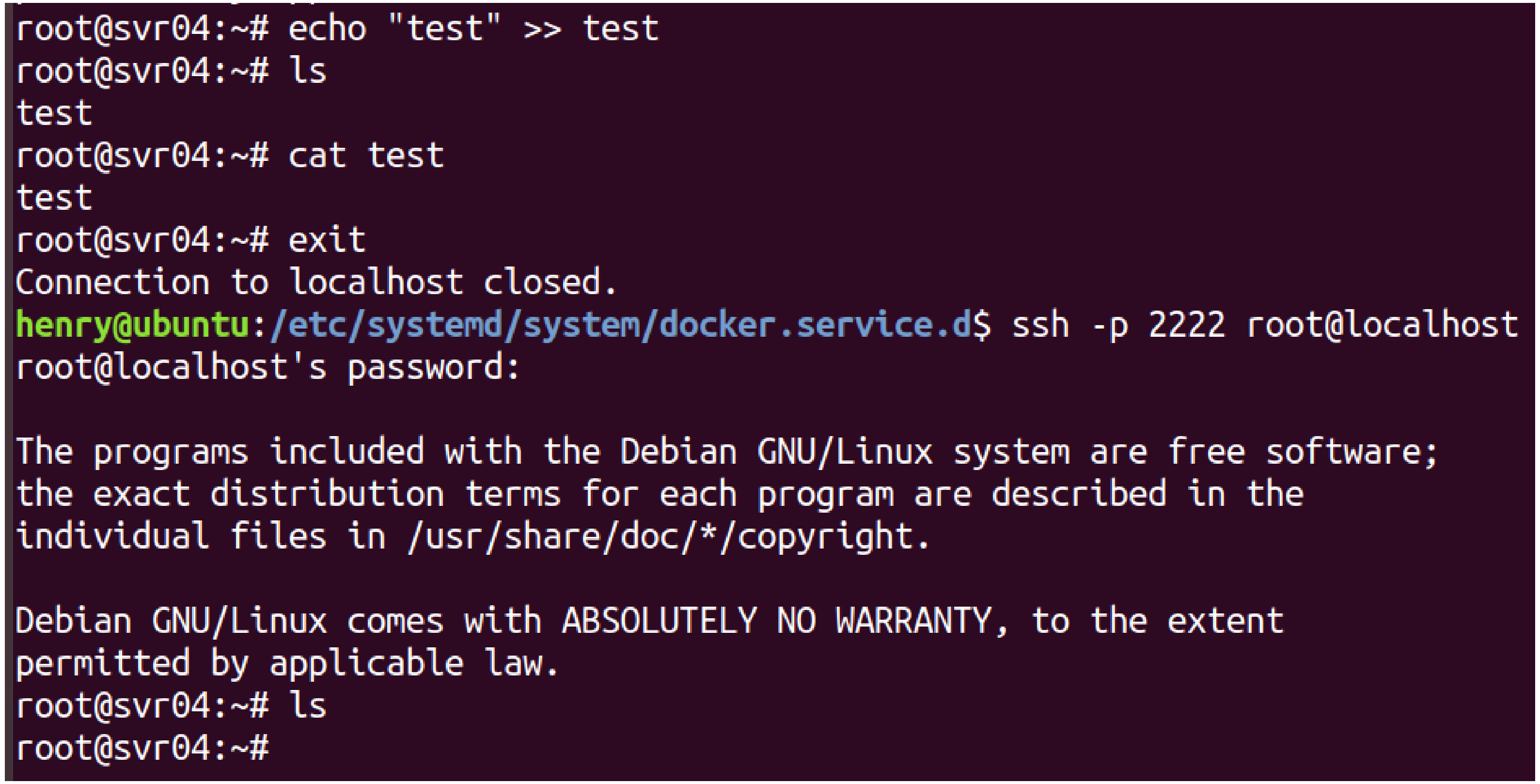
Introduction

Part of Cowrie is an emulated file system. Each honeypot visitor will get their own personal copy of this file system and this will deleted when they log off. They can delete or change any file, nothing will be preserved.

OCO: Identifying and Evading Deception

Identifying Cowrie

- While in Cowrie you can write to files, they're gone the second you quit your SSH session. Simply create a test file, write to it and relogin. If it doesn't exist, you're in Cowrie

