한국항행학회 논문지 제14권 제6호 2010년 12월

Stuxnet의 파일 은닉 기법 분석 및 무력화 방법 연구

Analysing and Neutralizing the Stuxnet's Stealthing Techniques

이경률*, 임강빈*

Kyung-Roul Lee*, Kang-Bin Yim*

요 약

본 논문은 현재 전 세계적으로 사이버전의 심각성을 자극하고 있는 악성코드인 Stuxnet에 대해 소개하고, Stuxnet이 전파되는 방법 및 감염 증상에 대해 분석하며 그 구조에 따른 치료 방안을 제안한다. Stuxnet과 같은 악성코드는 시스템 내에서 은닉되어 전파되기 때문에 이를 탐지하거나 치료하기 위해서는 이들이 사용하는 파일 은닉 기법을 분석해야 한다. 본 논문의 분석 결과, Stuxnet은 자신을 구성하는 파일들의 은닉을 위하여 유저레벨에서 라이브러리 후킹을 활용한 기법과 커널레벨에서 파일시스템 드라이버의 필터 드라이버를 활용한 기법을 사용하고 있다. 따라서 본 논문에서는 Stuxnet이 활용하는 파일 은닉 기법에 대한 분석 결과를 소개하고, 이를 무력화하기 위한 방안을 제시하며, 실험을 통해 실제 무력화가 가능함을 확인하였다.

Abstract

This paper introduces Stuxnet, a malicious ware that presently stimulates severity of the cyber warfare worldwide, analyses how it propagates and what it affects if infected and proposes a process to cure infected systems according to its organization. Malicious wares such as Stuxnet secretes themselves within the system during propagation and it is required to analyze file hiding techniques they use to detect and remove them. According to the result of the analysis in this paper, Stuxnet uses the library hooking technique and the file system filter driver technique on both user level and kernel level, respectively, to hide its files. Therefore, this paper shows the results of the Stuxnet's file hiding approach and proposes an idea for countermeasure to neutralize it. A pilot implementation of the idea afterward shows that the stealthing techniques of Stuxnet are removed by the implementation.

Key words : SCADA System, Stuxnet, Malicious Codes, Stealthing Techniques

I. 서 론

과거 악의적인 목적으로 다른 시스템에 침입하는 바이러스부터 시작하여 웜, 트로이목마, DoS, DDoS 등으로 이어지는 악성코드는 날로 진화하고 있으며, 그 전파속도 또한 빨라지고 있는 실정이다[1]. 하지만 현재 대부분의 악성코드들은 호기심이나 개인적인 목적을 위해 이용되어 왔으며 국가적 차원에서의 공격은 없었다. 그러나 최근 6월 발견된 Stuxnet은 핵심 시설의 파괴를 목표로 하고 있으며 개인이 아닌 국가적인 차원에서 제작되었다고 의심될 만큼 위협적이어서 최초의 사이버전을 야기하는 사이버미사일

* 순천향대학교 정보보호학과(Department of Information Security Engineering, Soonchunhyang University)
· 제1저자 (First Author) : 임강빈
· 투고일자 : 2010년 11월 1일
· 심사(수정)일자 : 2010년 11월 2일 (수정일자 : 2010년 12월 16일)
· 게재일자 : 2010년 12월 30일

이라고 불리기도 한다[2].

대부분의 악성코드들을 분석할 때에는 악의적인 기능, 예를 들면 키보드 데이터를 훔쳐내거나 시스템을 파괴하는 행위 등을 분석하는 것도 중요하지만 근본적으로 코드분석을 위해 악성코드 파일 자체를 확보해야 분석이 가능하기 때문에 파일 은닉 기법을 이해하는 것이 가장 중요하다. 특히 악성코드들은 피해자 컴퓨터에 감염된 후에 탐지되거나 무력화되지 않기 위해 파일 은닉 기법을 우선적으로 수행하며, 다른 시스템을 감염시킬 경우에도 자신을 은닉한 상태에서 감염시켜야 탐지 및 무력화되지 않으므로 이를 위해 다양한 접근 방법을 이용한다.

따라서 본 논문은 Stuxnet에서 사용하는 파일 은닉 기법을 분석하고 분석된 기법을 토대로 이를 무력화하기 위한 방안을 제시하였다. 또한 제시된 방안을 구현하였으며, 이를 실제 감염된 컴퓨터에서 실험함으로써 제안한 방안이 Stuxnet의 파일 은닉 기법을 무력화할 수 있음을 증명하였다.

