**보안 장비 종류**

**방화벽** :

-> 네트워크를 외부망과 내부망으로 분리시키고 그 사이에 배치시켜 정보의 악의적인 흐름, 침투 등을 방지하는 시스템

-> 보안 관리자가 미리 정해 놓은 보안 정책에 따라 차단하거나 허용하는 기능을 수행하는 소프트웨어 또는 하드웨어 기반 시스템

-> 허가된 접근, 서비스, 사용자만을 통과 시키는 정책을 수립하여 외부망으로부터 내부망 보호

- 주요기능 :

-> 접근 통제(접근 제어) : 송신자의 IP 및 포트 번호를 바탕으로 패킷 필터링 수행

-> 주소 변환 : NAT 기능을 이용하여 외부망과 내부망 구분

-> 인증: 트래픽에 대한 사용자 신분 증명

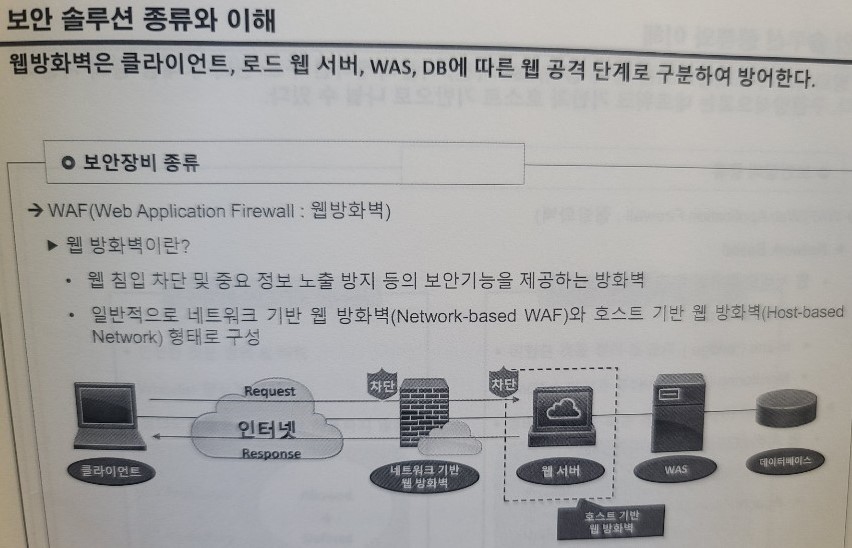
-> 감사기록/추적 기능 : 접속 정보 기록, 네트워크 사용에 따른 통계 정보

-> 프록시 : 어플리케이션 계층 필터링, 실제 IP 주소를 감춤

**WAF(Web Application Firewall : 웹 방화벽)(웹 방화벽은 클라이언트, 로드 웹 서버, WAS, DB에 따른 웹 공격 단계로 구분하여 방어한다.)**

- 웹 침입 차단 및 중요 정보 노출 방지 등의 보안기능을 제공하는 방화벽

- 일반적으로 네트워크 기반 웹 방화벽(Network-based WAF)와 호스트 기반 웹 방화벽(Host-based Network) 형태로 구성



- 주요 기능

-> 웹 공격 방지 : 논리적 공격, Client side 공격, 명령어 실행, 정보유출 등의 공격에 대한 보안 및 차단

-> Positive Security Model 지원 : 강제 접근 및 알려지지 않은 공격 차단

-> SSL 지원 : SSL 트래픽에 대한 복호화 지원

-> 어플리케이션 구조 변화에 대한 대응 : 서비스의 지속성을 보장

- Network Based

-> 웹 서버의 종류와 무관하게 보호 가능 : Reverse Proxy, In-line(bridge), Monitoring(On-armed)

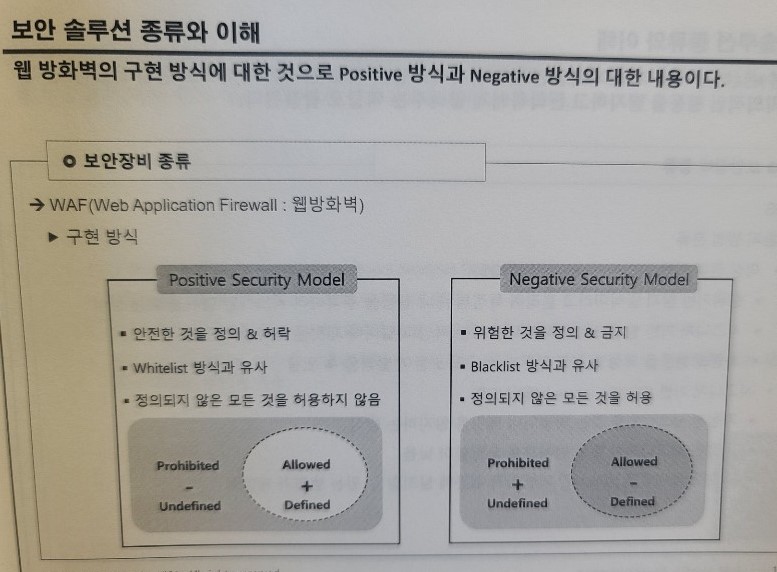
- Host Based(Agent)

