Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.28, NO.1, Feb. 2018 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2018.28.1.145

서버 SSL/TLS 취약점 자동 탐지를 위한 시스템 개발*

조 성 원,[1†] 최 현 상,[1] 허 규,[1] 조 상 현,[1‡] 김 영 갑[2]
[1]네이버, [2]세종대학교

A System for SSL/TLS Vulnerability Detection of Servers*

Sungwon Cho,[1†] Hyunsang Choi,[1] Gyu Heo,[1] Sanghyun Cho,[1‡] Young-Gab Kim[2]
[1]Naver Corporation, [2]Sejong University

요 약

SSL(Secure Socket Layer) 과 TLS(Transport Layer Security)는 네트워크 통신환경에서 주고받는 데이터를 보호하기 위해서 암호화를 해주는 통신 규약으로 널리 사용되고 있다. 그러나 이 통신 규약들의 설계 결함과 실제 구현에서의 오류로 인해서 보안 이슈가 끊이지 않고 있으며 특히 많은 서버들이 이를 이용하고 있기 때문에 보안취약점으로 인해 심각한 피해를 일으킬 수 있다. 본 논문에서는 서버의 SSL과 TLS 보안 취약점을 자동으로 빠르게 탐지하고 주기적으로 검사할 수 있는 시스템을 개발하였다. 본 연구에서 개발한 취약점 스캐닝 시스템은 기존의 취약점 검사 툴에 비하여 주요 SSL과 TLS 관련 취약점을 내부 네트워크에서 빠르게 검사하여 탐지하는 것이 가능하며 검사 스케줄링 및 시각화 기능을 제공한다. 개발한 취약점 스캐닝 시스템을 이용하여 네이버 망에서 실제 운용되고 있는 서버에서 SSL과 TLS 관련 취약점들을 탐지하여 그 결과를 분석하였다. 네이버 주요 서비스 및 관계사 도메인 213개에 대해 검사를 수행하여 각 도메인 당 평균 4.2개, 즉 총 816개의 취약점이 발견되었다. 7개 이상의 취약점이 발견된 도메인 21개 중 일반 사용자들이 직접 사용하는 도메인은 없었으나 46개의 도메인에서 2016년도 보안 취약점에 노출되어 있었다. 본 논문에서 개발한 취약점 검사 시스템을 이용해서 네이버 망의 서버에서 발견된 보안 취약점들을 빠르게 대응하여 패치를 수행할 수 있었다.

ABSTRACT

SSL (Secure Socket Layer) and TLS (Transport Layer Security) are widely used protocols for secure and encrypted communication over a computer network. However, there have been reported several security vulnerabilities of SSL/TLS over the years. The vulnerabilities can let an adversary carry out critical attacks on SSL/TLS enabled servers. In this paper, we have developed a system which can periodically scan SSL/TLS vulnerabilities on internal network servers and quickly detects, reports and visualizes the vulnerabilities. We have evaluated the system on working servers of Naver services and analyzed detected vulnerabilities. 816 vulnerabilities are found on 213 internal server domains (4.2 vulnerabilities on average) and most vulnerable servers are not opened to public. However, 46 server domains have old vulnerabilites which were found 2016. We could patch and response to SSL/TLS vulnerabilites of servers by leveraging the proposed system.

Keywords: Mobile advertisement, mobile ad injection, abusing

I. 서 론

웹 서버들은 사용자와의 인터넷 연결을 보호하기 위해 TLS(Transport Layer Security)와 SSL (Secure Socket Layer) 기술을 사용하고 있다. 이 기술들은 전송계층 종단간 보안과 데이터 무결성

Received(10. 27. 2017), Modified(01. 04. 2018), Accepted(01. 29. 2018)
* 본 논문은 NAVER 주식회사로부터 지원되었습니다.

† 주저자, amjo.nyang@gmail.com
‡ 교신저자, super.bg@navercorp.com(Corresponding author)

을 암호화 알고리즘을 통해 제공한다. 그러나 취약한 설정을 사용하거나 구현상의 문제가 있을 경우 연결을 보호할 수 없으며 서버의 리소스가 유출될 수도 있다. 예를 들어 Logjam attack[1]은 서버가 512bit Diffie-Hellman(DH) 키 교환 알고리즘을 사용할 수 있도록 설정한 경우에 발생하는 취약점을 이용한 공격이다. 현재의 컴퓨터 연산 속도는 512bit group의 discrete log를 몇 분 안에 구하기에 충분하다. 따라서 공격자는 중간자 공격을 통해 서버와 사용자가 512bit DH 알고리즘을 사용하도록 강제하여 송수신되는 모든 데이터를 복호화 할 수 있고, 변조 또한 가능하다.