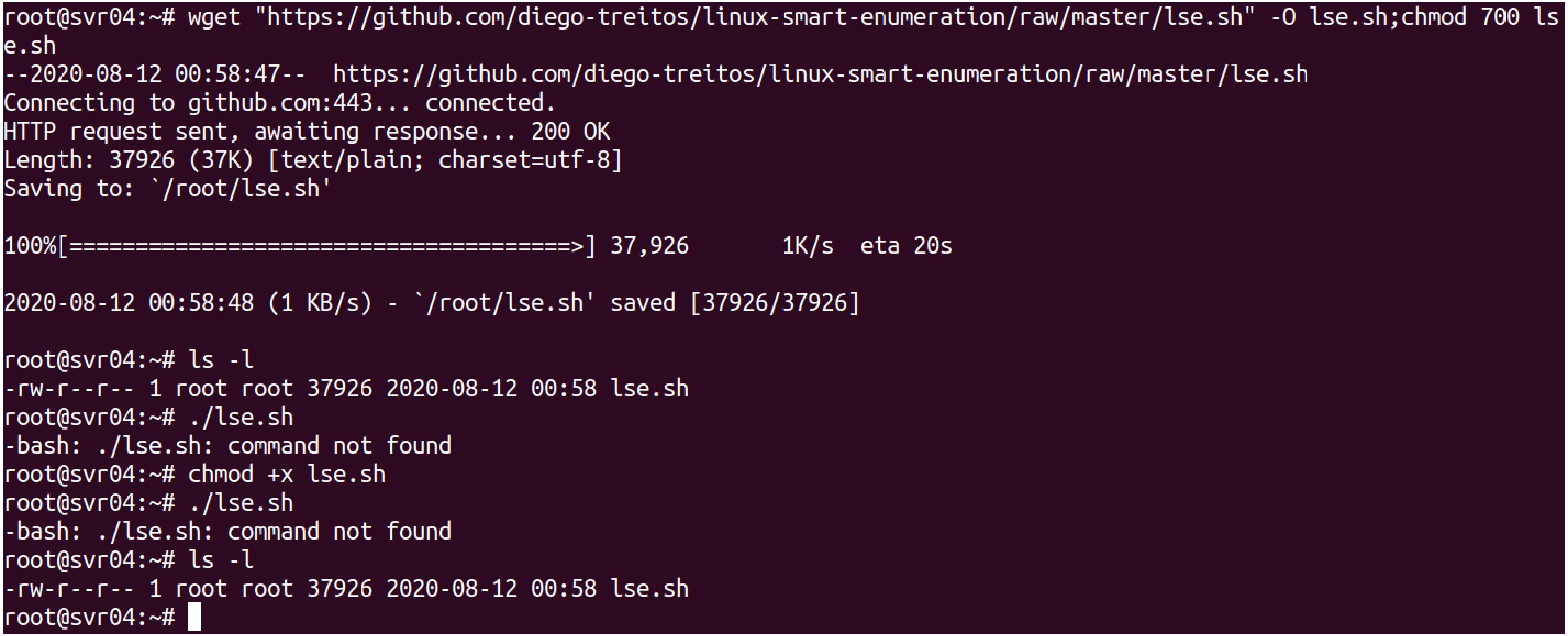


```
root@svr04:~# echo "test" >> test
root@svr04:~# ls
test
root@svr04:~# cat test
test
root@svr04:~# exit
Connection to localhost closed.
henry@ubuntu:/etc/systemd/system/docker.service.d$ ssh -p 2222 root@localhost
root@localhost's password:

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

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

OCO: Identifying and Evading Deception

Identifying Cowrie

- Downloading and running a script is not possible in Cowrie



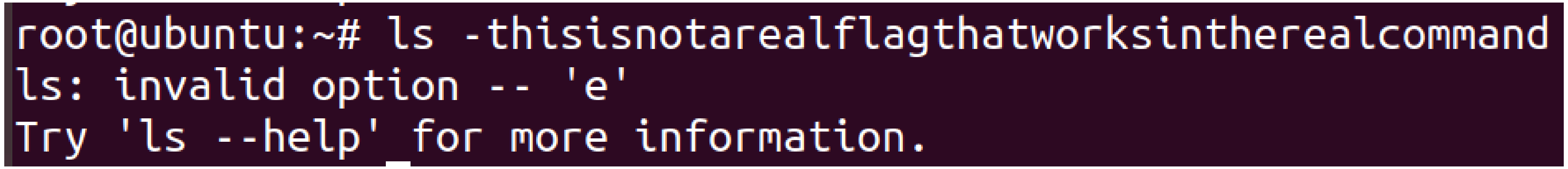
```
root@svr04:~# wget "https://github.com/diego-treitos/linux-smart-enumeration/raw/master/lse.sh" -O lse.sh;chmod 700 ls
e.sh
--2020-08-12 00:58:47--  https://github.com/diego-treitos/linux-smart-enumeration/raw/master/lse.sh
Connecting to github.com:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 37926 (37K) [text/plain; charset=utf-8]
Saving to: `/root/lse.sh'