-> 웹 서버/OS의 종류에 따라 제한 : ISAPI Filter, Apache Module

- Detect Method

-> Misuse (Knowledge Base) : 미리 정의된 Signature Pattern을 이용하여 탐지. 주기적인 Update필요

->Anomaly : 일반적인 행위를 기록, 비정상적인 행위 발생 시 탐지. 정상 / 비 정상 확인 어려움



**IDS(악의적인 행동을 탐지하고 관리자에게 알려주는 역할로 활용)**

- Intrusion Detection System로 침입탐지시스템의 약어

- 1980년 NSA의 제임스 엔더슨에 의해 처음 개념이 확립

- 장치 또는 소프트웨어 형태로 악의적인 네트워크나 악성행위 상태를 모니터링하여 알림 역할 수행

**분류**

- 대표적인 분류로 NIDS(네트워크 기반)와 HIDS(호스트 기반)로 분류

- NIDS(Network IDS) : 네트워크 트래픽을 분석하고 탐지하는 역할 수행

- HIDS(Host-based IDS) : 중요 운영체제 파일을 모니터링하는 역할 수행

- Application-based IDS : 어플리케이션 한정 범위 내에서 모든 현황들을 모니터링하는 역할 수행

**탐지 방법 분류**

**이상 징후 기반 IDS(Anomaly-based IDS)**

- 행위기반 탐지 방식이라고 불리며 특정 범위 내 행동을 추적하여 악의적인 행위를 찾는 방식

- 시그니처 기반 방법으로 탐지하지 못하는 악성 파일이나 패킷을 찾아낼 수 있음

- 사람의 행동을 악성 행위로 간주하는 등의 오탐이 발생할 수 있음

**시그니처 기반 IDS(Signature-based IDS)**

- 지정된 시그너처를 갖는 파일이나 패킷을 탐지하는 방식

- 지정된 시그너처만 찾는 방식으로 오탐률이 낮음

- 지정된 데이터에 의해서만 가능하기 때문에 탐지할 수 있는 범위가 제한적

**IPS(IDS 기능의 확장 형태인 IPS는 탐지 기능 뿐만 아니라 악의적인 행동을 탐지하고 차단까지 수행하는 역할을 가지며, 현재 많은 기업에서 필수적으로 많이 사용된다.)**

- Intrusion Prevention System로 침입차단시스템의 약어

- IDS의 확장 형태로 악의적인 활동을 모니터하는 기능도 포함하며 차단 기능도 수행

- 주요 기능은 주요 악의적인 활동을 차단

**탐지 방법**

**시그니처 기반 탐지(Signature-based Detection)**

- 시그니처 기반 IDS의 탐지 방법을 통해 미리 구성된 패턴과 비교 차단

**통계적 이상 징후 기반 탐지(Statistical anomaly-based Detection)**

- 이상 지후 기반 IDS 수준의 네트워크 트래픽을 모니터링하고 비교 차단

- 잘못된 설정으로 인한 오탐 경고가 발생할 수 있음

**상태 프로토콜 분석 탐지(Stateful protocol analysis Detection)**

- 프로토콜 분석을 상태 기반 특성에 추가하여 탐지

- HTTP, FTP 등 TCP 또는 UDP의 페이로드를 포함하는 프로토콜을 검사

- 공격자가 공격하는 행위를 어플리케이션 계층에서 확인 가능하며 세션으로 연계되어 있는 패킷에 대해서도 탐지가 가능

**대상 분류**

- NIPS(Network-based IPS) : 전체 네트워크 상 의심스러운 트래픽을 식별

- WIPS(Wireless IPS) : 무선 네트워크 프로토콜 대상을 식별

- NBA(Network behavior analysis) : 네트워크 트래픽을 분석하여 DDoS 공격 등의 비정상 트래픽 흐름의 위협을 식별

- HIPS(Host-based IPS) : 호스트에서 발생하는 이벤트를 분석하여 각 호스트에서 발생하는 의심스러운 위협을 식별

**IDS/IPS(악성 행위의 탐지 또는 차단의 기능으로 많은 기업에서 사랑받고 있으나 기술에 한계점 또한 존재한다. 한계점을 적절히 파악하여 효율적 보안 조치는 필수적으로 수행해야 할 것이다.)**

- 한계 :

-> 적용 조건에 의해 오탐과 미탐의 빈번함이 크게 다를 수 있으며, IPS 적용 룰에 따라 서비스 영향도에 지장 가능성이 존재

-> 시그니처 기반 탐지 방법의 경우 새로운 위협 발견화 해당 시그너처가 적용될 때까지의 지연 발생

-> 암호화된 공격 패킷의 경우 단일 솔루션으로는 차단 또는 탐지가 어려움

-> NIDS/NIPS 성격의 시스템이 TCP/IP 기반 공격 위협의 대상이 되어 위협 요인이 존재

**WIPS(Wireless Intrusion Prevention System)(지속적으로 무선랜 환경을 모니터하여 유해하거나 정의된 보안과 위배되는 동작을 탐지한 후, 각 탐지에 대해 적절한 방법으로 차단하거나 회피, 경고 등을 행하는 시스템이다.)**

- 인가되지 않은 무선단말기의 접속을 차단하고 보안에 취약한 AP(무선공유기)를 탐지하는 솔루션

- 영역 : 불법 디바이스 탐지/차단 -> 불법 AP, AD Hoc 네트워크, 인가/비인가 사용자 등

- DoS 탐지/회피

- Impersonation 탐지/차단 : MITM 공격 유형이며, 해커가 인가된 AP 혹은 사용자인 것처럼 위장하는 형태의 공격

- 패턴 매치 탐지

- 기능 :

-> 대략적인 위치 추적 : WIPS 기기를 기점으로 신호의 세기를 분석하여 위치를 파악

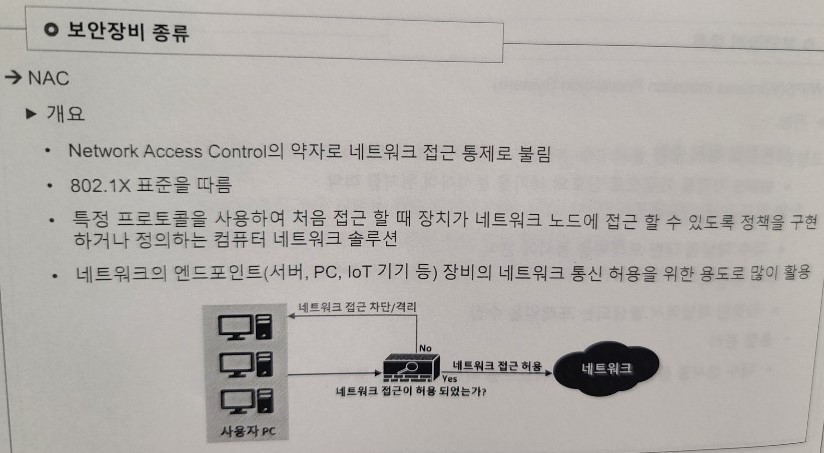
-> 다수 채널 동시 분석 : 다수 채널에 대한 트래픽을 동시에 분석

-> 무선 포렌식 : 타겟한 채널에서 발생되는 프레임을 수집

-> 통합 관리 : 다수 센서를 통합 관리하고 대쉬보드를 제공하여 통합 분석

- 인가되지 않은 디바이스 탐지/차단. 인가되지 않은 접속 탐지/차단. SSID, MAC 도용 및 DoS 공격 탐지/차단

**NAC(Network Access Control)(네트워크 인프라 관리 또는 통제를 위한 네트워크 솔루션이다.)(802.1X 표준을 따르는 NAC은 네트워크 인프라 관리 또는 통제를 위한 네트워크 솔루션.)**



**목표**

- 제로데이 공격 완화

- 인증, 인가, 계정모니터링을 통한 사용자의 장치, 응용프로그램 또는 보안 상태의 역할 기반 제어

- 안티 바이러스, 패치 또는 호스트 침입 방지 소프트웨어가 없는 엔드 포인트 단말이 다른 네트워크를 통해 오염 위험에 빠뜨리지 못하게 함

- 사용자 역할, 컴퓨터의 종류에 따라 정책을 조정하고 네트워크 영역에 접근 할 수 있도록 함

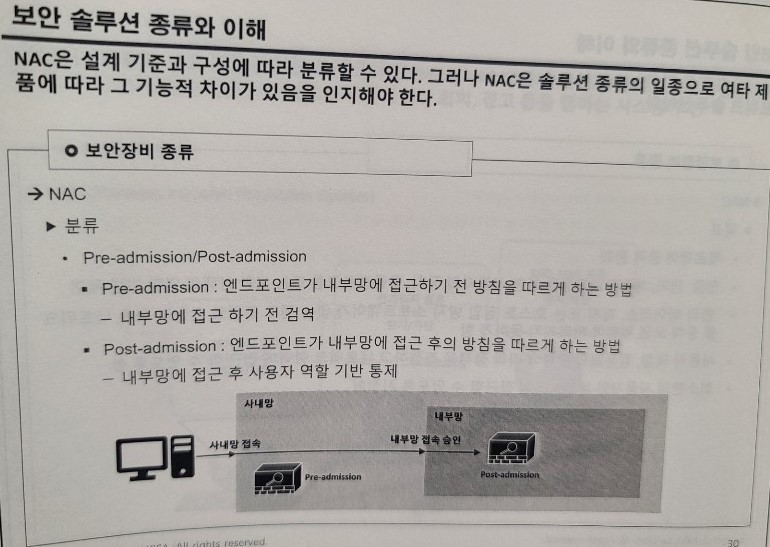
- 최소한의 사용자만 네트워크에 접근할 수 있도록 지정함

**주요 기능**

- 접근 제어/인증 : 내부직원 역할 기반 접근 제어. 네트워크의 모든 IP 기반 접근 제어

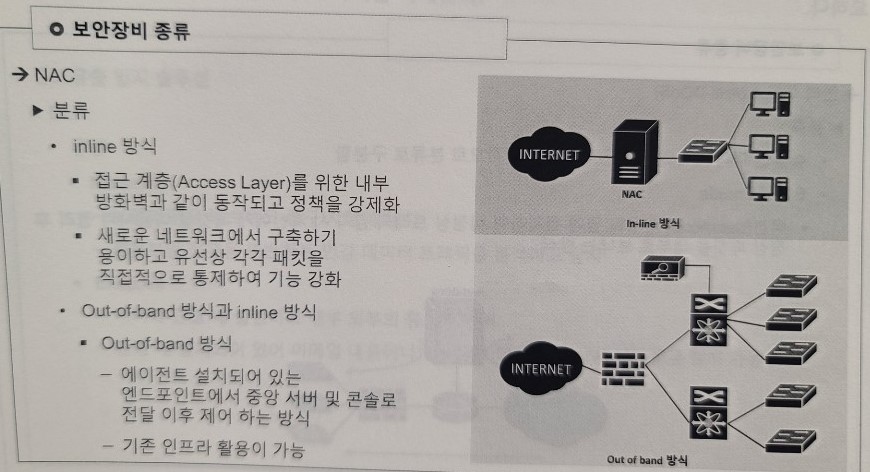
- PC 및 네트워크 장치 통제(무결성 검증) : 백신/패치/자산 관리(비인가 시스템 자동 검출)

