Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.29, NO.3, Jun. 2019 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2019.29.3.613

위협모델링과 공통평가기준을 활용한 인포테인먼트의 안전한 업데이트 보안요구사항 분석*

강 수 영,† 김 승 주‡
고려대학교 정보보호대학원

Analysis of Security Requirements for Secure Update of IVI(In-Vehicle-Infotainment) Using Threat Modeling and Common Criteria*

Soo-young Kang,† Seung-joo Kim‡
Center for Information Security Technologies(CIST), Korea University

요 약

차량 내부에 탑재되어 있는 인포테인먼트는 네비게이션뿐만 아니라 앱을 설치하여 다양한 기능을 제공하고 있으며, ECU에 명령을 전송하여 차량 전체를 제어하는 중요한 역할을 수행하고 있다. 인포테인먼트는 다양한 무선 통신 프로토콜을 지원하며, 이를 통해 해커들이 접근하기 위한 공격 표면이 되고 있다. 인포테인먼트가 해킹되어 악성 소프트웨어가 설치되면 차량 제어권을 획득하여 ECU에 악의적인 명령을 전송하여 운전자의 생명에도 영향을 끼칠 수 있다. 따라서 인포테인먼트 소프트웨어 업데이트에 대한 보안성 및 신뢰성을 검증하기 위한 방안이 필요하다. 보안성 및 신뢰성을 제공하기 위하여 개발 초기 단계부터 보안을 고려한 SDL에 따라 개발되어야 하며, 위협모델링 기법을 적용하여 체계적인 보안요구사항을 도출해야 한다. 따라서 본 논문은 인포테인먼트의 업데이트 보안요구사항을 도출하기 위하여 위협모델링을 수행하고, 공통평가기준의 보안기능요구사항 및 보증요구사항을 도출하여, 인포테인먼트에서 안전한 소프트웨어 업데이트를 위한 기준을 제시한다.

ABSTRACT

In-Vehicle Infotainment provides navigation and various functions through the installation of the application. And infotainment is very important to control the entire vehicle by sending commands to the ECU. Infotainment supports a variety of wireless communication protocols to install and update applications. So Infotainment is becoming an attack surface through wireless communcation protocol for hacker's access. If malicious software is installed in infotainment, it can gain control of the vehicle and send a malicious purpose command to the ECU, affecting the life of the driver. Therefore, measures are needed to verify the security and reliability of infotainment software updates, and security requirements must be derived and verified. It must be developed in accordance with SDL to provide security and reliability, and systematic security requirements must be derived by applying threat modeling. Therefore, this paper conducts threat modeling to derive infotainment update security requirements. Also, the security requirements are mapped to the Common Criteria to provide criteria for updating infotainment software.

Keywords: Threat Modeling, STRIDE, IVI(In-Vehicle-Infotainment), SOTA(Security Over-The-Air), CC(Common Criteria)

Received(03. 22. 2019), Accepted(06. 11. 2019)
* 본 연구는 과학기술정보통신부 및 정보통신기기획평가원의 대학 ICT연구센터육성지원사업의 연구결과로 수행되`ITP-2019-0-00403)
† 주저자, bbang814@gmail.com
‡ 교신저자, skim71@korea.ac.kr (Corresponding author)

I. 서 론

인포테인먼트(Infotainment)란 정보(Information)와 오락(Entertainment)의 합성어로 네비게이션 기능뿐만 아니라 인터넷, 메일, 미디어, 카메라 등의 다양한 기능을 제공하는 차량 내 기기이다. 인포테인먼트는 무선으로 소프트웨어를 설치하고 업데이트해야되기 때문에 다양한 통신 프로토콜을 지원하고 있으며, 해커들은 이를 이용하여 차량 제어권 획득을 시도하고 있다. 특히 다양한 종류의 소프트웨어를 설치하기 때문에 악성 소프트웨어를 설치하게 되면 차량 제어권을 획득할 수 있어 이에 대한 보안성 및 신뢰성 검증이 필요하다[1].

보안성 및 신뢰성 제공을 위하여 IoT 및 CPS 환경에서 사용되는 보안제품은 SDL(Security Development Lifecycle)에 따라 보안요구사항 분석, 설계 구현, 테스트, 유지보수 단계를 거쳐 개발된다. 이 중 가장 처음 단계인 보안요구사항 분석단계에서 보안요구사항이 잘못 도출되었을 경우 구현 및 테스트 단계에서 문제가 발생할 수 있으며, 이를 해결하기 위한 시간 및 비용이 증가한다. 조사에 따르면 요구사항 분석 단계에서 55%, 설계 단계에서 25%, 구현 단계에서 15%, 기타 단계에서 5%의 결함이 발견된다고 한다[2]. 요구사항 분석 단계에서 가장 많은 결함이 발견되고 이 결함이 테스트 단계에서 발견될 경우 결함을 해결하기 위한 비용은 기하급수적으로 증가하게 된다. 요구사항을 도출하는 과정은 매우 중요하며, 보안요구사항을 도출하기 위하여 국내외에서는 위협모델링을 활용하여 보안 위협을 도출하고 보안 위협에 대응할 수 있는 보안요구사항을 도출한다.

본 논문은 차량 환경에서 보안성 및 신뢰성이 제공되어야 하는 핵심 모듈인 인포테인먼트에 대해 위협모델링 기법을 적용하여 안전하게 업데이트하기 위한 보안요구사항을 분석 및 도출한다. 본 논문은 2장에서 관련연구에 대해 소개하고, 3장에서 위협모델링을 활용하여 위협을 도출하고 4장에서 공통평가기준에 의거하여 보안기능요구사항과 보증요구사항을 도출한다. 5장에서는 본 논문의 결론 및 향후 연구 방향을 제시하도록 한다.

II. 관련 연구

본 장에서는 차량 내부에서 ECU들을 제어하고 다양한 기능을 제공하는 인포테인먼트와 소프트웨어를 안전하게 업데이트 하는 기술인 SOTA, 이를 분석할 위협모델링에 대하여 서술하고자 한다.

2.1 IVI(In-Vehicle-Infotainment) 연구 동향

인포테인먼트는 Wi-Fi, 블루투스 및 이동통신과 같은 무선 통신 기술 및 스마트 기기가 발전함과 동시에 차량 탑승자에게 더 나은 편의성을 제공하기 위한 목적으로 개발되고 있다. 하지만 다양한 네트워크를 지원하기 때문에 공격자가 접근할 수 있는 접점이 되고 있으며, 이에 대한 해킹 사례가 발표되고 있다.

2015년 Blackhat USA 2015에서는 미국의 유명한 보안 엔지니어 Charlie Miller, Chris Valasek이 Jeep Cherokee의 인포테인먼트인 Uconnect에 원격으로 접속하여 차량 제어권을 획득한 해킹 과정을 시연하였다. 차량 제어권 획득한 후 라디오, 에어컨, 와이퍼 등을 무작위로 조정하여 인포테인먼트의 해킹 가능성과 보안에 대한 필요성을 부각시켰다[3].