100%[======================================>] 37,926       1K/s  eta 20s

2020-08-12 00:58:48 (1 KB/s) - `/root/lse.sh' saved [37926/37926]

root@svr04:~# ls -l
-rw-r--r-- 1 root root 37926 2020-08-12 00:58 lse.sh
root@svr04:~# ./lse.sh
-bash: ./lse.sh: command not found
root@svr04:~# chmod +x lse.sh
root@svr04:~# ./lse.sh
-bash: ./lse.sh: command not found
root@svr04:~# ls -l
-rw-r--r-- 1 root root 37926 2020-08-12 00:58 lse.sh
root@svr04:~# 
```

OCO: Identifying and Evading Deception

Identifying Cowrie

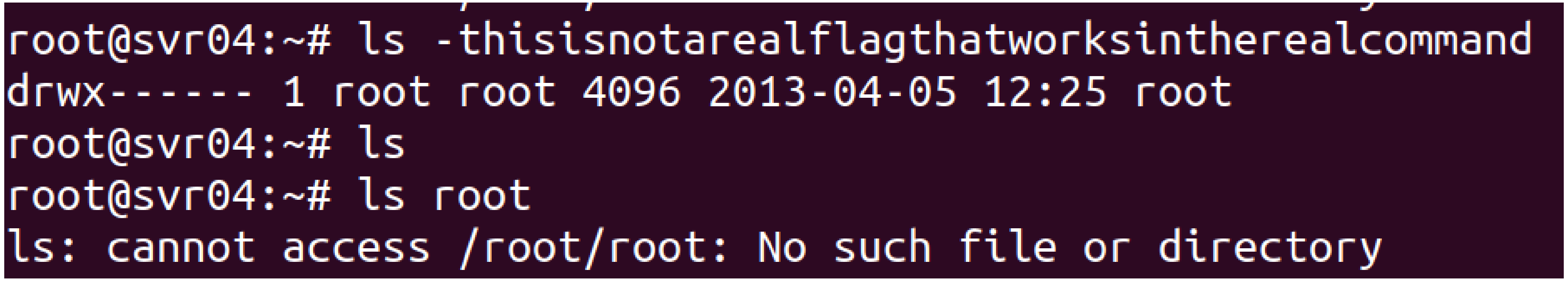
- Certain commands don’t generate errors when they should.

Ubuntu:



```
root@ubuntu:~# ls -thisisnotarealflagthatworksintherealcommand
ls: invalid option -- 'e'
Try 'ls --help' for more information.
```

Cowrie:



```
root@svr04:~# ls -thisisnotarealflagthatworksintherealcommand
drwx------ 1 root root 4096 2013-04-05 12:25 root
root@svr04:~# ls
root@svr04:~# ls root
ls: cannot access /root/root: No such file or directory
```

OCO: Identifying and Evading Deception

Identifying Honeyusers

- Able to enumerate all user accounts? Great. Here’s things to watch out for:
  - *Windows user doesn’t have a profile? Don’t log in to it.*
    - Profileless accounts have never been logged into. Splunk Enterprise Security, for example, can generate alerts on first login
    - Some HoneyUsers are made in a lazy fashion, where a profile is omitted
    - Last login date is Jan. 1, 1601—account has never been logged into
  - *Account disabled? Take it out of your password spray list*
    - Failed login attempts to disabled accounts can be monitored
  - *Are you able to see logon hours? If an account has logon hours set to “never”, it’s likely a HoneyUser. Avoid these if you can.*
- Password spray slowly if you are aware the victim has any monitoring capability
  - *Try to match password spray with the usual start of work day and time in the victim’s locality*

OCO: Identifying and Evading Deception

- General suggestion: Figure out what deception technology your victim uses, find its weaknesses
  - *Deception stops being deception when it is completely identical to whatever it's emulating, so a weak point exists—you just need to find it*
- Don't scan the entire network when you're inside
  - *You shouldn't be doing this even if cyber deception isn't used*
- Don't touch machines you have no reason to attack
  - *Especially if those machines have seemingly no breadcrumbs leading to them and they run an ancient operating system in an environment with otherwise up-to-date machines*

- For defensive teams:
  - *Following all of this advice greatly slows down the attacker*
  - *If you're using deception tools on production systems you're already at an advantage*

Future Research

- Document common vulnerabilities in other services and implement similar techniques to CORECPRO
- Automate breadcrumb generation to aid in steering attackers into traps
  - *Integrate with automation tools like Ansible to quickly deploy breadcrumbs*

Questions?

Sources

1. P. 216. “2011 Report to Congress”. *U.S.-China Economic and Security Review Commission.* https://www.uscc.gov/sites/default/files/annual_reports/annual_report_full_11.pdf
2. P. 7 “Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1” National Institute of Standards and Technology. https://nvlpubs.nist.gov/nistpubs/CSWP/NIST.CSWP.04162018.pdf
3. P. 7 “A ‘Kill Chain’ Analysis of the 2013 Target Data Breach”. *US Senate Committee on Commerce, Science, and Transportation. https://www.commerce.senate.gov/services/files/24d3c229-4f2f-405d-b8db-a3a67f183883*
4. Dominguez, Andrea. “The State of Honeypots: Understanding the Use of Honey Technologies Today”. *SANS Institute. https://www.sans.org/reading-room/whitepapers/detection/state-honeypots-understanding-honey-technologies-today-38165*
5. Strand, John. “Webcast: Getting Started in Cyber Deception”. *Black Hills Information Security*. https://www.youtube.com/watch?v=cCxbBz1UbnA
6. Nikos Virilis, et al. “Changing the game: The art of deceiving sophisticated attackers” *North Atlantic Treaty Organization*. https://ieeexplore.ieee.org/document/6916397
7. Strand, John. “Getting Started with Tracking Hackers with HoneyBadger”. *Black Hills Information Security.* https://www.youtube.com/watch?v=wsHDC1LD8_w

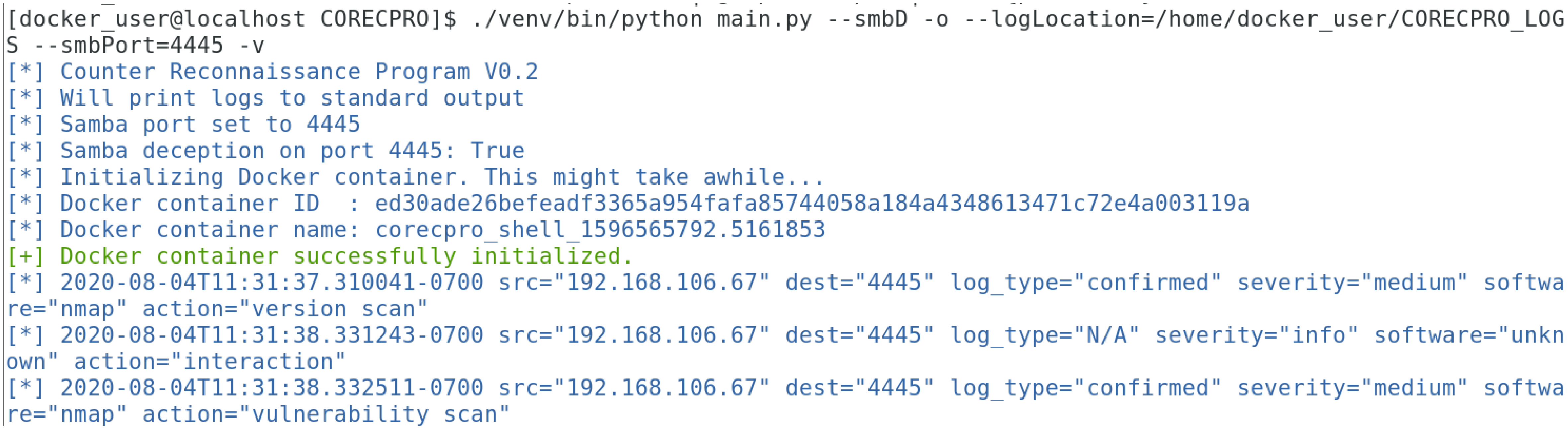
Sources

8. Strand, John. “Active Defense & Cyber Deception - Part 3”. *Black Hills Information Security.* https://www.youtube.com/watch?v=vmfB2u6rXtk
9. Strand, John. 11:04 “Active Defense & Cyber Deception - Part 2”. *Black Hills Information Security.* https://www.youtube.com/watch?v=qGwqYjJZclU

Backup Slides

Counter Reconnaissance Program

Current capabilities

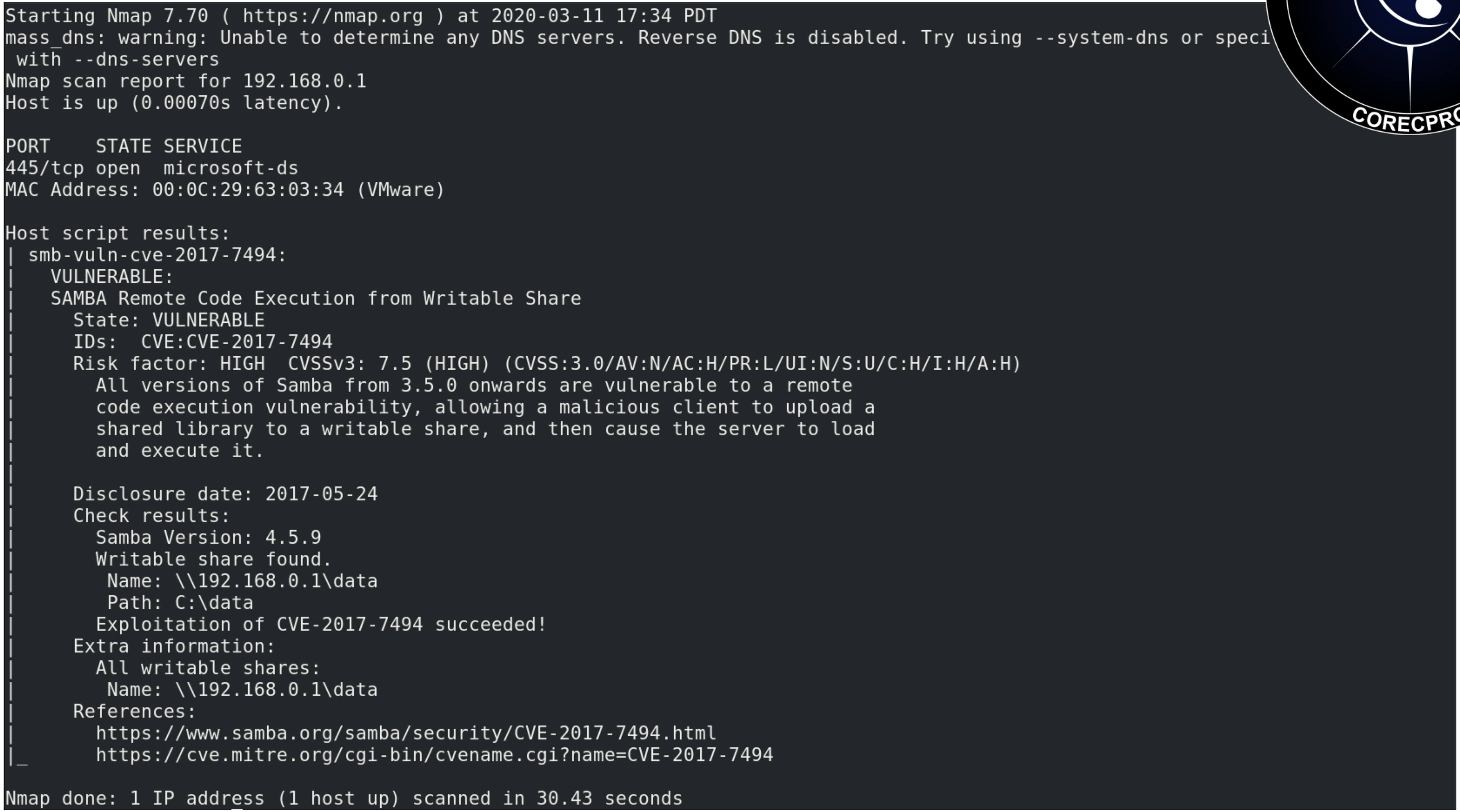