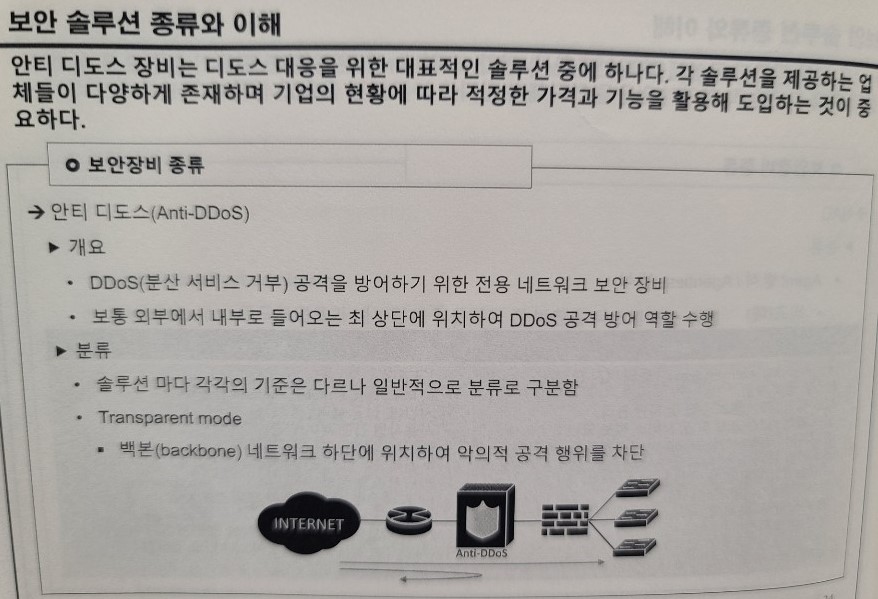
- 해킹/웜/유해 트래픽 탐지 및 차단 : 유해 트래픽 탐지 및 차단. 위협 행위 차단

**분류**

- Agent 방식 : 엔드포인트에서 정보를 얻기 위해 Agent를 통한 제어와 관리 수행

- Agentless 방식 : 설치되는 소프트웨어 없이 네트워크의 접근 통제 역할 수행

****

****

****

**정보유출 방지 솔루션(정보유출 방지 솔루션은 여럿 존재하며, 그 대응과 탐지하는 방식에 따라 조금 차이가 있다. 선천적 유출을 방지하여 사전에 정보 유출 방지를 위한 DLP와 유출이 되었을 때 컨텐츠를 보호하기 위한 DRM이 있으며 그 차이를 이해한다.)**

- DLP(Data Loss Prevention의 약어) : 기업의 중요 자산이 밖으로 유출되는 것을 방지하기 위한 솔루션. 누출 사건을 처리하는데 사용되는 기술적 범주로 분리

Standard Measure : 방화벽과 IDS, 안티바이러스 소프트웨어와 같이 일반적으로 내부와 외부의 공격을 차단

Advanced Measure : 비정상적인 전자 메일 교환에 대한 비정상적인 접근 탐지를 하기 위한 것으로 머신러닝 등의 사용자 활동 모니터링을 제공

Designed Systems : 민감 데이터 접근 가능자가 민감 데이터 전송을 시도하거나 복사하는 행위를 차단 또는 탐지

**유형**

- Network : 보통의 경우 네트워크 출구 지점에 설치. 정보보안 정책을 위반하는 민감 데이터 트래픽을 분석하고 탐지

- Endpoint : 서버나 단말 내 실행하고 내부 외부의 통신을 탐색. 단말 내 설치되어 있어 이메일 내용이나 인스턴트 메시지 등 상세 내역들을 탐지 가능

- Data identification : 민감한 데이터를 식별. 정규표현식, 메타 데이터 태그, 기계 학습, 행동 분석, 위협 모델링 분석, 패턴 탐지 등을 이용하여 내부 데이터를 식별

- Data leak detection : 인가되지 않은 장소에 발견된 누출 여부를 탐지

- Data at rest : 장기간 사용되지 않은 데이터 관리

- Data in use : 사용중인 데이터를 보호하여 불법적 활동을 모니터링하고 표시

- Data in motion : 네트워크를 통해 단말로 이동하는 데이터

**정보유출 방지 솔루션(선천적 유출을 방지하여 사전에 정보 유출 방지를 위한 DLP와 유출이 되었을 때 컨텐츠를 보호하기 위한 DRM이 있으며 그 차이를 이해한다.)**

- DRM :

-> 하드웨어 및 저작권이 있는 저작물 사용을 제한하기 위한 접근 제어 기술

-> 자산에 대해 사용, 수정, 등의 정책을 반영하여 통제

**기술 부문**

-> Verification(확인)

-> Production key : 가장 오래되고 복잡한 기법으로 과거 닌텐도 컨텐츠 보호를 위해 사용했던 방식으로 소프트웨어에 키를 넣어 자격증명을 수행

-> Limited install activations : 온라인 서버 인증을 요구하여 다른 컴퓨터에서 활성화 할 수 있는 설치 수를 제한

-> Persistent online authentication : 온라인 서버의 연결을 계속 지속하며 인증 상태를 유지

-> Encryption : 소프트웨어를 수정하여 다른 제한 조치를 우회할 수 없도록 수행. 데이터를 암호화하여 유출 시 읽기, 수정이 불가하도록 수행

-> Copy Restriction : 전자책과 문서등에 적용되며 있으며, 복사, 프린트, 전달, 백업 저장 기능들을 제한

-> Anti-tampering : 서명되지 않은 소프트웨어가 컨텐츠에 접근하는 것을 방지. 프로그램 자체 손상을 방지

-> Regional Lockout : 소프트웨어 구독 서비스 취소, 미등록 등으로 특정 지역을 벗어나거나 특정 소프트웨어를 통해 실행하지 못하도록 수행

-> Tracking

--> 워터마크 : 제작 또는 배포 중에 오디오, 비디오 등에 스테가노그래피 방식으로 내장하거나 문서 출력 시 문서 자체에 이미지가 내장되어 함께 출력

--> Metadata : 파일 내에 구매자의 이름, 계정 정보 또는 전자 메일 주소와 같은 정보를 미디어에 포함되어 저장.

**엔드포인트 보안 솔루션**

EDR(Endpoint Detection and Response의 약어)

- 가드너에 따르면 엔드포인트의 행위와 이벤트를 기록하고 수집된 데이터를 기반으로 다양한 기술을 활용하여 공격을 탐지하고 대응하는 솔루션을 의미

- 4가지 필수 기능 : 탐지, 대응, 조사, 치료

- 엔드포인트에서 다양한 정보를 수집해서 엔드포인트의 기시성을 확보한 뒤, 수집된 정보들을 기반으로 행위 분석, 머신러닝, IOC탐지 등의 기술로 알려진 혹은 알려지지 않은 위협 탐지 필요

- 악성코드 탐지와 위협을 격리하여 제거 하도록 기능 제공 필요

EPP(Endpoint Protection Platform의 약어)

