← Blog

Andrey Zhdanov

Chief Malware Analyst and Threat Hunter

Vladislav Azersky

Incident Response and Digital Forensics Analyst

The old way: BabLock, new ransomware quietly cruising around Europe, Middle East, and Asia

Group-IB uncovers a new stealthy ransomware strain

April 4, 2023 · min to read · Ransomware

BabLock Digital Forensics Incident Response Ransomware

The New Year holidays have always been a time of the year that our Digital Forensics and Incident Response unit spends in anticipation of something bad. The anticipation usually lasts until the end of the New Year holidays in Russia and some other Post-Soviet states. A fair share of ransomware gangs and their affiliates are Russian speaking, and they take a break during the winter holidays, just like law abiding citizens do. Once the holidays are over, these groups get back to work. So do cyber incident response teams. And this year was not an exception.

In mid-January 2023, Group-IB's Amsterdam-based Digital Forensics and Incident Response team was called in to investigate one of those post-New-Year-holidays attacks against an industrial sector company in Europe. During the investigation, Group-IB's experts established that the victim had been encrypted with a previously unknown ransomware strain. The strain, first uncovered by Group-IB researchers in January 2023, was codenamed **BabLock,** because its versions for Linux and ESXi share similarities with the leaked Babuk ransomware. Despite these slight similarities, the group has a very distinct modus operandi and custom sophisticated ransomware for Windows. Additionally, BabLock gang (also tracked under the name "Rorschach" by CheckPoint), unlike most of its "industry peers", is not using a Data Leak Site (DLS) and is communicating with its victims via email.

The absence of DLS, along with relatively modest ransom requests ranging from **50,000 to 1,000,000 USD**, allows the group to operate stealthily and remain under the radar of cybersecurity researchers. The strain has been active since at least June 2022, when its earliest known version for ESXi was released. Interestingly, all BabLock ransomware modules for *Windows* that Group-IB researchers found were compiled in 2021, according to timestamps.

In addition to Europe, the group allegedly carried out attacks in Asia and the Middle East, based on the BabLock samples submitted to VirusTotal. Notably, the group doesn't encrypt devices that use Russian and other languages spoken in the Post-Soviet space.

The artifacts gathered during the incident response engagement in Europe suggest BabLock employs sophisticated tactics such as the exploitation of CVE, DLL side-loading as well as complex anti-analysis and detect evasion techniques. This blog contains a comprehensive description of the BabLock attack: their toolset, the strain's samples for Windows, ESXi, and Linux as well as TTPs used by the BabLock gang mapped to MITRE ATT&CK®. Incident Response experts, SOC teams, and threat intelligence specialists will also find a list of all known indicators of compromise related to the new groups in the end of this blog post.

BabLock January Attack

To gain initial access to the victim's infrastructure, the attackers used a remote code execution (RCE) vulnerability in the **email software Zimbra Collaboration** (ZCS) 8.8.15 and 9.0, namely **CVE-2022-41352** that enables a threat actor to remotely execute arbitrary code. The vulnerability was discovered in September 2022 and has the **NIST CVSS Score 9.8 (out of 10)**. *The Zimbra Collaboration* software used by the victim had not been updated, which once again highlights the importance of installing patches on time.

After successfully exploiting this vulnerability, the attackers connected to the domain controller from a compromised Zimbra server via a Remote Desktop Protocol (RDP). To do so, they used a domain administrator account. We didn't find any details about the threat actors' obtaining the administrator account because the logs available for examination after the attack didn't contain such information. The day after the initial connection to the domain controller, the following files were copied:

BEST_uninstallTool.exe
cy.exe
winutils.dll
config.ini

BEST_uninstallTool.exe is a legitimate utility for uninstalling Bitdefender Endpoint Security Tools. The other files are related to ransomware.

To gain access to Linux systems, the attackers used a utility called PuTTY. As a result, a Linux version of the BabLock ransomware, s86.out, was copied to these systems. In addition, we detected connections to an external IP address, which is used by Cobalt Strike network infrastructure.

The whole attack took about 24 hours to complete. As a result, files in Windows systems and network shares as well as files belonging to VMware ESXi virtual machines were encrypted.

Notably, the attackers did not collect or transfer the victim's data. After the encryption, txt notes with ransom demands for decryption were created in each directory and two email addresses were provided to contact the operators.

The IT infrastructure of the organization is based primarily on VMware ESXi virtual systems. After they were encrypted, most of the information that could be useful to investigate the incident became unavailable.

In the next section, we'll focus on the analysis of the BabLock sample for Windows that we retrieved during the incident response engagement.

Ransomware for Windows

Malware that uses the **DLL side-loading** technique rarely leaves cybersecurity researchers indifferent. Chinese APT groups have always been considered the creators and active users of this

technique. The backdoor **PlugX, attributed by cybersecurity researchers to Chinese threat actors,** is the first to spring to mind regarding this technique.

The Windows version of the BabLock ransomware used in the investigated attack employs DLL side-loading to load **winutils.dll** targeting the vulnerable legitimate software file **cy.exe**. The vulnerable file was cydump.exe from the utility **Cortex XDR Dump Service Tool** belonging to the cybersecurity company Palo Alto Networks, Inc. (Figure 1).

Figure 1. Properties of cy.exe

We later found other samples of this malware family that used a legitimate file from software belonging to another cybersecurity company. The use of DLL side-loading by ransomware is somewhat surprising, but not extremely rare; for instance, this technique had earlier been used by **Polar** ransomware. This poses a question, however, whether using DLL side-loading in ransomware makes sense. Affiliates of notorious RaaS use ransomware mostly at later stages of attacks after bypassing most security controls. It is reasonable to assume that DLL side-loading is justified in

programs for quick attacks, such as the one described above, where there is a risk that the ransomware will be detected and neutralized before it launches.

As a result of this technique, launching the legitimate file cy.exe will load the malicious winutils.dll (original name: XLOADERDLL.dll), which is located in the same directory.

The winutils.dll module is compressed using a modified **UPX 3.96 packer**, which does not let it be unpacked using a standard version of UPX. Unpacking revealed other protective techniques in the DLL, such as **string obfuscation, junk code, and calling Native API functions using direct system calls** (syscall) (Figure 2). To obtain system call numbers, the contents of ntdll.dll are loaded into memory as a memory-mapped file and the code of the following Native API functions is parsed:

NtCreateFile
NtReadFile
NtWriteFile
NtClose
NtCreateProcess
NtAllocateVirtualMemory
NtReadVirtualMemory
NtWriteVirtualMemory
NtResumeThread
NtCreateThreadEx
NtQueryInformationProcess
NtFreeVirtualMemory
NtProtectVirtualMemory

