Journal of The Korea Institute of Information Security & Cryptology ISSN 1598-3986(Print)
VOL.33, NO.3, Jun. 2023 ISSN 2288-2715(Online)
https://doi.org/10.13089/JKIISC.2023.33.3.459

소규모 네트워크의 IoT 보안을 위한 저비용 악성코드 탐지 시스템 설계 방안 연구

신 상 윤,[1†] 이 다 희,[2] 이 상 진[3‡]
[1,3]고려대학교 (대학원생, 교수), [2]SK쉴더스 (책임연구원)

Design Method of Things Malware Detection System(TMDS)

Sangyoon Shin,[1†] Dahee Lee,[2] Sangjin Lee[3‡]
[1,3]Korea University (Graduate student, Professor),
[2]SK shieldus (Senior Researcher)

요 약

IoT 기기는 임베디드 장비와 컴퓨터 네트워크의 발전으로 그 수가 폭발적으로 늘어나고 있다. 이에 따라 IoT에 대한 사이버 위협이 증가하고 있으며, 현재 IoT 기기를 대상으로 악성코드를 유포하여 감염시키고 DDoS 공격에 악용하고 있다. 현재 이와 같은 공격의 대상이 되고 있는 IoT 기기는 설치 환경이 다양하며 기기의 자원이 제한적이다. 또한 IoT 기기는 한번 설정하면 소유자가 관리에 신경을 쓰지 않는 특성이 있다. 이 때문에 IoT 기기는 악성코드가 감염되기 쉬운 관리의 사각지대가 되어가고 있다. 이러한 어려움 때문에 IoT 기기는 악성코드의 위협이 항상 존재하며, 감염되면 대응이 제대로 이루어지고 있지 않다. 본 논문에서는 IoT 환경 특성을 고려하여 IoT 전용 악성코드 탐지 시스템을 설계하고 해당 시스템에서 사용하기 적합한 탐지 규칙을 제시할 것이다. 해당 시스템을 활용하면 이미 설치되어 사이버 위협에 노출되어 있는 IoT 기기의 구조를 변경하지 않고 저렴하고 효율적으로 IoT 악성코드 탐지 시스템을 구성할 수 있을 것이다.

ABSTRACT

The number of IoT devices is explosively increasing due to the development of embedded equipment and computer networks. As a result, cyber threats to IoT are increasing, and currently, malicious codes are being distributed and infected to IoT devices and exploited for DDoS. Currently, IoT devices that are the target of such an attack have various installation environments and have limited resources. In addition, IoT devices have a characteristic that once set up, the owner does not care about management. Because of this, IoT devices are becoming a blind spot for management that is easily infected with malicious codes. Because of these difficulties, the threat of malicious codes always exists in IoT devices, and when they are infected, responses are not properly made. In this paper, we will design an malware detection system for IoT in consideration of the characteristics of the IoT environment and present detection rules suitable for use in the system. Using this system, it will be possible to construct an IoT malware detection system inexpensively and efficiently without changing the structure of IoT devices that are already installed and exposed to cyber threats.

Keywords: IoT Malware, Malware Pattern Analysis, Malware Detection System

Received(03. 07. 2023), Modified(1st: 04. 03. 2023, 2nd: 05. 02. 2023), Accepted(05. 03. 2023)

† 주저자, loossy@korea.ac.kr
‡ 교신저자, sangjinlee@korea.ac.kr(Corresponding author)

I. 서 론

현재 IoT(Internet of Things) 기기는 임베디드 장비와 컴퓨터 네트워크의 발전으로 대중화되고 있다. 특히 개인과 소규모 사업자들이 센서, 카메라 및 기타 기기로 구성된 IoT 네트워크를 사용하면서 IoT 기기의 수가 폭발적으로 증가했다.

Explodingtopics 통계[1]에 따르면 Fig.1과 같이 2019년에 인터넷이 연결된 기기 중 IoT 기기가 차지하는 비율이 50%가 넘었으며 증가는 더 가파를 것으로 예상된다.

또한 IoT 기기는 일반적으로 사용자의 편의성 중심으로 개발이 진행되고 있으며, 이에 따라 사용자가 최대한 관리하지 않아도 되는 방향으로 발전하고 있다. 하지만 이러한 편의성 때문에 IoT 기기는 관리의 사각지대가 되어가고 있다.

그리고 IoT 기기들은 항상 인터넷에 연결되어 있으며 이는 항상 위협에 노출될 수 있다는 의미이기도 하다. 악성코드 유포지를 공개하는 urlhaus[2]의 통계 자료에 따르면 Fig.2와 같이 유포되는 악성코드 중 IoT 악성코드가 대부분을 차지하고 있으며 이는 지금도 IoT 기기에 많은 침해 사고가 발생하고 있다는 것을 의미한다.

개인 및 소규모 사업자가 많이 사용하는 IoT 기기는 대표적으로 IP Camera, DVR(Digital Video Recorder), NAS(Network Attached Storage)가 있는데, shadow server[3]에 따르면 Fig.3과 같이 개인 및 소규모 사업자가 많이 사용하는 video-system IoT 기기들이 2년 동안 악성코드에 상당히 많이 감염되었고 증가 폭이 가파른 것을

