Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.27, NO.6, Dec. 2017 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2017.27.6.1317

배너 그래빙을 통한 서버 정보 수집에 관한 연구

강 홍 구,[1†] 김 현 학,[1] 이 현 승,[1] 이 상 진[2‡]
[1]국방과학연구소, [2]고려대학교

Study on Collecting Server Information through Banner Grabbing

HongGoo Kang,[1†] HyeonHak Kim,[1] HyunSeung Lee,[1] Sang-jin Lee[2‡]
[1]Agency for Defense Development, [2]Korea University

요 약

서버 정보를 수집해 네트워크 지도를 구축하는 작업은 쉽게 발생하는 보안 사고들을 예방할 뿐만 아니라, 사이버전에 대비하고 적절한 정책을 제안하는데 방향을 제시해준다. 본 논문에서는 기존의 네트워크 스캐너인 Nmap과 ZMap을 분석하고, 네트워크 지도를 만들기 위해 서버 정보를 수집하는 기술로 배너 그래빙(banner grabbing)을 사용했다. 실시간으로 서버 정보를 수집하는 크롤링 도구를 구현하면서 정보 수집 대상이 받는 부하를 줄이기 위해 주소 생성 방식을 고안하고, 속도 향상을 위해 스레드를 나눴다. 구현한 크롤링 도구를 자체적으로 제시한 성능 평가 기준에 따라 기존 네트워크 스캐너를 사용하는 경우와 비교했다. 마지막으로, 크롤링 도구를 활용해 국내외 서버에서 정보를 수집한 DB를 바탕으로 국가별 위험 지표를 도출했으며 국가마다 차이는 있으나 수많은 사용자들이 위험한 공격에 노출되어 있는 실태를 확인했다.

ABSTRACT

To collect server information and construct network map enable us to prevent security breach, prepare for national cyber warfare and make effective policies. In this paper, we analyze well-known network scanners, Nmap and ZMap, and construct network map using banner grabbing. We use multiple threads in order to increase scanning speed and arrange IP lists by specific order to reduce the load on information gathering targets. Also, we applied performance tests to compare the real-time banner grabbing tool with the existing network scanners. As a result, we gathered server information from domestic and overseas servers and derived a risk index based on the collected database. Although there are slight differences among countries, we can identify the risky situation that many users in every country are exposed to several security breaches.

Keywords: Network Scanning, Server Crawling, Banner grabbing, Network Map

I. 서 론

점점 늘어나는 IoT 기기와 서버 및 PC의 수만큼 다양한 문제점이 새로 등장한다. 보안 사고를 막기 위해 알려진 취약점부터 대비해야 하나 말단 조직에서는 정보 기기의 취약성 파악에 높은 진입 장벽을 느낀다. 상위 관리 조직에서도 효과적으로 전체 조직의 상태를 파악할 방법이 없어 시간과 인력의 한계를 느낀다. 사이버전을 대비하는 군 및 국가 조직 또는 자산을 보호하려는 다수의 기업은 정보 기기 현황 및 실태를 파악해야 한다. 어떠한 기기가 네트워크에 연결되어 있는지 구성 연결도인 '네트워크 토폴로지'를 넘어 각 기기가 어떠한 운영체제나 애플리케이션을 사용하고 있는지 상세 정보를 담고 있는 '네트워크 지도'를 구축하는 일은 효과적인 보안 정책을 세우는 데에 도움이 된다.

Received(03. 27. 2017), Modified(08. 22. 2017), Accepted(09. 02. 2017)
† 주저자, soulbreeze_@korea.ac.kr
‡ 교신저자, sangjin@korea.ac.kr(Corresponding author)

네트워크 지도를 구축하는 데에는 정보 기기의 세부 사항을 알아내기 위한 네트워크 스캐닝이 필요한데, 해외에서는 다양한 목적으로 네트워크상의 서버나 정보 기기에 대한 정보 수집을 진행하고 있다. 2012년 10월 기준으로 KISA 허니넷에 유입된 트래픽 530만 건 중, 약 62%인 327만여 건이 해외에서 국내로 유입된 TCP 서비스 스캔이며 매년 증가하는 추세이다[1].

패치되지 않은 서버 버전 정보의 노출은 해당 서버의 알려진 취약점도 함께 노출한다. 공격자는 버전 정보와 알려진 취약점을 이용해 더 정교하고, 신속한 공격을 수행할 수 있다. 이를 예방하기 위해 국내에서도 버전 정보 현황을 포함한 네트워크 지도를 만들어야 한다. 네트워크 지도는 버전 정보 노출을 막아 알려진 취약점에 대비할 뿐만 아니라, 사이버전 대비 전략을 수립하는데 기반 자료로 활용할 수 있으며, 그 외에도 전산 시스템 운영 및 연구 목적으로 활용할 수 있다.

본 논문에서는 배너 그래빙을 이용해 국내에 네트워크 지도 구성 인프라를 구축할 수 있는 방안을 연구했다. 배너 그래빙을 기반으로 지속적인 정보 수집 서비스를 제공하는 네트워크 스캐닝 및 서버 크롤링 프레임워크(이하 크롤링 도구)를 구현했다. 그 후, 크롤링 도구의 성능을 평가하기 위한 기준을 최대한 정량적으로 제시하여 기존의 네트워크 스캐너를 복합적으로 이용하는 상황과 성능을 비교 평가한다. 추가로, 구현한 도구를 이용해 다양한 국가에서 실제로 서버 애플리케이션 버전 정보나 운영체제 정보, 제공하는 서비스 등의 정보를 수집하고, 이를 바탕으로 국가별 위험 지수를 비교한다.

II. 관련 연구

2.1 쇼단(shodan)

네트워크 지도를 구성할 수 있는 정보 수집 기술 연구를 수행하고 검색 엔진으로 제공하고 있는 곳이 '쇼단(shodan)'이다. 쇼단(https://www.shodan.io/)은 IP 주소를 기반으로 정보를 제공하는 미국의 검색엔진이다. 쇼단이 제공하는 정보는 양이 많을 뿐 아니라 정확도가 높아 해커와 보안전문가는 이를 기반으로 취약점을 탐색하기 시작했다.

검색엔진에 키워드를 입력하면 웹 기반 사용자 인터페이스 전용 HTTP 헤더를 검색하여 해당 키워드를 포함하고 있는 장치의 IP주소, 접속 가능한 포트, 국가와 도시, 위/경도 등의 정보를 표시해준다. 이러한 서비스를 제공하기 위해 데이터베이스를 구축하고 지속해서 갱신하고 있다[2]. 일반적으로 서버, 네트워크 장비, 관리자용 서비스, 웹캠, CCTV 등의 IP 주소를 수집하고 있으며, 복합기, 의학 장비 등 IP 주소를 가진 IoT 장비들도 역시 쇼단의 수집 범위에 포함되어 있다[3].