2016년 중국 보안팀인 텐센트 연구원들은 테슬라 S의 인포테인먼트 Wi-Fi 핫스팟에 악의적으로 접속하여 악성코드 감염을 통해 차량 제어권을 획득한 해킹 과정을 시연하였다. 차량 제어권 획득 후 차량의 브레이크를 원격 제어하거나 트렁크를 여는 시연을 하여 해킹 위협에 대해 공표하였다[4].

인포테인먼트에 대한 중요성이 부각됨에 따라 산업 표준화 비영리 단체인 GENIVI Alliance에서는 인포테인먼트에 대한 오픈소스 기반의 플랫폼을 표준화하고 있다[5]. 최근 대형 차량 제조사들은 GENIVI Alliance에 참여하고 있지 않은 차량 부품 회사의 제품을 납품받지 않겠다고 선언하여 유관 기관들이 점차 참여하고 있는 추세이다. 특히 인포테인먼트에서 소프트웨어가 설치되거나 패치될 경우 안전하게 업데이트 하는 SOTA(Secure Software Updates Over-The-Air)에 대한 부분을 표준화하여 악의적인 목적의 소프트웨어가 설치될 수 없도록 기준을 정의하고 있다.

2.2 SOTA 연구 동향

차량 제조 업체에서 리콜 조치를 위해 소프트웨어 결함을 보정하는 조치를 취하는데 많은 비용이 소모되고 있다. 이러한 비용을 줄이기 위해 무선으로 안

전하게 소프트웨어를 업데이트 하는 기술을 SOTA라고 한다[6]. 시장조사업체 IHS(Information Handling Services)에서 최근 발표한 오토모티브 리포트에 따르면 SOTA를 통해 절감할 수 있는 비용이 2015년에 약 27억 달러에서 2022년 350억 달러 이상으로 증가될 것으로 전망하고 있다. 시장 규모가 커지고 있으나 경량화에 초점을 두고 개발하여 보안을 접목한 연구가 지속적으로 진행되고 있다[7].

2016년 Euromicro Conference 2016에서는 차량에서 효율적이고 신뢰성을 제공할 수 있는 소프트웨어 업데이트 방식을 제안하였다. 최초 소프트웨어는 제조사로부터 다운로드 받아 설치하고, 패치가 발생하면 근처에 있는 다른 차량으로부터 소프트웨어를 다운로드 받아 설치하는 방식이다. 근처 차량과의 경량화된 통신을 통해 효율성을 제공하고 소프트웨어의 해쉬 체인을 이용하여 무결성을 보장한다[8].

2017년 유명 보안 컨퍼런스인 USENIX Security 2017에서는 스위스의 로잔 연방 공과대학 연구원들이 분산 소프트웨어 업데이트 프레임워크인 CHAINIAC을 제안하였다. 블록체인과 skiplist를 결합시킨 skipchain을 이용하여 업데이트에 대한 투명성 및 무결성을 제공하고, 중앙에서 업데이트를 제어할 때 발생할 수 있는 장애를 해결하기 위하여 전자서명을 사용하고, reply 공격을 방지하기 위하여 타임스탬프를 사용하여 업데이트에 대한 보안을 향상시킨 방식이다[9].

2017년 차량 관련 컨퍼런스인 AMAA 2017에서는 오스트레일리아 연구원들이 블록체인 기반 소프트웨어를 업데이트하는 방식을 제안하였다. 중간 배포지인 (CH)Cluster Head를 지정하고, CH에서 차량에 소프트웨어를 배포하여 제조사의 부하를 줄이고 소프트웨어가 변조되지 않음을 검증하기 위하여 블록체인 기술을 접목하였다[10].

2.3 위협모델링 연구 동향

위협모델링이란 제품의 데이터흐름에 따라 내부 구조를 파악하고 발생할 수 있는 위협을 식별하는 방법론이다. SDL에 따라 보안제품을 개발할 때 설계 단계부터 발생 가능한 취약점을 식별하고 잠재적인 위협을 도출하기 위해 사용한다.

위협모델링 기법 중 대표적인 기법은 마이크로소프트에서 개발한 STRIDE로 가장 범용적으로 사용되며 네트워크 환경에서 발생할 수 있는 보안 위협들을 다루고 있다[11]. STRIDE에서는 Table 1과 같이 6가지 보안속성에 대응하는 보안 위협에 대해 분석이 가능하며, 서버, 클라이언트 환경에서 발생할 수 있는 기본적인 보안 위협을 다루고 있어 인포테인먼트 업데이트 환경 분석 시 가장 적합한 기법이다.

Table 1. STRIDE Description

Security Property	Threat	Description
Authenticity	Spoofing	Impersonating something or someone else
Integrity	Tampering	Modifying data or code
Non-repudiability	Repudiation	Claiming to have not performed an action
Confidentiality	Information disclosure	Exposing information to someone not authorized to see it
Availability	Denial of Service	Deny or degrade service to users
Authorization	Elevation of Privilege	Gain capabilities without proper authorization

벨기에 가톨릭 대학에서 개발된 LINDDUN은 프라이버시 관점에서 위협모델일을 수행하는 기법이다[12]. LINDDUN을 수행하기 위해서는 DFD 정의

Table 2. LINDDUN Description

Threat	Description
Linkability	Not being able to hide the link between two or more identities of information
Identifiability	Being able to sufficiently identify the subject within a set of subjects
Non-repudiation	Not being able to deny a claim
Detectability	Being able to sufficiently distinguish whether an item of interest exists or not
Disclosure of information	Critical data is exposed
Unawareness	unaware of the consequences of sharing information.
Non-compliance	Not being compliant with legislation, regulations, and corporate policies.

하고 DFD 요소에 프라이버시 위협을 매핑시킨 후 위협 시나리오를 식별하고, 위협의 우선순위를 선정한다. 식별된 위협의 완화 전략 및 대응 솔루션을 제시하여 위협 모델링을 수행한다. LINDDUN에서는 Table 2와 같이 7가지 항목에 대한 보안 위협을 도출할 수 있다.

Trike는 데이터 흐름의 주도하는 자산을 식별하고 자산에 대해 CRUD(Create, Read, Update, Delete) 수행에 따른 위험도를 산출하는 위험 관리 기법이다[13]. Trike는 도구를 지원하기 때문에 분석 대상에서 발생할 수 있는 위험도를 용이하게 분석하고 한눈에 식별이 가능하다. CRUD 속성에 대한 내용은 다음과 같다.

