Buckle It Up (Or Shells Die!)

Will Schroeder (@harmj0y)

Lee Christensen (@tifkin_)

@harmj0y – Will Schroeder

- **Career:** Technical Architect at SpecterOps
- **Code:** Veil-Framework, Empire, PowerView/PowerUp, BloodHound, GhostPack
- **Cons:** DerbyCon (RIP), BlackHat, DEF CON, Troopers, others
- **Content:** Veteran trainer (*Adversary Tactics: Red Team Operations*/others), sometimes blogs at http://blog.harmj0y.net

@tifkin_ – Lee Christensen

- **Career:** Technical Architect at SpecterOps
- **Code / Interests:** Seatbelt, SpoolSample, UnmanagedPowerShell, ❤️ Windows Internals/AD/PowerShell and attacking new enterprise tech
- **Cons:** DerbyCon (RIP), BlackHat, DEF CON, Troopers, others
- **Content:** Veteran trainer (Adversary Tactics: Red Team Operations, Adversary Tactics: Vulnerability Research for Operators), rarely blogs at https://medium.com/@tifkin

tl;dr

- What Host-Based Situational Awareness is, and Why it Matters
- Data collection with Seatbelt
- Host-based Situational Awareness in the Attack Cycle
  - Defensive Enumeration
  - Exploitation and Vulnerability Research Target Selection
  - Credential Theft
  - User/System Behavioral Baselining
- Technique Selection (persistence, lateral movement, etc.)

Host-based Situational Awareness

What It Is

Why It Matters

Host-based Situational Awareness

- Perceiving/understanding the environment using host artifacts
    - Data collected in an environment should continually influence TTP selection throughout an engagement
    - Situational awareness is one of the main sources of this type of information
- Informs us about **capabilities**
    - What are we **capable** of doing as an attacker? What's possible?
- Informs us **strategically**
    - Given what's possible, what should we do next and how?

Using Data to Guide Ops

- Any action you perform is a detectable risk
    - *“Everything is stealthy until someone is looking for it.” - Lee*
- Your risk tolerance for detection depends on:
    - Assessment training objectives
    - The current attack strategy - Smash and grab? Low and slow?
- “Enlightened Actors” understand the impact of each action performed and make a risk-based decision before acting
- **Collect** relevant data, **Calculate** risk from that data, **Act** accordingly

Example Attack Considerations

- Don’t just run **sekurlsa::logonPasswords** first thing!
- Is it worth even pursuing credential extraction?
  - Are you elevated? Do you currently have local administrative rights?
  - Is there even a useful logon session currently on the system?
  - Is wdigest enabled, making a specific action worth the risk?
- Will defenses affect extraction actions?
  - What defensive products are current deployed? Are there exceptions?
  - Does something make this “impossible”? (RunAsPPL, Credguard, etc.)
  - How do you run target code? (C#, PowerShell, fork+run, in-process, etc.)

SA and Attack Phases

- **Initial Access**
  - Most fragile part of an engagement
  - Collect as much data (especially about defenses) as you can - if kicked out, you have a roadmap back in
- **Lateral Movement**
  - Much of this data can be enumerated remotely from a host that you have administrative rights to!
- **Strategic Hunting**
  - When searching for specific objectives (i.e. cookies for cloud platforms)

Weaponization

With Seatbelt

Seatbelt: Original Goal

- Handful of “safety checks” and security product enumeration in C#
  - Based on a few PowerShell host SA scripts we used previously
- Expanded slightly....
  - 100+ commands now
  - Clearinghouse for any host-based artifact that might be interesting from an attacker's perspective
  - **ALL** of these have been useful to us in one situation or another

Seatbelt: Current Goals

- **Identify data sources** that are useful for an attacker
- Point out **what’s possible** and provide **source code examples**
- **Data Interpretation Callouts**
  - Notify the operator of "interesting" artifacts
  - Data means nothing if you don't know how to interpret/understand it
  - (Admittedly, we have a lot of room for improvement here)

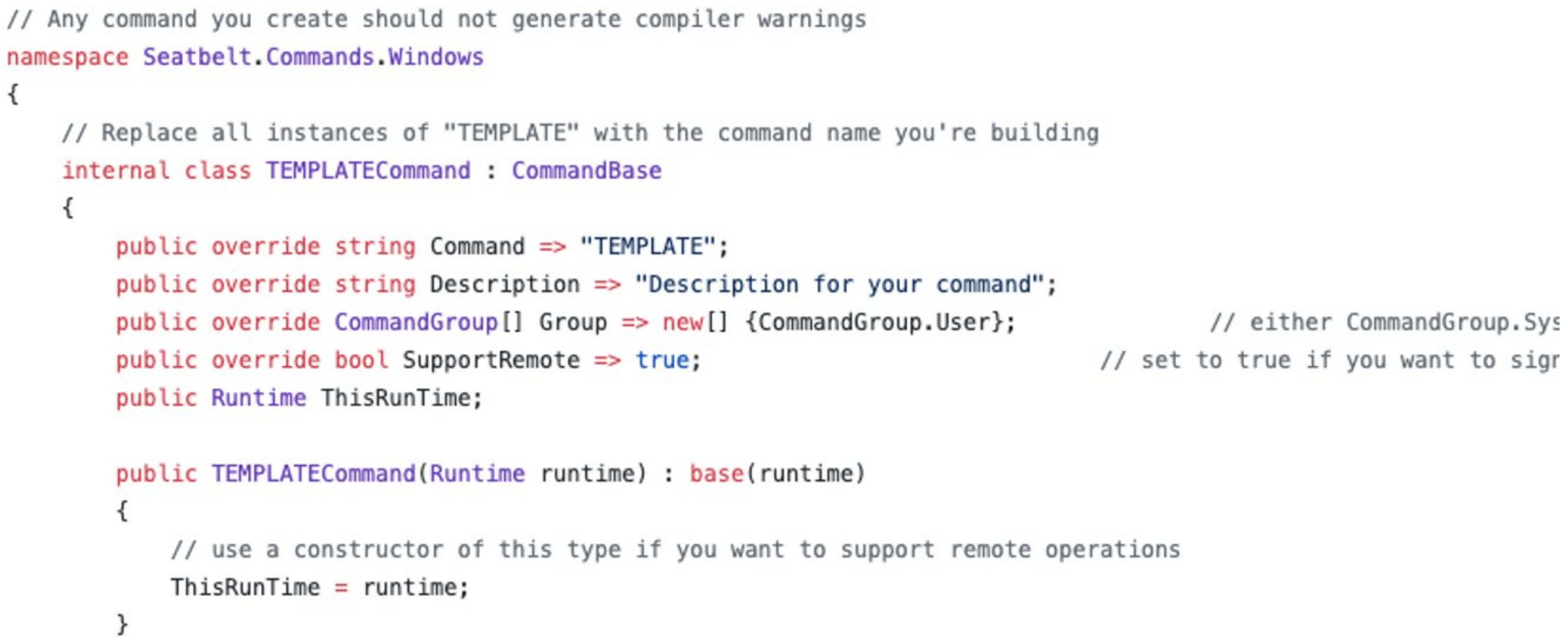
Reference: Seatbelt Collection Primitives

Method	Remote Support	Notes
Registry Reads	Yes (assuming admin rights)	Implemented with StdRegProv over WMI
File Reads	Yes (assuming admin rights)	Basic file reading
Event Log Reads	Yes (assuming admin rights)	.NET's EventLogQuery/EventLogSession
COM	Not currently	Some COM interfaces implement DCOM, some don't
API calls	Only for some	Some things like TCP connections are restricted to local host collection only.

Seatbelt Modularity

- Everything is drag and drop if you want to build custom internal modules
  - Easy to remove functionality too (reduces footprint on host)

- Template at **Seatbelt/Commands/Template.cs** (see next slide)



```
// Any command you create should not generate compiler warnings
namespace Seatbelt.Commands.Windows
{
    // Replace all instances of "TEMPLATE" with the command name you're building
    internal class TEMPLATECommand : CommandBase
    {
        public override string Command => "TEMPLATE";
        public override string Description => "Description for your command";
        public override CommandGroup[] Group => new[] {CommandGroup.User};              // either CommandGroup.Sys
        public override bool SupportRemote => true;                             // set to true if you want to sigr
        public Runtime ThisRunTime;

        public TEMPLATECommand(Runtime runtime) : base(runtime)
        {
            // use a constructor of this type if you want to support remote operations
            ThisRunTime = runtime;
        }
```

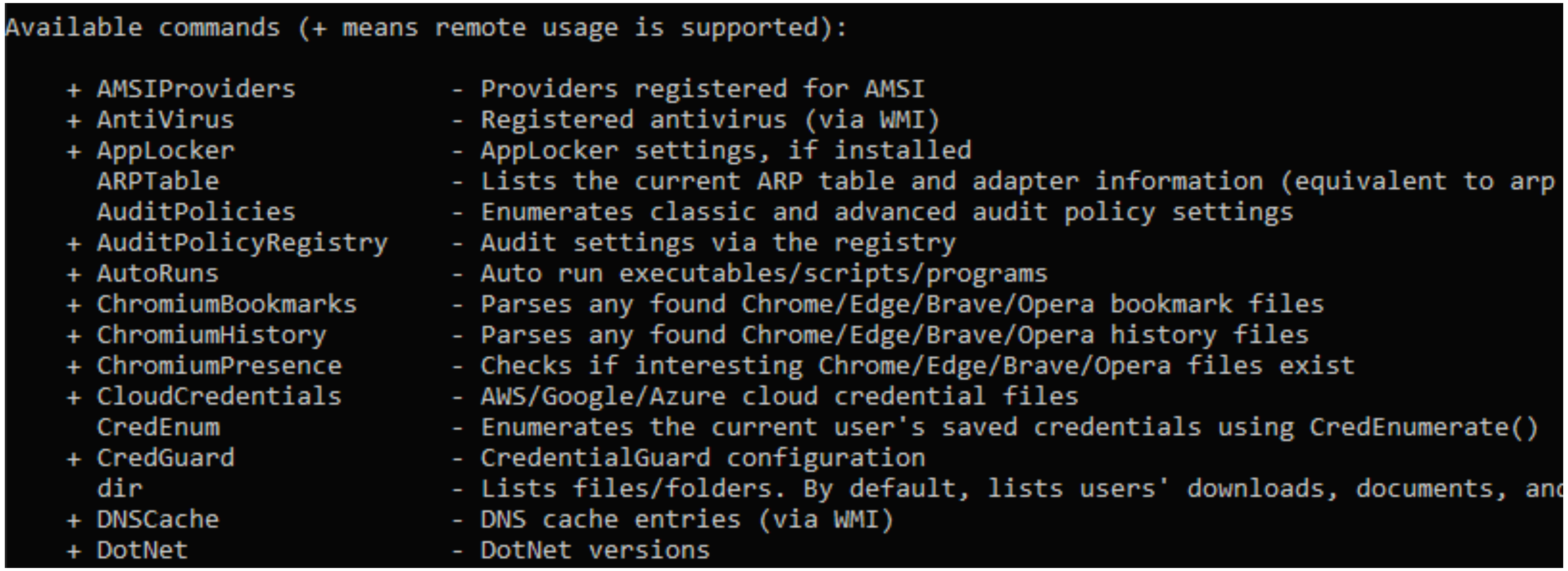
Seatbelt Command Groups

- Commands can be part of one or more "command groups"
- Run with **Seatbelt.exe -group=X**

-group=all	: all commands
-group=user	: user behavior-focused commands
-group=system	: system profiling
-group=slack	: Slack-specific modules
-group=chromium	: Chromium-specific modules
-group=remote	: modules useful for remote enumeration
-group=misc	: everything else

Seatbelt (Remote) Usage

- Run with **-computername=COMPUTER.DOMAIN.COM**
- Any commands with + support remote collection:



```
Available commands (+ means remote usage is supported):

    + AMSIProviders            - Providers registered for AMSI
    + AntiVirus                - Registered antivirus (via WMI)
    + AppLocker                - AppLocker settings, if installed
      ARPTable                 - Lists the current ARP table and adapter information (equivalent to arp
      AuditPolicies            - Enumerates classic and advanced audit policy settings
    + AuditPolicyRegistry      - Audit settings via the registry
    + AutoRuns                 - Auto run executables/scripts/programs
    + ChromiumBookmarks        - Parses any found Chrome/Edge/Brave/Opera bookmark files
    + ChromiumHistory          - Parses any found Chrome/Edge/Brave/Opera history files
    + ChromiumPresence         - Checks if interesting Chrome/Edge/Brave/Opera files exist
    + CloudCredentials         - AWS/Google/Azure cloud credential files
      CredEnum                 - Enumerates the current user's saved credentials using CredEnumerate()
    + CredGuard                - CredentialGuard configuration
      dir                      - Lists files/folders. By default, lists users' downloads, documents, and
    + DNSCache                 - DNS cache entries (via WMI)
    + DotNet                   - DotNet versions
```

Seatbelt Command Arguments

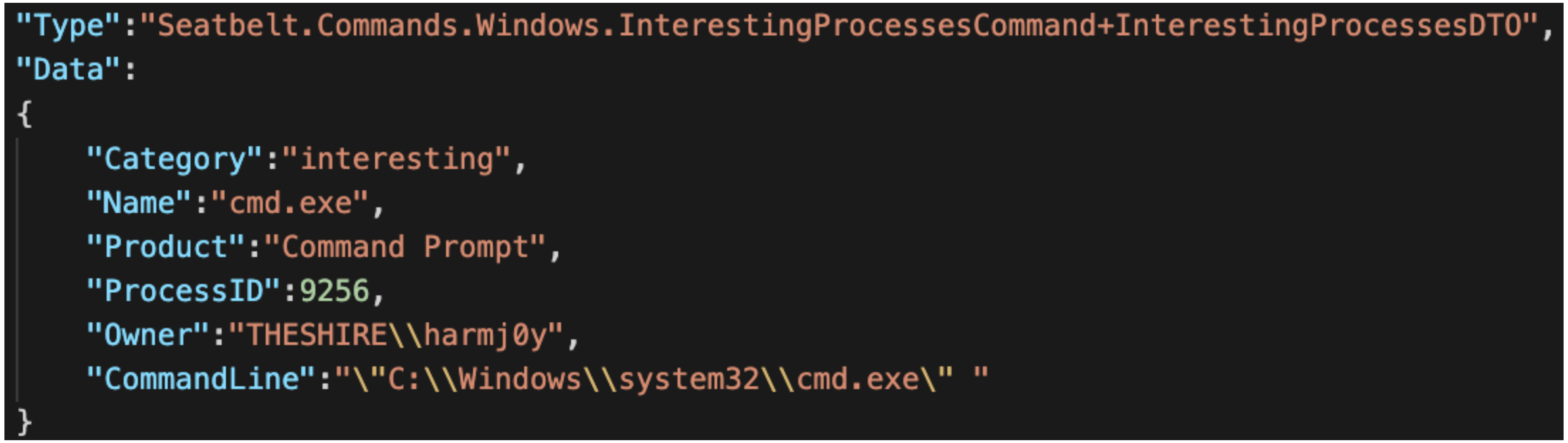
- Some modules support arguments (check module source)
  - Also, **-full** global argument prevents filtering of data

- **Seatbelt.exe "LogonEvents 60"** – returns logon events for the last 60 days instead of the default 10

- **Seatbelt.exe "SearchIndex C:\Path\"** – queries the search indexer for files in a specific path

Seatbelt Output

- **Text file output:** `-outputfile="C:\Temp\out.txt"`
- **JSON output:** `-outputFile="C:\Temp\out.json"`
  - Makes output digestible by automated systems!



```
"Type":"Seatbelt.Commands.Windows.InterestingProcessesCommand+InterestingProcessesDTO",
"Data":
{
    "Category":"interesting",
    "Name":"cmd.exe",
    "Product":"Command Prompt",
    "ProcessID":9256,
    "Owner":"THESHIRE\\harmj0y",
    "CommandLine":"\"C:\\Windows\\system32\\cmd.exe\" "
}
```