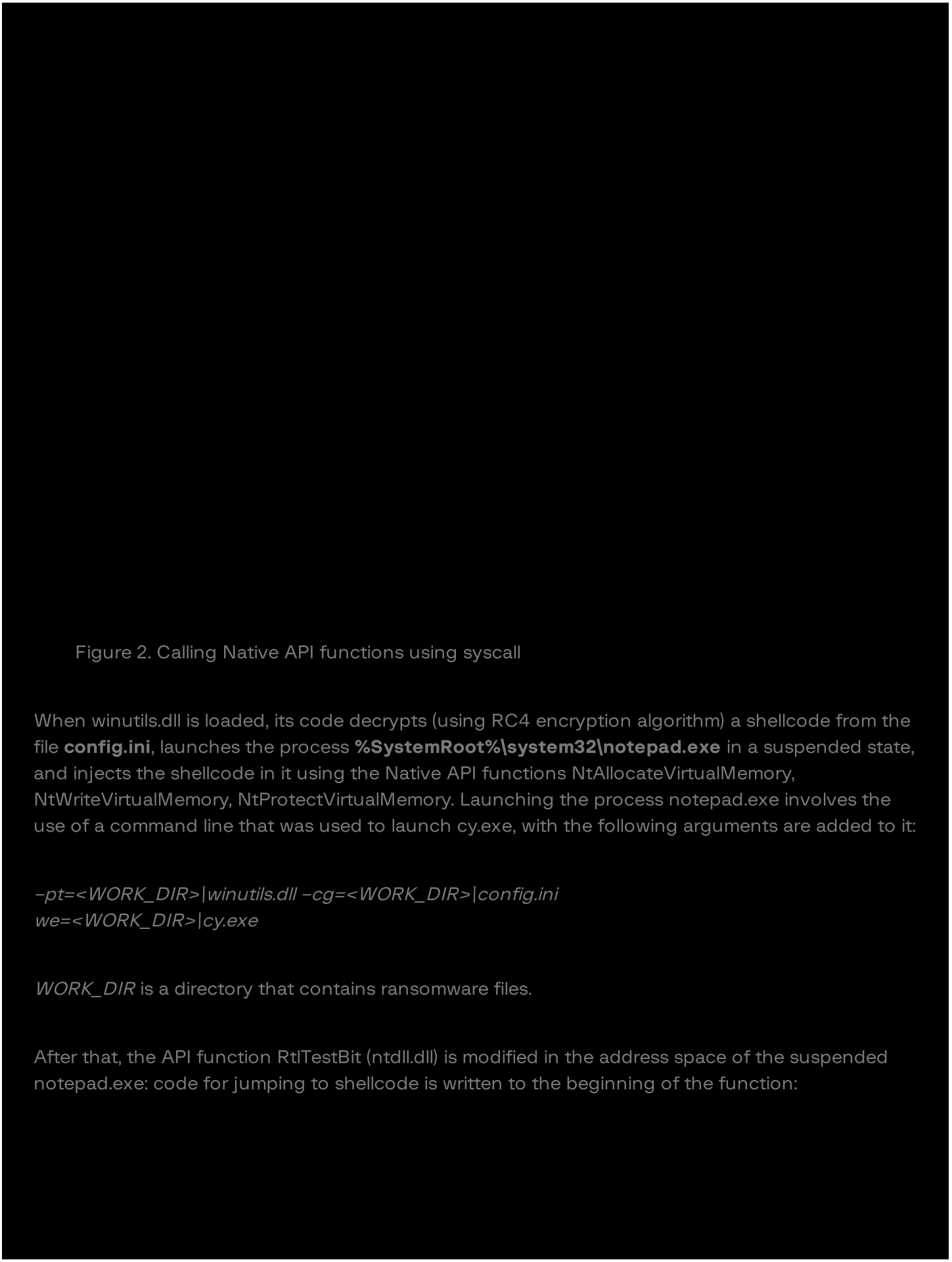

Figure 2. Calling Native API functions using syscall

When winutils.dll is loaded, its code decrypts (using RC4 encryption algorithm) a shellcode from the file **config.ini**, launches the process **%SystemRoot%\system32\notepad.exe** in a suspended state, and injects the shellcode in it using the Native API functions NtAllocateVirtualMemory, NtWriteVirtualMemory, NtProtectVirtualMemory. Launching the process notepad.exe involves the use of a command line that was used to launch cy.exe, with the following arguments are added to it:

–pt=<WORK_DIR>|winutils.dll –cg=<WORK_DIR>|config.ini
we=<WORK_DIR>|cy.exe

WORK_DIR is a directory that contains ransomware files.

After that, the API function RtlTestBit (ntdll.dll) is modified in the address space of the suspended notepad.exe: code for jumping to shellcode is written to the beginning of the function:

Figure 3. Code of the modified function RtlTestBit

With the help of the function NtCreateThreadEx, a thread is created and launched in the notepad.exe process; the modified RtlTestBit function is used as a thread function.

The shellcode loads the payload PE module contained in it directly into memory. The addresses of the necessary Windows API functions are obtained in the shellcode with the help of the Process Environment Block (PEB) and the algorithm for calculating hashes for function names is based on the popular **ROR13** algorithm. It is worth mentioning that similar PE-module loader code has been observed in two other malware families, namely the banking **Trojan KrBanker / BlackMoon** and the TA505-related bot **SDBbot**.

The payload is ransomware in the PE32+ DLL format. The ransomware uses an unknown protector. After the protector was removed, the program's main function remained virtualized; the strings were obfuscated using various methods, like in winutils.dll (Figure 4), and direct system calls (syscall) are used to call certain Native API functions:

NtCreateFile
NtReadFile
NtWriteFile
NtClose
NtQueryInformationFile
NtWaitForSingleObject
NtSetInformationFile
NtQueryEaFile