Table 3. Trike CRUD Description

Threat	Description
C Create	Create a new object. For example, add a row to a database table, call Object new, or allocate a memory buffer
R Read	View or use the contents of an object. For example, select values from a database, open and read a file, or receive network traffic
U Update	Change the contents of an object. For example, update values in a database, set a variable, or write to a file
D Delete	Remove or destroy an object. For example, delete a row from a database table, remove a file, or free memory

III. 위협모델링

인포테인먼트는 네비게이션 기능을 중점적으로 제공해왔으나, 최신 차량에 탑재되어 있는 인포테인먼트는 네비게이션 뿐만 아니라 미디어, 인터넷, 전화, 센서 컨트롤 등 다양한 기능을 포함하고 있다. 특히 인포테인먼트는 무선 통신이 가능하기 때문에 해커의 공격 대상이 되고 있으며, 인포테인먼트에 접근하여 변조된 소프트웨어 및 펌웨어로 업데이트 할 경우 차량에 포함되어 있는 ECU를 제어할 수 있기 때문에 운전자에게 큰 피해를 발생시킬 수 있다. 따라서 위협모델링을 통해 인포테인먼트 기능 중 업데이트 기능에서 발생할 수 있는 보안 위협을 모두 식별하고 이에 대응할 수 있는 방안을 도출해야 한다. 위협모델링은 2.3에서 선정한 STRIDE 기법을 이용하며, DFD, STRIDE, Attack tree, Risk management를 통해 Checklist 및 보안기능요구사항, 보증요구사항을 도출하고자 한다.

3.1 구조도

위협모델링을 수행하기 위해서는 시스템 구성요소 및 구조도를 파악하고 각 구성요소 간 데이터 흐름을 도식화한 DFD를 작성해야 한다. 객관적인 보안요구사항 도출을 위하여 국제 표준 및 산업계 컨소시엄에서 수행하고 있는 플랫폼을 조사하였으며, 이 중 GENIVI Alliance에서 오픈소스 기반의 인포테인먼트 플랫폼에 대해 분석하였다[14].

Table 4. The Comparison of Infotainment

OEM	Infotainment	SOTA	Architecture	GENIVI
Toyota	Entune	USB/ Infotainment	3	
BMW	iDrive	USB/ Infotainment	3	√
Mercedes-Benz	MBUX (Mercedes-Benz Infotainment System)	USB/ Infotainment	3	
Audi	Audi MMI	USB/ Infotainment	3	
Land Rover	Land Rover	USB/ Infotainment	3	√
Tesla	Tesla	USB/ Infotainment	2	
Chevrolet	Mylink	USB/ Infotainment	3	√
Jeep	Uconnect	USB/ Infotainment	3	
Volkswagen	Volkswagen	USB/ Infotainment	3	
Ford	SYNC3	USB/ Infotainment	3	
Hyundai	Blue Link	USB/ Infotainment	1 or 3	√
KIA	UVO	USB/ Infotainment	1 or 3	√

또한 분석 대상을 선정하기 위해, 브랜드 자산 가치를 전문적으로 연구하는 다국적 시장 조사 회사인 Kantar Millward Brown에서 발표한 "Most valuable global car brands" 10개와 국내 점유율이 가장 높은 2개, 총 12개의 OEM(Original Equipment Manufacturer)을 선정하였다.

선정된 12개의 OEM(Original Equipment Manufacturer) 차량 내 인포테인먼트를 분석하고, GENIVE 플랫폼 준수 여부에 대해서 조사하여 구조도를 파악하였다. 인포테인먼트는 소프트웨어 및 펌웨어를 업데이트하기 위하여 OEM 서버로부터 패치 파일을 다운로드 받아 설치한다. 다운로드 받는 방식은 총 3가지 방식으로 구분된다.

첫 번째 방식은 USB, 스마트폰과 같은 저장매체에 다운로드 받는 방식으로 이전에 보급된 차량에서 가장 많이 사용하고 있는 방식이다. 두 번째 방식은 인포테인먼트의 무선 네트워크를 통해 직접 다운로드 받는 방식으로 유일하게 Tesla만이 이 방식으로 사용하고 있다. 세 번째 방식은 첫 번째 방식과 두 번째 방식이 합쳐진 방식으로 일부 소프트웨어의 경우 저장매체를 사용해서 다운로드 받아 설치하고, 일부 소프트웨어는 인포테인먼트의 무선 네트워크를 통해 다운로드 받아 설치하는 방식으로 최신 보급된 차량

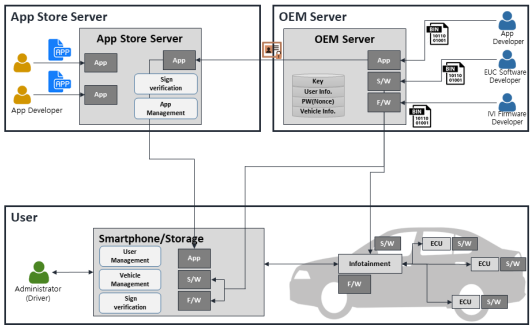

Fig. 1. The Analysis Target Infotainment

에 가장 많이 사용하고 있는 방식이다. 세 번째 방식은 첫 번째 방식과 두 번째 방식의 보안 위협을 포함하고 있기 때문에 세 번째 방식을 선정한다.

3.2 DFD(Data Flow Diagram)

DFD란 위협모델링 중 가장 첫 번째 단계로 1967년 Larry Constantine에 의해 제시된 개념으로, 시스템을 분석할 때 각 프로세스 간 데이터 흐름을 도식화하는 방법이다. 해커가 시스템을 공격할 때 데이터 흐름 경로에 따라 접근하기 때문에 데이터의 흐름이 명확하게 도식화되어야 하며, 주요 요소는 프로