```
[docker_user@localhost CORECPRO]$ ./venv/bin/python main.py --smbD -o --logLocation=/home/docker_user/CORECPRO_LOG
S --smbPort=4445 -v
[*] Counter Reconnaissance Program V0.2
[*] Will print logs to standard output
[*] Samba port set to 4445
[*] Samba deception on port 4445: True
[*] Initializing Docker container. This might take awhile...
[*] Docker container ID  : ed30ade26befeadf3365a954fafa85744058a184a4348613471c72e4a003119a
[*] Docker container name: corecpro_shell_1596565792.5161853
[+] Docker container successfully initialized.
[*] 2020-08-04T11:31:37.310041-0700 src="192.168.106.67" dest="4445" log_type="confirmed" severity="medium" softwa
re="nmap" action="version scan"
[*] 2020-08-04T11:31:38.331243-0700 src="192.168.106.67" dest="4445" log_type="N/A" severity="info" software="unkn
own" action="interaction"
[*] 2020-08-04T11:31:38.332511-0700 src="192.168.106.67" dest="4445" log_type="confirmed" severity="medium" softwa
re="nmap" action="vulnerability scan"
```

Counter Reconnaissance Program

Current capabilities

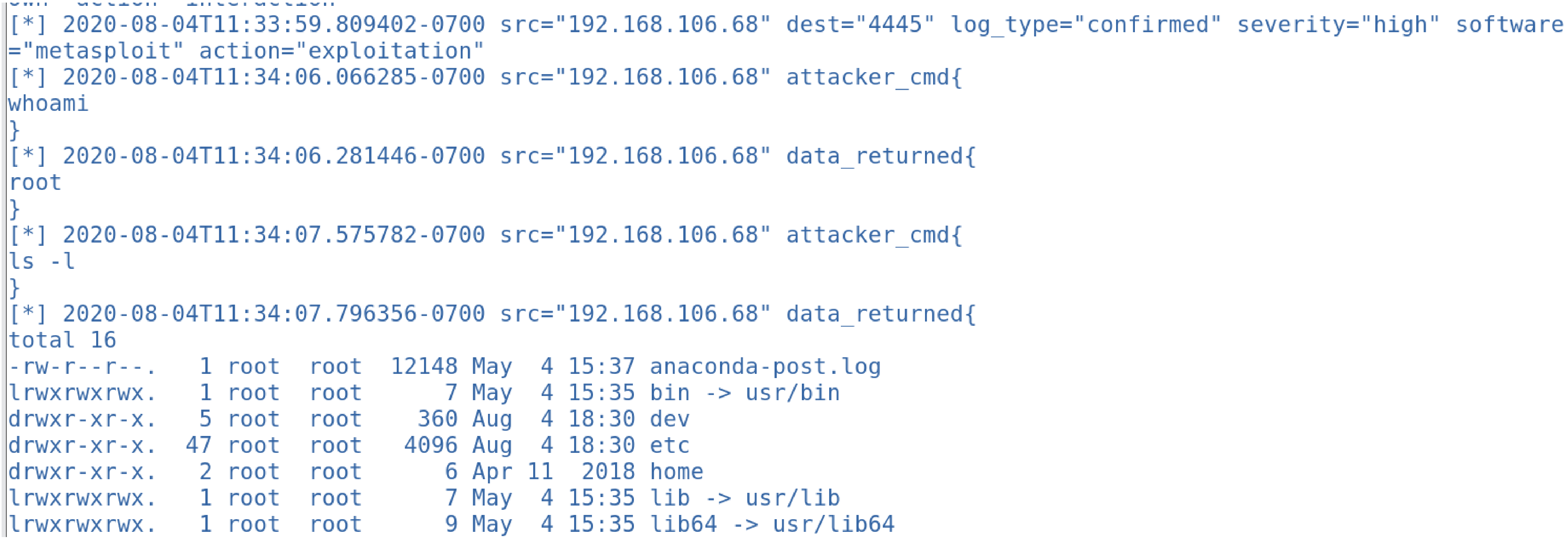
ACTA DEOS NUMQUAM MORTALIA FALLUNT
CORECPRO