몇 년 전에 이슈가 되었던 Heartbleed[2]는 오픈 소스 라이브러리인 OpenSSL의 취약점으로, 잘못된 프로토콜 구현이 원인이 되어 발생하는 취약점이다. SSL/TLS에는 상대와 연결이 계속 유지되는지 확인할 수 있도록 Heartbeat라는 프로토콜이 존재한다. Heartbeat는 어느 한 쪽이 메시지를 전송하면 상대방이 수신된 메시지를 그대로 다시 전송하는데 악의적으로 조작된 메시지를 받게 되면, 그 뒤의 메모리 내용을 상대방에게 전송하게 된다. 따라서 공격자는 Heartbeat 프로토콜이 허용하는 최대 메시지 길이인 64KB씩 서버 메모리의 데이터를 탈취할 수 있다. 앞에서 언급한 Logjam과 Heartbleed 외에도 많은 SSL/TLS 취약점이 발견되어 공개 되었으며 최근에도 지속적으로 보고가 되고 있다.

이러한 TLS 및 SSL의 취약점을 탐지하기 위해 취약점 검사 도구들이 지속적으로 개발되어 왔다. 그러나 많은 검사 도구들이 다양한 종류의 취약점을 검사하는 기능을 충분히 제공하지 않거나 외부 라이브러리에 의존성을 갖는다. 뿐만 아니라 주기적으로 취약점을 스캔하고 스캔 결과에 대한 관리 및 시각화 등의 기능들을 제공하고 있지 않다. 또한, 일부 툴은 검사의 기능은 뛰어나나 검사 API만을 제공하여 내부 네트워크의 검사 결과가 외부로 누출 될 수 있다는 단점을 갖고 있다.

본 연구에서는 기존 문제를 해결하고자 주기적으로 여러 서버의 SSL/TLS 취약점을 검사할 수 있는 SSL/TLS 스캐너 시스템을 개발하였고, 실제로 네이버(Naver)에서 운용중인 서버들을 대상으로 스캐닝을 수행하고 그 결과를 분석하였다. 특히, 본 연구에서 개발된 시스템은 보안상 중요한 취약점 항목과 관련 없는 부가적인 검사 항목이나 부수적인 기능을 빼고 외부 라이브러리나 구현체에 의존하지 않고 개발을 하여 검사 속도가 매우 빠르다. 뿐만 아니라 검사 스케쥴링 및 결과를 시각화 하는 기능을 제공하여 그 효용성을 극대화 하였다.

본 논문의 구성은 다음과 같다. Ⅱ장에서는 다른 취약점 검사 도구를 설명하고 각각을 본 연구에서 개발한 시스템과 비교하였다. Ⅲ장에서는 본 연구에서 개발한 SSL/TLS 취약점 검사 시스템에 대해서 상세히 설명한다. Ⅳ장에서는 네이버의 서버들에서 취약점 검사를 수행한 결과를 정리하고, Ⅴ장에서는 본 제안 시스템의 미비점 및 제한사항에 대해 언급하고, 마지막으로 Ⅵ장에서 본 연구의 결론에 대해 기술한다.

Ⅱ. 관련 연구

많은 SSL/TLS 취약점 검사 도구가 존재하지만, 본 논문에서는 Table 1과 같이 대표적인 3개 도구(즉, Qualys SSL Server Test, SSLyze, testssl)를 소개하고 본 연구에서 개발한 검사 도구와 비교하였다.

Qualys SSL Server Test[3]는 보안 업체인 Qualys Inc.가 무료로 제공하는 SSL/TLS 취약점 검사 도구이다. 웹 페이지 또는 API를 통해 검사할 서버의 도메인이나 IP 주소를 전송하면 보안 취약점이 존재하는지 여부와 함께 각종 TLS 설정 정보를 분석해서 알려준다. 또한, 자체적인 기준에 따라 A+부터 F까지 등급을 부여한다. SSLyze[4]는 Python과 nassl이라는 OpenSSL wrapper를 사용한 오픈 소스 검사 도구이다. Qualys 사의 검사 도구와는 다르게 CLI(command-line interface) 형태로만 제공되며, 직접 소스코드를 다운받아 실행할 수 있다. testssl.sh [5]는 단일 쉘 스크립트(shell script)로만 이루어진 오픈 소스 검사 도구이다. 사용자의 컴퓨터에 설치된 OpenSSL과 Linux socket을 사용한다.

Qualys SSL Server Test[3]는 웹 페이지를 통해 도메인이나 IP 주소를 입력 받아 1분 이내에 분석을 마치고, 보고서를 생성하여 보여준다. 먼저, 주체, 유효기간, 발급자, 서명 알고리즘, 인증서 해지 상태 등을 포함한 모든 인증서의 개별적인 정보와 인증서 체인을 표시해준다. 다음으로 서버에서 지원하는 모든 프로토콜과 암호화 알고리즘을 알려주고, 자주 사용되는 브라우저 및 봇에서 어떤 암호화 알고

Table 1. Comparison with Qualys, SSLyze and testssl (O: all supported, △: partially supported, X: not supported)

