Journal of the Korea Convergence Society
Vol. 8. No. 5, pp. 37-43, 2017 https://doi.org/10.15207/JKCS.2017.8.5.037

Registry 분석을 통한 악성코드 감염여부 탐지 방법 연구

홍성혁
백석대학교 정보통신학부

Research on Registry Analysis based Malware Detection Method

Sunghyuck Hong
Div. of Information Communication, Baekseok University

요 약 윈도우 운영체제(Operating System)에서 OS와 어플리케이션 프로그램 운영에 필요한 정보를 저장하기 위해 개발된 계층형 DB인 registry는 부팅에서 사용자 로그인, 응용 서비스 실행, 어플리케이션 프로그램 실행, 사용자 행위 등 모든 활동에 관여하기 때문에, registry를 분석을 통한 디지털증거획득이 많이 사용되고 있다. 최근 사용자가 인식하지 못하는 방법으로 악성코드가 시스템에 침투하여 귀중한 기술정보를 유출하거나 도용하여 금전적 피해가 많이 발생하고 있다. 따라서 본 연구에서는 고가의 디지털포렌식 프로그램 사용 없이 세어웨어 어플리케이션을 이용하여 악성코드를 탐지하는 방법을 제시하여 해킹의 피해를 분석하고 동일한 피해를 예방하기 위해 본 연구를 진행하였으며, 악성코드를 탐지하고 분석하기 위해 고가의 상용프로그램을 사용하지 않고도 정확히 분석할 수 있기 때문에 학문적 기여도는 클 것으로 기대한다.

• 주제어 : 레지스트리, 악성코드, 디지털포렌식, 멜웨어분석, 악성코드 감지

Abstract A registry is a hierarchy database which is designed to store information necessary for operating system and application programs in Windows operating system, and it is involved in all activities such as booting, logging, service execution, application execution, and user behavior. Digital forensic is widely used. In recent years, malicious codes have penetrated into systems in a way that is not recognized by the user, and valuable information is leaked or stolen, causing financial damages. Therefore, this study proposes a method to detect malicious code by using a shareware application without using expensive digital forensic program, so as to analysis hacking methods and prevent hacking damage in advance.

• Key Words : Registry, malware, digital forensics, malware analysis, malware detection

1. Registry 정의

Registry는 각종 응용프로그램과 시스템 작동에 관한 모든 등록 정보를 저장하여 사용하는 데이터베이스이다. 부팅부터 윈도우 계정으로 로그인, 서비스 실행등 대부분의 시스템 사용자 행위에 관한 기록을 하며, 디지털 증거 분석 관점에서 매우 유용하게 쓰는 데이터베이스이며, 윈도우 시스템의 휴지통, 패치, 이벤트 로그 등의 각종 설정에 관한 정보도 저장된다. 따라서 registry 정보를 분석하게 되면 시스템의 모든 활동들을 조사할 수 있고, 시스템 활동의 시간별 활동을 분석하거나 사용자 행동을 분석하여 디지털 범죄수사에 도움이 되며, 각종 보안시스템 및 게임, 사물인터넷 등 모든 분야에서 적용이 가능하다. [1,2,3,4,5,6]. 본 연구에서는 악성코드가 감염이 되었

을 때 윈도우 registry의 변화를 분석하여 악성코드가 활동을 하지 않더라도 감염 여부를 판단하여 악의적인 코드의 활동을 차단하여 해킹의 피해를 분석하고 동일한 피해를 예방하기 위해 본 연구를 진행하였다.

2. 분석 방법

윈도우 7이후 운영체제의 registry를 분석 도구를 이용하여 모든 registry 데이터를 분석하고, 분석 도구로는 RegShot, 파이널포렌식, RegRipper [7,8]로 이용하여 비교 검증하였다.

2.1 대상 컴퓨터

<Table 1> Test beds

Model	Samsung	TG	General PC
Number	1	3	2
Year	2010.12.	2007.05.	2012.8.
CPU	intel® Core 2 CPU Q8400 @ 2.66GHz	intel® Core 2 CPU 6320 @ 1.86GHz	intel® Core i7 CPU 3770 @ 3.4GHz
RAM	SAMSUNG DDR3 2GB * 1 SAMSUNG DDR3 1GB * 1	hynix DDR2 2GB * 2	SAMSUNG DDR3 4GB * 4
HDD/SSD	SAMSUNG SSD 830 Series 64GB	SAMSUNG SSD 830 Series 64GB	OCZ-AGILITY 3 128G SSD