- 파일 기반 악성소프트웨어 공격을 방지하고 악의적인 활동을 탐지

- 동적 보안 사고 및 경고에 대응하는데 필요한 조사 및 치료 기능을 제공하기 위해 엔드포인트 장치에 배포된 솔루션

- 주로 알려진 위협 등에 대해 빠른 처리를 위한 기능들을 수행

- 주요 기능 : 패치 관리. 취약 시스템 점검 및 조치. 디바이스 제어, 개인정보 유출 방지. 위협 탐지 및 대응, 안티 멀웨어

**EDR과 EPP 비교**

- EPP의 경우 즉각적인 대응과 다일 대상에 대한 분석 결과를 바탕으로 위협을 탐지하고 대응

- EDR의 경우 복합적인 대상의 정보 수집을 바탕으로 자동화 분석 대응 키워드가 여럿 존재

- EDR의 경우 복합적 기능을 포괄하고 알려진 위협 외에 알려지지 않은 위협도 탐지하는데 목적을 둠

- EDR이 상대적으로 EPP보다 많은 비용이 필요

**기타 보안 장비 및 솔루션**

**NPB**

- Network Packet Broker의 약어로 네트워크 장비 종류

- 라우터 또는 스위치 링크 사이에서 네트워크 트래픽을 지시하는 네트워크 장비

- 패킷의 통합/분산 역할 수행과 필터링, 로드 밸런싱 등의 네트워크 트래픽 제어 통합 역할 기능 수행

- 상세 기능

Aggregation : 여러 링크 / 세그먼트에서 나온 트래픽을 통합하는 기능

Filtering : 모니터링 툴에 가해지는 과부하를 방지하기 위해 트래픽을 조건에 맞게 걸러내는 기능

Load-balancing : 툴의 효율성을 높여주는 트래픽 균등 분배 기능

Regeneration : 여러 툴로 동일한 트래픽을 보내주는 기능

**UTM**

- Unified Threat Management의 약어로 통합 위협 관리 솔루션을 통칭

- 네트워크, 엔드 포인트 등 환경에서의 위협을 통합 관리 및 차단을 위한 솔루션

- 샌드 박스를 통한 악성코드 자동화 분석, 트래픽 분석, 엔드포인트 트래픽 분석 등의 여러 위협 요인들을 분석하여 차단해주는 역할 수행

- 장비 업체, 기종 별로 기능이 다양하여 정확한 개념을 정의하기가 어려움

**TMS**

- 네트워크 장비로부터 실시간 데이터를 수집/분석 하여 회선별 이용 모니터링 및 정보를 제공

- 수집된 데이터를 이용하여 과다트래픽, 유해트래픽 등의 이상징후를 감지하여 통보

- 도입 시 대시보드를 통한 가시성과 여러 네트워크 장비 간의 호환성을 고려해야 함

**NMS(Network Management System)**

- 네트워크 장비상태와 회선 상태 등 필요한 정보를 가지고 와서 각 시스템 모니터링

- SNMP Protocol을 이용함(string name이 항상 default값과 달라야 함)

- 최근에는 전통적인 NMS가 아닌 여러 이 기종 보안로그(네트워크 로그포함)를 그래픽 형식으로 보여주는 솔루션이 유행

- 장비 업체, 기종 별로 기능이 다양하여 정확한 개념을 정의하기가 어려움

**TMS**

- 네트워크 장비로부터 실시간 데이터를 수집/분석 하여 회선별 이용 모니터링 및 정보를 제공

- 수집된 데이터를 이용하여 과다트래픽, 유해트래픽 등의 이상징후를 감지하여 통보

- 도입 시 대시보드를 통한 가시성과 여러 네트워크 장비 간의 호환성을 고려해야 함

**NMS(Network Management System)**

- 네트워크 장비상태와 회선 상태 등 필요한 정보를 가지고 와서 각 시스템 모니터링

- SNMP Protocol을 이용함(string name이 항상 default값과 달라야 함)

- 최근에는 전통적인 NMS가 아닌 여러 이 기종 보안로그(네트워크 로그포함)를 그래픽 형식으로 보여주는 솔루션이 유행

**EMS(Enterprise Security Management의 약자로 통합 보안 관리 솔루션을 의미)**

- 관제센터 내 통합 보안 관제를 위한 목적으로 많이 활용

- 정확한 명세가 되어 있지 않아 하계와 산업계 등에서의 관점이 다양하게 해석

- IPS, IDS, NAC, 방화벽 등 다양한 보안 시스템을 하나의 통합 뷰로 관리하고 보안 정책을 일괄적으로 관리하여 서로 간의 상호연관 분석을 도와주는 솔루션

- 전문 보안 관리자의 경험적 지식 및 판단을 자동화하여 사내 보안 능력 향상을 목표로 함.

- 관리 프로세스 : 보안시스템 등의 감지기(Detector)에 의한 이벤트 정보 수집 -> 이벤트 정보 파싱 및 상관 분석 -> 침입행위 통보 또는 차단 등의 대응 조치

**SIEM(Security Information & Event Management의 약자로 보안 정보 및 이벤트 관리 솔루션을 의미)**

- 방대한 양의 데이터를 상관분석과 포렌식 기능을 제공하여 지능형 위협에 대한 모니터링 역할 수행

- ESM의 보안 영역에서 기업 전반으로 확대하고 기업 컴플라이언스 대응 기능을 확장한 형태

- 목표 : 기업 내 정보 시스템의 내, 외부 위협을 모니터링하여 공격과 정보 유출 방지

- 주요 기능

-> 로그 관리 : 이벤트 수집, 무결성 관리

-> 로그 분석 : 상관 분석, 포렌식 지원

-> 보안 : 외부 공격 탐지, 내부 위협 탐지

**SOAR(Security Orchestration, Automation and Response의 약어)**

- 보안의 패러다임을 바꾸는 기술

- 관제업무가 폭증하여 이벤트가 많아지는 어려움을 방지하고자 자동화 분석에 초점을 맞춤

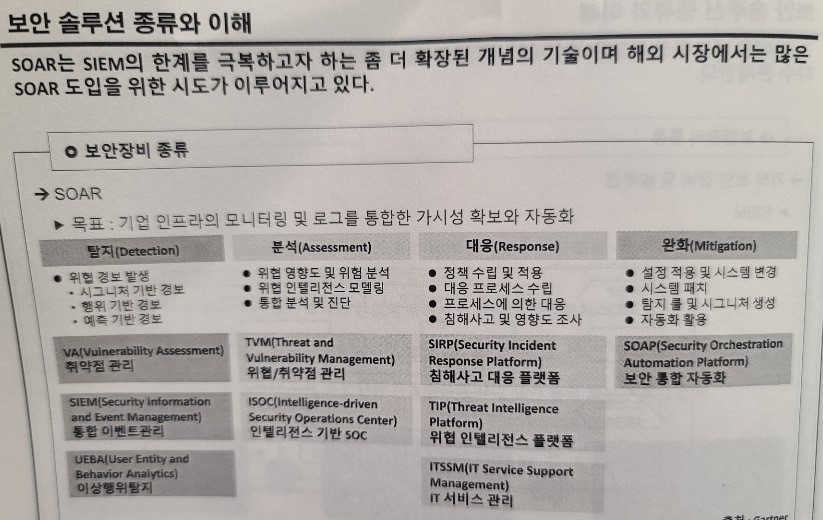
- SIEM의 한계로 관리자의 업무가 대응하기 위한 업무를 줄임(SIEM과 가장 큰 차이인 자동화가 핵심)

- 위협 인텔리전스를 주력으로 수행

- 각 부서 간의 전자 결재 등의 협의가 자동화되어 빠른 대응 가능

- 여러 보안 장비가 서로 상호 작용하는 솔루션인만큼 각 솔루션에서 API가 필요

-> 단순한 syslog 등의 로그 전송이 아닌 명령을 서로 주고 받는 기능을 위한 API 필요

****

**네트워크 해킹(네트워크 상의 연결과 관련된 모든 위협으로부터 공격 받는 것들을 의미한다. 이러한 공격은 보안 3요소와 연관되어 분류되거나 그 외 복합적인 융합 과정으로 해석되기도 한다.)**

- 네트워크 상에서 발생되는 해킹을 의미

- 둘 이상의 단말이 서로 소통하는 과정에서 이루어지는 공격을 의미

- 보안의 기본 요소에 따라 기밀성, 무결성, 가용성의 침해를 통해 공격 가능

**네트워크 위협 요소**

- 여러 통신 구간에 따라 보안의 3요소에 취약한 여러 위협들이 존재

- 보안의 3요소 뿐만 아니라 다른 요소나 아니면 복합적인 요소로 인한 침해사고 발생 가능

- 네트워크 장비를 속여 악의적인 방향으로 통신하도록 하거나 통신이 되지 않도록 하는 행위

- 네트워크 통신 경로의 중간에 개입하여 도청하는 행위

- 외부에서 네트워크 통신을 이용해 서비스가 정상 작동을 하지 못하도록 하는 행위(DoS)

