논문 19-44-04-16 The Journal of Korean Institute of Communications and Information Sciences '19-04 Vol.44 No.04
https://doi.org/10.7840/kics.2019.44.4.736

보안에 취약한 IoT장치 식별을 위한 검색 시스템

권 영 우◆°

A Search System to Identify Vulnerable IoT Devices

Young-Woo Kwon◆°

요 약

IoT 장치들은 네트워크에 항시 연결되어 있어 여러 보안 위협에 노출되어 있으며 수많은 IoT 장치들이 설치 및 배포 후 관리가 소홀하다는 문제점이 있다. IoT 장치의 취약점 분석을 위하여 포트스캐닝 기법이 널리 사용되고 있지만, 수천만 ~ 수억대의 IoT 장치들의 취약점을 탐지하고 대응하기가 쉽지 않다. 그러므로 IoT 장치를 식별하고 취약정도를 파악하여 심각한 보안 위협에 노출된 장치들을 선택적으로 대응하는 전략이 필요하다. 본 논문에서는 이러한 문제를 해결하기 위해 보안에 취약한 IoT 장치들을 식별하고 검색할 수 있는 시스템을 제안하고 실제 네트워크 스캐닝을 통해 제안한 시스템을 평가한다.

Key Words : IoT, Security, Vulnerability, Search engine, CVSS

ABSTRACT

Because IoT devices are always connected to the network, and are neglected in management after installation and distribution, they are easily exposed to various security threats. Although port scanning is widely used for vulnerability analysis of IoT devices, it is not easy to detect and respond to vulnerabilities of tremendous IoT devices. Therefore, a strategy is needed to identify IoT devices that are vulnerable to security threats and then selectively manage them due to the limited time and management cost. In this article, we introduce a search engine system to identify vulnerable IoT devices and then evaluate the system in the real-world network.

Ⅰ. 서 론

오늘날 사물인터넷 또는IoT(Internet of Things)는 우리 삶 속의 상당한 부분을 차지하고 있다. Gartner에 따르면 2012년부터 2017년까지 6년 연속 IoT를 10대 전략 기술로 선정하고 있으며 2020 년에는 260 억대에 육박할 것으로 전망하고 있다[1]. 하지만 IoT 장치들은 항시 네트워크에 연결되어 있어 보안 위협에 쉽게 노출되어 있다는 점이 가장 큰 문제로 여겨지고 있다. IoT 장치들은 다양한 시스템으로 구성되어 있어 장치별로 취약점이 다를 뿐만 아니라 취약점별로 중요도도 다르기 때문에 IoT 장치에 대한 보안 위협 탐지와 예방이 기술적으로 어렵다. 또한, IoT 장치들은 배포 및 설치 후 지속적인 관리가 필요함에도 여러 제약 사항들로 인하여 관리가 등한시 되고 있어 IoT 장치들에 대한 보안 문제는 더욱 심각해지고 있는 실정이다.

2016년에 발생한 Mirai 봇넷에 의한 사고는 IoT 장치가 어떻게 공격에 활용되는지 잘 보여주고 있다. 악성코드에 감염된 수십만 대의 IoT 장치들로 구성된 봇넷이 DNS 공급업체인 Dyn과 수백 웹사이트를 붕괴시킨 DDoS 공격 사고가 발생하였고, 현재까지 IoT

※ 이 성과는 2017년도 정부(과학기술정보통신부)의 재원으로 한국연구재단의 지원을 받아 수행된 연구임(NRF-2017R1C1B5075658)
◆° First and Corresponding Author : Kyungpook National University, School of Computer Science and Engineering, ywkwon@knu.ac.kr, 정회원
논문번호 : 201902-473-D-RU, Received February 20, 2019; Revised March 15, 2019; Accepted March 18, 2019

장치들로 인하여 발생한 가장 큰 보안 사고로 알려져 있다. 그리고 2016년 2월 전 세계 약 7만 3000여개의 CCTV가 해킹되어 실시간으로 방송되는 사건 발생하였고 이는 IoT 장치에 대한 보안 위협이 심각한 사회 문제로까지 연결되는 계기가 되었다.

