Journal of the Korea Convergence Society
Vol. 12. No. 5, pp. 9-16, 2021

ISSN 2233-4890 / eISSN 2713-6353
https://doi.org/10.15207/JKCS.2021.12.5.009

IoT 기기 보안을 위한 해시 기반의 SSDP

김효종[1], 한군희[2], 신승수[3*]
[1]동명대학교 컴퓨터미디어공학과 학생, [2]백석대학교 컴퓨터공학부 교수, [3]동명대학교 소프트웨어융합보안학과 교수

Hash-based SSDP for IoT Device Security

Hyo-Jong Kim[1], Kun-Hee Han[2], Seung-Soo Shin[3*]
[1]Student, Dept. of Computers & Media Engineering, Tongmyong University
[2]Professor, Division of Computer Engineering, Baekseok University
[3]Professor, Dept. of Software Convergence Security, Tongmyong University

요 약 전 세계적으로 COVID-19의 감염병이 장기화됨에 따라 재택근무 시 취약한 사물인터넷(IoT) 기기에 대한 네트워크 공격으로 인해 각종 보안 위협이 있다. 초기에는 사물인터넷(IoT) 기기의 사용자를 대상으로 RDP(Remote Desktop Protocol)의 취약점을 악용하고 스피어 피싱, APT 공격 등이 주로 이루어졌다. 이후 네트워크 공격의 기술이 점차 발전하여 사물인터넷 기기의 단순서비스검색프로토콜(SSDP)을 악용하여 DRDoS 공격이 지속적으로 증가하고 있다. SSDP의 인증절차의 문제점을 보완하기 위해 Notify 메시지와 M-Search 메시지 패킷에 서버 고유정보를 해시로 암호화하고 인증필드를 추가하여 인가된 IoT 기기의 여부를 판별하는 해시 기반의 SSDP을 제안한다. 해시 기반의 SSDP을 활용하면 추후 기하급수적으로 증가할 다양한 IoT 기기에 대한 정보 노출을 방지하고 증폭 공격을 사전에 차단할 것으로 기대된다.

주제어 : 사물인터넷, 단순서비스검색프로토콜, 분산서비스거부공격, 분산반사서비스거부공격, 중간자공격

Abstract Due to the prolonged infectious disease of COVID-19 worldwide, there are various security threats due to network attacks on Internet of Things devices that are vulnerable to telecommuting. Initially, users of Internet of Things devices were exploited for vulnerabilities in Remote Desktop Protocol , spear phishing and APT attacks. Since then, the technology of network attacks has gradually evolved, exploiting the simple service discovery protocol of Internet of Things devices, and DRDoS attacks have continued to increase. Existing SSDPs are accessible to unauthorized devices on the network, resulting in problems with information disclosure and amplification attacks on SSDP servers. To compensate for the problem with the authentication procedure of existing SSDPs, we propose a hash-based SSDP that encrypts server-specific information with hash and adds authentication fields to both Notify and M-Search message packets to determine whether an authorized IoT device is present.

Key Words : Internet of Things, Simple Service Discovery Protocol, Distributed Denial of Service, Distributed Reflective Denial of Service, Man in the Middle Attack

*This work was supported by the BB21+ Project in 2020
*Corresponding Author : Seung-Soo Shin(shinss@tu.ac.kr)
Received March 19, 2021 Revised May 3, 2021
Accepted May 20, 2021 Published May 28, 2021

1. 서론

정보기술과 통신기술의 발전으로 2015년부터 사물 인터넷(IoT : Internet of Things) 프로토콜을 활용한 공격사례는 지속적으로 늘어나고 있다[1]. 다양한 IoT 프로토콜 중에서 공격자들은 단순 서비스 검색 프로토콜(SSDP : Simple Service Discovery Protocol)을 활용한 분산 서비스 거부 공격(DDoS : Distributed Denial of Service)을 시도하고 사이버 대피소의 통계로 한국은 대략 100만개 정도의 공개된 SSDP 서버가 있다. 인터넷에 연결된 취약한 SSDP 서버들은 50Gb 이상의 트래픽 발생이 가능하고 공격의 위험성은 점차 증가하고 있다.

최근까지도 DDoS과 분산 반사 서비스 거부 공격(DRDoS : Distributed Reflective Denial of Service)은 보안상 이슈가 되고 있다. SSDP는 DHCP(Dynamic Host Configuration Protocol), DNS(Domain Name System)와 같은 서버 기반의 구성없이 소규모 환경에서 사용하기 위해 설계되었고, Internet Draft by Microsoft and Hewlett-Packard in 1999의 IETF 제안이 만료된 후, SSDP가 UPnP(Universal Plug and Play)프로토콜에 통합되었다[2].