Analysis Tools	Qualys	SSLyze	testssl.sh	Proposed
Analysis Time	〈 60s	〈 5s	〈 30s	〈 15s
Supported Platform	Web	Win/Linux	Linux	Linux
Internal Network Check	X	O	O	O
Report Detail Check	O	X	△	O
External Library Dependancy	Unknown	O	O	X
Certificate Information	O	O	O	X
Vulnerable Algorithm Scan	O	O	△	O
Browser Handshake Simulation	O	X	O	X
DROWN Attack Test	O	X	O	O
Renegotiation Support Test	O	X	△	O
Compression, Fallback SCSV Test	O	O	O	O
BEAST, POODLE, Heartbleed Test	O	O	O	O
Ticketbleed Test	O	X	X	X
OpenSSL EarlyCCS Test	O	O	O	O
OpenSSL CVE-2016-2107 Test	O	X	X	O
Session Resumption Check	O	O	X	X
HSTS, HPKP Setting Check	O	X	O	O
Protocol Intolerance Test	O	X	X	X
(EC)DH Server Param Reuse Test	O	X	X	O
Supported EC Curve Test	O	X	X	X

리즘을 선택하여 연결이 성립되는지 보여준다. 다음으로 (1) Heartbleed 등 개별적인 취약점 존재 여부와 (2) 지원하는 EC(elliptic curve)알고리즘 등 SSL/TLS의 상세한 설정, (3) HSTS(HTTP strict transport security) 지원 여부 등 HTTPS 설정에 관련된 부가적인 정보를 표시한다. 따라서 Qualys사의 도구는 어떠한 준비 과정 없이 바로 사용할 수 있는 강력한 도구라는 큰 장점이 있으나, Qualys 사의 서버에서 검사하므로 인터넷에 연결되어있지 않은 서버는 검사할 수 없으며 검사 결과가 유출될 수 있다는 단점이 있다. 특히 사내 내부망의 서버에 대한 검사에는 데이터 유출의 위험이나 커스터마이징(customizing)이 불가능 하다는 점 때문에 검사를 수행하기 어렵거나 불가능할 수 있다.

SSLyze [4]는 Python의 패키지 관리자인 pip를 사용하여 쉽게 설치할 수 있는 CLI 도구이다. C로 개발된 nassl이라는 OpenSSL의 wrapper를 사용하여 분석한다. SSLyze는 오픈소스이며 특정 취약점을 분석하는 플러그인을 호출하는 방법으로 되어 있어 확장이 용이하다. 또한 5초 안에 분석이 완료되는 등 속도가 매우 빠르다. 결과는 텍스트로 출력되며, Qualys사의 도구와 마찬가지로 인증서 정보, 지원하는 프로토콜과 암호화 알고리즘, 그리고 몇 개의 취약점 검사 결과가 포함되어있다. 그러나 Qulays사의 도구에서는 제공하는 브라우저에 따른 연결 알고리즘 정보나 DROWN, Ticketbleed 등의 취약점 검사를 제공하지 않으며, HSTS 설정 등 부가적인 HTTPS 관련 설정을 알려주지 않는다는 단점이 있다.

testssl.sh[5]는 단일 쉘스크립트로 된 CLI 도구이며, wget이나 curl 등을 사용하여 직접 스크립트를 다운받아 사용할 수 있다. 검사는 30초 정도 소요되며 Qualys사의 검사 도구와 마찬가지로 인증서 정보, 지원 프로토콜과 알고리즘, 개별 취약점 존재 여부와 부가적인 HTTPS 관련 설정을 표시한다. 그러나 이 도구는 사용자의 컴퓨터에 설치된 OpenSSL과 Linux socket을 사용하여 검사를 수행하므로 Linux 외의 환경은 지원하지 않으며 사용자의 컴퓨터에 설치된 OpenSSL 버전에 따라 일부 검사가 불가능한 단점이 있다. 또한, Insecure client-initiated renegotiation 검사를 하지 않는 등 Qualys 사의 검사 도구보다 적은 항목에 대해 검사한다.

본 연구에서 제안하는 시스템도 Qualys와 비교해 볼 때 수행하지 않거나 지원하지 않는 부가 기능들이 있다. 그러나 이는 테스트 항목 관련 알려진 공격이 없어서 구체적인 관련 보안 취약점을 찾을 수 없는 항목(예를 들어, Supported EC Curve

Test, Protocol Intolerance Test)이나 주요 보안 취약점과 관련 없는 부가적 기능과 관련된 항목(예를 들어, Browser Handshake Simulation, Certificate Information)이거나 혹은 본 시스템 개발 이후에 알려진 보안 취약점으로 추후 지원 예정인 항목(Ticketbleed Test)이다. 본 시스템은 부가적인 기능이나 보안상 위협적이지 않은 부분을 제외한 주요 취약점에 대해서 빠르게 검사하여 내부 네트워크의 서버들을 안전하게 하고자 함이 목적이므로 시스템의 목적상 여러 항목들이 제외 되었다.

SSL/TLS 자동 검사와 관련하여 다양한 연구가 진행되었다. Brubaker[6] 등과 Chen[7] 등은 mutation 기반 차이점 테스팅 (differential testing) 기법을 이용해서 인증서 검증(certificate validation)의 문제점을 찾는 연구를 하였다. 그러나 그들의 경우에는 인증서 검증 체크를 위해서 라이브러리의 호스트 이름을 검증하는 기능이 꺼져 있었기 때문에 몇 가지의 취약점을 찾지 못하는 한계점을 갖고 있다.