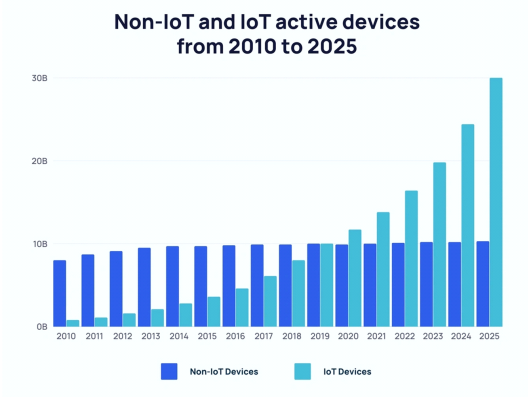

Fig. 1. IoT Device Percentage of Total Devices

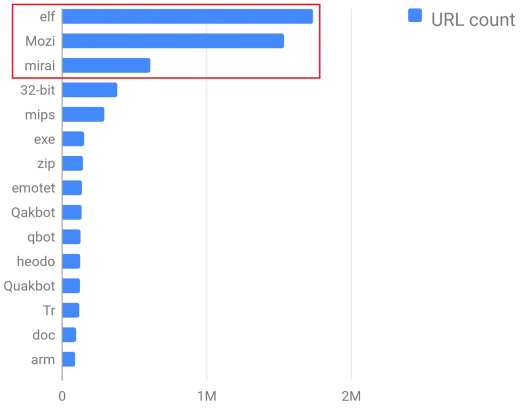

Fig. 2. IoT Malware URL Statistics

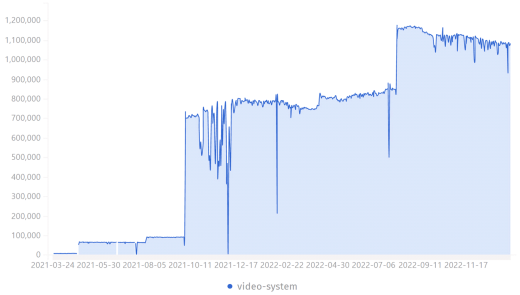

Fig. 3. Video-System devices of infected to malware

확인할 수 있다. 이러한 개인 및 소규모 사업자가 사용하는 IoT 기기들이 악성코드에 감염되어 DDoS 공격에 악용되고 있으며, 향후 민감한 영상 데이터나 개인 정보 유출과 같은 심각한 문제로 이어질 수 있다.

이와 같이 특수한 환경인 IoT 기기 대상으로 현 시장에서 IoT 보안 솔루션을 판매 중이지만, 대부분 기업을 위한 솔루션이며 가격이 비싸다. 그리고 개인 및 소규모 사업장은 사용하는 IoT 기기들을 위해 보안 솔루션에 많은 비용을 지불하기 어렵고, 기기 소유자들은 잘 동작하고 있는 네트워크 환경을 변경하는 것을 부담스러워 한다.

이에 따라, 본 논문에서 제시하는 문제점은 가장 많은 침해사고가 발생하고 있는 개인 및 소규모 사업장의 IoT 기기들이 보호받지 못하고 있다는 것이다. 이 대상들을 보호하기 위해서는 소규모 네트워크에 적합하고 저비용으로 쉽게 구축 가능한 IoT 전용 보안 시스템이 필요하다.

따라서 IoT 전용 보안 시스템 설계를 위한 3가지 핵심 문제를 도출하였다. 첫째, IoT 악성코드의 특

성인 다양한 아키텍처에 대응해야 한다. 둘째, IoT 기기가 설치된 저전력 환경을 고려해야 한다. 셋째, 저비용으로 쉽게 구축이 가능해야한다. 이에 본 논문에서는 핵심 문제를 고려하여 IoT 악성코드의 특성을 파악하고 효율적인 탐지 방법을 제안하며, 이를 이용하여 저비용으로 구축 가능한 IoT 악성코드 탐지 시스템을 제시한다.

II. 관련 연구

Khraisat[4]는 현재 연구된 여러 가지 IoT 전용 악성코드 탐지 시스템에 대한 제안과 방법론, 배포 전략, 유효성 검사 전략, 사용되는 Dataset에 대한 한계를 비판적으로 검토하였다. 이러한 연구들은 IoT 아키텍처로 인하여 공격을 모두 탐지하는데 문제가 있을 수 있다고 주장하며 다음과 같이 제안하였다. 첫째, 많은 양의 데이터가 있기 때문에 오경보가 적어야 한다. 둘째, IoT 센서의 에기치 않은 동작에도 적응할 수 있어야 한다. 셋째, 새로운 취약점이 발견되면 탐지할 수 있어야 한다. 넷째, 스스로 학습할 수 있는 빅데이터 기반 머신러닝, 딥러닝 기술을 적용해야 한다. 더 나아가 앞으로는 자체 구성, 자체 최적화, 자체 보호 및 자체 치유 기능을 포함해야 한다.

James[5]는 스마트 홈 환경과 IoT 서비스의 컴퓨팅 리소스, 공격 포인트, 통신 인프라 및 공격 비율이 증가함에 따라 상당한 보안 문제가 야기될 수 있음을 주장하면서 IoT 사이버 공격을 분석하고 침입 차단 시스템 구축 방법론을 제시하였다. 첫째, IoT 기기는 도청 공격을 당하고 있다고 인지하기 어렵기 때문에 데이터를 암호화 하여 기밀성을 확보해야 한다. 둘째, IoT 공격은 무차별 대입 공격으로 로그인 자격 증명을 시도하기 때문에 강력한 암호 정책을 적용하여 인증을 강화해야 한다. 셋째, IoT 공격의 목적은 DoS(Denial of Service)가 주요한 부분이기 때문에 접근 제어를 통하여 잘못된 패킷 요청을 하지 않도록 막아야 한다.

Alrawi[6]는 IoT 악성코드 생애주기에 대한 연구를 진행하였으며, 특수한 IoT 환경에 맞추어 MITRE ATT&CK을 참고한 분류 체계와 전용 악성코드 분석 시스템을 제시하였다. 또한 IoT 환경에서의 악성코드 대응에 대한 두 가지 현상을 분석하였다. 첫 번째, 전통적인 악성코드와 IoT 악성코드가 무엇이 다른지 분석하였다. IoT 환경은 end-point 보안 솔루션이 부족하며 IoT 대상 악성코드는 전통적인 악성코드와 달리 변종은 많으나 뿌리가 되는 악성코드는 몇 개 없는 것을 밝혀냈으며, 다양한 환경에서도 빠르게 전파되는 특징과 지속성 확보를 위하여 패킹, P2P 통신, 시스템 콜 사용 등 보안 우회 기능을 통한 변종으로 발전해나가는 것을 확인하였다. 두 번째, 현재 IoT 악성코드가 효율적으로 차단되고 있는지 분석하였다. 결론적으로 많이 부족한 상태이며 따라서 기기 소유자, 기기 생산자, 통신사 별 IoT 위협 대응에 대한 방법론을 제시하였다.