김영훈 외 2명[4]은 "쇼단에서 간단한 검색으로 다른 곳의 CCTV를 찾아볼 수 있으며 일부 시스템에 접근하여 시스템을 통제할 수 있다. 실제로 이것들이 악용될 경우, 인증 절차가 없는 디바이스에 접근하여 개인정보 및 사생활을 침해할 수 있다. 그뿐만 아니라 공공기반의 시스템을 장악하여 혼란을 일으킬 수 있다."라고 말한 바 있다.

2.2 기존 네트워크 스캐너

기존에 오픈소스로 개발된 네트워크 스캐너로 Nmap과 ZMap, Masscan이 있다. Nmap은 운영체제 핑거프린팅(OS fingerprinting) 기능과 여러 포트 스캐닝 기법을 제공하며, ZMap과 Masscan은 빠른 속도의 스캐닝 기능을 제공한다. 네트워크 스캐너들은 수집된 정보의 정확도와 스캐닝 속도를 높이기 위한 구조적 특성 및 세부 기능을 갖추고 있다.

2.2.1 Nmap

Nmap(7.12 버전)은 UNIX 환경에서 사용하는 오픈 소스 도구로, 오픈 스캔(open scan), 하프 오픈 스캔(half open scan), 스텔스 스캔(stealth scan) 등 다양한 포트 스캔 방식을 지원한다.

오픈 스캔[5]은 스캔 대상과 온전한 TCP 연결을 맺으면서 스캔하는 방법이다. 따라서 TCP 연결을 위한 3단계 핸드셰이크(3-way handshake)의 과정을 완전히 수행해야 스캔이 완료된다.

스텔스 스캔의 SYN 스캔[6]은 TCP 연결을 시도하는 척하면서 연결하지 않는 스캔 방식이다. 즉, 연결을 시도하기 위한 SYN 패킷을 보낸 후에 그에 따른 응답이 SYN, ACK인지 RST, ACK인지에 따라 해당 포트의 개폐 여부를 확인한다. 오픈 스캔과는 다르게 연결을 완료하기 전에 스캔 결과를 도출해낼 수 있으므로 오픈 스캔보다 빠르게 결과를 수집

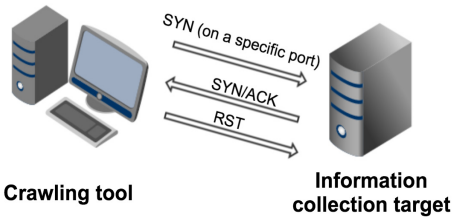


Fig. 1. The process of SYN Scan

할 수 있다.

스텔스 스캔의 FIN 스캔, 널(null) 스캔, 엑스마스(xmas) 스캔[7]은 가장 많은 서버를 스캔할 수 있지만, 패킷을 보낸 후 결괏값을 받지 못하면 포트가 열린 것으로 간주하는 역사상(inverse mapping)이기 때문에 부정확한 결과를 얻을 수 있다. 패킷이 송수신되는 과정에서 손실되어도 포트가 열린 것으로 판단하기 때문에 오탐이 많다.

2.2.2 ZMap

ZMap은 IPv4 대역의 모든 IP에 대해 지정한 포트가 열려있는지 50분 내로 알아낼 수 있는 네트워크 스캐너이다. 빠른 스캔 속도는 패킷 전송 스레드와 패킷 수신 스레드를 구분하는 구조로 인해 가능하다. 한 스레드가 스캔을 위한 패킷을 보낸 후, 응답이 오거나 타임아웃(timeout)이 될 때까지 기다리고 다음 패킷을 보내는 과정은 효율적으로 시간을 활용하지 못하기 때문이다. 또한 로우 소켓(raw socket)을 이용해 TCP 세션을 맺지 않으므로 더욱 빠르게 스캐닝한다. 탐지 패킷(probe packet)의 SYN 영역을 특정값으로 덮어서 패킷 수신 스레드는 다른 패킷과 구분해 정보를 수집할 수 있다.

그 외에도 순환군(cyclic group)을 활용한 임의 순열(random permutation)로 주소 생성(address generation)을 구현하였다. 이는 패킷을 보낼 IP에 이미 패킷을 보낸 적이 있는지 점검하지 않아도 중복되지 않도록 하며, 트래픽(traffic)으로 인한 과부하를 막는다.

ZMap은 정보 수집자의 인근 네트워크가 스캐닝 행위에 의해 가용성이 저해되지 않는다는 전제하에 최대 출력을 내며, 이론적인 최고 스캐닝 속도에 근접한다.[8]

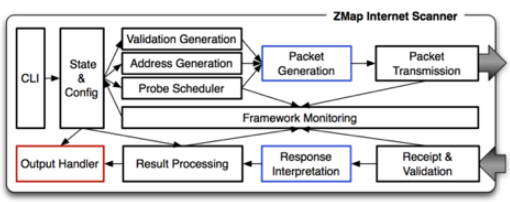


Fig. 2. Structure of ZMap

2.2.3 Masscan

Masscan은 ZMap과 동일하게 송수신 스레드를 구분하고 있으며 가장 큰 장점은 무작위 탐지에 있다. ZMap은 전체 IPv4 공간에서 분산된 IP를 반복적으로 탐지한다면, Masscan은 통계적 임의성(degree of statistical randomness)을 포기하고 연산 시간을 절약한다. 초당 천만 패킷을 전송하는데 BlackRock 알고리즘을 사용한다. 이는 암호화 알고리즘 DES의 여러 단계를 줄이며 구현한 것으로, DES가 암호화 결과로 pseudorandom 값을 내듯이 균일하게 분포한 IP값을 낸다.[9] Masscan은 입력한 IP에 대해 배너 그래빙을 지원하지만 IP 목록에 대한 정보 수집은 지원하지 않는다.

2.3 기존 웹 크롤러

2.3.1 구글(Google)의 리소스 수집

구글과 같은 검색 엔진의 경우 세계의 모든 서버가 정보 수집 대상이 된다. 저장소를 만들기 위해 정보 수집 URL을 선택하고 데이터 구조를 유지 관리하며 정보를 업데이트한다. 이 과정을 통틀어 크롤링 또는 스파이더링이라 부른다. URL은 꾸준히 달라지므로 변경되면 다시 크롤링해야 한다. 불일치성을 최소화하기 위해 URL에 주기적으로 우선순위를 지정하고 다시 방문해야 하며, 이를 증분적 크롤링(incremental crawling)이라고 한다[10].