Defensive Enumeration

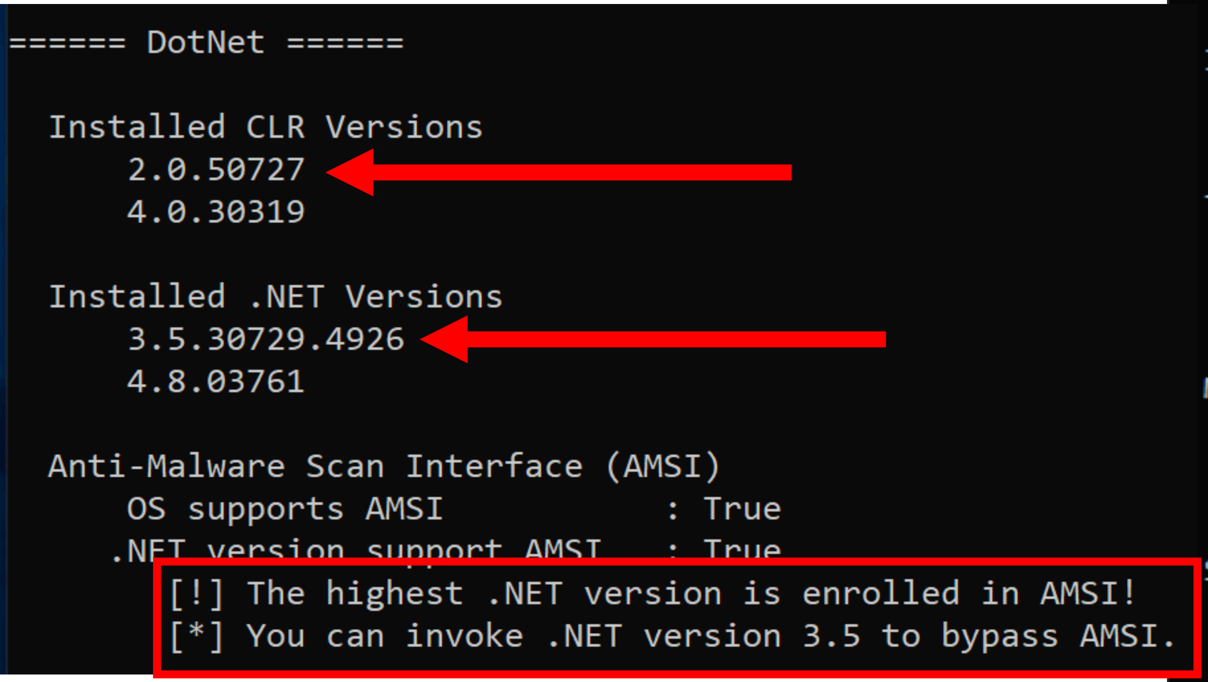
The Genesis

Defensive Enumeration / "Safety Checks"

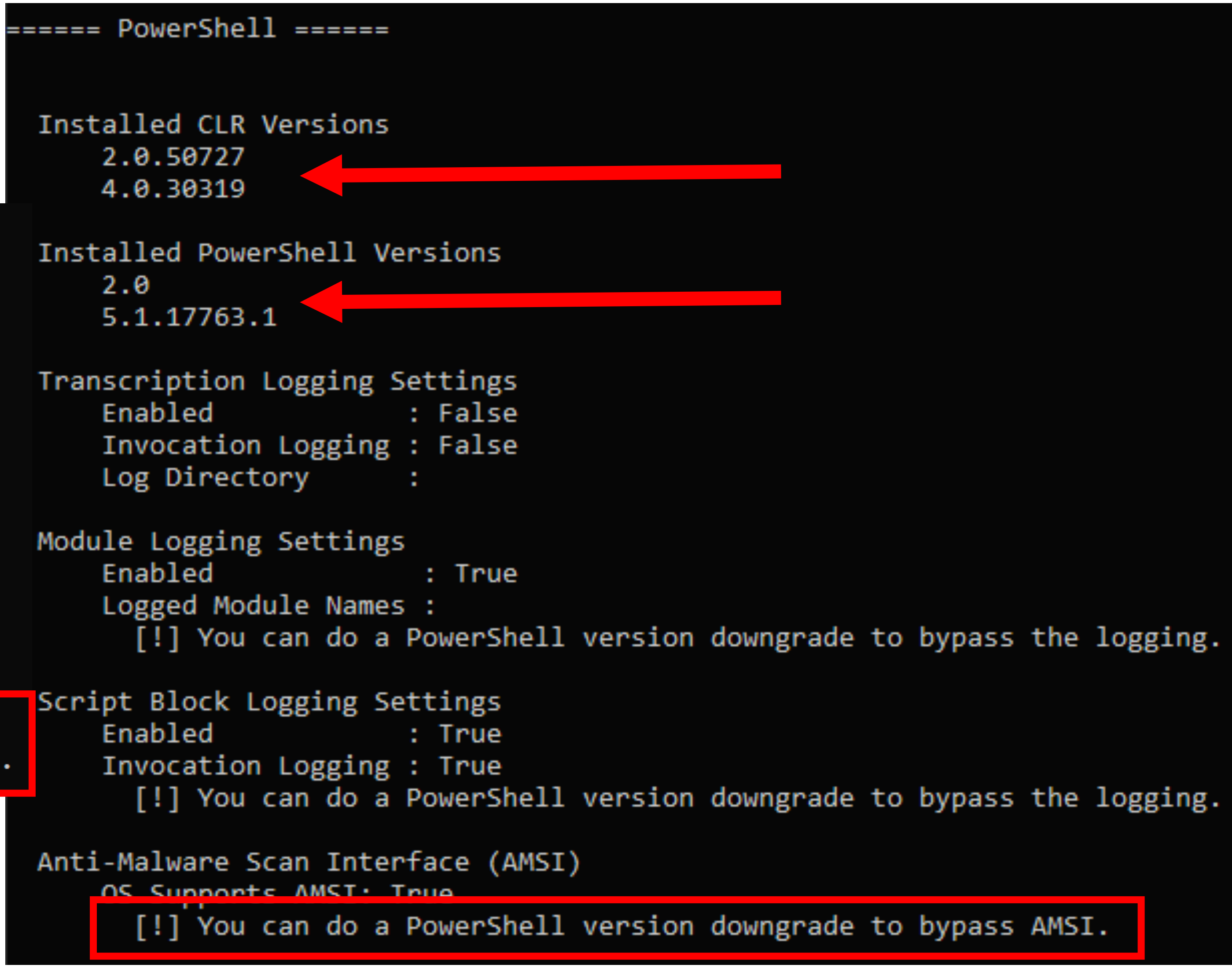
- What defensive tooling is there? How is it (mis)configured?
- What detective and preventative OS settings are enabled?
  - Example: What audit logs are enabled? Are host-based firewall enabled?
- Affects tools and technique selection
  - Empire? Beacon? Custom agent? How does IT admin machines?
- Enumeration can let you know *what* is possible and *what* might get you caught

Reference: Relevant Modules

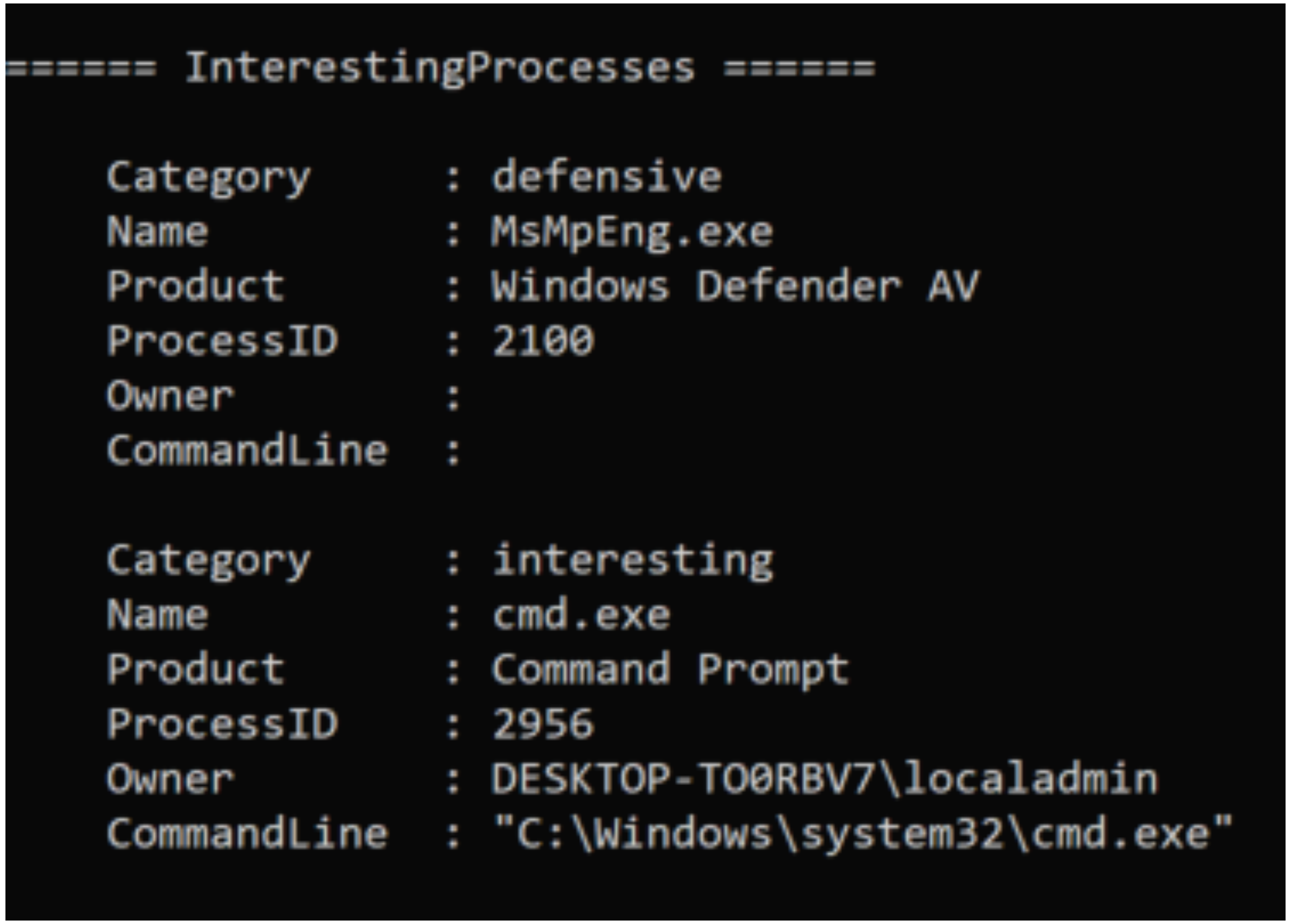
AMSIProviders
AntiVirus
AppLocker
AuditPolicies
AuditPoliciesRegistry
CredGuard / LSASettings (RunAsPPL)
InterestingProcesses
LAPS
LocalGPOs
McAfeeConfigs
NTLMSettings
PowerShell / DotNet
PSSessionSettings
RDPsettings
Sysmon
UAC
UserRightAssignments
WindowsDefender
WindowsEventForwarding
WindowsFirewall / NetworkProfiles



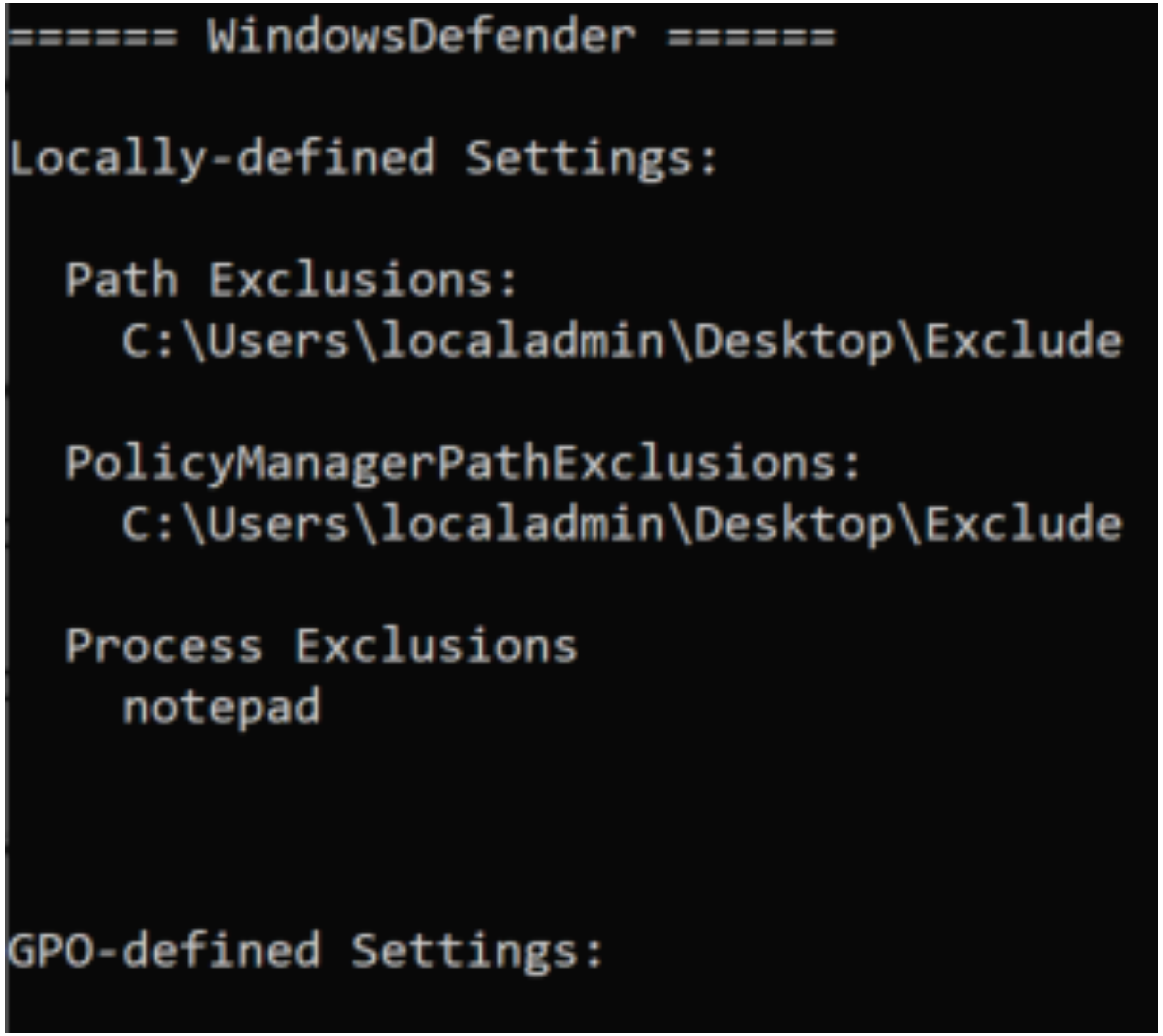
```
====== DotNet ======

  Installed CLR Versions
      2.0.50727
      4.0.30319

  Installed .NET Versions
      3.5.30729.4926
      4.8.03761

  Anti-Malware Scan Interface (AMSI)
      OS supports AMSI           : True
     .NET version support AMSI   : True
        [!] The highest .NET version is enrolled in AMSI!
        [*] You can invoke .NET version 3.5 to bypass AMSI.
```