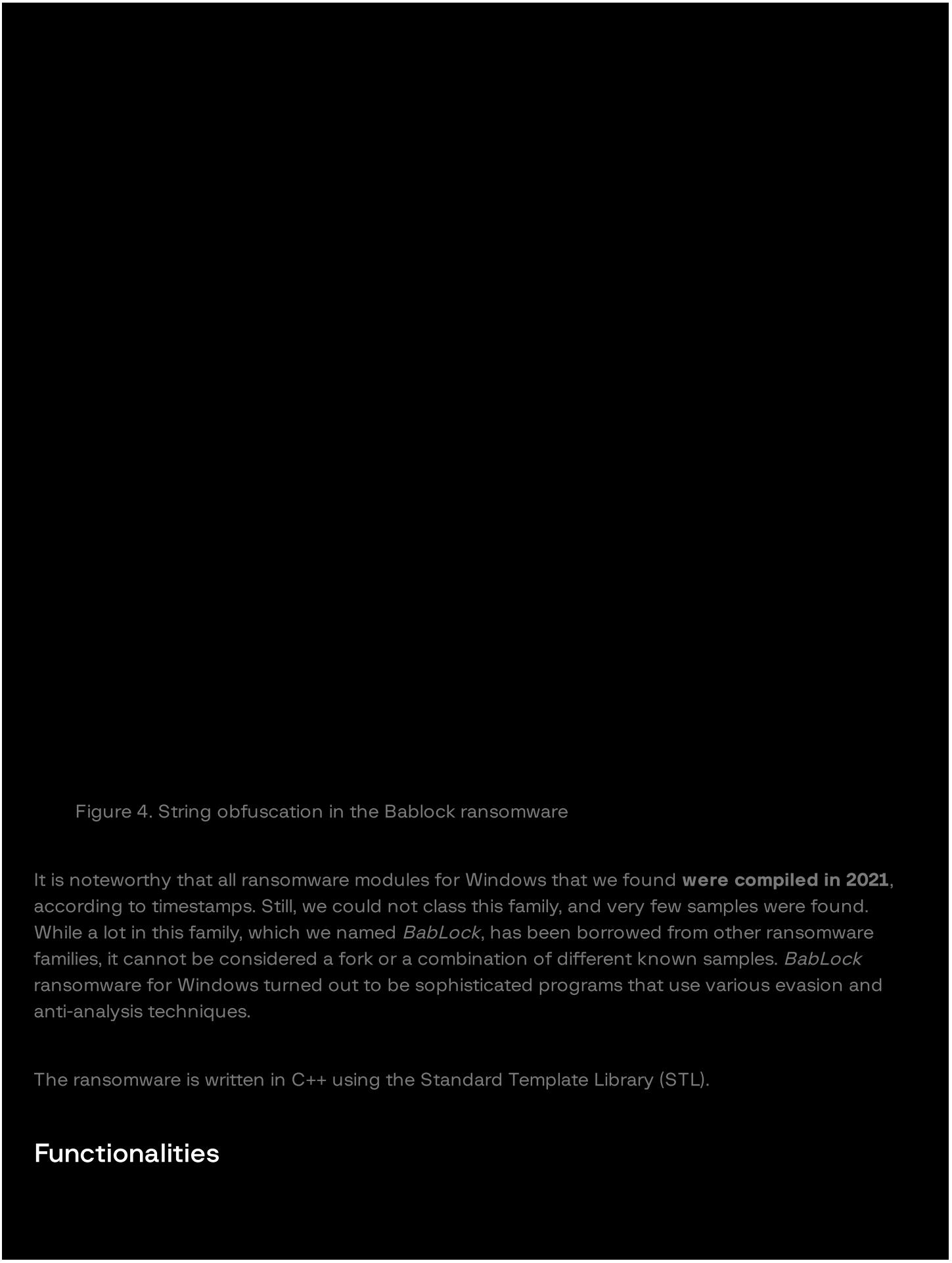

Figure 4. String obfuscation in the Bablock ransomware

It is noteworthy that all ransomware modules for Windows that we found **were compiled in 2021**, according to timestamps. Still, we could not class this family, and very few samples were found. While a lot in this family, which we named *BabLock*, has been borrowed from other ransomware families, it cannot be considered a fork or a combination of different known samples. *BabLock* ransomware for Windows turned out to be sophisticated programs that use various evasion and anti-analysis techniques.

The ransomware is written in C++ using the Standard Template Library (STL).

Functionalities

The ransomware does not encrypt files and shuts down if the default language of the system or user is one of the following:

Russian	419
Ukrainian	422
Belarusian	423
Tajik	428
Armenian	42B
Azerbaijani Latin	42C
Azerbaijani Cyrillic	82C
Georgian	437
Kazakh	43F
Kyrgyz	440

The language is checked using the Windows API functions **GetSystemDefaultUILanguage** and **GetUserDefaultUILanguage**.

For comparison, in addition to the list above, LockBit 3.0 (Black) checks for Tatar (444), Romanian Moldova (818) and Arabic Syria (2801).

Launching the ransomware requires specifying the correct value of the command-line argument "–run=". In the sample in question this value is, i.e. "–run=3306"; simply specifying 3306 in the command line without the argument is also possible. If the launch code is not specified or is specified incorrectly, the ransomware shuts down. This technique is used to evade sandbox analysis. The checked value 3306 is hardcoded in the form of a string in the sample. Other samples of *BabLock* for Windows use other values for launch.

Depending on the command-line parameters, the ransomware can encrypt a given object (directory, file, network resource) or the entire system. A description of command-line arguments is provided below.

The ransomware stops the following security, backup, database management and other system services:

vss	*ccSetMgr*	*veeam*
sql	*SavRoam*	*PDVFSService*
svc$	*RTVscan*	*BackupExecVSSProvider*
memtas	*QBFCService*	*BackupExecAgentAccelerator*
mepocs	*QBIDPService*	*BackupExecAgentBrowser*
sophos	*Intuit.QuickBooks.FCS*	*BackupExecDiveciMediaService*
veeam	*QBCFMonitorService*	*BackupExecJobEngine*
backup	*YooBackup*	*BackupExecManagementService*
GxVss	*YooIT*	*BackupExecRPCService*
GxBlr	*zhudongfangyu*	*AcrSch2Svc*

The ransomware also terminates the following processes of database management systems, email clients, office applications, etc.:

sql.exe	*mydesktopservice.exe*	*sqbcoreservice.exe*	*steam.exe*
oracle.exe	*ocautoupds.exe*	*excel.exe*	*thebat.exe*
ocssd.exe	*encsvc.exe*	*infopath.exe*	*thunderbird.exe*
dbsnmp.exe	*firefox.exe*	*msaccess.exe*	*visio.exe*
synctime.exe	*tbirdconfig.exe*	*mspub.exe*	*winword.exe*
agntsvc.exe	*mydesktopqos.exe*	*onenote.exe*	*wordpad.exe*
isqlplussvc.exe	*ocomm.exe*	*outlook.exe*	*wrapper.exe*

xfssvccon.exe dbeng50.exe powerpnt.exe dbsrv12.exe

To delete volume shadow copies and system state backups, disable recovery in *Windows* boot menu, clear *Windows event logs*, shut down certain services, and disabling the firewall, the ransomware executes the following commands:

```
vssadmin.exe Delete Shadows /All /Quiet
bcdedit.exe /set {default} recoveryenabled No
bcdedit.exe /set {default} bootstatuspolicy ignoreallfailures
wbadmin.exe DELETE SYSTEMSTATEBACKUP
wbadmin.exe DELETE SYSTEMSTATEBACKUP -deleteOldest
wbadmin.exe delete catalog -quiet
wbadmin.exe delete backup
wbadmin.exe delete systemstatebackup -keepversions:0
wevtutil.exe clear-log Application
wevtutil.exe clear-log Security
wevtutil.exe clear-log System
wevtutil.exe clear-log "windows powershell"
wmic.exe SHADOWCOPY /nointeractive
net.exe stop MSDTC
net.exe stop SQLSERVERAGENT
net.exe stop MSSQLSERVER
net.exe stop vds
net.exe stop SQLWriter
net.exe stop SQLBrowser
net.exe stop MSSQLSERVER
net.exe stop MSSQL$CONTOSO1
netsh.exe advfirewall set currentprofile state off
netsh.exe firewall set opmode mode=disable
```

When these commands are executed, the ransomware uses an artifact hiding technique called **process argument spoofing**: it creates a system program process in a suspended state, writes command-line arguments directly into the PEB, and then resumes the process.

It is worth noting that along with sophisticated solutions the program has simple mistakes. For instance, executing the command shown below will not delete volume shadow copies:

wmic.exe SHADOWCOPY /nointeractive

The command lacks the required argument "*DELETE*".

