기기 보안성의 향상을 위한 모바일 수행 오프로딩의 가능성

이하윤*, 조영필*, 권동현*, 백윤흥*
*서울대학교 전기정보공학부
e-mail : hyyi@sor.snu.ac.kr

The potential of mobile execution offloading for enhanced device security

Hayoon Yi*, Yeongpil Cho*, Donghyun Kwon*, Yunheung Paek*
*Dept. of Electrical and Computer Engineering, Seoul National University

요 약

지금까지 모바일을 위한 본격적인 보안 소프트웨어 연구는 대부분 모바일 기기의 성능 제약 때문에 서버를 활용하는 서버-클라이언트 모델로 진행이 되었다. 본 논문에서는 서버-클라이언트 모델과 비슷하게 서버를 활용하면서도 다른 추가적인 이점을 가질 수 있는 모바일 수행 오프로딩 프레임워크를 활용한 모바일 기기의 보안 가능성에 대해 이야기한다.

1. 서론

모바일 기기의 성능이 발전되면서 전화나 문자 등의 전통적인 휴대장치의 기능에 추가적으로 과거의 휴대장치에서는 수행하기 어려웠던 다양한 전자업무를 처리할 수 있게 되었기 때문에 모바일 기기는 현대인의 일상에 필수적인 요소로 자리잡았다. 하지만 그 편리성과 유용성으로 인해 사용자가 자신의 기기에 중요한 정보를 남기거나 인터넷뱅킹 등의 민감한 업무를 처리하는 빈도가 급증함에 따라 모바일 기기는 보안 공격의 대상으로 떠오르게 되었다.

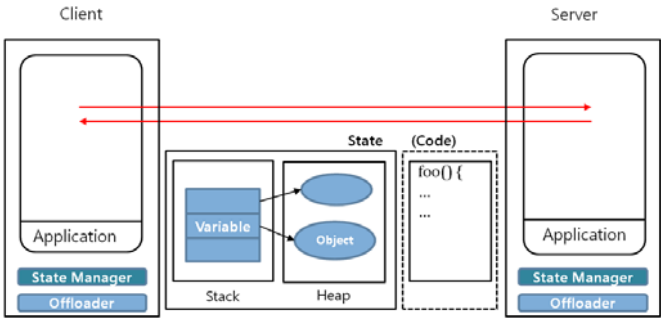
전통적으로 공격대상이 되어 보안 연구가 많이 진행되었던 일반 컴퓨터와는 달리 모바일 기기는 제한된 연산 성능, 메모리 양, 건전지 용량 등을 가지기 때문에 전통적인 보안 연구를 그대로 적용하기 어렵다. 일반 컴퓨터에 비해 연산 능력이 부족하기 때문에 일반 보안 소프트웨어를 실행하게 모바일 기기의 전체적인 성능이 저하되는 현상이 나타나게 된다. 사용자는 일반적으로 모바일 기기에 즉각적인 반응을 기대하며 사용하기 때문에, 작은 느려짐에도 민감하게 반응하므로 이런 접근은 현실적으로 어렵다.

앞서 언급된 특성 때문에 현재의 모바일 보안 소프트웨어는 크게 두 가지 분류로 발전하게 되었는데, 하나는 모바일 기기의 성능에 최대한 영향을 주지 않으면서 간단한 보안 처리를 해주는 소프트웨어이고, 다른 하나는 모바일 기기에서 필요한 정보를 간단히 취합하여 외부 서버장치로 그 정보를 보내서 면밀하게 분석하여 보안검사를 하는 소프트웨어이다. 전자의 경우는 간단한 보안 처리만을 해주는 것이 한계이고 실제로는 후자의 방법이어야 보다 본격적인 보안 처리가 가능한데, 이는 흔히 말하는 서버-클라이언트 모델로 소프트웨어가 개발되게 된다.

서버-클라이언트 모델은 그 소프트웨어를 작성함에 있어 서버 부분과 클라이언트 부분을 모두 개발해야 한다는 점에서 개발자에게 많은 부담을 주게 된다. 최근 몇 년 동안 MAUI[1]나 Clonecloud[2]를 시작으로 이러한 서버-클라이언트 모델과는 달리 서버를 활용하면서도 소프트웨어 개발자에게 일반적인 소프트웨어 개발 이상의 부담을 주지 않는 모바일 수행 오프로딩 프레임워크에 대한 연구가 많이 진행되었다.

본 논문에서는 바로 이 모바일 수행 오프로딩 프레임워크가 모바일 기기의 보안에 도움을 줄 수 있는 가능성을 가지고 있음을 보이고자 한다. 이를 위해 본 논문은 먼저 모바일 수행 오프로딩 프레임워크를 설명한 뒤에 그 프레임워크가 모바일 기기의 보안에 어떻게 도움을 줄 수 있는지를 설명한다.