SSDP의 문제점은 비인가 기기에서 IoT 서버 접근과 SSDP 서버에서 정보를 요청한 클라이언트 기기의 출발지에 대한 추가적인 검증을 하지 않아 발생되는 DRDoS 취약점이 존재한다[3]. SSDP을 활용한 증폭공격이 가능한 비인가 디바이스를 판별하고 비인가 디바이스의 요청 메시지를 필터링하여 증폭공격을 사전에 방지할 필요가 있다. SSDP에 보안필드를 추가하여 IoT 기기의 고유정보를 해시하여 서버와 추가 인증과 로그를 저장하고, SSDP에 대해 송/수신 패킷을 분석하여 증폭공격을 파악하여, 추후 증폭공격이 의심될 경우 메시지를 필터링하여 대역폭을 유지하는 해시기반의 SSDP을 제안한다.

본 논문의 구성은 다음과 같다. 2장에서 SSDP, DRDoS 증폭에 관련된 동향 분석과 SSDP의 문제점을 분석한다. 3장에서는 SSDP에 보안 필드를 적용하여 IoT 기기 보안을 위한 해시 기반의 SSDP을 시뮬레이션 하고, 4장에서는 SSDP과 IoT 기기 보안을 위한 해시 기반의 SSDP을 비교/분석한다. 그리고 5장에서 결론을 맺는다.

2. 관련 연구

본 장에서는 SSDP, 중간자공격, DRDoS 증폭공격의 연구 동향을 알아보고 기존의 SSDP 프로토콜의 문제점을 분석한다.

2.1 SSDP

1999년 미국에서 케빈 애슈턴이 만물 인터넷(IoE : Internet of Everything)개념을 처음 사용하고 기술이 발달함에 따라 IoT의 개념으로 변화되었다. IoT은 소형 디바이스, 자동차, 빌딩 등의 IT 인프라 들을 네트워크를 통해 실시간 정보를 송/수신하고 다양한 서비스를 제공한다[4]. 그리고 대부분 서비스에서 사용하는 소형 디바이스는 NFC(Near Field Communication), 블루투스와 같은 센서들로 모든 정보와 상호작용을 하고 일반 데스크탑, 서버 컴퓨터와는 다르게 제한된 하드웨어 자원을 효율적으로 사용한다[5].

SSDP는 실시간으로 기기의 정보검색이 가능하다. UDP(User Datagram Protocol) 통신을 사용하고 통신 구조는 HTTP(Hyper-Text Transfer Protocol)와 비슷하다[6]. SSDP의 통신방식은 UPnP 프로토콜을 지원하는 디바이스에서 자신의 디바이스와 서비스를 광고하기 위해 검색 기능 메시지를 멀티캐스팅 방식으로 전달하는 광고 메시지 전달방식과 검색방식인 M-search 통신의 2가지로 분류된다. SSDP 패킷의 기본구조는 IP Address, UDP, Multicast, SSDP M-Search, SSDP Notify로 구성된다. IP Address는 패킷 송수신을 위한 기기의 주소로 설정하고 Multicast의 UDP 통신으로 SSDP M-Search 또는 SSDP Notify를 전송한다.

2.2 중간자 공격과 DRDoS 증폭

네트워크 공격기법은 서버와 클라이언트가 존재할 경우, 공격자는 두 시스템 사이에서 중간자 공격(MITM : Man in the Middle)을 통해 클라이언트의 연결을 강제로 종료시키고 공격자는 클라이언트의 아이피로 변조하여 서버와 통신한다[7]. 대표적으로 Session hijacking, DDoS, DRDoS 등의 다양한 공격기법이 존재한다. DDoS 공격에서는 공격자의 IP를 숨길 수 있고 DRDoS에서는 출발지 IP를 변조하여 증폭공격을 진행한다[8,9].

또한 중간자 공격을 기반으로 MIT PC(Man in the PC)와 MITB(Man in the Browser) 공격기법들이 있다. MIT PC는 Keystroke logging, Operating System Exploit, Memory Hacking등의 공격기법으로 대상이 정해진 공격으로 Operating System을 기반으로 한다. MITB는 브라우저 보안의 취약점을 이용하여 웹 브라우저를 감염시키고 특정 웹페이지 수정, 송/수신 데

이터 수정 등의 다양한 악의적인 행위를 할 수 있다. 악의적인 행위는 백그라운드로 진행되고 사용자와 호스트에게는 보이지 않는다[10-13].

DRDoS는 DDoS의 발전형이다. DDoS 공격에서 필요한 악성 에이전트 설치 없이 네트워크의 취약점을 이용하여 공개된 정상서버를 DRDoS의 반사서버로 사용한다[14-16]. 1999년 DRDoS의 위험성에 대해 알려지고 대표적인 사례로 2014년 대규모의 공격이 프랑스에서 발생했다[17].