Creation of an Active Directory group policy

If the ransomware has been launched on the domain controller with administrator privileges, it uses a group policy to share access to hosts' disks, stop services and terminate processes on hosts, and spread itself in the local area network. The use of group policies in the ransomware is similar to how it is implemented in LockBit 2.0.

The ransomware extracts the following Group Policy Object (GPO) components — which determine a new group policy — into the root directory on the domain controller:

```
\Machine\Preferences\NetworkShares\NetworkShares.xml
\Machine\Preferences\Services\Services.xml
\Machine\comment.cmtx
\Machine\Registry.pol
\User\Preferences\Files\Files.xml
\User\Preferences\ScheduledTasks\ScheduledTasks.xml
```

GPO_GUID – is the GUID of the new group policy

NetworkShares.xml (Figure 5) is meant for giving shared access to domain hosts' disks so that the ransomware can access more files in the victim's network to encrypt them.

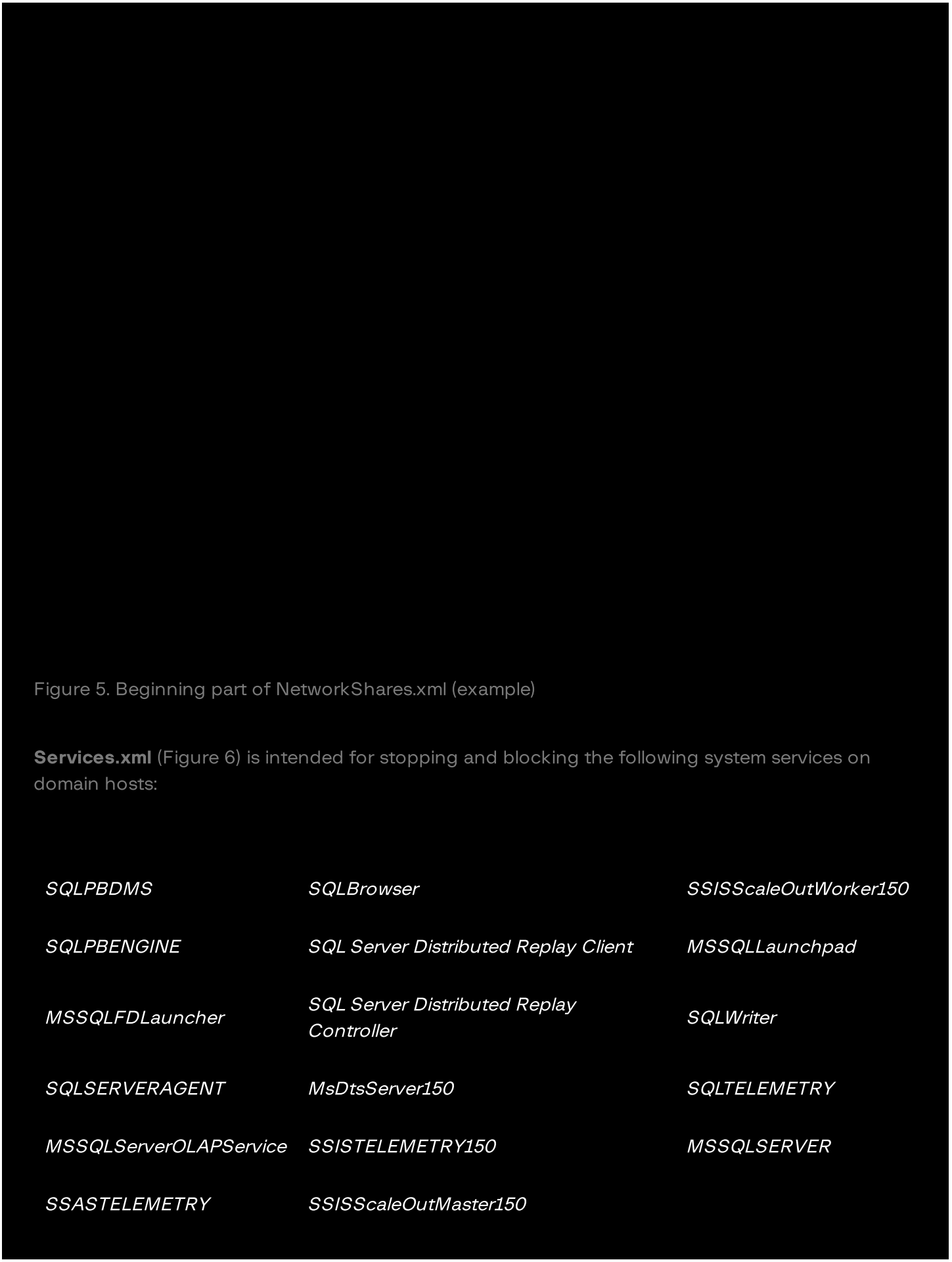

Figure 5. Beginning part of NetworkShares.xml (example)

Services.xml (Figure 6) is intended for stopping and blocking the following system services on domain hosts:

SQLPBDMS	*SQLBrowser*	*SSISScaleOutWorker150*
SQLPBENGINE	*SQL Server Distributed Replay Client*	*MSSQLLaunchpad*
MSSQLFDLauncher	*SQL Server Distributed Replay Controller*	*SQLWriter*
SQLSERVERAGENT	*MsDtsServer150*	*SQLTELEMETRY*
MSSQLServerOLAPService	*SSISTELEMETRY150*	*MSSQLSERVER*
SSASTELEMETRY	*SSISScaleOutMaster150*	

