메모리 트랩기법을 활용한 컨테이너 탈출 탐지 프레임워크

Container Escape Detection Framework using a Memory Trap Technique

최상훈, 전우진, 김성진*, 김형천*, 박기웅†
Sang-Hoon Choi, Woo-Jin Jeon, Sungjin Kim, Hyuncheon Kim, and Ki-Woong Park

(05006) 서울특별시 광진구 능동로 209 세종대학교 정보보호학과 시스템보안 연구실
(05006) 대전시 유성구 화암동 국가보안기술연구소*
(34186) 서울특별시 광진구 능동로 209 세종대학교 정보보호학과†
csh0052@gmail.com, woojinjeon929@gmail.com, woongbak@sejong.ac.kr†

요 약

최근 클라우드 플랫폼을 효율적으로 사용하기 위한 컨테이너 기술들이 주목을 받고 있다. 컨테이너 가상화 기술은 기존 하이퍼바이저와 비교하였을 때 이식성이 뛰어나고 집적도가 높다는 장점을 갖고 있다. 하지만 컨테이너의 가상화 기술은 하드웨어 자원을 가상화하는 시스템레벨 가상화 기술 대신, 하나의 커널을 공유하여 복수개의 인스턴스를 구동하는 운영체제 레벨 가상화 기술을 사용하기 때문에 인스턴스 간 공유 자원의 요소가 많아져 취약성 또한 증가하는 보안 문제를 가지고 있다. 이러한 컨테이너의 특징으로 인해 공격자는 운영체제의 취약점을 이용하여, 컨테이너 내부에서 호스트 운영체제의 루트 권한을 획득할 수 있게 된다. 본 논문에서는 컨테이너가 사용하는 메모리의 특정영역에 대한 변화유무를 감지하고, 감지 시에는 해당 컨테이너의 동작을 중지시키는 메모리 트랩 기법을 활용하여 컨테이너 내부에서 시작되는 호스트 운영체제에 대한 루트 권한 탈취 공격을 효율적으로 탐지 및 대응하기 위한 프레임워크를 제안한다.

1. 서론

최근 클라우드 플랫폼[1, 2]을 더 효율적으로 사용하기 위해 컨테이너 기반의 가상화 기술에 대한 관심이 증가하고 있다. 컨테이너 가상화 기술은 Linux 계열의 운영체제에서 제공하는 LXC(Linux Container)[3, 4]를 기반으로 운영체제 레벨의 가상화 기법을 활용하는 것이다. 컨테이너 기반의 가상화 기술은 기존 하이퍼바이저[5, 6]와 비교하여 이식성이 뛰어나고 자원 활용 효율이 높아 집적도가 높으며, 가볍다는 장점을 가지고 있어 낮은 사양의 환경에서 높은 집적도를 보인다. 최근 컨테이너 기반 클라우드 환경구축에 많이 활용되는 Docker[7] 및 Kubernetes[8]는 기존의 컨테이너 개념에 DevOps환경에 친화적인 이식성 및 복수개의 컨테이너를 관리하기 위한 오케스트레이션 기능을 확장시킨 것으로, 기존 컨테이너 기술의 장점인 낮은 오버헤드와 가상화의 장점인 독자적 서버 구성을 가능하게 한다. 이러한 기술의 확장을 통해 기존에 비해 낮은 오버헤드로 다양한 환경의 플랫폼을 구성 및 제공할 수 있게 되었다.

컨테이너 기술은 커널 영역을 공유하여 낮은 오버헤드로 독립적인 환경을 제공할 수 있는 장점이 있지만, 하나의 커널 영역을 공유함으로써 해당 커널의 취약점까지 공유하게 되어 공격자가 다른 컨테이너에 접근이 가능해져 부가적인 여러 보안 문제들이 발생한다. 이에 관한 기존의 연구들은[9, 10] 컨테이너 내부에서 수행되는 연산을 모니터링하여 보안 매커니즘을 제공한다는 장점이 있다. 하지만 이는 알려진 공격을 방지하는데 한정되고, 사전에 동일한 환경에서 분석이 되어야 한다는 한계점에 때문에 실제 컨테이너 플랫폼으로의 적용에는 어려움이 있었다.