하지만 수천만에서 수억대의 IoT 장치들 중에서 어떤 장치가 보안 위협에 노출되어 있고 위협이 어느 정도인지 파악하기란 쉽지 않다. 최신 기술 동향 에서도 알 수 있듯이 IoT 장치 혹은 네트워크에 연결된 시스템들의 분석을 위하여 여러 가지 포트스캐닝 기법을 사용하고 있지만 이를 바탕으로 한 취약점 분석은 제대로 이뤄지지 않고 있다[2,3].

그러므로 본 논문에서는 IoT 장치 취약점 검색을 위한 프레임워크를 설계하고 시스템을 구현을 한다. 제안하는 프레임워크는 포트스캐닝 기법을 사용하여 네트워크에 연결된 장치들을 찾고 취약점 데이터베이스를 바탕으로 보안에 취약한 정도를 파악한다. 수집된 장치들은 검색 엔진을 통하여 검색이 가능하며 보안 위협에 더 많이 노출된 장치를 먼저 보여줄 수 있다. 개발된 IoT 검색 시스템은 Shodan이나 Censys와 같은 최신 네트워크 장치 검색 시스템과 유사한 검색 성능을 보이면서 더욱 자세한 호스트 정보를 제공함을 보였다.

논문의 구성은 다음과 같다. 2장에서 논문에서 사용되는 관련 기술 및 연구를 소개하고, 3장에서 제안하는 시스템의 상세한 구조를 설명한다. 4장에서는 제안한 시스템을 사용한 실험 결과에 대해 논하고 5장에서 본 논문에서 연구 결과를 평가하고 향후 연구를 제시하며 논문을 마무리한다.

Ⅱ. 배경 및 관련 연구

오늘날 널리 사용되는 포트스캐닝 기법은 TCP, UDP, ICMP 등을 사용하여 네트워크 트래픽을 발생시키고 응답에 따라 호스트 및 네트워크 정보를 파악한다. 하지만 이러한 기법들은 네트워크 트래픽을 발생시킴에 따라 탐지가 용이하며 방화벽에 의해 차단되는 경우도 많으며 스캐닝 도구가 점유하는 리소스의 양도 많아 많은 시간과 노력이 소모된다. 이런 문제들을 해결하기 위하여 혹은 더 많은 정보 수집을 위하여 새로운 포트스캐닝 기법이 제안되고 있으며, 최신 포트스캐닝 도구들은 이들 기법들을 적극적으로 구현하여 배포하고 있다. 그러므로 본 연구에서는 새로운 포트스캐닝 기법을 직접 개발하기 보다는 기존의 포트스캐닝 기법을 구현한 도구들을 활용하여 프레임워크를 구현하는 전략을 선택하였고 본 장에서는 관련 연구를 소개한다.

2.1 문제점 분석

일반적으로 IoT 장치와 네트워크 장치들은 기술적으로 다른 점이 거의 없지만 보안 문제에서는 큰 차이점을 보이고 있다. 네트워크 장치들은 잘 관리된 환경에 위치하고 있으며 시스템 관리자에 의해 지속적인 관리를 받는 반면, IoT 장치들은 일반 생활환경에 노출되어 있으며 한번 설치가 되면 관리가 되지 않고 방치되어 있는 경우가 많다. 또한 IoT 장치 제조업체는 영세한 경우가 많아 제품에 대한 사후 지원이 제때 이루어지지 않아 보안 위협이 발생하더라도 즉각적인 대응이 이루어지지 못 하는 경우가 많다.

그림 1, 2, 3은 앞서 설명한 IoT 환경에서의 보안 위협을 도식화 한 것으로, 네트워크에 연결된 시스템들 (일반 네트워크 장비 혹은 IoT 장치 포함)의 보안 위협과 보안 패치 적용에 따른 위협 정도를 보여주고 있다. 그림 1은 일반적인 네트워크 시스템에서의 보안 위협을 보여주고 있다. 보안 취약점이 발견(Discovery)되면 전문가 그룹에 의해 취약점에 대한 분석 및 공개 (Disclosure)가 이루어지고 취약점에 대한 패치가 배포(Patch) 및 설치가 이루어진다. 보안 취