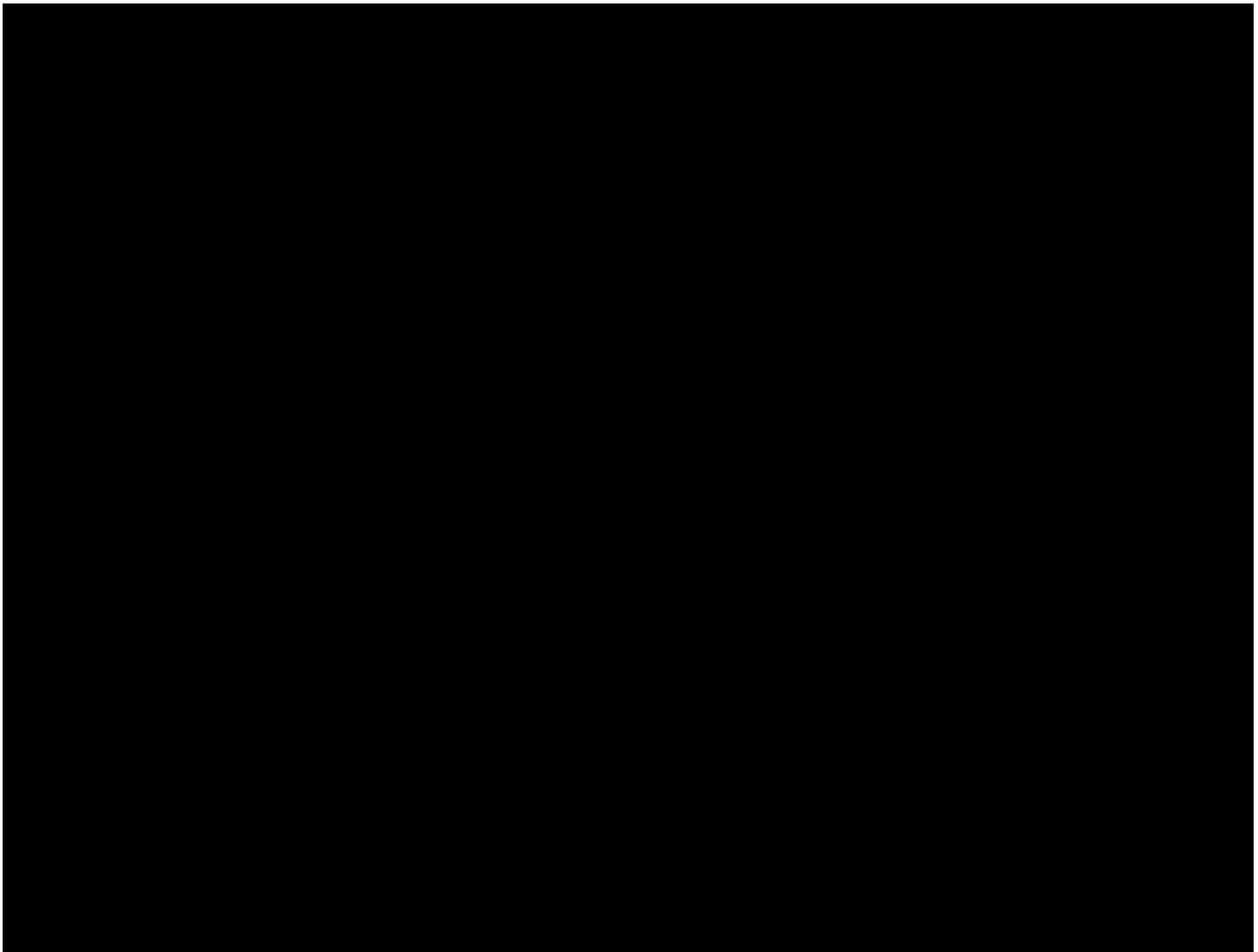

Figure 6. The beginning part of Services.xml (example)

The files **Registry.pol** and **comment.cmtx** (Figure 7) are intended for disabling **Windows Defender** on hosts by modifying relevant parameters in the system registry.

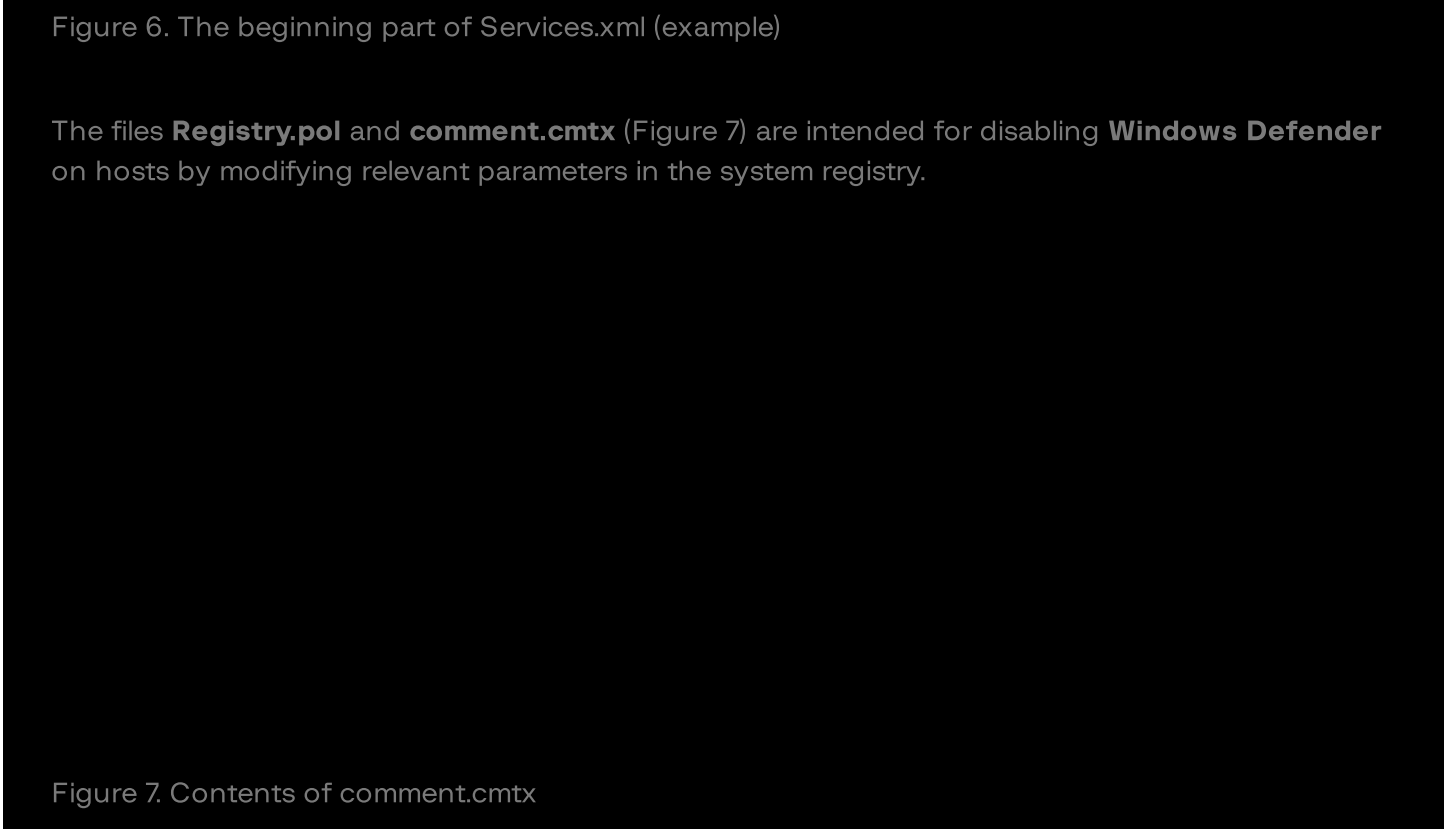

Figure 7. Contents of comment.cmtx

The ransomware copies its files to the shared Active Directory files directory *SYSVOL*:

```
\\< DNS_DOMAIN_NAME>\sysvol\< DNS_DOMAIN_ NAME>\scripts\
```

DNS_DOMAIN_ NAME is the name of the DNS domain.

Files.xml (Figure 8) is intended for copying ransomware files from the shared *Active Directory* files directory *SYSVOL* to the host's *%Public%* directory.

Figure 8. Contents of Files.xml (example)

ScheduledTasks.xml (Figure 9) is intended for creating two scheduled tasks on the host:

1_MMdd_Services
2_MMdd_<EXE_NAME>

EXE_NAME is the name of the main executable (legitimate file) of the ransomware (cy.exe);

MMdd is the month and day when *ScheduledTasks.xml* was created.

