Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.6, Dec. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.6.1393

웹 크롤러를 이용한 자동 패치 정보 수집 시스템*

김 용 건,[1†] 나 사 랑,[1] 김 환 국,[2] 원 유 재[1‡]
[1]충남대학교, [2]한국인터넷진흥원

Automatic Patch Information Collection System Using Web Crawler*

Yonggun Kim,[1†] Sarang Na,[1] Hwankuk Kim,[2] Yoojae Won[1‡]
[1]Chungnam University, [2]Korea Internet & Security Agency

요 약

다양한 소프트웨어를 사용하는 기업은 보안 업체에서 제공하는 패치관리시스템을 사용하여 소프트웨어의 취약점을 일괄적으로 관리해서 보안 수준을 높인다. 시스템 관리자는 최신 소프트웨어 버전을 유지하기 위해 신규 패치 정보를 제공하는 벤더 사이트를 모니터링 하지만 패치를 제공하는 주기가 불규칙적이고 웹 페이지 구조가 다르기 때문에 패치 정보를 검색하고 수집하는데 많은 비용과 모니터링 시간이 소요된다. 이를 줄이기 위해 키워드나 웹 서비스를 기반으로 패치 정보 수집을 자동화하는 연구가 진행되었으나 벤더 사이트에서 패치 정보를 제공하는 구조가 규격화되어 있지 않기 때문에 특정 벤더 사이트에서만 적용 가능했다. 본 논문에서는 패치 정보를 제공하는 벤더 사이트 구조와 특징을 분석하고 패치 정보 수집에 소모되는 비용과 모니터링 시간을 줄이기 위해서 웹 크롤러를 이용해 패치 정보 수집을 자동화하는 시스템을 제안한다.

ABSTRACT

Companies that use a variety of software use patch management systems provided by security vendor to manage security vulnerabilities of software to improve security. System administrators monitor the vendor sites that provide new patch information to maintain the latest software versions, but it takes a lot of cost and monitoring time to find and collect patch information because the patch cycle is irregular and the structure of web page is different. In order to reduce this, studies to automate patch information collection based on keyword or web service have been conducted, but since the structure to provide patch information in vendor site is not standardized, it was applicable only to specific vendor site. In this paper, we propose a system that automates the collection of patch information by analyzing the structure and characteristics of the vendor site providing patch information and using web crawler to reduce the cost and monitoring time consumed in collecting patch information.

Keywords: Web Crawler, Patch Information, Collection, Regular Expression

I. 서 론

인터넷이 보편화되고 소프트웨어 시장의 규모가 커지면서 다양한 소프트웨어가 개발되고 많은 시스템에서 사용되고 있다. 이로 인해 소프트웨어가 상용화 된 이후에 설계상의 결함으로 취약점이 발견되면 시스템의 결함으로 이어져 공격자가 악용할 경우 금전적 손실, 개인 정보 노출 등 보안 사고가 발생하며

Received(11.08 2018), Accepted(11. 24. 2018)
* 이 논문은 2018년도 정부(과학기술정보통신부)의 재원으로 정보통신기술진흥센터의 지원을 받아 수행된 연구임 (No.2016-0-00193, IoT 보안 취약점 검색·공유 및 시험 기술 개발)
† 주저자, sasinsg625@cnu.ac.kr
‡ 교신저자, yjwon@cnu.ac.kr(Corresponding author)

이를 예방하기 위해 소프트웨어의 취약점을 수정한 패치가 필수적이다. 소프트웨어 벤더는 사용자를 보호하기 위해 벤더 사이트를 통해 패치 정보를 제공한다. 기업에서 사용하는 시스템은 다양한 소프트웨어로 구성되어 있어서 취약점이 발견된 소프트웨어를 패치하기 위해서 다른 소프트웨어들과의 호환성과 패치가 된 이후의 안전성을 고려해야 한다. 하지만 패치가 나올 때마다 적용하기 어렵고 관리 비용이 발생하기 때문에 보안 업체에서 제공하는 패치관리시스템을 사용하여 소프트웨어의 취약점에 대한 패치를 일괄적으로 관리하고 있다[1].