따라서 본 논문에서는 컨테이너가 사용하는 메모리의 특정영역에 대한 변화유무를 감지하고, 감지 시에는 해당 컨테이너의 동작을 중지시키는 메모리 트랩 기법을 활용하여 컨테이너 내부에서 시작되는 호스트 운영체제에 대한 루트 권한 탈취 공격을 효율적으로 탐지 및 대응하기 위한 프레임워크를 제안한다.

본 논문의 구성은 다음과 같다. 2장에서는 컨테이너 기술의 취약점과 그에 따른 관련 연구를 서술하고, 3장에서는 컨테이너의 탈출을 효율적으로 탐지 할 수 있는 메모리 트랩을 활용한 프레임워크를 제안한다. 4장에서는 컨테이너 탈출 탐지에 관한 실험결과에 관해 설명하고, 5장에서는 결론을 기술한다.

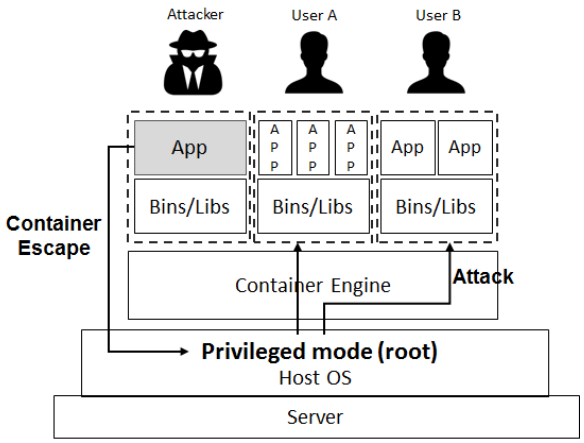
2. 컨테이너 보안 관련연구

본 장에서는 커널, 이미지와 같은 리소스의 공유로 인해 발생할 수 있는 컨테이너 가상화의 보안 문제 해결에 관한 기존 연구들에 대해 설명하고 한계점을 도출한다.

2. 1 컨테이너의 리소스 공유에 따른 보안 위협

† : 교신저자 (세종대학교 정보보호학과 교수)

컨테이너는 파일 시스템, 네트워크 및 커널과 같은 공유 리소스를 허용함으로써 가장 큰 장점인 뛰어난 이식성을 갖고 있다. 하지만 이에 따른 부가적인 보안 위험이 존재한다. 컨테이너 보안의 가장 큰 위협은 컨테이너별 커널 구성이 불가능하다는 데서 비롯된다.



(그림 1) 컨테이너 탈출을 통한 특권모드 취득

(그림 1)과 같이 컨테이너는 호스트 운영체제의 커널을 공유하여 사용한다. 컨테이너는 시스템 운영 레벨에서의 가상화이기 때문에 모든 사용자의 컨테이너는 루트 권한에서 실행된다. 따라서 컨테이너의 취약점을 통해 컨테이너의 탈출에 성공한다면 특권 모드(루트) 권한을 쉽게 획득할 수 있다. 특권 모드 권한을 획득한 공격자는 다른 사용자를 공격하거나 시스템에 장애를 일으켜 시스템 치명적인 공격을 할 수 있다.

2. 2 LXC 보안 관련 연구

컨테이너의 보안문제를 해결하기 위한 연구는 꾸준히 진행되어 왔다. 2015년 CNS[9]에서 발표된 연구에서 제시된 프레임워크는 리눅스 컨테이너 내부 환경에서 비정상적인 실행이 포착되면 이를 분석하고 보안 프로필을 생성하여 보호 메커니즘을 제공한다. 하지만 이 프레임워크는 악성 행위가 실행된 후 추적하고 프로필을 생성하여 방어하는 기법으로, 알려진 공격에 효율적이라는 제한된 환경을 기반으로 하고 있으며 악성행위 탐지를 위한 사전 테스트가 필요하다는 한계점이 있다. 같은 해에 GCUX[10]에서 발표된 연구에서는 컨테이너 환경에서 SELinux 또는 AppArmor와 같은 기존 보안 모듈을 사용하여 컨테이너 내의 필수 접근 제어 보안 강화를 제안하였다. 하지만 이러한 기술은 여러 보안 요소 중 하나일 뿐이며, 이들로 모든 보안 요건이 충족되지 않는다는 한계점을 갖는다.