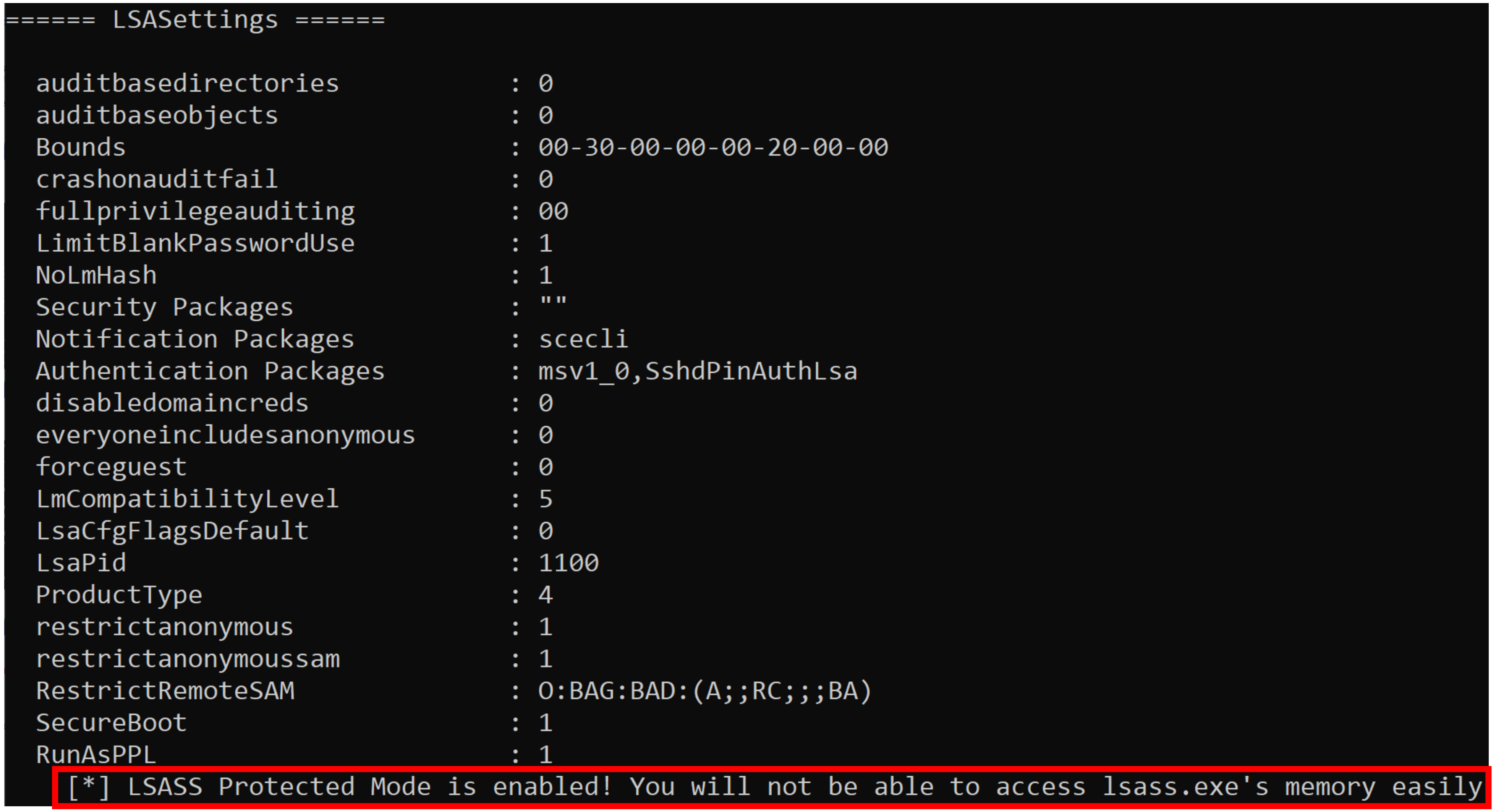
```
====== PowerShell ======

  Installed CLR Versions
      2.0.50727
      4.0.30319

  Installed PowerShell Versions
      2.0
      5.1.17763.1

  Transcription Logging Settings
      Enabled            : False
      Invocation Logging : False
      Log Directory      :

  Module Logging Settings
      Enabled             : True
      Logged Module Names :
        [!] You can do a PowerShell version downgrade to bypass the logging.

  Script Block Logging Settings
      Enabled            : True
      Invocation Logging : True
        [!] You can do a PowerShell version downgrade to bypass the logging.

  Anti-Malware Scan Interface (AMSI)
      OS Supports AMSI: True
        [!] You can do a PowerShell version downgrade to bypass AMSI.
```



```
====== InterestingProcesses ======

    Category     : defensive
    Name         : MsMpEng.exe
    Product      : Windows Defender AV
    ProcessID    : 2100
    Owner        :
    CommandLine  :

    Category     : interesting
    Name         : cmd.exe
    Product      : Command Prompt
    ProcessID    : 2956
    Owner        : DESKTOP-TO0RBV7\localadmin
    CommandLine  : "C:\Windows\system32\cmd.exe"
```



```
====== WindowsDefender ======

Locally-defined Settings:

  Path Exclusions:
    C:\Users\localadmin\Desktop\Exclude

  PolicyManagerPathExclusions:
    C:\Users\localadmin\Desktop\Exclude

  Process Exclusions
    notepad



GPO-defined Settings:
```


```
====== LSASettings ======

  auditbasedirectories                 : 0
  auditbaseobjects                     : 0
  Bounds                               : 00-30-00-00-00-20-00-00
  crashonauditfail                     : 0
  fullprivilegeauditing                : 00
  LimitBlankPasswordUse                : 1
  NoLmHash                             : 1
  Security Packages                    : ""
  Notification Packages                : scecli
  Authentication Packages              : msv1_0,SshdPinAuthLsa
  disabledomaincreds                   : 0
  everyoneincludesanonymous            : 0
  forceguest                           : 0
  LmCompatibilityLevel                 : 5
  LsaCfgFlagsDefault                   : 0
  LsaPid                               : 1100
  ProductType                          : 4
  restrictanonymous                    : 1
  restrictanonymoussam                 : 1
  RestrictRemoteSAM                    : O:BAG:BAD:(A;;RC;;;BA)
  SecureBoot                           : 1
  RunAsPPL                             : 1
    [*] LSASS Protected Mode is enabled! You will not be able to access lsass.exe's memory easily
```

Exploitation and Vulnerability Research Target Selection

AKA “Attack Surface Analysis”

Exploitation/Vuln Research Target Selection

- Anything that guides to specific things on a system to attack
- Common scenarios:
  - Privilege Escalation – through abuse of insecure configurations (think PowerUp), abuse of custom binaries, etc.
  - Lateral Movement/Domain Escalation – misconfigurations or vulnerable binaries might (often) lead to avenues to exploit other systems in the network
- **Big questions** – What things run elevated? How fast can we triage them? And are they remotely accessible?

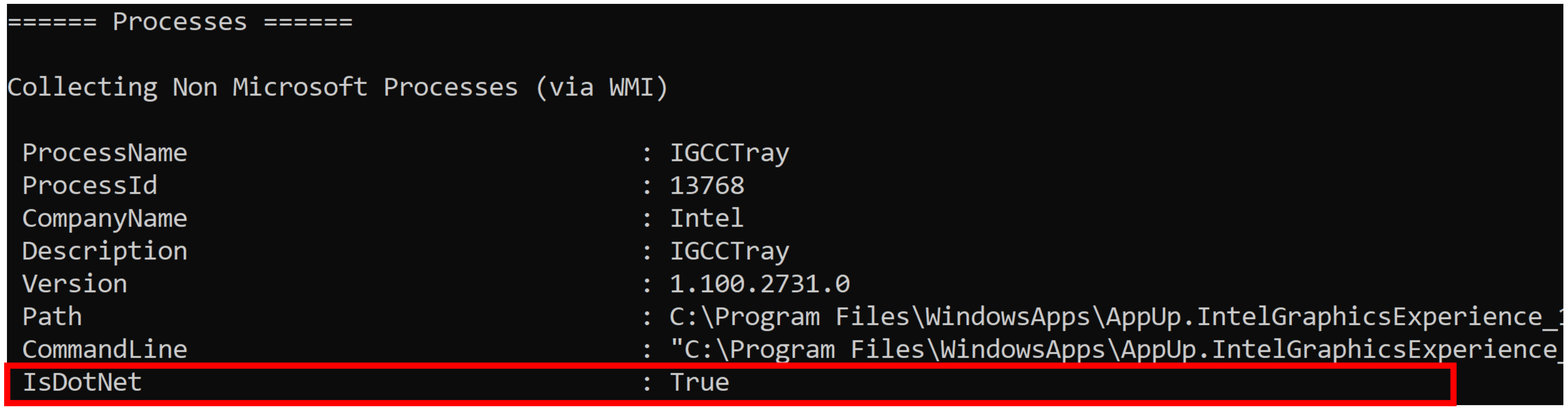
Reference: Relevant Modules

AutoRuns
EnvironmentPath
EnvironmentVariables
FileInfo
Fileinfo
Hotfixes
InstalledProducts
InterestingProcesses
MicrosoftUpdates
NamedPipes
OSInfo
reg
RPCMappedEndpoints
ScheduledTasks
Services
TcpConnections
UdpConnections

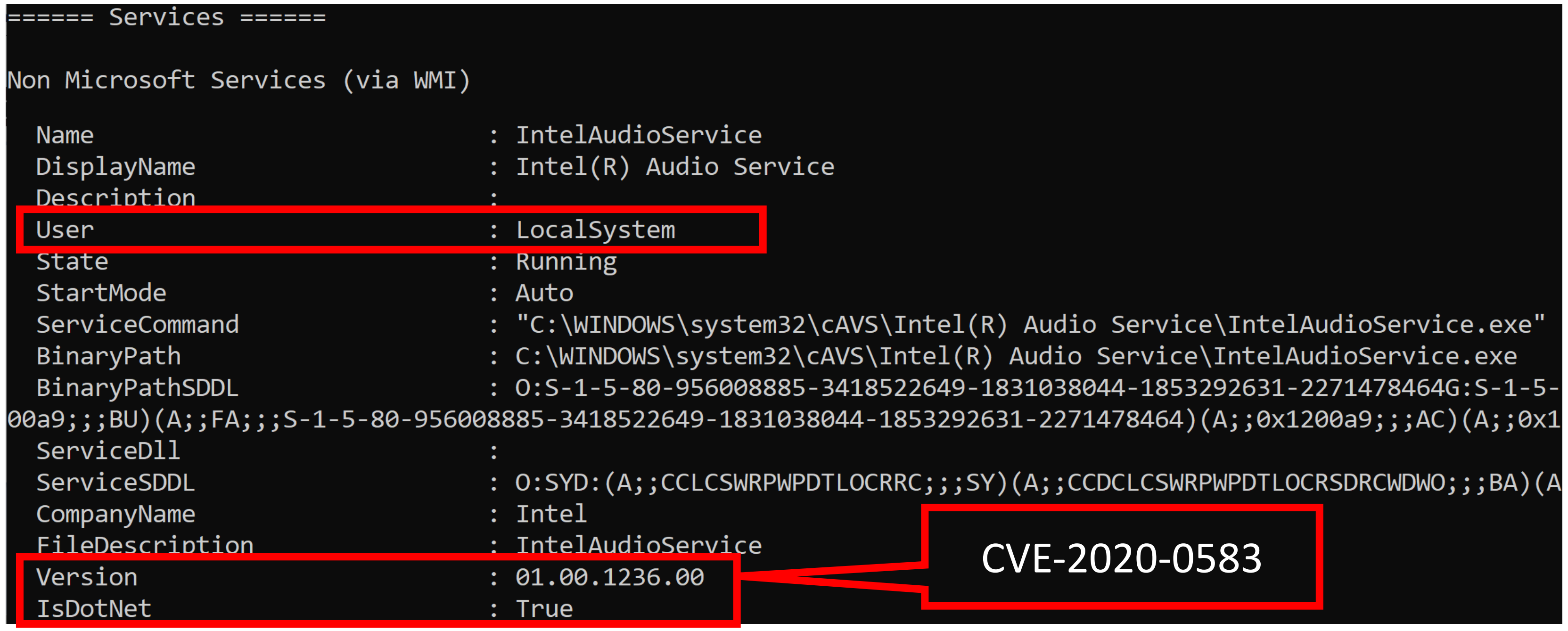
Services/Processes/ScheduledTasks

- We want to know:
  - What elevated programs/tasks are running on the system
  - Whether the program running is .NET or not (quicker for us to triage)
  - This often leads to privesc through pulling apart custom binaries

- The abuse of custom binaries is one of the most common ways we escalate privileges on a host (and in domains!)



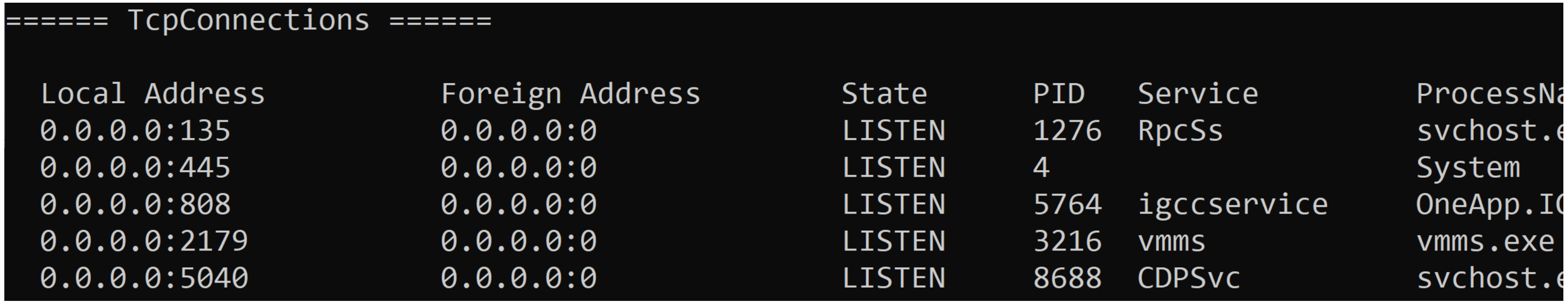
```
======= Processes =======

Collecting Non Microsoft Processes (via WMI)

 ProcessName                                : IGCCTray
 ProcessId                                  : 13768
 CompanyName                                : Intel
 Description                                : IGCCTray
 Version                                    : 1.100.2731.0
 Path                                       : C:\Program Files\WindowsApps\AppUp.IntelGraphicsExperience_1
 CommandLine                                : "C:\Program Files\WindowsApps\AppUp.IntelGraphicsExperience_
 IsDotNet                                   : True
```