시스템 관리자는 최신 소프트웨어 버전을 유지하기 위해 수시로 벤더 사이트에서 신규 패치 정보를 검색해서 수집하면 데이터베이스에 저장하고 사전 검증을 통해 안전성이 확보되면 클라이언트로 배포하여 패치를 진행한다[2]. 하지만 벤더 사이트에서 소프트웨어의 취약점에 대한 패치를 제공하는 시기가 불규칙적이다. 또한 소프트웨어 벤더마다 웹 페이지 구조가 다르기 때문에 신규 패치 정보를 수집하는데 많은 비용과 모니터링 시간이 소요된다.

본 논문에서는 패치 정보를 제공하는 벤더 사이트 구조와 특징을 분석하고 패치 정보 수집에 소모되는 비용과 모니터링 시간을 줄이기 위해서 웹 크롤러를 이용해 패치 정보 수집을 자동화하는 시스템을 제안한다.

II. 관련 연구

2.1 키워드를 이용한 패치 정보 수집

Fig.1.은 키워드를 이용한 패치 정보 수집 과정을 보여준다. 소프트웨어의 취약점에 대한 신규 패치 정보를 발표하는 벤더 사이트마다 URL(Uniform Resource Locator)과 웹 페이지 구조가 다르지만 패치 정보를 제공하는 공통적인 특징으로 사용되는 주요 키워드(fixed, patches, solution, remediation)를 이용하면 웹 페이지 구조가 달라도 패치 정보 수집이 가능하다[3]. 하지만 다른 벤더 사이트에서 동일한 키워드를 사용하지 않는다면 패치 정보가 제공되고 있어도 식별되지 않아서 다양한 벤더 사이트를 대상으로 키워드로만 수집 가능한 것은 한정적이다.

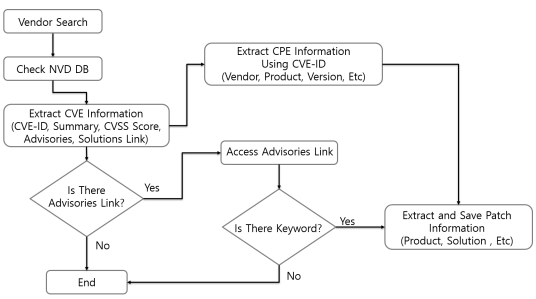


Fig. 1. Patch information collection using keyword

2.2 보안패치 자동분배를 위한 패치 DB 자동구성 방안

Fig.2.는 보안패치를 위한 DB(DataBase)를 구성하는 방법을 보여준다. 소프트웨어 벤더에서 제공하는 패치 정보를 자동으로 검색하고 수집해서 데이터베이스에 저장하기 위해 벤더 사이트에서 제공하는 웹 서비스에 따라 HTTP(HyperText Transfer Protocol)와 FTP(File Transfer Protocol)접근으로 구분하였다. HTTP로 접근이 가능한 벤더 사이트는 해당 웹 페이지 문서를 파싱하여 게시 글에서 링크를 추출해서 파일의 확장자가 포함되면 패치로 수집하였고 FTP로 접근이 가능한 경우에는 디렉토리 경로와 패치 파일명으로 검색하여 수집한다[4]. 이 연구는 벤더 사이트에서 웹 서비스에 따라 패치 정보를 다른 방법으로 검색해서 수집을 자동화하는 방법을 제안했다. 하지만 링크에 파일의 확장자가 있거나 디렉토리 경로를 제공하는 경우에만 수집이 가능한 방법이다.

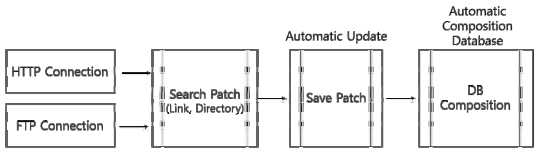


Fig. 2. Automatic composition database for security patch

III. 소프트웨어 벤더 사이트 분석

3.1 소프트웨어 벤더 사이트 구조

소프트웨어 벤더에서 패치 정보를 제공하는 공통점을 찾기 위해 사이트 구조와 특징을 분석한다.

Fig.3.은 벤더 사이트에서 소프트웨어의 취약점에

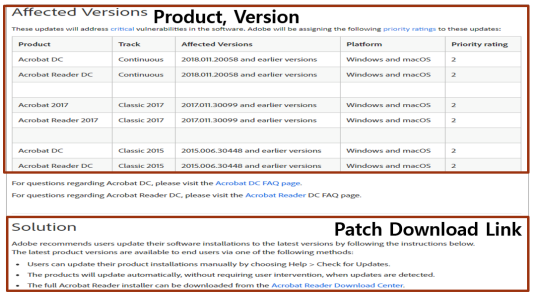
Fig. 3. Structure of vendor site

대한 패치 정보를 제공하는 구조를 보여준다. 벤더 사이트의 메인 웹 페이지에는 다양한 카테고리를 제공하고 있으며 그 중에서 보안 카테고리를 통해 들어가면 패치 정보를 확인할 수 있는 웹 페이지가 있다.

