

기말 과제

- UDP_flooding.exe 악성코드 분석 보고서 -

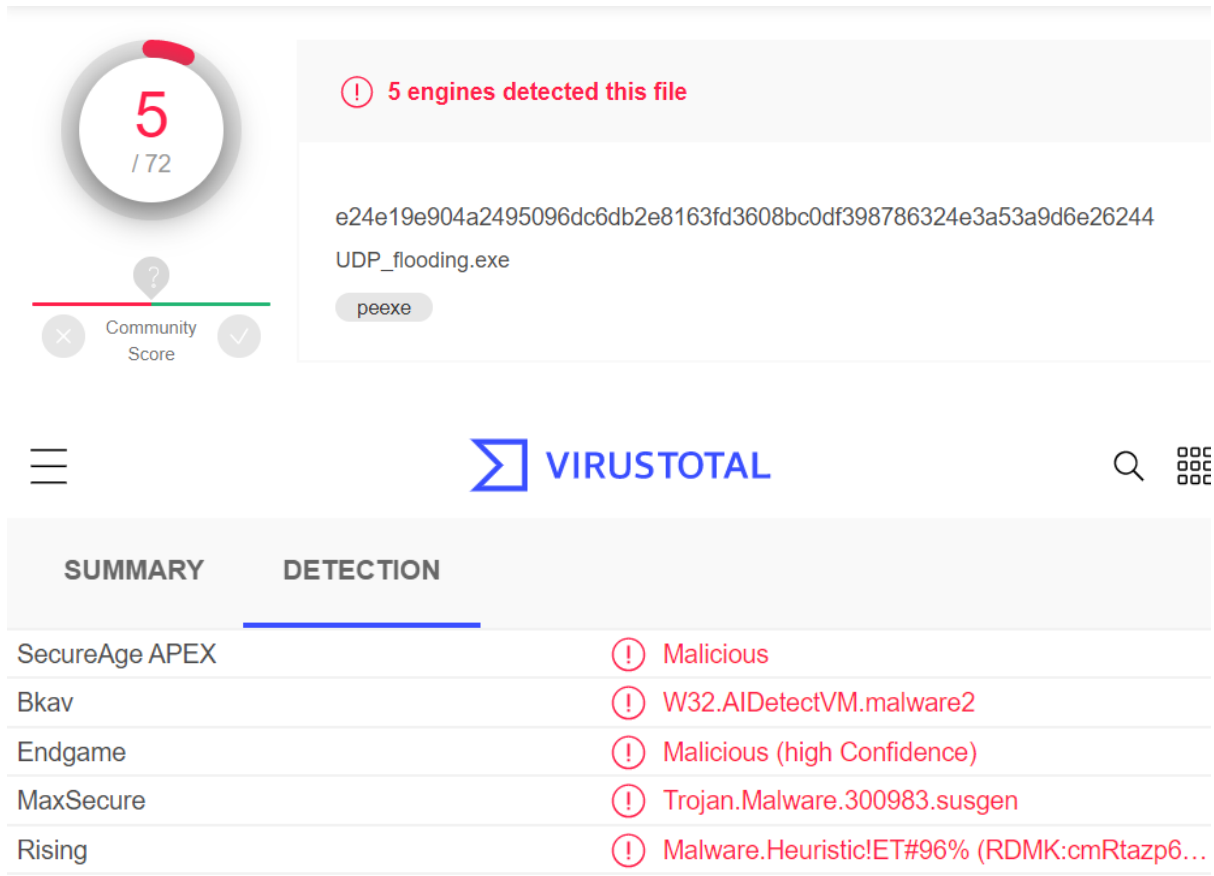
정보보호학과 2018111353 이승아

윈도우즈 보안과 악성코드 기초 _ 최은정 교수님

1. 정적 분석

프로그램을 실행해보기 전에 해당 프로그램의 악의적인 여부를 판단하기 위해 여러가지 정적 분석 도구를 사용해 보았다.

1) 악성코드 시그니처 판단 - VIRUSTOTAL

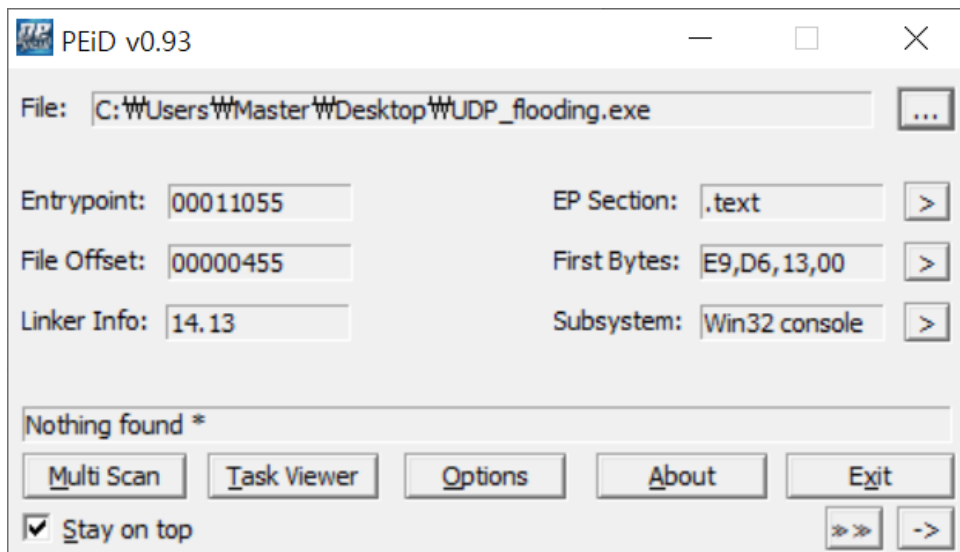


The image shows the VirusTotal analysis results for a file named 'UDP_flooding.exe'. The file's SHA256 hash is 'e24e19e904a2495096dc6db2e8163fd3608bc0df398786324e3a53a9d6e26244'. The analysis shows that 5 out of 72 engines detected the file as malicious. The detection results are as follows:

Engine	Detection Result
SecureAge APEX	Malicious
Bkav	W32.AIDetectVM.malware2
Endgame	Malicious (high Confidence)
MaxSecure	Trojan.Malware.300983.susgen
Rising	Malware.Heuristic!ET#96% (RDMK:cmRtazp6...

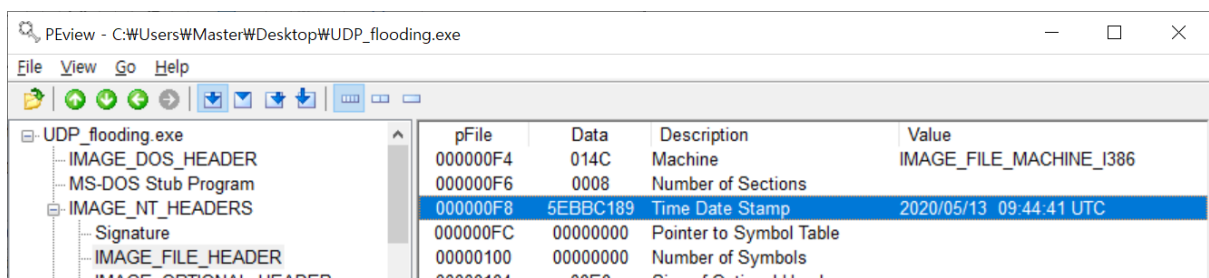
- 5개의 안티 바이러스 엔진에서 악성코드 시그니처가 발견되었다.
- 공통적으로 트로이목마와 관련된 악성코드임을 추측할 수 있는데, 이 프로그램은 겉보기에는 정상적인 프로그램으로 보이지만 실행 시 악성코드를 실행할 수 있는 가능성이 있다.
- malicious: 이 프로그램에 악의적인 요소가 있음을 추측할 수 있다.
- W32.AIDetectVM.malware2: 트로이목마 관련 시스템에 악성코드를 유발할 수 있음을 의미하는 시그니처이다.
- Trojan.Malware.300983.susgen: 이 프로그램이 시스템 파일을 수정, 새 바이러스 폴더를 생성하여 컴퓨터를 감염시키고 손상시키기 위해 윈도우즈 서비스를 설치할 수 있음을 의미한다.
- Malware.Heuristic! ET: Windows XP, Windows Vista, Windows 7, Windows 8 또는 Windows 10 컴퓨터에 악의적인 위협으로 작용하여 시스템 파일을 수정하고 새 폴더를 생성할 수 있는 가능성이 있다.