2.3 SSDP 분석

SSDP의 구성은 Notify와 M-Search로 구성된다. Notify는 DHCP와 DNS등과 같은 외부서버의 도움을 받지 않고 IoT 서버에서 기기와 통신하기 위해 주기적으로 Notify라는 광고를 한다. M-Search는 기기에서 IoT 서버를 탐색하기 위해 사용하고 해당 과정을 통해 IoT 서버와 통신정보를 교환한다. 일반적인 SSDP 서버는 광고 또는 기기의 탐색요청에 의해 이벤트가 발생할 경우, 기기에 대한 인증절차 없이 통신정보를 송신한다. 따라서, SSDP의 문제점은 다음과 같다.

첫 번째, 비인가 기기에서 IoT 서버로 접근이다. 비인가 기기에서 IoT 서버로 접근하여 Location 필드를 통해 정보를 획득할 수 있다. Location 필드는 XML (Extensible Markup Language)형태로 deviceType, modelName, serialNumber, UDN 등 다양한 정보가 노출된다. 일부 보안이 적용된 IoT 기기의 경우 serialNumber는 임의로 지정한 더미값이 적용된다. 두 번째, SSDP 서버에서 정보를 요청한 클라이언트 기기의 출발지에 대한 추가적인 검증을 하지 않아 발생되는 DRDoS 취약점이 존재한다.

본 논문에서 제안하는 방식은 SSDP의 통신필드에 IoT 기기의 추가적인 인증을 할 수 있는 필드를 추가하고, SSDP 서버는 Notify를 진행할 때, 서버의 UUID (Universally Unique Identifier)의 노출을 최소화하기 위해 SHA-512 해시로 암호화를 한다. 그리고 클라이언트에서 M-Search를 진행할 때, 통신필드 중 마지막에 Device 필드를 추가하고 Device 필드의 값은 기기의 UUID를 설정하여 SSDP 서버로 송신하는 해시 기반의 SSDP이다. SSDP의 인증절차의 문제점을 보완하기 위해 해시 기반의 SSDP는 기존 SSDP에서 Notify 메시지와 M-Search 메시지 패킷에 서버 고유정보를 해시로 암호화하고 인증필드를 추가하여 인가된 IoT 기기의 여부를 판별한다.

3. 해시 기반의 SSDP

3.1 시스템의 구성 및 개발 환경

해시 기반의 시스템은 SSDP Server, Work PC, Smart Device, Media Server로 구성된다. SSDP 서버는 Raspberry Pi 3 B+와 Maria DataBase로 구성된다. 클라이언트는 일반적인 Work PC와 스마트 기기, 그리고 미디어 서버 등 여러 기기를 대상으로 한다. SSDP 서버와 클라이언트는 UDP 통신으로 비연결성을 지향하고 SSDP의 형태는 평문의 HTTP 프로토콜과 유사하다.

SSDP 서버는 일정한 주기마다 내부망을 통해 멀티캐스트로 광고를 하고 스마트 기기는 특정 플래그를 이용하여 SSDP 서버를 검색하고 통신에 대한 필요한 정보를 송/수신한다. SSDP의 구성도는 Fig. 1과 같다[18].

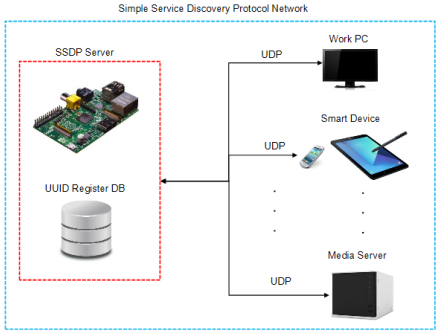

Fig. 1. SSDP Configuration

SSDP 서버를 위한 하드웨어 구성은 Rasberry Pi3 B+모델로 CPU는 1.4GHz의 Quad Core와 RAM은 1GB SDRAM(Synchronous Dynamic Random Access Memory), 그리고 네트워크 환경은 330Mbps와 502.11 b/g/n/ac, Bluetooth 4.2가 적용된다. SSDP 구성에 필요한 하드웨어 구성은 〈표-1〉와 같다.

Table 1. Hardware Configuration

Model	Rasberry Pi3 B+
CPU	1.4GHz, Quad Core
RAM	1GB SDRAM
Ethernet	330Mbps Ethernet
Wireless	502.11.b/g/n/ac, BT4.2

SSDP 서버와 Client Device를 시뮬레이션하기 위해 OS(Operation System)는 Microsoft의 Windows 10

Pro 20H2&Rasberry OS를 사용한다. Dev language는 Microsoft사의 C# language와 Ryan Dahl의 Nodejs, 그리고 Apache Web, SSH(Secure SHell), FTP(File Transfer Protocol) 프로그램을 사용한다. 시뮬레이션 구성에 필요한 소프트웨어 개발환경은 〈표-2〉와 같다.