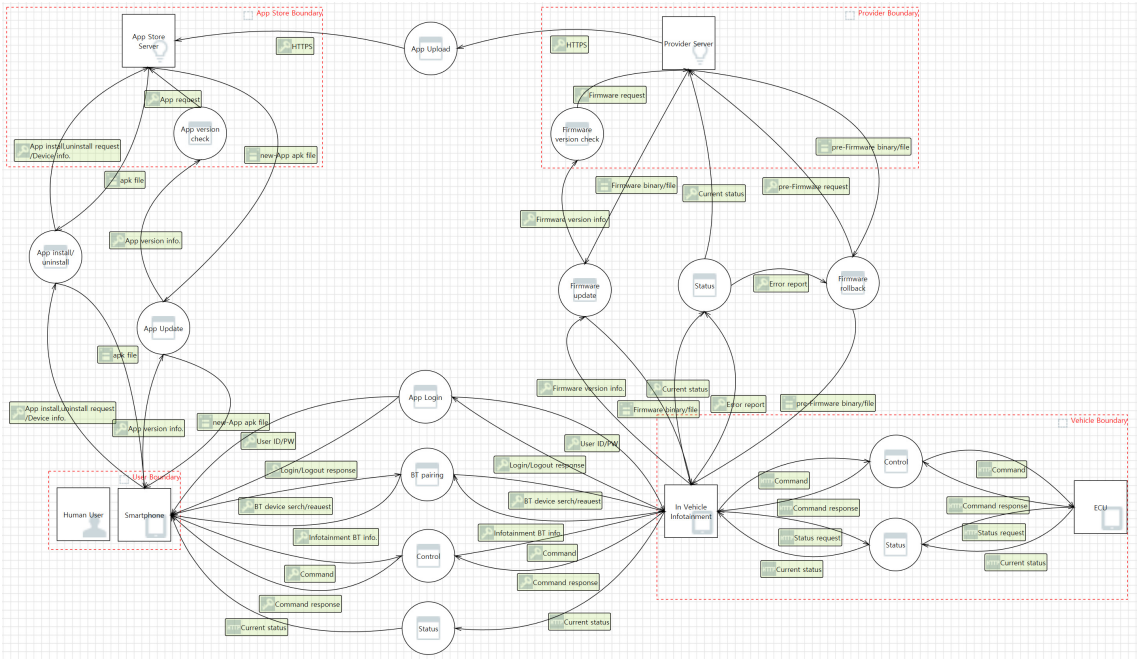

Fig. 2. DFD(Data Flow Diagram)

세스(Process) 데이터 흐름(Data flow), 데이터 저장소(Data Store), 외부 객체(External Entity)로 구성된다.

인포테인먼트에 대한 DFD는 세 번째 방식을 기반으로 작성하였으며, 해당 방식은 도요타, BMW, 벤츠, 아우디, 랜드로버, 시보레, 폭스바겐 등 대형 OEM이 채택하고 있는 구조이다. OEM 서버는 소프트웨어, 펌웨어가 패치될 경우 패치된 버전을 저장하고 있으며, 인포테인먼트에서는 주기적으로 버전을 체크하며, 패치가 존재할 경우 다운로드 받아 설치하고 장애가 발생 시 안전한 최신 버전으로 롤백한다. 대상 구조도 기반 DFD는 Fig. 2와 같다.

전체 DFD 중 가장 중요한 부분은 SOTA 서버와 클라이언트로, 각 ECU들은 클라이언트에 포함되어 있다. 소프트웨어 업데이트에 가장 중요한 기능은 현재 버전을 확인하고, 서버에 있는 패치 파일이 더 상위 버전일 경우 패치 바이너리 전체를 다운로드 받을 수도 있고, 패치된 일부 바이너리만 다운로드 받아 설치할 수 있다. 패치가 설치된 이후 크래시 발생 시 보안 영역에 저장되어 있는 패치 바이너리 중 가장 최근에 정상적으로 동작했던 패치 바이너리로 롤백하거나 아니면 서버에 이전 버전의 패치 바이너리를 요청하거 다시 다운로드 받아 설치한다. 인포테인먼트 뿐만 아니라 연결되어 있는 ECU 소프트웨어가 업데이트 될 경우에도 동일한 방식으로 동작하며 ECU에 패치 바이너리를 포워딩한다. 이러한 핵심 기능은 Fig. 3과 같다.

OEM 서버에서 Admin HMI는 인가된 관리자로부터 API call을 수행하는 인터페이스이며, Device Registry는 클라이언트를 등록하고 차량 식별자(VIN: Vehicle Identification Number)와 인포테인먼트 식별자(UUID: Universal Unique Identifier)를 매핑하고, External Resolver는 패

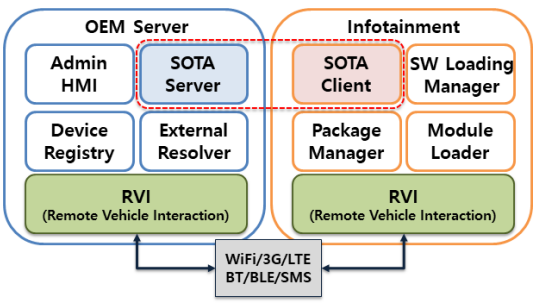

Fig. 3. SOTA Core module

Table 5 .DFD Components

Component	Symbol	Description
Process	○	Receives input data and produces output with a different content or form
Data flow	→	Used in a DFD to represent a situation when the system must retain data
Data storage	═	Used in a DFD to represent a situation when the system must retain data
External entity	□	Person, department or other information System that provides data to the system or receives outputs from the system

치 바이너리를 관리하며 패치 식별자(packageID)와 패치가 필요한 차량을 매핑하는 역할을 수행한다. 인포테인먼트에 포함되어 있는 Package Manager는 패치 바이너리를 설치, 업그레이드, 삭제하는 핵심 역할을 수행하며, SW Loading Manager는 패치 알림, 사용자 승인 요청, 패치 로딩 준비와 관련된 역할을 수행하며, Module Loader는 ECU로 패치 바이너리를 포워딩하고 로딩하는 역할을 수행한다.

3.3 Attack Library

위협모델링을 수행하며 공격 기법에 대해 명확히 하기 위하여 알려진 취약점 및 공격 기법들을 모두 수집하여 나열한 것을 Attack Library라고 한다.

Attack Library는 2가지 요소를 고려하여 수집해야 한다. 첫째, 청중에 대한 수준으로, Attack Library의 내용과 구조, 공격 수준에 대해 큰 영향을 미친다. 둘째, 추상화 및 상세화에 대해 고려하여 공격 기법들을 수집해야 한다. 추상화를 시킬 경우 하위 항목에 대한 공격 기법을 모두 포함하므로 더 많은 공격 기법을 수집할 수 있으나 공격에 대한 상세 내용이 부족할 수 있다. 상세화 시킬 경우 명확한 공격 기법을 알 수 있지만 범위가 줄어들며 시간이 많이 소요될 수 있다. 이를 고려한 인포테인먼트에 대한 Attack Library는 Table 6과 같다.