2) 패킹이나 난독화 흔적 - PEiD



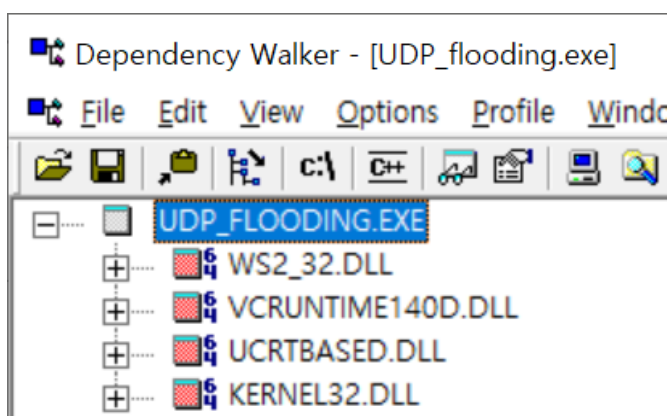
- 이 프로그램에 대한 정보를 제대로 확인할 수 없었다. 따라서 PE View를 사용하여 PE Header에 대해 더 자세한 정보를 찾아보았다.

3) PView

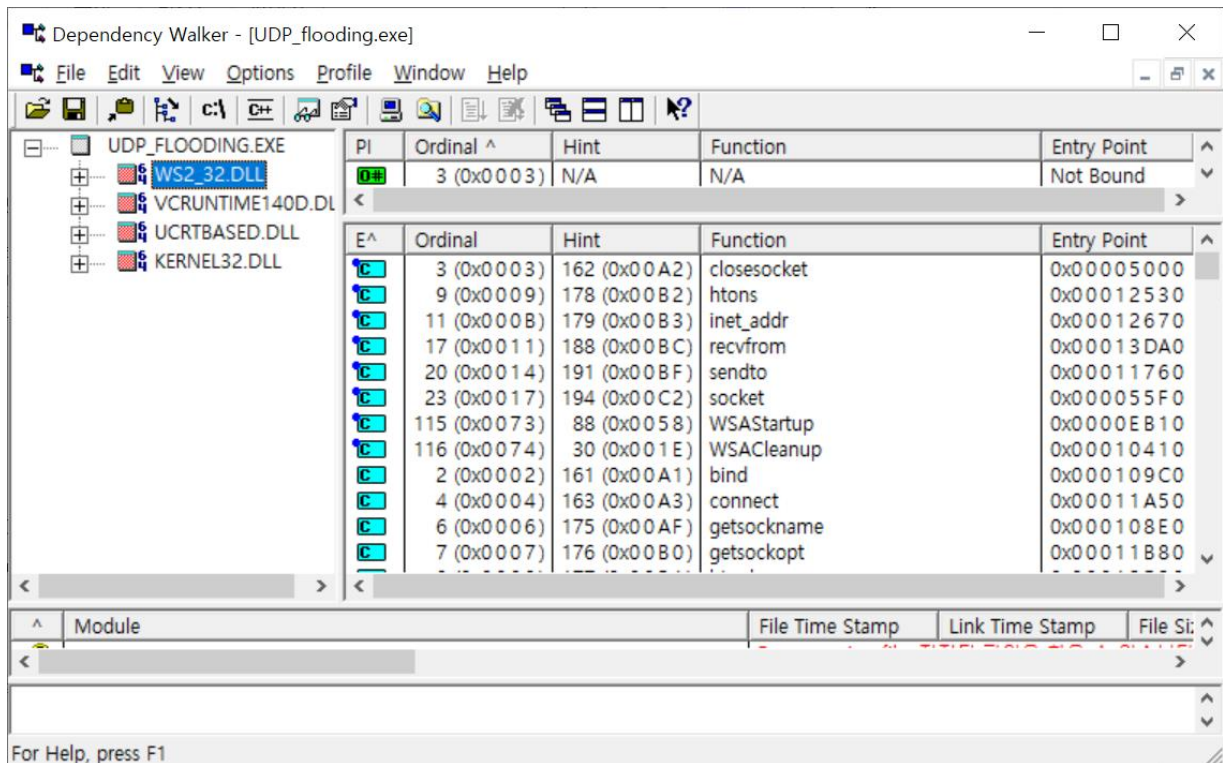


- 해당 프로그램이 2020년 5월 13일에 컴파일 되었음을 확인할 수 있다.

4) Dependency Walker



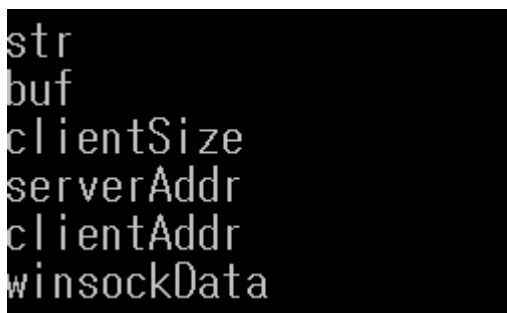
- 패킹되어있지 않기 때문에 어떤 API함수가 사용되었는지 확인할 수 있다.
- 다음과 같은 4개의 dll을 확인할 수 있다.



- 먼저 WS2_32.dll은 네트워크 연결을 처리하는데 사용되는 동적 라이브러리이다. 트로이목마를 설치하고 사용할 수 있으므로 문제가 될 수 있다. 위와 같이 소켓과 관련한 export 함수들을 확인할 수 있다. 윈도우에서 소켓을 사용하기 위해 동적으로 링크되어야 하는 라이브러리이기 때문에, 이 프로그램이 소켓을 사용하여 네트워크에서 어떠한 행위를 할 수 있음을 추측해볼 수 있다.
- Vcruntime140.dll은 Windows 운영 시스템을 위해 Microsoft에서 개발한 C RUNTIME LIBRARY와 관련된 파일이다. 이전에 PEiD로는 정보를 얻지 못했는데, 해당 프로그램이 C언어로 컴파일되었음을 추측할 수 있다.
- UCRTBASED.dll은 C:\Windows 또는 C:\Windows\System32 폴더에 있는 경우 위장한 멀웨어일 가능성이 있다고 한다. 따라서 동적 분석을 통해 ucrtbase.dll 프로세스를 검사하여 악성 행위를 하는지에 대해 자세히 알아볼 필요가 있다.

5) strings

이 프로그램이 소켓 통신과 관련된 프로그램임을 추측할 수 있는데 더 자세한 네트워크 기반의 증거를 찾아보기 위해 strings 도구를 사용해보았다.



- 우선 변수 이름으로 추정되는 여러 문자열들이 보이는데 서버 주소와 클라이언트 주소, 원속데이터 같은 문자열로 보았을 때 클라이언트와 서버 간의 원속을 통한 어떤 네트워크 통신이 있음을 예측할 수 있다.

```
IMHACKER_2018111353
1.
114.70.37.17
2.
sendto() error.
2018111353
```

- "IMHACKER_2018111353"이나 "2018111353"과 같은 문자열이 데이터로서 주고받아졌을 가능성이 있다. 또한 통신에 사용되었을 114.70.37.17이라는 명확한 아이피 주소까지 알 수 있다.

```
4.
The value of ESP was not properly saved across a function call.
This is usually a result of calling a function declared with
one calling convention with a function pointer declared with a
different calling convention.
6. UDP flooding
A cast to a smaller data type has caused a loss of data. If th
is was intentional, you should mask the source of the cast with
the appropriate bitmask. For example:
```

- "UDP_flooding"이라는 문자열이 눈에 띈다. 앞서 나온 단서들과 함께 추측해보았을 때 이 프로그램은 윈도우 소켓을 이용하여 UDP flooding이라는 악성 행위를 할 확률이 크다.

```
Stack around 'api-ms-win-core-registry-l1-1-0.dll' corrupted
bin\MSPDB140.DLL
VCRUNTIME140D.dll
api-ms-win-core-registry-l1-1-0.dll
advapi32.dll
RegOpenKeyExW
RegQueryValueExW
RegCloseKey
SOFTWAREWow6432Node\Microsoft\VisualStudio\14.0\Setup\VC
ProductDir
DLL
MSPDB140
MSPDB140
PDBOpenValidate5
recvfrom() error.
->
```

- 앞에서 dependency walker로 확인하지 못했던 advapi32.dll이 보인다. 이 라이브러리는 서비스 관리자나 레지스트리 같은 추가 핵심 윈도우 컴포넌트에 접근이 가능하며, 레지스트리, 시스템 종료와 재시작, 윈도우의 서비스의 시작/종료/생성, 계정 관리 등의 기능 지원하는 역할을 한다. 또한 지정

된 레지스트리 키를 오픈하는 RegOpenKeyEx 함수나 오픈된 레지스트리 키 값을 읽는 RegQueryValueEx 함수와 같은 문자열들을 보았을 때 이 프로그램이 레지스트리를 이용하여 특정 행위를 할 것으로 예측이 된다. 레지스트리에 어떤 값이 등록되고 어떤 변화가 나타나는지와 같은 내용을 중점으로 동적 분석을 해볼 필요가 있어 보인다.