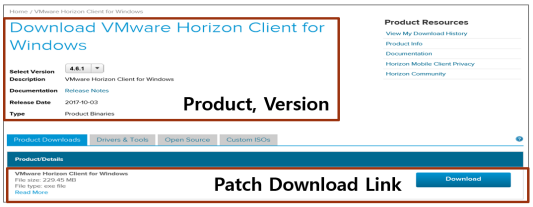
3.2 패치 정보를 제공하는 웹 페이지

Fig.4.는 벤더 사이트에서 패치 정보를 제공하는 웹 페이지를 보여준다. 웹 페이지에서 소프트웨어의 취약점 정보를 공지하면서 취약점에 영향 받은 소프트웨어 제품명, 버전과 소프트웨어의 취약점이 발생한 부분을 수정한 패치 파일을 다운 받을 수 있는 링크를 제공하고 있는데, 소프트웨어 벤더 마다 웹 페이지 구조가 다르다. Adobe 벤더는 제품과 버전 별로 한 단락이나 테이블 구조를 통해 패치 파일을 다운받을 수 있는 링크를 제공하는 반면, VMware는 소프트웨어 버전을 선택하는 자바스크립트 이벤트 동작에 따라 제공하는 링크가 다르다.

Table 1.은 소프트웨어 벤더 10개를 대상으로



Adobe



VMware

Fig. 4. Web page providing patch information

Table 1. Structure of web page

	Vendor	Structure Providing Patches
1	Google	Table
2	IBM	Table, Sentence
3	Adobe	Table, Javascript Event
4	Oracle	Table
5	Microsoft	Table, Sentence
6	Cisco	Sentence
7	Vmware	Sentences, Javascript Event
8	McAfee	Table, Javascript Event
9	Apple	Sentence
10	Redhat	Sentence

웹 페이지에서 패치 정보를 제공하는 구조를 조사한 결과이다. 대부분의 경우 패치 정보를 한 단락이나 테이블 구조를 이용해 제공하지만 일부 소프트웨어 벤더는 웹 페이지에서 소프트웨어 제품명과 버전을 선택하는 자바스크립트 이벤트를 통해 패치 정보를 제공하고 있다.

IV. 웹 크롤러를 이용한 패치 정보 수집 시스템

4.1 패치 정보 수집 시스템

해당 장에서는 웹 크롤러를 이용한 패치 정보 수집 시스템을 제안한다. 제안하는 시스템과 관련 연구들과의 비교 분석은 Table 2.와 같다.

패치 정보 수집 시스템은 관련 연구의 한계점인 특정 벤더 사이트에서만 적용 할 수 있다는 문제점을 개선하기 위해 정규표현식과 XPath를 사용하였고 시스템 관리자가 벤더 사이트를 모니터링 하는 시간과 비용을 줄이기 위해 이메일 전송 기능과 스케줄링 기능을 추가하였다. 또한 벤더 사이트에서 패치 정보를 제공하는 구조가 변경되어도 시스템 관리자가 패치 정보를 수집하는 정책을 생성 할 수 있어 확장성을 제공한다.

Fig.5.은 패치 수집 시스템 구조를 보여준다. 패

Table 2. Related research Comparison

	Existing	Proposed
Technology	Keyword, Extension, Directory Path	Regular expression, XPath
Scalability	Impossibility	Possibility
E-mail	Impossibility	Possibility
Scheduling	Impossibility	Possibility