Table 6 . Attack Library

Type	Author	Title	Reference
Public	MITRE	CVE(Common Vulnerabilities and Exposures)	[15]
	MITRE	CWE(Common Weakness Enumeration)	[16]
	MITRE	CAPEC(Common Attack Pattern Enumeration and Classification)	[17]
	OWASP	Embedded Application Security	[18]
Book	Alex Omar	The Car Hacker's Handbook A Guide for the Penetration Tester	[19]
Standard	ITU-T	Secure software update capability for intelligent transportation system communication devices	[20]
Paper	Keen Security Lab of Tencent	OVER-THE-AIR: HOW WE REMOTELY COMPROMISED THE GATEWAY, BCM, AND AUTOPILOT ECUS OF TESLA CARS	[21]
	Keen Security Lab of Tencent	FREE-FALL: TESLA HACKING 2016	[4]
	Charlie Miller, Chris Valasek	Remote Exploitation of an Unaltered Passenger Vehicle	[3]
	Bjoern M. Luettmann and Adam C. Bender	Man-in-the-Middle Attacks on Auto-Updating Software	[22]
	Ang Cui, Michael Costello and Salvatore J. Stolfo	When Firmware Modifications Attack: A Case Study of Embedded Exploitation	[23]
	K. Chen	Reversing and Exploiting an Apple Firmware Update	[24]
	중략		
Technical document	New York University (Laboratory of Secure Systems)	The Update Framework	[25]
	WONDER HOWTO	How to Hijack Software Updates to Install a Rootkit for Backdoor Access	[26]
	Institute for Defence Studies and Anlayses	THE PETYA CYBER ATTACK	[27]
	CrySyS Lab(Laboratory of Cryptography and System Security)	sKyWIper (a.k.a. Flame a.k.a. Flamer): A complex malware for targeted attacks	[28]

3.4 STRIDE

STRIDE란 1999년 Microsoft에서 개발한 위협 모델링 기법으로 위장(Spoofing), 변조(Tampering), 부인(Repudiation), 정보 노출(Information Disclosure), 서비스 거부(Denial of Service), 권한 상승(Elevation of Privilege)에 대한 위협을 도출할 수 있다.

위협모델링 기법은 다양하지만 인포테인먼트 업데이트 기능은 서버, 클라이언트 구조를 따르고 있기 때문에 기본적인 네트워크 환경에서의 위협을 다루는 STRIDE 기법을 선정한다. 인포테인먼트 업데이트에 대한 STRIDE는 Table 7과 같다.

Table 7. STRIDE for Infotainment DFD

Type	No	Name	STRIDE	Threat Description	Threat No
Entity	E1	User	S	The threat that intercepts and steals passwords through sniffing	T1
				The threat that gussing attack to acquire a password to trick administrators to access	T2
			R	The threat that the attacker denies access and connection	T3
Entity	E2	Developer	S	The threat that an unauthorized developer uploads a patch file via spoofing attack.	T4
			R	The threat that an attacker denies connection attempts and connections	T5
Process	P3	Software Update	S	An attack that an attacker spoofs patch files during an update	T13
			T	The threat that the patch file is being modified during the update	T14
			R	The threat that denies updates to patch files	T15
			I	The threat that exposes sensitive information in patch files	T16
			D	The threat that disables the update by not updating the patch file in the event of a failure	T17
			E	The threat of updating the patch file by another authorized administrator	T18
Process	P4	Software Rollback	S	The threat that an attacker rolls back to a random version	T19
			T	The threat that modifies information needed for rollback	T20
			R	The threat that repudiation received information needed for rollback	T21
			I	The threat that expose information needed for rollback	T22
			D	The threat that disables the rollback by modifying the information necessary to rollback in the event of a failure	T23
			E	The threat that unauthorized user is rollback	T24
Process	P5	Software version check	T	The threat that version information is tampered	T25
			I	The threat that version information is exposed	T26
			D	The threat that modifies version information in the event of a failure, making the update more difficult	T27
				Omit	
Data Store	D1	In Vehicle Infotainment	T	The threat of modifying files and memory in infotainment system	T87
			R	The threat to repudiation access to the infotainment system(system file, memory, etc.)	T88
			I	The threat of exposed files and memory in infotainment system	T89
			D	The threat that makes user cannot get service properly due to not being able to access system files and memory of infotainment.	T90
Data Store	D2	OEM Server	T	The threat is tampered to important information on the OEM server(user information, vehicle information, encryption key, etc.)	T91
			R	The threat to repudiation access to the OEM server(system file, memory, etc.)	T92
			I	The threat is exposed to important information on the OEM server(user information, vehicle information, encryption key, etc.)	T93
			D	The threat that makes user cannot get service properly due to not being able to access OEM's severs	T94

3.5 Attack Tree

STRIDE를 통해 도출한 위협과 발생할 수 있는 알려진 공격 방법을 수집한 Attack Library를 기반으로 공격 시나리오를 도식화한 Attack Tree를 도출한다[29]. 공격 시나리오는 OEM 서버와 인포테인먼트 분류되며, 이에 대해 자세한 Attack Tree는 Table 8과 같으며, 각 공격 노드들은 STRIDE에 의해 도출한 위협과 매핑되어 일관성과 완전성을 제공한다.

Table. 8. Attack Tree

OEM Server				
1	OEM update server attack			
	1.1	Important information leakage		
		1.1.1	Packet dump	
			1.1.1.1	Version information sniffing
			1.1.1.2	Sniffing critical information in the package file
			1.1.1.3	Administrator account sniffing
		1.1.2	Packet spoofing	
			1.1.2.1	Version information tampering
			1.1.2.2	Package file tampering
			1.1.2.3	Digital Signature Certificate tampering
		1.1.3	Obtain DB information	
			1.1.3.1	SQL Injection
	1.2	Denial of service		
		1.2.1	Resource consumption	
			1.2.1.1	SYN Flooding
			1.2.1.2	Smurf attack
			1.2.1.3	Ping of death
			1.2.1.4	Land attack
			1.2.1.5	Teardrop attack
Infotainment				
2	Infotainment device attacks			
	2.1	Important information leakage		
		2.1.1	Obtain storage data	
			2.1.1.1	Reuse of memory
			2.1.1.2	Physical access
	2.2	Denial of service		
		2.2.1	Resource consumption	
			2.2.1.1	Infinite loop
		2.2.2	Memory tampering	
			2.2.2.1	Buffer overflow attack
	2.3	Root acquisition		
		2.3.1	Firmware acquisition	
			2.3.1.1	Internet download
			2.3.1.2	UART port dump
			2.3.1.3	JTAG port dump
			2.3.1.4	Memory dump
		2.3.2	Firmware tampering	
			2.3.2.1	Malicious code injection

3.6 Risk Management

위협모델링을 통해 도출한 위협을 도출한 후 각 위협이 미치는 영향 정도를 파악하기 위하여 우선순위를 지정해야 한다. 제품에 대한 모든 위협을 수용하고 피해를 관리하기 위해서는 비용 및 시간에 대한 제한이 있기 때문에 우선순위를 지정하고 위험도가 높은 부분에 대해 집중 관리가 필요하다.