- mspdb140(.dll)이나 VisualStudio와 같은 문자열이 들어간 경로를 보았을 때, 해당 프로그램이 visual studio로 컴파일되었을 가능성이 있다.

```
C:\Users\LG\source\repos\UDP_flooding\Debug\UDP_flooding.pdb
```

- 해당 프로그램이 다음과 같은 경로에서 실행되었을 것으로 추측된다.

```
_std_type_info_destroy_list
except_handler4_common
_vcrtd_GetModuleFileNameW
_vcrtd_GetModuleHandleW
_vcrtd_LoadLibraryExW
VCRUNTIME140D.dll
_acrt_iob_func
fgets
_stdio_common_vfprintf
strlen
_CrtDbgReport
_CrtDbgReportW
seh_filter_exe
set_app_type
_setusermatherr
configure_narrow_argv
initialize_narrow_environment
get_initial_narrow_environment
initterm
initterm_e
exit
exit
set_fmode
_p__argc
_p__argv
cexit
c_exit
register_thread_local_exe_atexit_callback
configthreadlocale
set_new_mode
_p__commode
strcpy_s
strcat_s
_stdio_common_vsprintf_s
seh_filter_dll
```

- 그외에도 다양한 문자열들이 보이는데, 이후 동적 분석과 리버싱을 통해 상세히 알아볼 필요가 있다고 생각했다.

2. 동적 분석

다음과 같이 프로그램을 더블클릭하여 단순 실행시켜 동적 분석 도구를 통해 행위를 관찰해 보았다.



1) Procmon으로 행위 판단

Process Monitor - Sysinternals: www.sysinternals.com					
File Edit Event Filter Tools Options Help					
Time	Process Name	PID	Operation	Path	
오후 5:00	UDP_flooding.exe	3876	Process Start		
오후 5:00	UDP_flooding.exe	3876	Thread Create		
오후 5:00	UDP_flooding.exe	3876	QueryNameInformationFile	C:\Documents and Settings\Administrator\바탕 화면\UDP_flooding.exe	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\Documents and Settings\Administrator\바탕 화면\UDP_flooding.exe	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\WINDOWS\system32\ntdll.dll	
오후 5:00	UDP_flooding.exe	3876	QueryNameInformationFile	C:\Documents and Settings\Administrator\바탕 화면\UDP_flooding.exe	
오후 5:00	UDP_flooding.exe	3876	CreateFile	C:\WINDOWS\Prefetch\UDP_FLOODING.EXE-06A66CB9.pf	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\UDP_flooding.exe	
오후 5:00	UDP_flooding.exe	3876	CreateFile	C:\Documents and Settings\Administrator\바탕 화면	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\WINDOWS\system32\kernel32.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\System\CurrentControlSet\Control\Terminal Server	
오후 5:00	UDP_flooding.exe	3876	RegQueryValue	HKLM\System\CurrentControlSet\Control\Terminal Server\TSAppCompat	
오후 5:00	UDP_flooding.exe	3876	RegCloseKey	HKLM\System\CurrentControlSet\Control\Terminal Server	
오후 5:00	UDP_flooding.exe	3876	FileSystemControl	C:\Documents and Settings\Administrator\바탕 화면	
오후 5:00	UDP_flooding.exe	3876	QueryOpen	C:\Documents and Settings\Administrator\바탕 화면\WS2_32.dll	
오후 5:00	UDP_flooding.exe	3876	QueryOpen	C:\WINDOWS\system32\ws2_32.dll	
오후 5:00	UDP_flooding.exe	3876	CreateFile	C:\WINDOWS\system32\ws2_32.dll	
오후 5:00	UDP_flooding.exe	3876	CreateFileMapping	C:\WINDOWS\system32\ws2_32.dll	
오후 5:00	UDP_flooding.exe	3876	CreateFileMapping	C:\WINDOWS\system32\ws2_32.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\System\CurrentControlSet\Control\SafeBoot\Option	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Policies\Microsoft\Windows\Safety\CodeIdentifiers	
오후 5:00	UDP_flooding.exe	3876	RegQueryValue	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safety\CodeIdentifiers\TransparentEnabled	
오후 5:00	UDP_flooding.exe	3876	RegCloseKey	HKLM\SOFTWARE\Policies\Microsoft\Windows\Safety\CodeIdentifiers	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKCU\Software\Policies\Microsoft\Windows\Safety\CodeIdentifiers	
오후 5:00	UDP_flooding.exe	3876	CloseFile	C:\WINDOWS\system32\ws2_32.dll	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\WINDOWS\system32\ws2_32.dll	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\WINDOWS\system32\advapi32.dll	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\WINDOWS\system32\rpcrt4.dll	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\WINDOWS\system32\secur32.dll	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\WINDOWS\system32\msvcr71.dll	
오후 5:00	UDP_flooding.exe	3876	QueryOpen	C:\Documents and Settings\Administrator\바탕 화면\WS2HELP.dll	
오후 5:00	UDP_flooding.exe	3876	QueryOpen	C:\WINDOWS\system32\ws2help.dll	
오후 5:00	UDP_flooding.exe	3876	CreateFile	C:\WINDOWS\system32\ws2help.dll	
오후 5:00	UDP_flooding.exe	3876	CreateFileMapping	C:\WINDOWS\system32\ws2help.dll	
오후 5:00	UDP_flooding.exe	3876	CreateFileMapping	C:\WINDOWS\system32\ws2help.dll	
오후 5:00	UDP_flooding.exe	3876	CloseFile	C:\WINDOWS\system32\ws2help.dll	
오후 5:00	UDP_flooding.exe	3876	Load Image	C:\WINDOWS\system32\ws2help.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\Secur32.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\RPCRT4.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ADVAPI32.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	
오후 5:00	UDP_flooding.exe	3876	RegQueryValue	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon\LeakTrack	
오후 5:00	UDP_flooding.exe	3876	RegCloseKey	HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Winlogon	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Diagnostics	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\msvcr71.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\WS2HELP.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\WS2_32.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\ntdll.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Image File Execution Options\kernel32.dll	
오후 5:00	UDP_flooding.exe	3876	RegOpenKey	HKLM\System\CurrentControlSet\Control\Session Manager	
오후 5:00	UDP_flooding...	620	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\ RNG\Seed	SUCCESS Type: REG_BINARY, Length: 80, Data: 8A 33 7B
오후 5:00	UDP_flooding...	620	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\ RNG\Seed	SUCCESS Type: REG_BINARY, Length: 80, Data: AD 0D 7C
오후 5:00	UDP_flooding...	620	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\ RNG\Seed	SUCCESS Type: REG_BINARY, Length: 80, Data: 4B E0 CD
오후 5:00	UDP_flooding...	620	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\ RNG\Seed	SUCCESS Type: REG_BINARY, Length: 80, Data: 40 72 A8
오후 5:00	UDP_flooding...	620	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\ RNG\Seed	SUCCESS Type: REG_BINARY, Length: 80, Data: 24 68 D3
오후 5:00	UDP_flooding...	620	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\ RNG\Seed	SUCCESS Type: REG_BINARY, Length: 80, Data: A7 5E E9
오후 5:00	UDP_flooding...	620	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\ RNG\Seed	SUCCESS Type: REG_BINARY, Length: 80, Data: 89 73 05
오후 5:00	UDP_flooding...	620	RegSetValue	HKLM\SOFTWARE\Microsoft\Cryptography\ RNG\Seed	SUCCESS Type: REG_BINARY, Length: 80, Data: 0D A1 0B

- RegSetValue 함수를 사용하는 프로세스 행위를 관찰해 보았다. RegSetValue는 레지스트리의 값을 저장하는 행위를 뜻하며 악성행위와 크게 연관있는 유의미한 부분은 찾아내지 못했다.