Somorovsky [8]는 TLS 취약점을 시스템적 퍼징(fuzzing)을 통해 찾아낼 수 있는 TLS-Attacker라는 도구를 만들었다. TLS-Attacker는 상태기계(state machine) 기반의 퍼징 등을 통해 SSL/TLS 제로데이(zero-day) 취약점을 탐지하는 데에는 유리하나 현재 동작중인 서버에서 취약점을 빠르게 탐지하고 관리하는데에 적합하지는 않다.

Ruiter와 Poll[9]은 오토마타 러닝(automata learning) 알고리즘을 이용해서 TLS 프로토콜의 모델을 인지하고 수동으로 버그를 탐지하는 연구를 수행하였다. 그들의 연구는 TLS handshake 과정에서 교환되는 메시지의 차이점에서 TLS state machine을 유도하는 방식에 초점을 맞추고 있어 앞의 TLS-Attacker와 유사한 연구로 볼 수 있다.

Suphannee[10]등의 연구에서는 HVLearn이라는 블랙박스 테스팅(black-box testing) 프레임워크를 제안하여 SSL/TLS 호스트 이름의 검증을 분석하였다.

III. SSL/TLS 스캐너 시스템 개발

본 논문에서 제안한 SSL/TLS 스캐너의 구조는 Fig. 1과 같으며, 시스템의 주요 기능은 다음과 같다.

- 핵심 보안 취약점과 관련 없는 항목을 제거하고 부가적인 정보 제공 등의 기능을 구현하지 않았고, 또한 병렬처리로 검사를 수행하여 검사 속도가 매우 빠르다. 특히 OpenSSL 등 외부 라이브러리에 의존하지 않으며, Go[11]의 기본 TLS 라이브러리만 사용하여 어떤 환경에서나 검사할 수 있다.
- Web UI를 제공하여 검사할 서버를 쉽게 추가, 수정 및 삭제할 수 있으며, "매일 오전 10시 검사" 등의 주기적인 자동 검사를 수행하게 할 수 있다.
- 모든 검사 결과를 DB에 저장하며, Elastic Stack으로 쉽게 시각화가 가능하다.

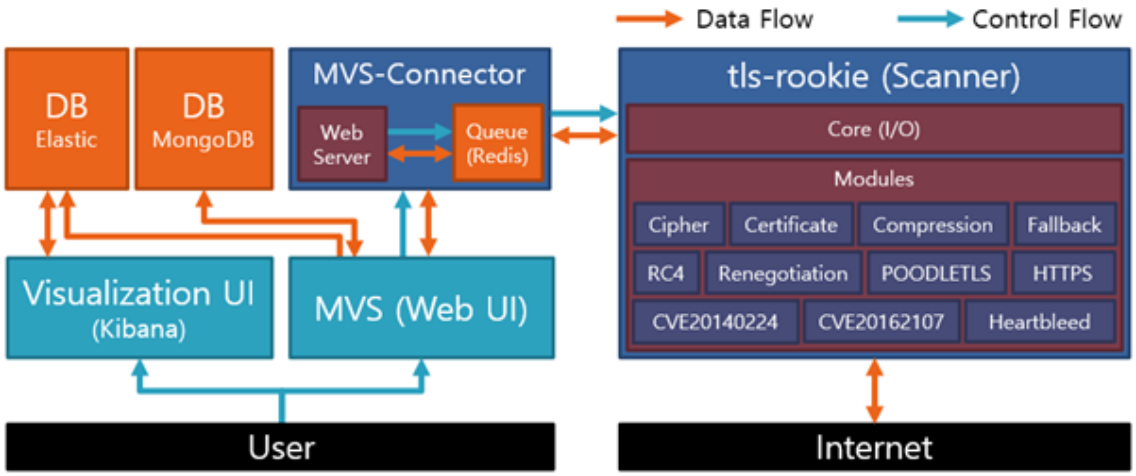

Fig. 1. Proposed SSL/TLS Scanner Overview

SSL/TLS 스캐너 시스템은 웹 인터페이스를 제공해주는 스캐너와 Go로 개발된 검사 도구 tls-rookie로 구성되어 있다. 데이터는 MongoDB[12]와 Elasticsearch[13]를 사용하여 두 곳에 저장되며, 시각화를 위해 Kibana[14]를 사용하였다.

3.1 MVS

MVS(Massive Vulnerability Scanner)는 NodeJS [15]를 이용하여 개발되었으며 취약점 검사 도구와 데이터를 교환하는 HTTP 기반 API를 정의한다. 따라서 해당 API를 구현한 가벼운 스크립트를 개발하는 것만으로도 어떠한 형태의 취약점 검사 도구와도 쉽게 연동할 수 있다. 구현된 MVS용 취약점 검사 도구는 웹 인터페이스를 통해 등록 할 수 있다. Web UI를 통해 검사할 서버를 추가, 수정 및 삭제할 수 있으며 각 서버마다 취약점 검사 도구를 추가하거나 삭제할 수 있다. 각 취약점 검사 도구 마다 특정 일 주기로 시간을 정하여 검사 예약 기능을 제공한다. 검사 결과는 가장 최근 검사 결과부터 순차적으로 표시해주며, 관리자 이메일을 입력할 경우 메이져(major) 등급 이상의 취약점이 발견되면 자동으로 메일을 발송해준다. MVS의 모든 데이터는 MongoDB에 저장하는데 검사 결과의 빠른 검색과 시각화를 위해 MongoDB 뿐만 아니라 Elasticsearch에도 결과를 동시에 저장한다.

