RAM Core Dump Exploitation을 방지하기 위한 RAM Encryption

신용명, 김강석, 예홍진
아주대학교 일반대학원 지식정보보안학과
e-mail:sione03@ajou.ac.kr, kangskim@ajou.ac.kr, hjyeh@ajou.ac.kr

RAM Encryption for preventing RAM Core Dump Exploitation.

Yongmyeong Shin, Kangseok Kim, Hongjin Yeh
Dept. of Knowledge Information Security, Graduate School of Ajou University

요 약

최근 많은 연구들은 RAM이 이전에 우리가 생각하는 것처럼 안전하지 않다는 것을 증명했다. RAM에는 암호화되지 않은 data가 저장되는데, RAM에 저장되어있는 데이터를 보호하기 위한 연구들이 진행되고 있다. 최근 cold boot attack이나 core dump exploitations은 과거 우리가 생각했던 것만큼 RAM이 안전하지 않다는 것을 증명하였다. 이런 유형의 공격으로부터 RAM을 보호하기 위한 방법으로 RAM의 암호화 방안을 제안한다.

1. 서론

컴퓨터를 기반으로 하는 산업의 규모가 커지고, 서비스가 증가함에 따라, 현대 시스템이 제 기능을 제대로 발휘하기 위해서 과거에 안전하다고 여겨졌던 부분을 다시 평가할 필요가 있다. 이런 부분들 중 하나가 바로 RAM에 데이터를 저장할 때 암호화하지 않고 사용하는 것에 대한 부분이다.

컴퓨터 시스템에서 데이터를 RAM에 저장하는 주된 이유는 컴퓨터의 성능을 향상시키기 위함이다. 패스워드나 인증서와 같은 데이터를 RAM에 저장함으로써, 속도가 느린 disk를 사용할 때보다 더 쉽게 데이터를 이용할 수 있다. 그러나 지금까지 RAM의 성능향상만 고려해 왔을 뿐 보안성을 고려해오지 않았다. 이제까지 hard disk나 USB와 같은 대용량 저장장치의 암호화에 대해서는 많은 연구 및 개발이 되어왔지만, RAM에 있는 data에 대한 암호화 방안에 대해서는 연구가 별로 없었다. 그러나 cold boot attack과 core dump exploitation과 같은 새로운 유형의 공격방법이 등장하면서 RAM의 취약성이 이슈화되었다.

이 논문에서 RAM의 보안성 향상을 위해 RAM의 암호화 및 RAM을 암호화함으로써 발생 할 수 있는 성능저하 문제에 대한 연구를 제안한다.

2장 관련연구에서는 cold boot attack과 core dump exploitation이 무엇이며, 어떤 방식으로 RAM의 dump image를 획득하는지 살펴본다. 3장 구현방안에서는 RAM의 암호화 및 복호화 방안에 대해 다룬다. 또한 RAM 암호화 시 고려해야 할 점들에 대해서도 살펴볼 것이다. 마지막으로 결론에서는 향후 보완되어야할 점에 대해서 다룸으로써 마무리하겠다.

2. 관련 연구

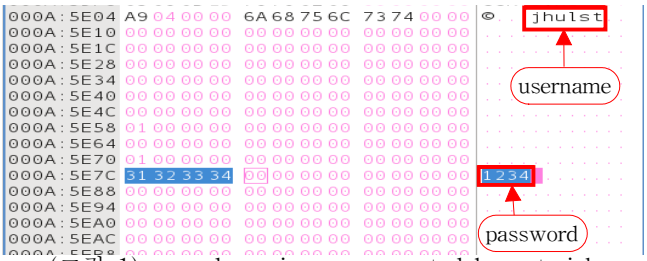
Cold boot attack과 core dump exploitation은 RAM image를 이용해서 패스워드와 같은 중요한 정보를 추출하는 공격이다.

대부분의 전문가들은 컴퓨터는 전원이 종료되면 메모리에 있는 데이터가 바로 지워진다고 생각했었다.
일반적으로 RAM은 마더보드에서 제거된 후 일반적인 온도에 노출된다면 수초기간동안 점차적으로 데이터가 소실된다. 만약 낮은 온도에서 칩을 보관한다면 데이터는 수분 또는 심지어 수 시간까지 보관될 수 있다.[1] 1978년에 있었던 실험은 RAM이 액체 질소와 함께 저온 상태로 보관된다면 한주 동안 데이터를 보관할 수 있는 것을 증명해 보였다.[2]
Attacker는 목표 컴퓨터 시스템에 cold boot attack을 수행해서 RAM image를 얻을 수 있다. 이미 말한 바와 같이 RAM에 저장되어 있는 전하는 컴퓨터 전원을 종료한 후에 즉시 사라지지 않는다. attacker는 컴퓨터의 전원을 제거한 후 일반 적인 온도에서 약 10초 이내에 USB와 같은 휴대용 저장장치 등을 이용해서 light weight operating system을 부팅시키면, RAM의 최근 이미지를 얻을 수 있다.