본 논문의 구성은 다음과 같다. 제2장에서는 Stuxnet이 감염된 후 나타나는 증상과 전파 방법에 대해 조사하였으며, 제3장에서는 Stuxnet이 사용하는 파일 은닉 기법에 대해 분석하였다. 분석된 내용을 토대로 유저레벨, 커널레벨별 무력화 방안에 대해 제4장에 기술하였으며, 제5장에 결론을 도출하였다.

Ⅱ. Stuxnet 감염 증상 및 전파 방법

2-1 Stuxnet의 분포 현황

Stuxnet은 최초로 산업자동화제어시스템(SCADA System)을 타켓으로 제작된 악성코드로 폐쇄망으로 운용되는 주요 기반시설을 공격하며 벨라루스에서 2010년 6월 처음 발견되었다[3]. Stuxnet이라는 명칭을 가지게 된 이유는 바이러스 코드 안에 “Stuxnet”으로 시작하는 파일의 이름이 많아서 USB 자동실행 취약점, 네트워크 공유 취약점, LNK 파일 취약점을 이용하여 마이크로소프트사의 MS10-046(서버 서비스의 취약점으로 인한 원격 코드 실행 문제점)[4], MS10-061(윈도우 쉘 취약점으로 인한 원격코드 실행 문제점)[5], MS08-067(인쇄 스풀러 서비스의 취약점으로 인한 원격 코드 실행 문제점)[6] 등을 통해 전파한다. Stuxnet은 명령제어서버와 P2P 업데이트를 이용하는데 그 50% 이상이 이란에 분포되어 있으며 여러 나라의 주요 산업시설에 전파되고 있다. 현재 분포된 현황을 그림 1에 나타내었다[7]. 또한 감염된 컴퓨터의 파일을 읽고 쓰거나 삭제, 프로그램 생성 등의 명령을 이용할 수 있으며 데이터를 탈취한 뒤 흔적을 없애기 때문에 분석에 어려움이 있다.

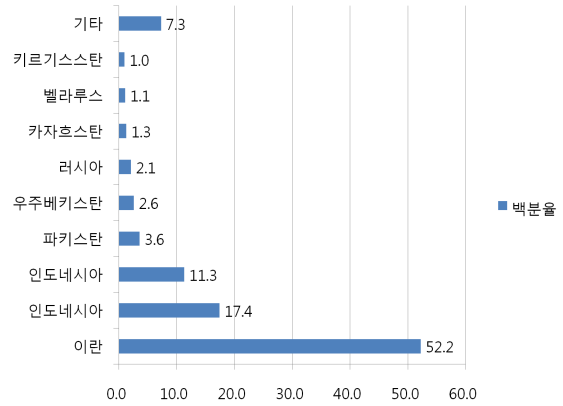

그림 1. 국가별 Stuxnet 분포 현황
Fig. 1. World wide distribution of Stuxnet.

2-2 Stuxnet의 구성 요소

Stuxnet에서 사용되는 파일은 그림 2와 같이 다수의 핵심 파일과 그 링크파일로 이루어져 있다. 파일 리스트 중 LNK 파일은 전파에 사용되는 파일이며 “~wtrxxxx.tmp” 파일은 실제 악의적인 행위를 하는 파일이다. “xxxx” 부분은 랜덤한 값으로 생성된다. LNK 파일은 바로 가기 파일이며, 링크된 디렉토리나 파일을 실행할 수 있도록 경로 정보를 저장하고 있다. 따라서 LNK 파일 내의 경로에 해당하는 부분을 Stuxnet, 즉 “~wtr4141.tmp”가 위치한 경로를 저장함으로써 LNK 파일이 존재하는 해당 디렉토리에 접근하기만 하면 자동으로 감염되는 형태를 이루고 있다. 실제 LNK 파일 내의 경로 정보는 그림 3, 그림 4, 그림 5, 그림 6과 같다.

USB 장치를 이용하여 악성코드 Stuxnet을 실행하면 그림 7과 같이 파일을 은닉한다. “~wtr4141.tmp” 파일이 로드되어 드라이버 파일을 생성하며 레지스

트리에 드라이버 정보를 등록하여 드라이버를 로드한다. 생성된 드라이버 파일은 “mrxcls.sys”와 “mrxnet.sys” 파일이며, 이 중 “mrxnet.sys” 파일이 파일 은닉 기능을 담당하지만 생성된 드라이버 파일 자체는 은닉하지 않는다.