월드 와이드 웹 초창기에는 문서가 자유롭게 변경될 수 있다는 인식으로 URL이 해당 문서의 식별방법이 아닌 리소스 지시(locator)를 위해 사용됐다. 초기의 검색 엔진 구축은 만약 문서가 나중에도 여전히 가치가 있다면, 그 URL이 안정적으로 유지될 것이라고 가정했다. 그래서 URL의 시드를 기반으로 반복적으로 페이지를 크롤링하고 새로 크롤링이 필요한 URL을 추출했다. 그러나 웹 이용자들이 방문하고자 하는 사이트는 신선한 엔터테인먼트나 뉴스 등 사회적 영향도가 큰 부분을 차지하기 때문에 효과적

인 크롤링을 위해서는 범위(coverage), 새로운 정도(freshness), 오래된 정도(age)로 가중치를 두는 증분적 크롤링이 필요하다[11]. 이러한 점은 크롤링의 방식과 정보 표현 방법 2가지 측면에서 효과적인 서버 관리를 위해 필요한 점이다.

III. 크롤링 도구 설계

3.1 정보 수집 방식 선정

서버 및 정보 기기로부터 서비스 및 애플리케이션 정보, 운영체제 정보를 수집하는 방법에는 핑거프린팅 방식과 배너 그래빙을 이용하는 방식이 있다.

Nmap은 운영체제 탐지(OS detection)나 서비스 및 애플리케이션 버전 탐지 등의 핑거프린팅을 지원한다. 오픈 소스 도구인 만큼 핑거프린팅 알고리즘에 패킷 송수신에 따른 TTL, 시퀀스 넘버 등이 많이 반영되어 있어 세부적인 정보를 알려준다. 그러나 기본적으로 탐지 패킷을 많이 주고받고, 초반에 보낸 패킷으로 탐지가 되지 않을 경우 탐지를 위해 꾸준히 여러 시도를 하므로 주고받는 패킷의 수가 점점 늘어나 시간이 많이 소요된다. 탐지 과정에서 패킷 수를 측정해 보면 Nmap과 스캔 대상이 최소 60개의 패킷을 주고받는다.

배너 그래빙은 스캔 대상과 네트워크 연결을 맺어서 운영체제 정보나 서비스 및 애플리케이션 정보를 담고 있는 배너를 받아오는 방법이다. 배너에서 필요한 부분만을 파싱한 결과에 관련 정보가 담겨있는 경우, 간단하게 버전을 확인할 수 있다. 연결을 맺는 과정에 필요한 패킷만을 사용하므로 패킷 수를 최소화하고 연산량을 줄여 소요 시간이 짧다. 온전히 연결되지 않는 경우나 배너를 보내지 않도록 설정한 대상으로부터 정보를 수집할 수 없다는 한계를 가진다.

핑거프린팅 방식은 세부적인 정보 수집 능력을 보이지만, 소요시간이 길고, 많은 패킷 송수신량으로 인해 비교적 높은 연산부담 및 네트워크 부담을 가진다. 배너 그래빙은 수집한 정보가 핑거프린팅 방식만큼 세부적이진 않지만, 소요시간이 아주 짧고, 연산부담이 낮고 정보 수집 대상 서버와 인근 네트워크 가용성에 장애를 일으킬 가능성이 작다.

3.2 부하 및 주소 생성

앞서 언급했듯이 배너 그래빙에 의해 스캔 대상이 받는 부하는 한 IP에 10개 이하로 적다. 정보 수집 대상이 패킷을 처리하느라 가용성에 장애가 발생하거나, 네트워크 과부하로 인한 서비스 문제가 일어날 가능성이 타 방식에 비해 낮다. 하지만 인접한 대상은 같은 기관에서 관리하거나 근처 지역에 있을 수 있어 공통의 라우터, 네트워크 자원을 공유하기 때문에, 짧은 시간 간격을 두고 연달아 정보를 수집하는 상황을 피해야 한다.

이를 위해 주소 생성 방식을 고안했다. /24 형태로 저장된 IP 목록(list)에서 마지막 8비트를 고정하고 0~7비트, 8~15비트, 16~23비트 순으로 IP를 정렬해 해당 순서로 스캔하는 것이다. RFC 1466의 일부인 Fig. 3에 따르면 IP 주소 체계의 맨 앞 8비트가 달라지면 대륙이 바뀌거나 국가나 조직의 단위가 바뀌게 된다[12]. 따라서 전 세계를 대상으로 정보를 수집할 때 맨 앞 8비트부터 차례로 바꾸는 방식은 지리나 네트워크에서 인접한 대상에 연달아 탐지 패킷을 보낼 가능성이 가장 낮다.

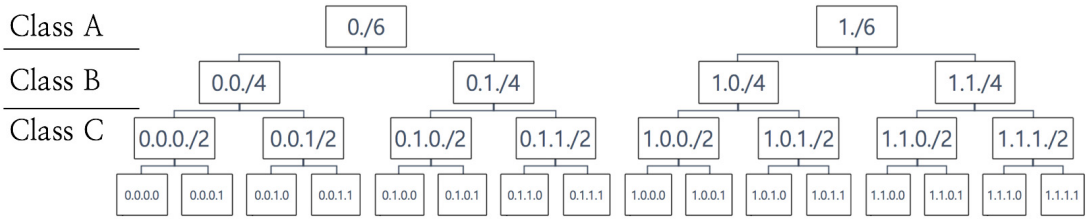
앞에서부터 8비트씩을 바꿔나가면(Fig. 4 방식2) 뒤에서부터 8비트씩을 바꾸어나가는 방식(Fig. 4 방식1)에 비해 각 클래스 단위의 네트워크가 받는 부

Table 1. Comparison of characteristics of the two methods of collecting information

Comparison \ Method	Fingerprinting	Banner Grabbing
Accuracy	High~Medium	Medium~Low
Collection rate	Slow	Fast
Number of packets used	more than 60	7~10
Target server load	High	Low

Multi-regional	192.0.0.0 - 193.255.255.255
Europe	194.0.0.0 - 195.255.255.255
Others	196.0.0.0 - 197.255.255.255
North America	198.0.0.0 - 199.255.255.255
Central/South America	200.0.0.0 - 201.255.255.255
Pacific Rim	202.0.0.0 - 203.255.255.255

Fig. 3. IP distribution by continent



Method 1	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Method 2	1	9	5	13	3	11	7	15	2	10	6	14	4	12	8	16

Fig. 4. Packet transmission order according to address generation method

하를 줄일 수 있다.

Fig. 4는 방식1과 방식2의 차이를 클래스별로 직관적으로 확인하기 위해 IP 주소 체계가 4비트라고 가정하고 그린 트리 구조이다. 이 구조에서의 순서대로 탐지 패킷을 보내면, A 클래스가 단위 시간당 받는 부하는 방식1이 방식2의 2배이다. B 클래스에서는 방식1이 방식2보다 4배 높은 부하를 받고, C 클래스에서는 8배 높은 부하를 받는다. 동일한 계산방식을 그대로 적용했을 때, 32비트 IP 주소 체계에서는 방식1은 방식2보다 A 클래스가 256배의 부하를 받으며, B 클래스는 256*256배(216), C 클래스는 256*256*256배(224)의 부하를 받는다.