```
Starting Nmap 7.70 ( https://nmap.org ) at 2020-03-11 17:34 PDT
mass_dns: warning: Unable to determine any DNS servers. Reverse DNS is disabled. Try using --system-dns or speci
 with --dns-servers
Nmap scan report for 192.168.0.1
Host is up (0.00070s latency).

PORT    STATE SERVICE
445/tcp open  microsoft-ds
MAC Address: 00:0C:29:63:03:34 (VMware)

Host script results:
| smb-vuln-cve-2017-7494:
|   VULNERABLE:
|   SAMBA Remote Code Execution from Writable Share
|     State: VULNERABLE
|     IDs:  CVE:CVE-2017-7494
|     Risk factor: HIGH  CVSSv3: 7.5 (HIGH) (CVSS:3.0/AV:N/AC:H/PR:L/UI:N/S:U/C:H/I:H/A:H)
|       All versions of Samba from 3.5.0 onwards are vulnerable to a remote
|       code execution vulnerability, allowing a malicious client to upload a
|       shared library to a writable share, and then cause the server to load
|       and execute it.
|
|     Disclosure date: 2017-05-24
|     Check results:
|       Samba Version: 4.5.9
|       Writable share found.
|        Name: \\192.168.0.1\data
|        Path: C:\data
|       Exploitation of CVE-2017-7494 succeeded!
|     Extra information:
|       All writable shares:
|        Name: \\192.168.0.1\data
|     References:
|       https://www.samba.org/samba/security/CVE-2017-7494.html
|_      https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2017-7494