III. IoT 특성 분석

먼저 일반 컴퓨팅 환경과 IoT 환경은 많이 다르기 때문에 전통적인 방법으로는 IoT의 위협에 대응할 수 없다. IoT 환경은 지극히 제한적이며 악성코드는 다양한 아키텍처를 지원하면서 임베디드 리눅스에 최적화 되어 있다. 따라서 IoT 전용 악성코드 탐지 시스템을 설계하기에 앞서 IoT 환경과 악성코드에 대한 특성을 먼저 파악할 필요가 있다.

3.1 IoT 환경 특성 분석

IoT 기기는 연동을 통한 기기 간 동작과 데이터 송수신을 자동화하여 사용자의 편의성을 증진시키고 개입을 최소화하는 방향으로 발전하고 있다. IoT 기기들은 사용자가 처음 한번 설정하면 이후로는 스케줄에 맞추어 동작하거나, 기기 간 통신을 하며 사용자와 상호작용을 거의 하지 않고 동작하는 기기가 대부분이다. 따라서 사용자는 자신이 소유한 IoT 기기가 어떤 것이 있는지 인지조차 못하고 있는 경우도 있다.

IoT 기기는 일반 컴퓨팅 환경과 다르게 임베디드 리눅스를 기반으로 구성된 제품들이 많으며 32bit와 64bit, 그리고 ARM, MIPS, m86k 등 다양한 아키텍처 기반으로 동작한다. 또한 저전력 컴퓨팅환경이기 때문에 굉장히 제한적인 리소스상에서 동작한다. 따라서 개발 환경이 제약적일 수밖에 없으며 보안 모듈 적용도 마찬가지로 제약적인 환경이다. 이러한 환경으로 인하여 현재 사용되고 있는 리눅스의 보안 솔루션으로는 해결하기 어려운 부분이 존재한다. 이는 과거 PC가 사이버 보안 위협에 대한 대응책이 마련되지 않은 상태에서 컴퓨터의 리소스와 네트워크가 빠르게 발전한 경우와 유사할 수도 있다.

IoT 사이버 보안 위협에 대한 대책 마련이 없는 것은 아니다. 전통적인 네트워크 사이버 위협 방어 솔루션 기업인 Paloalto[7]와 F5[8]에서 IoT 관련 솔루션을 내놓았지만, 현재 판매되는 장비에 IoT 전용 탐지 규칙을 적용하여 파는 수준으로 가정용으로 쓰기엔 장비가 크고 비용이 많이 든다. 그리고 End-point IoT 보안 솔루션에는 Microsoft와 Kaspersky의 제품이 있다. Microsoft[9]는 Microsoft Defender를 활용한 IoT 탐지 솔루션이지만 네트워크 이벤트 모니터링 수준이며 솔루션의 대상도 OT(Operation Technology)환경이다. Kaspersky[10]는 공유기에 침입 방지 시스템을 포함시켜서 판매하고 있지만 해당 공유기는 가격이 비싸며 이미 설치된 IoT 네트워크 환경을 변경해야 되는 부담이 있다. 이러한 IoT 보안 솔루션 제품들은 스마트 홈이나 소규모 사업장에서 사용하고 있는 IoT 기기 환경에 적용하기에는 적합하지 않다. 그리고 IoT 보안 관련 연구들은 대부분 머신러닝[11], 저전력 환경에서의 성능 측정[12], 특성 분류[13] 위주의 연구이다.

따라서 가장 많이 침해를 받고 있는 개인 및 소규모 사업장용 IoT 기기들은 여전히 위협에 노출되어 있다. 사용자들은 IoT 기기에 존재하는 민감 정보나 자산이 위협에 노출되어 있다는 인식이 부족하다. 그리고 관련 보안 솔루션들을 적용해야 될 환경이 다양한 아키텍처이기 때문에 대응하기에 까다롭고, 비용 측면도 고려해야 한다. 따라서 실제 IoT 기기들이 침해를 받고 악용되고 있지만 대응에 어려움을 겪고 있으며 제대로 대응되고 있지 않다.

3.2 IoT 악성코드 특성 분석

Alrawi[6]에 따르면 IoT 악성코드는 다양한 코드가 변종으로 발전하는 전통적인 악성코드와 다르게 mirai, qbot, tsunami 등 소수의 코드가 뿌리가 되어 변종으로 발전해왔다. 따라서 변종으로 발전된 IoT 악성코드 간 코드 유사도가 높을 것으로 가정하여 분석하였다.

3.2.1 IoT 악성코드 수집 및 분석 방법론

IoT 관련 악성코드 수집은 과거의 IoT 악성코드와 현재 유포되는 악성코드들을 virustotal[14], virusshare[15], triage[16], urlhaus[17] 등 다양한 채널을 통하여 IoT 관련 탐지명 및 태그를 기준으로 수집하였다. 수집한 IoT 악성코드 아키텍처와 그 수는 Table 1과 같다.

Table 1. Architecture of collected IoT malware

architecture	malware number
arm	19426
mips	9893

다양한 채널에서 악성코드 샘플이 수집되어 탐지명이 명확하지 않은 샘플도 존재하기 때문에 해당 샘플들은 virustotal[14]에서 탐지명을 파악했으며 Table 2와 같다. 그 결과 IoT 악성코드 유포와 감염이 많은 순서대로 mirai계열 악성코드가 가장 많은 비율을 차지하고 있었으며, 그 다음으로 qbot계열 악성코드가 많다. 이는 Fig.4와 같이 IoT 악성코드가 2022년도에 Mirai와 qbot 변형 악성코드가 가장 많이 유포된 것을 확인할 수 있으며, 수집된 악성코드의 비율과 실제 유포되는 비율이 비슷한 것을 확인할 수 있다[18]. 그리고 mirai의 변종으로 mozi, moobot, satori, wicked, hajime, enemybot 등 다양한 변종이 존재하며, qbot 역시 gafgyt, sakura, enemybot, rapperbot, mozi, vbot, echobot 등 변종이 다양하다. 하지만 virustotal에서는 변종 샘플들도 하나의 탐지명으로

Table 2. Family of collected IoT malware

family name	number
mirai family	18178
qbot family	9675
tsunami family	586
etc	880