이와 같이 기존 연구들은 선행적으로 악성코드가 분석이 선행 되어야 하고 컨테이너에 특화되지 않은 기존 보안 모듈을 활용한다는 점에서 한계점을 보여준다.

3. 컨테이너 메모리 트랩 프레임워크

본 장에서는 Docker 환경에서 구동되고 있는 컨테이너들의 메모리 트랩을 이용해 내부에서 발생할 수 있는 악성 행위를 효율적으로 탐지하고 차단할 수 있는 프레임워크를 제안한다.

3. 1 메모리 트랩을 위한 컨테이너 메모리 추출

LXC는 호스트에서 여러 개의 독립된 리눅스 시스템들을 구동시키기 위한 운영체제 레벨의 가상화 기법이다. LXC는 리눅스 커널의 cgroups를 활용하여 자원(CPU, Memory, Block I/O, Network)의 할당 범위를 운영 시스템 레벨에서 제한하며, 프로세스 트리, 네트워크, 사용자 ID, 마운트 된 파일 시스템 등의 운영 환경을 완전히 고립시키기 위해 namespace를 활용한다.

메모리를 통해 컨테이너 탈출을 탐지하기 위해서는 메모리 정보를 파일로 추출하는 것이 선행되어야 한다. 호스트 운영체제에서는 LXC 위에 구동되는 응용프로그램(APP)을 하이퍼바이저와 같이 하나의 프로세스로 처리한다. 이에 따라 호스트 운영체제에서 컨테이너 프로세스를 디버깅 모드로 접근하여 컨테이너의 APP 메모리 정보를 파일로 추출할 수 있다.

본 연구에서는 컨테이너가 사용하는 메모리 특정 영역을 파일로 추출하기 위해 GDB[11](GNU Debugger) 를 활용하였다. GDB란 프로세스가 실행되고 있는 동안 프로세스의 실행을 추적할 수 있도록 하여 내부 변수들의 값을 모니터링 하거나, 디버깅을 할 수 있도록 리눅스에서 제공해주는 유틸리티이다. 호스트 운영체제에 구동되고 있는 컨테이너 APP의 PID를 추출하여 GDB에 매개변수로 전달하여 모니터링 대상의 컨테이너 APP을 추적하고, 덤프를 수행할 메모리 영역을 지정하여 메모리 정보를 파일로 추출할 수 있다.

3. 2 컨테이너 탈출 취약점

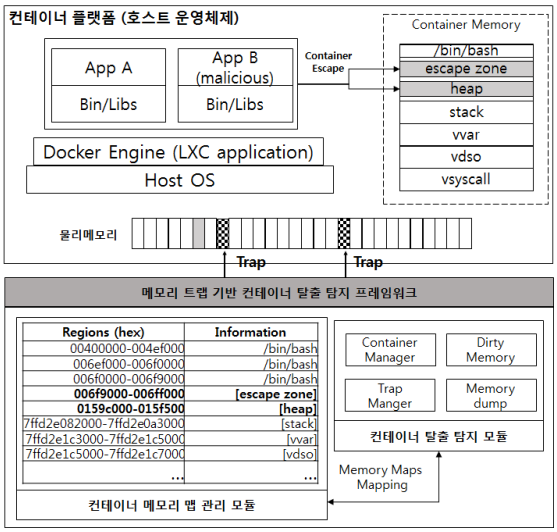
컨테이너의 APP은 루트권한으로 실행될 수 있기 때문에 컨테이너 탈출 시 호스트 운영체제의 루트권한을 획득하여 호스트 시스템을 통제 할 수 있다. 따라서 컨테이너 탈출이 가능하게 되면 호스트 운영체제에 구동중인 다른 컨테이너에 직접적인 공격이 가능하다.