3.2 tls-rookie 및 mvs-connector

tls-rookie는 Go 1.8에 기본적으로 포함되어있는 TLS 라이브러리[6]를 기반으로 개발되었다. SSLyze와 같이 모듈 방식으로 구성되어 있으며 Certificate, Compression, Fallback, RC4, Renegotiation, POODLETLS, CVE20140224, CVE20162107, HTTPS, Heartbleed의 10개 모듈이 개발되어 있다. 이 모듈은 (1) 프로토콜 지원 상태, (2) FALLBACK_SCSV 지원 여부, (3) TLS 압축 지원 여부, (4) 재협상 지원 여부, (5) NULL, ANON, MD5, DES, RC4, EXPORT-grade 등 취약한 암호화 알고리즘 지원 여부, (6) SSL/TLS POODLE 공격에 취약한지 여부, (7) BEAST 공격에 취약한지 여부, (8) DHE/ECDHE 키의 길이가 충분한지 여부, (9) Heartbleed 공격에 취약한지 여부, (10) CVE-2016-2107 취약 여부, (11) CVE-2014-0224 취약 여부, (12) 인증서 신뢰 / 만료 여부, (13) 인증서 서명 알고리즘 및 키 길이가 안전한지 여부, (14) HSTS 및 HPKP가 설정되어 있는지 여부를 검사하여 디버그 용도의 코드를 제외하고 총 29개의 결과 코드를 출력한다. 또한 CVSS V3 등급에 따라 보통, 마이너, 메이저, 크리티컬 (trivial, minor, major, critical)의 등급이 같이 출력된다. 다만, tls-rookie는 취약점만을 점검하기 위한 CLI 형태의 도구이므로, 인증서 주체나 인증 경로 등 부가적인 정보는 출력하지 않으며, 브라우저에 따른 연결 가능 여부 및 알고리즘 등을 확인하지 않는다. 또한, MVS와 연동하기 위해 mvs-connector를 개발하였다. 이는 MVS의 취약점 검사 도구 API를 구현하고 검사 요청을 저장하는 queue를 구현하고 있어 검사가 끝나기 전에 다른 검사 요청을 받아도 모두 수용할 수 있다.

3.3 Elasticsearch, Kibana

MVS가 Elasticsearch에 저장한 검사 결과를 바탕으로 Kibana에서 결과를 검색하고 시각화할 수 있다. Kibana 웹 인터페이스를 사용하여 “help라는 단어가 포함되어있는 도메인의 검사 결과 중 2017년 06월 15일 이후 결과”와 같은 검색이나 “n일 동안 가장 많이 발생한 마이너 (minor) 등급의 취약점 번호 상위 5개를 태그 클라우드 (tag cloud) 형태로 표시”나 “각 도메인별 발생한 취약점 개수를 막대 그래프로 표시” 같은 시각화를 쉽게 할 수 있다.

3.4 동작 과정

본 스캐닝 시스템을 동작시키기 위해서는 먼저 사용자가 MVS Web UI에서 서버를 등록하고 각 서버에 tls-rookie 검사 예약을 해야 한다. 예약이 되면 사용자가 즉시 검사를 실행시킬 수도 있고 MVS 스케줄러에 의해 자동으로 검사가 실행될 수도 있다. 두 경우 모두 MVS는 tls-rookie의 mvs-connector로 검사 요청을 보낸다. 요청은 먼저 Redis 기반 queue에 저장되며 아무것도 검사하지 않고 있는 process에 가장 먼저 들어온 요청을 전달한다. 요청을 받은 process는 tls-rookie 바이너리를 실행시켜 검사 결과를 얻고, 이를 MVS로

전송한다. MVS는 최종적으로 결과를 받아 DB에 저장하고 필요하다면 메일을 발송한다. 사용자는 MVS Web UI를 통해 이전의 모든 검사 기록을 시간 역순으로 조회할 수 있다.