Figure 9. Contents of ScheduledTasks.xml (example)

The scheduled task *1_MMdd_Services* (Figure 10) terminates processes on domain hosts by executing the following command for each process from the list:

C:\Windows\System32\taskkill.exe /IM "<PROC_NAME>" /F

Names of terminated processes (*PROC_NAME*):

wxServer.exe	*supervise.exe*	*sync-taskbar.exe*	*vxmon.exe*
wxServerView.exe	*Culture.exe*	*sync-worker.exe*	*sqlbrowser.exe*
sqlmangr.exe	*Defwatch.exe*	*wsa_service.exe*	*tomcat6.exe*
RAgui.exe	*httpd.exe*	*synctime.exe*	*Sqlservr.exe*

Figure 10. Part of 1_0123_Services (example)

The second task, **2_MMdd_cy.exe**, (Figure 11) calls the main ransomware executable, located in the host's *%Public%* directory. The ransomware's command-line arguments for the task are created based on the arguments of the launched ransomware.

Figure 11. 1_0123_cy.exe (example)

Scheduled tasks are launched immediately after they are created.

Group policies on domain computers are updated using the following PowerShell command:

```
powershell.exe -Command "Get-ADComputer -filter * -Searchbase 'AD_SEARCHPATH' | foreach{
```

where *AD_SEARCHPATH* is the *AD* path for searches:

DC=<DC1>,DC=<DC2>

where *DC1, DC2* are domain components

File encryption

The ransomware encrypts files on disks and available network resources. Before encryption, the ransomware mounts hidden volumes. To search for network resources, the ransomware also enumerates Active Directory computers using LDAP queries.

During encryption, the ransomware skips files and directories with the following names:

AppData	*ProgramData*	*ntuser.ini*	*bootfont.bin*
Boot	*All Users*	*thumbs.db*	*ntldr*
Windows	*autorun.inf*	*NTUSER.DAT*	*config.ini*
WINDOWS	*boot.ini*	*ntuser.dat.LOG1*	*1_config.ini*
Windows.old	*bootfont.bin*	*ntuser.dat.LOG2*	*Policies*
Ahnlab	*bootsect.bak*	*thumbs.db*	*NETLOGON*
Tor Browser	*bootmgr*	*Program Files*	*SYSVOL*
Internet Explorer	*bootmgr.efi*	*Program Files (x86)*	*begin.txt*
Google	*bootmgfw.efi*	*#recycle*	*finish.txt*

Opera *desktop.ini* *scripts*

PID is the ransomware process identifier.

Files with the following extensions are not encrypted either:

".exe", ".dll", ".sys", ".com", ".EXE", ".DLL", ".SYS", ".COM"

To encrypt files the ransomware uses high-performance implementation of multithreading using an I/O (input/output) completion port. I/O completion port–based multithread encryption is also implemented in ransomware such as **LockBit 3.0** / **BlackMatter**, **DarkSide**, **REvil**, and the latest version of **Hive v6**.

The actual encryption of data is similar to how it is implemented in the **Babuk** ransomware family, but multithread encryption in Babuk is simpler. Babuk for Windows in general is much simpler than the ransomware in question. We decided to reflect the connection with Babuk in the name of the new ransomware family, the more so because Linux versions of the ransomware were developed based on the source code of Babuk for NAS and ESXi. We also decided to take into account the complexity of the Windows version and the similarity with LockBit in terms of group policy use. Hence, we named the new family BabLock.

Encryption involves the stream cipher HC-128 with a 256-bit key and an initialization vector *(IV).* For each file, a 32-byte private key is generated using the **Crypto API function CryptGenRandom**. From this key, 32-byte public and shared keys are calculated by way of Diffie–Hellman key exchange implemented using *Curve25519*. The attackers' public key used for the exchange is contained in the ransomware code. From the obtained shared key, a *SHA-512* hash is calculated, whose first 32 bytes are used as the *HC-128* key, while the other 32 bytes are used as the *IV*.

The ransomware encrypts the starting 16-megabyte block in the files with the extensions "*.sql*", "*.mdf*", "*.mdb*", "*.db*", "*.dbf*", "*.wdb*", "*.accdb*", "*.rar*,", "*.zip*", and "*.7z*". In files with other extensions, one 16-kilobyte block is encrypted with a 256 offset. If the size of a file is less than 512 bytes, the file is encrypted fully.

The list of extensions above contains another simple mistake: for .rar files the comma comes before the quotation mark in the source code, which means that .rar files will not be encrypted.

After the data in a file is encrypted, a 68-byte block of metadata is added to the end of the file. The metadata is encrypted using a 32-bit XOR operation with a dword.

Offset	Size	Description
000h	32	Calculated public key for the file.
020h	4	Hash of an XCRC32 encryption key and HC128 IV.
024h	4	00000001h
028h	8	Decryption ID corresponding to the first 8 bytes of the attackers' public key.
030h	16	Encrypted files marker 75 EC 81 78 B9 DB 87 B0 E3 99 75 5D 8D 03 F9 65
040h	4	XOR metadata encryption key. The value of the key is generated for each file.

Names of encrypted files are as follows:

<FILENAME>.<RANSOM_EXT>.<RND_N>

FILENAME is the original name of the file.

RANSOM_EXT is the extension, which is hardcoded in the ransomware code ("*slpqne*");

RND_N is a random two-digit number from 00 to 98 (inclusive).

In each processed directory, the ransomware creates *_r_e_a_d_m_e.txt* text files with a ransom demand for decrypting files (Figure 12).