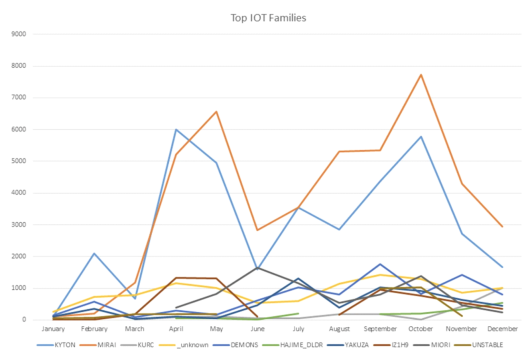

Fig. 4. Top IoT Families by month in 2022

Similarity	Confid	Change	EA Primary	Name Primary	EA Secondary	Name Secondary	Com	Algorithm	Match	Basic	Basic I	Match	Instruc	Instruc	Match	Edges
1.00	0.99	-------	0000F1F0	strcoll	0001B630	strcoll		Name Hash	2	2	2	7	7	7	2	2
0.98	0.98	-I-----	00012B18	__divsi3	00014B04	__divsi3		Name Hash	14	14	14	70	74	75	16	16
0.93	0.94	-I-JE--	0001013C	random_r	00017638	random_r		Name Hash	6	6	6	30	35	35	7	7
0.90	0.92	-I--E--	000110FC	sbrk	000196CC	sbrk		Name Hash	6	6	6	18	21	25	9	9
0.87	0.98	GI-----	00012A64	strspn	0001B920	strspn		Name Hash	7	8	7	18	20	19	7	11
0.83	0.89	GI--E--	00010298	initstate_r	000177A0	initstate_r		Name Hash	7	7	8	41	49	58	8	8
0.78	0.96	GI--EL-	0000F298	strcpy	000156B0	strcpy		Name Hash	3	3	4	4	7	9	2	3
0.73	0.90	GI--EL-	0000DF70	_stdio_term	0001A584	_stdio_term		Name Hash	9	9	12	25	42	82	9	11
0.69	0.97	-I--E-C	0000FEDC	setstate	0001741C	setstate		Name Hash	1	1	1	26	29	39	0	0
0.68	0.97	-I--E--	0000F7B8	inet_addr	000156D4	inet_addr		Name Hash	1	1	1	7	9	10	0	0
0.67	0.97	-I--E--	00010B60	__pthread_return_0	00018C24	__pthread_return_0		Name Hash	1	1	1	1	2	2	0	0
0.65	0.97	-I--E--	0000DCCC	__errno_location	00015598	__errno_location		Name Hash	1	1	1	1	2	7	0	0
0.64	0.79	GI--E-C	0001007C	setstate_r	00017898	setstate_r		Name Hash	5	5	6	34	47	56	3	5
0.60	0.94	GI--E--	0000F7DC	connect	0001583C	connect		Name Hash	3	3	6	3	11	42	1	3
0.58	0.95	GI--E--	00010E4C	sigaction	00019224	sigaction		Name Hash	4	8	4	18	55	31	1	11
0.53	0.78	GI--E--	00010F30	__default_sa_rest...	000192B8	__default_sa_restorer		Name Hash	3	5	3	3	13	8	1	5

Fig. 5. IoT Malware Binary Similarity Analysis

나오기 때문에 하나의 탐지명 내부에 다수의 변종이 존재한다.

수집된 샘플들은 Fig.5와 같이 bindiff[19]와 ida pro[20]를 사용하여 코드 간 바이너리 유사도를 분석하였으며 여러 샘플 간 유사도를 측정하기 위하여 이러한 비교 작업을 자동화 하였다.

IoT 악성코드의 유사도 비교 분석을 통하여 패턴을 찾아낼 때 다음과 같은 분석 대상을 확보하면 더욱 쉽게 패턴을 얻어낼 수 있다.

첫째, 탐지명이 확실한 샘플을 확보하는 것이다.

탐지명이 확실하면 어떤 악성코드에서 변종으로 진화되었는지 확인하기 쉬우며, 유출된 악성코드 소스 코드를 참조할 수도 있다.

둘째, 공격자가 다양한 아키텍처를 대상으로 배포하고 있는 샘플을 확보하는 것이다. 이는 하나의 소스코드를 다양한 아키텍처로 크로스 컴파일 했다는 사실이 명확하기 때문에 코드의 구조를 분석할 때 타 아키텍처를 참조하여 상호 보완이 가능하다. 또한, 패킹이나 strip 옵션을 사용하여 컴파일 한 경우 분석하기 어려운데, 하나의 아키텍처라도 위의 사항들이 적용되어 있지 않다면 참고하여 분석하여 패턴을 생성할 수 있다.

3.2.2 IoT 악성코드 바이너리 유사도 상세 분석

먼저 탐지명이 명확한 IoT 악성코드를 각각 선별하여 코드를 분석하였으며, 바이너리 유사도 비교를 진행하였다.

분석 결과 enemybot과 sakura 악성코드에서 Fig.6과 같이 유사도가 높은 rand_cmwc 함수를 찾아낼 수 있었다. 그리고 이 함수들을 Fig.8과 같이 어셈블리를 통하여 코드의 유사도를 검증하였다. 이 함수는 IoT 악성코드의 SendUDP, MakeRandomStr, MakeIPPacket 함수에 쓰이고 있었으며 이를 통해 네트워크 통신할 때 쓰이는 함수라는 것을 파악할 수 있었다. 이러한 과정에서

```
1  int rand_cmwc()
2  {
3    int v1; // [sp+0h] [bp-2Ch]
4    int v2; // [sp+14h] [bp-18h]
5
6    v1 = c + 18782 * Q;
7    c = (c + 18782 * Q) >> 32;
8    v2 = v1 + c;
9    if ( v1 + c < c )
10   {
11     ++v2;
12     ++c;
13   }
14   Q = -2 - v2;
15   return -2 - v2;
16 }
```

```
1  int rand_cmwc()
2  {
3    int v0; // r1
4    int v2; // [sp+0h] [bp-28h]
5    int v3; // [sp+10h] [bp-18h]
6
7    i_4126 = (i_4126 + 1) & 0xFFF;
8    v2 = c + 18782 * Q[i_4126];
9    c = (c + 18782 * Q[i_4126]) >> 32;
10   v3 = v2 + c;
11   if ( v2 + c < c )
12   {
13     ++v3;
14     ++c;
15   }
16   v0 = i_4126;
17   Q[i_4126] = -2 - v3;
18   return Q[v0];
19 }
```