IV. SSL/TLS 스캐너 취약점 탐지 결과

개발된 스캐너 시스템을 이용하여 네이버 도메인 213개에 대해 TLS 취약점 검사를 수행하였다. MVS와 tls-rookie는 독립된 컴퓨터에서 작동하였으며, 각 컴퓨터는 Intel® Xeon® CPU E5-2660 v4 @ 2.00GHz 코어 4개와 8GB 메모리를 장착하고 있다. 도메인 전체를 검사하는데 소요된 시간은 10분 30초였다. 같은 도메인 213개에 대해서 Qualys를 이용해 취약점 검사를 수행하였는데 총 1시간 26분 5초의 시간이 소요되었으며 testssl.sh로 검사를 수행했을 때는 2시간 11분 24초의 시간이 소요되었다. 이 검사 결과에서도 알 수 있듯이 본 논문에서 제안하는 시스템은 기존 시스템에 비해 검사 속도가 매우 빠른데 그 이유는 1. 주요 취약점과 관련 없는 테스트 항목 및 보안상 위협이 알려지지 않은 항목을 제외, 2. 부가적인 정보 제공 등의 기능을 제외, 3. 병렬처리를 통해 속도를 극대화, 4. 외부 라이브러리나 구현체에 의존하지 않음의 이유 때문이다. 테스트에서 사용된 213개의 도메인은 외부에 오픈되어 있는 도메인들인데 Qualys의 경우, 내부 망에서 접근 가능한 서버의 테스트로 인하여 정보 유출 가능성이 있기 때문에 공개되어 있는 도메인들에 대해서만 테스트를 수행하였다.

213개의 도메인 중 TLS를 지원하지 않는 도메인 19개를 제외한 194개에서 총 816개의 취약점이 검출되어, 각 도메인마다 평균 4.2개의 취약점이 존재

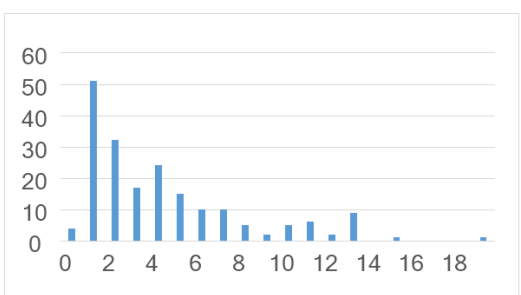

Fig. 2. Number of Domains (X) Corresponding to Number of Vulnerabilities (Y)

Table 2. TLS Scanner Result Code

Result Code	Description
0X	For Internal Use (Debug)
1X	Vulnerable version used or client-side reneogation, using unsafe version of protocols
3X-4X	Using unsafe encryption such as DES or vulnerable parameters such as 512bit key
5X	Logical bug such as Heartbleed found
7X	Certificate problem such as a certificate signed by SHA1
9X	HTTP protocol debug information (HSTS, HPKP)

했다. 각 결과 코드 (X)에 따라 검출된 도메인 개수(Y)는 Table 2와 같으며 검출된 결과 코드 개수에 따른 도메인의 개수는 Fig. 2에서 보이는 바와 같다.

검출된 취약점 중 12(SSLv3 지원)와 30(RC4 알고리즘 지원)의 경우 IE6 등 오래된 브라우저에서 접속을 가능하게 하기 위해 해당 설정이 필요한 경우가 있다. 37 및 38 (BEAST)는 TLS 1.0 및 SSL 3.0에서 발생하는 취약점으로 MITM(Man-In-The-Middle attack) 공격자가 IV(Initialization Vector)를 예측하고, 암호화된 평문의 형태를 추측할 수 있다면 평문을 얻을 수 있는 취약점이다. 주로 HTTP 세션 쿠키나 인증 정보가 유출될 수 있다. 그러나 현재는 대부분의 브라우저가 패치되어 공격자가 IV값을 예측할 수 없게 되어 더 이상 BEAST 취약점이 문제되지 않는다. 이 코드 4개를 제외하고 검출된 결과 코드 개수에 따른 도메인의 개수를 보면 Fig. 3과 같다.

검사 과정과 결과에서 몇 가지 알 수 있는 결과는 다음과 같다.

- 일부 도메인은 검사할 때마다 검출되는 취약점이 조금씩 달라진다. 이 현상은 같은 도메인 뒤에 설정이 다른 여러 대의 서버가 존재하고, 로드밸런서(load-balance) 등에 의해 연결되는 서버가 계속 변경되기 때문이다.
- 7개 이상의 취약점이 발견된 도메인 21개 중 단 7개만 리다이렉션이나 오류 응답 없이 페이지를 표시(HTTP status code 2xx success) 하였으며,

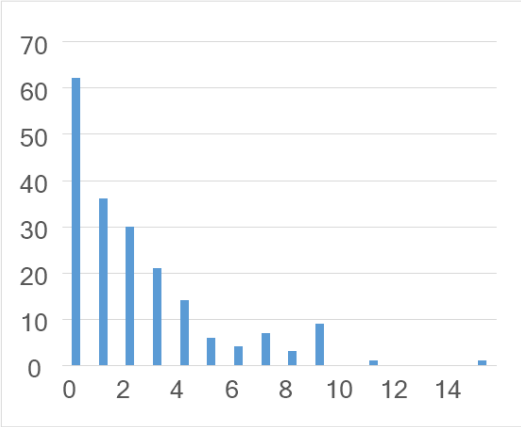

Fig. 3. Number of Domains (X) Corresponding to Number of Vulnerabilities (Y) Except Vulnerability Code 37 and 38