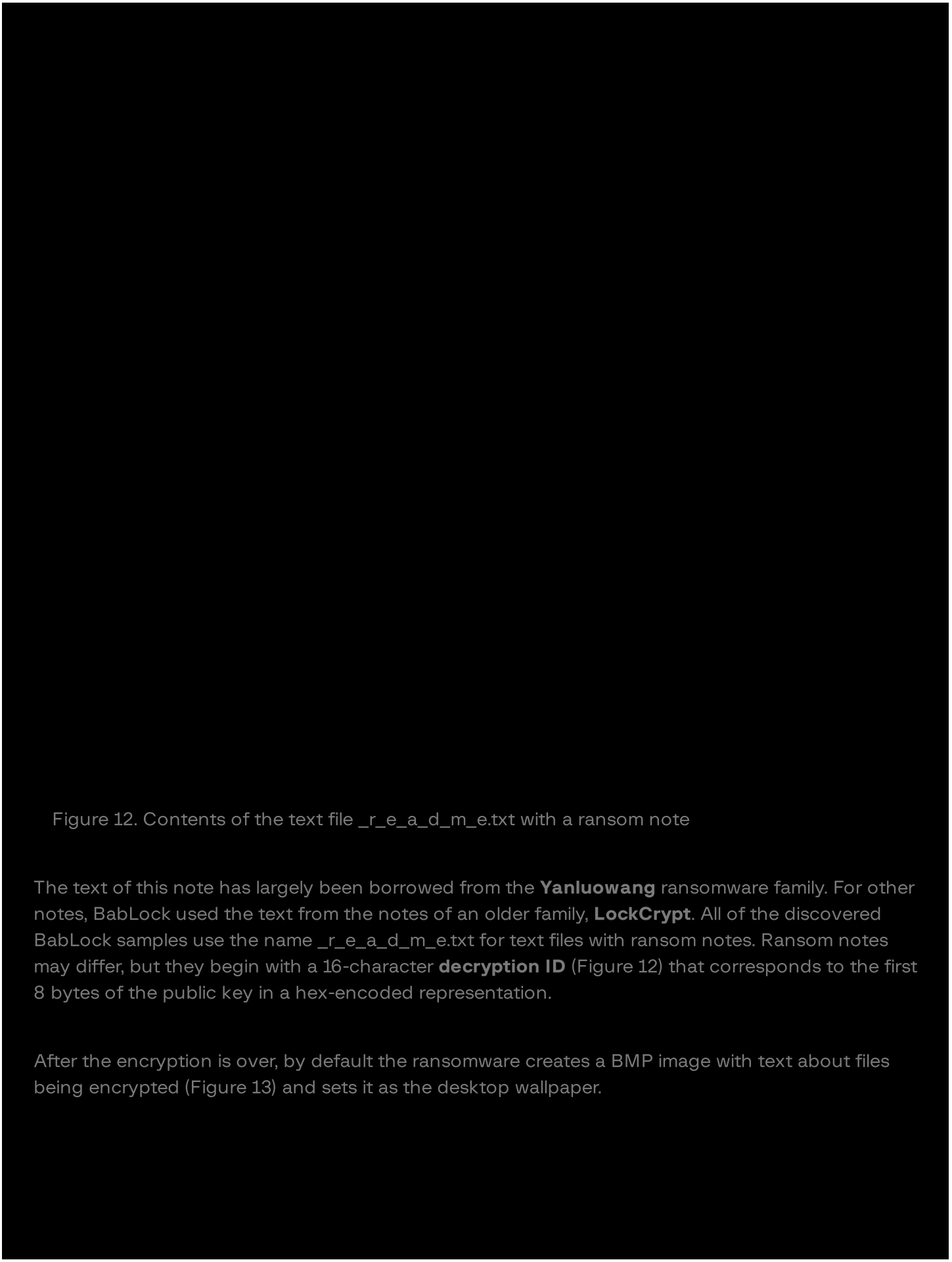

Figure 12. Contents of the text file _r_e_a_d_m_e.txt with a ransom note

The text of this note has largely been borrowed from the **Yanluowang** ransomware family. For other notes, BabLock used the text from the notes of an older family, **LockCrypt**. All of the discovered BabLock samples use the name _r_e_a_d_m_e.txt for text files with ransom notes. Ransom notes may differ, but they begin with a 16-character **decryption ID** (Figure 12) that corresponds to the first 8 bytes of the public key in a hex-encoded representation.

After the encryption is over, by default the ransomware creates a BMP image with text about files being encrypted (Figure 13) and sets it as the desktop wallpaper.

Figure 13. A BMP image as a desktop background with text about files being encrypted

Command-line parameters

Parameter	Description
–run=<KEY>	Specify the value of the key to launch the ransomware. The ransomware is launched if the key value is correct (3306). The correct KEY value is contained in the ransomware code. The key can be specified in the command line without specifying the argument itself “–run=”.
–nomutex=1	Do not check the mutex. The mutex is used to check for simultaneous launch of several ransomware copies. Mutex: “80CE038F-6317-984D-C0A7-FA0A1EED0199”
–path=<PATH>	Encrypt files in a specified path.
–log=1	Create text log files: <PID>_l.log (log file) <PID>_e.log (list of encrypted files) PID is the ransomware process identifier.
–nodel=1	Do not delete ransomware executables after shutting down.

Ransomware for Linux

The *Linux* version of the ransomware is a 32-bit program for Linux in ELF format written in **Go 1.18.3** (release date: 2022-06-01).

GOVERSION variable	*go1.18.3*
GOROOT variable	*C:|Program Files|Go*
GOPATH variable	*C:|Users|Administrator|go*
Additional packages used for creating the ransomware	*golang.org/x/crypto/chacha20* *golang.org/x/crypto/curve25519*

The ransomware was developed based on the source code of Babuk ransomware for NAS, which was made public in September 2021. The ransomware was compiled on a computer running Windows not later than **January 6, 2022**.

Functionalities

The ransomware encrypts files in a path specified in a command line. The ransomware's command-line parameters are:

s86.out <PATH> [esxi]

PATH is the path for encrypting files.

esxi is the encryption mode.

During encryption, the ransomware skips directories that start with the following substrings:

/proc	*.system/thumbnail*	*/usr/bin*	*/bin*
/boot	*.system/opt*	*/usr/etc*	*/lib*
/sys	*.config*	*/usr/sbin*	*/lib32*

/run	*.qpkg*	*/usr/lib*	*/lib64*
/dev	*/mnt/ext/opt*	*/usr/syno*	*/libx32*
/etc	*/tmp*	*/var/packages*	*/root*
/home/httpd	*/usr/sh*	*/usr/local/packages*	*/sbin*

The ransomware skips files whose paths start with the following substrings:

/lib	*home/httpd*	*.qpkg*	*/sbin*
/bin	*.system/thumbnail*	*/mnt/ext/opt*	
/proc	*.system/opt*	*/tmp*	
/boot	*.config*	*/var/run*	

In esxi encryption mode, the ransomware only encrypts files with the following extensions: "*.log*", "*.vdmk*", "*.vmem*", "*.wsvp*", "*.wmsn*".

File encryption

The ransomware performs multithread encryption of files. Data is encrypted using the *ChaCha20* stream cipher algorithm. For each file, a 32-byte private key is generated (*/dev/urandom*), from which 32-byte public and shared keys are calculated by way of Diffie–Hellman key exchange implemented using *Curve25519*. The attackers' public key used for the exchange is contained in the ransomware code. From the obtained public key, a *SHA-256* hash is calculated, which is used as the *ChaCha20* key. From this key, a *SHA256* hash is calculated, whose bytes [10:22] are used as *nonce*.

In files with the extensions "*.log*", "*.vdmk*", "*.vmem*", "*.wsvp*", and "*.wmsn*" the ransomware encrypts ten 16384-byte blocks, each with an offset in the file divisible by 10 megabytes. In files with other extensions, only the first 16384-byte starting block is encrypted.

After data in a file is encrypted, a 46-byte block of metadata is added to the end of the file.

Offset	Size	Description
000h	32	Calculated public key for the file.
020h	6	Encrypted files marker CE AD 38 DE F9 00 – one block is encrypted. CE AD 38 DE F9 01 – several blocks are encrypted.
026h	8	Decryption ID, which corresponds to the first 8 bytes of the attackers' public key.

In each processed directory, the ransomware creates *_r_e_a_d_m_e.txt* text files with a ransom demand for decrypting files. The contents of *_r_e_a_d_m_e.txt* are identical to ones created by *BabLock* for *Windows*.

Ransomware for ESXi

In the attack in question, we were unable to obtain a sample of BabLock for ESXi, but we found samples of BabLock for ESXi that were used in other attacks.

The ransomware is a 64-bit program for *Linux* in ELF format compiled using GNU Compiler (GCC).