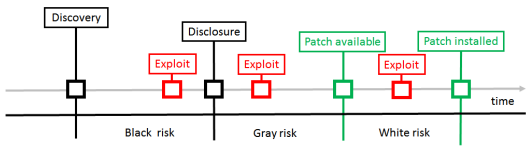


그림 1. 일반적인 네트워크 시스템에서의 보안 위협
Fig. 1. Security threat in a general network

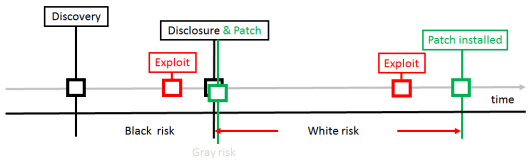


그림 2. 보안 위협 1: 보안 패치가 늦게 적용되는 경우
Fig. 2. Security threat 1: applying a security patch late

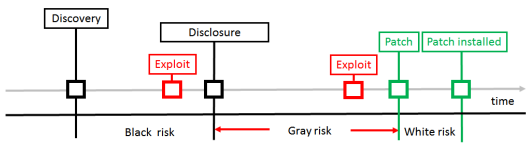


그림 3. 보안 위협 2: 보안패치가 늦게 발표되는 경우
Fig. 3. Security threat 2: releasing a security patch late

약점을 활용한 공격은 언제든지 이루어질 수 있지만 가장 많은 위협이 발생하는 구간은 그림 1에서 Gray risk 기간 (보안 위협이 공개되고 패치가 배포되기 전)이다.

하지만 IoT 환경에서는 일반적인 네트워크 시스템과는 다른 보안 위협 패턴을 가지고 있다. IoT 환경에서는 수많은 IoT 장치들이 무선으로 연결되어 흩어져 있으므로 보안위협에 대한 패치가 있더라도 쉽게 적용할 수 없어 설치가 늦어지게 되고 이로 인한 보안사고가 발생하거나(그림 2), IoT 장치를 개발하는 곳이 영세한 곳이 많아 패치의 개발이 늦어지는 경우가 많다(그림 3). 이런 이유로 한 연구에 따르면 평균 40% 이상의 IoT 장치의 패치가 전혀 이루어지지 않아 일반적인 네트워크 시스템보다 보안에 훨씬 취약한 것으로 조사되고 있다. 그러므로 본 연구에서는 보안 위협 발생 시 선제적으로 대응을 할 수 있는 IoT 장치의 취약점 분석 및 검색 시스템을 제안한다.

2.2 관련 연구

2.2.1 포트 스캐닝 도구

인터넷 규모의 스캐닝을 위한 연구는 오랜 시간 동안 이루어져 왔는데, Nmap[4] 과 ZMap[5]은 최근에 가장 널리 사용되고 있는 인터넷 규모의 스캐닝을 위한 도구로 잘 알려져 있다. Nmap과 ZMap은 모두 네트워크 트래픽을 발생시켜 해당 시스템의 정보를 습득하는데, Nmap은 TCP SYN, TCP connect, UDP 등의 메커니즘을 사용하여 호스트 및 네트워크 정보를 탐지하며 ZMap은 TCP SYN, UDP, ICMP echo 등을 사용하여 정보를 추출한다. 특히, ZMap은 IPv4 주소 공간 전체를 5분 만에 탐색할 수 있다고 알려져 있다. 본 논문에서는 Nmap과 ZMap을 모두 사용하여 호스트 정보를 취득한다.

2.2.2 IoT 장치 검색 엔진

앞서 기술한 포트스캐닝 기법과 도구들을 활용하여 IoT 장치 검색 엔진 및 서비스 개발에 대한 연구가 이루어져왔는데 본 절에서는 Shodan[6]과 Censys[7]에 대해서 소개한다. 2009년에 처음 서비스를 시작한 Shodan은 인터넷에 연결된 장치들을 검색하기 위한 검 색엔진이다. Shodan은 주로 웹서버 위주의 데이터(HTTP/HTTPS: 80, 8080, 443, 8443등) 와 FTP, SSH, Telnet, SNMP, SIP, RTSP 등과 같은 포로토콜 및 포트 정보를 수집하고 이들의 취약점 분석을 통해 관련 정보를 제공한다. Censys는 ZMap을 사용하여 IPv4 네트워크 영역을 스캐닝하고 그 결과를 토대로 더욱 자세한 스캐닝 작업을 ZGrab이라는 도구를 사용하여 수행하고 수집된 정보들은 데이터베이스에 저장한다. 저장된 검색 서비스를 통해 접근이 가능하며 Raw Data에 대한 액세스를 직접 제공하여 추가적인 데이터 분석도 가능하게 한다. 또한 다양한 웹 API를 제공하여 Censys에 직접 검색을 요청하거나 데이터를 얻을 수 있다. Shodan과 Censys는 본 논문에서 소개하는 시스템과 유사한 점이 있지만 취약점 분석 방법이 다르고 검색 엔진을 통해 취약점 정보를 제공한다는 점에서 차별성을 보인다.