본 논문에서는 자산을 기반으로 위험관리를 수행하는 Trike 기법을 선정하여 우선순위를 결정한다. 자산은 OEM 서버, 인포테인먼트로 분류하고, 위험도는 개인정보 영향도에 따라 4단계로 분류한다. None 단계는 해당 위협이 발생하더라도 위험도가 매우 낮은 수준이며, Low 단계는 중요도가 낮은 정보 노출로 인해 경미한 위협이 발생하여 위험도가 낮은 수준이다. Medium 단계는 중요도가 높은 정보 노출 또는 서비스에 영향을 주는 위협이 발생하여 위험도가 높은 수준이며, High 단계는 중요도가 매우 높은 정보 노출 또는 서비스 제어 권한에 대한 위협이 발생하여 위험도가 매우 높은 수준을 의미한다. Trike 분석을 통해 각 자산에서 제어 권한을 탈취당한 경우

Table 9. Risk level

Level	Risk	Example
None	Not relevant	Expose very low-priority information: dimmy information, etc.
Low	Minor risks	Low-Critical Information Disclosure: Patch update date, version, etc.
Medium	Significant risk	High-priority information exposure or threats affecting services: Logs, patch file exposure and modulation, etc.
High	Critical risk	Very High-priority information exposure or privilege acquisition: Encryption key, root acquisition, etc.

Table 10. Trike for Infotainment

Create Read Update Delete		Risk											
		Spoofing		Tampering		Repudiation		Information Disclosure		Denial of Service		Elevation of Privilege	
Asset	OEM Server		Low		Med				High		Med	High	High
		Low		Med				High				High	
	Infotainment		Med		Low				Med		Med	High	High
		Med		Low		Low						High	

가장 높은 위험도가 발생하며 분석 결과는 Table 10 과 같다.

3.7 Checklist

인포테인먼트의 안전한 업데이트를 위하여 DFD, STRIDE, Attack Library, Attack Tree, Risk Management를 기반으로 체크리스트를 도출하였다. 체크리스트는 인포테인먼트, 개발자, 업데이트 서버/배포 서버, 클라이언트, 애플리케이션, 네트워크로 총 6개의 분야로 분류된다.

Table 11. Checklist for Infotainment

Category	Checklist	
Infotainment	C1	Check download through internet
	C2	Check acquisition by packet sniffing when updating firmware
	C3	Check whether the firmware can be extracted through the boot loader
	C4	Check for existence of firmware verification routine
	C5	Check physical removal of JTAG ports
	C6	Check logical removal of JTAG ports
	C7	Check physical removal of UART ports
	C8	Check logical removal of UART ports
	C9	Check for infotainment important information (patch, version, status, etc.) acquisition
	C10	Check Flash memory data encryption
Infotainment	C11	Check modulation by overwriting the memory
Developer	C12	Check digital signature certificate management
Update server/ Deployment server	C13	Check identification and authentication features
	C14	Check account information for identification and authentication(Check whether information is leaked to ID / PW through packet dump at login)
	C15	Check replay attack during identification and authentication (Timestamp, Sequence, etc.)
	C16	Check the number of login attempt limit
	C17	Check secure password generation mechanism for identification and authentication(English capital letters, lower case letters, numbers, special characters, etc.)
	C18	Check password masking (*****) processing for identification and authentication
	C19	Check generate audit record for identification and authentication failure
	C20	Check trace for identification and authentication failure
	C21	Check the default password force change
	C22	Check manage critical file storage and configuration files
	C23	Check the session lock and session termination, when

Category	Checklist	
		there is no operation
	C24	Check the session management when accessing the same account or same authority
	C25	Check the system shell supported(SSH, Telnet, etc.)
	C26	Check information acquisition via system shell
	C27	Check the weak API usage (strcpy, strcat, system, etc.)
	C28	Check the incorrect API usage(out-of-bound, use-after-free)
	C29	Check the user's file access
	C30	Check the sensitive information in stored file
	C31	Check the privilege and access control for directory and file
	C32	Check whether audit records are accessible only by authorized administrators
	C33	Check the countermeasure for audit threshold
	C34	Check the countermeasure for audit storage full
Client	C35	Check the rollback to modification version or downgrade
	C36	Check the recovery after install update patch
	C37	Check the agent neutralization
	C38	Check the self-integrity check during booting and periodically
	C39	Check the data acquisition in local storage
	C40	Check whether important information is stored in a file(password, key, etc.)
	C41	Checking if ports are open via port scanning
	C42	Check that there are no shared folders Check that there is no shared path
	C43	Check the media automatic run(USB, CD-ROM, etc.)

Category	Checklist	
	C44	Check to keep the latest version of software in client
	C45	Check the vaccine installation, and keep the latest version of vaccine
Application	C46	Check identification and authentication
	C47	Check account information for identification and authentication(Check whether information is leaked to ID / PW through packet dump at login)
	C48	Check replay attack during identification and authentication (Timestamp, Sequence, etc.)
	C49	Check the number of login attempt limit
	C50	Check secure password generation mechanism for identification and authentication(English capital letters, lower case letters, numbers, special characters, etc.)
	C51	Check password masking (*****) processing for identification and authentication
	C52	Check generate audit record for identification and authentication failure
	C53	Check trace for identification and authentication failure
	C54	Check the default password force change
	C55	Check the session lock and session termination, when there is no operation
	C56	Check the session management when accessing the same account or same authority
	C57	Check session encryption
	C58	Check data acquisition in local storage
	C59	Check data acquisition in memory
	C60	Check the existence of the apk integrity tamper

Category	Checklist	
		detection routine
	C61	Check the safety of apk obfuscation and obfuscation
	C62	Check encryption key via apk decompile
	C63	Check for existence of important information such as account information, system information, personal information in app source code
	C64	Check requesting unnecessary permissions for app operation
	C65	Check whether system information is exposed through deliberate error induction
	C66	Check for the latest SSL / TLS
	C67	Check for the existence of routing check code
Network	C68	Check replay attack during identification and authentication (Timestamp, Sequence, etc.)
	C69	Check Transmission data encryption
	C70	Check Transmission data integrity
	C71	Check for trust after sending DoS packet (SYN Flooding, Smurf attack, Ping of death, Land attack, Teardrop attack)
	C72	Attack packet detection and flow control

IV. 보안요구사항 도출

인포테인먼트의 소프트웨어 및 펌웨어를 안전하게 업데이트 하기 위해서 위협모델링을 수행하고, 도출된 체크리스트와 매핑되도록 보안기능요구사항을 도출하였다. 또한 미국 NHTSA(미국 도로교통안전국)에서 2017년 NPRM(Notive Proposed Rulmaking)이라는 법 제정을 위한 요구사항에 H/W Security가 FIPS-140 Level 3를 의무적으로 준수해야 한다고 발표했다[30]. 이에 따라 해당 수준을 준수하는 보증요구사항을 도출하였다.

4.1 보안기능요구사항

정보보호제품 보안기능요구사항은 공통평가기준에 의거한다[31]. 위협모델링을 통해 도출된 체크리스트를 기반으로 공통평가기준 2부와 매핑된 SFR(Security Functional Requirement)을 통해 인포테인먼트에서 안전하게 업데이트하기 위한 기능요구사항을 도출하였다.