The ransomware was developed based on the source code of Babuk for ESXi, which was made public in September 2021, and is virtually identical to the original.

The ransomware encrypts files in a path specified in a command line and only encrypts files with the following extensions:

"*.log*", "*.vdmk*", "*.vmem*", "*.wsvp*", and "*.wmsn*".

File encryption

The ransomware performs multithread encryption of files, with 16 threads used for encryption.

Data is encrypted using the **Sosemanuk** stream cipher algorithm. For each file, a 32-byte private key is generated (*/dev/urandom*), from which 32-byte public and shared keys are calculated by way of

Diffie–Hellman key exchange implemented using *Curve25519*. The attackers' public key used for the exchange is contained in the ransomware code. From the obtained shared key, a *SHA-256* hash is calculated, which is used as the *Sosemanuk* key.

The ransomware encrypts the first 100 megabytes in files by 10-megabyte blocks. For comparison, *Babuk* for ESXi encrypts the first 500 megabytes by 1-megabyte blocks.

After the data in a file is encrypted, a 40-byte block of metadata is added to the end of the file.

Offset	Size	Description
000h	32	Calculated public key for the file.
020h	8	Decryption ID, which corresponds to the first 8 megabytes of the attackers' public key.

The original *Babuk* does not have a *Decryption ID* metadata field.

The names of encrypted files are as follows:

<FILENAME>.<RANSOM_EXT>

- *FILENAME* is the original name of the file.
- *RANSOM_EXT* is the extension, which is hardcoded in the ransomware code ("*slpqne*").

In each processed directory, the ransomware creates *_r_e_a_d_m_e.txt* text files with a ransom demand for decrypting files.

Conclusion

We believe that the group BabLock is not related to any particular RaaS affiliate program and that it performs "quiet" occasional attacks using proprietary ransomware.

The geographical scope of the group's attacks and the fact that they check whether the ransomware is launched on computers that use Post-Soviet countries' languages to prevent encryption could suggest that the group might be Russian speaking, but the information is insufficient to attribute the attack.

The group managed to remain unnoticed for a long time, because they conducted few attacks and did not employ double or triple extortion techniques. To pressure its victims, BabLock only threatens to launch attacks again, according to their ransom note. Another factor that contributed to the group's low profile could be that for encrypting *Linux* systems BabLock used ransomware based on the published source code of *Babuk* with insignificant modifications. However, it is the version for *Windows*, its complexity, and the evasion and anti-analysis techniques used that caught our eye. It would make more sense for the threat actors to use a simpler program based on *Babuk* to encrypt *Windows* systems, but they preferred developing their own, more sophisticated program, which overall is not similar to other families. It is also unusual that all of the samples for *Windows* that we have discovered were dated 2021.

Recommendations

1. Regularly installing critical updates for operating systems and software used.
2. Setting up strong password policies for both local and domain accounts. Verifying that different passwords are used for local administrators on all the hosts in the infrastructure.
3. When managing access rights, sticking to the principle of minimal required privileges in the system, with a special focus on service accounts as well as accounts used for automated tasks and remote access.
4. Prohibiting direct RDP access to workstations and servers from outside the internal network.
5. Ensuring that the storage period for operating system event logs and security controls logs lasts for at least three months.
6. Collecting the following logs from the VMware ESXi hypervisor:
   auth.log, hostd.log, syslog.log, vmksummary.log, vmkernel.log.
7. In the Linux segment, setting up the auditd tool, designed for monitoring operating system events.
8. Setting up the collection of Windows events, relating to:

    successful and unsuccessful login attempts

    enabling/disabling and blocking accounts

    creating local and domain accounts

    adding users to groups, especially ones granting elevated privileges

creating services, processes, and scheduled tasks

changing domain and local security policies

executing commands using various command interpreters

critical triggering of the built-in Windows security tool (Windows Defender)

accessing objects in network shares

clearing event logs

9. Implementing the centralized collection of events in the Linux/Windows infrastructure and their transfer to a data collection system (e.g., ELK, SIEM).
10. Using Group-IB Managed Extended Detection and Response (MXDR) to protect end devices.
11. Using Group-IB Attack Surface Management (ASM) to control the security posture of the organization's infrastructure.
12. Use Threat Intelligence to keep track of the group and changes in its tactics, techniques, and procedures.

BabLock Ransomware MITRE ATT&CK

Tactic	Technique	Description
TA0001 Initial Access	T1190 Exploit Public-Facing Application	As an initial access vector, the attacker used vulnerabilities in the email software Zimbra and as a result gained RDP access to an email system server.

T1106 Native API	The ransomware uses direct system calls (syscall) to launch certain Native API functions.
T1053.005 Scheduled Task/Job: Scheduled Task	For the ransomware to spread in the victim's infrastructure, a group policy (GPO) is used, which creates scheduled tasks on domain hosts to launch the ransomware and stop SQL system services.
T1059.001 Command and Scripting Interpreter: PowerShell	A PowerShell script is used to update group policies on domain computers.

Indicators of compromise

Windows

cy.exe
4874d336c5c7c2f558cfd5954655cacfc85bcfcb512a45fb0ff461ce9c38b86d
winutils.dll
2fd264f58ba82a2675280ec8c6759612def2bcc62aa6160f5e23071f67bb67ab
config.ini
03c41019faf7e4cc26ca0dd3a2c41b2115e4c4ebd561402079bc4a20256c1813

Shortcut.exe
88081a21e500e831d86666ca5d7a3d348f7c03bc5c471b6d17d8b18a022f25be
libexpa.dll
aa48acaef62a7bfb3192f8a7d6e5229764618ac1ad1bd1b5f6d19a78864eb31f
config.ini
b99d114b267ffd068c3289199b6df95a9f9e64872d6c2b666d63974bbce75bf2

winutils.dll
66bcad0829a59c424d062b949c2a556b11c509b17515dffecb9cbf65f13f3dc6

Dumped payload
38c610102129be21d8d99ac92f3369c6650767ed513e5744c0cda54e68b33812
e14b88795bde45cf736c8363c71a77171aa710a4e7fa9ce38470082cb1bdadbb

Linux/ESXi

7d62a33e9a2fedff6cf27aaa142ff15838a766ccd4a8d326424611e155442775
83052cc23c45ecaa09fe5c87fd650c7f8e708aea46756a2b9d452d40ce3b9c00

b711579e33b0df2143c7cb61246233c7f9b4d53db6a048427a58c0295d8daf1c
de5a53131225dd97040d48221d9afd98760f7ff2f55613f0d08436891ca632b9

The threat actors' email addresses

dcqyvp1@onionmail.org
DcqYvp@onionmail.org
dyhdsak@onionmail.org
dyhdsak1@onionmail.org
jzmc2t@tutanota.com
jzmc2t@onionmail.org
ngoueeb@onionmail.org
ngoueeb1@onionmail.org
vvbured@onionmail.org
vvbured1@onionmail.org
wvpater@onionmail.org
wvpater1@onionmail.org

EU & NA: +31 20 226 90 90

MEA: +971 4 568 1785

info@group-ib.com