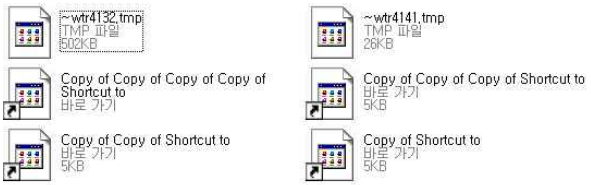

그림 2. Stuxnet 전파에 사용되는 파일 리스트
Fig. 2. File components of Stuxnet.

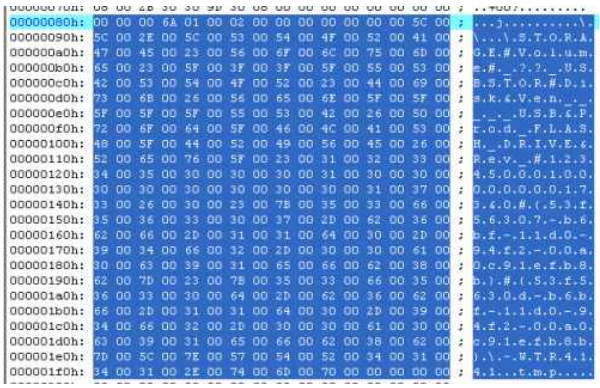

그림 3. “Copy of Shortcut to” 파일 내 경로 정보
Fig. 3. File path in “Copy of Shortcut to”.

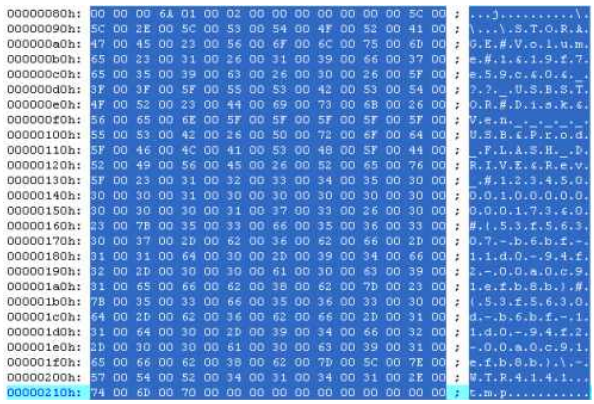

그림 4. “Copy of Copy of Shortcut to” 파일 내 경로 정보
Fig. 4. File path in "Copy of Copy of Shortcut to”.

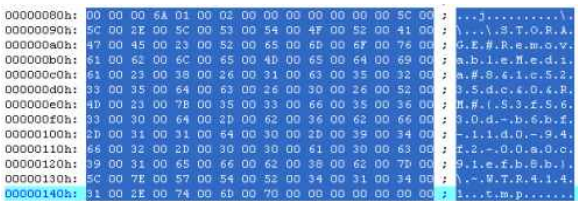

그림 5. “Copy of Copy of Copy of Shortcut to” 파일 내 경로 정보
Fig. 5. File path in “Copy of Copy of Copy of Shortcut to”.

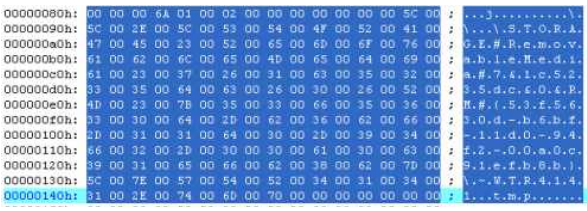

그림 6. “Copy of Copy of Copy of Copy of Shortcut to” 파일 내 경로 정보
Fig. 6. File path in “Copy of Copy of Copy of Copy of Shortcut to”.