Fig. 6. Comparison of enemybot(left) and sakura(right) rand_cmwc decompile function

github.com/geniosa/qbot/blob/master/client.c

```
75  uint32_t rand_cmwc(void)
76  {
77          uint64_t t, a = 18782LL;
78          static uint32_t i = 4095;
79          uint32_t x, r = 0xfffffffe;
80          i = (i + 1) & 4095;
81          t = a * Q[i] + c;
82          c = (uint32_t)(t >> 32);
83          x = t + c;
84          if (x < c) {
85                  x++;
86                  c++;
87          }
88          return (Q[i] = r - x);
89  }
```

Fig. 7. qbot rand_cmwc function

Fig.7과 같이 해당 악성코드들의 뿌리가 될 수 있는 qbot의 공개되어 있는 소스코드[21]를 참고하여 해당 함수의 의미를 빠르게 파악할 수 있었다.

또한 mirai 악성코드와 탐지명을 알 수 없는 악성코드에 대하여 바이너리 유사도를 비교 분석하여 유사도가 가장 높은 checksum_tcpudp 함수를 도출할 수 있었다. 공개된 mirai 소스코드[22]를 참고하여 의미 파악과 함수의 사용하는 부분에 대한 분석을 진행하였고, 어셈블리를 통하여 유사도에 대해 검증을 하였다.

IoT 악성코드 바이너리 유사도 비교 분석 및 검증을 통하여 변종 악성코드 간 일치하는 패턴을 찾을 수 있었다. 본 논문에서 다루지 않은 악성코드도 이와 같이 바이너리 유사도 비교 분석을 통하여 다수의 악성코드를 탐지할 수 있는 패턴을 손쉽게 도출할 수 있다. 이를 활용하여 침입 방지 시스템에 snort 탐지 규칙을 적용하여 탐지하면 소량의 탐지 규칙만으로도 넓은 탐지 커버리지를 확보할 수 있다.

3.2.3 IoT 악성코드 바이너리 유사도 분석 결과

패턴 탐지 실험에 사용된 악성코드는 가장 많이 유포되어 감염되고 있는 ARM과 MIPS 아키텍처 기반 악성코드 총 29,319개를 수집하였다. 아키텍처별 악성코드를 샘플링하여 바이너리 유사도 분석을 진행하고 공통된 패턴을 수집하여 탐지 테스트를 진행하였다.

탐지에 사용된 snort 규칙은 Table 4와 같다. IoT 악성코드는 http 프로토콜과 ftp 프로토콜을 모두 이용하여 유포되기 때문에 해당 규칙을 snort에서 tcp, udp 에 모두 적용시켜서 사용하였다.

탐지 결과 Table 3과 같이 ARM 아키텍처는 88%, MIPS 아키텍처는 91%의 탐지율을 보였으며, 사용된 snort 규칙은 Table 4와 같이 ARM

Table 3. IoT Malware Detection Result

type	total	detection	ratio
arm	19426	17028	88%
mips	9893	9029	91%

Table 4. IoT Malware Detection Rules

type	name	rule
arm	rand_cmwc	content:"|49 3C A0 E3 5E 30 83 E2 00 40 A0 E3|"; content:"|0B E5|"; distance:2; within:3; content:"|0B E5|"; distance:2; within:3;
	checksum_tcpudp	content:"|20 28 B0 E1 04 00 00 0A 00 38 A0 E1 23 38 A0 E1 02 00 83 E0 20 28 B0 E1|";
mips	rand_cmwc_mipsb_little_endian	content:"|03 A0 F0 21 24 03 49 5E 00 00 10 21|";
	rand_cmwc_mipsl_big_endian	content:"|21 F0 A0 03 5E 49 02 24 21 18 00 00|";
	checksum_tcpudp_mipsb	content:"|94 A2 00 00 24 E7 FF FE 28 E3 00 02|"; content:"|21 10 60 FF FB 24 A5 00 02|"; distance:3; within:10;
	checksum_tcpudp_mipsl	content:"|00 00 A2 94 FE FF E7 24 02 00 E3 28 21|"; content:"|FB FF 60 10 02 00 A5 24|"; distance:3; within:9;

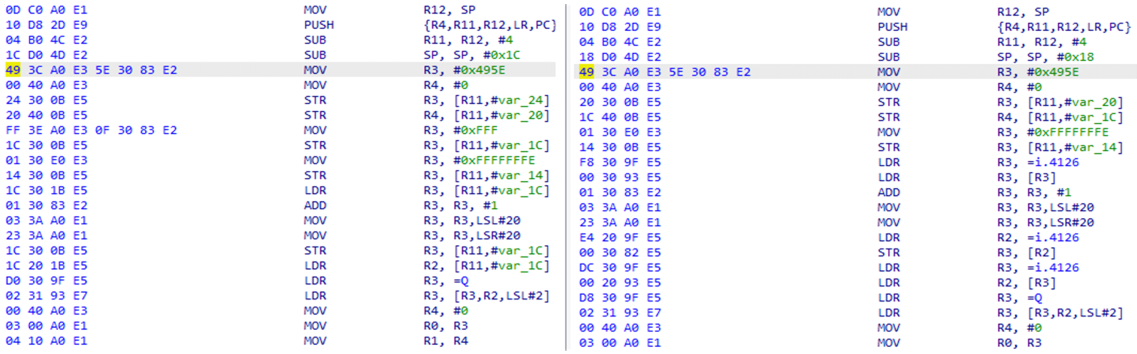

Fig. 8. Comparison of enemybot(left) and sakura(right) rand_cmwc assembly code

아키텍처는 2개, MIPS 아키텍처는 4개를 사용하였다. 이 결과 IoT 악성코드는 소수의 코드가 발전되어 변종이 됨을 확인할 수 있었으며, 따라서 바이너리 유사도 분석을 통하여 패턴을 생성한다면 소량의 패턴을 가지고 상당히 많은 IoT 악성코드를 탐지할 수 있다.