- 접근제어 설정 미흡으로 비정상적인 접근을 차단 또는 탐지하지 못하여 내부에 침투하는 행위

- 내부의 자산을 파악하기 위해 훔쳐 보는 스캐닝 같은 행위(네트워크 스캐닝)

- 정상적인 통신을 강제로 끊고 공격자가 세션을 갈취하는 등의 행위(세션 하이제킹)

**보안 요소별 공격 유형 분류**

**기밀성**

- 스니핑(Sniffing) 유형 공격

-> ‘킁킁 거리다’의 의미로 네트워크 흐름의 중간에서 도청, 감시 등의 공격을 하는 행위

-> 패스워드 등의 데이터를 통신하는 과정에서 중간에 도청하여 확인 할 수 있는 위협이 될 수 있음

-> 공격 예) 패스워드 스니핑 : 네트워크 상에서 패스워드를 몰래 감청하여 탈취하는 공격

**무결성**

- 스푸핑(Spoofing)유형 공격 :

-> ‘변조시키다’의 의미로 사용자가 원하는 행위를 속여 비정상적인 행동을 하는 행위

-> 통신 경로 조작 등을 이용해 정상적인 행동을 비정상적인 행동으로 강제 변환 시킬 수 있음

-> 통신 예) DNS 스푸핑 : DNS를 변조하여 정상적인 URL을 조회하였으나 공격자가 만든 페이지로 이동하게 만드는 공격

-> ARP 스푸핑 : ARP를 이용하여 Mac 주소를 속이는 공격

**가용성**

- DoS 유형 공격

-> Denial of Service의 약자로 서비스 거부 공격을 수행하여 서버나 시스템이 동작하지 못하도록 공격하는 행위

-> 공격 예) SYN Flood 공격 : SYN 패킷을 이용하여 네트워크 트래픽 용량을 초과 시켜 정상 접근이 어렵도록 하는 공격

- HTTP Flood 공격 : HTTP 프로토콜 중 GET 또는 POST 메소드를 이용하여 웹 페이지 접근 한도를 초과 시키는 공격

**복합 공격 또는 미분류 대상의 공격**

- 스캐닝(Scanning) 유형 공격

-> 대상 시스템에 혹은 네트워크 환경에 구성을 파악하기 위한 행위

-> 보안의 3요소 공격유형에 포함되기 어려운 이유 : 스캐닝은 공격은 비밀로 감춰져 있는 정보를 탐색하는 행위가 아님(Whois 등 공개된 정보). 악의적인 목적인지의 의도를 파악하기 어려움

-> 공격 예) 포트 스캐닝 : 특정 대상의 단말이 어떤 서비스를 제공하는지 확인하기 위한 행위

- 세션 하이제킹 공격 : 서로 신뢰하는 있는 양 단말의 연결을 중간에서 갈취하여 공격자 한 쪽 사용자로 속여 연결

- 보안의 3요소 한 방향에 치우치지 못한 이유

-> 세션 연결을 중간에 탈취하기 위해 세션 연결 정보를 알아야 하므로 기밀성 침해 공격

-> 세션을 속여 마치 특정 사용자의 연결처럼 속이는 행위이기 때문에 무결성 침해 관련 공격

-> 기밀성과 무결성 침해가 복합적으로 이루어진 공격으로 3요소의 한 부분이라고 보기 어려움

**스캐닝(특정 대상의 대략적인 이해와 구조를 파악하기 위한 사전 단계로 사용되며, 공격의 방향 또는 위협의 대략적인 방향성을 잡기 위해 보통 공격자가 활용하는 방법이다.)**

- 특정 시스템 또는 인프라에서 어떤 단말 또는 서비스가 존재하는지 확인하기 위한 단계

- 침투를 하기 위한 사전 단계로서 주로 활용

- ATT&CK 메트릭스에서 Discovery 전략 단계로 구분

- 네트워크를 기반한 정보 수집이기 때문에 정보 내용이 선명하지 않음

-> 단말의 전원 꺼짐, 네트워크 요청과 응답으로 부터 오는 정보의 부족으로 100% 정확한 정보 수집은 어려움

- 공격자 관점에서의 스캐닝

-> 공격의 범위를 산정

-> 공격 대상을 식별

-> 사용할 공격 도구 또는 취약점을 사전의 파악하여 준비하는 단계

**방어자 관점에서의 스캐닝**

- 진단의 기능 : 취약점 진단, 모의해킹 등의 기술적 진단 업무 중에서 업무를 수행하기 전에 자산이나 취약점 진단 대상을 파악하기 위한 단계. 진단 업무 수행 전 도구의 준비 등을 수행하기 위한 단계로 활용

- 점검의 기능 : 사내 부서, 협업 부서 등 사내 여러 부서의 보안 수준이 제대로 되어 있는지 확인. 사내에서 규정에 어긋나는 행위 등을 파악하기 위한 단계로 활용

- 감사의 기능 : 타 사의 보안 수준과 실제 보안 규정과 절차를 잘 이행하고 있는지 확인. 사내에서 규정에 어긋나는 행위 등을 파악하기 위한 단계로 활용

**정보 수집을 위한 기술**

- 공개 소스 기반 정보 수집 : 회사 홈페이지, 뉴스, 블로그, 트위터 등을 통해 수집하고자 하는 대상의 정보를 수집. OSINT(Open Source Intelligence)으로도 불리며, 공개되어 있는 사이트를 기반으로 정보 수집. 획득 가능 정보(위치, 전화번호, 네트워크 주소, 내부 정책, 직원 이름, E-mail, 부서, 기업 현황 등). 관련 도구(Shodan, Censys 등)

- 사회공학적 기법 정보 수집 : 채팅 또는 휴대폰을 이용한 소통과 E-mail 스푸핑 공격을 통해 정보 수집. 사람 심리의 약점을 이용해 공격하는 기법

- 도메인을 통한 정보 수집 : 도메인 질의를 이용하여 정보를 수집. 획득 가능 정보 등록된 조직, 담당자, 개인 정보, 도메인 정보, 프로젝트, 서비스, 기술 등. 관련 사이트(whois.com, ipvoid.com, archive.org 등). 관련 도구(dig, maltego 등)

- 네트워크 정찰 기반 정보 수집 : 스캐닝 도구, 정보 수집 도구 등을 활용해서 현재 활성화 되어 있는 네트워크 또는 인프라 정보를 수집. 정확한 정보를 수집하기는 어렵지만 공격의 대상을 식별하고 결정하는데 주요 역할을 수행. 획득 가능 정보(네트워크 흐름 정보. IP 주소 또는 대역, 단말, 통신 장비 등). 관련 도구(Nmap, Unicornscan, tracerpite/tracert, DMitry 등)

**네트워크 스캐닝(스캔을 통해 서비스를 제공하는 서버의 작동 여부와 제공하고 있는 서비스를 확인할 수 있다. 스캐닝에 사용하는 프로토콜은 ICMP, TCP, UDP로 나눌 수 있다.)**

**NMAP이란?(100% 정보를 얻기는 불가능. 대략적으로 정보를 얻는다.)**

- 대표적인 스캐닝 도구로 nmap이 존재

- 보통 네트워크를 통해 제공하고 있는 서비스, 포트, HOST 정보 등을 수집

- TCP 기반의 프로토콜의 질의(Request) 응답(Response) 매커니즘을 기반한 스캐닝 수행

- 도구의 기능

-> 열려 있는 포트 확인

-> 제공하는 서비스 확인

-> 동작하는 Daemon의 버전

-> 운영체제 종류 및 버전

-> 취약점 점검 등의 역할

**스캔 종류**

- Ping & ICMP 스캔 : Ping은 네트워크와 시스템이 정상적으로 작동하는지 확인. Echo Request와 Echo Reply를 이용한 방법. ICMP(Internet Control Message Protocol)를 사용

ICMP

- Window, Linux에서는 기본 차단이다.