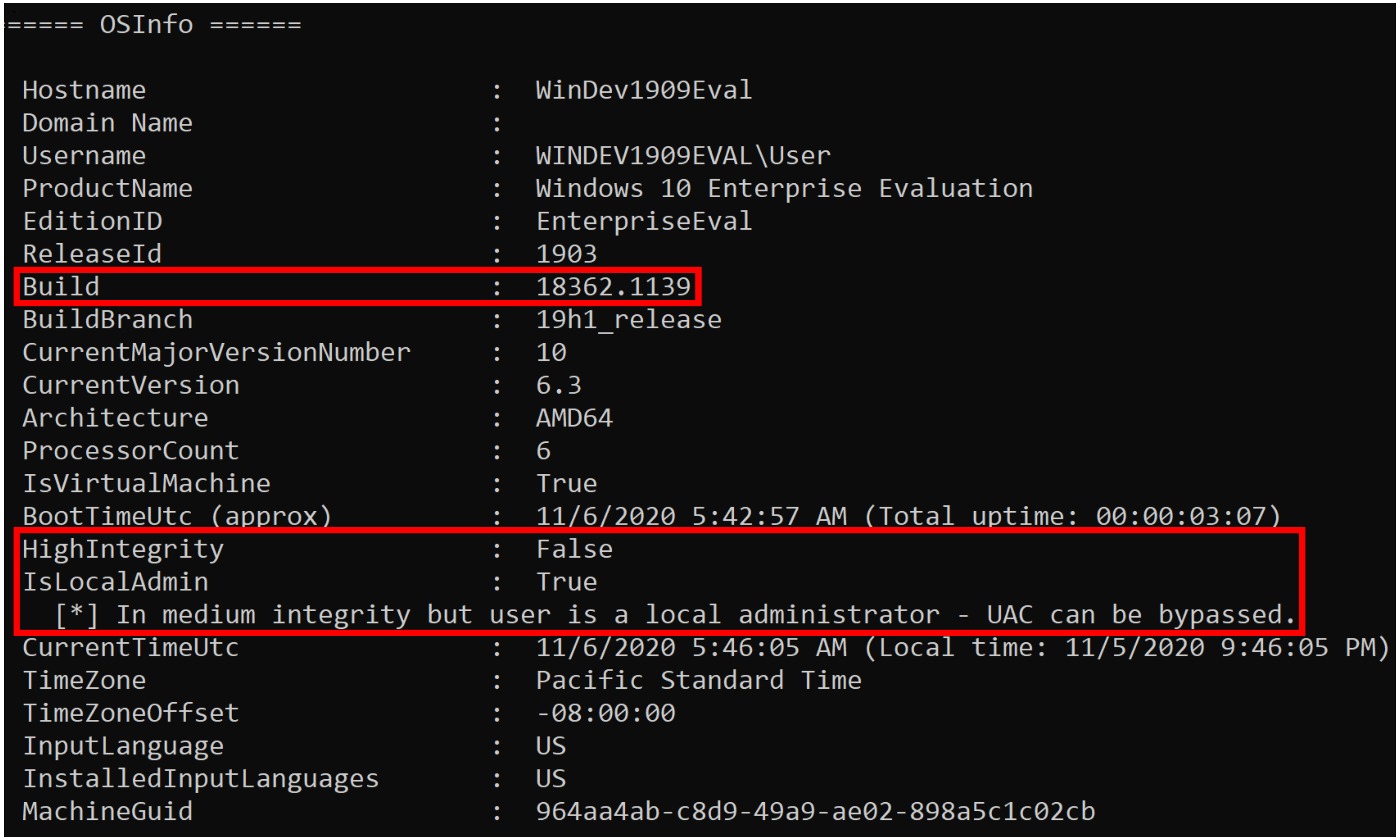
```
====== Services ======

Non Microsoft Services (via WMI)

  Name                                   : IntelAudioService
  DisplayName                            : Intel(R) Audio Service
  Description                            :
  User                                   : LocalSystem
  State                                  : Running
  StartMode                              : Auto
  ServiceCommand                         : "C:\WINDOWS\system32\cAVS\Intel(R) Audio Service\IntelAudioService.exe"
  BinaryPath                             : C:\WINDOWS\system32\cAVS\Intel(R) Audio Service\IntelAudioService.exe
  BinaryPathSDDL                         : O:S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464G:S-1-5-
00a9;;;BU)(A;;FA;;;S-1-5-80-956008885-3418522649-1831038044-1853292631-2271478464)(A;;0x1200a9;;;AC)(A;;0x1
  ServiceDll                             :
  ServiceSDDL                            : O:SYD:(A;;CCLCSWRPWPDTLOCRRC;;;SY)(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A
  CompanyName                            : Intel
  FileDescription                        : IntelAudioService
  Version                                : 01.00.1236.00
  IsDotNet                               : True
```

TcpConnections/UdpConnections

- Tells us processes that listen for UDP/TCP connections, as well as the service associated with the process (if applicable)
  - If it's listening on all interfaces (0.0.0.0), potential candidate for RCE
  - If it's bound locally (127.0.0.1) or on all interfaces, and running as SYSTEM, potential target for local privesc



```
====== TcpConnections ======

  Local Address          Foreign Address        State      PID   Service         ProcessNa
  0.0.0.0:135            0.0.0.0:0              LISTEN     1276  RpcSs           svchost.e
  0.0.0.0:445            0.0.0.0:0              LISTEN     4                     System
  0.0.0.0:808            0.0.0.0:0              LISTEN     5764  igccservice     OneApp.IG
  0.0.0.0:2179           0.0.0.0:0              LISTEN     3216  vmms            vmms.exe
  0.0.0.0:5040           0.0.0.0:0              LISTEN     8688  CDPSvc          svchost.e
```



```
===== OSInfo ======

  Hostname                      :  WinDev1909Eval
  Domain Name                   :
  Username                      :  WINDEV1909EVAL\User
  ProductName                   :  Windows 10 Enterprise Evaluation
  EditionID                     :  EnterpriseEval
  ReleaseId                     :  1903
  Build                         :  18362.1139
  BuildBranch                   :  19h1_release
  CurrentMajorVersionNumber     :  10
  CurrentVersion                :  6.3
  Architecture                  :  AMD64
  ProcessorCount                :  6
  IsVirtualMachine              :  True
  BootTimeUtc (approx)          :  11/6/2020 5:42:57 AM (Total uptime: 00:00:03:07)
  HighIntegrity                 :  False
  IsLocalAdmin                  :  True
    [*] In medium integrity but user is a local administrator - UAC can be bypassed.
  CurrentTimeUtc                :  11/6/2020 5:46:05 AM (Local time: 11/5/2020 9:46:05 PM)
  TimeZone                      :  Pacific Standard Time
  TimeZoneOffset                :  -08:00:00
  InputLanguage                 :  US
  InstalledInputLanguages       :  US
  MachineGuid                   :  964aa4ab-c8d9-49a9-ae02-898a5c1c02cb
```

Credential Theft

What You’re All Here For, Right?

Credential Theft

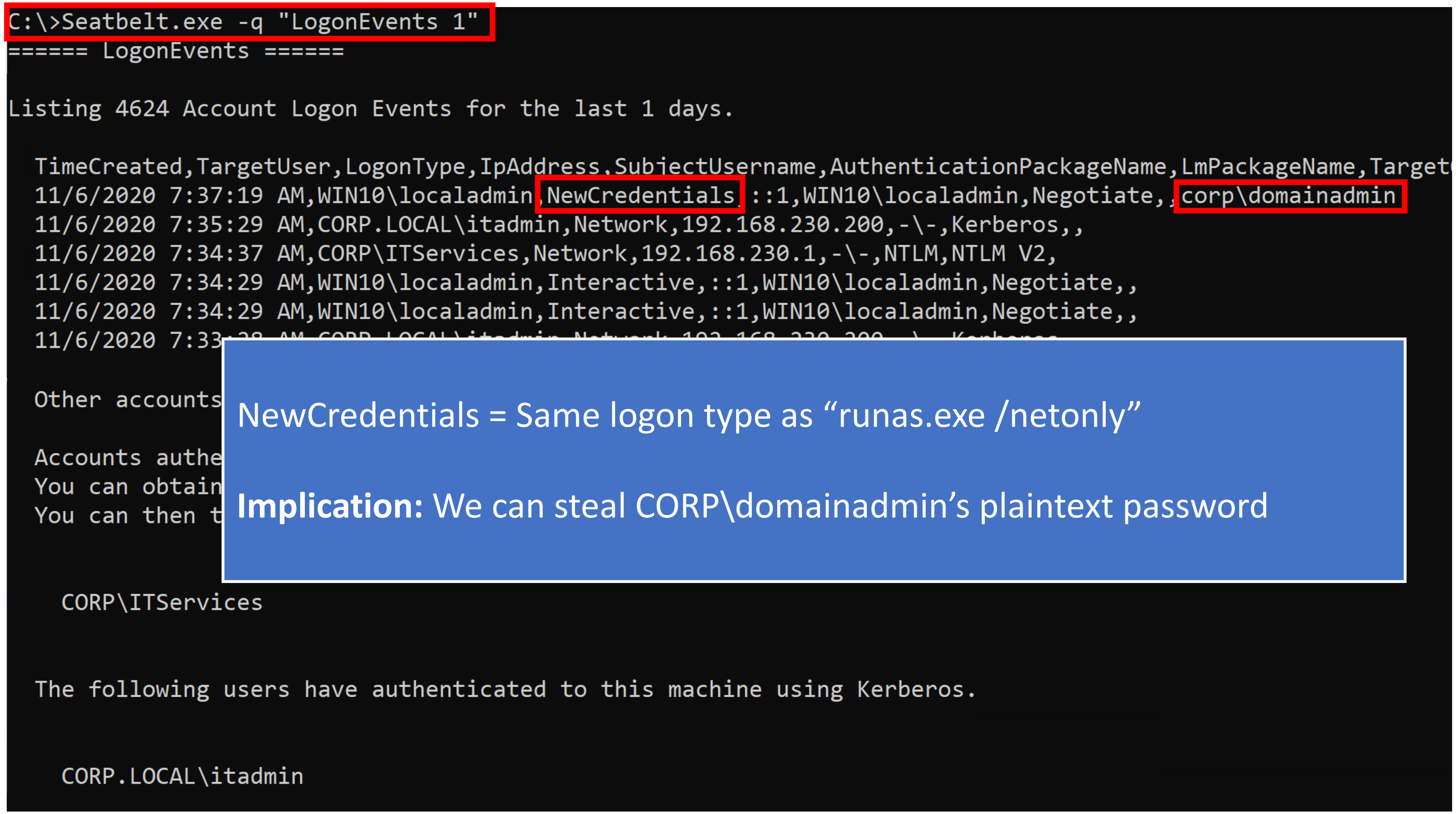
- Obviously an essential part of the attack cycle, but often over simplified
    - Remember: credentials, more than just sekurlsa::logonpasswords!
- For Seatbelt, this includes collection/modules that either:
    - Directly allow for the recovery of credentials
    - Somehow *affect* the recovery of credential material

Reference: Relevant Modules

CloudCredentials
CredEnum
CredGuard
dir/reg
DpapiMasterKeys
ExplicitLogonEvents
InterestingFiles
InterestingProcesses
LogonSessions
LSASettings
McAfeeSiteList
NTLMSettings
LogonEvents
PowerShellEvents
PowerShellHistory
ProcessCreationEvents
ProcessOwners
SearchIndex
SecPackageCreds
SecurityPackages
SysmonEvents
WindowsAutoLogon
WindowsCredentialFiles
WindowsVault

LogonEvents (Security Event ID 4624)

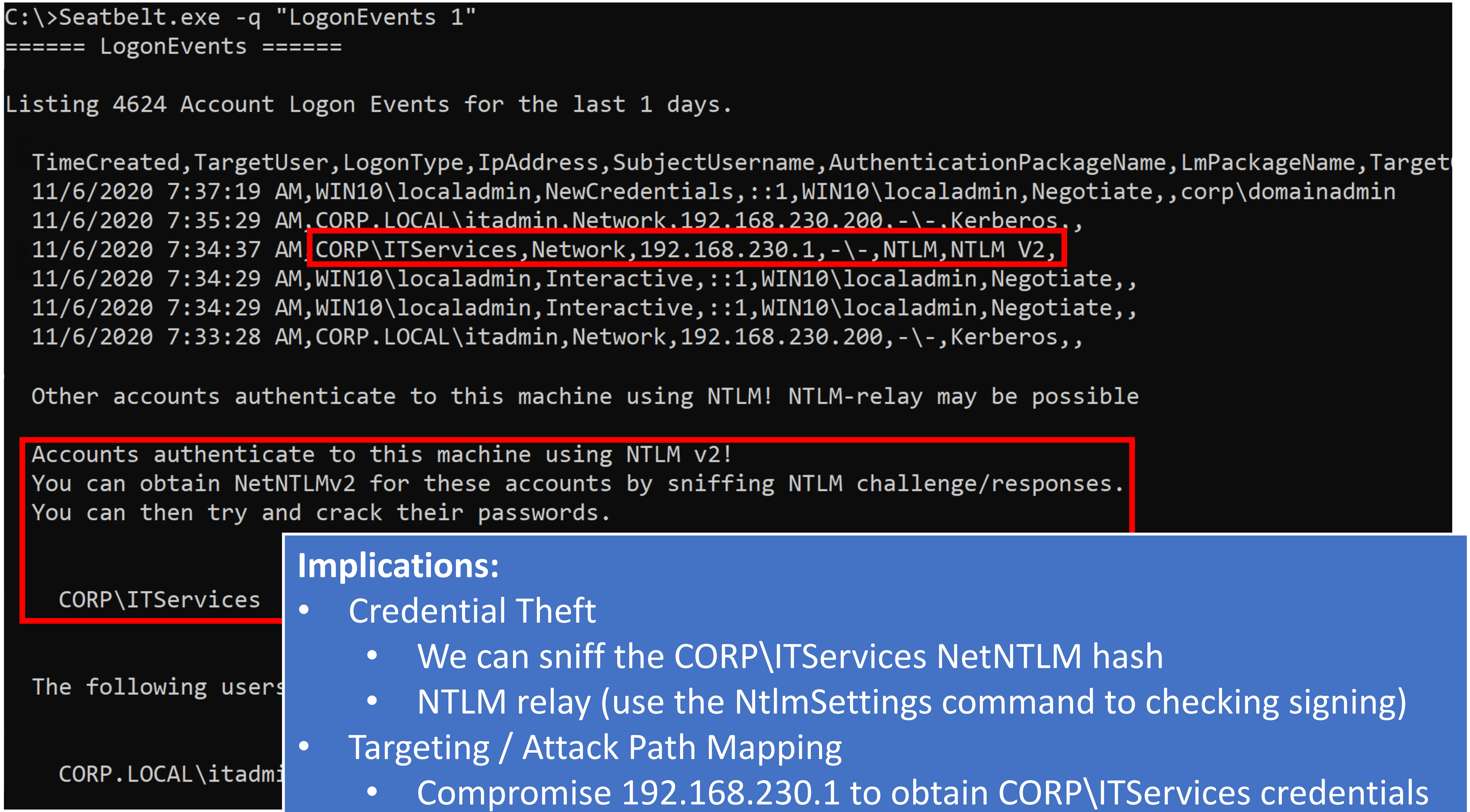
- What accounts perform inbound logins to this machine and when?
  - Examples: Nessus scanners, random IT accounts, SCCM push
- What protocol(s) do these accounts use when logging in?
  - If NTLM: sniff NetNTLMv1/v2 hashes or NTLM relay
- Where do these account login from?
  - Might give us information on where a sensitive admin is located
- Requires admin (Reads the Security event log)



```
C:\>Seatbelt.exe -q "LogonEvents 1"
====== LogonEvents ======

Listing 4624 Account Logon Events for the last 1 days.

  TimeCreated,TargetUser,LogonType,IpAddress,SubjectUsername,AuthenticationPackageName,LmPackageName,Target
  11/6/2020 7:37:19 AM,WIN10\localadmin,NewCredentials,::1,WIN10\localadmin,Negotiate,,corp\domainadmin
  11/6/2020 7:35:29 AM,CORP.LOCAL\itadmin,Network,192.168.230.200,-\-,Kerberos,,
  11/6/2020 7:34:37 AM,CORP\ITServices,Network,192.168.230.1,-\-,NTLM,NTLM V2,
  11/6/2020 7:34:29 AM,WIN10\localadmin,Interactive,::1,WIN10\localadmin,Negotiate,,
  11/6/2020 7:34:29 AM,WIN10\localadmin,Interactive,::1,WIN10\localadmin,Negotiate,,
  11/6/2020 7:33

  Other accounts

  Accounts authe
  You can obtain
  You can then t


    CORP\ITServices


  The following users have authenticated to this machine using Kerberos.


    CORP.LOCAL\itadmin
```

NewCredentials = Same logon type as “runas.exe /netonly”