탐지되지 않은 악성코드에 대하여 분석했을 때 유형은 다음과 같았다.

첫째, 임베디드 리눅스 기반 IoT 악성코드가 아닌 안드로이드 JNI(Java Native Interface)에 사용되는 so파일로 구성된 라이브러리 악성코드 이다. 둘째, 과거 리눅스 IRC 봇 악성코드에서 IoT DDoS 악성코드로 발전된 Kaiten 및 Tsunami 악성코드이며 이 악성코드는 바이너리 패턴이 너무 세분화 되어 다량의 탐지를 할 수 있는 공통 패턴을 찾을 수 없었다. 셋째, UPX 패커를 수정하여 사용하는 mirai 악성코드로 해당 샘플은 일반적인 방식으로 실행 압축을 하지 않았기 때문에 공통 패턴을 찾을 수 없었다.

그러나 qbot과 mirai 악성코드가 arm과 mips 아키텍처에서 각각 88%, 91%의 탐지 결과가 나온 것을 보았을 때, IoT 전용 악성코드는 여전히 뿌리가 되는 악성코드의 소스를 차용하고 있는 것을 확인할 수 있다. 따라서 이 사실을 활용하여 도출한 바이너리 유사도 분석을 통한 공통 패턴 탐지 규칙은 유의미한 결과라고 할 수 있다.

이와 같이 바이너리 유사도 비교 분석을 통한 탐지 패턴을 생성한다면, IoT 악성코드 변종이 많이 발전하여 위에 제시된 snort 규칙에 탐지되지 않는다 하더라도 보다 쉽고 빠르게 대응할 수 있을 것이다.

IV. 악성코드 탐지 시스템 설계

IoT 악성코드는 현재까지도 사이버 위협이 지속되고 있으며 기능이 발전하여 진화하면 더 큰 위협으로 다가 올 수 있다. 현재 문제되는 IoT 악성코드에 감염된 기기는 가정용이나 소규모 사업장에서 사용하는 IoT 기기가 대부분이다. 또한 많은 연구가 진행되어 발전되고 있는 스마트 시티 및 스마트 공장에도 향후 악성코드의 위협이 있을 수 있다. 하지만 아직 IoT 악성코드 감염의 위협에 대한 대응은 미흡한 수준이다. 따라서 IoT 환경을 고려한 악성코드 탐지 시스템을 적용하여 위협을 예방해야 한다.

4.1 설계 방안

IoT 환경과 악성코드에 대한 특성 분석 결과 특징적인 부분이 많으며 전통적인 보안 관점으로 해결할 수 없는 부분도 존재한다. 따라서 IoT 전용 악성코드 탐지 시스템을 설계하기 위하여 다음과 같은 사항들을 고려하였다.

첫째, 악성코드 탐지 시스템은 패킷 기반 탐지 방식으로 고려해야 한다. IoT 기기는 아키텍처가 다양하기 때문에 모든 End-point를 고려하여 대응하기 어렵다. 따라서 Agent 기반보다 Network 기반 탐지 대응 방식을 사용하면 악성코드 탐지 시스템을 구성하고 관리하기 쉬우며 빠른 대응이 가능하다.

둘째, 저전력 기기를 활용해야 하며 구성할 때 가격이 저렴해야 한다. IoT는 저전력 환경에서 설치되고 동작하는 경우가 많기 때문에 저전력 환경에 적용할 수 있어야 한다. 그리고 IoT 네트워크를 구성할 때 많은 기기가 사용되며 네트워크 구성 또한 다양하게 구성되기 때문에 많은 영역에 설치하기 위하여 소

비 전력과 탐지 장비의 가격 경쟁력을 갖추어야 한다.

셋째, 트래픽 미러링을 활용하여야 한다. 현재 다수의 IoT 기기가 공유기를 활용하여 설치되어 있으며, 이 공유기의 트래픽 미러링 기능을 활용하면 기존의 네트워크의 구조를 크게 변경하지 않고 악성코드 탐지 시스템을 적용할 수 있다. 또한, 탐지 시스템을 인라인으로 적용하여 트래픽 부하 및 탐지 시스템 이상으로 인한 네트워크 장애를 피할 수 있으며, 탐지 시 이상 행위를 인지하지 못한 IoT 기기 사용자에게 알람을 발생하여 악성코드 침입에 대한 위협을 효율적으로 대응할 수 있다 .

최근 IoT 기기들이 클라우드와 연결되면서 컴퓨팅 리소스를 클라우드에서 해결하고 있기 때문에 IoT 악성코드 탐지도 클라우드에서 할 수 있는 연구가 진행되고 있다[23]. 해당 연구에서 클라우드를 이용할 수 있는 환경은 엣지 컴퓨팅 환경과 IoT 서비스가 클라우드와 연동되어야 하는 환경이 필요하다. 하지만 본 논문에서 다루는 현재 침해사고가 일어나고 있는 IoT 기기들은 이미 설치가 되어 있는 상태이며 클라우드가 연동되어 있지 않은 구형 장비도 많다. 소규모 네트워크에 기존에 설치된 IoT 장비를 클라우드 연동이 가능한 장비로 교체하는 비용이 많이 들고 물품관리에 지표가 될 수 있는 내용연수에서도 IoT 관련 기기들은 최소 6년에서 9년까지 사용하도록 고시되어 있기 때문에 구형 장비들이 바로 교체되기 어렵다[24]. 그러므로 최대한 기존 네트워크 구조를 변경하지 않으면서 저비용으로 구축 가능한 악성코드 탐지 시스템을 설계하였다. 따라서 본 연구에서는 클라우드를 활용한 악성코드 탐지 시스템에 대해서는 다루지 않는다.