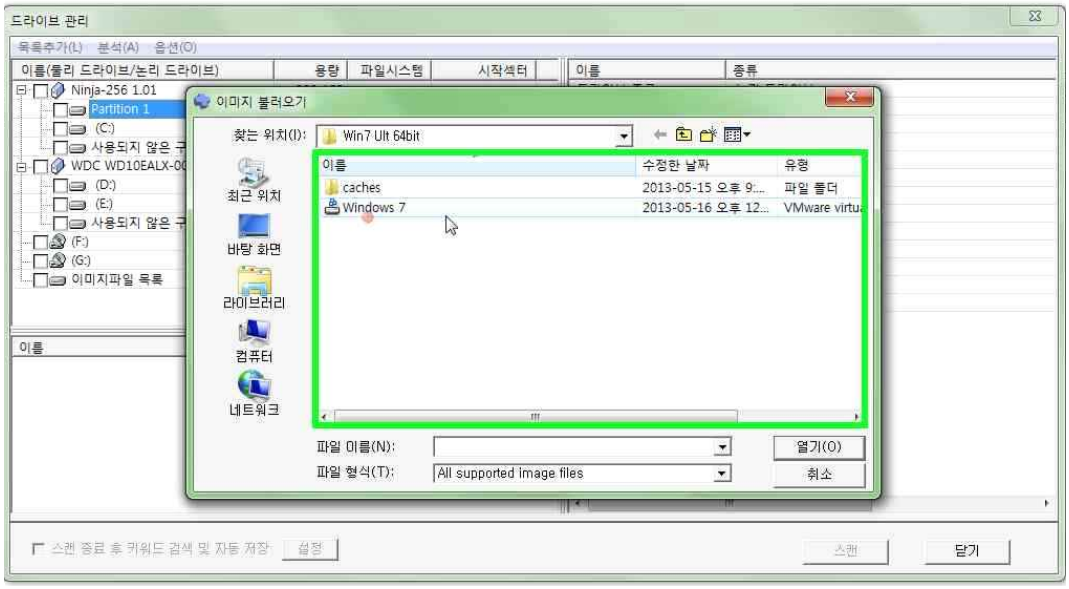
2.2 대상 운영체제

<Table 1>은 실험을 실시한 컴퓨터 스팩이며, 가장 보편성인 버전을 선택 하여 실험을 진행하였으며, 일반적으로 사용하고 있는 윈도우 7, 32bit 운영체제에서 registry를 비교 분석하였다.

3. 분석도구

분석 도구로는 RegShot, 파이널 포렌식스 3.1 및 X-way forensics 16.6, RegRipper등이 있으며, 본 연구에서는 shareware인 RegShot을 사용하였으며, 분석도구에 의한 오류를 최소화 하였고, 윈도우 설치 후 증거 이미지 파일을 분석하기 위해 WMware Player를 이용한 [Fig 1]와 같이 가상화 환경 구축을 통해 실험대상 운영체제를 가상 디스크 파일로 만들어 분석하였다 [9,10].

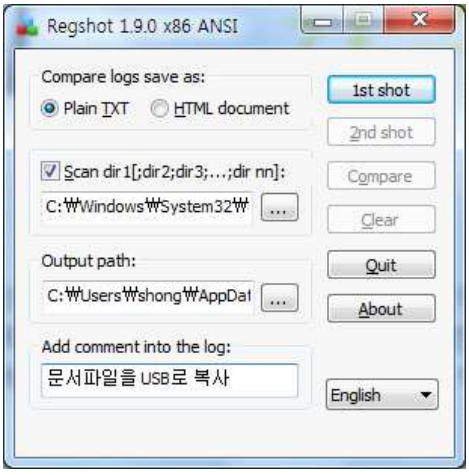
윈도우 내에서 발생 가능한 모든 이벤트를 수행 전 RegShot을 통해서 분석한 registry 값과 이벤트 후 RegShot을 통해서 분석한 registry 값을 비교하여 이벤트에 관여한 registry 값을 기록하는 방법으로 진행 하였다 [11]. [Fig 2] 참조. 아래 그림과 같이 윈도우 이벤트 전에 Regshot을 통해 초기 상태를 기록하고 [Fig 2], 이벤트 발생 후 Regshot을 통해 변화된 상태를 기록한 후 두 기록을 비교하여 변화된 registry 값을 찾아 기록하였