1초에 약 1,400개의 IP에 패킷을 전송한다고 가정하고, 본 논문에서의 주소 생성 방식이 클래스별로 주는 부하를 살펴보면 A 클래스 단위는 1초에 약 5.5회의 탐지 패킷을 받게 되며, 각 B 클래스는 46.8초에 1회의 탐지 패킷을, C 클래스 단위는 한 번 탐지 패킷을 받고 3시간 이상 지난 후에 다시 탐지 패킷을 받게 된다.

시스코는 지난해 12월 20일 발표한 '시스코 2015~2020 글로벌 클라우드 인덱스'에서 평균 다운로드/업로드/레이턴시 속도를 기준으로 전 세계 고정형 및 모바일 네트워크 상위 10개 국가의 순위를 매겼다. 발표된 상위 10개 국가의 고정형 네트워크 평균 다운로드 속도는 66Mbps, 평균 업로드 속도는 56Mbps이다[13]. 정보 수집 과정에서 탐지 패킷은 최대로 payload를 받았을 때 총합 5.4kB 수준으로, 가장 자주 탐지 패킷을 주고받는 A 클래스의 네트워크에서도 초당 30.24kB가 된다. 이는 네트워크 평균 다운로드/업로드 속도에 크게 못 미치는 수준이며 가용성에 문제를 일으킬 가능성이 작다.

3.3 구조적 특성

구조는 전송 스레드와 수신 스레드로 나눈다. ZMap에서는 속도를 높이기 위해 스레드를 나누어 SYN 패킷을 보내고, 그에 대한 SYN, ACK을 받는다. 이를 배너 그래빙 속도 향상에도 활용한다. 따라서 열린 포트를 탐색하는 포트 스캔 과정과 통신 가능한 대상에 대한 배너 그래빙을 동시에 수행해 크롤링 속도를 높인다. 수신 스레드에서 SYN 패킷에 대한 응답과 배너 응답을 모두 받아 크기에 따라 분류하고 처리할 수 있도록 한다.

IV. 크롤링 도구 구현

설계한 방식들을 종합하여 크롤링 도구를 구현했다. Windows 운영체제 환경에서 구동되며, 기능 흐름도는 Fig. 5와 같다. 활용 포트를 추가하거나, 다른 기능 확장이 쉽도록 기능은 전부 모듈화했다.

4.1 주소 생성 방식

스캔을 시작할 때 스캔할 마지막 8비트 영역을 입력받고, 새로운 8비트 영역을 스캔 시작할 때마다 화면상에 표시해 준다. 주소 생성 함수를 통해 단순한 반복문을 이용하여 중복되지 않으면서 IP 목록을 스캔해 나갈 수 있다.

4.2 패킷 송신 스레드

패킷 송신(sender) 스레드는 SYN 스캔과 같은 알고리즘으로 동작한다. 주소 생성 함수에서 출력되는 IP 순서대로 지정한 포트에 SYN 패킷을 전송한다. 각 대상에서 돌아오는 응답은 패킷 수신(receiver)

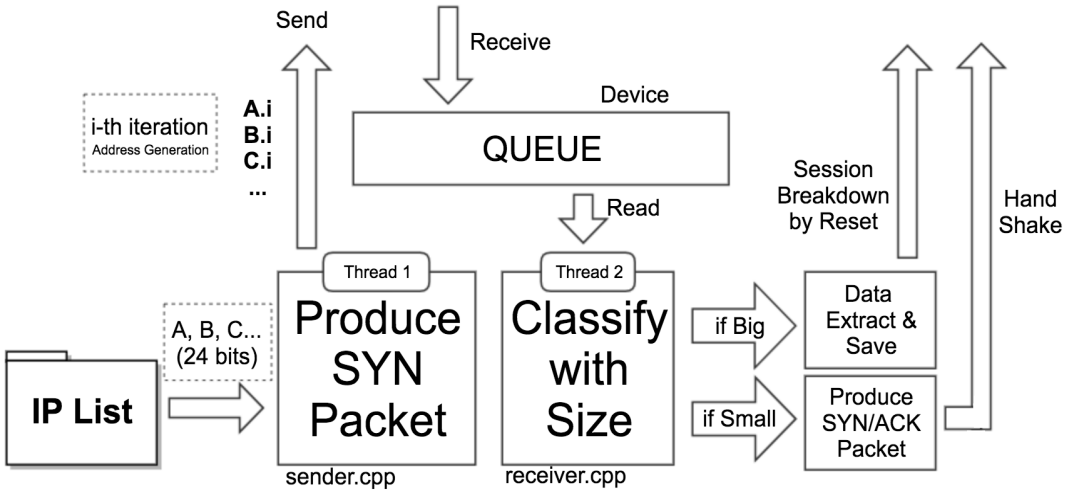


Fig. 5. Functional Flow chart of Crawling Tool

스레드에서 담당해 처리한다.

4.3 패킷 수신 스레드

대상의 해당 포트가 열려있는 경우, 응답으로 SYN, ACK이 장치(device)의 큐(queue)를 거쳐 패킷 수신 스레드로 들어온다.

패킷 수신 스레드는 패킷의 크기에 따라 처리 방식을 달리하는데, 별도의 Payload가 없는 SYN, ACK의 경우 크기가 작으므로 SYN, ACK 패킷을 확인하여 3-way handshake를 완료하고 필요한 경우 해당 IP에 배너 정보를 요청하는 패킷을 송신한다.

SSH(22번 포트)나 FTP(21번 포트), SMTP(25번 포트)의 경우 3-way handshake가 완료되면 해당 IP에서 자동으로 배너 정보를 제공하지만, HTTP(80번 포트)의 경우 TCP payload에 'GET / HTTP 1.1'을 담아 추가로 전송해주어야 응답 패킷에 서버 정보가 담겨 온다. 이처럼 배너 정보를 담고 있는 패킷은 패킷 수신 스레드에서 크기가 큰 상태로 분류하고 데이터를 추출해 SQLite DB에 저장한다. 그리고 동시에 맺을 수 있는 세션 수를 넘기지 않도록 패킷의 RST 플래그를 1로 설정한 RST 패킷을 통해 해당 IP와의 연결을 종료하며 세션을 파괴한다.

V. 크롤링 도구 활용 결과

구현한 크롤링 도구를 활용해 국내외의 서버 정보를 수집했다. 22번 포트를 통해 조사한 SSH 서버와 80번 포트를 통해 조사한 웹 서버, 21번 포트를 통해 조사한 FTP 서버, 25번 포트를 통해 조사한 메일 서버 통계를 소개하고 관련한 취약점을 함께 분석한다. 정보 수집에 소요되는 시간은 네트워크 상황에 따라 크게 달라지나 본 실험 환경에서는 국내 IP 별로 4개의 포트에 대해 약 6시간, 일본은 약 12시간, 말레이시아는 약 35분, 필리핀은 약 15분이 소요됐다. 각 국가별 IP 수는 KRNIC의 통계자료 등을 이용했으며, 이는 2017년 8월 6일 기준이다[14].