4.2 구조 및 구현

현재 IoT 악성코드 감염 대상이 되는 대부분의 기기들은 Fig.9와 같이 공유기 하단에 설치되어 있다. IoT 전용 악성코드 탐지 시스템은 패킷이 들어오고 나가는 공유기에서 트래픽을 미러링한 후 해당 트래픽을 분석하도록 구성한다. 트래픽 미러링은 공유기 자체에서 지원하는 기능을 활용하거나 공유기에 설치되어 있는 리눅스의 iptables에 있는 TEE 모듈을 사용하면 된다. 이와 같은 방법을 활용하여 현재 IoT 기기들의 제품 변경이나 설치된 네트워크를 변경하지 않고 구성이 가능하다.

IoT 전용 악성코드 탐지 시스템은 Raspberry

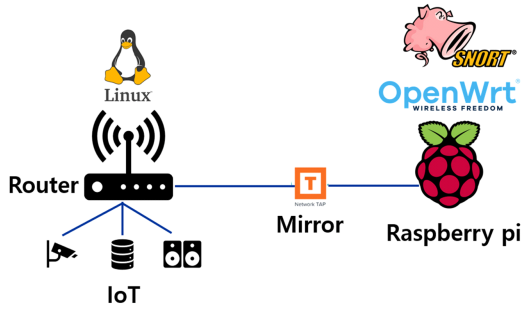

Fig. 9. Things Malware Detection System Structure

pi 4 B로 구성하였다. 해당 기기는 35달러에서 75달러로 가격이 비교적 저렴하며, 네트워크 포트가 Gigabit 통신을 지원하여 트래픽 부하에 잘 견딜 수 있다.

운영체제로는 OpenWrt[25]를 사용하였는데 오픈소스 방화벽으로 유명한 pfsense는 ARM 아키텍처를 지원하지 않기 때문에 ARM 아키텍처를 지원하는 오픈소스 방화벽 운영체제 중 유지보수, 성능, 사용성이 뛰어나 가장 많은 사용자가 활용하고 있는 OpenWrt를 적용하였고, snort 패키지를 설치하여 범용적인 네트워크 악성코드 탐지 규칙을 사용할 수 있도록 하였다.

4.3 탐지 성능 실험 및 결과 비교 분석

본 논문에서 구성한 IoT 환경 전용 악성코드 탐지 시스템의 성능을 비교하기 위하여 판매되고 있는 netgate 2100 BASE pfsense+ security gateway 제품[26]을 사용하였다. Table 5와 같이 제품을 비교하였을 때, netgate 제품이 네트워크 성능이 더 좋고 소비전력이 높으며 가격이 약 최대 10배까지 차이가 나는 것을 확인하였다. 해당 제품은 4포트로 구성되어 있으며 가정용으로 사용하기 적합하다. 또한 제품 내부에 오픈 소스 방화벽으로

Table 5. Comparison of Malware Detection System Device

	netgate 2100	raspberry pi 4B
Power	24W	15W
Network benchmark	964Mbps	725Mbps
Price	$349	$35~$75

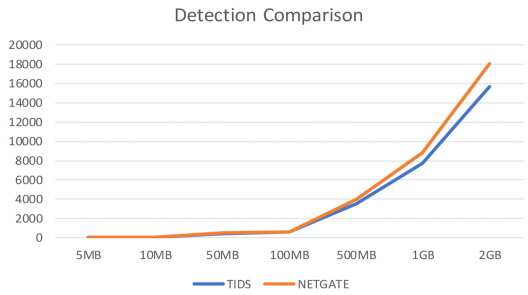

Fig. 10. IoT Malware Detection Comparision

유명한 pfsense가 설치되어 있다. 해당 솔루션에서 snort 패키지를 설치하고 실험을 진행하였다.

탐지 테스트는 IoT 악성코드의 유포 방식과 동일하게 http 프로토콜을 이용하여 수집한 ARM 아키텍처와 MIPS 아키텍처 악성코드 파일을 다운로드하는 방식으로 진행하였다.

전송 용량을 변경하며 실험을 진행했을 때 Fig.10과 같이 총 전송 용량이 많아질수록 탐지의 차이가 생기기 시작했다. 하지만 IoT 환경에서는 트래픽이 많은 대역폭을 차지하지 않으며 낮은 대역폭에서 탐지의 차이가 10%내외의 차이를 보였다. 이는 탐지 시스템에 들어오는 패킷을 필터링하여 선별적으로 패킷을 전달한다면 물리적인 성능의 차이를 극복할 수 있을 것이다.

또한 비교에 사용한 netgate 제품은 현재 349달러인데 반해, 본 논문에서 제시하는 IoT 악성코드 탐지 시스템의 라즈베리파이는 소비 전력이 더 낮으며 최소 35달러로 구성 가능하여 가격 차이가 약 10배가량 차이난다. 따라서 본 논문에서 제시하는 악성코드 탐지 시스템은 IoT 환경에 적합하다고 할 수 있다.

V. 결 론

앞으로 IoT 기기가 점점 더 증가할 것으로 예상되고 있으며 그에 따른 사이버 위협도 증가할 것이다. IoT 환경은 사용자가 최대한 관리하지 않도록 설계되어 있어 기기의 이상 행위를 인지하기 힘들고, 다양한 설치 환경과 아키텍처로 인하여 사이버 위협이 발생하면 대응하기 어려운 상황이다.

따라서 본 논문에서는 IoT 특성에 대하여 연구하여 IoT 환경 및 악성코드에 대하여 분석하였으며, 이를 통하여 IoT 전용 악성코드 탐지 시스템을 구현하였다. 이에 대한 기대 효과는 다음과 같다.

첫째, 저렴하고 구성하기 쉬운 소규모 네트워크 IoT 전용 방어 시스템을 적용할 수 있다. 일반 가정 및 소규모 사업장에서는 비싼 장비를 쓸 수 없고 현재 시판되어 있는 제품들의 가격은 너무 높다. 본 논문에서 제시한 시스템을 구성한다면 적은 비용으로 IoT 사이버 위협을 예방할 수 있다.

둘째, 소량의 탐지 규칙으로 다수의 악성코드를 탐지할 수 있다. 본 논문에서 제시한 IoT 악성코드의 특성을 고려한 분석 방법론을 활용하여 패턴을 생성한다면 보다 효율적으로 탐지가 가능하다. 또한 탐지 규칙을 원격에서 전달할 수 있다면 현재 해결되지 않고 있는 IoT 사이버 위협에 대하여 빠르게 대응하여 해결할 수 있다.