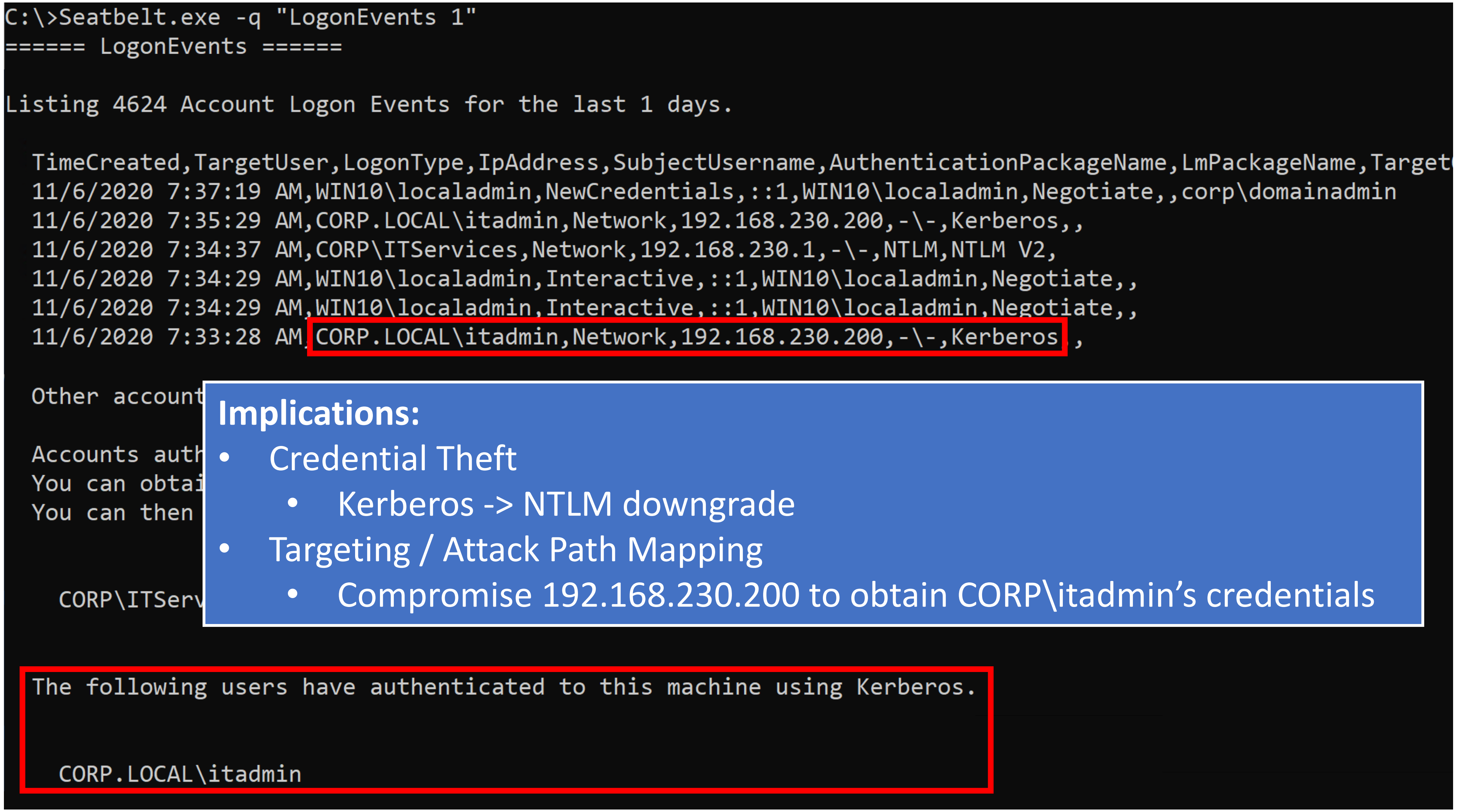
Implication: We can steal CORP\domainadmin’s plaintext password



```
C:\>Seatbelt.exe -q "LogonEvents 1"
====== LogonEvents ======

Listing 4624 Account Logon Events for the last 1 days.

  TimeCreated,TargetUser,LogonType,IpAddress,SubjectUsername,AuthenticationPackageName,LmPackageName,Target
  11/6/2020 7:37:19 AM,WIN10\localadmin,NewCredentials,::1,WIN10\localadmin,Negotiate,,corp\domainadmin
  11/6/2020 7:35:29 AM,CORP.LOCAL\itadmin,Network,192.168.230.200,-\-,Kerberos,,
  11/6/2020 7:34:37 AM,CORP\ITServices,Network,192.168.230.1,-\-,NTLM,NTLM V2,
  11/6/2020 7:34:29 AM,WIN10\localadmin,Interactive,::1,WIN10\localadmin,Negotiate,,
  11/6/2020 7:34:29 AM,WIN10\localadmin,Interactive,::1,WIN10\localadmin,Negotiate,,
  11/6/2020 7:33:28 AM,CORP.LOCAL\itadmin,Network,192.168.230.200,-\-,Kerberos,,

  Other accounts authenticate to this machine using NTLM! NTLM-relay may be possible

  Accounts authenticate to this machine using NTLM v2!
  You can obtain NetNTLMv2 for these accounts by sniffing NTLM challenge/responses.
  You can then try and crack their passwords.


    CORP\ITServices


  The following users


    CORP.LOCAL\itadmi
```

Implications:

- Credential Theft
  - We can sniff the CORP\ITServices NetNTLM hash
  - NTLM relay (use the NtlmSettings command to checking signing)
- Targeting / Attack Path Mapping
  - Compromise 192.168.230.1 to obtain CORP\ITServices credentials


```
C:\>Seatbelt.exe -q "LogonEvents 1"
====== LogonEvents ======

Listing 4624 Account Logon Events for the last 1 days.

  TimeCreated,TargetUser,LogonType,IpAddress,SubjectUsername,AuthenticationPackageName,LmPackageName,Target
  11/6/2020 7:37:19 AM,WIN10\localadmin,NewCredentials,::1,WIN10\localadmin,Negotiate,,corp\domainadmin
  11/6/2020 7:35:29 AM,CORP.LOCAL\itadmin,Network,192.168.230.200,-\-,Kerberos,,
  11/6/2020 7:34:37 AM,CORP\ITServices,Network,192.168.230.1,-\-,NTLM,NTLM V2,
  11/6/2020 7:34:29 AM,WIN10\localadmin,Interactive,::1,WIN10\localadmin,Negotiate,,
  11/6/2020 7:34:29 AM,WIN10\localadmin,Interactive,::1,WIN10\localadmin,Negotiate,,
  11/6/2020 7:33:28 AM,CORP.LOCAL\itadmin,Network,192.168.230.200,-\-,Kerberos,,

  Other account

  Accounts auth
  You can obtai
  You can then


    CORP\ITServ


  The following users have authenticated to this machine using Kerberos.


    CORP.LOCAL\itadmin
```

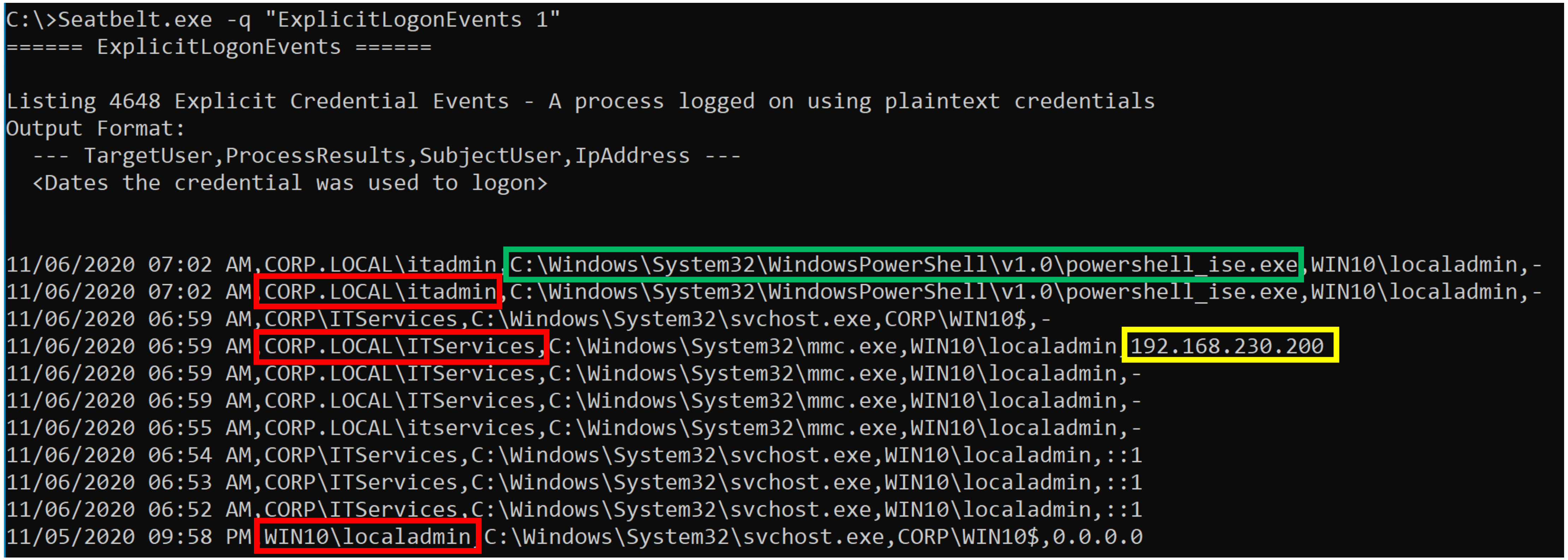
Implications:

- Credential Theft
  - Kerberos -> NTLM downgrade
- Targeting / Attack Path Mapping
  - Compromise 192.168.230.200 to obtain CORP\itadmin's credentials

ExplicitLogonEvents (Security Event 4648)

- When a program logs on as a user using a plaintext credential, map that back to the program that triggered the logon event.
  - *Somehow* that plaintext credential is being used by that program and is ALWAYS recoverable.
- You can use the timestamp to determine if this occurs at some regular internal
  - e.g. might be a scheduled task or some background task an installed program performs at a regular interval

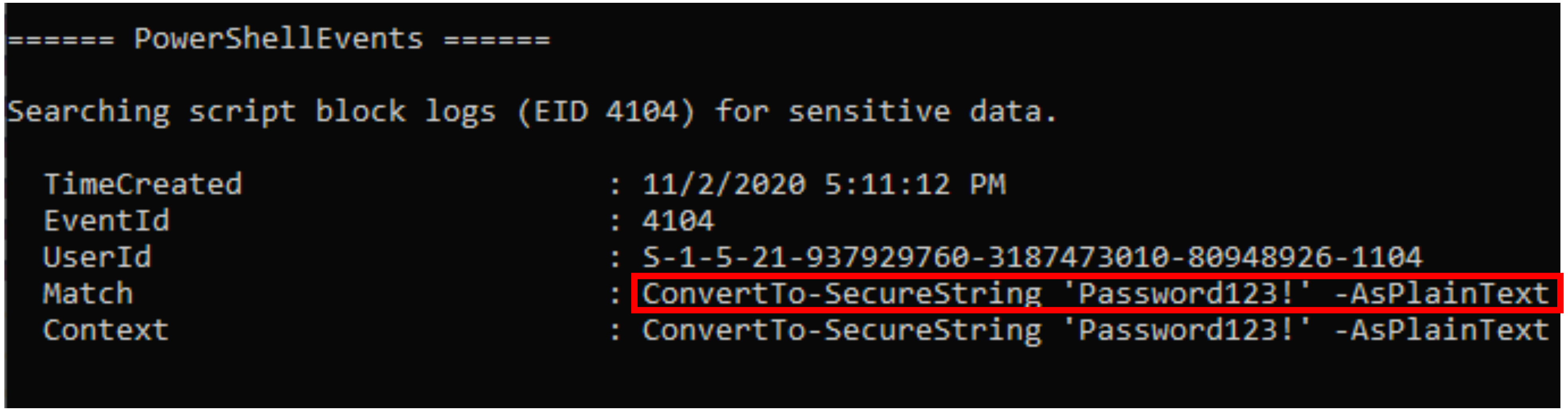
ExplicitLogonEvents



```
C:\>Seatbelt.exe -q "ExplicitLogonEvents 1"
====== ExplicitLogonEvents ======

Listing 4648 Explicit Credential Events - A process logged on using plaintext credentials
Output Format:
  --- TargetUser,ProcessResults,SubjectUser,IpAddress ---
  <Dates the credential was used to logon>


11/06/2020 07:02 AM,CORP.LOCAL\itadmin,C:\Windows\System32\WindowsPowerShell\v1.0\powershell_ise.exe,WIN10\localadmin,-
11/06/2020 07:02 AM,CORP.LOCAL\itadmin,C:\Windows\System32\WindowsPowerShell\v1.0\powershell_ise.exe,WIN10\localadmin,-
11/06/2020 06:59 AM,CORP\ITServices,C:\Windows\System32\svchost.exe,CORP\WIN10$,-
11/06/2020 06:59 AM,CORP.LOCAL\ITServices,C:\Windows\System32\mmc.exe,WIN10\localadmin,192.168.230.200
11/06/2020 06:59 AM,CORP.LOCAL\ITServices,C:\Windows\System32\mmc.exe,WIN10\localadmin,-
11/06/2020 06:59 AM,CORP.LOCAL\ITServices,C:\Windows\System32\mmc.exe,WIN10\localadmin,-
11/06/2020 06:55 AM,CORP.LOCAL\itservices,C:\Windows\System32\mmc.exe,WIN10\localadmin,-
11/06/2020 06:54 AM,CORP\ITServices,C:\Windows\System32\svchost.exe,WIN10\localadmin,::1
11/06/2020 06:53 AM,CORP\ITServices,C:\Windows\System32\svchost.exe,WIN10\localadmin,::1
11/06/2020 06:52 AM,CORP\ITServices,C:\Windows\System32\svchost.exe,WIN10\localadmin,::1
11/05/2020 09:58 PM,WIN10\localadmin,C:\Windows\System32\svchost.exe,CORP\WIN10$,0.0.0.0
```

Scraping "Sensitive" Event Logs

- PowerShellEvents/SysmonEvents/ProcessCreationEvents
  - All of these modules are run through a common set of regexes built to detect the leakage of passwords on command line binaries (psexec/net/etc.)
  - PowerShellEvents is readable from a non-admin context!

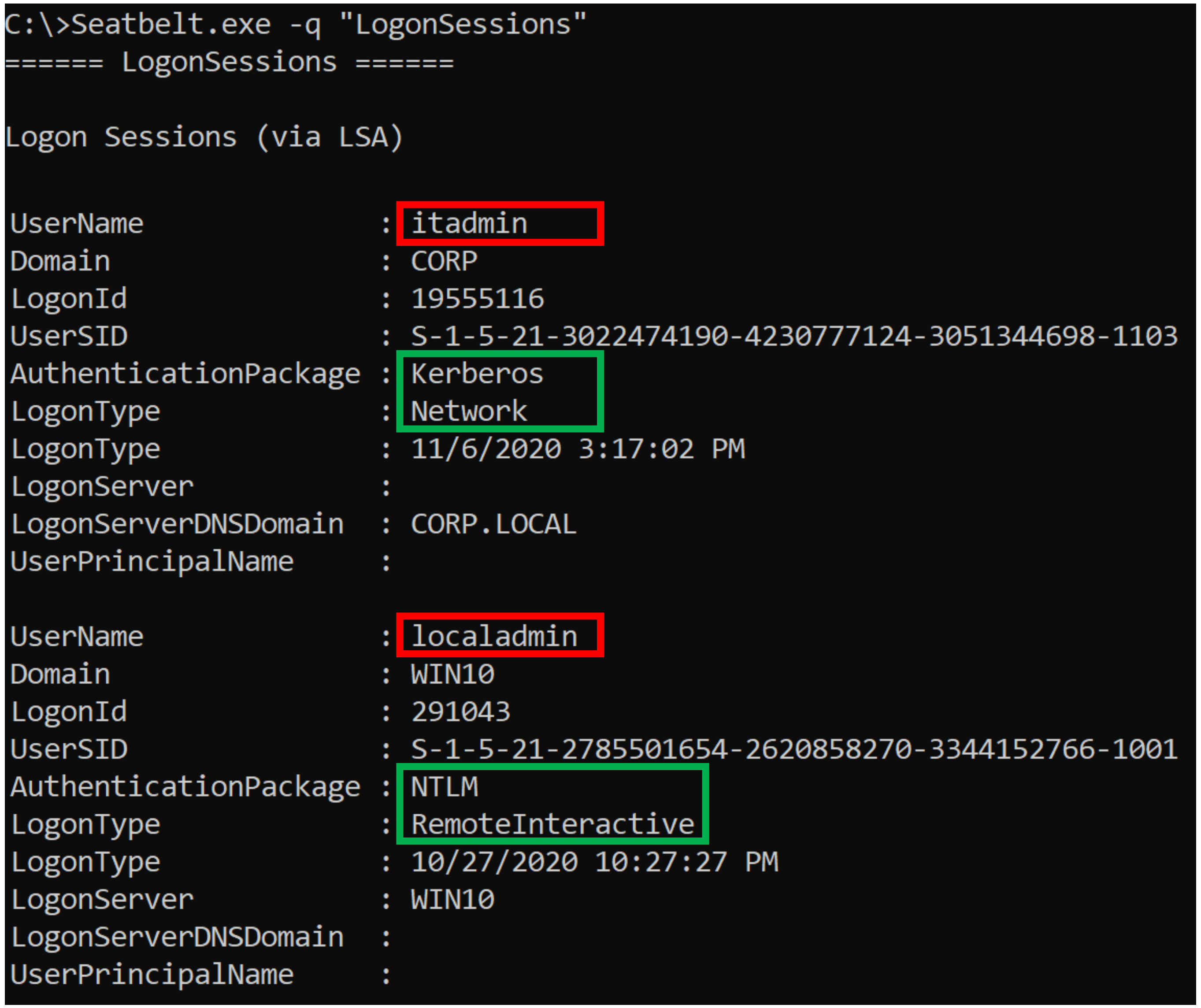


```
====== PowerShellEvents ======

Searching script block logs (EID 4104) for sensitive data.

  TimeCreated                      : 11/2/2020 5:11:12 PM
  EventId                          : 4104
  UserId                           : S-1-5-21-937929760-3187473010-80948926-1104
  Match                            : ConvertTo-SecureString 'Password123!' -AsPlainText
  Context                          : ConvertTo-SecureString 'Password123!' -AsPlainText
```