우리는 메모리를 활용하여 컨테이너 탈출을 탐지 및 차단하기 위해 현재 공개되어있는 컨테이너 탈출 취약점을 분석하였다. 현재 공개된 컨테이너 탈출 취약점은 컨테이너의 성능을 극대화하기 위해 활용하는 공유된 vDSO(virtual dynamic shared object) 라이브러리로부터 발생한다. vDSO는 모든 프로세스가 공유하는 라이브러리에서 코드를 실행할 수 있기 때문에, 루트 권한으로 실행된 APP에서 호스트 운영체제로 탈출할 수 있다. 오픈소스로 공개되어있는 취약점[12]은 Dirty-Cow[13]를 활용하여, vDSO 메모리 공간 안에 있는 *clock-gettime()* 함수변조를 통해 호스트 운영체제로 탈출시킨다.

3. 3 컨테이너 탈출 탐지 프레임워크 설계

우리는 프로세스(Container APP)의 메모리 분석을 통해 컨테이너 탈출 시 변화되는 메모리 정보를 추출하여 메모리 로드 정보를 분석한 결과 두 가지 특이사항을 찾을 수 있었다.

첫 번째는 컨테이너 탈출 시 프로세스(Container APP) 힙 영역의 vDSO 데이터 변조 관련 데이터가 로드된다는 것이고, 두 번째는 컨테이너 내부에서 호스트 운영체제의 쉘을 실행하기 위한 메모리의 고정된 데이터 영역(0x006f9000 - 0x006ff000)에 쉘 관련 데이터를 인젝션 시킨다는 것이다. 즉 컨테이너 내부에서 호스트 운영체제의 특권명령 쉘 권한을 획득하기 위해서는 반드시 2개의 메모리 영역이 변조되어야한다. 우리는 이러한 분석결과를 바탕으로 트랩영역을 추출하였다.



(그림 2) 컨테이너 탈출 탐지 프레임워크 구조

메모리 트랩 기반 컨테이너 탈출 탐지 프레임워크는 (그림 2)와 같다. 위와 같이 우리가 제안한 메모리 트랩 기반의 컨테이너 탈출 탐지 프레임워크는 2개의 모듈로 구성되어있다. 컨테이너 메모리 맵 관리 모듈은 호스트 운영체제에서 구동되고 있는 컨테이너 APP에 대한 메모리 정보를 할당 정보를 분석하고, 컨테이너 탈출 시 변조되는 메모리 주소를 컨테이너 탈출 탐지 모듈에 전달한다. 컨테이너 탈출 탐지모듈에는 4가지 세부 모듈로 구성되어있다. Trap Manager 모듈은 메모리 맵 관리 모듈로부터 전달받은 컨테이너의 메모리영역으로부터 트랩이 설치되어야 할 메모리영역을 추출하며, Memory Dump 모듈은 컨테이너의 메모리 변화 정보를 감시하기 위한 메모리의 일부 영역을 파일로 추출한다. Dirty Memory 모듈은 덤프 된 메모리 정보를 분석하여 트랩 영역에 컨테이너 사용자가 접근하였는지 파악하고, Container Manager 모듈은 컨테이너 사용자가 컨테이너 탈출을 시도하였을 때 해당 컨테이너를 강제 종료시킨다.

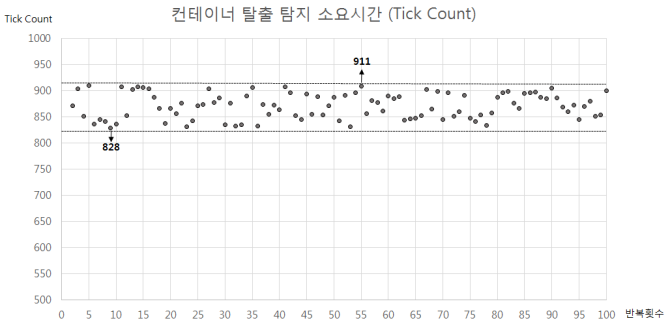
4. 컨테이너 탈출 탐지 성능 평가

본 장에서는 구현된 메모리 트랩 시스템에 대한 효용성을 검증하기 위해 트랩 모듈이 적재된 클라우드 플랫폼에서 컨테이너 탈출을 탐지하는데 소요되는 Tick Count를 측정하였다. 실험은 컨테이너 응용 도구인 Docker 1.3 환경에서 진행하였다.