- 12(SSLv3), 30(RC4), 37 및 38 (BEAST)을 제외하고 가장 많이 검출된 취약점은 39 (짧은 DH키 사용)이다. 512bit 키는 개인이나 회사 급의 공격자에게 1024bit 키는 정부 기관 급의 공격자에게 취약할 수 있다. 검출된 99개의 도메인 중 512bit 키를 사용하는 도메인은 19개로, 모두 그래프 2에서 5개 이상의 취약점이 검출된 도메인이다.
- 최신 버전에서는 기본적으로 사용하지 않도록 설정되어있는 NULL, ANON, DES, EXPORT-grade 암호화 알고리즘을 사용하는 도메인이 각 7, 13, 33, 22개로 확인되었다. 이 알고리즘을 모두 비활성화해도 Windows XP의 IE6 환경에서 접속하는데 어떠한 문제도 없으므로, 일반적인 상황에서 하위 호환 문제가 발생하지 않는다.
- CVE-2016-2107은 OpenSSL의 구현 오류로 발생한 취약점이며, AES CBC에 대한 padding oracle 공격이다. OpenSSL측은 2016년 5월 3일 security advisory를 통해 OpenSSL을 업그레이드 할 것을 안내했다. 해당 취약점은 46개의 도메인에서 검출되었다. 즉, 적어도 46개 도메인에 대한 서버는 2016년 5월 3일 이후의 OpenSSL 업그레이드가 진행되지 않았으므로 이후에 발견된 보안 취약점에도 노출되어 있다.

V. 논 의

본 논문에서 제안한 시스템인 MVS 플랫폼과 SSL/TLS 취약점 스캐너인 tls-rookie의 한계점은 다음과 같다. 먼저 MVS 플랫폼은 여러 취약점 검사 도구를 연동하여 많은 서버를 검사할 수 있으므로 검사를 실행하거나 검사 결과를 열람할 수 있는 권한은 각 서버별로 분리되어야 한다. 그러나 권한 관리 기능은 구현되지 않았다. 또한, MVS 플랫폼은 취약점만을 보고하기 위해 제작되었으므로 tls-rookie에서 인증서 체인이나 HSTS 설정 등의 SSL/TLS의 상세 정보를 제공한다고 해도 사용자에게 적절히 표시할 방법이 없다. 실제로 tls-rookie에서는 HSTS 설정 여부를 확인하고 debug 코드 중 하나를 할당하여 MVS로 전송한다. 따라서 취약점 목록만 표시되는 MVS UI상 사용자가 바로 확인하기 어렵다. 이러한 MVS의 특성으로 인해 tls-rookie에서는 각 브라우저 환경에서의 handshake simulation 기능이나 인증서 체인을 표시하는 기능이 구현되지 않았다. 또한 tls-rookie는 SSLyze나 testssl.sh 같은 다른 검사도구와 달리 오픈소스가 아니므로 외부 개발자들의 도움을 받기 힘들다. tls-rookie 첫 구현 이후로 발견된 취약점인 ticketbleed 검사 기능은 구현되지 않았으며, TLS 1.3에 대한 검사 역시 지원하지 않는다.

VI. 결 론

네트워크 통신환경에서 주고받는 데이터를 보호하기 위해서 SSL/TLS가 널리 이용되고 있다. 그러나 설계 결함과 실제 구현에서의 오류로 인해서 보안 이슈가 끊이지 않고 있는데, 본 논문에서는 이러한 보안 취약점을 SSL/TLS 취약점을 자동으로 빠르게 검사하기 위한 취약점 검색 도구를 개발 하였다. MVS는 tls-rookie에 종속적인 도구가 아니라, 어떠한 취약점 스캐너라도 붙일 수 있는 플랫폼이므로 TLS 취약점 외에 다른 취약점 검사 도구와 연동하여 자동 검사를 수행할 수 있다. 특히 주요 취약점 검사 기능을 제외한 보안상 위협이 알려지지 않은 테스트 및 부가적인 정보 제공 등의 기능 구현을 제외하고 병렬처리를 통해 속도를 극대화하여 매우 빠른 속도로 검사를 수행한다. 검사 결과가 외부로 유출될 우려가 없으므로 내부 네트워크 검사에 사용하기 용이하다.

개발된 도구를 이용하여 네이버 및 관계사 도메인 213개에 대해 검사를 수행하였고, 도메인 213개에 대해 검사를 수행하여 각 도메인 당 평균 4.2개, 즉 총 816개의 취약점이 발견되었다. 하위 호환성을 위한 설정으로 취약점이 발생한 경우와 웹 브라우저에서 패치가 완료되어 문제가 없는 취약점을 제외하면 도메인 당 평균 2.3개의 취약점이 검출되었다. 7개 이상의 취약점이 발견된 도메인 21개 중 일반 사용자들이 직접 사용하는 도메인은 없었으며 50% 이상이 다른 페이지로 redirection 되거나 오류 페이지가 표시되었다. 99개의 도메인이 1024bit 이하의 DH키를 사용하며, 30개 이상의 도메인이 취약한 암호화 알고리즘인 NULL, ANON, DES, EXPORT-grade를 지원하는 것으로 파악되었고, 46개의 도메인에서 2016년에 발견된 보안 취약점에도 노출되어 있었다. 개발한 취약점 검사 시스템을 이용해서 발견된 보안 취약점들을 빠르게 대응하여 패치를 수행할 수 있었다. 향후, TLS 1.3 대응과 ticketbleed 등 새로 발견된 취약점에 대한 검사 기능을 구현할 예정이다.