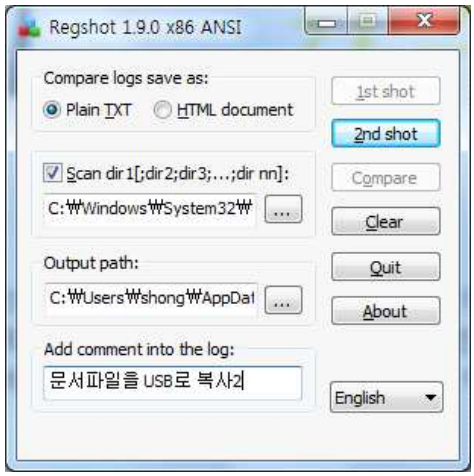


다 [Fig 3]. 아래 그림 분석을 토대로 이벤트 별 registry 값의 변화를 기록하고, 각 이벤트별 루트 키와 엔트리 이름, 데이터 종류, 값들을 정리하고, 교체 점검을 위해서 다른 포렌식 도구를 이용하여 변동된 값들을 교체 검증하였다. 변동된 값들이 방대하여 루트 키별로 중요한 데이터 값 변화를 기록 하였다 [12-15]. [Fig 4]는 레지스트의 변화된 것을 비교하는 것이며, [Fig 5]는 이밴트 전후의 결과 값을 비교하는 것이다.

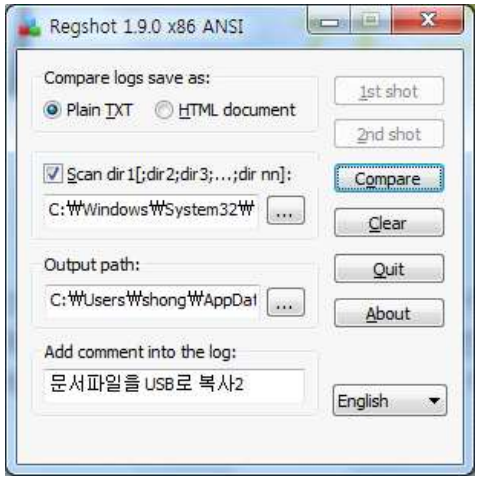
각각의 분석과정에서는 수기는 물론, 필요시 사진촬영, 화면 캡처 등을 기본으로 하였고, 이들 자료를 기반으로 실험일지를 작성하여 다양한 경우에서 나온 방대한 자료들을 처리하는 와중에 혼란이 발생되지 않도록 하였다. 실험일지에는 일련번호, 실험제목, 참여연구원, 실험일시, 실험환경(컴퓨터기종, 운영체제, 외부장치 모델넘버 및 시리얼 번호 등), 실험방법, 실험결과, 특징, 비고 등을 기록하여 향후 정확성에 논란이 있는 경우에 근거자료로 사용될 수 있도록 하였다.

윈도우 상에서 발생한 모든 이벤트에 해당하는 registry 값의 변화를 기록하였고, 또한 모든 registry 키의 마지막 저장 시간을 기록하여 시간 정보도 함께 분석하였다. 이는 파일의 마지막 수정 시각과 매우 유사한 개념으로서, 해당 registry 키가 언제 마지막으로 수정되었는지를 FILETIME 구조체로 저장한다.