Table 2. Software Development Environment

Operation System	Windows 10 Pro 20H2, Rasberry OS
Dev language	C#, Node JS
Other	Apache Web, SSH, FTP

3.2 시스템의 흐름도

SSDP의 흐름도는 Fig. 2에서 ①번과 같다. Fig. 2에서 ①의 과정은 다음과 같이 진행된다. 일정 시간마다 SSDP 서버에서 Notify 메시지를 멀티캐스트로 광고를 진행하고 Client Device는 M-Search 메시지를 통해 SSDP 서버를 검색한다. 검색된 SSDP 서버는 Client Device에서 요청한 메시지의 플래그에 따라 응답패킷을 구성하여 HTTP /1.1 형식으로 데이터를 송/수신한다.

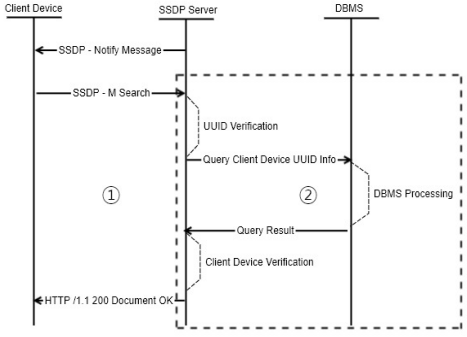

Fig. 2. Hash-based SSDP Flow Chart

SSDP에서는 인증절차가 존재하지 않으며 인가되지 않은 사용자가 무분별하게 SSDP 서버로 접근하여 악의적인 행위에 대한 위험성이 존재한다. 이러한 문제점을 해결할 방안으로 Fig. 2에 데이터베이스(DBMS : DataBase Management System)를 추가한다. Fig. 2에서 ②의 과정은 다음과 같다. Client Device의 검색요청에 따라 SSDP 서버는 해당 기기에 대한 UUID 검증단계를 진행한다. 첫 번째, M-Search 패킷의 필드 중 인증필드의 존재유무와 UUID의 형태와 일치여부를 확인한다. 두 번째, 인가된 IoT 기기 여부를 조회하기 위해 SSDP 서버에서 DBMS로 질의 요청을 하고 인가된 IoT 기기의 경우, 정상적인 데이터를 송/수신한다.

인증절차가 추가된 IoT 기기 보안을 위한 해시 기반의 SSDP를 사용하기 위해 선행 작업으로 인가된 사용자에 의해 IoT 서버 데이터베이스에 기기의 고유번호를 등록한다. 해시 기반의 SSDP에서 IoT 기기 등록은 다음과 같이 진행된다. 먼저, IoT 기기는 SSDP 서버로 3-way-handshake 과정 후 Client Device에서 SSDP 서버로 아이디와 패스워드를 HTTP/1.1 POST 형식으로 송신한다. IoT Device로부터 아이디와 패스워드를 수신 받은 SSDP 서버는 데이터 검증 후 DBMS로 질의한다.

DBMS의 인가된 사용자 정보 조회결과에 따라 로그인 세션이 발급된다. 로그인 세션을 발급받은 인가된 사용자는 기기등록 페이지로 이동하여 기기의 UUID 정보를 SSDP 서버를 통해 DBMS로 등록 작업 또는 삭제 작업을 진행한다. IoT 기기의 등록과정은 Fig. 3과 같다.

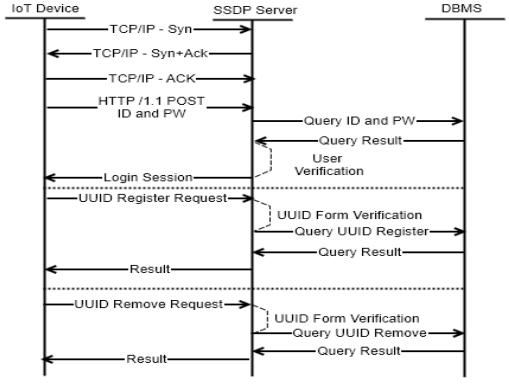

Fig 3. Registration Process of IoT Device

3.3 시스템 구조

시스템 구조는 SSDP에 보안필드를 추가한 Notify 메시지 형식과 M-Search 메시지 형식이다.

3.3.1 SSDP Notify

Notify 메시지는 SSDP 서버에서 사용하는 메시지로 클라이언트 기기와 정보교환을 위해 멀티캐스트를 통해 광고한다. 하지만 SSDP 서버의 UUID가 노출되는 취약점이 존재하여 해당필드의 부분을 해시를 통해 노출된 일정 부분을 암호화한다.