본 논문에서 제시한 바이너리 유사도를 통한 분석 방법론으로 공통 패턴을 생성할 수 없는 악성코드가 존재한다. 이러한 악성코드들은 향후 연구를 통하여 탐지 규칙을 세분화하고 탐지 규칙 간 상관관계 분석을 통한 공통 패턴을 도출할 수 있을 것이다. 또한 IoT 환경 전용 악성코드 탐지 시스템은 대역폭이 높은 환경에서 성능이 저하되는 현상이 존재한다. 이는 향후 연구를 통하여 프로토콜 및 패킷 필터를 적용한다면 대역폭이 큰 트래픽 환경에서도 높은 성능을 기대해 볼 수 있을 것이다.

References

[1] Exploding Topics, “80+ Amazing IoT Statistics(2023-2030)”, https://explodingtopics.com/blog/iot-stats, Nov. 2022

[2] urlhaus statistics, “Statistics - URLhaus - Abuse.ch”, https://urlhaus.abuse.ch/statistics/, Feb. 2023

[3] shadowserver, “IoT device statistics Time series”, https://dashboard.shadowserver.org/statistics/iot-devices/time-series/?date_range=other&d1=2021-02-07&d2=2023-02-08&type=video-system&group_by=type&style=stacked, Feb. 2023

[4] Khraisat, Ansam, and Ammar Alazab, “A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, vali-

dation strategy, attacks, public datasets and challenges." Cybersecurity, vol. 4, pp. 1-27, Dec. 2021.

[5] James, Fathima. "IoT cybersecurity based smart home intrusion prevention system." 2019 3rd Cyber Security in Networking Conference (CSNet). pp. 107-113, Oct. 2019.

[6] Alrawi, Omar, et al. "The Circle Of Life: A Large-Scale Study of The IoT Malware Lifecycle." USENIX Security Symposium 2021. pp. 3505-3522, Aug. 2021.

[7] Paloalto, "Paloalto IoT Security Solution", https://docs.paloaltonetworks.com/iot/iot-security-admin/iot-security-solution, Feb. 2023

[8] F5, "Securing the Number One Attack Target on the Internet: IoT Devices", https://www.f5.com/pdf/solution-center/f5-network-security-for-iot.pdf, Feb. 2023

[9] Microsoft Defender for IoT, "System architecture for OT system monitoring", https://learn.microsoft.com/en-us/azure/defender-for-iot/organizations/architecture, Feb. 2023

[10] Kaspersky, "Kaspersky IoT Infrastructure Security", https://os.kaspersky.com/solutions/kaspersky-iot-infrastructure-security/, Feb. 2023

[11] NGUYEN, Xuan-Ha, et al. "Realguard: A lightweight network intrusion detection system for IoT gateways", Sensors, no. 2, pp. 432, Jan. 2022

[12] ZITTA, Tomas, et al. "Experimental load test statistics for the selected IPS tools on low-performance IoT devices." Journal of Electrical Engineering, no. 4, pp. 285-294, Aug 2019

[13] KHRAISAT, "Ansam: ALAZAB, Ammar. A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges.", Cybersecurity, no 4, pp. 1-27, Dec 2021

[14] VirusTotal, "Virus Total", https://virustotal.com/, Feb. 2023

[15] Virusshare, "Forensics, C.: Virusshare", https://virusshare.com/, Feb. 2023

[16] triage, "Recorded Future Triage", https://tria.ge/, Feb. 2023

[17] urlhaus, "URLHaus | Malware URL exchange", https://urlhaus.abuse.ch/, Feb. 2023

[18] fortinet, "2022 IoT threat review", https://www.fortinet.com/blog/threat-research/2022-iot-threat-review, Feb. 2023

[19] Bindiff, "Zynamics BinDiff", https://www.zynamics.com/bindiff.html, Feb. 2023

[20] ida pro, "IDA Pro - Hex Rays", https://hex-rays.com/ida-pro/, Feb. 2023

[21] qbot source code, "leaked qbot source code", https://github.com/geniosa/qbot/blob/master/client.c, Feb. 2023

[22] mirai source code, "leaked mirai source code", https://github.com/jgamblin/Mirai-Source-Code/blob/master/mirai/bot/checksum.c, Feb. 2023

[23] Syed, Naeem Firdous, Mengmeng Ge, and Zubair Baig. "Fog-cloud based intrusion detection system using Recurrent Neural Networks and feature selection for IoT networks.". Computer Networks, vol. 225, pp. 109662, Apr. 2023

[24] Korean Law information Center, "Administrative ruels useful life", https://www.law.go.kr/%ED%96%89%EC%A0%95%EA%B7%9C%EC%B9%99/%EB%82%B4%EC%9A%A9%EC%97%B0%EC%88%98/(2018-14,20180927), Feb. 2023

[25] openwrt, "Open Wireless RouTer",

https://openwrt.org/, Feb. 2023
〔26〕 netgate, "NETGATE 4100 BASE PFSENSE+ SECURITY GATEWAY", https://shop.netgate.com/products/4100-base-pfsense, Feb. 2023

〈저자소개〉

신 상 윤 (Sangyoon Shin) 정회원
2014년 2월: 홍익대학교 컴퓨터정보통신공학과, 산업공학과 졸업
2021년 2월~현재: 고려대학교 정보보호대학원 정보보호학과 석사과정
〈관심분야〉 정보보호, 악성코드, 취약점분석

이 다 희 (Dahee Lee) 정회원
2018년 2월: 세종대학교 정보보호학 학사 졸업
2020년 12월~현재: SK쉴더스 책임연구원
〈관심분야〉 정보보호, IoT보안

이 상 진 (Sangjin Lee) 종신회원
1989년 10월~1999년 2월: ETRI 선임 연구원
1999년 3월~현재: 고려대학교 교수
2008년 3월~현재: 고려대학교 디지털포렌식연구센터 센터장
〈관심분야〉 디지털포렌식