4.1 실험환경

실험 환경에서는 CPU Xeon E5-2609(2.5GHz)와 32GB의 메모리, SSD 128GB의 하드웨어가 장착되어있는 서버를 사용하였다. Container APP이 구동되는 운영체제는 Ubuntu 14.04-64bit를 사용하였다. 본 실험에서는 vDSO의 취약점을 활용하여 컨테이너를 탈출하였다.

4.2 컨테이너 탈출 감지 성능평가



(그림 3) 메모리 트랩 기반 컨테이너 탈출 감지 성능평가

우리는 메모리 트랩을 통한 컨테이너 탈출 탐지의 성능을 검증하기 위해 공격시점의 Tick Count와 탐지되었을 때의 Tick Count를 비교하였다. 총 100번의 컨테이너 탈출 공격을 시도하였고 실험에 대한 결과는 (그림 3)과 같다. 컨테이너를 탈출에 대한 공격에 대해 탐지까지 최소 828 Tick이 소요되었고, 최대 911 Tick이 소요되었다. 컨테이너 탈출 공격이 시도되었을 때 평균 871 Tick 이내에 공격을 탐지하고 차단할 수 있었다. 이와 같은 실험을 통해 우리는 메모리 트랩 기반 탐지 프레임워크의 성능이 우수하다는 것을 검증하였다.

5. 결론

본 논문에서는 컨테이너 환경에서 발생할 수 있는 컨테이너 탈출 취약점을 탐지하기 위한 메모리 트랩 기법을 제안하였다. 우리가 제안한 메모리 트랩 기법은 성능의 효율성을 극대화하기 위해 특정 메모리 영역에 트랩을 만들어 컨테이너 탈출을 탐지했다. 메모리 트랩 기반 컨테이너 탐지 프레임워크를 실제 컨테이너 실행 환경에 적용하기 위해 Docker 플랫폼에 메모리 트랩 시스템을 적재하여 실험해보았다. 실험결과 공격을 탐지하는데 평균 871 Tick이 소요되었다. 이는 우리가 제안한 메모리 트랩 기반의 컨테이너 탈출 탐지 프레임워크가 효율적이라는 것을 입증하였다.

추후 연구에서는 컨테이너 기술이 클라우드 플랫폼과 함께 활성화되었을 때 발생할 수 있는 보안적 측면에서의 공유 리소스 연구를 수행할 것이며, 이러한 연구 결과를 종합하여 컨테이너 보안 강화에 관한 연구를 수행할 예정이다.

Acknowledgement

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (NRF-2017R1C1B2003957, NRF-2016R1A4A1011761)

참고문헌

[1] Mell, Peter, and Tim Grance. "The NIST definition of cloud computing." (2011).

[2] Armbrust, Michael, et al. "A view of cloud computing." Communications of the ACM 53.4 (2010): 50-58.

[3] Seo, Kyoung-Taek, et al. "Performance comparison analysis of linux container and virtual machine for building cloud." Advanced Science and Technology Letters 66.105-111 (2014): 2.

[4] Helsley, Matt. "LXC: Linux container tools." IBM devloperWorks Technical Library (2009): 11.

[5] Chisnall, David. The definitive guide to the xen hypervisor. Pearson Education, 2008.

[6] Habib, Irfan. "Virtualization with kvm." Linux Journal 2008.166 (2008): 8.

[7] Merkel, Dirk. "Docker: lightweight linux containers for consistent development and deployment." Linux Journal 2014.239 (2014): 2.

[8] kubernetes, https://kubernetes.io/

[9] Mattetti, Massimiliano, et al. "Securing the infrastructure and the workloads of linux containers." Communications and Network Security (CNS), 2015 IEEE Conference on. IEEE, 2015.

[10] Hayden, Major, and Richard Carbone. "Securing Linux Containers." GIAC (GCUX) Gold Certification, Creative Commons Attribution-ShareAlike 4.0 International License (2015).