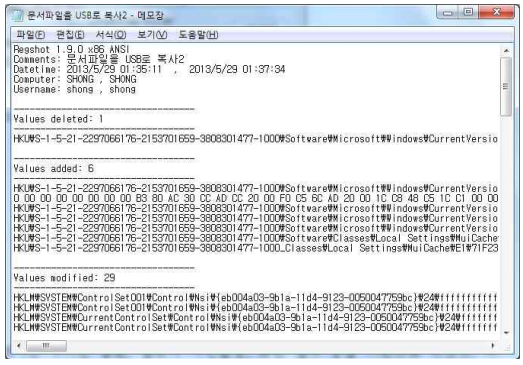




[Fig. 3] Record of After Windows Event



[Fig. 4] Compare 1st and 2nd shot



마이크로소프트 Knowledge Base에 따르면, FILETIME 구조체는 1601-01-01를 기점으로 100 nano 단위로 시각을 표시한다. 마지막 저장 시간은 registry 키가 수정 또는 생성, 접근, 삭제될 경우에 갱신된다. 이 때 마지막 저장 시간은 아쉽게도, registry 값(Value)이 아닌 registry 키(Key)에 대한 시각 정보만 얻을 수 있다. 마지막 저장 시간분석 툴로는, Windows Forensics and Incident Recovery 의 저자 Harlan Carvey가 언급한 바 있는 Keytime.exe (http://www.windowsir.com/tools.html)가 사용되고 있다. registry 키의 마지막 저장 시간은, 포렌식 전문가에게 어떤 이벤트 발생의 대략적인 날짜와 시각 정보를 알려준다. 그러나 키의 last time을 알더라도, 실제로 어떤 registry 값이 변경되었는지를 아는 것은 어려운 일기 때문에, 본 연구에서는 registry를 훌륭한 로그로 활용하기 위해서는 registry 키의 마지막 저장 시간과 함께 해당 파일시스템에서 발견한 MAC(modified, accessed, or created) time 등 다른 정보를 함께 조합하여 분석을 하였다 [15].

4. 악성코드 분석 준비

<Table 2>는 분석 도구명의 기능을 설명한 것으로 샘플 악성코드(키로거_Backdoor)를 통해 악성코드 분석을 간접 경험을 할 수 있다.

<Table 2> Function of Analysis Tools

Tool Name	Function
winalysis v3.0	Check system updates
strings	Check internal characters in executable files
BinText v3.00	same as strings
PEid v0.95	Packing, Compile information
IceSword v1.20en	Explorer Rootkit
Filemon v7.03	Check from process and file usage
PEview v0.9	Check Windows file configuration
Regmon v7.03	Check from process to registry contents
Stud_PE v2.4	Check total file information

4.1 샘플 악성코드

<Table 3>은 현재 테스트 중인 샘플 악성코드로써 제조사별로 다양하게 실험하였고, 결과 값은 AhnLab에서 제조한 in-Trojan/Prorat.349228.O을 중심으로 기록하였다.

<Table 3> Sample Malicious Codes' names

Manufacturer	Name of Malware
AhnLab-V3	- Win-Trojan/Prorat.349228.O
Avast	- Win32:Prorat-FE
Kaspersky	- Backdoor.Win32.Prorat.dz
McAfee	- BackDoor-AVW
nProtect	- Backdoor/W32.Prorat.349228.W
ViRobot	- Backdoor.Win32.Prorat.349228.G

4.2 분석 환경

실제 샘플 악성코드 분석을 위해 설정한 분석 환경이다. 윈도우를 설치 후 스냅샷을 통하여 기본 상태를 기록하고 새로운 이밴드 후 분석이 끝난 이후에는 초기 모드로 되돌리기 위해 스냅샷(snapshot) 기능을 이용하였으며, 악성코드를 분석하는 다양한 툴들의 특징은 아래와 같다.

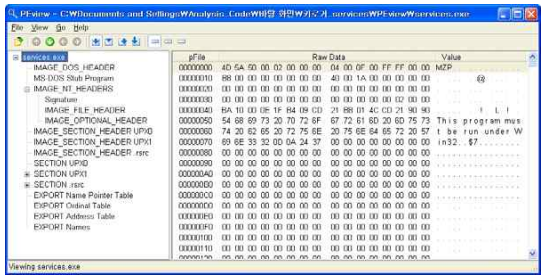
4.3 악성코드 분석 툴 소개

4.3.1 샘플 악성코드 바이너리 분석

악성코드 binary상의 ASCII code, PE구조 확인, Import & Export, 패킹 등과 같이 악성코드를 실행하지 않고 확인 할 수 있는 내용들을 분석한다.

1) PEview v0.9

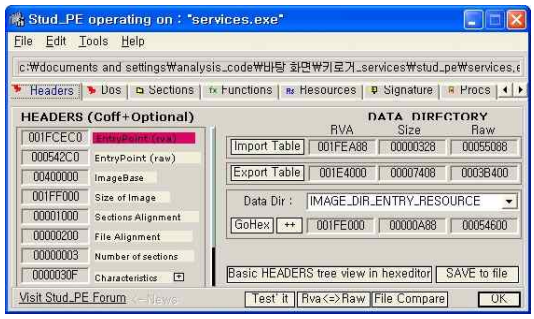
[Fig 5]는 샘플 악성코드의 binary 전체 분석이 가능하며, PE구조 형태로 분류하여 핵사코드와 아스키코드 값으로 분석할 수 있는 장점이 있다.



[Fig. 5] Screen Capture of PEview v0.9

2) Stud_PE v2.4

[Fig 6]는 샘플 악성코드의 바이너리 값을 사용자가 보기 쉽도록 정렬되어 있으며, Import, Export 함수 목록을 상세히 정리 되어 있기 때문에 사용에 편리한 장점이 있다.