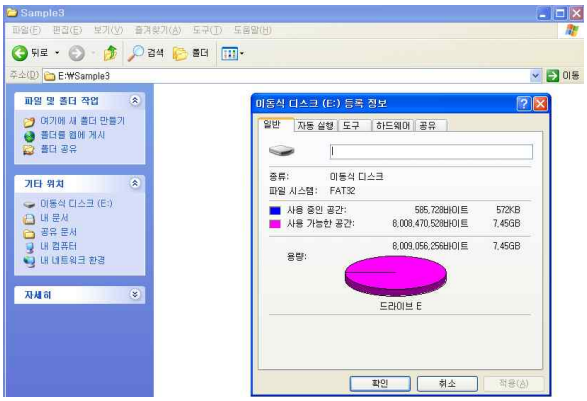

그림 7. Stuxnet 실행 후 파일 은닉 결과
Fig. 7. Hiding files after running Stuxnet.

Ⅲ. Stuxnet의 파일 은닉 기법

3-1 유저레벨에서의 파일 은닉

감염 후 악성코드 파일인 “~wtr4141.tmp” 파일을 검색하면 그림 8과 같이 검색되지 않는 것을 확인할 수 있다. 유저레벨에서는 DLL 인젝션을 통해 “kernel32.dll” 내의 FindFirstFileW, FindNextFileW, NtQueryDirectoryFile 함수를 IAT 후킹을 통해 은닉하

고 있으며 이를 그림 9에 나타내었다.

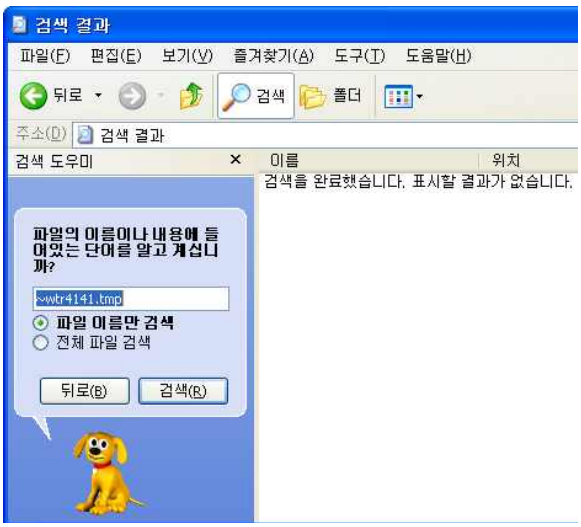

그림 8. 무력화 전 Stuxnet 파일 검색 결과
Fig. 8. File search result before neutralizing Stuxnet.

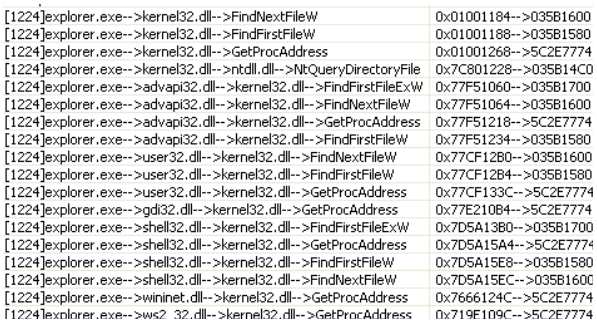

그림 9. Stuxnet에 의해 후킹된 함수 리스트
Fig. 9. List of functions hooked by Stuxnet.

3-2 커널레벨에서의 파일 은닉

파일 은닉 기능을 담당하는 “mrxnet.sys” 파일이 로드되면 디바이스 스택에 추가적인 디바이스들이 어태치된 것을 확인할 수 있다. 특히 윈도우즈 파일 시스템 드라이버인 “Ntfs” 디바이스를 확인한 결과 “\Driver\MRxNet”이라는 드라이버명을 가진 디바이스가 하위 계층에 어태치된 것을 확인할 수 있으며 이를 그림 10에 나타내었다. 실제 “mrxnet.sys” 드라이버를 디버거로 확인하면 IoAttachDeviceToDeviceStack 함수가 import된 것을 알 수 있으며 이를 그림 11에 나타내었다. IoAttachDeviceToDeviceStack 함수를 이용하여 파일 은닉을 위해 삽입된 필터를 위한 디바이스는 총 14개이며 이에 대한 리스트는 그림 12와 같다.

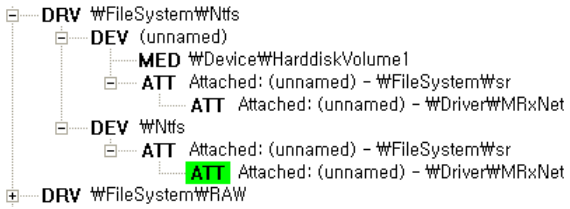

그림 10. NTFS 드라이버 스택
Fig. 10. NTFS driver stack.

