블록체인 프레임워크 기반 IoT 자산관리시스템

(Study IoT Asset Management System Based on Block-Chain Framework)

요 약

본 논문에서는 관리자가 자신이 소유하고 있는 IoT 시스템을 관리할 수 있도록 하는 도구를 개발한다. 장비 에이전트는 오픈 소스 블록체인 프레임워크 기반으로 구성해 데이터의 불변성을 보장하고, 장비에 연결된 AP에 대한 추적성을 확보하여 자산에 대한 위치를 파악할 수 있다. 관리자는 블록체인 장부에서 장비의 연결 내역을 추적할 수 있다. 추가로 네트워크 형성 과정 중 발생하는 ARP 프로토콜의 ARP 추가 요청에 대한 신뢰를 없애 ARP poisoning 공격을 방지할 수 있는 가능성에 대해 연구한다.

■ 중심어 : 블록체인, 프레임워크, 사물인터넷, 자산관리시스템, 정보보호

Abstract

In this paper, we develop the tools enabling to manage the IoT system that the manager him(her)self owns. Since equipment agents consists based on open-source block-chain framework, we can secure the invariance on data and furthermore can locate the resources by searching the AP connected to the equipments. Also the manager can trace the connecting details on equipments from their block-chain accounts. In addition, we work on the possibility of protecting ARP poisoning attacks by removing the credibility on additional ARP requests being generated during the process of network creation.

■ keywords : Block-chain, Framework, IoT, Asset management system, Information security, ARP

Ⅰ. 서 론

시장조사업체 가트너는 2017년 말에 전 세계 IoT 기기 수가 2016년보다 31% 늘어난 84억 개에 이를 것이라고 예상했다. 이러한 추세는 기기 수가 200억 개를 넘어서는 2020년까지 유지될 것으로 전망하였다.

현행 ISMS 표준 체계에서는 정보자산을 분류하고 이를 식별할 것을 요구하고 있다 [1]. 접속할 수 있는 장비가 늘어날수록 관리해야 할 접속 지점은 증가한다. IoT 기기가 폭발적으로 증가하는 추세지만, 보안 관리자가 손수 관리할 수 있는 장비의 수는 제한되어 있다.

블록체인은 장비를 익명 인증하는 기능이 있다. 블록체인에 저장되는 디지털 자산에는 암호키를 기반으로 한 소유자가 배정되어 있고, 블록체인에 참여하는 장비는 하나 이상의 암호키를 가지고 있다. RSA 공개키는 1024비트의 긴 값으로 구성되어 있고, 이 값으로 표현할 수 있는 경우의 수는 십진수로 계산해서 309자리의 긴 수로서, IoT 기기들의 많은 양을 충분히 표현할 수 있다 [2,3]. IoT 기기들은 네트워크에 연결되어야 제 역할을 수행할 수 있다. 보안 담당자는 이 점을 이용해 기기들을 전부 체크하는 대신, 네트워크의 접속점(Access Point)을 조사해 자산을 파악할 수 있다.

한편 IP 기반 사물인터넷 장비는 기존 IP 프로토콜 기반 공격을 동일하게 받을 수 있다. 사물인터넷 응용도 기존의 개발 환경 적용이 쉬운 TCP/IP 기반으로 개발될 수 있으며, 이러한 시스템들은 이미 잘 알려진 공격들을 수행할 수 있게 될 수 있다.

본 연구에서는 블록체인 기반 IoT 자산관리 시스템을 제안한다. IoT 장비를 식별할 수 있도록 네트워크의 접속 지점에 블록체인 에이전트를 설치한다. 블록체인의 특성을 이용해 위변조 불가능한 자산 식별 및 이동 목록을 만들고, 검증할 수 있도록 만들어 보안 관리자가 이를 이용해 자신이 관리하는 장비 목록과 현황을 정확히 파악할 수 있는 시스템을 만드는 것을 목표로

접수일자 : 2019년 06월 18일 (학회측 사용란)
수정일자 : 1차 2019년 0월 0일 (학회측 사용란)
게재확정일 : 2019년 06월 xx일 (학회측 사용란)
교신저자 :

한다. 장부는 특정 기관의 독자 책임 없이 모든 기관이 공유할 수 있도록 한다.