[Fig. 6] Screen Capture of Stud_PE v2.4

4.3.2 샘플 악성코드 실행 분석

악성코드를 실행한 후 실행 전과 후를 비교 분석하여 어떤 레지스트리 변화가 일어 났는지 실시간으로 확인한다. [Fig 7]과 같이 스냅샷으로 서로 비교한 결과이다.

4. 결론

registry 분석 도구를 이용하여 샘플 악성코드를 가상 환경에서 실행 시키고, 실행 전과 실행 후의 레지스트 값을 분석한 결과 <Table 4>과 같은 결과가 나왔다. 분석하는 방법은 [Fig 7] 분석 도구에서 Filters 버튼을 클릭하면 Files, Groups, Registry, Services를 설정하여 이후 변경되는 정보를 설정하여 결과 값을 얻었다.

(디폴트로 C:, C:₩₩WINDOWS₩₩SYSTEM32 로 설정되어 있으며 사용에 맞게 Include 버튼으로 추가 가능 하다.)

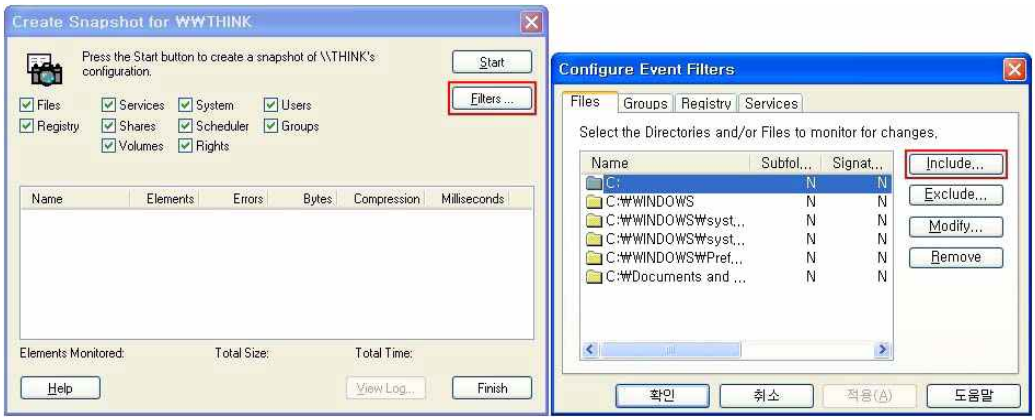
<Table 4> 통해 C:에서 WINDOWS 폴더가 변경되었음을 확인할 수 있다. 위 그림을 통해 WINDOWS 폴더 내에 services.exe 파일이 새로 생성되었고, C:₩₩WINDOWS에서 Prefetch, system, system32폴더가 변경된 것을 확인할 수 있다. 동일한 방식으로 system, system32, Prefetch 폴더 내부 파일을 체크하면 Malicious 코드가 어떤 새로운 파일들을 생성했는지 확인 할 수 있다. 파일뿐만 아니라 변경된 Registry, Groups, Rights, Scheduler, Shares, Services, System, Users, Volumes를 확인 할 수 있다. 아래는 변경, 생성된 파일 정보들이다.

(Prefetch폴더는 MS Windows에서 메모리 미리 읽기 기능을 통한 빠른 실행을 위해 존재한다. 변경 사항들의 정보로 Forensic분야에서 많이 활용되고 있다.)

<Table 4> Files and Directories After Event

Directory name	File Information
WINDOWS₩system₩	sservice.exe
WINDOWS	services.exe
WINDOWS₩Prefetch₩	NET1.EXE-029B9DB4.pf CONIME.EXE-13EEEA1A.pf NET.EXE-01A53C2F.pf CMD.EXE-087B4001.pf FSERVICE.EXE-2E4F6E14.pf SERVICES.EXE-14D2412D.pf RUNDLL32.EXE-2313F110.pf SERVICES.EXE-2B0DDD57.pf
WINDOWS₩system32₩	fservice.exe winkey.dll reginv.dll

스냅샷을 한 이후 악성코드를 실행하고, 다시 Snapshot 버튼 옆의 Test를 버튼을 클릭하면 변경된 내용을 찾아준다.



따라서, <Table 4>에서 나타난 파일들이 시스템 내에서 발견이 된다면 악성코드에 감염이 된 것이며, 악성코드의 활성여부와 상관없이 탐지를 할 수 있기 때문에 본 연구는 기여하는 바가 크며, 향후 윈도우 10에 관한 결과 분석은 추후에 진행할 예정이다.