Core dump란 수행 중인 프로세스가 RAM에 존재할 때의 이미지이다. Linux는 프로세스가 비정상적으로 종료

될 때 core dump를 생성한다.[3, 4] Linux는 이런 기능을 디버깅 목적으로 제공한다. core dump를 조사함으로써 프로그램 개발자는 문제의 원인을 좀 더 쉽게 알아낼 수 있다. 그러나 Linux의 이런 core dump 제공기능은 본래의 취지와 달리 악용될 수 있다. 만약 attacker가 Linux 시스템에서 보안과 관련된 application을 수행하고 있는 계정정보를 획득했다면, attacker는 시스템에 침입하여, 해당 application에 kill signal과 같은 core dump를 발생시키는 시그널을 발생시켜 core dump image를 획득 할 수 있다. 하지만 이 방법은 프로세스를 중단시키기 때문에 시스템 운영자는 침입이 발생했다는 것을 알 수 있다. 프로세스 충돌을 발생시키지 않고도 core dump를 발생 시킬 수도 있는데, gcore라는 core dump utility를 사용해서 아무도 모르게 RAM image를 획득 할 수 있다.[3]

(그림 1)은 asterisk라는 open-source VoIP server프로그램을 이용해서 core dump image를 획득한 후, hex editor를 이용해서 그 내용을 검사한 것이다.[1]
Asterisk는 SIP Protocol을 지원하는데, SIP는 패스워드를 이용해서 사용자 인증을 수행한다. 인증이 완료되면, 패스워드는 RAM상에서 그 사용자이름 근처에 저장된다. 'jhulst'가 사용자 이름이고, '1234' 가 패스워드이다.



(그림 1) core dump image generated by asterisk

(그림 1)에서 보는 것과 같이, 패스워드 및 모든 중요한 데이터가 RAM에 암호화 되지 않은 채 저장되어있다. Attacker는 이런 정보를 획득함으로써, 정상적인 사용자를 사칭할 수 있다.

3. 구현 방안

이 논문은 Core Dump Exploitation을 방지하는데 목적을 두고 RAM Encryption 방안 을 연구한다. 보안성 테스트는 Core Dump Exploitation을 대상으로 실시하지만, cold boot attack역시 RAM의 image를 이용해서 정보를 추출해 내기 때문에 cold boot attack에 대한 대비책이 될 수 있다.

RAM의 암•복호화 알고리즘은 대칭키 암호화 방식을 사용하고, 이를 수행하기 위해 3개의 system call을 사용한다.

첫 번째 system call은 key를 생성하는 system call로써, key를 시스템의 시간정보를 이용해 key를 랜덤으로 생성한 후, 이 생성된 key를 프로세스의 주소 공간이 아닌 다른 곳에 보관한다. key를 프로세스 주소 공간이 아닌 다른 곳에 보관하는 이유는 key는 plain-text 형태로 저장되어야 하는데, 이 key가 해당 프로세스의 주소 공간에 저장된다면, core dump image를 통해 key값을 노출되기 때문에 key를 해당 프로세스와 다른 주소 공간에 저장한다.

두 번째 system call은 중요한 데이터를 저장하고 있는 변수의 주소 값을 인자로 넘겨받아, 이 주소에 저장되어 있는 데이터를 암호화 하는 system call이다.

세 번째 system call은 암호화된 데이터를 복호화 하는 system call로써, 프로세스가 암호화된 데이터를 저장하고 있는 변수의 주소 값을 인자로 넘겨받아 그 주소에 저장되어 있는 암호화된 데이터를 다시 plain-text 형태로 복호화하는 system call이다.

(그림 2)는 3개의 system call을 소스코드에서 사용하는 간단한 예를 보여준다.

```
-----
system call_for_key_creation();
-----
-----
char passwd[20];
system call_for_encryption(passwd);
-----
-----
system call_for_decryption(passwd);
if(strcmp(str,passwd) == 0){
-----
```

(그림 2) 사용 예

4. 결론

RAM의 취약성은 반드시 해결해야 할 문제이다. 하지만 위에서 제시한 암호화 방식은 완전한 해결책이 아니며, 더 많은 연구가 요구된다. 특히 암호화 및 복호화에 사용되는 비밀 키를 보관하는 방식에 대해서는 더 많은 연구가 필요하다고 본다. 프로세스가 실행 될 때마다 사용자로부터 RAM 암호화에 사용될 비밀 키를 입력 받을 지 또는 OS가 랜덤 방식으로 비밀 키를 생성해서 각 프로세스마다 다른 비밀 키를 적용할지, 즉 비밀 키를 어떻게 생성하는지부터, 이 생성된 비밀 키를 RAM에 어떻게 보관할지에 관한 방안에 대해서 추가적인 연구가 진행되어야 한다.

참고문헌

[1] J. Alex Halderman, Seth D. Schoen, etc. "Lest We Remember: Cold Boot Attacks on Encryption Keys" February. 2008.

[2] Walter Link and Herbert May. "Eigenschaften von MOS-Ein-Transistorspeicherzellen bei tiefen Temperaturen. Archiv f¨ur Elektronik und¨ Ubertragungstechnik," June 1979

[3] Mosafa El-Said and David Connett "On The Insecurity of RAM" 2010

[4] Joshua Hulst "Vulnerabilities in RAM Core Dumps " April. 2009.