또한 이 구조가 기존의 ARP 테이블과 비슷한 역할을 수행하는 점을 이용해 능동적으로 ARP Poisoning 공격을 막도록 하는 에이전트를 개발한다.

Ⅱ. 배경

1. IoT 환경에서의 기기인식 및 정보보호

기업의 보안 담당자는 자사에 존재하는 정보기기 리스트를 관리할 필요가 있다. 정보기기 리스트를 관리함으로써, 보안 담당자는 어떤 장비들이 네트워크에 어떤 영향을 미칠 수 있을지 알 수 있다. IoT 장비들은 네트워크에 연결되어 있을 필요가 있다. 네트워크의 접속 지점(Access Point)을 통해 어떤 장비들이 연결되는지 알 수 있고, 해당 기록을 점검해 장부를 만들면 보안 담당자는 특정 시점에 어떤 장비들이 연결되고 연결 해제되었는지 알 수 있을 것이다. 성능이 비교적 낮고 실시간성이 중요한 IoT 장비에서 정보를 얻는 대신, IoT 장비와 네트워크의 연결 지점에서 정보를 얻어 효율적인 자산관리가 가능하다.

2. 블록체인기반 ARP Poisoning 방어

기존의 ARP poisoning 공격은 주로 ARP 테이블을 정적으로 만들어 추가적인 ARP 추가를 받지 않는 방법으로 방어하였다 [4]. 하지만 이 방법은 ARP 테이블의 수정을 거부하는 방법이며, 변동이 잦은 장비에서 적용하기 힘들다는 단점이 있다. 또한, 이 방법은 ARP 테이블의 휘발성 문제를 해결하지 못한다.

MAC 주소를 이용해 장비의 연결을 관리하는 블록체인 구조를 만들면, ARP 테이블을 변조하는 공격을 방지할 수 있는 부가 효과를 얻는다. 공격자의 장비는 블록체인 지갑을 가지고 있지 않아 네트워크에 참여할 수 없으며, 각 장비는 블록체인이 구성하는 MAC 표를 참조해 ARP 테이블을 구성해 단순히 ARP 테이블의 변경을 막는 것보다 더 안정적인 운영이 가능하다.

MAC 주소는 변경이 가능한 정보이다. 하지만 블록체인에 MAC 정보가 포함된 객체를 저장하고, 이 객체와 기기를 연결하면 MAC 정보를 변경해 장비를 속이는 공격 또한 막을 수 있다.

또한, ARP 테이블은 따로 데이터를 저장해두지 않아 휘발성 자료지만, 네트워크에 연결된 장비의 정보는 사고 발생 시 중요한 참고 자료가 될 수 있다.

기존 DB를 이용해 ARP 테이블을 관리하는 솔루션을 만들면, 공격자의 ARP 요청이 DB의 ARP 요청보다 장비에 더 빨리 도달할 수 있으며, 장비를 검증하기 위해 별도의 조치가 필요하다. 허가형 블록체인을 이용해 각 피어가 자체 상태에서 ARP 테이블을 검증하게 만들고, 인증된 장비만 네트워크에 접속할 수 있게 할 수 있다.

Ⅲ. 시스템의 설계 및 구현

1. 시스템 설계

시스템의 요소를 장비 에이전트, 블록체인 네트워크로 나누어 설계한다. 장비 에이전트는 블록체인 네트워크에서 해당 장비가 등록됐는지 확인하고, 등록되지 않았다면 예비로 등록한다. 장비는 이미 허가형 블록체인인 Hyperledger Fabric에서 이미 공개키 쌍을 받아 접속할 수 있지만, 해당 절차는 블록체인 밖에서 이루어지기 때문에 블록체인 내부에서 이를 등록하는 절차가 필요하다.

등록이 확인되었으면, 장비가 현재 연결된 AP의 정보를 얻고(AP 확인), 해당 장비가 연결된 AP가 블록체인에 등록된 장비인지 확인한다(등록 AP 확인). 등록된 AP가 아니라면 에이전트를 종료한다.