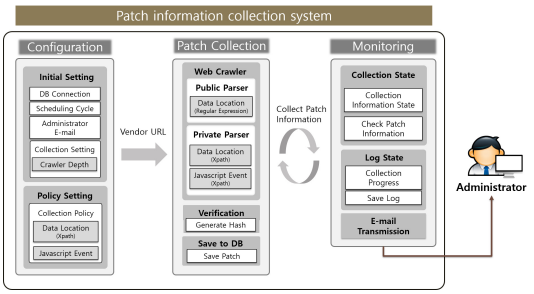


Fig. 5. Structure of patch information collection system

치 정보 수집 시스템은 환경 설정, 패치 정보 수집, 모니터링 및 알림 기능들로 구성된다. 환경 설정은 패치 정보 수집하기 위해 필요한 DB 설정, 스케줄링 설정, 이메일 설정, 수집 설정이 있다. 또한 웹 페이지 내의 데이터 위치와 자바스크립트 동작 순서를 고려한 수집 정책 설정이 있다. 패치 정보 수집은 웹 크롤러를 이용해 벤더 사이트 내의 웹 페이지를 수집하고 범용 파서와 전용 파서를 사용하여 패치 정보를 수집한다. 모니터링 및 알림은 패치 정보가 수집되는 과정을 보여주며 신규 패치 정보가 수집되면 관리자에게 이메일로 전송하는 기능을 포함한다.

Fig. 6.는 패치 수집 시스템이 동작하는 순서도를 보여준다. 시스템이 동작하기 위해서는 사전에 패치

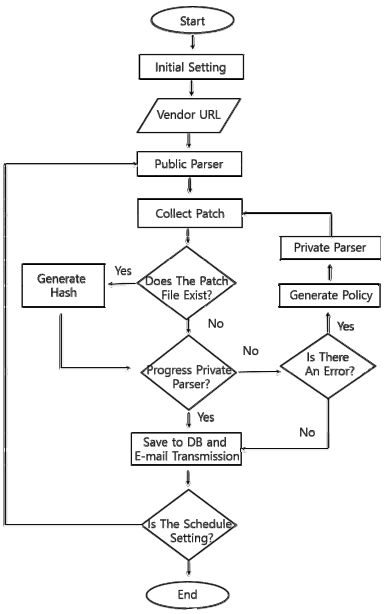


Fig. 6. Flowchart of patch information collection system

수집 대상인 벤더 URL을 가져오고 신규 패치가 수집되면 저장되는 데이터베이스 정보, 패치 수집을 주기적으로 수행하는 스케줄링, 웹 검색 효율성을 높이기 위해 웹 크롤러의 Depth를 설정하고 패치 수집 결과를 전송 받을 관리자의 이메일 계정을 등록하는 초기 설정이 필요하다. 설정이 완료되면 데이터베이스에 저장된 벤더 URL을 가져와 웹 크롤러를 진행하여 웹 페이지를 수집하고 1차적으로 임의의 웹 페이지에 적용 가능한 범용 파서를 통해 패치 정보를 수집한다. 이 때 웹 페이지에서 패치 정보를 수집 못한 오류가 발생할 경우 전용 파서를 통해 2차 수집을 진행한다. 이 과정이 완료되면 수집한 결과를 데이터베이스에 저장하면서 이메일로 관리자에게 전송하고 스케줄링 주기 여부를 판단하여 지속적인 모니터링을 수행한다.

4.2 웹 페이지 수집

4.2.1 웹 크롤러

벤더 사이트 내에서 보안 게시판을 검색하고 패치 정보가 제공되는 웹 페이지 수집을 자동화하여 모니터링 시간을 줄이기 위해 웹 크롤러를 사용한다.

Fig.7.는 웹 크롤러 프로세스를 보여준다. 웹 크롤러는 벤더 사이트 내의 링크를 통해 웹 페이지의 HTML(HyperText Markup Language)태그, 문서 내용, URL 정보를 자동으로 수집하는 기술[5,6]로 다량의 정보 수집에 적합하기 때문에 패치 수집 시스템에서 웹 페이지를 수집하는데 사용한다.

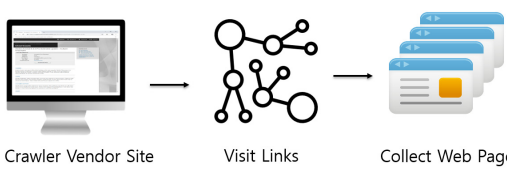


Fig. 7. Web crawler process

4.2.2 정규표현식을 이용한 패치 정보 식별

패치 정보를 자동 수집하기 위해서는 웹 크롤러를 통해 수집한 많은 웹 페이지 중에서 패치 정보가 제공되는 웹 페이지를 구분해야 한다.