2.2.3 NIST 취약점 데이터베이스 (NVD)

미국 정부의 산하 기관인 NIST에서 관리되는 데이터베이스인 NVD는 CVE 포맷[1]에 의해 관리되며 공개된 API를 사용하여 자유롭게 다른 서비스와 연계할 수 있다. CVE는 소프트웨어 취약점 및 그 공격법에 대해서 기술하는 방법으로써 고유한 ID로 구분이 가능하고 각 CVE는 CVSS[8]라는 취약점 점수를 가지고 있다. 취약점 점수는 여덟 개의 항목을 가지고 기본 점수를 계산하며 시간과 사용자의 환경을 고려하여 기본 지표 점수를 조정을 할 수 있다. 이를 위하여 CVSS에서는 시간 지표(Temporal Metric)와 환경 지표(Environmental Metric)를 정의하고 있어 일관된 점수가 아닌 시간 및 환경에 따라 취약점 점수가 다르게 계산이 가능하다.

Ⅲ. IoT 장치 식별 프레임워크

3.1 보안 취약 IoT 장치 검색을 위한 시스템

보안에 취약한 IoT 장치를 식별하기 위해서는 네트워크에 연결된 모든 장치들을 검색 후 이들의 취약점을 분석해야한다. 이를 위하여 기존에 사용되고 있는 포트스캐닝 도구인 Nmap, ZMap을 채택하여 프레임워크를 개발하였다. 프레임워크는 IPv4 네트워크에 연결된 장치들의 빠른 검색을 위하여 ZMap을 기본으로 하여 동작하며 Nmap을 통해서 더욱 자세한 정보를 습득하게 된다. 한편, 스케줄러는 취약점 데이터베이스로부터 취약점 정보를 가져오며 이를 데이터베이스에 저장한다. 포트스캐닝 도구를 사용하여 식별된 장치들의 정보는 취약점 분석을 거쳐 저장되고 검색 엔진을 통해서 제공된다.

그림 4는 앞서 소개한 각 구성 요소를 도식화한 그

1) CVE (Common Vulnerabilities and Exposures): https://cve.mitre.org/

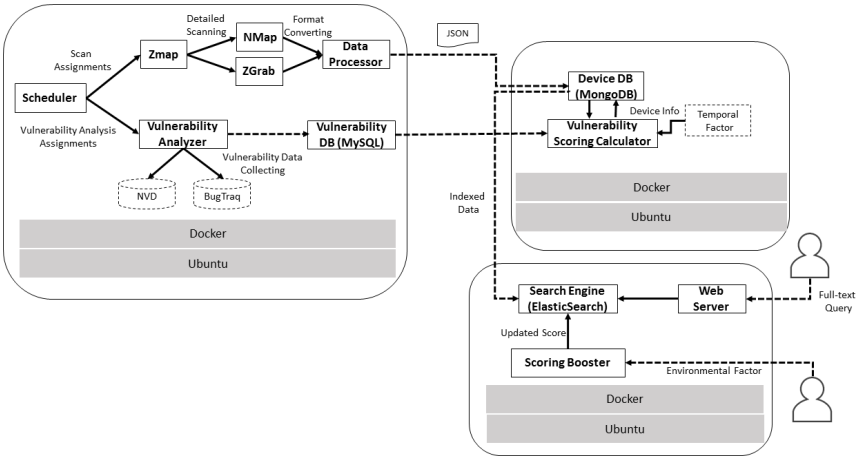


그림 4. 전체 프레임워크의 구조
Fig. 4. Framework overview

림이다. Master-Worker 모델을 기본 구조로 설계하였으며, 스케줄러(Scheduler), 취약점 분석도구(Vulnerability Analyzer), 취약점 점수 계산(Vulnerability Scoring Calculator) 모듈, 데이터 처리(Data Processor) 모듈 등이 서로 독립된 구조로 구현하였다. 그리고 수많은 IoT 장치의 정보를 저장 및 갱신하고 향후 데이터 구조의 변경에 유연하게 대처하기 위하여 NoSQL DB를 선택하였다.