Modules:

Module	Address	Size	Path	Company
UDP_flooding.exe	0x400000	0x1c000	C:\Windows\Documents and Settings\Administrator\...	
hnetcfg.dll	0x65cb0000	0x56000	C:\Windows\system32\hnetcf...	Microsoft Corpo..
mswsock.dll	0x71980000	0x3f000	C:\Windows\system32\mswso...	Microsoft Corpo..
ws2help.dll	0x719d0000	0x8000	C:\Windows\system32\ws2hel...	Microsoft Corpo..
ws2_32.dll	0x719e0000	0x17000	C:\Windows\system32\ws2_3...	Microsoft Corpo..
imm32.dll	0x762e0000	0x1d000	C:\Windows\system32\imm32.d...	Microsoft Corpo..
msvcrt.dll	0x77bc0000	0x58000	C:\Windows\system32\msvcrt.d...	Microsoft Corpo..
user32.dll	0x77cf0000	0x90000	C:\Windows\system32\user32...	Microsoft Corpo..
rpcrt4.dll	0x77d80000	0x92000	C:\Windows\system32\rpcrt4.d...	Microsoft Corpo..
gdi32.dll	0x77e20000	0x49000	C:\Windows\system32\gdi32.d...	Microsoft Corpo..
secur32.dll	0x77ef0000	0x11000	C:\Windows\system32\secur3...	Microsoft Corpo..
advapi32.dll	0x77f50000	0xa8000	C:\Windows\system32\advapi...	Microsoft Corpo..
kernel32.dll	0x7c800000	0x130000	C:\Windows\system32\kernel...	Microsoft Corpo..
ntdll.dll	0x7c930000	0x9b000	C:\Windows\system32\ntdll.d...	Microsoft Corpo..

- 위와 같이 프로그램이 로딩한 라이브러리를 확인할 수 있다. 정적 분석에서 dependency walker와 strings로 볼 수 있었던 dll들이 실제로 동작하고 있었음을 알 수 있다.

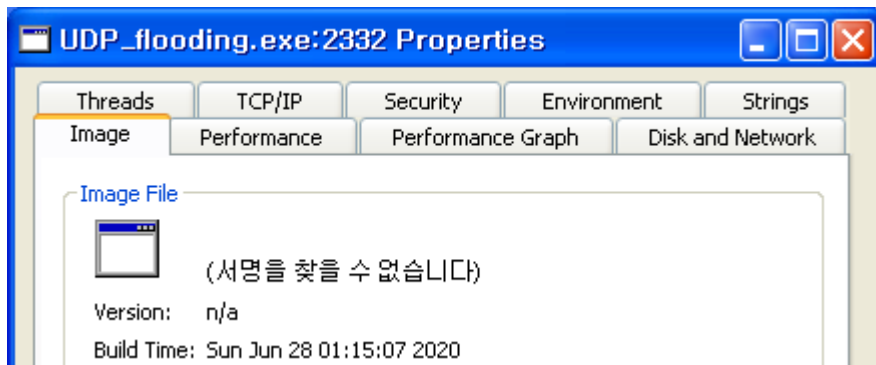
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:10004	SUCCESS	Length: 19
오류 ...	UDP_flooding.exe	3876	UDP Receive	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:10004	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30
오류 ...	UDP_flooding.exe	3876	UDP Unknown	gowoon-d31936d6,localdomain:1324 -> 114.70.37.17:7777	SUCCESS	Length: 30

- 정적 분석을 통해 udp 통신을 통해 악성행위를 할 가능성이 크다고 추측했는데 동적 분석을 통해 실제 udp 통신을 하는 듯한 흔적도 발견할 수 있었다. 통신 대상의 아이피 주소는 114.70.37.17이고 포트번호는 10004이다. 처음에 해당 주소로 길이 19의 데이터를 전송하고 그 대상으로부터 길이 30의 데이터를 받는다. 그 후에 같은 아이피 주소의 7777번 포트로 길이 30의 데이터를 10번 보내는 것으로 추측해 볼 수 있다. 이러한 반복 행위가 앞서 발견한 문자열 UDP flooding과 큰 관련이 있을 것 같다. 위의 행위에 대한 자세한 내용은 네트워크 행위를 좀 더 구체적으로 관찰할 수 있는 wireshark와 같은 도구를 사용해 알아보아야겠다.

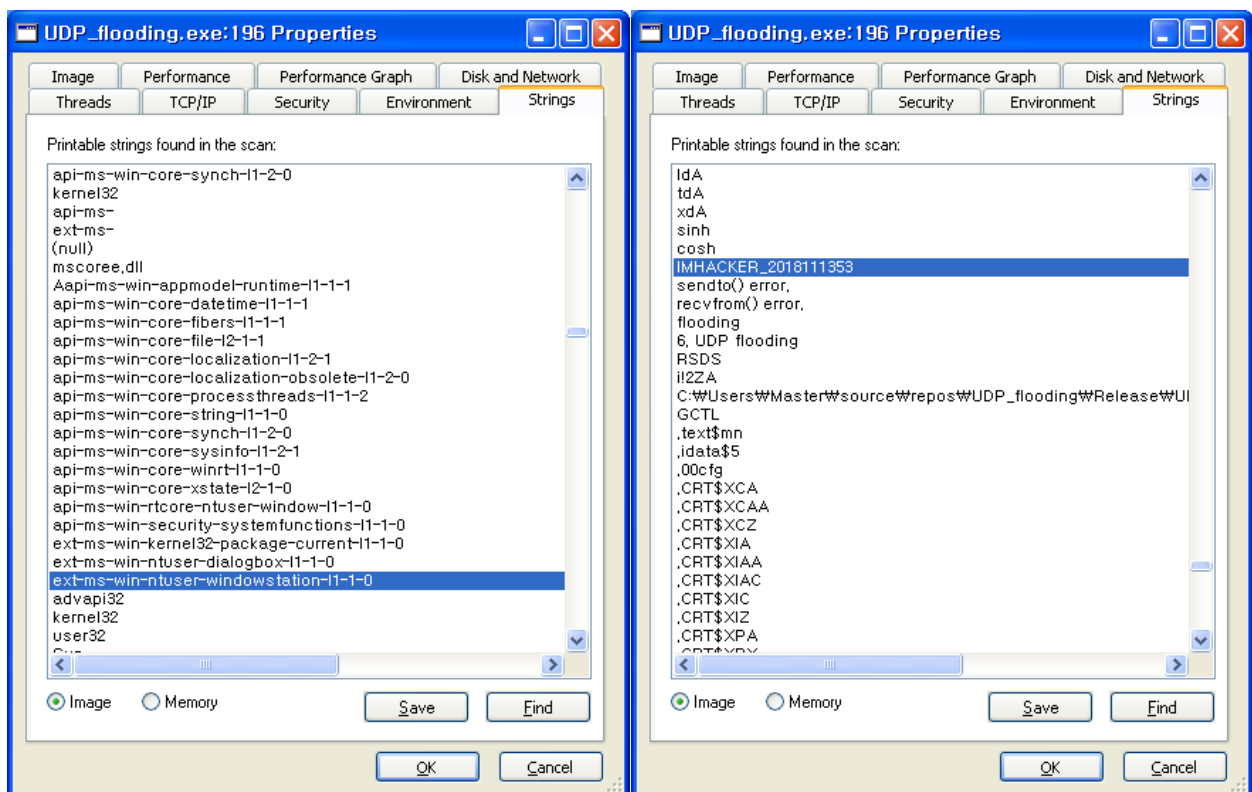
2) Process Explorer로 행위 관찰

explorer.exe	9,38	21,424 K	25,120 K	1596 Windows Explorer	Microsoft Corporation
rundll32.exe		2,400 K	3,776 K	1888 Run a DLL as an App	Microsoft Corporation
vmtoolsd.exe		10,604 K	15,228 K	1896 VMware Tools Core Ser...	VMware, Inc.
ctfmon.exe		964 K	3,496 K	1924 CTF Loader	Microsoft Corporation
Procmon.exe		14,480 K	17,360 K	2540 Process Monitor	Sysinternals - www.s...
procexp.exe		15,580 K	18,688 K	2020 Sysinternals Process E...	Sysinternals - www.s...
procexp.exe	1,56	16,076 K	18,536 K	1848 Sysinternals Process E...	Sysinternals - www.s...
UDP_flooding.exe	3,13	488 K	1,812 K	804	
conime.exe		956 K	3,328 K	2216 Console IME	Microsoft Corporation