Fig.8.은 소프트웨어 버전 표기법을 보여준다. 소프트웨어 벤더에서 제공하는 소프트웨어 버전은 소

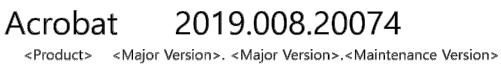


Fig. 8. Software version

프트웨어의 특정 상태를 표기하는데 사용하며 취약점을 수정하면 버전 번호를 증가시켜 관리한다. 소프트웨어 벤더마다 버전 번호에 의미를 부여하거나 작성 방법을 가지고 있어 국제적으로 통일된 규칙은 존재하지 않는다. 하지만 대부분의 버전 번호는 점(.)으로 구분되며 순서대로 주 버전과 부 버전 등으로 나눈다는 규칙성을 가진다.

벤더 사이트에서 소프트웨어 제품명과 버전 별로 패치 정보를 제공한다는 특징을 가지고 있기 때문에 일정한 규칙을 가진 구조를 표현하는데 사용되는 정규표현식[7]으로 웹 페이지에서 버전을 식별하여 패치 정보 제공 여부를 확인한다.

4.3 패치 수집 방법

4.3.1 범용 파서

Fig.9.은 범용 파서 프로세스를 보여준다. 범용 파서는 임의의 웹 페이지에 적용할 수 있는 방법으로 웹 페이지에서 패치 정보를 한 단락이나 테이블 형식 구조로 제공한다면 정규표현식으로 소프트웨어 버전이 식별된 데이터 위치를 찾고 주변 단락이나 테이블 구조 내에서 패치 정보를 수집한다.

하지만 웹 페이지에서 소프트웨어 제품명과 버전을 선택하는 자바스크립트 이벤트 동작이 있거나 버전이 식별된 데이터 위치 주변에 패치 파일을 다운로드 할 수 있는 링크가 없다면 정보의 정확도와 수집률이 떨어진다. 이를 보완하기 위해서 웹 페이지에서 패치 정보가 제공되는 데이터 위치와 자바스크립트 이벤트 동작을 고려한 수집 방법을 사용한다.

Fig. 9. Public parser process

4.3.2 전용 파서

Fig.10.은 전용 파서 프로세스를 보여준다. 전용 파서는 시스템 관리자가 직접 웹 페이지에서 패치 정보가 제공되는 데이터 위치와 자바스크립트 이벤트 동작을 고려해서 생성한 수집 정책을 기반으로 패치 정보를 수집하는 방법이다. XPath(Xml Path language)는 웹 페이지 구조에서 지정한 정보를 찾기 위한 경로 표시법[8]으로 웹 페이지에서 패치 정보가 제공되는 위치와 자바스크립트 이벤트가 동작하는 위치를 찾을 수 있다. XPath로 수집 정책을 생성하면 웹 크롤러를 이용해 지정한 위치의 패치 정보를 가져오고 클릭 이벤트를 통해 자바스크립트 이벤트 동작을 수행할 수 있어서 패치 정보의 정확도와 수집률을 높일 수 있다.

Fig. 10. Private Parser Process

4.4 이메일 전송

소프트웨어 벤더에서 중요한 소프트웨어들에 대해서는 시스템 관리자에게 패치 정보를 알림해주는 서비스를 제공한다. 패치 정보 수집 시스템은 소프트웨어 벤더에서 서비스를 제공하지 않는 소프트웨어들의 패치 정보도 이메일 전송을 통해 알 수 있다.

Fig.11.은 이메일 전송 프로세스를 보여준다. 패치 정보 수집 시스템은 스케줄링 시간을 기준으로 주기적으로 벤더 사이트를 모니터링하고 수집한 패치 정보를 시스템 관리자에게 이메일로 전송한다.

Fig.12.는 VMware 벤더 사이트에서 패치 정보를 수집한 결과를 보여준다. 이처럼 패치 수집 시스템을 확인하지 않아도 신규 패치 정보가 수집되면 이

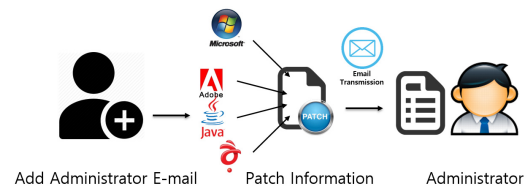


Fig. 11. Email transmission process

Fig. 12. Collect patch information

메일을 통해 수집 여부를 알 수 있어 업무의 효율성이 향상되고 모니터링 시간을 줄일 수 있다.