그림 5는 개발된 IoT 장치 식별 프레임워크의 동작 과정을 도식화 하고 있다. 스케줄러는 ZMap은 주기적으로 실행하여 22번 혹은 80번 포트[2]가 열린 장치 및 잘 알려진 포트 위주로 스캐닝 작업을 수행하고, 응답이 있는 장치에 대해서는 Nmap을 사용해 더욱 자세한 포트스캐닝 작업을 수행한다. Nmap을 통한 포트스캐닝 작업 수행 시 무작위로 잘 알려진 100개의 포트를 선택하되 한번 선택된 포트는 최소한 7일의 기간이 지난 후에 선택하여 포트 스캐닝 작업을 수행한다. 예를 들어 첫째 날에는 22, 80, 51413[3] (transmission daemon이 사용하는 포트)을 대상으로 스캐닝 작업을 수행하고, 둘째 날에는 22, 80, 3306 (mysql이 사용하는 포트)을 대상으로 스캐닝 작업을 수행하고, 셋째 날에는 22, 80, 21 (FTP 서버가 사용하는 포트)을 선택한다.

수행되는 스캐닝 작업의 결과는 취약점 분석도구을 통해 IoT 장치에 대한 취약 정도를 점수화하여 데이터베이스에 저장한다. 그리고 새로운 취약점이 발견되었거나 패치가 이루어졌을 때에는 취약 점수를 다시

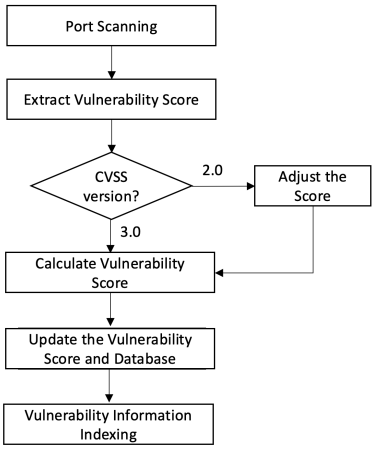


그림 5. 시스템 동작 순서
Fig. 5. Workflow of the framework

계산하는데, 이때 CVSS의 시간 지표를 함께 고려한다. 계산된 취약점 점수와 취약점 정보는 데이터베이스에 저장되며 하루 단위로 인덱싱 되어 검색 엔진(ElasticSearch)을 통한 검색이 가능하다. 사용자가 검색 질의문을 입력하면 CVSS의 환경 지표를 반영한 점수를 다시 계산하고 이를 사용하여 최종적으로 검색 결과를 정렬한다. 검색 기능은 키워드 검색을 기본적으로 지원하며 IoT 장치 정보 중 주소, 도메인 네임, OS, 포트 번호 및 서비스 이름과, 취약점 정보 등을 중심으로 인덱싱하여 검색에 활용된다. 마지막으로 검색 결과는 취약점 점수를 기준으로 정렬되어 최종 사용자에게 제공된다.

2) 22번은 SSH이며 80번은 HTTP 프로토콜에 의해 사용된다.
3) transmission 서비스에서 사용되는 포트번호

3.2 보안 취약 정도에 따른 점수 계산

NVD 취약점 정보는 약 2 ~ 3일 마다 업데이트되고 있으며 새로운 취약점 정보가 있거나 기존 취약점 정보 중 점수 변경 혹은 CPE(Common Platform Enumeration) 정보가 변경되었을 때 데이터베이스를 갱신한다. 새로운 취약점이 발견될 때 마다 해당 취약점에 대한 점수가 계산되어 NVD에 추가되는데 기존에 존재하는 CPE에 새로운 취약점이 추가 되는 경우도 있으며, 새로운 취약점이 해당 CPE에 처음 발견되어 추가되는 경우도 발생한다.