[11] GDB, https://www.gnu.org/s/gdb

[12] Container Escape Code, https://github.com/scumjr/dirtycow-vdso

[13] Dirty Cow, https://dirtycow.ninja/

Next Generation Computing Conference 2017

한국차세대컴퓨팅학회 하계학술대회

장 소 제주 제주한라대학교

일 시 2017. 6. 16(금) 12:30 ~ 17(토) 12:00

주최·주관 사단법인 한국차세대컴퓨팅학회 Korean Institute of Next Generation Computing / 제주한라대학교 공학기술교육혁신센터

후 원 제주한라대학교

2017년도
한국차세대컴퓨팅학회
하계학술대회

장　소 : 제주 제주한라대학교

일　시 : 2017. 6. 16(금) 12:30 ~ 17(토) 12:00

주최·주관 : 한국차세대컴퓨팅학회, 제주한라대학교 공학기술교육혁신센터

후　원 : 제주한라대학교

2017년도 한국차세대컴퓨팅학회 하계학술대회

★ 대 회 장 : 백성욱 교수(세종대학교)

★ 조직위원장 : 문석환 교수(제주한라대학교)

★ 조직 위원

김덕환 교수(인하대학교)
노병희 교수(아주대학교)
신병석 교수(인하대학교)
염성관 교수(제주한라대학교)
이상웅 교수(가천대학교)
최 린 교수(고려대학교)

★ 학술위원장 : 조성제 교수(단국대학교)

★ 학술 위원

권구락 교수(조선대학교)
김봉재 교수(선문대학교)
김성백 교수(제주대학교)
김황남 교수(고려대학교)
노상욱 교수(가톨릭대학교)
민 홍 교수(호서대학교)
박기웅 교수(대전대학교)
박운상 교수(서강대학교)
석준희 교수(고려대학교)
신석주 교수(조선대학교)
이문규 교수(인하대학교)
이미영 박사(세종대학교)
이종원 교수(세종대학교)
임승호 교수(고려대학교)
정용주 교수(가천대학교)
정진만 교수(한남대학교)
허준영 교수(한성대학교)

2017년도 한국차세대컴퓨팅학회 하계학술대회 프로그램

<table>
<tr><th colspan="2">날짜 및 시간</th><th colspan="2">일정</th><th>장소</th></tr>
<tr><td>6. 15
(목)</td><td>17:30 ~ 20:30</td><td colspan="2">한국차세대컴퓨팅학회
조직위원/학술위원회의</td><td></td></tr>
<tr><td rowspan="8">6. 16
(금)</td><td>12:30 ~ 13:00</td><td colspan="2">등 록</td><td rowspan="8">제주한라대학교</td></tr>
<tr><td>13:00 ~ 14:30</td><td colspan="2">Keynote Speech
인공지능을 통한 산업혁명 - 의료의 현재 그리고 미래
주식회사 뷰노 김현준 전략이사</td></tr>
<tr><td>14:30 ~ 15:00</td><td colspan="2">개 회 식
- 이·취 임 식
- 개 회 사
- 우수논문 시상</td></tr>
<tr><td colspan="2">논문 발표</td></tr>
<tr><td>15:00 ~ 16:00</td><td>Oral Session 1</td><td>Oral Session 2</td></tr>
<tr><td>16:00 ~ 16:15</td><td colspan="2">Coffee break</td></tr>
<tr><td>16:15 ~ 17:30</td><td>Oral Session 3</td><td>Oral Session 4</td></tr>
<tr><td>6. 17
(토)</td><td>10:00 ~ 12:00</td><td colspan="2">Poster Session</td><td>제주한라대학교</td></tr>
</table>