4. 결 론

본 논문에서는 벤더 사이트에서 패치 정보를 제공하는 구조와 특징을 분석한 결과를 통해 웹 페이지에서 패치 정보를 식별 할 수 있었다. 또한 XPath를 이용해 패치 정보의 수집률과 정확도를 높이면서 웹 크롤러를 이용해 패치 정보 수집을 자동화하는 시스템을 제안하였다. 이 시스템을 이용하면 소프트웨어 벤더에서 패치 정보를 제공하는 주기를 알지 못해도 벤더 사이트를 지속적으로 방문하면서 신규 패치 정보를 수집하고 이메일 전송을 통해 시스템 관리자가 수집 여부를 알 수 있어 모니터링 하는 시간을 줄일 수 있었다. 하지만 벤더 사이트 내의 웹 페이지가 방대하기 때문에 패치 정보를 식별하고 수집하는데 까지 많은 시간이 소모된다. 향후 패치 정보 수집에 대한 성능 비교와 수집 시간을 줄이기 위한 연구가 필요 하다.

References

[1] JunHee Kim and Yoojae Won, "Patch Integrity Verification Method Using Dual Electronic Signatures," Journal of Information Processing Systems, vol. 13, no. 6, pp. 1516-1529, Dec. 2017

[2] Inyong Lee, Suyoung Lee, Jaeik Cho and Jongsub Moon, "Effective Patch Database Composing For Multi-OS and S/W," Korea Information Science Society, 34(1D), pp. 100-103, Oct. 2007

[3] Jin-Ho Song, Yong-Gun Kim and Yoo-Jae Won, "Research for collecting efficient patch information," *Korea Information Science Society*, pp. 1164-1166, Dec. 2017

[4] Dong-Og Min, Tae-Shik Shon, Jung-Taek Seo, Won-Bon Koo, Jung-Ah Jang and Jong-Sub Moon, "Automatic Composition Database For Security Patch Auto-Distribution," Korea Information Science Society, 31(1A), pp.367-369, Apr. 2004

[5] Moon-Soo Chang and June-Young Jun, "A Method of Efficient Web Crawling Using URL Pattern Scripts," Journal of Korean Institute of Intelligent Systems, 17(6), pp. 849-854, Dec. 2007

[6] Seong-Chan Jo, Young-Duk Seo and Doo-Kwon Baik, "A study of improving URL duplication problem for distributed web crawling system," Korea Information Science Society, pp. 251-253, June. 2015

[7] Kwang-Man Ko and Hong-Jin Park, "Development of the Pattern Matching Engine using Regular Expression," The Korea Contents Society Association, 8(2), pp. 33-40, Feb. 2008

[8] Chee-Yong Chan, Pascal Felber, Minos Garofalakis and Rajeev Rastogi, "Efficient filtering of XML documents with XPath expressions," The International Journal on Very Large Data Bases, vol. 11, no. 4, pp. 354-379, Dec. 2002

〈저자소개〉

김 용 건 (Yonggun Kim) 학생회원
2017년 2월: 한남대학교 컴퓨터공학과 학사
2017년 8월~현재: 충남대학교 컴퓨터공학과 석사과정
〈관심분야〉 IoT 보안, 소프트웨어 보안, 블록체인 보안 등

나 사 랑(Sarang Na) 정회원
2011년 2월: 세종대학교 컴퓨터공학과 학사
2013년 8월: 세종대학교 컴퓨터공학과 석사
2013년 9월~현재: 연세대학교 정보대학원 박사과정
〈관심분야〉 IoT 보안, 취약점 분석, 텍스트 마이닝

김 환 국(Hwankuk Kim) 종신회원
2001년~2006년: 한국전자통신연구원 연구원
2017년 2월: 고려대학교 정보보호대학원 공학박사
2007년~현재: 한국인터넷진흥원 보안기술R&D2팀 팀장
〈관심분야〉 SW 취약점 분석, 네트워크 보안, IoT 보안

원 유 재 (Yoojae Won) 종신회원
1985년 2월: 충남대학교 계산통계학과 학사
1987년 2월: 충남대학교 계산통계학과 석사
1998년 2월: 충남대학교 컴퓨터공학과 박사
1987년 2월~2001년 2월: 한국전자통신연구원(ETRI) 팀장
2001년 3월~2004년 8월: 안랩유비웨어, 안철수연구소 CTO
2004년 9월~2014년 2월: 한국인터넷진흥원 인터넷침해대응센터 센터장
2014년 2월~현재: 충남대학교 컴퓨터공학과 교수
〈관심분야〉 사이버 침해대응, 시스템 및 네트워크 보안, IoT 보안 등