표 1은 이러한 상황을 나타낸 것으로 Solaris 운영체제에 해당하는 취약점은 1 ~ 4번이며, 해당 장치에서 제공하는 HTTP서버와 SSH 서비스에서 가지고 있는 취약점은 5번과 6번에 해당한다. 하나의 장치에 대해서 총 6개의 취약점이 발견되었으며 각각의 취약점 점수가 다름을 알 수 있다. 이를 해결하기 위하여 장치의 취약점 점수를 다음과 같이 계산한다. 먼저, IoT 장치가 가지는 모든 CPE와 관련이 있는 모든 취약점 정보를 NVD로부터 가져온 후 취약점 점수 버전에 따른 차이를 고려한 새로운 점수를 계산한다. 취약점 버전 3.0(CVSS 3.0)의 점수를 기본으로 사용하고 버전 2.0의 점수만 존재하는 경우에는 기존 연구 결과[9][10]를 사용하여 점수를 보정한다. 취약점 점수 버전 1.0과 2.0에 대한 분석 결과를 보면 버전 2.0이 버전 1.0보다 높은 점수를 가지고 있는데 평균 약 1.5가 더 높으며 중간 값은 1.2가 더 높음을 알 수 있다[9]. 또 다른 연구에서는 CVSS 3.0의 평균값은 약 버전 2.0에 비해 약 0.6 에서 0.9가 더 높음을 보이고 있다[10]. 그러므로 본 연구에서는 CVSS 값에 대한 보정으로 0.75를 더한 값($S_{CVSSv3} = S_{CVSSv2} + \frac{0.6+0.9}{2}$)을 사용한다. 각 취약점에 대한 보정이 끝나면 다음 식을 통하여 최종 점수를 계산한다.

표 1. 취약점 점수 계산 예
Table 1. Example of calculating a vulnerability score

	CVE	Original Score	New Score
1	CVE-2008-0964	9.3	10
2	CVE-2008-0965	9.3	10
3	CVE-2008-3838	7.2	7.95
4	CVE-2008-3839	4.7	5.45
5	CVE-2009-1890	7.1	7.85
6	CVE-2014-1692	7.5	8.25
Final Score		16.06	17.63

$$Score = \frac{\sum_{i=1}^{n} Score_i}{\log(1+n)} \quad (1)$$

Ⅳ. 실험 및 결과

이번 장에서는 본 연구에서 제안하는 프레임워크의 성능 및 기능을 검증하기 위하여 수행한 실험과 그 결과에 대해서 논의한다. 본 연구에서는 다음과 같은 환경에서 포트스캐닝 작업이 수행되었다.

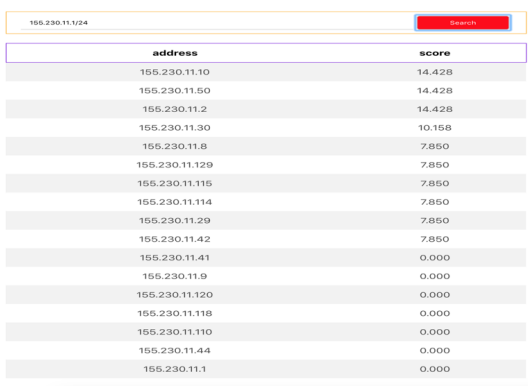
1) 포트스캐닝 대상 IP: 경북대학교 학내 망 (약 65,000 여개의 IP 주소)

2) 포트스캐닝 기간: 2018년 4월 1일 ~ 2018년 4월 30일, 매일 00:00 AM ~ 04:00 AM 실행

3) 스캐닝 대상 포트: Zmap을 사용하여 20, 21, 22, 23, 53, 67, 80, 8080, 443, 3306, 51413번을 탐색하였으며 Nmap에서는 Well-Known 포트 위주로 무작위로 100개의 포트를 스캔.

4) 포트스캐닝 결과: 총 65,536개의 IP 주소에 대한 스캐닝을 약 한 달 동안 수행하였으며, 하루 약 2,000개의 IP에 대해서 스캐닝 작업이 수행되었다. 전체 IP 중 약 10,500 개의 IP에서 응답이 있었으며 CPE 정보를 포함하는 경우는 3,733개 이었다.