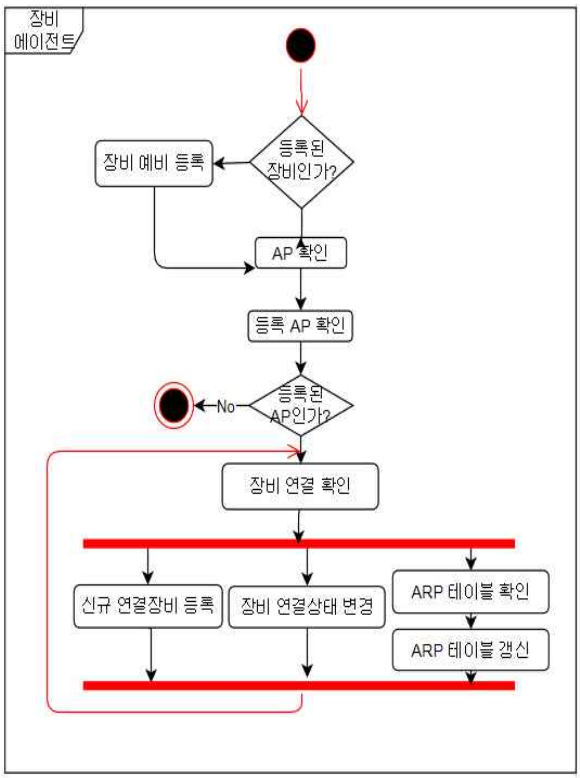

그림 1. 장비 에이전트 activity diagram

장비 연결이 확인되었으면, 하위 장비가 연결되거나 연결 해제되는 이벤트가 발생할 때를 기다린다. 새로운 하위 장비가 연결되면 블록체인에 해당 장비의 연결 상태를 업데이트하고(신규 연결 장비 등록), 기존 하위 장비가 장비 에이전트가 실행되는 장비에 연결되거나 연결 해제되는 이벤트가 발생하면 장비 연결 상태의 변경을 블록체인에 업데이트한다(장비 연결 상태 변경).

장비는 ARP 테이블을 관찰하고 있다가(ARP 테이블 확인), ARP 요청이 오면 해당 장비의 MAC을 블록체인 데이터에서 검색해 확인한다. 해당 MAC이 블록체인 네트워크에 없다면 장비는 ARP 테이블의 내용을 원래대로 복구한다(ARP 테이블 갱신).

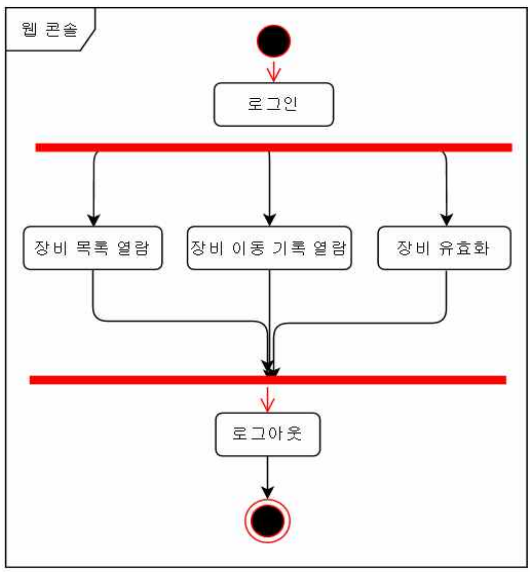

그림 2. 웹 콘솔 activity diagram

사용자는 웹 콘솔을 사용해 블록체인 장비와 상호작용한다. 블록체인이 장비를 인증하지만, 사용자는 인증하지 않기 때문에 사용자를 인증하기 위한 지식 기반이나 생체 기반의 인증을 수행해야 한다.

블록체인에 등록된 장비 목록과 장비 이동기록을 열람하고(장비 목록 열람, 장비 이동기록 열람), 블록체인에 한 번 이상 접속했지만, 관리자의 인가를 받지 않은 장비를 유효화한다(장비 유효화).

시스템의 요소를 장비 에이전트, 블록체인 네트워크로 나누어 설계한다. 장비 에이전트는 블록체인 네트워크에서 해당 장비가 등록됐는지 확인하고, 등록되지 않았다면 예비로 등록한다(장비 예비 등록). 장비는 이미 허가형 블록체인인 Hyperledger Fabric에서 이미 공개키 쌍을 받아 접속할 수 있지만, 해당 절차는 블록체인 밖에서 이루어지기 때문에 블록체인 내부에서 이를 등록하는 절차가 필요하다.