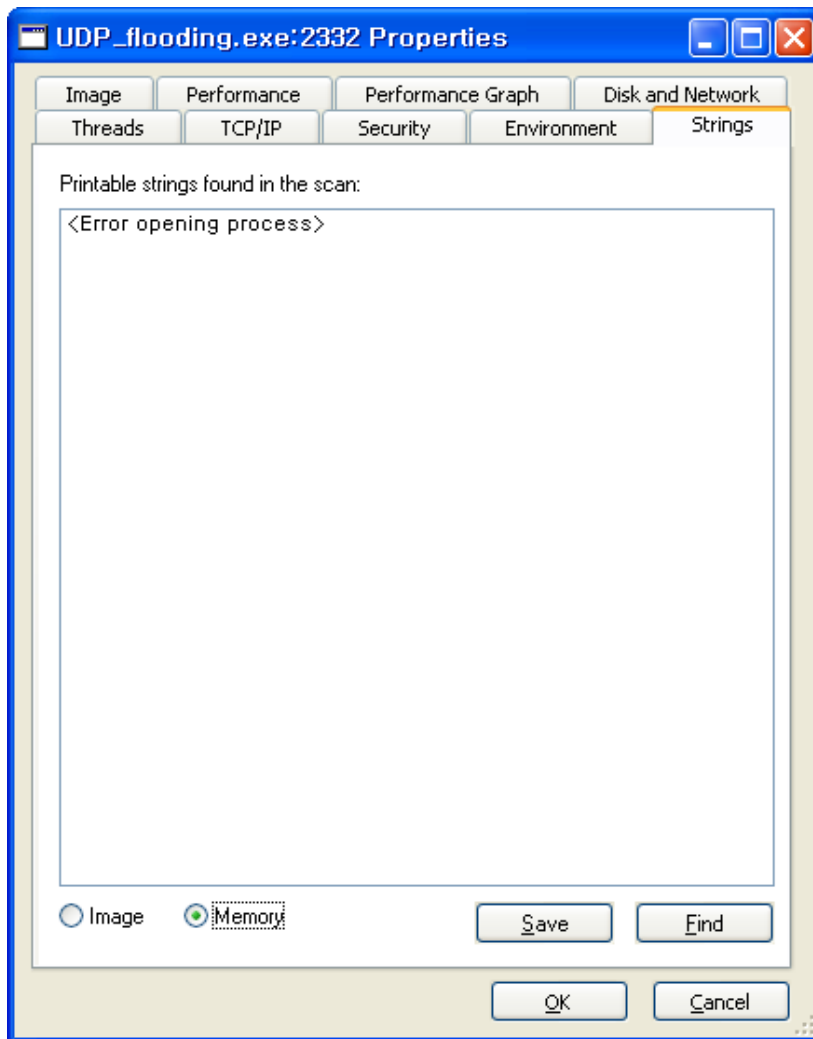
- UDP_flooding.exe을 더블 클릭하여 실행하였더니 해당 프로세스가 Process Explorer 화면 상에서 잠깐 나타났다가 곧바로 사라졌다. 아주 단시간에 실행되고 종료되는 프로세스임을 알 수 있다.



- 이 프로세스의 속성을 확인해보니, 서명을 찾을 수 없었다. 마이크로소프트에서 서명한 바이너리가 아니라는 뜻으로, 검증되지 않은 악성 프로그램일 확률이 크다는 것이다.



- Strings에서 Image 탭에서는 하드디스크에 파일이 존재할 때의 문자열을 볼 수 있다. 정적 분석 시에 Strings 도구로 확인했던 문자열들과 비슷했다.



- Memory 탭에서 볼 수 있는 활성화된 메모리상의 문자열은 확인할 수 없었다. 악성코드일 경우 메모리에 로드되었을 때 정보가 더 많이 보일 수 있는데 이 프로그램에서의 해당 사항은 없었다. 정적 분석에서 발견했듯이 겉보기엔 정상 프로그램인 것처럼 보이는 트로이목마의 속성을 가졌을 가능성이 있다.

3) Regshot

```

Regshot 1.8.3-beta2
Comments:
Datetime: 2020/6/28 08:26:54 , 2020/6/28 08:26:56
Computer: G0U00N-D31936D6 , G0U00N-D31936D6
Username: Administrator , Administrator

-----
Values added:2
HKU\S-1-5-21-343818398-1647877149-1801674531-5000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75948700-EF1F-11D0-9888-006097DEACF9}\Count\HRZR_EHACNGU:P:Wqbphzragi
HKU\S-1-5-21-343818398-1647877149-1801674531-5000\Software\Microsoft\Windows\Shell\NoRoam\HUICache\WC:Documents and Settings\Administrator\바탕 화면\UDP_flooding.exe: "UDP_flooding"

-----
Values modified:5
HKLM\SOFTWARE\Microsoft\Cryptography\ RNGSeed: 42 65 59 91 3E 08 37 20 2C E0 BF E8 00 A2 8D 95 C4 64 F3 12 5F 93 A6 01 2C F9 11 E3 35 C8 BA 57 71 3C 21 34 85 F1 D5 ED 6C E0 EA 73 3;
HKLM\SOFTWARE\Microsoft\Cryptography\ RNGSeed: 8D A7 22 CE BC 63 8F 7A 56 78 CD 14 17 09 DD B1 FB 2A 14 F1 F0 90 E8 1D D0 38 20 62 E9 81 08 EB 50 18 97 F0 78 16 68 81 A7 13 07 3E 9;
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Prefetcher\TracesProcessed: 0x0000006D
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Prefetcher\TracesProcessed: 0x0000006F
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Prefetcher\TracesSuccessful: 0x00000016
HKLM\SOFTWARE\Microsoft\Windows NT\CurrentVersion\Prefetcher\TracesSuccessful: 0x00000017
HKU\S-1-5-21-343818398-1647877149-1801674531-5000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75948700-EF1F-11D0-9888-006097DEACF9}\Count\HRZR_EHACNGU: 01 00 00 0;
HKU\S-1-5-21-343818398-1647877149-1801674531-5000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75948700-EF1F-11D0-9888-006097DEACF9}\Count\HRZR_EHACNGU: 01 00 00 0;
HKU\S-1-5-21-343818398-1647877149-1801674531-5000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75948700-EF1F-11D0-9888-006097DEACF9}\Count\HRZR_HUFPHG: 01 00 00 00;
HKU\S-1-5-21-343818398-1647877149-1801674531-5000\Software\Microsoft\Windows\CurrentVersion\Explorer\UserAssist\{75948700-EF1F-11D0-9888-006097DEACF9}\Count\HRZR_HUFPHG: 01 00 00 00

-----
Total changes:7
  
```

- Procmon을 통해 볼 수 있었던 레지스트리 관련 행위를 Regshot 도구를 통해 더 자세히 확인할

수 있었다. 레지스트리 상에서의 변화된 부분은 총 7개이다.

4) CurrPorts

System	4	UDP	137	netbios-ns	192, 168, 44, 129
System	4	UDP	138	netbios-dgm	192, 168, 44, 129
System	4	UDP	445	microsoft-ds	0, 0, 0, 0
UDP_flooding.exe	872	UDP	1536		0, 0, 0, 0

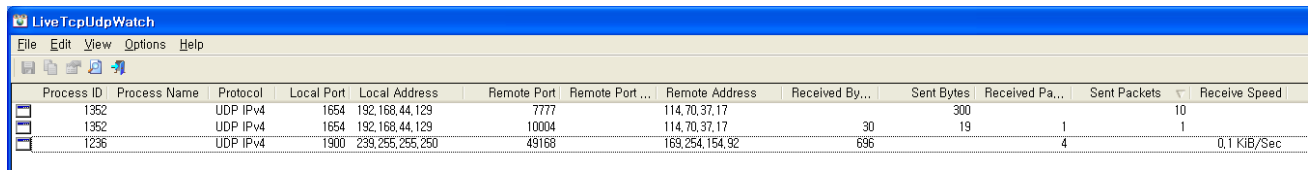
- CurrPorts는 로컬 컴퓨터에서 현재 열려 있는 모든 TCP/IP 및 UDP 포트 목록을 표시하는 네트워크 모니터링 툴이다. 앞에서 UDP 네트워크 통신을 통해 악성행위를 할 가능성을 알아냈기 때문에 이 부분에 초점을 맞춰 UDP 통신을 위한 준비작업이 있는지를 확인해 보았다.