Table 12. Security Requirements

Class	Component	
FAU	FAU_GEN.1	Audit data generation
	FAU_GEN.2	User identity association
	FAU_STG.1	Protected audit trail storage
	FAU_STG.3	Action in case of possible audit data loss
	FAU_STG.4	Prevention of audit data loss
FCS	FCS_CKM.1	Cryptographic key generation
	FCS_CKM.2	Cryptographic key distribution
	FCS_CKM.3	Cryptographic key access
	FCS_CKM.4	Cryptographic key destruction
	FCS_COP.1	Cryptographic operation
FDP	FDP_ACC.1	Subset access control
	FDP_ACF.1	Security attribute based access control
	FDP_IFC.1	Subset information flow control
	FDP_IFF.1	Simple security attributes
	FDP_ETC.1	Export of user data without security attributes
FIA	FIA_AFL.1	Authentication failure handling
	FIA_UAU.1	Timing of authentication
	FIA_UAU.4	Single-use authentication mechanisms
	FIA_UAU.7	Protected authentication feedback
	FIA_UID.1	Timing of identification
	FIA_SOS.1	Verification of secrets
FMT	FMT_MOF.1	Management of security functions behaviour
	FMT_MSA.1	Management of security attributes

	FMT_MSA.2	Secure security attributes
	FMT_REV.1	Revocation
	FMT_SMR.1	Security roles
	FMT_SAE.1	Time-limited authorisation
FPT	FPT_FLS.1	Failure with preservation of secure state
	FPT_PHP.1	Passive detection of physical attack
	FPT_RCV.1	Manual recovery
	FPT_RPL.1	Replay detection
	FPT_TST.1	TSF testing
FTA	FTA_MCS.1	Basic limitation on multiple concurrent sessions
	FTA_MCS.2	Per user attribute limitation on multiple concurrent sessions
	FTA_SSL.1	TSF-initiated session locking
	FTA_SSL.3	TSF-initiated termination
FTP	FTP_ITC.1	Inter-TSF trusted channel

4.2 보증요구사항

차량 환경은 H/W Security를 고려해야 하는 환경으로 FIPS-140 Level 3를 의무적으로 준수하기 위한 보증 요구사항이 필요하다. FIPS-140 Level 1은 가장 낮은 수준의 보안수준으로 암호화를 제공하는 수준이며, Level 2는 인증되지 않은 물리적 접근을 방지하는 보안 수준이다. Level 3는 인증되지 않은 물리적 접근이 발생할 경우 이를 탐지하고 대응할 수 있는 보안 수준이며, Level 4는 최고 수준으로 인가되지 않은 모든 접근 시도를 탐지하고 암호화에 대한 완벽한 보안 기능을 제공하는 수준이다. 이 중 차량 관련 기준은 FIPS-140 Level 3를 준수해야 하며, 이 등급은 보안 기능 중 신뢰된 경로(Trusted path, FTP_TRP.1)를 필수적으로 준수하고 보증수준은 EAL3 이상 수준이다. EAL3 수준에서 추가적인 보증요구사항 중 보안 정책 모델(Security Policy Model, ADV_SPM.1)을 검증해야하는데 이 수준은 EAL6 이상 수준이기 때문에 이를 기반으로 보증요구사항을 도출하였다.

Table 13. Assurance Requirements

Class	Component	
ASE	ASE_CCL.1	Conformance claims
	ASE_ECD.1	Extended components definition
	ASE_INT.1	ST introduction
	ASE_OBJ.2	Security objectives
	ASE_REQ.2	Derived security requirements
	ASE_SPD.1	Security problem definition
	ASE_TSS.1	TOE summary specification
ADV	ADV_ARC.1	Security architecture description
	ADV_FSP.3	Functional specification with complete summary
	ADV_SPM.1	Formal TOE security policy model
	ADV_TDS.2	Architectural design
AGD	AGD_OPE.1	Operational user guidance
	AGD_PRE.1	Preparative procedures
ALC	ALC_CMC.3	Authorisation controls
	ALC_CMS.3	Implementation representation CM coverage
	ALC_DEL.1	Delivery procedures
	ALC_DVS.1	Identification of security measures
	ALC_LCD.1	Developer defined life-cycle model
ATE	ATE_COV.2	Analysis of coverage
	ATE_DPT.1	Testing: basic design
	ATE_FUN.1	Functional testing
	ATE_IND.2	Independent testing - sample
AVA	AVA_VAN.2	Vulnerability analysis

V. 결론 및 향후 연구 방향

무인 자동차, 자율 주행 등 차량 패러다임이 변화함에 따라 차량 내부에서 ECU등을 관리하는 인포테인먼트에 대한 중요성이 점차 증대되고 있다. 인포테인먼트에서는 네비게이션 기능뿐만 아니라 외부 네트워크와 연결되어 여러 가지 기능을 제공하고 있다. 하지만 외부 네트워크와 연결되어 있어 공격자가 접근할 수 있는 경로가 되고 있으며 실제 해킹 사례가 보고되고 있다.

특히 인포테인먼트는 다양한 소프트웨어가 설치되어 있어 패치 발생 시 안전하게 업데이트되어야 한다. 만약 악성 소프트웨어 설치 시 공격자가 차량에 대한 제어권을 획득할 수 있으며 이로 인해 차량 운전자의 생명에 영향을 끼칠 수 있다.

이러한 문제를 해결하기 위하여 본 논문은 STRIDE 위협모델링 기법을 통해 인포테인먼트의 소프트웨어 업데이트 환경에서 발생할 수 있는 위협을 분석하고, Trike 기법을 이용하여 위협의 우선순위 지정하여 위험도를 분석하였다. 이를 기반으로 해당 환경에서 요구되는 보안기능요구사항 및 보증요구사항을 체계적인 방법으로 도출하였으며, 이를 적용할 경우 보안성이 향상될 수 있을 것으로 기대된다.

앞으로도 차량 패러다임은 지속적으로 변화할 것이며, 변화하는 환경에도 적용할 수 있는 보안성 평가기준을 만들고 효율적으로 적용할 수 있는 방안이 필요할 것으로 사료된다.

References

〔1〕 Craig Smith, "THE CAR HACKER'S HANDBOOK," http://opengarages.org/handbook/ebook/, Jan. 2016.

〔2〕 Paul Ammann, Jeff Offut, "INTRODUCTION TO SOFTWARE TESTING Edition 2," https://cs.gmu.edu/~offutt/softwaretest/, Dec. 2016.

〔3〕 Charlie Miller, Chris Valasek, "Remote Exploitation of an Unaltered Passenger Vehicle", Black Hat USA 2015, Aug. 2015.