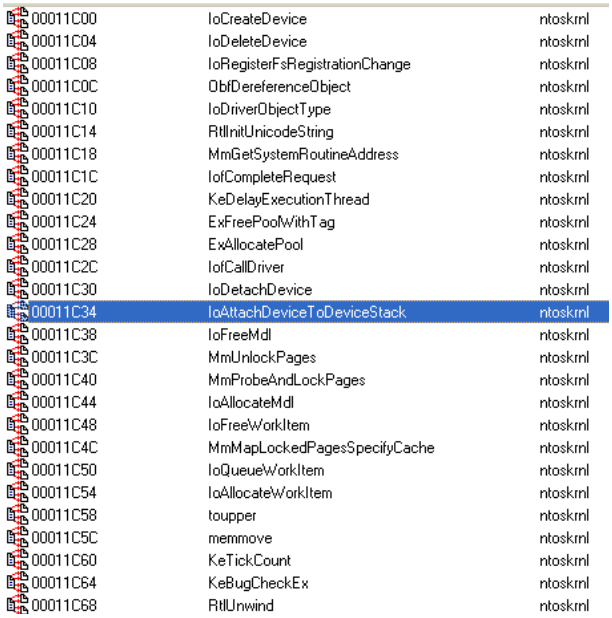

그림 11. “mrxnet.sys” 드라이버 내 임포트된 어태치 함수
Fig. 11. Imported function to attach the“mrxnet.sys” driver.

그림 12. 삽입된 디바이스 리스트
Fig. 12. List of Inserted devices.

Ⅳ. 파일 은닉 무력화 방안

4-1 유저레벨에서의 무력화 방안

IAT 후킹은 그림 13과 같은 방법에 의해 은닉을 위한 함수가 호출된다. IAT 는 프로세스가 로드될 때 메모리 공간에 저장되기 때문에 후킹이 탐지된 함수의 주소를 후킹 함수 주소가 아닌 실제 함수 주소로 변경함으로써 무력화할 수 있다. Victim 프로세스에서 Function 1을 호출하면 IAT 내의 Function 1 주소를 참조하여 실제 Function 1을 호출하는데 후킹되었다면 IAT 내의 Function 1의 주소가 후킹 함수 주소로 변경되어 있으므로, 악의적인 기능을 수행할 수 있다. 따라서 이를 무력화하기 위해서는 Function 1의 후킹을 탐지한 후 IAT 내의 Function 1의 주소를 실제 Function 1의 주소로 대체함으로써 무력화가 가능하다. 상기의 함수를 무력화한 결과 은닉된 파일이 검색됨을 확인할 수 있으며, 이를 그림 14에 나타내었다.

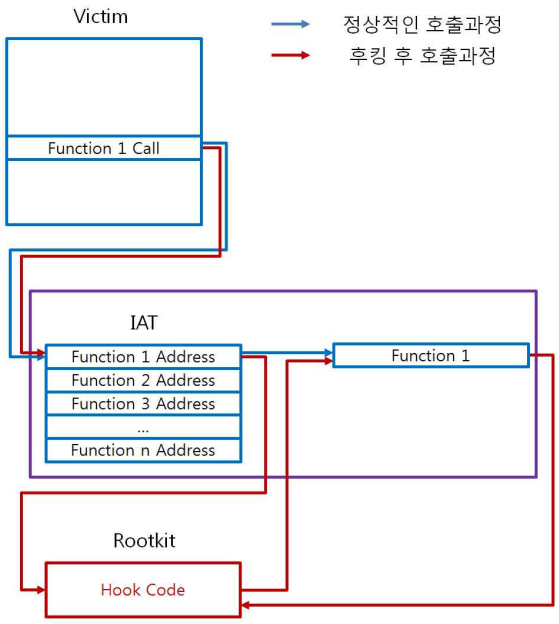

그림 13. IAT 후킹 시 함수 호출 과정
Fig. 13. Flow of function calls after IAT hook.