Table 2. Number of IP addresses by country

Country	Number of IP address
Korea	112,431,360
Japan	203,421,184
Malaysia	3,423,232
Philippine	5,461,472

5.1 FTP(21번 포트) 배너 수집 결과

Table 3. FTP Banner Collection Results

Country	Banner	Count	Ratio (%)	
Korea	FTP Server 1.2.4 (FTPD)	8206	31.7	Number of Banner Responses
	MikroTik FTP server (MikroTik 5.20)	4621	17.9	
	Microsoft FTP Service	1873	7.2	33,979
	MikroTik FTP server (MikroTik 2.9.27)	1771	6.8	
	vsFTPd 2.2.2	857	3.3	Number of servers verified
	vsFTPd 2.0.5	784	3.0	
	Filezilla Server version 0.9.41 beta	556	2.1	25,872
	vsFTPd 3.0.2	305	1.2	Response Ratio
	ProFTPD 1.2.9	147	0.6	
	Etc	6752	26.1	0.030%
Japan	FTP Server version 6.00LS	16072	35.0	Number of Banner Responses
	ProFTPD 1.3.5a	10992	23.9	
	vsFTPd 2.0.5	6315	1357	189,252
	vsFTPd 2.2.2	5029	10.9	
	ProFTPD 1.3.3e	817	1.8	Number of servers verified
	ProFTPD 1.3.5b	615	1.3	
	ProFTPD 1.3.4a	514	1.1	45,982
	ProFTPD 1.3.4c	386	0.8	Response Ratio
	Version 6.4/OpenBSD/Linux-ftpd-0.17	251	0.5	
	Etc.	4991	10.9	0.095%
Malaysia	Pure-FTPd privsep TLS	782	35.8	Number of Banner Responses
	Microsoft FTP Service	287	13.2	
	ProFTPD 1.3.4b	46	2.1	2,835
	vsFTPd 3.0.2	44	2.0	Number of servers verified
	vsFTPd 2.2.2	39	1.8	
	VxWorks 5.4.2	36	1.6	2,182
	FileZilla Server version 0.9.41 beta	30	1.4	Response Ratio
	ProFTPD 1.3.3c	27	1.2	
	Etc.	891	40.8	0.083%
Philippine	sv312.mngsystem.com FTP server (Version 6.00LS)	79	35.1	Number of Banner Responses
	Microsoft FTP Service	48	21.3	
	MikroTik FTP server (MikroTik 6.30.4)	18	8.0	421
	FileZilla Server version 0.9.41 beta	13	5.8	Number of servers verified
	vsFTPd 2.2.2	10	4.4	
	vsFTPd 2.0.5	8	3.6	225
	vsFTPd 3.0.2	7	3.1	Response Ratio
	FTP server (GNU inetutils 1.4.1)	6	2.7	
	Etc.	36	16.0	0.008%

5.2 SSH(22번 포트) 배너 수집 결과

Table 4. SSH Banner Collection Results

Country	Banner	Count	Ratio (%)	
Korea	OpenSSH_3.0.2p1	724	5.8	Number of Banner Responses
	OpenSSH_4.3	1318	10.5	12,560
	OpenSSH_5.1p1 Debian-5	510	4.1	
	OpenSSH_5.3	3900	31.1	Number of servers verified
	OpenSSH_6.1p1 Ubuntu-2ubuntu2.8	235	1.9	12,544
	OpenSSH_5.5p1 Debian-6+squeeze3	238	1.9	
	OpenSSH_6.6	227	1.8	Response Ratio
	OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8	235	1.9	0.112%
	OpenSSH_6.6.1	788	6.3	
	Etc.	4369	34.8	
Japan	OpenSSH_4.3	5669	8.0	Number of Banner Responses
	OpenSSH_5.1p1 FreeBSD-20080901	1955	2.8	70,826
	OpenSSH_5.3	23538	33.2	Number of servers verified
	OpenSSH_5.8p2_hpn13v11 FreeBSD-20110503	10583	14.9	70,822
	OpenSSH_6.6.1	8842	12.5	Response Ratio
	OpenSSH_7.2p2	1511	2.1	0.035%
	Dropbear_2015.67	1590	2.2	
	Etc.	17134	24.2	
Malaysia	OpenSSH_4.3	158	5.6	Number of Banner Responses
	OpenSSH_5.3	1251	44.6	3,063
	OpenSSH_5.8	93	3.3	Number of servers verified
	OpenSSH_6.6.1	277	9.9	2,804
	OpenSSH_6.6.1p1 Ubuntu-2ubuntu2	67	2.4	Response Ratio
	OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8	82	2.9	0.082%
	OpenSSH_6.6p1 Ubuntu-2ubuntu1	62	2.2	
	Etc.	814	29.0	
Philippine	OpenSSH_4.3	33	5.3	Number of Banner Responses
	OpenSSH_5.3	141	22.7	718
	OpenSSH_6.0	33	5.3	Number of servers verified
	OpenSSH_6.2	59	9.5	620
	OpenSSH_6.6.1	43	6.9	Response Ratio
	OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8	19	3.1	0.011%
	OpenSSH_7.2	35	5.6	
	Etc.	257	41.5	

5.3 HTTP(80번 포트) 배너 수집 결과

Table 5. HTTP Banner Collection Results

Country	Banner	Count	Ratio (%)	
Korea	Microsoft-HTTPAPI/2.0	11625	15.0	Number of Banner Responses
	Boa/0.94.14rc21	10093	13.0	78,062
	Apache	7405	9.5	
	AkamaiGHost	5598	7.2	Number of servers verified
	nginx	3262	4.2	77,557
	DNVRS-Webs	2880	3.7	
	PWS/8.2.0.6	2846	3.7	
	httpd	2616	3.4	Response Ratio
	GoAhead-Webs	2118	2.7	0.069%
	Apache/2.2.15 CentOS	1768	2.3	
	Etc.	27346	35.3	
Japan	Apache	166059	45.2	Number of Banner Responses
	Apache/2.2.15 FreeBSD DAV/2 PHP/5.3.5 with Suhosin-Patch	19187	5.2	367,618
	Apache/2.2.3 CentOS	16732	4.6	
	Apache/2.2.15 CentOS	16513	4.5	Number of servers verified
	Apache/2.2.25 FreeBSD PHP/5.3.27 ...	12252	3.3	366,998
	Microsoft-HTTPAPI/2.0	12251	3.3	
	Apache/2.2.22 FreeBSD PHP/5.3.10 ...	11062	3.0	Response Ratio
	Apache/2.2.31	10968	3.0	0.180%
	AkamaiGHost	10463	2.9	
	Etc.	91511	24.9	
Malaysia	Microsoft-HTTPAPI/2.0	1989	23.4	Number of Banner Responses
	Apache	1665	19.6	8,726
	Apache/2.2.15 (CentOS)	566	6.7	
	nginx	313	3.7	Number of servers verified
	DNVRS-Webs	265	3.1	8,511
	lighttpd/1.4.31	239	2.8	
	Apache/2	209	2.5	Response Ratio
	Apache-Coyote/1.1	174	2.0	0.249%
	Etc.	3091	36.3	
Philippine	Microsoft-HTTPAPI/2.0	1421	41.6	Number of Banner Responses
	Apache	588	17.2	3,522
	Apache/2.2.25	249	7.3	
	BigIP	128	3.8	Number of servers verified
	Apache/2.2.15	126	3.7	3,413
	Apache/2.2.3	60	1.8	
	Apache/2.4.6	59	1.7	Response Ratio
	nginx	55	1.6	0.064%
	Apache/2.4.7	46	1.3	
	Apache/2.2.22	42	1.2	
	Apache/2.4.18	29	0.8	
	Etc.	610	17.9	