등록이 확인되었으면, 장비가 현재 연결된 AP의 정보를 얻고(AP 확인), 해당 장비가 연결된 AP가 블록체인에 등록된 장비인지 확인한다(등록 AP 확인). 등록된 AP가 아니라면 에이전트를 종료한다. 근전도 데이터 취득은 전완근의 주요 근육다발인 수근과 신근, 굴근과 손가락의 인대와 연결되어 있는 부분을 선정하여 4가지 위치로 나누어 취득하였다.

AP가 연결되면 장비 에이전트는 AP의 정보를 확인한다. 그리고 블록체인 네트워크에서 장비 정보를 질의해 해당 AP가 등록되었는지 확인하고, 장비의 MAC 등록 여부도 확인한다.

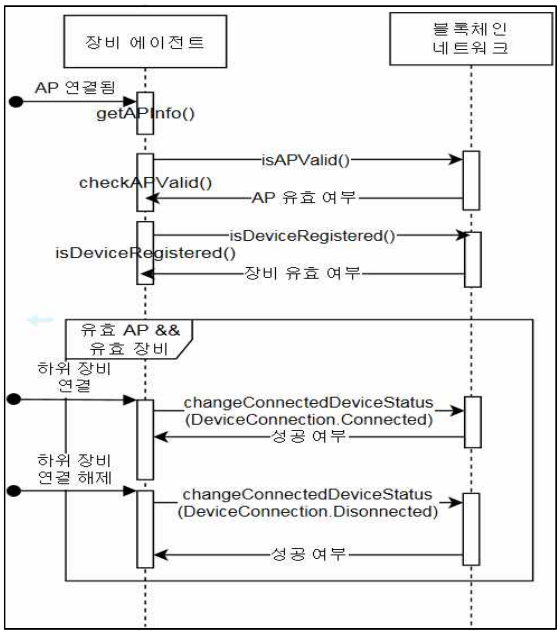

그림 3. 장비 에이전트 sequential diagram

이후 장비가 등록되어 있으며 AP도 유효하다면, 장비 에이전트는 ARP 테이블을 읽어 하위 장비가 연결되거나 연결이 해제되어 변경이 발생했을 시 이를 네트워크에 등록한다.

2. 시스템 구현

사용한 오픈소스 블록체인 프레임워크는 Hyperledger Fabric v1.1이며, 추가로 Hyperledger Composer 0.2, Hyperledger Node.js SDK를 이용해 구현하였다. 개발 OS는 Ubuntu 16.04 LTS, 사용된 도구는 Node.js 8.12, Docker 17.03, Docker-compose 1.8, npm 6.4, git 2.17.1, Python 2.7.4, VScode다. 프론트엔드 페이지를 구성할 때 Vue.js 2를 사용하였고, 블록체인 네트워크 개발에 사용한 언어는 HTML, ECMAScript(2018)이다. 자산 식별 콘솔 메인 화면에서는 장비의 현황 등 개략적인 상황을 한눈에 볼 수 있도록 하였다. 화

면 왼쪽 상단의 메뉴 버튼에서 콘솔을 조작할 수 있다.

IV. 실험 및 분석

1. 실험 환경

벤치마크는 Hyperledger Caliper 기본 설정인 1-peer-2-org, 2-peer-2-org 구성을 이용한다. ARP poisoning 실험은 Raspberry Pi 3 Model B 2개, 노트북을 이용한다. 성능 벤치마크 시 사용되는 네트워크의 구성도는 다음과 같다.

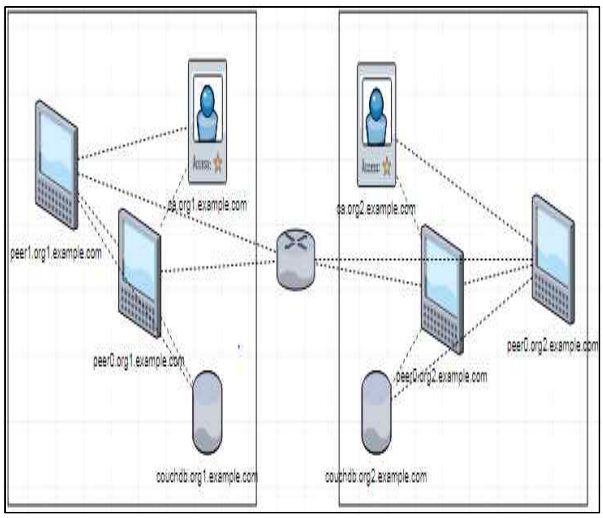

그림 4. 2-peer-2-org 네트워크 구성도

ARP 공격 시도 시 네트워크 구성은 다음과 같다.