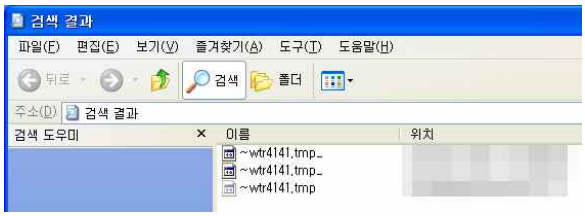

그림 14. IAT 후킹 무력화 후 은닉된 파일 탐지 결과
Fig. 14. Hidden files shown after neutralizing the IAT hook.

이 결과는 파일 필터 드라이버에 의해 은닉되지 않은 시점에서 분석한 결과이기 때문에 탐지가 가능하지만 드라이버가 로드된 후에는 IAT 후킹 무력화를 하여도 파일 탐지가 되지 않는다. 따라서 추가적으로 파일 필터 드라이버에 대한 분석이 필요하다.

4-2 커널레벨에서의 무력화 방안

필터 드라이버를 이용한 파일 은닉 기법은 파일 은닉을 위한 디바이스 하위에 위치하여 최초 IRP가 생성되어 상위 계층으로 전달되는 과정에서 CompleteRoutine을 등록하면 IRP가 상위계층에서 처리된 후 필터 드라이버 계층에 도착하여 등록된 CompleteRoutine가 호출되므로 호출된 CompleteRoutine 함수 내에서 수신된 IRP 내의 결과를 조작함으로써 파일 은닉이 가능하다. 또한 IoCompleteRequest 함수를 호출하여 요청이 정상적으로 완료되었다는 것을 리턴함으로써 파일 은닉이 가능하다. 따라서 이를 무력화하기 위해서는 어태치된 필터 디바이스를 모두 디태치함으로써 CompleteRoutine을 호출하지 못하게 하여 파일 필터 기능이 수행되지 않도록 할 수 있다. 이를 실제로 구현한 드라이버 소스 코드의 일부를 그림 15에 나타내었다.

윈도우 운영체제는 NTFS 파일 시스템을 사용하므로 “\\NTFS” 명을 가진 파일 시스템 디바이스 스택의 최하위 디바이스 오브젝트를 구한다. 구해진 디바이스 오브젝트는 Stuxnet에 의해 삽입된 필터 드라이버이지만 전체 삽입된 디바이스 오브젝트 리스트 중 처음이 아니므로 디바이스 리스트의 첫 오브젝트를 추출하기 위해 드라이버 오브젝트의 디바이스 오브젝트를 구하여 저장한 후, 각 디바이스 오브젝트가 어태치된 상위 디바이스 오브젝트를 구하여 디태치한다.

```
RtlInitUnicodeString ( &NTFSName,  L"\\NTFS");


status = IoGetDeviceObjectPointer ( &NTFSName, FILE_ALL_ACCESS, &pFileObject, &NTFSObject );
if ( !NT_SUCCESS ( status ) )
{
    DbgPrint  ( "IoGetDeviceObjectPointer () Fail!!!" ) ;
    return;
}

TopObject = NTFSObject->DriverObject->DeviceObject;

NextObject = TopObject;

for(index = 0; index < 13; index++)
{
  DeviceExtension = NextObject->DeviceObjectExtension;
  AttachedObject = DeviceExtension->AttachedTo;
  if(AttachedObject != NULL)
   IoDetachDevice(AttachedObject);
  NextObject = NextObject->NextDevice;
  if(NextObject == NULL)
   break;

}
```

그림 15. Stuxnet 파일 필터 드라이버 무력화를 위한 코드
Fig. 15. Code for neutralizing the filter drivers of Stuxnet.

다음 디바이스 오브젝트는 NextDevice 필드에 저장되어 있으므로 순차적으로 첫 디바이스 오브젝트부터 마지막 디바이스 오브젝트까지 디태치를 수행함으로써 무력화할 수 있으며 이를 수행한 결과 디바이스 스택에서 Stuxnet의 디바이스가 삭제된 것을 확인할 수 있다. 무력화 후 악성코드 실행파일인 "~wtr4141.tmp" 파일을 검색하면 숨겨진 파일로 탐지됨을 확인할 수 있다. 탐지 결과를 그림 17, 그림 18에 나타내었다.

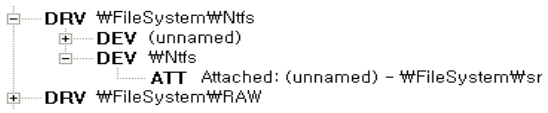

그림 16. 필터 드라이버 무력화 후 Stuxnet 디바이스가 삭제된 디바이스 스택
Fig. 16. Device stack after neutralizing filter drivers.