SSDP Notify의 필드부분은 Request Method, Request URL, Request Version, HOST, Cache-Control, Location, Server, NT, USN, NTS로 구성된다. Request Method, Request URL, Request Version의 필드형식은 “요청 메소드종류, 범위, 요청메소드버전”로 정의된다. HOST 필드는 서버주소, Cache-Control 필드는 최대 광고 유효시간을 max-age로 정의한다. 그리고 Location 필드는 루트 디바이스에 대한 UPnP URL 정보, 서버는 UPnP/1.1 제품/버전과 OS/버전에 대한 정보, NT는 광고 유형, USN은 기기의 고유번호, NTS는 광고 하위유형이다. 이와 같은 내용은 Fig. 4와 같다. 기존 IoT 서버의 SSDP 패킷에서 SSDP 서버의 고유정보가 노출되는 USN 필드의 취약점을 해시로 암호화하여 노출을 최소화한다.

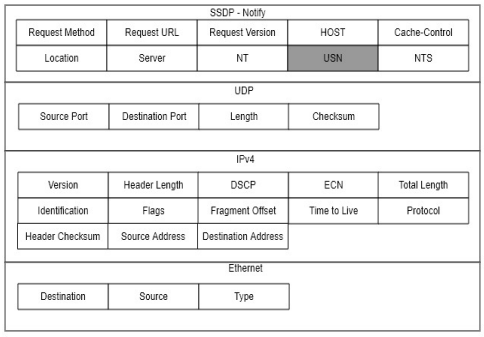

Fig. 4. SSDP-Notify

3.3.2 SSDP Search

SSDP Search 메시지의 다양한 필드 옵션 중에서 Device 필드를 추가하여 SSDP Search 메시지를 송신하는 SSDP 서버에서 Device의 UUID를 여부를 확인하고 인가되지 않을 경우, 로그를 저장하고 차단한다.

Fig. 5에서 SSDP Search의 필드부분은 Request Method, Request URL, Request Version, HOST, MAN, MX, ST, USER-AGENT, Device로 구성된다. Request Method, Request URL, Request Version의 필드형식은 요청 메소드 범위와 요청 메소드 정보, 그리고 메소드 버전으로 정의된다.

HOST 필드는 서버에 대한 주소, 그리고 MAN 필드는 HTTP Extension Framework에 필요하고 ssdp:discovery로 설정한다. MX 필드는 최대 대기시간이 포함되고 ST 필드는 검색대상을 지정한다. User-Agent 필드는 OS버전으로 설정하고 추가된 Device 필드는 IoT 기기의 고유정보를 설정한다.

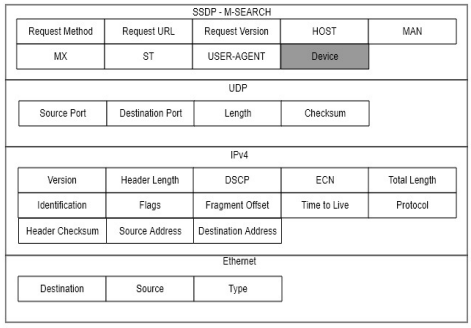

Fig. 5. SSDP - Search

기존 IoT 기기의 SSDP 패킷에서 IoT 기기에 대한 인증절차를 추가하기 위해 Device 필드를 추가한다. 이후 Device 필드에 IoT 기기의 고유정보를 해시로 암호화해서 저장한다.

3.4 시스템 시뮬레이션

해시 기반 SSDP의 시뮬레이션은 IoT 기기의 정보등록과 인증과정으로 진행된다. IoT 기기의 정보등록 과정은 세션으로 인가된 사용자를 인증하고 IoT 기기의 고유정보를 해시로 변환한다. 해시로 변환된 고유정보를 UUID 형식으로 등록 작업을 진행하고 잘못된 IoT 기기일 경우, 인가된 사용자의 권한으로 IoT 기기에 대한 정보를 삭제할 수 있다.

IoT 기기의 인증과정은 SSDP 서버의 백 엔드 단계에서 진행한다. IoT 기기로부터 정보요청 이벤트가 발생할 경우, 패킷의 Device 필드부터 검증한다. Device 필드의 검증은 데이터베이스로부터 인가된 기기의 고유번호와 일치여부를 확인한다. 비인가 디바이스일 경우, 관리자에게 알리고 로그정보를 저장한다.

3.4.1 SSDP Server Pseudocode

시스템 시뮬레이션에서 SSDP 서버는 세션으로 인가된 사용자를 인증하고 고유정보를 UUID 형식으로 등록 작업을 진행한다.