5.4 SMTP(25번 포트) 배너 수집 결과

Table 6. SMTP Banner Collection Results

Number of Banner Responses	
Japan	263,257
Korea	104,645
Malaysia	2,793
Philippine	988
Response Ratio (%)	
Japan	0.129
Korea	0.093
Malaysia	0.082
Philippine	0.018

5.5 수집 결과 분석

5.5.1 FTP 배너 수집 결과 분석

FTP 서버 정보를 수집한 결과를 살펴보면, 국내에서 가장 많이 사용되고 있는 FTP Server 1.2.4에서는 공격자가 임의 코드를 실행할 수 있는 CVE-2011-4130 취약점이 있다.

국내와 말레이시아에서 공통으로 사용하고 있는 FileZilla server 0.9.41 beta에서는 OpenSSL SSL/TLS handshake 취약점인 CVE - 2014-0224이 존재한다.

국내와 일본, 말레이시아에서 사용되는 ProFTPd 1.3.3g 이전의 FTP 서버 애플리케이션은 임의 코드 실행이 가능한 CVE-2011-4130, Use-After - Free 취약점이 있다.

일본에서 많이 사용하는 ProFTPd 1.3.5는 원격의 공격자가 명령을 내릴 수 있도록 허용하는 CVE-2015 - 3306 취약점이 있다. 이는 default 컴파일만 해도 발생하지 않는 기본적인 취약점이다.

vsFTPd 2.3.3 이전의 버전은 일본, 말레이시아, 필리핀에서 사용하는데, vsf_Filename_passes_filter 함수가 서비스 거부(DoS, Denial of Service) 공격을 가능하도록 허용하는, CVE-2011-0762 취약점이 있다.

말레이시아와 필리핀에서 사용하고 있는 vsFTPd 3.0.2는 deny_file 속성이 'unknown' 값을 가지도록 설정한 벡터를 이용하면 접근 제한을 우회할 수 있는 CVE-2015-1419 취약점이 있다.

5.5.2 SSH 배너 수집 결과 분석

SSH 서버 정보를 수집한 결과로는, 모든 국가에서 OpenSSH 5.3이 가장 많이 사용되고 있다. 이 서버 애플리케이션 버전에서는 2010년부터 2016년까지 등록된 다양한 CVE가 있다. OpenSSH 4.3도 조사한 모든 국가에서 공통으로 사용한다. OpenSSH 5.6 이전의 버전은 J-PAKE 프로토콜에서 인증 우회가 가능한 CVE-2010-4478 취약점을 가지며, OpenSSH 6.4 이전의 버전은 Makefile.inc가 특정 자료 구조를 초기화하지 않아 원격의 공격자가 서비스 거부 공격을 시도할 수 있도록 하는 CVE-2014-1692 취약점이 있다.

OpenSSH 4.3은 추가로 원격의 공격자가 특정 SSH 패킷을 이용해 서비스 거부 공격을 시도할 수 있도록 하는 CVE-2006-4924 취약점이 있다.

일본에서 많이 사용하는 OpenSSH 5.8p2는 J-PAKE 프로토콜을 실행하기 위해서 Makefile.inc를 수정할 때, 특정 구조체를 초기화하지 않아서 메모리 충돌(memory corruption)이 발생할 위험이 있는 CVE-2014-1692 취약점이 있다.

5.5.3 SMTP 배너 수집 결과 분석

SMTP 배너 정보를 수집한 결괏값은 대부분 어떤 업체나 기관에서 배너를 노출하고 있는지 알려준다. 이는 통계를 내기에는 특징적이지도 않고 적절하지 않은 정보이다. 다만, 드물게 HTTP 배너와 같이 웹 서버 버전 정보를 제공했다.

5.5.4 HTTP 배너 수집 결과 분석

결과 중에는 프린터, IP 카메라 등 기기 정보도 일부 나왔으나 그 비율이 아주 낮다.

모든 국가에서 많이 등장한 배너 'Apache'는 이용자들이 간단한 설정을 통해 구체적인 버전 정보 노출을 숨긴 결과이다. 하지만 그 외의 결과에는 구체적인 버전이나 운영체제 정보가 담겨 있다.

조사한 모든 국가에서 공통으로 사용하는 Apache/2.2.15 버전은 CVE-2011-3192와 CVE - 2012 - 0883 취약점이 있다. 이는 각각 메모리 및 CPU consumption을 통한 서비스 거부 공격이나 로컬 사용자의 관리자 권한 획득에 해당한다. Apache 2.2.15와 필리핀과 일본에서 사용하는 Apache 2.2.3은

dav_xml_get_cdata 함수에서 CDATA에 포함된 공백을 제대로 검사하지 않아 공격자에게 서비스 거부 공격 시도를 허용한다.

필리핀에서 사용하는 Apache/2.4.6은 원격의 공격자가 서비스 공격을 시도할 수 있도록 허용하고, 임의 코드를 실행할 수 있도록 하는 CVE-2014 - 0226 취약점이 있다.

국내와 말레이시아, 필리핀에서 가장 많이 사용하고 일본에서도 일부 사용하는 서버 헤더 Microsoft - HTTPAPI /2.0는 서버 버전이 Windows 2003 Sp2, Windows 7, Windows 2008, Windows 2008 R2 중 하나임을 알려준다. HTTPAPI /2.0은 FastCGI가 실행될 때, IIS에서 버퍼 오버플로우를 발생시켜 공격자가 임의 코드를 실행할 수 있도록 하는 CVE-2010-2730 취약점이 있다.

국내에서 약 13.0%가 사용하는 Boa/0.94 14rc21은 원격의 공격자가 윈도우 제목을 수정하거나, 임의의 명령어를 입력 또는 파일을 덮어쓸 수 있는 CVE-2009-4496 취약점이 있다.

말레이시아에서 사용하는 lighttpd/1.4.31은 SQL injection(CVE-2014-2323), 디렉터리 탐색(directory traversal, CVE-2014- 2324) 취약점이 있다.