2. 모바일 수행 오프로딩 프레임워크



(그림 1) 모바일 수행 오프로딩 프레임워크

수행 오프로딩이란 프로그램의 수행 중에 그 수행 자체를 자동적으로 하나의 기기에서 다른 기기로 넘겨주는 것을 이야기한다. 모바일 수행 오프로딩에서는 일반적으로 모바일에서 시작된 프로그램의 수행 도중에 연산이 많은 부분을 서버로 자동적으로 넘겨 수행한 뒤 다시 그 프로그램의 나머지 수행을 모바일로 돌려보내 수행한다.

지금까지 연구된 대부분의 모바일 수행 오프로딩 프레임워크는 그림 1 과 같은 형태로 동작을 하게 된다. 언뜻 보면 일반적인 서버-클라이언트 모델의 소프트웨어와 큰 차이가 없어 보이지만, 그 내부 동작에 있어 주요한 차이점이 존재한다.

서버-클라이언트 모델의 소프트웨어는 서버에서 동작하는 소프트웨어와 클라이언트 기기에서 동작하는 소프트웨어의 코드와 State 가 서로 다른데, 모바일 수행 오프로딩 프레임워크 내에서 동작하는 소프트웨어는 서버와 클라이언트 모두 동일한 코드와 동일한 State 를 공유하게 된다. 여기서 State 라 함은 그림 1 에 나타나 있는 것과 같이 소프트웨어의 수행 과정 가운데 나타나는 프로그램 스택이나 프로그램 힙 메모리에 저장되는 각종 객체들 즉 이름 그대로 프로그램의 상태를 나타내줄 수 있는 자료를 이야기한다.

기존의 서버-클라이언트와는 다르게 모바일 오프로딩 프레임워크에서 동일한 코드로 모바일 기기와 서버에서 유연하게 프로그램이 수행될 수 있는 이유는 코드 내에서 어떤 부분을 서버에서 수행하면 좋을지를 프레임워크에서 분별하여 자동적으로 기기와 서버를 오가며 프로그램을 수행하기 때문이다. 그 분별 방식은 프레임워크에 따라 달라질 수 있는데, 프로그래머가 간단히 표시해주는 방식과 프레임워크에서 자동적으로 수행 중이나 수행 전에 코드 분석을 하여 분별하는 방식이 있다. 이러한 오가는 동작을 관리해주는 것이 그림 1 에 나타난 프레임워크 내의 Offloader 모듈이다.

이 때 코드 내에서 어느 부분은 어디에서 수행하면 좋을지 결정이 되었으면 실제로 오가며 수행하는 것이 문제인데, 그 것을 가능하게 해주는 정보가 바로 State 이다. 앞서 언급했듯이 State 는 프로그램의 상태를 나타내주는 정보이기 때문에 이 정보만으로 프로그램의 상태를 재구성할 수 있게 된다. 오프로딩 프레임워크는 코드 수행 중에 서버로나 다시 모바일로 오프로딩이 일어나야 할 때 그 때까지 수행된 프로그램의 상태를 재구성할 수 있도록 State 를 넘겨주게 된다. 이를 받은 모바일이나 서버 측의 프레임워크에서는 State 를 이용하여 마치 그 곳에서 지속적으로 프로그램이 수행된 것처럼 프로그램의 상태를 재구성하여 오프로딩이 수행된 지점부터 코드를 마저 수행하게 된다. 이런 State 를 넘겨줄 수 있도록 모으거나 넘겨받은 State 를 통해 프로그램의 상태를 재구성하는 것을 관리하는 일은 그림 1 의 프레임워크에 나타난 State Manager 가 해주게 된다.

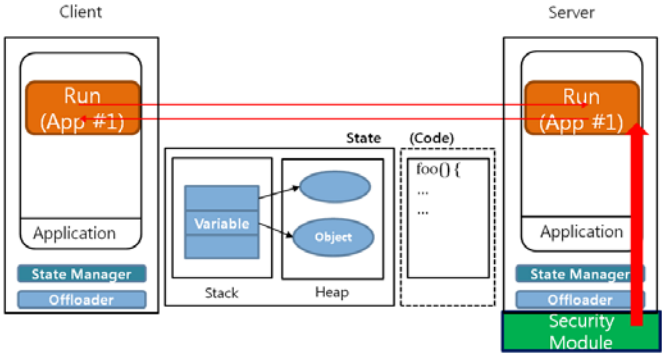
이러한 일련의 과정을 모바일 오프로딩 프레임워크가 자동적으로 해주기 때문에, 프레임워크 내에서 동작하는 소프트웨어를 작성하는 개발자는 그 코드가 어떻게 서버를 활용하면서 모바일 기기에서 동작할지를 생각하지 않아도 된다. 개발자는 마치 모바일 기기 자체에서 처음부터 끝까지 프로그램이 수행되는 것처럼 단일 코드만을 작성하게 되므로 서버의 성능을 활용할 수 있으면서도 개발의 부담을 덜 수 있게 된다.

모바일 오프로딩 프레임워크에서 또 하나 중요한 특징은 개발자가 정한 특정한 목적성을 가지고 프레임워크가 소프트웨어의 수행과정에서 서버로의 오프로딩을 결정한다는 점이다. 기존 연구들에서는 대부분 프로그램의 수행 시간을 단축하거나 모바일 기기의 전력 소모를 줄이기 위한 목적으로 프레임워크가 만들어졌지만[3], 기존 프레임워크에 활용되었던 각종 기술들을 보안의 목적성에 맞춰 활용하면 모바일 기기의 보안을 위한 프레임워크도 구성이 가능할 것으로 보인다.