ACKNOWLEDGMENTS

이 논문은 2017학년도 백석대학교 대학연구비에 의하여 수행된 것임

REFERENCES

[1] R. M. Saidi, S. A. Ahmad, N. M. Noor and R. Yunos, "Windows registry analysis for forensic investigation," 2013 The International Conference on Technological Advances in Electrical, Electronics and Computer Engineering (TAEECE), Konya, pp. 132-136, 2013.

[2] Han-Gi Kim, Doon-Won Kim, and Jong-Seong Kim, “A Forensic Analysis Study on One Modification of Registry Access”, Journal of the Korea Institute of Information Security and Cryptology, Vol. 26, No 5, pp. 1131-1139, 2016

[3] B. Yoo, J. Bang, K. S. Lim and S. Lee, "Analysis of Window Transactional NTFS(TxF) and Transactional Registry(TxR) in the Digital Forensic Perspective," 2nd International Conference on Computer Science and its Applications, Jeju, Korea (South), pp. 1-5, 2009.

[4] Shuhui Zhang, Lianhai Wang and Lei Zhang, "Extracting windows registry information from physical memory," 3rd International Conference on Computer Research and Development, Shanghai, pp. 85-89, 2011.

[5] Yoon-Su Jeong, Kun-Hee Han, Sang-Ho Lee, "Access Control Protocol for Privacy Guarantee of Patient in Emergency Environment", The Journal of Digital Convergence, Vol. 12, No. 07.6, pp. 279-284, 2014.

[6] Jung-Hoon Kim, Jun-Young Go, Keun-Ho Lee, "A Scheme of Social Engineering Attacks and Countermeasures Using Big Data based Conversion Voice Phishing", Korea Convergence Society, Vol. 6, No. 1, pp. 85-92, 2015.

[7] MyounJae Lee, "Prevention Method for Wireless LAN Threats and War Driving Attack", The Journal of Digital Convergence, Vol. 12, No. 10.7, pp. 501-508, 2014.

[8] WooSik Bae, "Mutual authentication and Formal Verification in M2M Environment", The Journal of Digital Convergence, Vol. 12, No. 09.5, pp. 219-224, 2014.

[9] J. Upchurch and X. Zhou, "Malware provenance: code reuse detection in malicious software at scale," 2016 11th International Conference on Malicious and Unwanted Software (MALWARE), Fajardo, PR, USA, pp. 1-9, 2016.

[10]B. Dixon, "Investigating clustering algorithm DBSCAN to self select locations for power based malicious code detection on smartphones," 2017 Third International Conference on Mobile and Secure Services (MobiSecServ), Miami Beach, FL, USA, pp. 1-7, 2017.

[11] Cheol-Joo Chae, Han-Jin Cho, “Smart Fusion Agriculture based on Internet of Thing”, Journal of the Korea Convergence Society, Vol. 7. No. 6, pp. 49-54, 2016.

[12] A. Mohsin, S. Asghar and T. Naeem, "Intelligent security cycle: A rule based run time malicious code detection technique for SOAP messages," 2016 19th International Multi-Topic Conference (INMIC), Islamabad, pp. 1-10, 2016.

[13] L. Zhiyong, S. Sai, H. Chengdong and W. Xueyu, "Detection Method of WEB Malicious Code Based on Link Analysis," 2016 International Symposium on Computer, Consumer and Control (IS3C), Xi'an, pp. 511-514, 2016.

[14] J. Lee, H. Kim, H. Yoon and K. Lee, "How to detect unknown malicious code efficiently?," 2015 International Conference on Computing, Communication

and Security (ICCCS), Pointe aux Piments, Mauritius, pp. 1-5, 2015.

[15]Byung-chul Kim, “A Internet of Things(IoT) based exploration robot design for remote control and monitoring”, Journal of digital Convergence, Vol. 13, No. 1, pp. 185-190, 2015.

저자소개

홍 성 혁 (Sunghyuck Hong) [정회원]

- 1995년 2월 : 명지대학교 컴퓨터공학과 (공학사)
- 2007년 8월 : Texas Tech University, Computer Science (공학박사)
- 2007년 9월 ~ 2012년 2월 : Texas Tech University, Office of International Affairs, Senior Programmer
- 2012년 3월 ~ 현재 : 백석대학교 정보통신학부 부교수

<관심분야> : Network Security, Hacking, Secure Sensor Networks