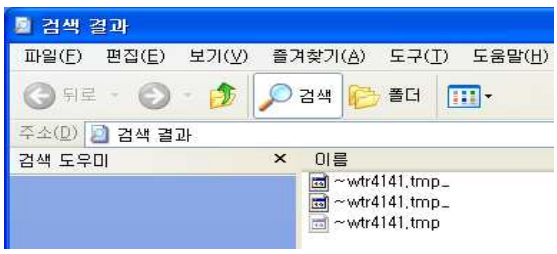

그림 17. 필터 드라이버 무력화 후 탐지된 Stuxnet 파일
Fig. 17. Files shown after neutralizing filter drivers.

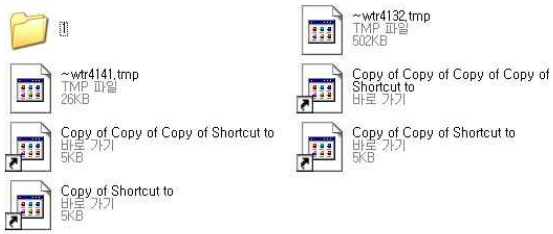

그림 18. 필터 드라이버 무력화 후 USB 내 저장된 파일 리스트
Fig. 18. List of files in USB memory after neutralizing.

Ⅴ. 결 론

본 논문은 최초의 사이버전을 야기하는 사이버 미사일로 불리우며 최근 사회적 문제를 일으키고 있는 악성코드인 Stuxnet의 감염 후 증상과 전파 방법에 대해 조사하였고, Stuxnet이 파일 은닉을 위해 사용하는 유저레벨 및 커널레벨에서의 기법을 분석하였다. 분석된 내용을 토대로 파일은닉을 무력화하기 위한 방안을 제시하였으며, 무력화를 위한 실제 코드를 구현하여 이를 검증하였다.

논문의 결과는 향후 악성코드의 파일 은닉을 무력화하기 위한 효과적인 방안으로 활용될 수 있다. 다만 최근의 악성코드 변화양상을 볼 때 향후 이러한 악성코드의 변종이 다수 출현할 것으로 보인다. 악성

코드의 행위가 심각해질수록 이들은 자신을 은닉하기 위하여 보다 고도화된 기법을 활용할 것이 자명하다. 따라서 보다 빠른 분석을 위해 이들이 사용하는 은닉 기법을 효과적으로 무력화하기 위해서는 악성코드의 은닉기술 분석을 위한 정형화된 프로세스의 개발이 요구된다.

참 고 문 헌

[1] 안철수연구소, “ASEC 리포트”, ASEC, 2010년 9월

[2] 아이뉴스24, “최악 악성코드 스턱스넷, 그 정체는?”, 2010년 10월

[3] 연합뉴스, “행안부, 스턱스넷 감염차단 긴급조치”, 2010년 10월

[4] Microsoft, “http://support.microsoft.com/kb/2286198/ko”, 2010년 8월

[5] Microsoft, “http://support.microsoft.com/kb/2347290/ko”, 2010년 10월

[6] Microsoft, “http://support.microsoft.com/kb/958644/ko”, 2009년 1월

[7] Aleksandr Matrosov, Eugene Rodionov, David Harley, Juraj Malcho, “Stuxnet Under the Microscope Revision 1.1”

이 경 률 (李庚栗)

2008년 8월 : 순천향대학교 정보보호학과 (공학사)

2010년 8월 : 순천향대학교 정보보호학과 (공학석사)

2010년 9월~현재 : 순천향대학교 정보보호학과 박사과정

관심분야 : vulnerability analysis, obfuscation, system security, insider threats

임 강 빈 (任綱彬)

1992년 2월 : 아주대학교 전자공학과 (공학사)

1994년 2월 : 아주대학교 전자공학과 (공학석사)

2001년 2월 : 아주대학교 전자공학과 (공학박사)

1999년 3월~2000년 2월 : (미)아리조나주립대학교 연구원

2003년 3월~현재 : 순천향대학교 정보보호학과 교수

2005년 3월~현재 : 한국정보보호학회 이사

2009년 3월~현재 : 한국인터넷정보학회 이사

관심분야 : insider threats, security assurance, vulnerability analysis, malware analysis, code obfuscation