References

[1] D. Adrian, K. Bhargavan, Z. Durumeric, P. Gaudry, M. Green, J.A. Halderman, N. Heninger, D. Springall, E. Thome, L. Valenta, B. VanderSloot, E. Wustrow, S.Z. Beguelin, and P. Zimmermann, "Imperfect Forward Secrecy: How Diffie-Hellman Fails in Practice," Proceedings of the 22nd ACM SIGSAC Conference on Computer and Communications Security, pp. 5-17, Oct. 2015.

[2] Z. Durumeric, J. Kasten, F. Li, J. Amann, J. Beekman, M. Payer, N. Weaver, J. A. Halderman, V. Paxson, and M. Bailey. "The matter of Heartbleed," Proceedings of the 2014 ACM Internet Measurement Conference, pp. 475-488, Nov. 2014.

[3] https://www.ssllabs.com/ssltest/

[4] https://github.com/nabla-c0d3/sslyze

[5] https://testssl.sh/

[6] C. Brubaker, S. Jana, B. Ray, S. Khurshid, and V. Shmatikov, "Using Frankencerts for Automated Adversarial Testing of Certificate Validation in SSL/TLS Implementations," Proceedings of the 2014 IEEE Symposium on Security and Privacy, pp. 114-129, May 2014.

[7] Y. Chen and Z. Su., "Guided Differential Testing of Certificate Validation in SSL/TLS Implementations," Proceedings of the 2015 10th Joint Meeting on Foundations of Software Engineering, pp. 793-804, Sep. 2015.

[8] J. Somorovsky "Systematic Fuzzing and Testing of TLS Libraries," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, pp. 1492-1504, Oct. 2016.

[9] J. De Ruiter and E. Poll., "Protocol State Fuzzing of TLS Implementations," Proceedings of the 24th USENIX Conference on Security Symposium, pp. 193-206, Aug. 2015.

[10] S. Sivakorn,G. Argyros, K. Pei, A.D. Keromytis, and S. Jana, "HVLearn: Automated Black-box Analysis of Hostname Verification in SSL/TLS Implementations," Proceedings of the 38th IEEE Symposium on Security & Privacy, May 2017.

[11] https://golang.org/pkg/crypto/tls/

[12] https://www.mongodb.com/

[13] https://www.elastic.co/products/elasticsearch

[14] https://www.elastic.co/products/kibana

[15] https://nodejs.org/

〈저자소개〉

조 성 원 (Sungwon Cho) 정회원
2014년 2월 KAIST 전산학부 입학
2017년 1월~2017년 6월: 네이버 인턴
2017년 12월~현재: 랜딧 개발자
〈관심분야〉 정보보호, 서비스 보안

최 현 상 (Hyun-sang Choi) 정회원
2004년 8월: 고려대학교 컴퓨터학과 졸업
2006년 8월: 고려대학교 컴퓨터학과 석사
2012년 2월: 고려대학교 컴퓨터학과 박사
2012년 3월~2013년 2월: 고려대학교 연구교수
2013년 2월~2014년 2월: UC Berkeley Postdoc
2014년 3월~2016년 5월: Secui 책임
2016년 5월~현재: Naver 연구원
〈관심분야〉 정보보호, 전자공학, 통신공학

허 규 (Gyu Heo) 정회원
2008년 2월: 순천향대학교 정보보호학과 졸업
2007년 12월~현재: Naver 보안팀 연구원
〈관심분야〉 정보보호, IoT보안, 서비스 보안

조 상 현 (Sanghyun Cho) 정회원
1997년 2월: 고려대학교 컴퓨터학과 졸업
1999년 2월: KAIST 전산학과 석사
2005년 2월: KAIST 전산학과 박사
2005년 3월~2006년 9월: 고려대학교 정보보호대학원 연구교수
2006년 9월~2007년 9월: KAIST 전산학과 시스템보안센터 연구원
2007년 9월~현재: Naver Security Leader
〈관심분야〉 정보보호, 서비스 보안, 이상탐지

김 영 갑 (Young-Gab Kim) 정회원
2001년 8월: 고려대학교 컴퓨터학과부전공
2003년 8월: 고려대학교 컴퓨터학과 석사
2006년 8월: 고려대학교 컴퓨터학과 박사
2006년 9월~2008년 3월: 고려대학교 정보보호대학원 연구교수
2008년 3월~2010년 1월: 국가평생교육진흥원 선임전문원
2010년 2월~2013년 2월: 고려대학교 연구교수
2013년 3월~2015년 2월: 대구가톨릭대학교 IT공학부 조교수
2015년 3월~현재: 세종대학교 정보보호학과 부교수
〈관심분야〉 사물인터넷 보안, 보안공학, 위험분석