그림 6에서 IP 대역으로 검색을 한 결과와 그림 7에서 특정 장치를 선택하였을 때 열려진 포트를 보여주고 있다. 키워드 검색도 가능한데 linux, cisco, windows로 검색을 하였을 때 336개, 3,284개, 94개의 장치들이 검색되었음을 확인하였다. 한편, 검색 결과는 앞서 설명한 취약점 점수 계산을 통해 정렬되어 중요한 장치를 먼저 보여주었고 (그림 6), 그림 8에서는 날짜별로 취약점수의 변화를 보여주고 있는데, 취약점


155.230.11.1/24 Search

address	score
155.230.11.10	14.428
155.230.11.50	14.428
155.230.11.2	14.428
155.230.11.30	10.158
155.230.11.8	7.850
155.230.11.129	7.850
155.230.11.115	7.850
155.230.11.114	7.850
155.230.11.29	7.850
155.230.11.42	7.850
155.230.11.41	0.000
155.230.11.9	0.000
155.230.11.120	0.000
155.230.11.118	0.000
155.230.11.110	0.000
155.230.11.44	0.000
155.230.11.1	0.000

그림 6. 검색화면 - 검색 키워드: IP대역
Fig. 6. Search result - keyword: IP range

ports

portID	state	protocol	service
22	open	tcp	ssh
88	open	tcp	kerberos-sec
445	open	tcp	microsoft-ds
548	open	tcp	afp
5900	open	tcp	vnc

그림 7. 특정 호스트에서 스캔된 서비스 목록
Fig. 7. Search results showing open services

그림 8. 취약점 점수 변화
Fig. 8. Changes of a vulnerability score

이 새롭게 발견되면 점수가 상승하고 보안패치가 적용되면 다시 점수가 하락하는 모습을 볼 수 있었다.

스캔 결과를 비교하기 위하여 Censys와 Shodan에서 같은 네트워크 영역에서 장치들을 검색한 결과, Censys에서는 4,322개의 호스트가 검색되었고Shodan에서는 8,158 개의 서비스가 검색되었다. Shodan의 경우 검색 결과가 호스트 단위가 아닌 서비스 별로 되므로 한 호스트가 여러 개의 서비스를 가지고 있는 경우 전체 검색 호스트 결과 수는 논문의 시스템보다 못 미치는 것을 알 수 있었다. 또한 Censys와 Shodan의 경우는 기본 정보(지역, 실행되고 있는 서비스 등)만 검색 결과로 제공하고 있어 해당 호스트에 대한 위협 요소를 분석할 수 없다. 하지만 본 논문에서 제안하는 시스템은 서비스와 취약점 정보를 함께 결과로 제공하므로 IoT 장치 관리에 있어 많은 도움이 될 것으로 예상된다.

본 연구의 실험 결과에 대한 첫 번째 위협 요소로는 포트스캐닝 대상이다. 국내에서는 포트스캐닝을 이용한 정보수집이 제한되어 있어 대학 학내망을 대상으로 허가 하에 포트스캐닝을 수행하였으나 다음과 같은 제약사항이 있어 다양한 실험을 하지 못하였다. 먼저 Nmap의 -A 옵션을 사용하여 스캔을 할 경우에는 방화벽에 의해 차단되어 실험 진행이 어렵게 되는 경우가 많았으며 심각한 경우에는 해킹으로 의심되어 리포팅이 되는 경우가 발생하였다. 그리고 학내 망을 대상으로 포트스캐닝을 하더라도 네트워크에 부하를 줄 수 있으므로, 제한된 시간 (매일 새벽 시간)동안 제한된 포트 (최대 10개)를 대상으로 포트스캐닝 작업이 진행되었다. 이런 제약사항으로 인하여 정확한 CPE 정보를 얻을 수 없어 점수를 계산할 수 없는 경우가 많이 발생하였으며 이러한 문제를 해결하기 위한 연구를 계속 진행할 계획이다.