Nmap done: 1 IP address (1 host up) scanned in 30.43 seconds
```

Counter Reconnaissance Program

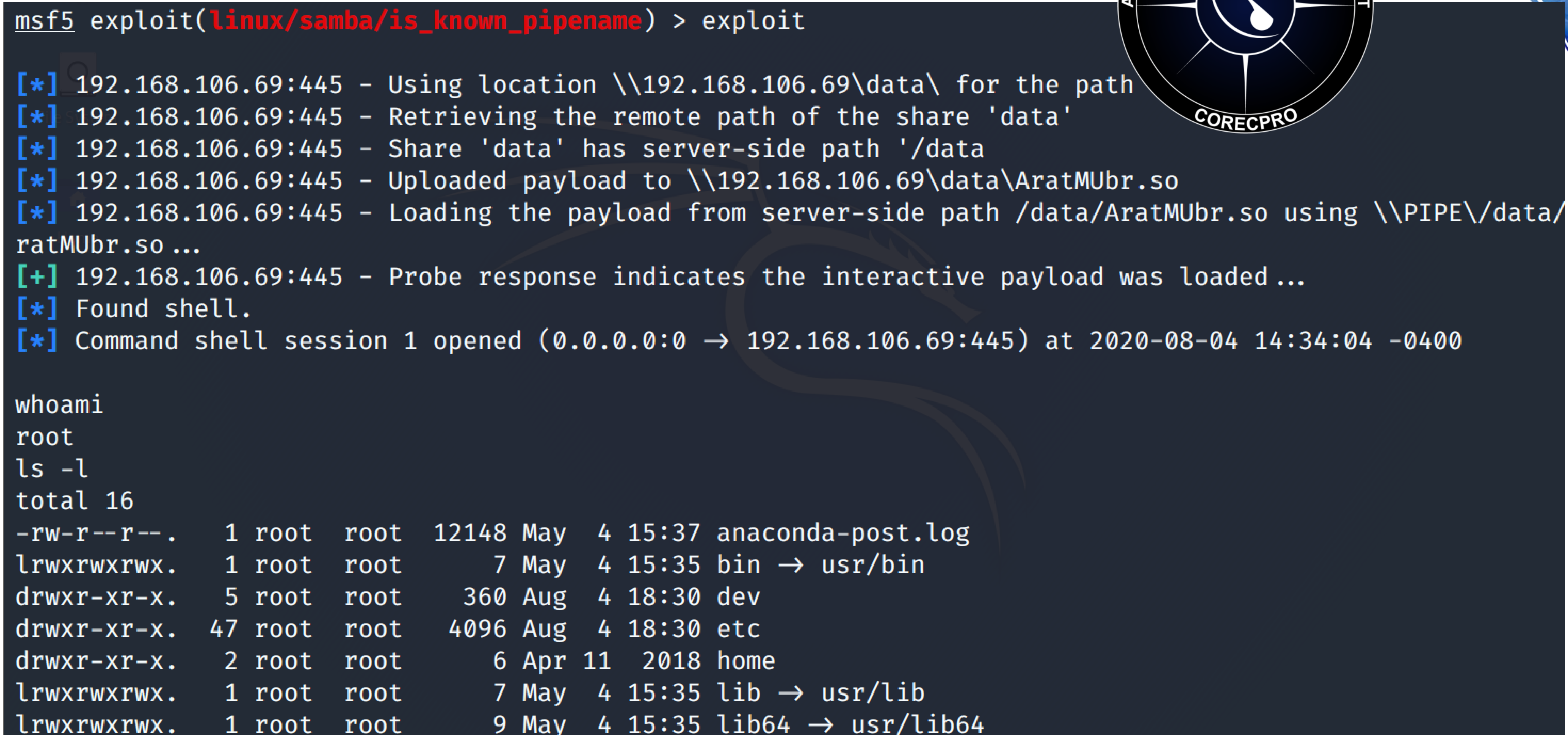
Current capabilities