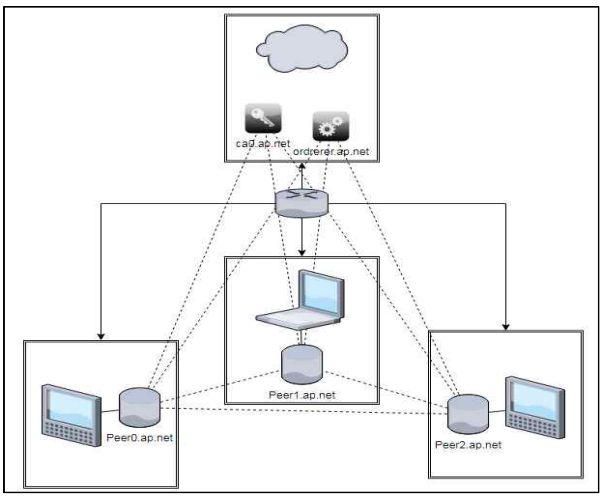

그림 5. 네트워크 구성도

ARP 공격은 Node.js의 arpjs@1.1.12 라이브러리를 이용한 스크립트를 이용한다. 공격자 Rasberry Pi 장비를 정해 스크립트를 실행해 공격자의 네트워크에 ARP 요청 패킷을 전파한다.

2. 실험 결과

1) Hyperledger Caliper 벤치마킹 결과

조직 수, 피어 수와 상관없이 초당 27개 트랜잭션을 처리할 수 있다는 결과가 나왔다. 특이 사항으로 전체 상태 DB를 관리하는 couchDB 인스턴스의 CPU 점유율이 높게 나타나므로, 실제 블록체인 네트워크 구성 시 DB 인스턴스의 CPU 사용량에 주의를 기울여야 할 것이다.

performance metrics

Name	Succ	Fail	Send Rate	Max Latency	Min Latency	Avg Latency	Throughput
ap-network	4	20	27 tps	0.83 s	0.72 s	0.75 s	3 tps

resource consumption

TYPE	NAME	Memory(max)	Memory(avg)	CPU(max)	CPU(avg)	Traffic In	Traffic Out
Process	node bench-client.js(avg)	-	-	NaN%	NaN%	-	-
Docker	dev-peer0.org2.example.co...1.0.5	99.9MB	99.8MB	8.94%	4.47%	79.5KB	65.9KB
Docker	dev-peer0.org1.example.co...1.0.5	101.9MB	101.3MB	18.12%	9.06%	145.0KB	149.5KB
Docker	peer1.org1.example.com	15.6MB	15.6MB	1.85%	0.93%	1000B	0B
Docker	peer0.org2.example.com	286.5MB	286.3MB	13.29%	6.65%	295.2KB	571.8KB
Docker	peer0.org1.example.com	257.7MB	257.3MB	21.49%	10.76%	486.8KB	850.7KB
Docker	peer1.org2.example.com	30.4MB	30.4MB	1.79%	0.90%	1000B	0B
Docker	couchdb.peer0.org2.example.com	104.6MB	104.3MB	54.33%	27.17%	84.5KB	153.9KB
Docker	couchdb.peer1.org1.example.com	101.6MB	101.6MB	3.09%	1.55%	1000B	0B
Docker	orderer.example.com	23.8MB	23.4MB	2.22%	1.11%	54.3KB	109.1KB
Docker	couchdb.peer0.org1.example.com	104.7MB	104.2MB	76.33%	38.17%	119.3KB	242.5KB
Docker	ca.org2.example.com	6.2MB	6.2MB	0.00%	0.00%	1000B	0B
Docker	couchdb.peer1.org2.example.com	120.7MB	120.7MB	1.27%	0.64%	1000B	0B
Docker	ca.org1.example.com	22.0MB	21.9MB	15.61%	7.80%	6.6KB	7.5KB