세션별 세부 프로그램

2017. 6. 15(목)		
17:30~20:30	**한국차세대컴퓨팅학회 조직위원/학술위원 회의**	
2017. 6. 16(금)		
12:30~13:00	**등록**	
13:00~14:30	Keynote Speech **인공지능을 통한 산업혁명 – 의료의 현재 그리고 미래** **주식회사 뷰노 김현준 전략이사**	
14:30~15:00	**개회식, 이취임식, 시상식**	
논문 발표	**Oral Session 1**	**Oral Session 2**
15:00~15:15	추상적 표현을 이용한 영화 장면에서 딥특징 기반 동작 인식 / 세종대학교 / 울라 아민, 아마드 자밀, 무하마드 칸, 메흐무르 이르판, 이미영, 박준렬, 백성욱	Sparse Regularization을 적용한 Autoencoder의 성능평가 / 고려대학교 / 손혜주, 배지윤, 이민혁, 석준희
15:15~15:30	메소드 참조 빈도를 이용한 안드로이드 애플리케이션 유사도 측정 / 단국대학교, 제주대학교 / 김규식, 마수드, 김성백, 조성제	아두이노 기반 역점역 알고리즘 개발 / 금오공과대학교 / 권종훈, Angsanto Stephen Ryan, 임완수
15:30~15:45	메모리 트랩기법을 활용한 컨테이너 탈출 탐지 프레임워크 / 국가보안기술연구소, 세종대학교 / 최상훈, 전우진, 김성진, 김형천, 박기웅	Deep Convolutional Neural Networks 와 Bag-of-Words를 이용한 생물 의학 문서 분류 / 세종대학교 / 울라 누어, 양수연, 오범수, 라힘 나시르, 압둘 말릭, 최진우, 백성욱
15:45~16:00	앙상블 학습을 사용한 RF 위협체 역추정 모델의 성능평가 / 가톨릭대 / 박헌우, 김철표, 노상욱	Inter-AP Contention Detection via Buffer Queue Length / 인하대학교 / 성윤동, 노영태
16:00~16:15	Coffee break	
논문 발표	**Oral Session 3**	**Oral Session 4**
16:15~16:30	정규화된 극도의 학습 기계를 이용한 선택적 특징 기반의 치매진단 / 가천대학교 / 라마 라메쉬 쿠마, 이상웅	스마트그리드에서 소비자 사생활 보호를 위한 RCRP 라우팅 프로토콜 / 조선대학교 / 푸다사이니아모드, 쿠르니아디이르반다, 강문수, 신석주

16:30~16:45	발화행태 특징을 활용한 응급상황 신고자 연령분류 / 세종대학교 / 윤산성, 박한샘, 손귀영, 유진필, 권순일, 백성욱, 전석봉	경로 추정을 이용한 자기장 기반 실내 위치 인식 / 고려대학교 / 신재민, 최린
16:45~17:00	마이닝 top-k 불확실한 빈번한 패턴 / 세종대학교 / 레 청 투웅, 베이 보, 이미영, 박성호, 백성욱	클라우드 컴퓨팅을 활용한 위치 기반 IoT 스마트 디바이스 / 인하대학교 / 윤다빈, 김덕환
17:00~17:15	임베디드 브릿지 장치 기반의 SIP 서비스 거부 공격 방어 및 대응 시스템 구현 / 아주대학교 / 김대순, 이승운, 이철웅, 노병희	보행자 추측 항법과 저전력 블루투스 지문인식을 이용한 실내 내비게이션 시스템 / 고려대학교 / 김도윤, 구경현, 최린
17:15~17:30	랜덤 커널과 콘볼루션 기능을 이용한 이미지 검색 / 세종대학교 / 아마드 자밀, 이미영, 백성욱	익명 오러베이 네트워크 서비스 동향 / 아주대학교 / 이승준, 신승훈, 노병희

2017. 6. 17(토)