```
[*] 2020-08-04T11:33:59.809402-0700 src="192.168.106.68" dest="4445" log_type="confirmed" severity="high" software
="metasploit" action="exploitation"
[*] 2020-08-04T11:34:06.066285-0700 src="192.168.106.68" attacker_cmd{
whoami
}
[*] 2020-08-04T11:34:06.281446-0700 src="192.168.106.68" data_returned{
root
}
[*] 2020-08-04T11:34:07.575782-0700 src="192.168.106.68" attacker_cmd{
ls -l
}
[*] 2020-08-04T11:34:07.796356-0700 src="192.168.106.68" data_returned{
total 16
-rw-r--r--.   1 root  root  12148 May  4 15:37 anaconda-post.log
lrwxrwxrwx.   1 root  root      7 May  4 15:35 bin -> usr/bin
drwxr-xr-x.   5 root  root    360 Aug  4 18:30 dev
drwxr-xr-x.  47 root  root   4096 Aug  4 18:30 etc
drwxr-xr-x.   2 root  root      6 Apr 11  2018 home
lrwxrwxrwx.   1 root  root      7 May  4 15:35 lib -> usr/lib
lrwxrwxrwx.   1 root  root      9 May  4 15:35 lib64 -> usr/lib64
```

Counter Reconnaissance Program

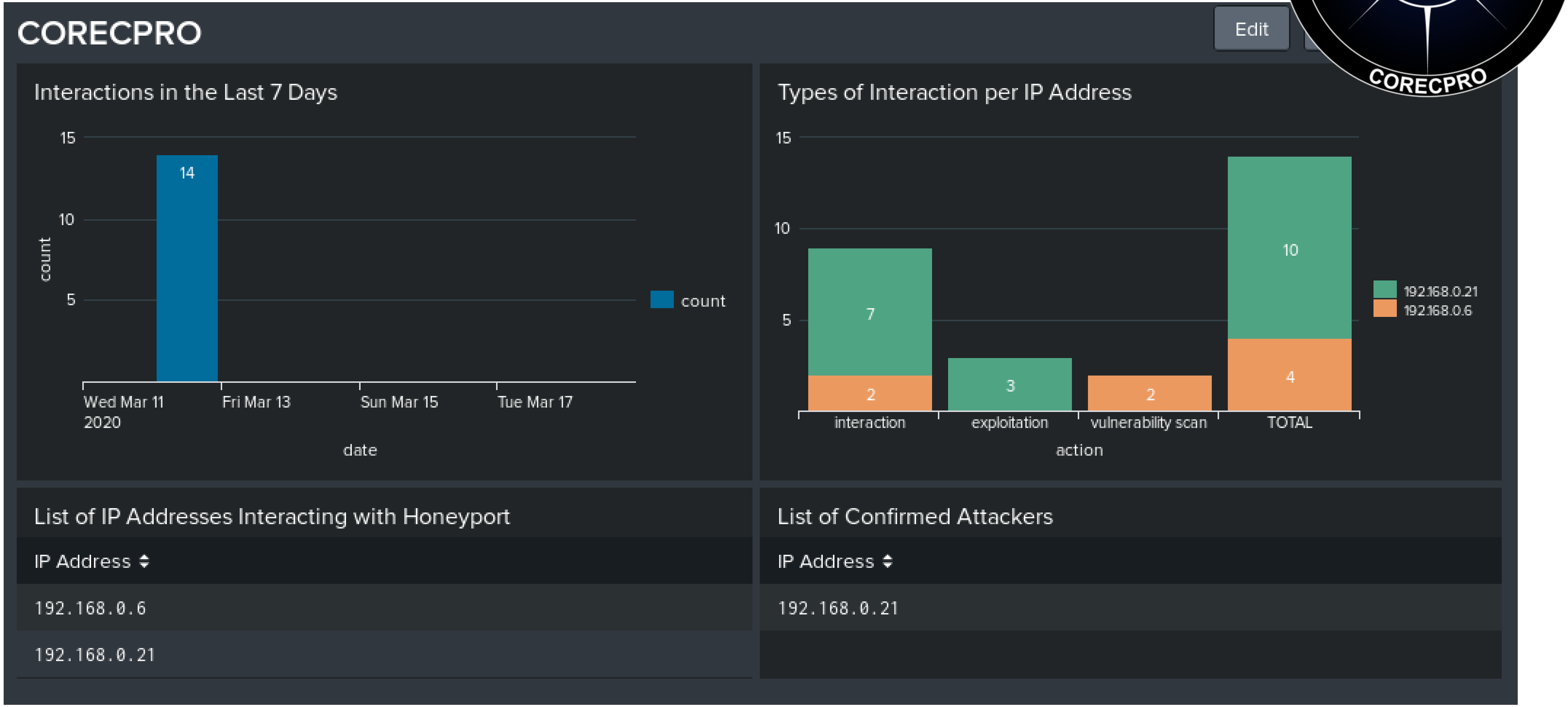
Current capabilities



```
msf5 exploit(linux/samba/is_known_pipename) > exploit

[*] 192.168.106.69:445 - Using location \\192.168.106.69\data\ for the path
[*] 192.168.106.69:445 - Retrieving the remote path of the share 'data'
[*] 192.168.106.69:445 - Share 'data' has server-side path '/data
[*] 192.168.106.69:445 - Uploaded payload to \\192.168.106.69\data\AratMUbr.so
[*] 192.168.106.69:445 - Loading the payload from server-side path /data/AratMUbr.so using \\PIPE\/data/
ratMUbr.so ...
[+] 192.168.106.69:445 - Probe response indicates the interactive payload was loaded ...
[*] Found shell.
[*] Command shell session 1 opened (0.0.0.0:0 → 192.168.106.69:445) at 2020-08-04 14:34:04 -0400

whoami
root
ls -l
total 16
-rw-r--r--.   1 root  root  12148 May  4 15:37 anaconda-post.log
lrwxrwxrwx.   1 root  root      7 May  4 15:35 bin → usr/bin
drwxr-xr-x.   5 root  root    360 Aug  4 18:30 dev
drwxr-xr-x.  47 root  root   4096 Aug  4 18:30 etc
drwxr-xr-x.   2 root  root      6 Apr 11  2018 home
lrwxrwxrwx.   1 root  root      7 May  4 15:35 lib → usr/lib
lrwxrwxrwx.   1 root  root      9 May  4 15:35 lib64 → usr/lib64
```

Counter Reconnaissance Program

Current capabilities



CORECPRO

Interactions in the Last 7 Days

Types of Interaction per IP Address

List of IP Addresses Interacting with Honeyport

IP Address ⬍
192.168.0.6
192.168.0.21

List of Confirmed Attackers

IP Address ⬍
192.168.0.21