3. 모바일 수행 오프로딩 프레임워크의 보안 가능성

기존에 모바일 보안을 위해 서버를 활용한 경우는 자료를 서버로 보내서 검사를 하거나[4], 모바일 기기의 동작을 기억했다가 서버에서 이를 재현해보면서 수행해보면서 보안 검사를 하는[5] 등의 방식이 있었다. 이들은 모바일 기기의 성능을 크게 저해하지 않으면서 서버의 성능을 십분 활용하는 면에서 좋은 접근들이었지만, 모바일 수행 오프로딩 프레임워크를 활용한다면 이들보다 더 나은 모바일 기기 보안이 가능할 것이다.

기존의 연구들은 보통 서버에서 따로 한번 미리 수행해보거나, 이미 수행되었던 동작을 재현하거나 하는 등으로 모바일 기기에 도는 프로그램의 보안성을 검사했다. 하지만 모바일 수행 오프로딩 프레임워크를 활용한다면 실제 사용자가 수행하는 프로그램의 동작 중에 그 프로그램이 서버를 사용하기 위해 구현되어 있지 않았더라도 자동적으로 그 프로그램을 보안 검사를 위한 서버로 넘겨주는 것이 가능하다. 이런 환경에서는 기존의 연구처럼 프로그램 수행 전에 따로 수행을 해보는 시간을 절약할 수도 있고, 프로그램 수행을 재현하는 것에 비해 훨씬 즉각적으로 보안 공격에 대한 반응과 대처를 할 수 있을 것이다.



(그림 2) 보안을 위한 모바일 오프로딩 프레임워크

예상되는 프레임워크의 형태는 그림 2 와 같이 기존의 모바일 수행 오프로딩 프레임워크의 서버 측에

보안 검사를 위한 추가적인 보안 모듈이 추가된 형태이다.

이러한 프레임워크 내에서는 보안 소프트웨어의 작성자는 더 이상 모바일 기기에서 서버로 정보를 가져오는 것에 대한 고민을 하지 않아도 된다. 프레임워크에서 자동적으로 모바일에서 수행되는 소프트웨어를 서버에서 재구성할 수 있도록 해주기 때문에 보안 소프트웨어 개발자는 프레임워크에 어떠한 조건이 충족되면 소프트웨어를 서버로 넘겨달라고 지정해주기만 하면 된다.

또한 프레임워크에 추가되는 보안 모듈에는 기존의 서버-클라이언트 모델과 마찬가지로 전통적인 형태의 컴퓨터 보안을 위한 연구를 기반으로 한 프로그램들을 모바일 환경에 맞춰 조금만 변형하면 활용이 가능하다. 더 나아가 이미 모바일 수행 오프로딩 프레임워크에서 오프로딩을 위해 State 를 모으게 되는데, 이러한 프로그램의 State 를 분석하는 형태의 보안 연산이 추가적으로 개발될 수 있을 것이다. 프로그램의 State 는 현재 프로그램의 상태를 가장 잘 표현해주기 때문에, 악성 프로그램의 분석에 유용한 정보가 될 수 있다.

4. 결론

본 논문에서는 증가되는 모바일 기기의 보안공격에 대응하기 위한 보안 소프트웨어 개발을 위해 전통적인 서버를 활용하는 보안을 더 손쉽게 수행할 수 있으면서 동시에 State 정보를 활용한 보안 대처를 추가적으로 할 수 있는 모바일 수행 오프로딩 프레임워크가 활용될 수 있는 가능성이 있음을 탐색했다.

Acknowledgement

본 연구는 교육과학기술부/한국과학재단 우수연구센터 육성사업(과제번호 2012-0000470), 중소기업청에서 지원하는 2012 년도 산학연공동기술개발사업(No. C0019562), 미래창조과학부 및 한국산업기술평가관리원의 산업융합원천기술개발사업(정보통신) [No. 10047212, 1kB 이하 암호문 간의 연산을 지원하는 동형 암호 원천 기술 개발 및 응용 연구] 및 IDEC 의 지원을 받아 수행하였습니다.

참고문헌

[1] Eduardo Cuervo et al, "MAUI: making smartphones last longer with code offload", International conference on Mobile systems, applications, and services, 2010

[2] Byung-gon Chun et al, " CloneCloud: elastic execution between mobile device and cloud ", Conference on Computer systems, 2011

[3]Karthik Kumar et al, "A Survey of Computation Offloading for Mobile Systems", Mobile Networks and Applications, 2013

[4]Jon Oberheide et al, "Virtualized In-Cloud Security Services for Mobile Devices", Workshop on Virtualization in Mobile Computing, 2008

[5]Georgios Portokalidis et al, "Paranoid Android: Versatile Protection For Smartphones", Annual Computer Security Applications Conference, 2010