LogonSessions

- Lets us know who is logged into a machine, when they logged on, and what they're logon session type is:
  - Network logon sessions: credentials (usually) not in memory
  - Non-network logon sessions: credentials often in memory

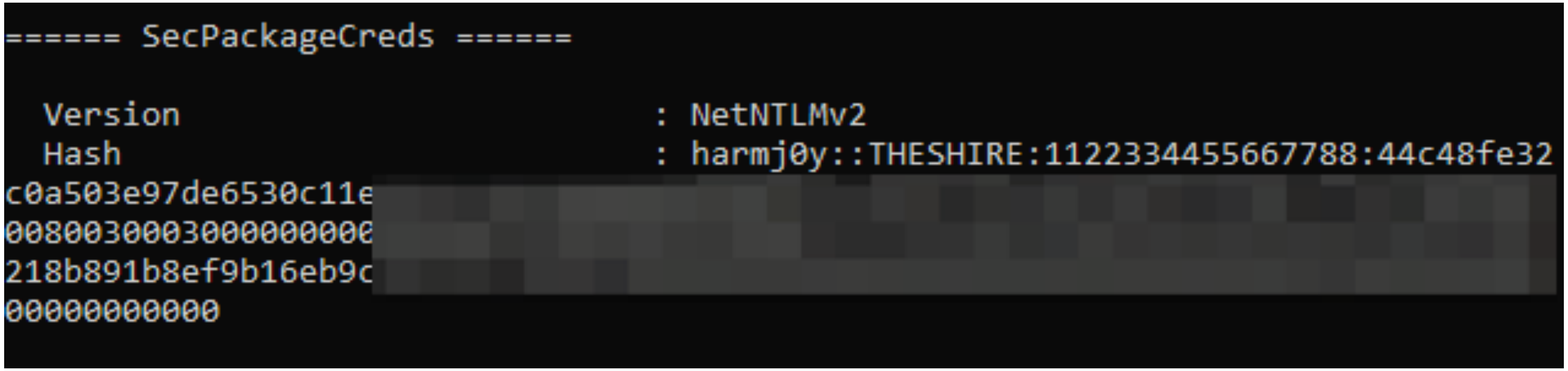
- Can be run against a remote system to see if credential-focused lateral movement is even worth it!



```
C:\>Seatbelt.exe -q "LogonSessions"
====== LogonSessions ======

Logon Sessions (via LSA)

 UserName              : itadmin
 Domain                : CORP
 LogonId               : 19555116
 UserSID               : S-1-5-21-3022474190-4230777124-3051344698-1103
 AuthenticationPackage : Kerberos
 LogonType             : Network
 LogonType             : 11/6/2020 3:17:02 PM
 LogonServer           :
 LogonServerDNSDomain  : CORP.LOCAL
 UserPrincipalName     :

 UserName              : localadmin
 Domain                : WIN10
 LogonId               : 291043
 UserSID               : S-1-5-21-2785501654-2620858270-3344152766-1001
 AuthenticationPackage : NTLM
 LogonType             : RemoteInteractive
 LogonType             : 10/27/2020 10:27:27 PM
 LogonServer           : WIN10
 LogonServerDNSDomain  :
 UserPrincipalName     :
```

SecPackageCreds

- Obtains credentials from security packages.
- Currently extracts NetNTLMv1 or NetNTLMv2 hashes of the current user from the NTLM package right now (a la Internal-Monologue)



```
====== SecPackageCreds ======

  Version                              : NetNTLMv2
  Hash                                 : harmj0y::THESHIRE:1122334455667788:44c48fe32
c0a503e97de6530c11e[redacted]
0080030003000000000
218b891b8ef9b16eb9c
00000000000
```

User/System Behavioral Baselining

Figuring Out “Normal”

Reference: Relevant Modules

ChromiumBookmarks
ChromiumHistory
ChromiumPresence
ExplorerMRUs
ExplorerRunCommands
FileZilla
FirefoxHistory
FirefoxPresence

Idletime
IEFavorites
IETabs
IEUrls
LocalGroups
OfficeMRUs
OutlookDownloads
PuttyHostKeys
PuttySessions
RDCManFiles

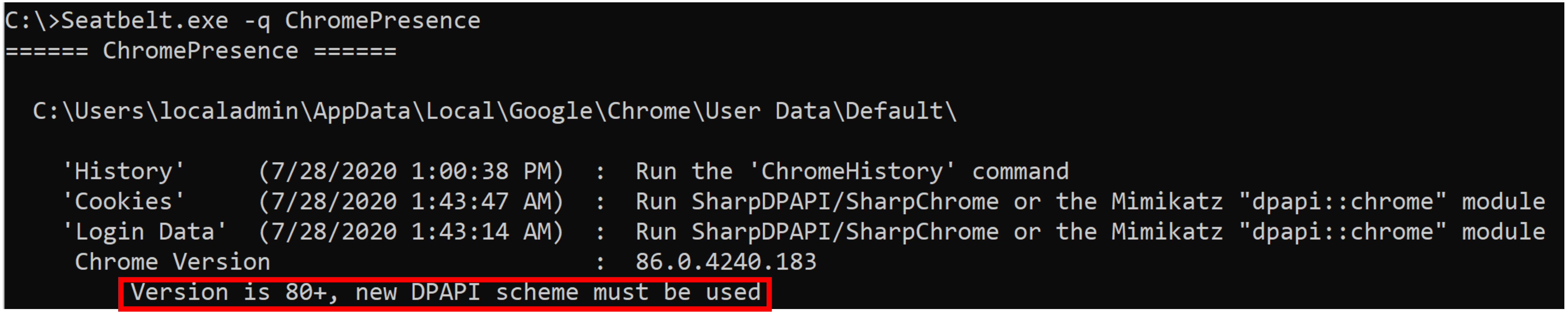
RDPSavedConnections
RDPSessions
RecycleBin
SlackDownloads
SlackPresence
SlackWorkspaces
SuperPutty
TokenGroups
TokenPrivileges

Behavioral Baselining

- We want to get a sense of what users actually ***use*** this system for
- Installed versions of Browsers?
  - Where do they navigate to with them? (bookmarks/tabs/history/etc.)
  - Any credentials? (passwords/cookies)
- Remote administration tools? (ssh/ftp clients/RDP/etc.)
- Recent documents/commands/paths?
  - Office/Explorer MRUs, recent documents, etc.

ChromiumPresence/History/Bookmarks

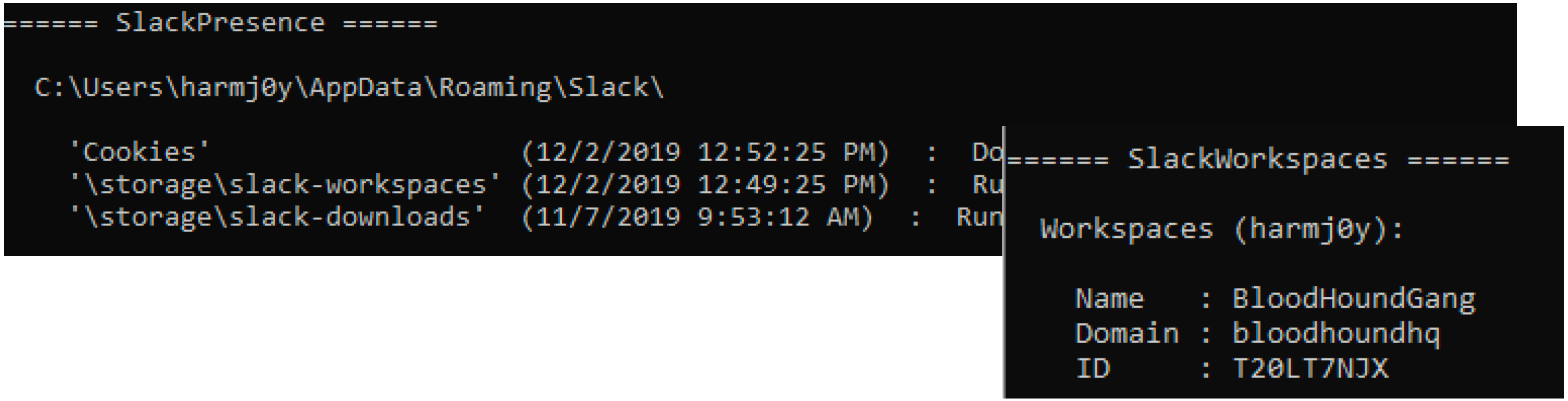
- Targets all Chromium-based browsers (Chrome/Edge/Brave/Opera)
  - Also works remotely!



```
C:\>Seatbelt.exe -q ChromePresence
====== ChromePresence ======

  C:\Users\localadmin\AppData\Local\Google\Chrome\User Data\Default\

    'History'     (7/28/2020 1:00:38 PM)  :  Run the 'ChromeHistory' command
    'Cookies'     (7/28/2020 1:43:47 AM)  :  Run SharpDPAPI/SharpChrome or the Mimikatz "dpapi::chrome" module
    'Login Data'  (7/28/2020 1:43:14 AM)  :  Run SharpDPAPI/SharpChrome or the Mimikatz "dpapi::chrome" module
     Chrome Version                       :  86.0.4240.183
         Version is 80+, new DPAPI scheme must be used
```

SlackPresence/Downloads/Workspaces

- If you have access to Slack/storage/slack-workspaces and Slack/Cookies, you can easily clone someone's Slack access!
  - https://posts.specterops.io/abusing-slack-for-offensive-operations-2343237b9282

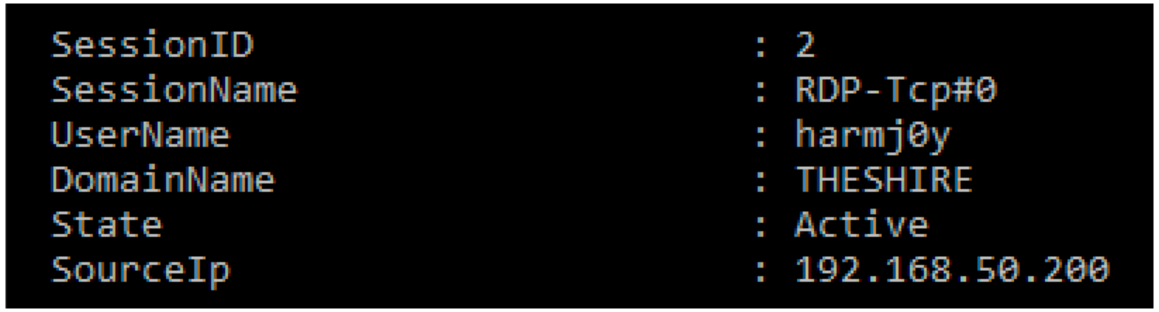


```
====== SlackPresence ======

  C:\Users\harmj0y\AppData\Roaming\Slack\

    'Cookies'                    (12/2/2019 12:52:25 PM)   :  Do
    '\storage\slack-workspaces'  (12/2/2019 12:49:25 PM)   :  Ru
    '\storage\slack-downloads'   (11/7/2019 9:53:12 AM)   :  Run
```

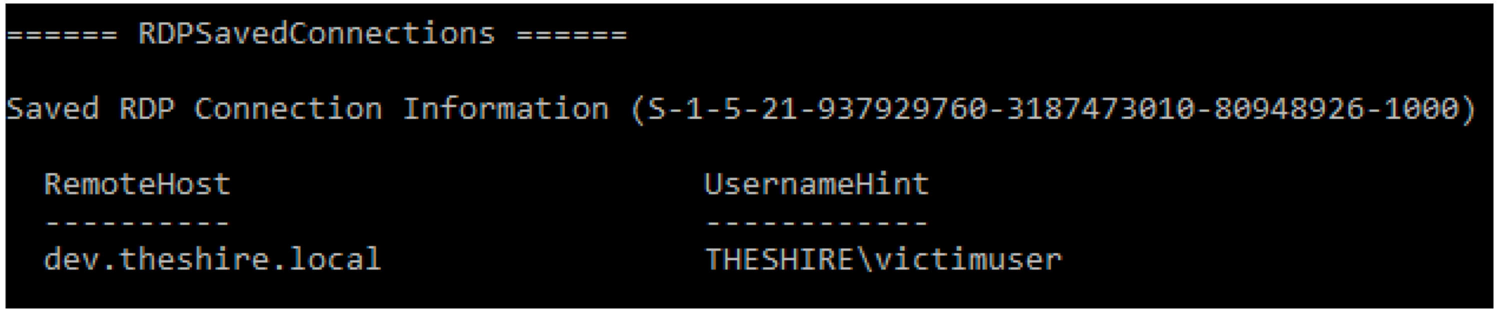
```
====== SlackWorkspaces ======

  Workspaces (harmj0y):

    Name    : BloodHoundGang
    Domain  : bloodhoundhq
    ID      : T20LT7NJX
```

RDPSavedConnections/RDPSessions/RDCManFiles/ LogonSessions

- Is it "normal" for RDP to be used in the environment?
  - Are there saved credentials we can abuse?
  - Are there periodic logon sessions we can steal creds from?



```
SessionID                              : 2
SessionName                            : RDP-Tcp#0
UserName                               : harmj0y
DomainName                             : THESHIRE
State                                  : Active
SourceIp                               : 192.168.50.200
```



```
====== RDPSavedConnections ======

Saved RDP Connection Information (S-1-5-21-937929760-3187473010-80948926-1000)

  RemoteHost                         UsernameHint
  ----------                         ------------
  dev.theshire.local                 THESHIRE\victimuser
```