SSDP 서버의 의사코드는 변수 location, 변수 USN으로 2개의 변수가 있다. 변수 location은 디바이스 기기의 아이피 주소가 저장되고 변수 USN은 서버의 정보가 저장된다. 그리고 alive-message-wait 메소드로 디바이스 기기의 메시지 요청을 대기하고 bye-message-wait 메소드는 디바이스 기기의 연결 종료를 한다. 마지막 Server-wait에서 서버의 시작과 로그저장을 한다.

SSDP 서버의 의사코드는 Fig. 6과 같다.

```
location <- ip address
USN <- upnp:rootdevice
USN <- urn:schemas-upnp-org:device
:MediaServer:1
USN <- urn:schemas-upnp-org:service
:ConnectionManager:1
----------RETURN alive-message-wait;
advertise-alive
----------RETURN bye-message-wait;
advertise-bye
------------------Server-wait;
server-start
server-log
```

Fig. 6. Data Return

3.4.2 SSDP Device Authorization

시스템 시뮬레이션에서 SSDP 기기는 패킷의 기기 필드 설정 여부를 검증하고 데이터베이스로부터 인가된 기기의 고유번호의 존재여부를 확인한다. 만약 비인가 디바이스일 경우, 로그정보를 저장한다.

디바이스 인증은 인가된 사용자가 기기의 고유정보를 등록할 경우, 기기 인증서버에서 변환작업을 진행한다. 변환작업은 RFC 4122 규약에 따라 세 가지의 조건이 존재한다. 첫 번째로 UUID에 대한 유일성 보장으로 다른 기기와의 중복이 불가능하다. 두 번째로 일정한 규칙에 따라 128비트의 고정 크기를 가진다. 세 번째로 변경이 불가능한 값을 가진다. 인가된 사용자가 세션으로 인증한 뒤, 기기의 고유정보를 등록한 결과는 Fig. 7과 같다.

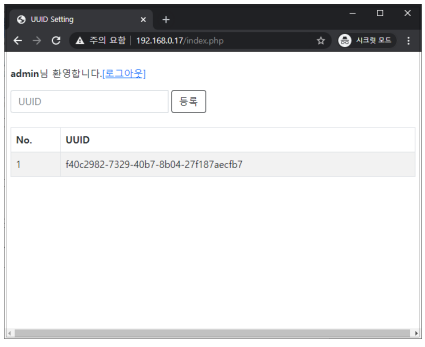

Fig. 7. UUID Result

4. 분석 및 평가

공격 트래픽을 대상으로 기기에서 SSDP 서버로 송신하는 패킷의 형태와 크기, 그리고 플래그 종류에 따라 SSDP의 취약점을 분석한다. 그리고 보안을 적용한 프로토콜과 보안을 적용하지 않은 프로토콜에 대한 증폭량을 측정하여 비교/분석하고 대응방안을 제시한다.

4.1 SSDP의 취약점 분석

SSDP은 출발지 IP에서 패킷의 특정 플래그(ST)를 ssdp:all로 설정하여 SSDP 서버로 송신한다. 수신 받은 SSDP 서버는 요청 IP의 요청에 따라 패킷을 구성하여 응답한다. SSDP의 취약점은 출발지 IP에 대한 검증이 미흡하여 악의적인 사용자가 출발지 IP를 위/변조하여 특정 플래그를 조작한 패킷을 요청할 경우, SSDP 서버는 요청 플래그에 따라 패킷을 구성하여 위/변조된 IP로 패킷을 송신한다. 짧은 시간에 많은 패킷을 수신 받은 타켓 IP는 대역폭이 소모되고 불필요한 패킷에 대한 처리로 서비스 거부가 발생한다.

출발지 IP에서 멀티캐스트 주소로 플래그 ssdp: all로 136Byte 패킷을 SSDP 서버에 송신한다. 패킷을 수신 받은 SSDP 서버 IP는 플래그 요청에 따라 서버에 대한 정보와 현재 네트워크에 대한 정보를 여러 패킷으로 나누어 송신한다. SSDP의 데이터 요청 정보는 Fig. 8과 같다.

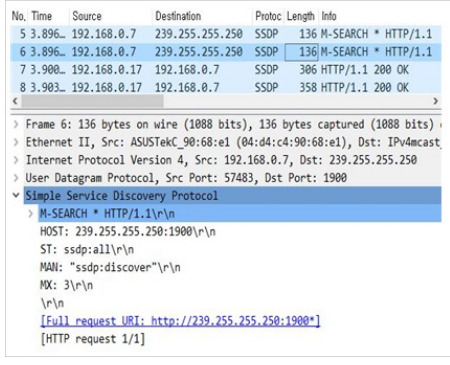

Fig. 8. SSDP Data Request

SSDP의 응답 형식은 HTTP/1.1의 형식과 유사하고 SSDP 서버로부터 정보 수신이 정상적으로 완료되었을 경우, HTTP 상태 코드를 SSDP 서버가 클라이언트로 반환한다. SSDP 서버로부터 증폭된 데이터는 여러 개의 패킷으로 분할하여 IoT 기기로 송신된다. SSDP 서버에서 IoT 기기로 증폭된 패킷의 크기는 1511Byte로 요청 대비 응답이 크다. SSDP 서버로부터 증폭된 패킷의 크기는 Fig. 9와 같다.