V. 결 론

향후 연구로 제한된 포트 스캐닝으로 인하여 부정확한 정보를 수집되는 문제를 해결하기 위한 연구를 수행하고, CPE 정보 외에 다른 요소 (예, 서비스명, 상태) 들을 고려하여 취약점 점수를 계산할 수 있게 취약점 계산 알고리즘을 개선하고 분석 성능을 향상시키기 위한 연구를 진행할 계획이다. 또한 추가적인 분석 기법[11]을 사용하여 특정 취약점이 전파되고 패치가 적용되는 과정을 파악하여 보안 사고로 인한 피해를 더욱 줄일 수 있도록 할 계획이다. 마지막으로 본 연구와 유사하게 검색엔진을 통해 검색되는 IoT 장치들을 보안 위협으로부터 보호하기 위한 방법으로 포트 스캐닝으로부터 스캐닝이 되지 않게 설정을 변경하고 필요 없는 서비스를 중지할 것을 권고하고 있다[12]. 이러한 권고사항을 적용하기 위하여 본 연구에서 제안하는 시스템을 통해 특정 장치에서 제공되는 서비스와 취약 정도를 파악하고 보안 권고사항을 추천할 수 있다.

최근 IoT 장치는 생활과 밀접한 여러 분야에서 활발하게 적용되고 있지만 보안에 대한 준비는 미흡한 실정이다. 본 연구에서는 IoT 취약점 원격식별을 위한 프레임워크를 설계하였고 IoT 및 네트워크 장치들을 검색하여 이들의 취약 정도를 수치화할 수 있게 하였다. 또한, 수집된 장치들의 정보는 검색 엔진을 통해 쉽고 빠르게 접근이 가능하며 웹 서비스로 제공되어 타 시스템과의 연동을 통해 다양한 서비스 개발이 가능할 것으로 기대한다.

References

[1] Gartner, Inc., *Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020*, 2013.

[2] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, “Surveying port scans and their detection methodologies,” *The Computer J.*,

vol. 54, no. 10, pp. 1565-1581, 2011.

[3] A. Bonkoski, R. Bielawski, and J. A. Halderman, "Illuminating the security issues surrounding lights-out server management," in *Proc. 8th USENIX Workshop on Offensive Technol.*, Aug. 2013.

[4] G. F. Lyon, "Nmap network scanning: The official Nmap project guide to network discovery and security scanning," in *Proc. Insecure*, 2009.

[5] Z. Durumeric, E. Wustrow, and J. A. Halderman, "ZMap: Fast internet-wide scanning and its security applications," in *Proc. 22nd USENIX Secur. Symp.*, Aug. 2013.

[6] Shodan, Retrieved 09/20/2017 from https://www.shodan.io/.

[7] Z. Durumeric, D. Adrian, A. Mirian, M. Bailey, and J. Alex Halderman, "A search engine backed by internet-wide scanning," in *Proc. 22nd ACM CCS'15*, pp. 542-553, Denver, Colorado, USA, Oct. 2015.

[8] *Common Vulnerability Scoring System* (CVSS) v3.0: Specification Document, Retrieved 9/20/2017 from https://www.first.org/cvss/specification-document.

[9] K. Scarfone and P. Mell, "An analysis of CVSS version 2 vulnerability scoring," in *Proc. 3rd Int. Symp. Empirical Software Eng. and Measurement*, 2009.

[10] O. Santos, *The Evolution of Scoring Security Vulnerabilities: The Sequen*, https://blogs.cisco.com/security/cvssv3-study, Retrieved 09/07/2017.

[11] M. H. Bhuyan, D. K. Bhattacharyya, and J. K. Kalita, "Network anomaly detection: methods, systems and tools," *IEEE Commun. Surveys & Tuts.*, vol. 16, no. 1, pp. 303-336, 2014.

[12] K.-H. Han and S.-H. Lee, "A study on the security threats of IoT devices exposed in search engine," *J. KIEE*, vol. 65, no. 1, pp. 128-134, 2016.

권 영 우 (Young-Woo Kwon)

2003년 2월 : 경북대학교 컴퓨터과학과 졸업
2005년 2월 : 광주과학기술원 정보통신 공학과 석사
2014년 7월 : Virginia Tech, Dept. of CS, 공학박사
2014년 8월~2017년 6월 : Utah State University, Dept. of CS, 조교수
2017년 8월~현재 : 경북대학교 컴퓨터학부 조교수
<관심분야> 분산시스템, IoT, 보안
[ORCID:0000-0003-0625-8232]