Technique Selection

Examples and a Walkthrough

Example: Persistence

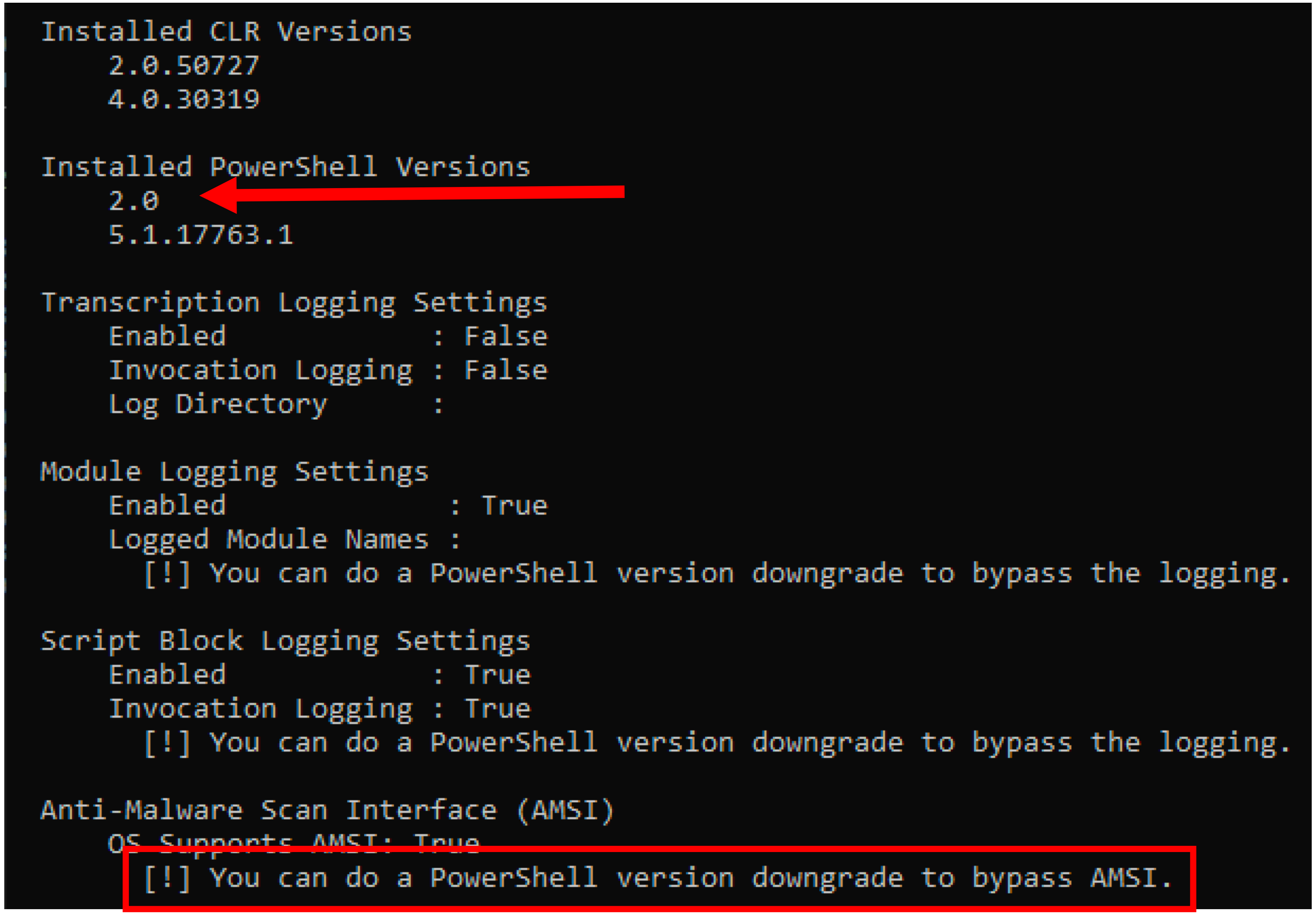
- Do we even need to drop persistence?
  - LastShutdown – last time the system was shutdown
  - PoweredOnEvents – reboot timings for the last week
- What runs normally that we can hijack (or .dll sideload)?
  - WMIEventFilter/WMIFilterBinding/WMIEventConsumer
  - AutoRuns
  - ScheduledTasks
  - Services

Example: Lateral Movement

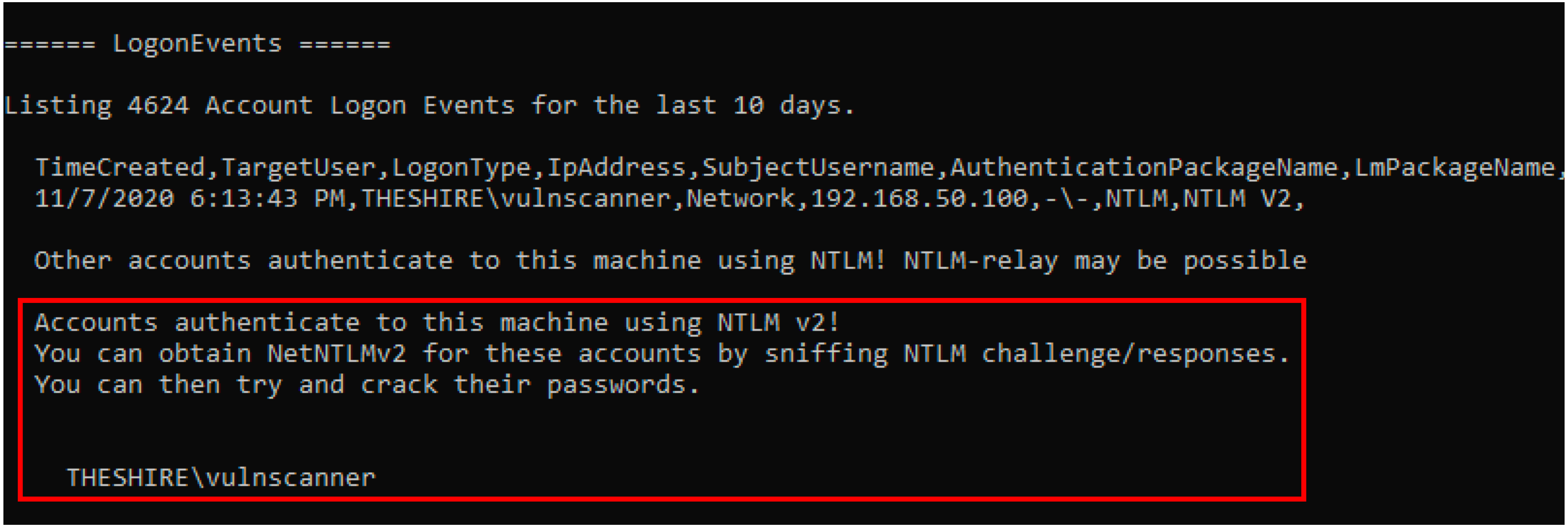
- Are there any specific firewall port restrictions? (**WindowsFirewall**)
- All existing "attack surface" analysis, but run/applied remotely
- What runs normally that we can hijack (or .dll sideload)?
  - Same as persistence (**ScheduledTasks**, WMI, etc.)
- Current EDR products (**InterestingProcesses**)
- **UAC** – affects what local accounts could be used for lateral movement

Full Walkthrough

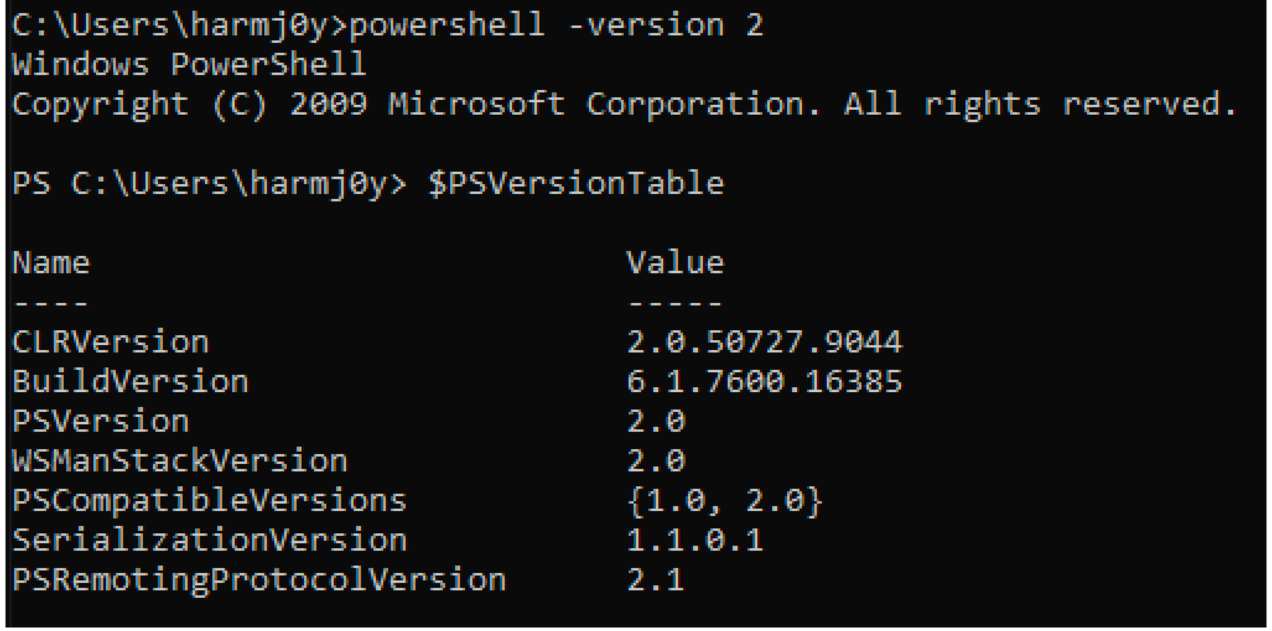
- The following few slides will talk through escalating on a system and moving laterally to a second
- We'll talk through individual data source results, and how they affect the next steps of the attack process
  - **Big point –** we're hoping to talk through our thought process, and how the new data sources affect the decisions we make moving to the next step in an attack chain



```
Installed CLR Versions
    2.0.50727
    4.0.30319

Installed PowerShell Versions
    2.0
    5.1.17763.1

Transcription Logging Settings
    Enabled             : False
    Invocation Logging  : False
    Log Directory       :

Module Logging Settings
    Enabled               : True
    Logged Module Names   :
      [!] You can do a PowerShell version downgrade to bypass the logging.

Script Block Logging Settings
    Enabled             : True
    Invocation Logging  : True
      [!] You can do a PowerShell version downgrade to bypass the logging.

Anti-Malware Scan Interface (AMSI)
    OS Supports AMSI: True
      [!] You can do a PowerShell version downgrade to bypass AMSI.
```

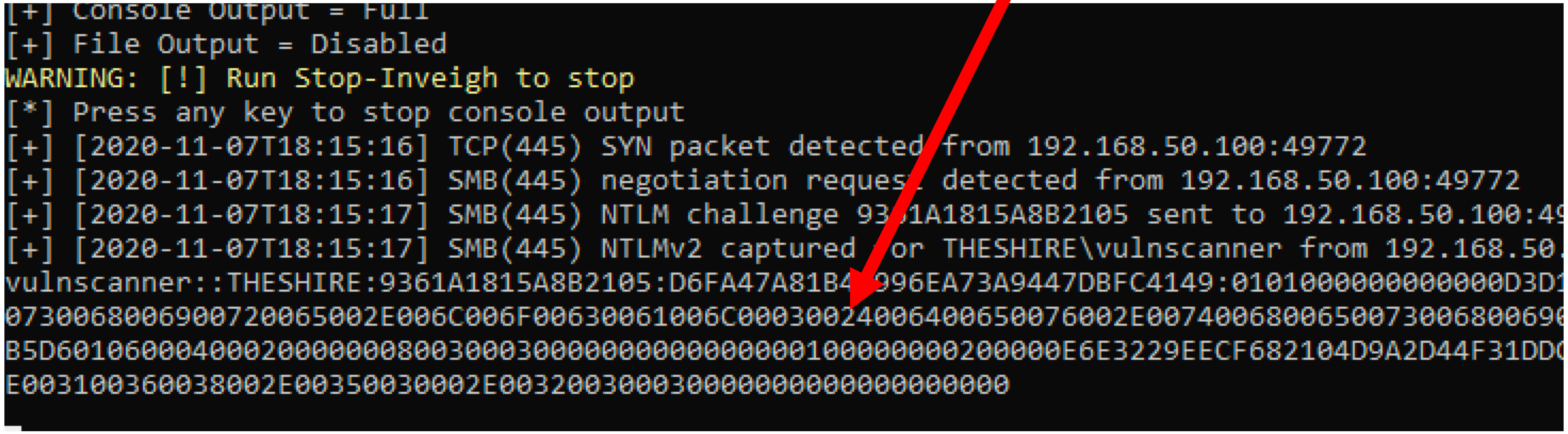


```
====== LogonEvents ======

Listing 4624 Account Logon Events for the last 10 days.

  TimeCreated,TargetUser,LogonType,IpAddress,SubjectUsername,AuthenticationPackageName,LmPackageName,
  11/7/2020 6:13:43 PM,THESHIRE\vulnscanner,Network,192.168.50.100,-\-,NTLM,NTLM V2,

  Other accounts authenticate to this machine using NTLM! NTLM-relay may be possible

  Accounts authenticate to this machine using NTLM v2!
  You can obtain NetNTLMv2 for these accounts by sniffing NTLM challenge/responses.
  You can then try and crack their passwords.


    THESHIRE\vulnscanner
```

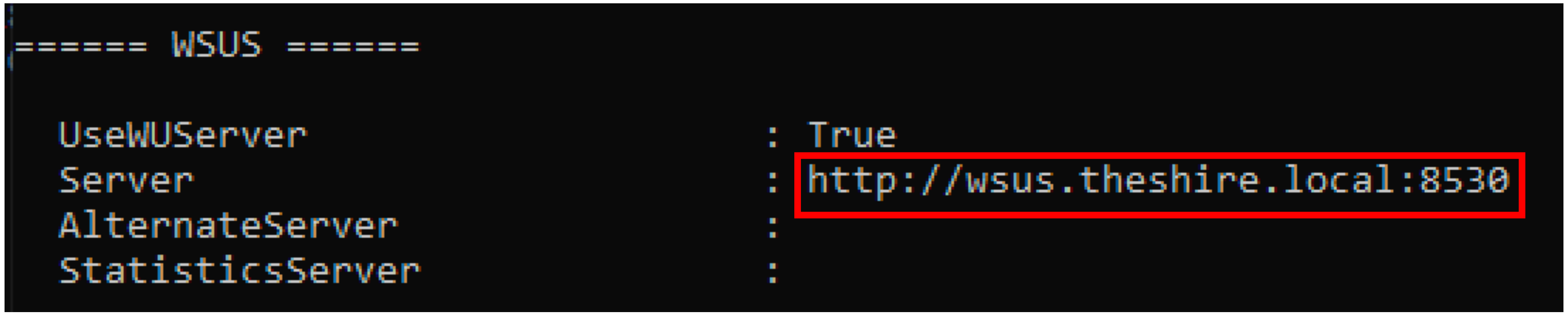


```
C:\Users\harmj0y>powershell -version 2
Windows PowerShell
Copyright (C) 2009 Microsoft Corporation. All rights reserved.

PS C:\Users\harmj0y> $PSVersionTable

Name                           Value
----                           -----
CLRVersion                     2.0.50727.9044
BuildVersion                   6.1.7600.16385
PSVersion                      2.0
WSManStackVersion              2.0
PSCompatibleVersions           {1.0, 2.0}
SerializationVersion           1.1.0.1
PSRemotingProtocolVersion      2.1
```

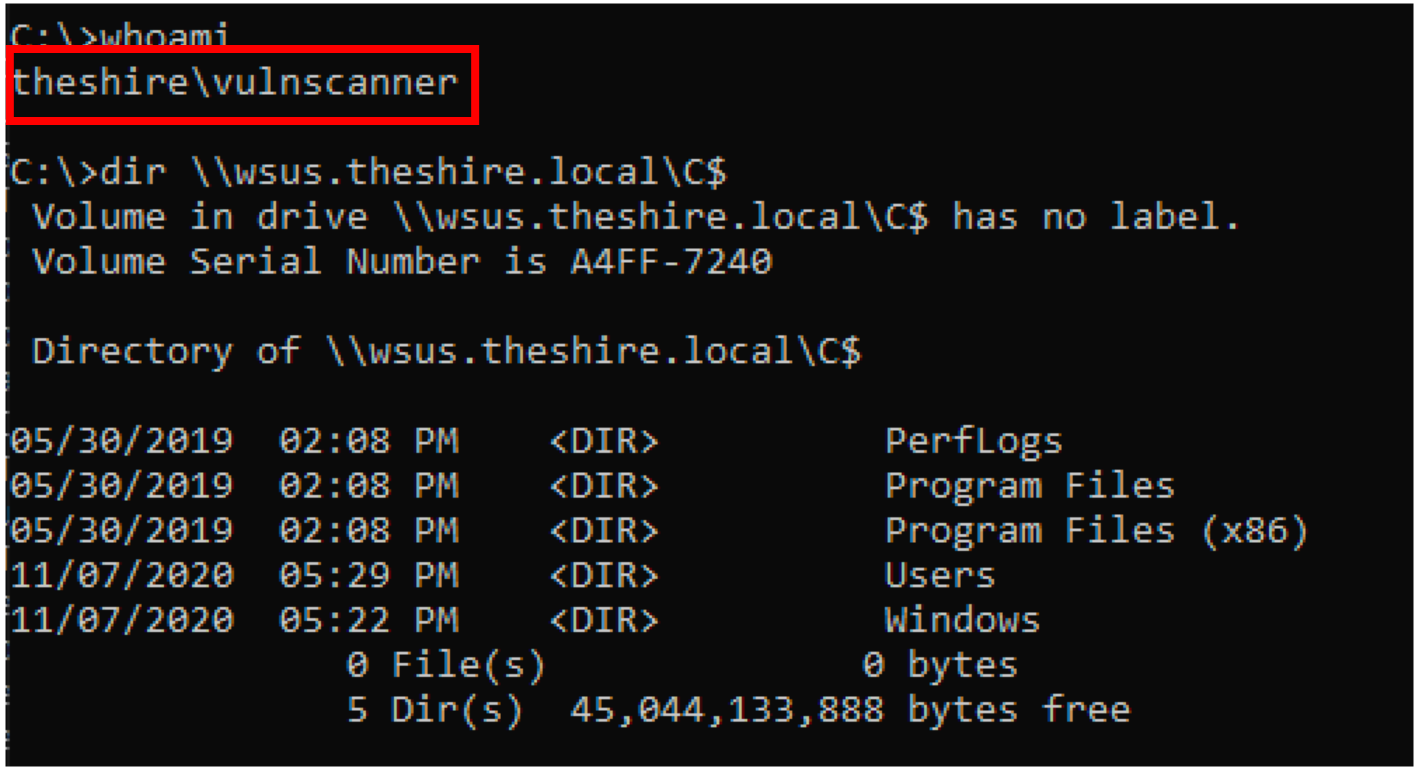
Crackable!