The screenshot shows a 'Properties' window for the process 'UDP_flooding.exe'. The window contains the following information:

- Process Name: UDP_flooding.exe
- Process ID: 2476
- Protocol: UDP
- Local Port: 1652
- Local Port Name:
- Local Address: 0.0.0.0
- Remote Port:
- Remote Port Name:
- Remote Address:
- Remote Host Name:
- State:
- Sent Bytes:
- Received Bytes:
- Sent Packets:
- Received Packets:
- Process Path: C:\Documents and Settings\Administrator\바탕화면
- Product Name:
- File Description:
- File Version:
- Company:
- Process Created On: 2020-06-28 오후 6:13:55
- User Name: GOWOON-D31936D6\Administrator
- Process Services:
- Process Attributes: A
- Added On: 2020-06-28 오후 6:13:55
- Creation Timestamp: 2020-06-28 오후 6:13:55
- Module Filename: C:\Documents and Settings\Administrator\바탕화면
- Remote IP Country:
- Remote IP ASN:
- Remote IP Company:
- Window Title: C:\Documents and Settings\Administrator\바탕화면

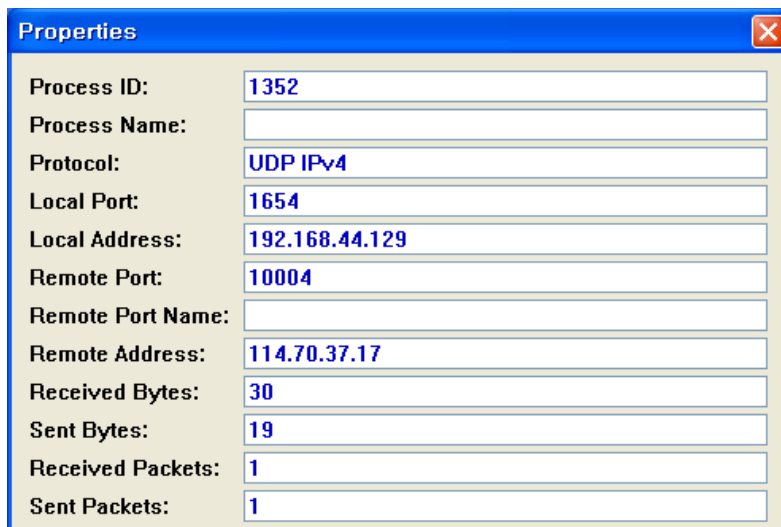
- 1652번 포트를 통해 UDP연결을 위한 작업을 했을 것으로 보인다.

5) LiveTcpUdpWatch



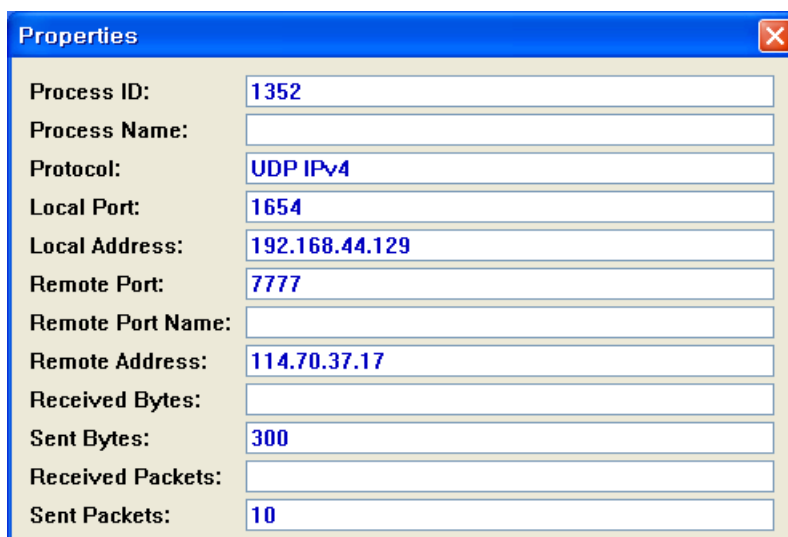
Process ID	Process Name	Protocol	Local Port	Local Address	Remote Port	Remote Port ...	Remote Address	Received By...	Sent Bytes	Received Pa...	Sent Packets	Receive Speed
1352		UDP IPv4	1654	192.168.44.129	7777		114.70.37.17		300		10	
1352		UDP IPv4	1654	192.168.44.129	10004		114.70.37.17	30	19	1	1	
1236		UDP IPv4	1900	239.255.255.250	49168		169.254.154.92	696		4		0.1 KiB/Sec

- LiveTcpUdpWatch는 시스템의 모든 TCP 및 UDP 활동에 대한 실시간 정보를 표시하는 Windows 용 도구라고 한다. UDP_flooding.exe을 실행해본 결과 다음과 같이 UDP 통신을 한 아이피 주소와 포트 번호, 주고받은 바이트 수 및 패킷의 개수를 알 수 있었다.



Properties	
Process ID:	1352
Process Name:	
Protocol:	UDP IPv4
Local Port:	1654
Local Address:	192.168.44.129
Remote Port:	10004
Remote Port Name:	
Remote Address:	114.70.37.17
Received Bytes:	30
Sent Bytes:	19
Received Packets:	1
Sent Packets:	1

- 114.70.37.17:10004의 주소에 19 바이트의 패킷 하나를 보내고 30 바이트의 패킷 하나를 받은 것을 확인할 수 있다. 이는 Procmon에서 확인한 결과와 일치한다.



Properties	
Process ID:	1352
Process Name:	
Protocol:	UDP IPv4
Local Port:	1654
Local Address:	192.168.44.129
Remote Port:	7777
Remote Port Name:	
Remote Address:	114.70.37.17
Received Bytes:	
Sent Bytes:	300
Received Packets:	
Sent Packets:	10

- 이후 같은 아이피 주소의 다른 포트 번호인 7777로 300 바이트의 총 10개의 패킷을 보내는 것을 확인할 수 있다. 114.70.37.17:10004 주소로부터 포트번호 7777라는 주소 정보를 받아 그 주소로 UDP flooding 공격을 하는 행위로 예측해볼 수 있다.

6) Wireshark

Capturing from 로컬 영역 연결 [Wireshark 1.12.0 (v1.12.0-0-g4fab41a from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

Filter: udp Expression... Clear Apply Save

No.	Time	Source	Destination	Protocol	Length	Info
1	0.00000000	192.168.44.129	114.70.37.17	UDP	61	Source port: 1657 Destination port: 10004
2	0.02267100	114.70.37.17	192.168.44.129	UDP	72	Source port: 10004 Destination port: 1657
3	0.02427900	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
4	0.02452000	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
5	0.02467600	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
6	0.02484500	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
7	0.02501900	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
8	0.02518700	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
9	0.02533200	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
10	0.02548400	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
11	0.02566300	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777
12	0.02581400	192.168.44.129	114.70.37.17	UDP	72	Source port: 1657 Destination port: 7777

- wireshark에서 필터를 udp로 설정하여 통신 내용을 자세히 살펴보았다.

```

source: 192.168.44.129 (192.168.44.129)
destination: 114.70.37.17 (114.70.37.17)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
[-] User Datagram Protocol, Src Port: 1038 (1038), Dst Port: 10004 (10004)
  Source Port: 1038 (1038)
  Destination Port: 10004 (10004)
  Length: 27
  [+ Checksum: 0xd6ed [validation disabled]
    [Stream index: 0]
[-] Data (19 bytes)
  Data: 494d4841434b45525f32303138313131333533
  [Length: 19]

```

0000	00 50 56 ff f2 e8 00 0c	29 12 32 47 08 00 45 00	.PV.....).2G..E.
0010	00 2f 01 bf 00 00 80 11	b4 7e c0 a8 2c 81 72 46	./.....~....rF
0020	25 11 04 0e 27 14 00 1b	d6 ed 49 4d 48 41 43 4b	%...'...IMHACK
0030	45 52 5f 32 30 31 38 31	31 31 33 35 33	ER_20181 11353

- 먼저 악성 프로그램이 실행된 로컬 컴퓨터에서 114.70.37.17:10004의 주소로 'IMHACKER_2018111353'이라는 데이터를 전송한다. 이 데이터는 strings를 통해 의미있게 보았던 문자열과 일치한다.

```

Source: 114.70.37.17 (114.70.37.17)
Destination: 192.168.44.129 (192.168.44.129)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
[- User Datagram Protocol], Src Port: 10004 (10004), Dst Port: 1038 (1038)
  Source Port: 10004 (10004)
  Destination Port: 1038 (1038)
  Length: 38
  [+ Checksum: 0xbf7e [validation disabled]
    [Stream index: 0]
[- Data (30 bytes)
  Data: 4f4b41595f3131342e37302e33372e31373a373737370a00...
  [Length: 30]
0000  00 0c 29 12 32 47 00 50 56 ff f2 e8 08 00 45 00 ..).2G.P V.....E.
0010  00 3a 78 81 00 00 80 11 3d b1 72 46 25 11 c0 a8 .:x.....=.rF%...
0020  2c 81 27 14 04 0e 00 26 bf 7e 4f 4b 41 59 5f 31 ,.'....& .~OKAY_1
0030  31 34 2e 37 30 2e 33 37 2e 31 37 3a 37 37 37 37 14.70.37 .17:7777
0040  0a 00 00 00 00 00 00 00 .....