- LAN환경에서 아무리 ICMP스캔을 하더라도 기본적으로 차단되어있기 때문에 효율적이지 않다. ICMP에서 얻을 수 있는 정보는 호스트가 켜져있는지 꺼져있는지 확인한다. 좀 더 효율적인것이 필요하다.

- LAN에 해당하면 ARP 프로토콜을 이용해 스캐닝을 한다.

**- 오픈 스캔 :**

전통적인 TCP 3-way Handshake를 이용해 정상적인 연결을 바탕으로 Open된 포트 정보를 추출.

TCP connect 스캔 : Open된 포트의 경우 target 시스템에서 SYN/ACK 패킷이 응답.

Close된 포트의 경우 target 시스템에서 RST/ACK 패킷이 응답

**- 하프-오픈 스캔** : TCP 3-way Handshake 방식의 연결을 비정상적으로 종료 하는 방식

-> 표적 시스템 로그에 기록되는 것을 피할 수 있으나 방화벽이나 IDS에는 탐지

-> TCP 하프 오픈 스캔 : SYN을 보낸 후 표적에서 SYN/ACK 응답이 오면 표적 HOST가 살아있다고 추측. 출발지에서 목적지로 ACK를 보내는 대신 RST를 보내 세션을 성립하지 않고 로그를 남기지 않음

**- 스텔스 스캔 :**

세션을 완전히 성립하지 않고 공격 대상 시스템의 포트 활성화 여부를 알아내는 스캔

시스템 세션 연결 관련된 로그가 남지 않음. ACK, NULL, X-MAS 스캔 등이 존재

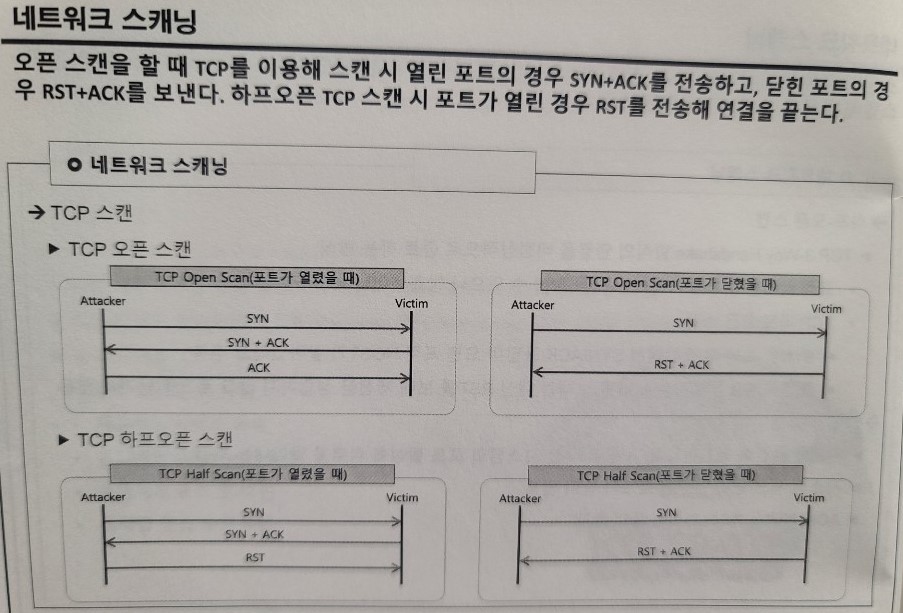
- X-MAS 스캔 : 모든 flag를 보내거나 FIN, PSH, URG 플래그를 보내는 스캔

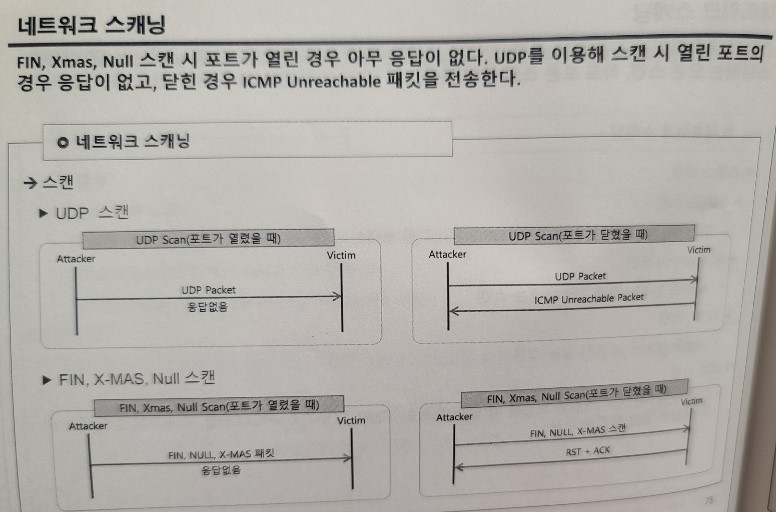
- ACK or FIN 스캔 : ACK 또는 FIN 플래그만 보내는 스캔

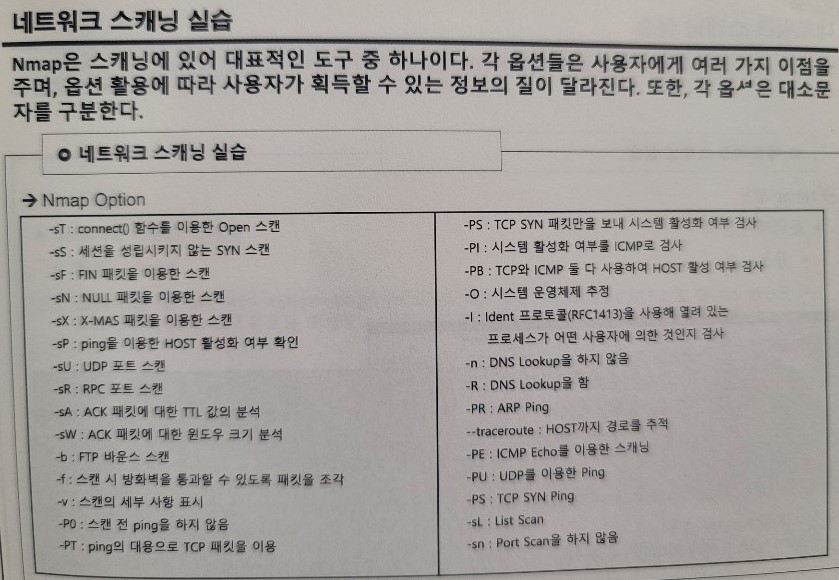
- NULL 스캔 : flag를 하나도 보내지 않는 스캔

**UDP 스캔**

­- 대상 HOST에 UDP패킷을 보냈을 때 닫힌 포트는 ICMP\_PORT\_UNREACH를 보내고 열린 포트는 아무 응답이 없는 방법을 이용







**네트워크 스캐닝 실습**

- Nmap으로 -sP 옵션 사용 시, 검색하고자 하는 대역대의 시스템 중 활성화된 시스템을 확인할 수 있다. EX) nmap -sP 192.168.0.0/24

- Nmap으로 -O 옵션 사용 시, 운영체제(Operating System) 정보를 획득할 수 있다. Ex) nmap -O 211.171.14.207. 장비(Machine)의 경우 Device type이 embedded라고 표시

- Nmap으로 --top-ports N(개수) 옵션 사용 시, 가장 많이 쓰는 상위 N개의 포트를 스캐닝 할 수 있다. Ex) nmap –top-ports 5 211.171.14.207

- Nmap으로 --script 옵션 사용 시, Nmap에서 제공하는 스크립트를 이용한 스캐닝이 가능 하며, 스크립트는 Nmap이 설치된 위치의 scripts 폴더에 존재한다. EX) nmap -p 139, 455 --script=smb-check-vulns 스크립트를 이용한 MS08-067 취약점 및 Conficker Worm 감염 여부 확인

- Ping을 이용한 스캔 시 리눅스는 TTL 값이 64, 윈도우는 TTL 값이 128인 것을 확인할 수 있다.

- 각 운영체제마다 고유의 TTL 값이 있다. Ping을 보낼 시 돌아오는 TTL 값을 확인 해 운영체제 추측이 가능하다.