논문 발표	Poster Session 1	Poster Session 2
10:00~12:00	대용량 클라우드 스토리지 환경을 위한ROI 기반의 효율적인 영상정보 전송 기법 / 제주대학교 / 이동혁, 박남제	차세대 컴퓨팅을 위한 고속 데이터 연결용 상하 호환 접속이 가능한 능동광케이블 / 한국광기술원 / 황성환, 김계원, 박유영, 허상휴, 조경재
	병해충 검색을 위한 통합 시스템 설계 및 구현 / 세종대학교 / 이여진, 유성준, 정다운, 구영현, 박철호, 윤학림, 박종한	네트워크 시뮬레이터에서의 단일 홉 지연시간 모니터링 및 분석 시스템 / 아주대학교 / 이강혁, 이승운, 노병희
	잡음에 강건한 실시간 홍채 검출 / 숭실대학교 / 강은정, 김계영	블록체인 기술의 사물인터넷 응용서비스 적용 사례 분석 / 아주대학교 / 정은선, 강태학, 이철웅, 노병희
	웹툰의 글로벌 서비스를 위한 플랫폼 설계 및 개발 / 전자부품연구원, 세종대학교 / 문연국, 윤희화, 이동현, 황홍선, 백성욱	Rasperryi-Pi와 USRP를 사용한 집단 생태모방의 DESYNC-TDMA 구현 / 아주대학교 / 배홍열, 이철웅, 노병희
	웹툰 번역 품질 향상을 위한 크라우드 소싱 번역 시스템 개발 / 전자부품연구원, 세종대학교 / 문연국, 윤희화, 이동현, 이청호	USRP를 사용한 Congnitive Radio를 기반으로 한 사물인터넷(IoT) 서비스 플랫폼 구현 / 아주대학교 / 김성래, 이규민, 노병희
	수질 오염 측정을 위한 종합 모니터링 센서 및 수중 통신에 관한 연구 / 전자부품연구원, 세종대학교 / 문연국, 박경수, 이동현, 윤희화	MQTT 기반의 홈 IoT 서비스 제어 구조 / 아주대학교 / 박현정, 박성미, 신승훈, 노병희
	통제 구역 보안을 위한 효율적인 건물 내 방문객 관리 시스템 / 전자부품연구원, 세종대학교 / 문연국, 박경수, 이동현, 윤희화, 백성욱	해외 빅 데이터 유통 사례 분석을 통한 효율적인 빅 데이터 포털 운영에 관한 연구 / 세종대학교 / 박성호, 이미영, 최진우, 박준렬, 백성욱

	선형 회귀 분석을 이용한 Intel SGX 상의 전력 수요 예측 / 인하대학교 / 윤예진, 임종혁, 이문규	이용자 데이터 보호를 위한 클라우드 임치제도 도입방안 연구 / 한국클라우드컴퓨팅연구조합 / 채규환, 신민희, 김진택
	긴급 전화 통화에서 연령 분류를 위한 MFCC 특성에 관한 연구 / 세종대학교 / 라힘 나시르, 이미영, 권순일, 전석봉, 백성욱	수중 환경에서 무선 통신의 신뢰성 향상을 위한 적응적 베이스밴드 코딩 시스템 / 전자부품연구원, ㈜다인인더스 / 박경수, 이동현, 이덕배, 문연국
	코어 수에 따른 병렬화 수준이 Spark 성능에 미치는 영향 분석 / 단국대학교 / 김민제, 장시형, 나연묵	합성 3D 영상 기반 신체 부위 분류 / 서강대학교 / 이햇살, 박성주, 김동칠, 박운상
	네트워크 카메라 스트리밍 데이터에서의 원본 영상 추출 / 인하대학교 / 손예별, 신동진, 권희용, 이문규	경제시장 데이터 기반한 미래예측알고리즘 개발 / 호남대학교 / 백란
	CAC: 영상통신을 활용한 소통형 출입통제 시스템 / 세종대학교 / 안성규, 구예은	혼합현실에서 손동작 인식을 활용한 가상 객체의 인터렉션 체험 도구 / 가천대학교 / 신동혁, 김동언, 정용주
	화재발생시 IoT기반 접근가능형 스마트 RC카 구현 / 조선대학교 / 이지환, 구재환, 김지인, 권구락	Intel CPU의 L3캐시 효율 측정을 이용한 CPU/내장GPU간의 영상 처리 작업 분배 / 인하대학교 / 김범준, 신병석
	k-최근접 이웃 알고리즘을 이용한 근전도 기반 마우스 기능 구현 / 아주대학교 / 이주영, 최선탁, 김준호, 조위덕	스마트 비디오 리트리버(SVR): 비디오 인덱싱 및 검색을 위한 키프레임 기반 프레임워크 / 키스탄 페샤와르 이슬라미야 대학, 세종대학교 / 칸 무스타킴, 사자드 무함마드, 메흐무르 이르판, 백성욱
	PIR센서를 이용한 단일 공간 분할 영역 이동 추적 알고리즘 / 아주대학교 / 최선탁, 조위덕	