그림 6. 조직 2개, 피어 2개 상태에서 벤치마크 결과

2) ARP Spoof 공격 방어

공격자는 ARP 스크립트를 이용해 지속적으로 ARP 요청을 보낸다. 라즈베리 파이 장비의 ARP 테이블이 영향을 받지 않는 것을 확인할 수 있다.

```
[client@localhost ~]$ arp -a
_gateway (172.30.1.254) at 00:07:89:29:ee:09 [ether] on wlp
? (172.30.1.10) at 00:25:8d:5e:13:b1 [ether] on wlp2s0
? (172.30.1.20) at 00:f8:cb:79:ab:80 [ether] on wlp2s0
? (172.30.1.30) at 00:f3:26:46:f5:29 [ether] on wlp2s0
? (172.30.10.10) at 00:25:8d:5e:13:b1 [ether] on wlp2s0
? (172.30.10.20) at 00:25:8d:5e:13:b1 [ether] on wlp2s0
```

그림 7. 블록체인 피어의 ARP 테이블 확인

CPU 부하는 블록체인 상태를 관리하는 CouchDB에 집중된 것을 볼 수 있으며, 그다음으로 노드 인증 정보를 관리하는 CA, 트랜잭션을 받아 보내는 각 Peer, Orderer 순으로 CPU 점유율이 나타났다.

메모리 사용량은 Peer 0 (Anchor Peer), CouchDB, Orderer, CA, 일반 Peer 순으로 나타났다. 각 조직의 Peer 0은 Anchor Peer로 조직의 다른 Peer를 대표하는 역할을 하므로, 메모리 사용률이 가장 높다. 처리율은 초당 3 트랜잭션으로 나타났는데, 이를 극복하기 위해 장비 에이전트에서 ARP 테이블을 큐잉하는 등 추가 조치가 필요하다.

ARP Spoof 공격을 성공적으로 방어하였지만, 트랜잭션 처리량이 낮은 문제점이 보였다. Composer 환경하에서는 블록체인 지갑을 가지고 피어를 설치하지 않은 채로 장비를 가입할 수 있으므로, 성능이 높은 장비를 Peer로 두고 IoT 장비는 클라이언트로 지정해 블록체인 지갑과 Composer 스크립트를 넣고 실행해야 할 것이다.

III. 결론 및 향후 과제

기존 환경에서는 AP에 어떤 기기가 들어가고 나가는지에 대한 정보를 저장하지 않아 사고 발생 시 현황 분석에 어려움을 겪었다. 또한, 장비를 고의로 다른 장비에 연결시키는 Rouge AP 등의 공격이 가능했다. 이 시스템을 통해 보안 관리자는 네트워크 현황을 더욱 자세히 파악할 수 있으며, 휘발되지 않는 정보를 통해 사고 발생 시 더욱 용이한 원인 추적을 수행할 수 있다.

장비의 이동은 많은 데이터를 발생시킨다. 보안 관리자가 이 정보를 모두 식별하는 대신, 장비의 연결 및 연결 해제 정보를 적절하게 처리해 해당 장비가 어떤 방향으로 이동하는지 파악하는 등 해당 데이터를 어떻게 사용할 수 있는지에 대한 연구를 수행하여 보안성 향상에 기여할 수 있을 것이다.

Hyperledger의 합의 알고리즘 중 단일 장비만을 이용하는 Solo는 개발용으로, 장비의 결함 안전성(Fault Tolerance)을 보장해주지 않는다. Kafka나 PoET 합의 알고리즘에서도 블록체인 네트워크의 구조에 결함이 있으면 이론적 안전성이 보장되지 않을 수 있다. 따라서 블록체인의 안전성을 보장할 수 있는 안전한 네트워크 구조에 관한 연구도 진행되어야 할 것이다.

참 고 문 헌

[1] 한국인터넷진흥원. KISA 정보보호 및 개인정보보호관리인증체계, https://isms.kisa.or.kr/main/isms/intro

[2] Hyperledger working group. “Introduction to Hyperledger Business Blockchain Design Philosophy and Consensus. Hyperledger Architecture”, Vol. 1. 2017.

[3] Dylan Y., Peter M., Nik R., Karen S. “Blockchain Technology Overview”, NIST Interagency/ Internal Report (NISTIR), 8202. https://dx.doi.org/10.6028/NIST.IR.8202, 2018.

[4] Jeff K, Kevin L. “ARP Poisoning (Man-in-the-Middle) Attack and Mitigation Techniques”, https://www.cisco.com/c/en/us/products/collateral/switches/catalyst-6500-series-switches/white_paper_c11_603839.html, 2016.

저 자 소 개