- Telnet과 같이 원격지의 시스템에 로그인을 시도하였을 때 나타나는 안내문을 배너라고 하는데, 기본적으로 이러한 배너는 애플리케이션 버전 등을 나타낸다. 이를 통하여 정보 수집이 가능하다.

- Naver와 같은 홈페이지의 경우 보안상의 이유로 ICMP 패킷을 차단하는 경우가 존재한다. 이러한 경우 서버가 구동이 되고 있는지 확인하기 위해서는 활성화 된 서비스가 있는 포트를 대상으로 서버가 존재하고 실행되고 있는지 스캔할 수 있다.

- SYN 스캔

-> hping3 명령어를 이용하여 아래와 같이 명령어를 입력

-c : 보낼 패킷 수 / -p : 포트 / -S : SYN 패킷 Flag

hping3 -S [www.naver.com](http://www.naver.com) -p 80 -c 2

RST패킷을 보낸 이유 : 3 way-handshake 방식으로 연결하고자 하는 서버에게 강제적으로 세션연결을 종료 시킴으로써 연결에 대한 흔적을 남기지 않기 위해 수행함.

ttl = time to live

홉을 지날 때 마다 -1 경로를 못찾으면 홉 + 1, ttl=0이 되면 삭제

**근거리 네트워크 공격(근거리 네트워크(LAN) 내부에서 일어날 수 있는 위협은 외부와 크게 다르지는 않지만 OSI 7 Layer에서 2계층에 해당하는 Mac주소 또는 프로토콜을 이용한 공격이 수행되는 것이 특징이다.)**

- LAN 범위 내에서 수행될 수 있는 공격에 한정하여 OSI 7 Layer 중 2계층에서 이루어지는 공격을 의미

- 주요 공격 : ARP를 이용한 MITM 공격

-> Man In the Middle 의 약자로 중간자 공격이라 불림

-> 통신하는 두 단말 사이에 중간자가 침입하여 양단의 통신을 도청하거나 조작

-> 중간자 공격을 막기 위해 TLS/SSL 프로토콜을 이용한 공개 키 기반 인증을 사용

-> 2계층에서 주로 활용하는 공격으로는 ARP 스푸핑을 연계한 공격이 존재

**근거리 네트워크 공격 종류**

**주요 공격**

- STP(Spanning Tree Protocol) 공격 :

-> STP는 스위치 네트워크 환경에서 사용되는 프로토콜

-> 루핑(정상적인 연결을 수행하지 못하고 통신이 쳇바퀴처럼 도는 현상) 공격을 방지하는 프로토콜

-> STP를 이용하여 스위치의 연결 흐름을 제어하여 악의적인 목적의 스위치로의 연결을 전환 가능

- MAC 스푸핑 공격

-> 연결되어 있는 MAC 주소를 변조하는 공격

-> 특정 장치로 위장하거나 해당 장치의 서비스를 거부하는 공격으로 활용

**주요 네트워크 공격**

- CAM(Content Addressable Memory) 테이블 오버플로우

-> MAC 주소 테이블은 스위치에서 연결되어 있는 장치로의 데이터 송수신을 위하여 MAC 주소별 특정 경로 선택을 위한 IP 또는 포트를 매칭하고 저장하고 있는 공간

-> 스위치 전원이 올라왔을 때 MAC 테이블을 임의적으로 과도하게 늘려 저장하도록 수행

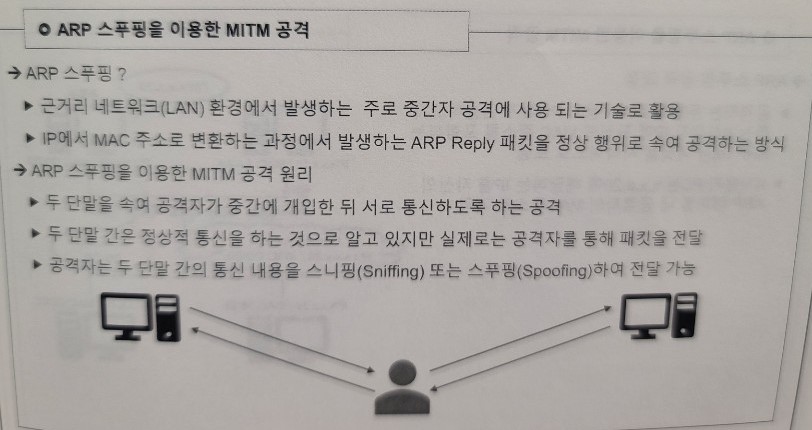
-> 저장된 테이블 수가 다량이면 정상적인 연결이 새로 들어올 경우 이미 잡혀 있는 메모리의 한계가 존재하기 때문에 정상 연결이 불가하도록 하는 DoS 유형의 공격

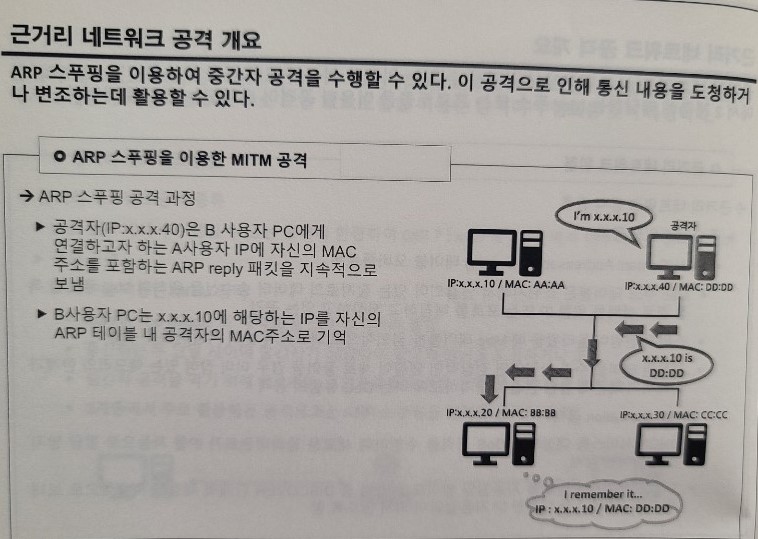
- DHCP Starving 공격

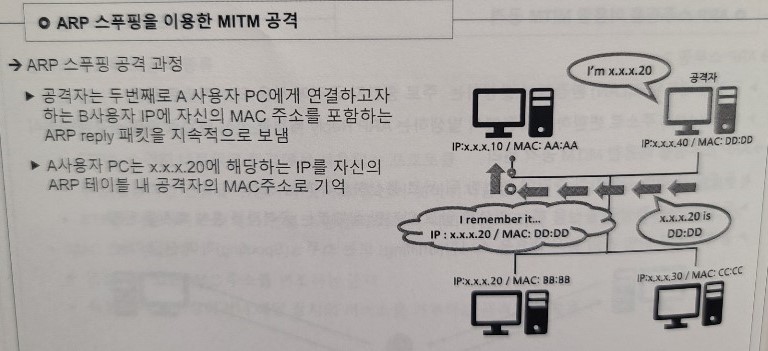
-> DHCP 서비스를 대상으로 DoS 공격을 수행하여 새로운 클라이언트가 IP를 자동으로 할당 받지 않도록 하는 공격

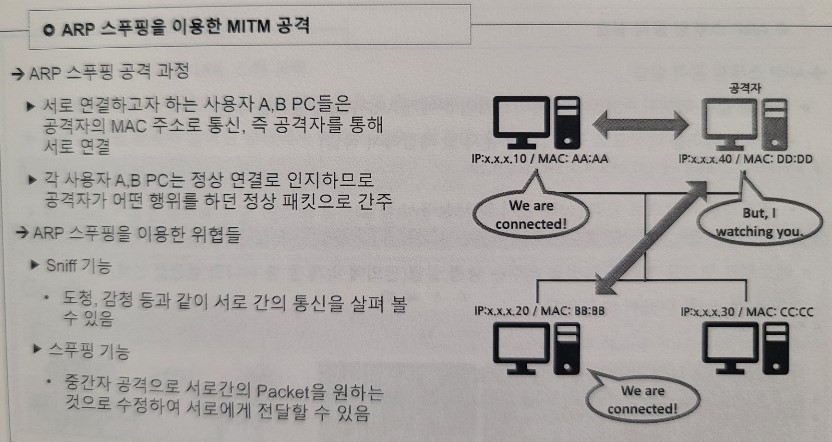
-> DHCP 서비스에서 IP를 자동할당 받아오는 과정 중 DISCOVER 단계의 패킷을 지속적으로 보내어 DHCP서비스를 이용한 IP 자동할당이 되지 않도록 함.