VI. 다른 크롤링 도구와의 비교

현재, 오픈소스로 공개되어있고, 공식적으로 연구되어있는 네트워크 스캐너는 Nmap과 ZMap, Masscan 등이 있다. 본 논문에서 구현한 크롤링 도구는 비동기화 방식으로 포트의 개폐 여부를 확인함과 동시에 서버의 정보를 가져온다. 이 목적을 달성하기 위한 크롤링 도구의 효율성을 확인하기 위해 기존의 네트워크 스캐너와 비교한다. ZMap의 SYN 스캔을 통해 지정한 포트가 열린 서버를 확인하고, 해당 IP를 Nmap이 확인하는 경우가 첫 번째 비교 대상이다. 배너 그래빙을 지원하는 Masscan에 본 논문에서 구현한 주소 생성 방식으로 IP를 입력하며 반복적으로 실행하는 경우가 두 번째 비교 대상이다.

구현한 크롤링 도구는 실시간으로 서버를 스캐닝하면서 데이터베이스를 업데이트하는 지속적인 서비스를 구축한 것이고, Masscan 및 ZMap과 Nmap은 일회성 도구라는 차이는 있으나 두 방법 모두 대상 서버의 정보를 얻어온다는 점에서 동일한 기능을 하기 때문에 속도와 수득률 측면에서 비교했다.

6.1 성능 평가 기준

기본적으로 크롤링 도구의 성능은 정보 수집 속도와 정보의 정확도에 달려있다. 크롤링을 통해 얻어낸 서버 정보는 사실 그 종류가 다양하므로 서로 다른 정보들에 대해 정보의 양이나 기능의 효율성을 정량적으로 비교하기가 쉽지 않다. 그런데도, 대략적인 비교를 위해 식 (1), (2)와 같은 기준을 세웠다.

$$\frac{DB\text{ 저장 크기}}{\text{받은 데이터 크기}} \fallingdotseq \text{수득률} \tag{1}$$

$$\frac{DB\text{ 저장 크기}}{\text{보낸 데이터 크기}} \fallingdotseq \text{크롤링 속도} \tag{2}$$

크롤링 도구의 성능은 시간과는 독립적이어야 한다. 시간에 관한 변수는 컴퓨터 성능이나 그 지역 네트워크 환경에 많은 영향을 받기 때문이다. 크롤링 도구의 성능은 컴퓨터나 네트워크 환경에 영향받지 않는 절대적인 평가 기준이 되어야 한다. 따라서 위 식과 같이 수득률은 받은 데이터 크기 당 DB 저장 크기, 크롤링 속도는 보낸 데이터 크기 당 DB 저장 크기(정보 수집 효율)로 정하면 컴퓨터 성능이나 네트워크 환경에 영향받지 않는 합리적인 평가 기준이 된다.

본 연구에서 개발한 크롤링 도구는 ZMap과 같이 송신 스레드와 수신 스레드를 구분지어 놓는 방식을 이용한다. 즉, ZMap과 마찬가지로 네트워크 환경이 고정되면 업로드 속도가 최대로 결정된다. 업로드 속도는 (보낸 데이터 크기 / 시간) 이라고 할 수 있으므로, 식 (3)이 되어, 시간당 DB 저장 크기와 다운로드 속도가 결정된다.

$$\begin{aligned}&(\text{크롤링 속도}) \times (\text{업로드 속도}) \\ &= \frac{DB\text{ 저장 크기}}{\text{시간}} = \frac{\text{수득률}}{\text{다운로드 속도}}\end{aligned} \tag{3}$$

즉, 크롤링 속도가 높더라도 시간당 DB 저장 크기는 업로드 속도에 의존하며, 이는 네트워크 환경에 따라 달라진다. 마지막 식에 의하면, 수득률이 높더라도 마찬가지로 다운로드 속도에 의존한다. 하지만 반대로 업로드 속도가 고정된 경우, 크롤링 속도가 높을수록 시간당 DB 저장 크기가 증가하기 때문에 성능이 더 좋다.

6.2 성능 평가

성능 평가 시에 모든 IP 대역과 모든 포트를 대상으로 하면, 시간도 오래 걸리고, 데이터의 크기가 커져 다루기 쉽지 않기 때문에 일본 IP 대역 중 일부 대역과 21, 22, 25, 80번 주요 4개 포트에 대해서만 테스트를 5회 진행했다.

구현한 크롤링 도구, 반복적으로 주소를 입력한 Masscan 실행, ZMap과 Nmap의 복합적 사용에 따른 수득률과 크롤링 속도 결과는 Table 7~9와 같다.

Table 7. Performance evaluation result of the crawling tool implemented in this study

	send (KB)	recv (KB)	db (KB)	db/recv	db/send
1	139488	17070	756	0.0443	0.0054
2	123592	14176	784	0.0553	0.0063
3	138866	16264	732	0.0450	0.0053
4	138906	16316	740	0.0454	0.0053
5	139161	16545	752	0.0455	0.0054
Avg.	136003	16074	753	0.0471	0.0055

Table 8. Performance evaluation result of the Masscan(banner grabbing)

	send (KB)	recv (KB)	db (KB)	db/recv	db/send
1	16060	967	76	0.0786	0.0047
2	16061	959	78	0.0813	0.0049
3	16077	966	76	0.0787	0.0047
4	16056	914	77	0.0842	0.0048
5	16058	955	77	0.0806	0.0048
Avg.	16062	952	77	0.0807	0.0048

Table 9. Performance evaluation result of complex use of Nmap & ZMap

	send (KB)	recv (KB)	db (KB)	db/recv	db/send
1	18638	1298	81.6	0.0377	0.0026
2	18254	1138	72.6	0.0383	0.0024
3	17248	748	45.6	0.0366	0.0016
4	18022	1047	64.8	0.0371	0.0022
5	18247	1178	70.8	0.0361	0.0023
Avg.	18082	1082	67.1	0.0372	0.0022

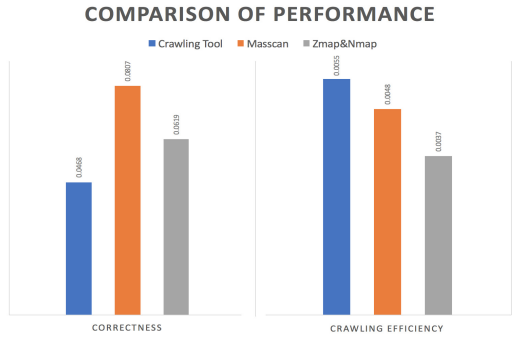


Fig. 6. Comparison graph of Performance between implemented crawling tool, Masscan and complex use of ZMap & Nmap

구현한 크롤링 도구의 평균 수득률은 0.0471, 평균 크롤링 속도는 0.0055이다. Masscan의 평균 수득률은 0.0807, 평균 크롤링 속도는 0.0048이다. 기존 네트워크 스캐너의 복합사용 평균 수득률은 0.0372, 평균 크롤링 속도는 0.0022이다. 기존 네트워크 스캐너 ZMap과 Nmap을 이용하는 경우는 Masscan을 사용하는 경우보다 정보의 수득률과 크롤링 속도 면에서 모두 떨어진다. Masscan은 크롤링 도구에 비해 정보의 수득률이 높으나 크롤링 속도는 크롤링 도구가 약 15% 더 높다.