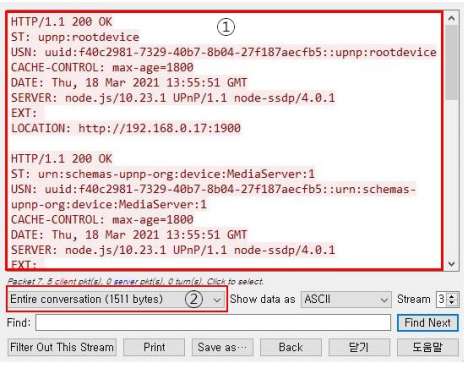

Fig. 9. Amplification result

4.2 대응방안

SSDP의 문제점은 비인가 기기에서 IoT 서버에 대한 접근으로 외부의 기기로 서버의 루트 주소로 접근하여 취약한 SSDP 서버에 대한 정보와 SSDP를 사용하는 기기에 대한 정보 등 민감한 정보가 유출된다. 그리고 IoT 기기에 대한 패킷의 출발지를 검증하지 않아 서비스 반사 취약점으로 연계되며 SSDP가 외부로부터 연결이 활성화가 되어있을 경우, 외부의 패킷요청에 SSDP 증폭공격이 가능하다.

이러한 취약점을 해결하기 위해 SSDP 서버의 고유정보를 해시로 암호화하여 노출을 최소화하고 M- Search 메시지의 추가인증 필드를 구성하여 사전에 인가된 IoT 기기만 통신이 가능하도록 구성한다.

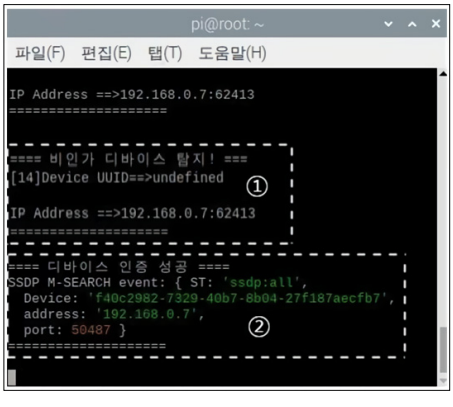

Fig. 10. Verification result

SSDP의 로그 정보는 Fig. 10에서 ①번과 같다. ①번은 IoT 기기에서 추가 인증필드 검증에서 실패하여 인가되지 않은 IoT 기기로 판별하여 로그로 저장 후 관리자에게 출력했다. 해시 기반의 SSDP의 로그 정보는 Fig. 10에서 ②번과 같다. ②번은 사전에 인가된 IoT 기기의 검증에 성공하여 해당 기기의 주소, 포트, 고유정보 등을 관리자에게 출력하고 통신을 허용한다.

5. 결론

SDP는 기기가 네트워크에 연결된 다른 기기를 찾고 통신할 수 있게 허용하는 프로토콜이다. 공격자는 SSDP의 취약점을 악용하여 시스템에 지속적으로 요청패킷을 보낸다. 이런 공격은 쉽게 50GB 이상의 트래픽을 발생시키고 더불어 웹사이트나 네트워크의 부하로 인한 서비스 중단 사태를 발생시킨다. 이에 대한 기존 SSDP의 인증절차의 문제점을 보완하기 위해 Notify 메시지와 M-Search 메시지 패킷에 서버 고유정보를 해시로 암호화하고 인증필드를 추가하여 인가된 IoT 기기의 여부를 판별하는 해시 기반의 SSDP을 제안한다.

향후 연구방향으로 SSDP 프로토콜의 요청패킷을 분석하여 증폭공격을 식별하고 증폭공격이 의심되는 경우 대응을 회피하여 다량의 응답패킷 발생으로 인한 네트워크 부하 방지가 필요하다.

REFERENCES

[1] H. G. Moon & D. J Park. (2020). Edge-Centric Metamorphic IoT Device Platform for Efficient On-Demand Hardware Replacement in Large- Scale IoT Applications. *Journal of the Korea Institute of Information and Communication Engineering, 24(12),* 1688-1696.

[2] S. C Lee & D. H Shin. (2020). TCP/IP Using Minimal Resources in IoT Systems. *Journal of the Korea Society of Computer and Information, 25(10).* 125-133.