〔4〕 Keen Security Lab of Tencent, "FREE-FALL: TESLA HACKING 2016", Black Hat USA 2016, Aug. 2016.

〔5〕 GENIVI Alliance, https://www.genivi.org/, Mar. 2019.

〔6〕 Martin Klimke, Klaus Scheibert, Axel Freiwald, Björn Steurich, "Secure and seamless integration of Software Over The Air (SOTA) update in modern car board net architectures," ESCAR Europe 2015, Nov. 2015.

〔7〕 IHS(Information Handling Services), "Over-the-air Software Updates to Create Boon for Automotive Market," Sep. 2015.

〔8〕 Marco Steger, Carlo Boano, Michael Karner, Joachim Hillebrand, Werner Rom, Kay Römer, "SecUp: Secure and Efficient Wireless Software Updates for Vehicles," 2016 Euromicro Conference on Digital System Design, Aug. 2016.

〔9〕 Kirill Nikitin, Eleftherios Kokoris-Kogias, Philipp Jovanovic, Linus Gasser, Nicolas Gailly, Ismail Khoffi, Justin Cappos, Bryan Ford, "Proactive Software-Update Transparency via Collectively Signed Skipchains and Verified Builds," USENIX Security 2017, Aug. 2017.

〔10〕 Marco Steger, Ali Dorri, Salil S. Kanhere, Kay Römer, "Secure Wireless Automotive Software Updates using Blockchains," Advanced Microsystems for Automotive Applications 2017, pp 137-149, Jan. 2018.

〔11〕 Adam Shostack, "Threat Modeling: Designing for Security," https://adam.shostack.org/blog/category/threat-modeling/, Jun. 2019.

〔12〕 DistriNet Research Group, "LINDDUN: Privacy Threat Modeling," https://linddun.org/, Jun. 2019.

〔13〕 Trike, http://www.octotrike.org/, Jun. 2019.

〔14〕 GENIVI Infotainment Architecture, https://at.projects.genivi.org/wiki/display/GRK/2_Reference+Architecture+and+Compliance+Specification, Jun. 2018.

〔15〕 MITRE CVE, https://cve.mitre.org/, Jun. 2019.

〔16〕 MITRE CWE, https://cwe.mitre.org/, Jun. 2019.

〔17〕 MITRE CAPEC, https://capec.mitre.org/, Jun. 2019.

[18] OWASP, https://www.owasp.org/index.php/OWASP_Embedded_Application_Security, Jun. 2019.
[19] Alex Omar, "The Car Hacker's Handbook A Guide for the Penetration Tester," Feb. 2016.
[20] ITU-T, "Secure software update capability for intelligent transportation system communication devices," Mar. 2017.
[21] Sen Nie, Ling Liu, Yuefeng Du, Wenkai Zhang, "OVER-THE-AIR: HOW WE REMOTELY COMPROMISED THE GATEWAY, BCM, AND AUTOPILOT ECUS OF TESLA CARS", Black Hat 2017, Aug. 2017.
[22] Bjoern M. Luettmann, Adam C. Bender, "Man-in-the-Middle Attacks on Auto-Updating Software", Bell Labs Technical Journal, pp 131-138, May. 2007.
[23] Ang Cui, Michael Costello, Salvatore J. Stolfo, "When Firmware Modifications Attack: A Case Study of Embedded Exploitation," NDSS Symposium 2013, Apr. 2013.
[24] K. Chen, "Reversing and Exploiting an Apple Firmware Update," Black Hat USA 2009, Jul. 2009.
[25] New York University (Laboratory of Secure Systems), "The Update Framework," https://theupdateframework.github.io/, Jun. 2019.
[26] WONDER HOWTO, "How to Hijack Software Updates to Install a Rootkit for Backdoor Access," https://null-byte.wonderhowto.com/how-to/hack-like-pro-hijack-software-updates-install-rootkit-for-backdoor-access-0149225/, , Jun. 2019.
[27] Institute for Defence Studies and Anlayses, "THE PETYA CYBER ATTACK," http://cert-mu.govmu.org/, Jun. 2019.
[28] CrySyS Lab(Laboratory of Cryptography and System Security), "sKyWIper (a.k.a. Flame a.k.a. Flamer): A complex malware for targeted attacks," May. 2012.
[29] Bruce Schneier, "Attack Tree," Dr. Dobb's journal, Aug. 1999.
[30] NIST, "FIPS 140-2 Level 3 Security Policy", https://csrc.nist.gov/CSRC/media/projects/cryptographic-module-validation-program/documents/security-policies/140sp1275.pdf, Mar. 2014.
[31] Common Criteria, "CC v3.1 Release 5," https://www.commoncriteriaportal.org/cc/, Jun. 2019.

〈저자소개〉

강 수 영 (Soo-Young Kang) 학생회원
2006년 2월: 순천향대학교 컴퓨터공학부 공학사
2008년 2월: 순천향대학교 컴퓨터공학부 공학석사
2008년 5월~2010년 10월: 한국인터넷진흥원(KISA) 연구원
2010년 10월~2014년 10월: 안랩(Ahnlab) 주임연구원
2013년 3월~현재: 고려대학교 정보보호대학원 박사과정
〈관심분야〉 보안성 평가/인증, 위협 모델링, 소프트웨어 보안

김 승 주 (Seungjoo Kim) 종신회원
1994년~1999년: 성균관대학교 정보공학과(학사, 석사, 박사)
1998년~2004년: 한국인터넷진흥원(KISA) 팀장
2004년~2011년: 성균관대학교 정보통신공학부 부교수
2011년~현재: 고려대학교 사이버국방학과/정보보호대학원 정교수
2017년~현재: 고려대학교 사이버무기시험평가연구센터(CW-TEC) 부센터장
2004년~현재: 한국정보보호학회 이사
2007년: 국가정보원장 국가사이버안전업무 유공자 표창
2010년: 방송통신위원회 정보통신망 침해사고 민관합동조사단 위원
2011년~현재: (사)화이트해커연합 HARU 및 국제해킹대회 SECUINSIDE 설립자 및 이사
2012년: 선관위 디도스 특별검사팀 자문위원
2014년~2015년: 육군사관학교 초빙교수
2014년~2016년: 다음카카오 프라이버시 정책 자문위원회 위원
2015년~현재: 방위사업청 방산기술보호 자문관
2016년~2018년: 개인정보분쟁조정위원회 위원
2016년~현재: 산업통상자원부 전략물자기술 자문위원
2016년~현재: 한국카카오뱅크 정보보호부문 자문교수
2017년~현재: 국방보안연구소 정보보호분야 자문위원
2017년~현재: 여신금융협회 신용카드 단말기 시험 인증위원회 위원
〈관심분야〉 보안공학 및 SDL, 위협 리스크 모델링, 보안성 평가/인증, 암호학, Usable Security