VII. 결 론

스캔한 IP 수 대비 배너 값을 응답으로 보낸 서버의 비율은 한국을 기준으로 일본이 1.44배, 말레이시아가 1.63배 많았다. 필리핀은 오히려 0.33배로 적게 나타났다. 서버로 이용되는 IP의 비율에 따라 달라질 수 있는 값이므로 국가적 보안성을 평가할 수는 없으나, 결과적인 위험성을 간접적으로 나타낸다. 배너 값 노출에 의한 위험성은 조사한 다른 국가에 비해 우리나라가 낮았다. 하지만 응답률이 비교적 낮다고 해도 취약점이 있는 서버를 다수 사용하는 만큼 위험에 노출되어 있다.

DB에 저장된 수많은 IP의 운영체제 정보, 서버 애플리케이션 정보는 간단한 검색으로 알려진 취약성을 즉각적으로 파악할 수 있는 데이터이다. IP 목록을 수정하면 본 논문에서의 크롤링 도구를 조직 내부망, 폐쇄 망에 대한 정보 수집에도 사용할 수 있으므로 취약한 기기를 파악하며 말단 조직에서도 낮은 난이도로 보안 관리를 수행할 수 있다.

더 큰 규모로 정보를 관리하기 위해서는, 다른 웹

크롤러에서 정보에 가중치를 두고 수집하는 을 수행하는 것처럼 서버 정보를 저장하고 효과적으로 관리하는 방법에 대한 연구가 필요하다.

다른 크롤링 도구와 비교하면서, 배너 그래빙 방식으로 서버 정보를 수집할 때에도 도구마다 송수신 패킷 크기와 송수신 패킷 비율이 다를 수 있음을 알게 됐다. 배너 그래빙 방식에 대한 추가 연구를 통해 정보 수집 대상 서버와 인근 네트워크 가용성에 장애가 발생할 가능성을 더욱 낮출 수 있기를 기대한다.

또한 정의한 수득률과 크롤링 속도가 어떤 형태의 함수로 크롤링 도구의 성능을 구성하는지 밝힌다면 성능을 구체적인 수치로 비교 가능하다. 크롤링 도구의 성능을 수득률과 크롤링 속도의 함수로 정의하고 함수 형태를 조사하는 후속 연구가 필요하다.

더 큰 규모에서 취약한 기기 분포를 파악하고 네트워크 지도를 만드는 작업은 사이버 방어 전략을 수립하는 데 도움을 줄 수 있을 것으로 기대한다. 새로운 취약점을 발견했을 때 이 취약점을 가지게 되는 서버 운영체제 버전이나 애플리케이션 버전 정보를 확보한다면, 조직 단위 또는 국가 단위에서 얼마나 많은 서버가 해당 취약점에 노출되어 있는지 신속하게 파악할 수 있다. 취약한 서버는 좀비로 악용될 가능성이 있다. 이를 방지하기 위해 국가 차원에서의 서버 시스템 보안 강화 체계가 필요하며 본 연구가 이에 활용되길 기대한다.

References

[1] Korea Internet & Security Agency, "The Monthly Trends and Analysis of Internet Invasion Accident," Vol.10, Oct. 2012.

[2] Shodan, "The search engine for 'Internet of Things'," https://www.shodan.io/ (accessed Jan. 9th, 2017)

[3] Kim Zetter, "The Biggest Security Threats We'll Face in 2016," https://www.wired.com/2016/01/the-biggest-security-threats-well-face-in-2016/ (accessed Jan. 1st, 2016)

[4] Yeong Hoon Kim, Joon Keun Yang and Hak Beom Kim, "M2M/IoT Trends and Security Threats," Journal of the Korea Institute of Information Security & Cryptology, Vol. 24(6), pp. 53, Dec. 2014.

[5] G.F. Lyon, Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning, Insecure, USA, Chapter 5.3, pp. 100-101, 2008.

[6] G.F. Lyon. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning, Insecure, USA, Chapter 5.2, pp. 96-99, 2008.

[7] G.F. Lyon. Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning, Insecure, USA, Chapter 5.5, pp. 107-110, 2008.

[8] Zakir Drumeric, Eric Wustrow and J. Alex Halderman, "ZMap: Fast Internet-wide Scanning and its Security Applications," Proceedings of the 22nd USENIX Security Symposium, pp. 605-619, Aug. 2013.

[9] Graham. Robert David, "MASSCAN: Mass IP port scanner," https://github.com/robertdavidgraham/masscan (accessed Jul. 28th 2017)

[10] McCurley. Kevin S, "Incremental Crawling," Encyclopedia of Database Systems, Springer New York, pp. 1-5, pp. 1417-1421, 2016.

[11] Brin. Sergey and Lawrence Page, "The anatomy of a large-scale hypertextual web search engine," Computer networks and ISDN systems 30.1, pp. 107-117, 1998.

[12] Gerich, Elise, "Guidelines for management of IP Address Space," RFC 1466, May 1993.

[13] Cisco Visual Networking, "Cisco Global Cloud Index: Forecast and Methodology 2015-2020," White Paper, 2016.

[14] KRNIC, "IP Address/AS Number Overseas status: Status by Countries," Korea Network Information Center, http://krnic.or.kr/jsp/infoboard/stats/countryCurrent.jsp (accessed Aug. 6th, 2017).

〈저자소개〉

강 홍 구 (HongGoo Kang) 정회원
2017년 2월: 고려대학교 학사 졸업
2017년 7월~현재: 국방과학연구소 2본부 3부
〈관심분야〉 네트워크 포렌식, 네트워크 보안

김 현 학 (HyeonHak Kim) 정회원
2017년 2월: 고려대학교 학사 졸업
2017년 7월~현재: 국방과학연구소 2본부 3부
〈관심분야〉 암호이론, 취약점 분석

이 현 승 (HyunSeung Lee) 학생회원
2017년 2월: 고려대학교 학사 졸업
2017년 7월~현재: 국방과학연구소 2본부 3부
〈관심분야〉 네트워크

이 상 진 (Sang-jin Lee) 종신회원
1987년 2월: 고려대학교 학사 졸업
1989년 2월: 고려대학교 석사 졸업
1994년 8월: 고려대학교 박사 졸업
1989년 10월~1999년 2월: ETRI 연구원 역임
1999년 3월~2001년 8월: 고려대학교 자연과학대학 조교수
2001년 9월~현재: 고려대학교 정보보호대학원 교수
〈관심분야〉 디지털 포렌식, 심층 암호, 해쉬 함수