[3] H. E Yang, Y. M Oh & Y. J Lee. (2020). The Mobile Anti-Virus Game Using IoT. *Proceedings of KIIT Conference.*

[4] J. H. Seol & K. Y Lee. (2008). Implementation of Middleware Security System for Home Networking. *Journal of the Korea Institute of Information and Communication Engineering, 12(5),* 863-869.

[5] K. O Park & J. K Lee. (2017). A Countermeasure Technique for Attack of Reflection SSDP in Home IoT. *Convergence Society for SMB, 7(2).* 1-9.

[6] J. H Oh & K. H Lee. (2016). Attack Scenarios and Countermeasures using CoAP in IoT Environment. *Journal of the Korea Convergence Society, 7(4),*

33-28.

[7] J. W. Seo & S. J. Lee. (2015). A study on the detection of DDoS attack using the IP Spoofing. *Journal of the Korea Institute of Information Security & Cryptology, 25(1).* 147-153.

[8] Y. Liu, H. C. Baek, J. H. Park & S. B. Kim. (2017). An Improved Model Design for Traceback Analysis Time Based on Euclidean Distance to IP Spoofing Attack. *Journal of convergence security, 17(5),* 11-18.

[9] H. D. Lee, H. T. Ha, H.C. Baek, C. G. Kim & S. B. Kim. (2012). Efficient Detction and Defence Model against IP Spoofing Attack through Cooperation of Trusted Hosts. *Journal of the Korea Institute of Information and Communication Engineering, 16(12),* 2649-2656.

[10] J. S. Jeon, Y. S. Jeong & W. Y. Soh. (2005). Design of Packet Generator for TCP/UDP Protocols Using Packet Sniffing and IP Spoofing. *In Proceedings of the Korean Information Science Society Conference.* (pp. 649-651).

[11] B. T. Kang & H. K. Kim. (2011). A study on the vulnerability of OTP implementation by using MITM attack and reverse engineering. *Journal of the Korea Institute of Information Security & Cryptology, 21(6),* 83-99.

[12] C. S. Lim, W. K. Lee & T. C. Jo. (2010). An Effective Protection Mechanism for SSL Man-in-theMiddle Proxy Attacks. *Journal of KIISE : Computing Practices and Letters, 16(6),* 693-697.

[13] S. J. Bang et el. (2018). A Security Analysis of IoT Hub by manufacturer through MITM Attack. The Korean Institute of Information Scientists and Engineers.

[14] M. Kuhrer, T. Hupperich, C. Rossow & T. Holz. (2014). Hell of a Handshake: Abusing TCP for Reflective Amplification DDoS Attacks. *8th {USENIX} Workshop on Offensive Technologies ({WOOT} 14).*

[15] N. K. Baik. (2020). Multi-level detection method for DRDoS attack. *Journal of the Korea Institute of Information and Communication Engineering, 24(12),* 1670-1675.

[16] Y. A. Hur & K. H. Lee. (2015). A Study on Countermeasures of Convergence for Big Data and Security Threats to Attack DRDoS in U-Healthcare Device. *Journal of the Korea Convergence Society, 6(4),* 243-248.

[17] H. S. Choi, H. D. Park & H. J. Lee. (2015). A Study on Amplification DRDoS Attacks and Defenses. *Journal of Korea Institute of Information, Electronics, and Communication Technology, 8(5),* 429-437.

[18] H. J. Kim, S. Y. Cjoi & S. S Shin. (2021). Designing a Response Scheme to Prevent Distributed Reflection Dos. Proceedings of the Korean Society for Internet Information.

김 효 종(Hyo-Jong Kim) [정회원]

· 2016년 2월 : 동명대학교 정보보호학과(공학사)
· 2017년 2월 ~ 현재 : 동명대학교 컴퓨터미디어공학과 석사과정
· 관심분야 : 웹 크롤링, 빅데이터 분석, 네트워크 보안
· E-Mail : khj47561404@gmail.com

한 군 희(Kun-Hee Han) [종신회원]

· 2001년 3월 ~ 현재 : 백석대학교 컴퓨터공학부 교수
· 관심분야 : 멀티미디어, 유비쿼터스, DB보안, 암호 프로토콜/알고리즘
· E-Mail : hankh@bu.ac.kr

신 승 수(Seung-Soo Shin) [정회원]

· 2001년 2월 : 충북대학교 수학과 (이학박사)
· 2004년 8월 : 충북대학교 컴퓨터공학과(공학박사)
· 2005년 3월 ~ 현재 : 동명대학교 SW융합보안학과 교수
· 관심분야 : 네트워크 보안, IoT, 데이터분석
· E-Mail : shinss@tu.ac.kr