```

- 그 다음 114.70.37.17:10004의 주소에서 로컬 컴퓨터로 'OKAY_114.70.37.17:7777'이라는 데이터를 보낸다. 이는 114.70.37.17:10004의 주소에서 'IMHACKER_2018111353'라는 데이터를 보낸 로컬 주소가 공격자 주소라는 것을 확인하고 공격 대상의 주소를 알려주는 것으로 추측할 수 있다.

```

Source: 192.168.44.129 (192.168.44.129)
Destination: 114.70.37.17 (114.70.37.17)
[Source GeoIP: Unknown]
[Destination GeoIP: Unknown]
[- User Datagram Protocol], Src Port: 1038 (1038), Dst Port: 7777 (7777)
  Source Port: 1038 (1038)
  Destination Port: 7777 (7777)
  Length: 38
  [+ Checksum: 0x395b [validation disabled]
    [Stream index: 1]
[- Data (30 bytes)
  Data: 323031383131313335330000362e2055445020666c6f6f64...
  [Length: 30]
0000  00 50 56 ff f2 e8 00 0c 29 12 32 47 08 00 45 00 .PV.....).2G..E.
0010  00 3a 01 c0 00 00 80 11 b4 72 c0 a8 2c 81 72 46 .:......r...rF
0020  25 11 04 0e 1e 61 00 26 39 5b 32 30 31 38 31 31 %....a.& 9[201811
0030  31 33 35 33 00 00 36 2e 20 55 44 50 20 66 6c 6f 1353..6. UDP flo
0040  6f 64 69 6e 67 20 bc ba oding ..

```

- 로컬 컴퓨터는 114.70.37.17:7777로 다음과 같은 패킷을 10번 반복하여 전송한다. 대상 주소는 114.70.37.17:10004에서 알려준 주소와 일치한다. 보내는 데이터는 '2018111353'와 같은 문자열로 볼 수 있으며 뒤의 UDP flooding이라는 문자열을 보았을 때 위의 10번의 패킷 전송이 UDP flooding 공격인 것으로 볼 수 있다.

3. 리버싱

리버싱을 하면 악성코드의 행위에 대하여 더 자세하게 분석할 수 있다. Ollydbg 툴을 활용하여 해당 악성 프로그램을 리버싱해보았다. (중간에 프로그램의 오류로 Ollydbg와 Ollydbg shadow를 함께 사용하였다.)

004014ED	. 59	POP ECX
004014EE	> E8 2E440000	CALL UDP_floo.00405921
004014F3	. 8B38	MOV EDI,DWORD PTR DS:[EAX]
004014F5	. E8 21440000	CALL UDP_floo.0040591B
004014FA	. 8BF0	MOV ESI,EAX
004014FC	. E8 5E400000	CALL UDP_floo.0040555F
00401501	. 50	PUSH EAX
00401502	. 57	PUSH EDI
00401503	. FF36	PUSH DWORD PTR DS:[ESI]
00401505	. E8 46FBFFFF	CALL UDP_floo.00401050

먼저 제일 처음 00401505부분에서 함수코드 내부로 들어가보았다.

004010B3	. 2BF1	SUB ESI,ECX	[pWSAData RequestedVersion = 202 (2.2.) WSAStartup
004010B5	. 50	PUSH EAX	
004010B6	. 68 02020000	PUSH 202	
004010BB	. FF15 14114100	CALL DWORD PTR DS:[<&WS2_32.#115>]	[Arg1 = 004165BC UDP_floo.00401010 WSACleanup
004010C1	. 83F8 FF	CMP EAX,-1	
004010C4	. 75 2A	JNZ SHORT UDP_floo.004010F0	
004010C6	. 68 BC654100	PUSH UDP_floo.004165BC	[Arg1 = 004165D0 UDP_floo.00401010
004010CB	. E8 40FFFFFF	CALL UDP_floo.00401010	
004010D0	. 83C4 04	ADD ESP,4	
004010D3	. FF15 0C114100	CALL DWORD PTR DS:[<&WS2_32.#116>]	[Arg1 = 004165D0 UDP_floo.00401010
004010D9	. 83C8 FF	OR EAX,FFFFFFFF	
004010DC	. 5F	POP EDI	
004010DD	. 5E	POP ESI	
004010DE	. 8B8C24 D40300	MOV ECX,DWORD PTR SS:[ESP+3D4]	
004010E5	. 33CC	XOR ECX,ESP	
004010E7	. E8 4E020000	CALL UDP_floo.0040133A	
004010EC	. 8BE5	MOV ESP,EBP	
004010EE	. 5D	POP EBP	
004010EF	. C3	RETN	
004010F0	> 68 D0654100	PUSH UDP_floo.004165D0	
004010F5	. E8 16FFFFFF	CALL UDP_floo.00401010	
004010FA	. 83C4 04	ADD ESP,4	

다음과 같이 소켓 통신을 준비하기 위한 윈속 초기화 과정에 사용되는 함수들이 실행되는 것을 확인할 수 있다.

004010EE	. 5D	POP EBP	[Arg1 = 004165D0 UDP_floo.00401010
004010EF	. C3	RETN	
004010F0	> 68 D0654100	PUSH UDP_floo.004165D0	
004010F5	. E8 16FFFFFF	CALL UDP_floo.00401010	[pAddr = UDP_floo.004165E8 inet_addr NetShort = 2714 ntohs Protocol = IPPROTO_UDP Type = SOCK_DGRAM Family = AF_INET socket
004010FA	. 83C4 04	ADD ESP,4	
004010FD	. 0F57C0	XORPS XMM0,XMM0	
00401100	. B8 02000000	MOV EAX,2	
00401105	. 66:0F134424 1	MOVLPS QWORD PTR SS:[ESP+18],XMM0	
0040110B	. 66:294424 20	MOVAPS QWORD PTR SS:[ESP+20],XMM0	
00401110	. 66:894424 10	MOV WORD PTR SS:[ESP+10],AX	
00401115	. 68 E8654100	PUSH UDP_floo.004165E8	
0040111A	. FF15 08114100	CALL DWORD PTR DS:[<&WS2_32.#11>]	
00401120	. 68 14270000	PUSH 2714	
00401125	. 894424 18	MOV DWORD PTR SS:[ESP+18],EAX	
00401129	. FF15 24114100	CALL DWORD PTR DS:[<&WS2_32.#9>]	
0040112F	. 6A 11	PUSH 11	
00401131	. 6A 02	PUSH 2	
00401133	. 6A 02	PUSH 2	
00401135	. 66:894424 1E	MOV WORD PTR SS:[ESP+1E],AX	
0040113A	. FF15 1C114100	CALL DWORD PTR DS:[<&WS2_32.#23>]	
00401140	. 8BF8	MOV EDI,EAX	

소켓의 타입으로 보아 UDP 프로토콜을 사용하는 소켓을 생성하였다.

00401174	> 68 0C664100	PUSH UDP_floo.0041660C	[Arg1 = 0041660C UDP_floo.00401010
00401179	. E8 92FFFFFF	CALL UDP_floo.00401010	
0040117E	. 83C4 04	ADD ESP,4	

위의 호출이 끝나면 소켓 생성에 성공한다.

004011A2	. 68 34664100	PUSH UDP_floo.00416634	[Arg1 = 00416634 UDP_floo.00401010
004011A7	. E8 64FEFFFF	CALL UDP_floo.00401010	
004011AC	. 83C4 04	ADD ESP,4	

004011A7부분에서 함수 내부로 들어가보았다.

00401181	. 8D4424 10	LEA EAX,DWORD PTR SS:[ESP+10]	[ToLength = 10 (16.) pTo Flags = 0 DataSize Data Socket sendto
00401185	. 6A 10	PUSH 10	
00401187	. 50	PUSH EAX	
00401188	. 6A 00	PUSH 0	
0040118A	. 56	PUSH ESI	
0040118B	. 8D8424 E00100	LEA EAX,DWORD PTR SS:[ESP+1E0]	
00401192	. 50	PUSH EAX	
00401193	. 57	PUSH EDI	
00401194	. FF15 18114100	CALL DWORD PTR DS:[<&WS2_32.#20>]	
00401195	. 90	CMP EAX,ESI	

0012FB78	00000064	Socket = 64	Registers (FPU)
0012FB7C	0012FD60	Data = 0012FD60	EAX 0012FD60 ASCII "IMHACKER_2018111353"
0012FB80	00000013	DataSize = 13 (19.)	ECX 0012FB2C
0012FB84	00000000	Flags = 0	EDX 00418098 UDP_floo.00418098
0012FB88	0012FBA0	pTo = 0012FBA0	EBX 7FFDE000
0012FB8C	00000010	ToLength = 10 (16.)	ESP 0012FB78
0012FB90	00155A50		EBP 0012FF78
0012FB94	00418CEC	UDP_floo.00418CEC	ESI 00000013
0012FB98	00000064		EDI 00000064

sendto() 함수를 사용하여 "IMHACKER_2018111353"이라는 데이터를 소켓을 통해 전송하는 것을 확인할 수 있다. 위의 호출이 끝나면 데이터 전송이 완료된다.