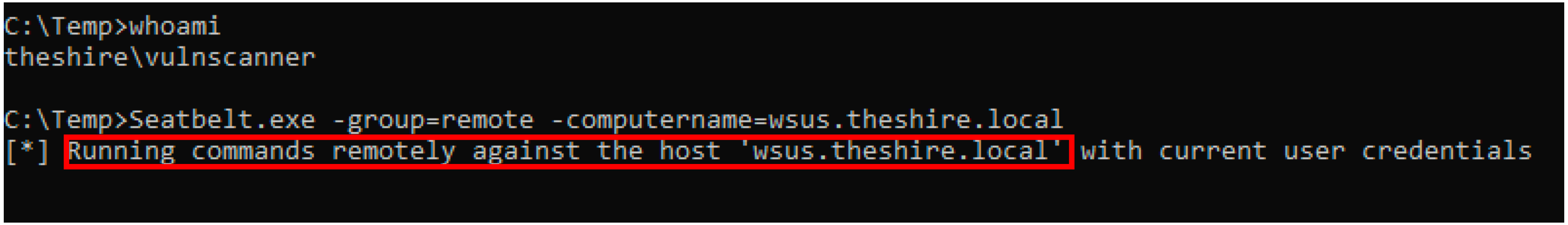
```
[+] Console Output = Full
[+] File Output = Disabled
WARNING: [!] Run Stop-Inveigh to stop
[*] Press any key to stop console output
[+] [2020-11-07T18:15:16] TCP(445) SYN packet detected from 192.168.50.100:49772
[+] [2020-11-07T18:15:16] SMB(445) negotiation request detected from 192.168.50.100:49772
[+] [2020-11-07T18:15:17] SMB(445) NTLM challenge 9361A1815A8B2105 sent to 192.168.50.100:49
[+] [2020-11-07T18:15:17] SMB(445) NTLMv2 captured for THESHIRE\vulnscanner from 192.168.50.
vulnscanner::THESHIRE:9361A1815A8B2105:D6FA47A81B4[illegible]996EA73A9447DBFC4149:0101000000000000D3D1
0730068006900720065002E006C006F00630061006C0003002400640065076002E00740068006500730068006900
B5D6010600040002000000080030003000000000000000010000000200000E6E3229EECF682104D9A2D44F31DDC
E0031003600380002E00350030002E0032003000300000000000000000000
```



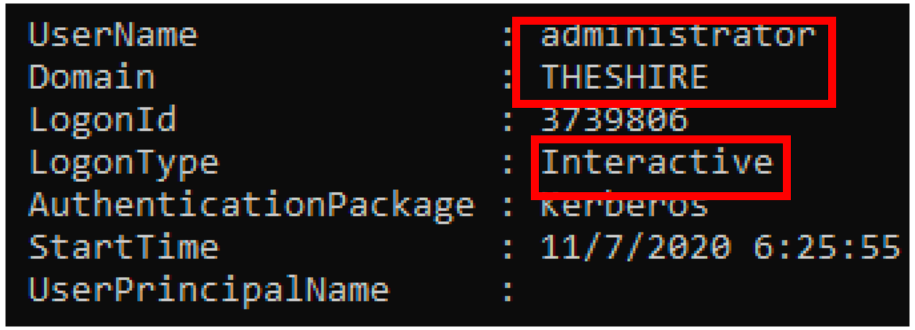
```
====== WSUS ======

  UseWUServer                      : True
  Server                           : http://wsus.theshire.local:8530
  AlternateServer                  :
  StatisticsServer                 :
```



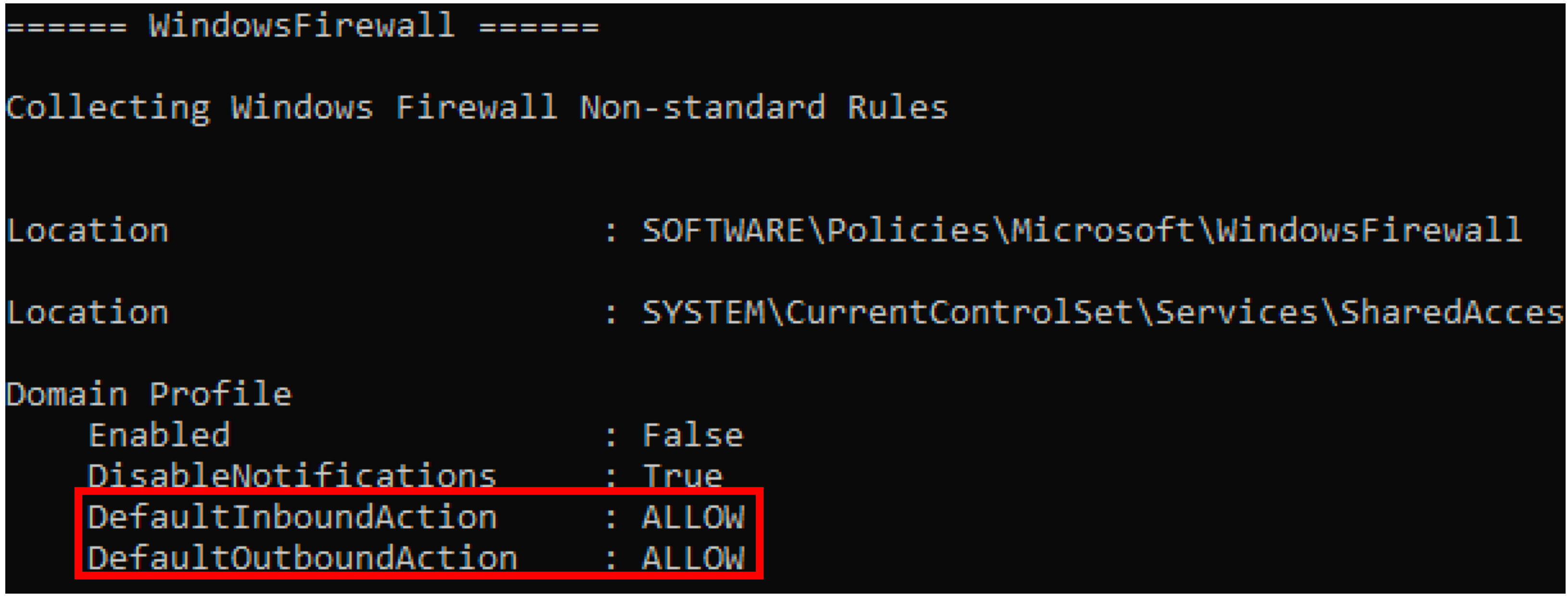
```
C:\>whoami
theshire\vulnscanner

C:\>dir \\wsus.theshire.local\C$
 Volume in drive \\wsus.theshire.local\C$ has no label.
 Volume Serial Number is A4FF-7240

 Directory of \\wsus.theshire.local\C$

05/30/2019  02:08 PM    <DIR>          PerfLogs
05/30/2019  02:08 PM    <DIR>          Program Files
05/30/2019  02:08 PM    <DIR>          Program Files (x86)
11/07/2020  05:29 PM    <DIR>          Users
11/07/2020  05:22 PM    <DIR>          Windows
               0 File(s)              0 bytes
               5 Dir(s)  45,044,133,888 bytes free
```



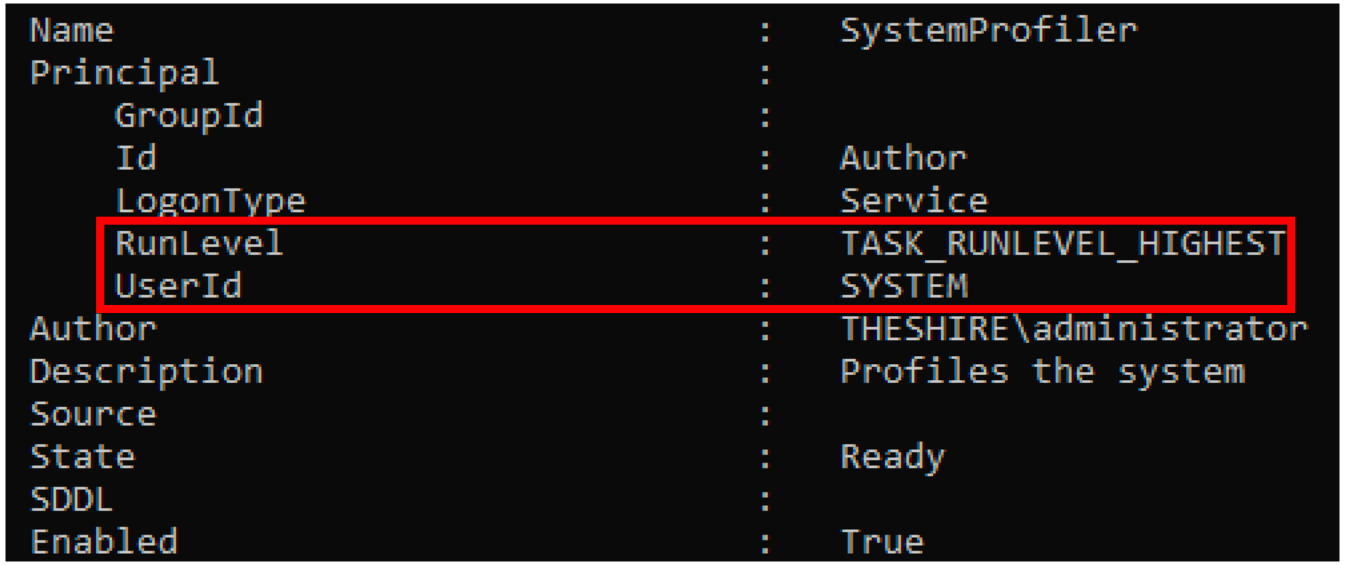
```
C:\Temp>whoami
theshire\vulnscanner

C:\Temp>Seatbelt.exe -group=remote -computername=wsus.theshire.local
[*] Running commands remotely against the host 'wsus.theshire.local' with current user credentials
```



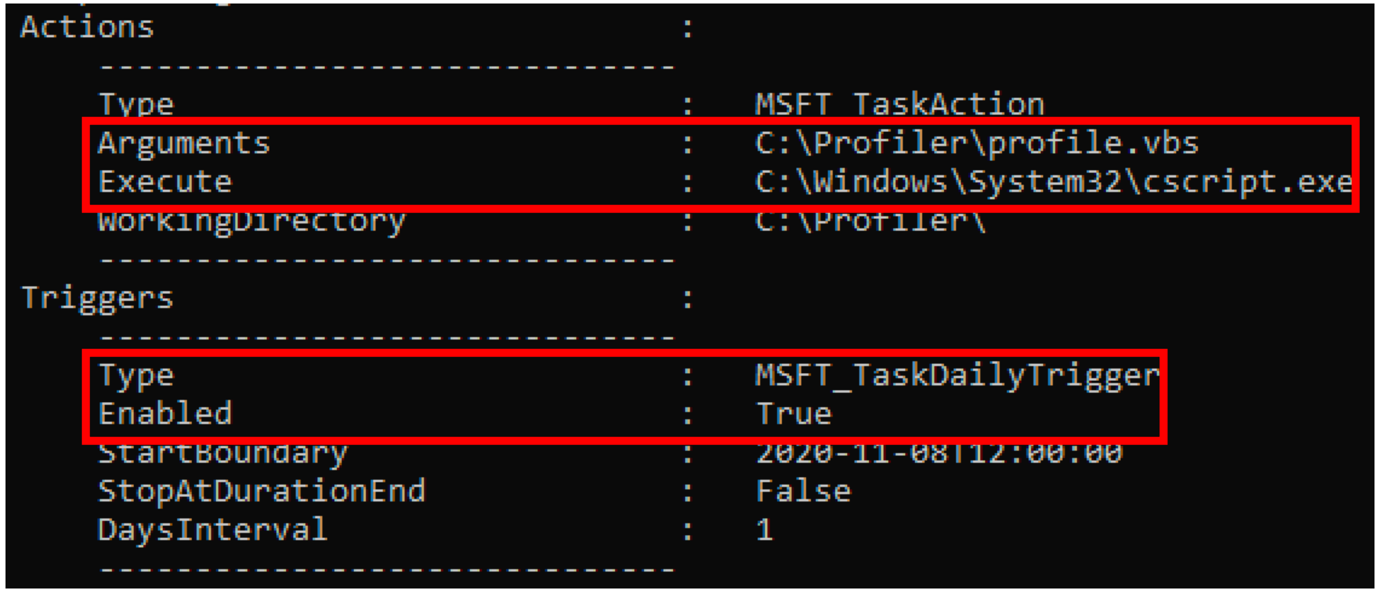
```
UserName                : administrator
Domain                  : THESHIRE
LogonId                 : 3739806
LogonType               : Interactive
AuthenticationPackage   : Kerberos
StartTime               : 11/7/2020 6:25:55
UserPrincipalName       :
```



```
====== WindowsFirewall ======

Collecting Windows Firewall Non-standard Rules


Location                     : SOFTWARE\Policies\Microsoft\WindowsFirewall

Location                     : SYSTEM\CurrentControlSet\Services\SharedAcces

Domain Profile
    Enabled                  : False
    DisableNotifications     : True
    DefaultInboundAction     : ALLOW
    DefaultOutboundAction    : ALLOW
```



```
Name                              :    SystemProfiler
Principal                         :
    GroupId                       :
    Id                            :    Author
    LogonType                     :    Service
    RunLevel                      :    TASK_RUNLEVEL_HIGHEST
    UserId                        :    SYSTEM
Author                            :    THESHIRE\administrator
Description                       :    Profiles the system
Source                            :
State                             :    Ready
SDDL                              :
Enabled                           :    True
```



```
Actions                                 :
    ------------------------------
    Type                                :    MSFT_TaskAction
    Arguments                           :    C:\Profiler\profile.vbs
    Execute                             :    C:\Windows\System32\cscript.exe
    WorkingDirectory                    :    C:\Profiler\
    ------------------------------
Triggers                                :
    ------------------------------
    Type                                :    MSFT_TaskDailyTrigger
    Enabled                             :    True
    StartBoundary                       :    2020-11-08T12:00:00
    StopAtDurationEnd                   :    False
    DaysInterval                        :    1
    ------------------------------
```

Wrapup

- Seatbelt aims to be a clearinghouse for any host-based artifact that might be interesting from a security perspective
  - Many/most Seatbelt commands can be run remotely before lateral movement!

- The more data you collect, the better decisions you can make on an engagement
  - Various Seatbelt data sources can help you make better decisions before you execute the next step of your attack path

Thanks!

- Any questions?
- @harmj0y on Twitter and the BloodHound Slack, will [at] specterops.io
- @tifkin_ on Twitter / lee [at] specterops.io

www.specterops.io
@specterops
info@specterops.io