00401208	. vE9 0B010000	JMP UDP_floo.0040131B	[Arg1 = 0041667C UDP_floo.00401010
00401210	. > 68 7C664100	PUSH UDP_floo.0041667C	
00401215	. E8 F6FDFFFF	CALL UDP_floo.00401010	
0040121A	. 83C4 04	ADD ESP,4	
0040121B	. 8D8424 D00100	LEA EAX,DWORD PTR SS:[ESP+1D0]	

004011D0	. 83C4 14	ADD ESP,14	[pFromLen pFrom Flags = 0 BufSize = 200 (512.) Buffer Socket recvfrom
004011E0	. 8D4424 0C	LEA EAX,DWORD PTR SS:[ESP+C]	
004011E4	. 50	PUSH EAX	
004011E5	. 8D4424 24	LEA EAX,DWORD PTR SS:[ESP+24]	
004011E9	. 50	PUSH EAX	
004011EA	. 6A 00	PUSH 0	
004011EC	. 68 00020000	PUSH 200	
004011F1	. 8D8424 E00100	LEA EAX,DWORD PTR SS:[ESP+1E0]	
004011F8	. 50	PUSH EAX	
004011F9	. 57	PUSH EDI	
004011FA	. FF15 20114100	CALL DWORD PTR DS:[<&WS2_32.#17>]	
00401200	. 8BF8	MOV EDI,EAX	

Registers (FPU)	
EAX	0012FD60 ASCII "OKAY_114.70.37.17:7777"
ECX	0012FB2C
EDX	00418098 UDP_floo.00418098
EBX	7FFDE000

위의 호출에서 소켓 통신으로 recvfrom() 함수를 통해 데이터를 받는다. 받은 데이터는 Buffer에 저장되며 데이터의 내용은 "OKAY_114.70.37.17" 이다.

00401208	. vE9 0B010000	JMP UDP_floo.0040131B	[Arg1 = 0041667C UDP_floo.00401010
00401210	. > 68 7C664100	PUSH UDP_floo.0041667C	
00401215	. E8 F6FDFFFF	CALL UDP_floo.00401010	
0040121A	. 83C4 04	ADD ESP,4	
0040121D	. 8D8424 D00100	LEA EAX,DWORD PTR SS:[ESP+1D0]	
00401224	. 50	PUSH EAX	
00401225	. 68 9C664100	PUSH UDP_floo.0041669C	
0040122A	. E8 E1FDFFFF	CALL UDP_floo.00401010	
0040122F	. 8D8424 D00100	LEA EAX,DWORD PTR SS:[ESP+1D8]	
00401236	. 68 B0664100	PUSH UDP_floo.004166B0	
0040123B	. 50	PUSH EAX	
0040123C	. E8 C3220000	CALL UDP_floo.00403504	[Arg2 = 004166B0 ASCII "._:" Arg1 UDP_floo.00403504
00401241	. 33C9	XOR ECX,ECX	

00403511	. 50	PUSH EAX	Arg3 Arg2 Arg1 UDP_floo.004065D9
00403512	. FF75 0C	PUSH DWORD PTR SS:[EBP+C]	
00403515	. FF75 08	PUSH DWORD PTR SS:[EBP+8]	
00403518	. E8 BC300000	CALL UDP_floo.004065D9	
0040351D	. 83C4 0C	ADD ESP, 0C	

0012FB64	004064FC	UDP_floo.004064FC
0012FB68	00000000	
0012FB6C	0012FD60	Arg1 = 0012FD60 ASCII "OKAY_114.70.37.17:7777"
0012FB70	004166B0	Arg2 = 004166B0 ASCII "._:"
0012FB74	00153A04	Arg3 = 00153A04
0012FB78	0012FF78	
0012FB7C	00401241	RETURN to UDP_floo.00401241 from UDP_floo.00403504
0012FB80	0012FD60	ASCII "OKAY_114.70.37.17:7777"
0012FB84	004166B0	ASCII "._:"
0012FB88	0041669C	UDP_floo.0041669C
0012FB8C	0012FD60	ASCII "OKAY_114.70.37.17:7777"
0012FB90	00155A50	
0012FB94	00418CEC	UDP_floo.00418CEC
0012FB98	000000C4	

```

ESP 0012FB34
EBP 0012FB64
ESI 0012FD65 ASCII "114.70.37.17:7777"
EDI 0012FD60 ASCII "OKAY"
EIP 00406693 UDP_floo.00406693

C 0 ES 0023 32bit 0(FFFFFFFF)
P 1 CS 001B 32bit 0(FFFFFFFF)
A 0 SS 0023 32bit 0(FFFFFFFF)
Z 0 DS 0023 32bit 0(FFFFFFFF)
S 0 FS 003B 32bit 7FFDD000(FFF)
T 0 GS 0000 NULL

```

0012FB64	0012FB78	RETURN to UDP_floo.0040351D from UDP_floo.004065D9
0012FB68	0040351D	ASCII "OKAY"
0012FB6C	0012FD60	ASCII "._:"
0012FB70	004166B0	
0012FB74	00153A04	
0012FB78	0012FF78	
0012FB7C	00401241	RETURN to UDP_floo.00401241 from UDP_floo.00403504
0012FB80	0012FD60	ASCII "OKAY"
0012FB84	004166B0	ASCII "._:"
0012FB88	0041669C	UDP_floo.0041669C
0012FB8C	0012FD60	ASCII "OKAY"
0012FB90	00155A50	
0012FB94	00418CEC	UDP_floo.00418CEC
0012FB98	000000C4	

다음의 과정에서, 받은 데이터의 문자열에서 "_" 와 ":" 을 기준으로 아이피 주소와 포트 번호를 각각 추출해내는 것을 확인할 수 있다.

```

Registers (FPU)
EAX 0012FD65 ASCII "114.70.37.17"
ECX 661A23CD
EDX 00000007

```

아이피 주소가 정상적으로 추출되었다.

004012C0	> 6A 10	PUSH 10	ToLength = 10 (16.) pTo Flags = 0 DataSize Data = UDP_floo.004166DC Socket sendto
004012C2	. 8D4424 14	LEA EAX, DWORD PTR SS:[ESP+14]	
004012C6	. 50	PUSH EAX	
004012C7	. 6A 00	PUSH 0	
004012C9	. 57	PUSH EDI	
004012CA	. 68 DC664100	PUSH UDP_floo.004166DC	
004012CF	. FF7424 1C	PUSH DWORD PTR SS:[ESP+1C]	
004012D3	. FF15 18114100	CALL DWORD PTR DS:[<&WS2_32.#20>]	
004012D9	. 3BC7	CMP EAX, EDI	

0012FB7C	004166DC	ASCII "2018111353"
0012FB80	0000001E	

위의 과정에서 sendto() 함수가 10번 실행된다. 소켓을 통해 "2018111353"이라는 데이터가 10번 보내지는 것을 확인할 수 있다.

004012E3	. 68 E8664100	PUSH UDP_flo.004166E8	[Arg1 = 004166E8
004012E8	. E8 23FDFFFF	CALL UDP_flo.00401010	UDP_flo.00401010
004012ED	. 83C4 04	ADD ESP,4	
004012F0	. FF7424 08	PUSH DWORD PTR SS:[ESP+8]	[Socket
004012F4	. FF15 10114100	CALL DWORD PTR DS:[&WS2_32.#3>]	closesocket
004012FA	. FF15 0C114100	CALL DWORD PTR DS:[&WS2_32.#116>]	WSACleanup

이후 closesocket() 함수를 통해 소켓이 종료된다.

4. 결론

UDP_flooding.exe는 겉보기에는 정상적인 프로그램으로 보이지만 실행 시 악성코드를 실행할 수 있는 트로이목마와 관련된 악성 코드이다. 이 프로그램의 주요 악성 행위는 UDP 소켓 통신을 통한 Flooding 공격이다. 프로그램 실행 시 소켓이 생성되어 서버의 역할을 하는 114.70.37.17:10004 주소로 ""IMHACKER_2018111353" 라는 데이터가 보내진다. 이 데이터를 서버가 받으면 공격 대상지 주소를 "OKAY_114.70.37.17:7777" 라는 문자열로 알려준다. 이를 통해 공격 대상지 주소를 알게 된, 이 프로그램이 실행된 로컬 컴퓨터는 해당 공격 대상지 주소로 "2018111353"라는 데이터를 10번 보내는 Flooding 공격을 수행하고 이 악성 프로그램은 종료된다.