**ARP 스푸핑(ARP 스푸핑은 근거리 네트워크에서 사용되는 대표적인 중간자 공격으로 두 단말 간의 통신을 속여 자신을 통해 통신하도록 만들어 중간에서 네트워크 통신 내용을 스니핑 또는 스푸핑 하도록 하는 기술이다.)**









**DoS 공격의 이해와 종류**

**DoS**

- Denial of Service의 줄임말

- 서비스를 일시적 또는 무기적으로 중단시키는 공격 유형을 통틀어 뜻함

- 시스템의 과부하나 시스템 오류를 일으켜 일부 또는 모든 합법적 요청이 정상 동작하지 않도록 수행

- 대표적인 공격 예시 : Ping of Death 공격

**DDoS**

- 여러 시스템이 대상 시스템의 대역폭이나 리소스를 초과하여 서비스를 거부하는 **방법**

- 보통 시스템에 넘쳐나는 트래픽으로 시스템을 손상시킴

- 시스템의 결함이 아니더라도 가용성 침해 공격이 가능

- DoS와 DDoS를 동일선상에서 비교하는 것이 아닌 DoS 중 하나의 형태로 생각해야함

- 대표적 공격 예시 : SYN Flooding 공격

**7.7 DDoS 공격**

- 공격자는 악성코드를 웹 페이지 및 Spam-Relay 기법으로 배포하였으며, 피해 PC는 악성코드 감염 후 악성코드에 포함된 URL List와 Timer를 이용하여 공격하였다.

- 공격 시나리오 : 취약한 웹 사이트에 악성코드를 은닉 하거나, Spam-Relay 기법을 통해 악성코드를 배포. 피해 PC에 악성코드 감염 후, 공격 URL List와 Timer에 의해 공격

- 공격 기법 :

HTTP Get Flooding, CC(cache-control) 공격 형태의 Web Server 의 자원을 고갈시키기 위한 공격.

UDP Flooding, ICMP, Ping Flooding 네트워크 대역폭 고갈 공격이 함께 이루어짐

- 공격 특징 :

전파 경로 : 보안에 취약한 웹 서버 공격 및 Spam Mail을 통해 전파

- Agent : C&C 서버 없이 자체 공격 리스트와 Timer에 의해 정해진 시간에 공격 후 자가파괴 가능

**3.3 DDoS 공격**

- 네트워크의 특성을 이용한 공격 기법을 이용하여 서비스를 거부하도록 만들었으며, 주요 P2P 서버를 사용하여 악성코드를 배포하였다.

- 공격 기법 :

HTTP Get Flooding, CC(cache-control) 공격 형태의 Web Server 의 자원을 고갈시키기 위한 공격.

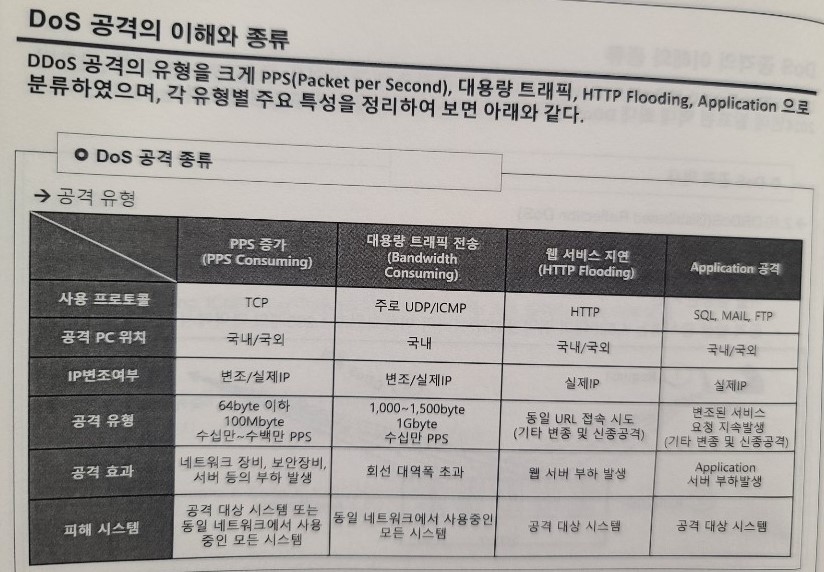
UDP Flooding, ICMP, Ping Flooding 네트워크 대역폭 고갈 공격이 함께 이루어짐

- 공격 특징 :

전파 경로 : P2P 사이트 등을 통하여 악성코드를 배포. 주요 유포지 : sharebox, superdown, filecity, bobofile

- Agent : C&C(Command & Control) 서버가 명령을 내리는 형식





**DoS 공격 종류**

**Flood 공격?**

- 보통 공격 대상자에게 작업이 가능한 범위 이상으로 데이터를 흘려 보내 가용성 침해를 일으키도록 하는 공격 방식을 의미

- Flood 공격 종류

- Flag 조작에 따른 공격 : SYN Flood 공격, ACK Flood 공격, SYN/ACK Flood 공격, FIN/RST/PSH Flood 공격.

- 기타 네트워크 특징을 이용한 공격 : UDP Flood 공격. Application-layer Flood 공격, HTTP GET/POST Flood 공격, DB Query Flood 공격, DNS Query Flood 공격, ICMP Flood 공격.

**DoS 공격 종류 - BPS**

**UDP Flood 공격(Flood 공격의 일종인 UDP Flood 공격은 비교적 단순한 공격으로 현재 대부분의 os에서 자체적으로 제한하기 때문에 거의 사용되지 않은 공격 중 하나다.)**

- UDP를 이용한 비연결형 DDoS 공격의 일종

- 대량의 UDP를 발생하여 정상적인 서비스를 하지 못하도록 가용성 침해를 일으키는 공격

- UDP 패킷을 받은 시스템은 어플리케이션이 없다는 것을 확인한 후 주로 ICMP 프로토콜을 사용하여 Destination Unreachable 메시지를 출발지로 전달

- 대부분 공격자가 UDP 패킷의 출발지 주소를 임의로 생성하여 보내 공격자로 패킷이 전달되지 않음.

- 현재 대부분의 OS에서 ICMP 응답비율 제한음으로 UDP Flood 공격의 위험을 경감

- 공격 원리 :

-> 임의의 주소를 시작 지점으로 대상 시스템에게 UDP 패킷을 대량으로 전송

-> UDP 패킷을 받은 시스템은 관련된 어플리케이션을 확인한 후 오류에 해당하는 ICMP Destination Unreachable 패킷을 출발 주소로 보냄

-> 다량의 데이터를 처리하여 정상적인 데이터가 왔을 때 즉각적인 대응이 어려움

**DoS 공격 종류 – PPS**

**IP Spoofed SYN Flooding 공격**

- IP 변조 후 다량의 SYN 패킷을 공격 대상 서버로 전송

- 공격 받은 서버는 다수의 SYN\_RECEIVED 세션 상태가 발생

- 서버의 CPU 및 Connection 자원의 고갈을 유발

**TCP Connection Flooding 공격(3-way handshake 정상 완료)**

­- IP를 변조하지 않고, 다량의 SYN 패킷을 공격 대상 서버로 전송

- 공격 받은 서버는 다수의 ESTABLISHED 세션 상태가 발생

- 서버의 CPU 및 Connection 자원의 고갈을 유발

**TCP Out-Of-State Packet Flooding 공격(ACK/SYN + ACK/FIN 등)**

- 다량의 ACK/SYN + ACK/FIN/RST 등의 패킷을 공격 대상 서버로 전송

- 일부 네트워크 장비 및 서버의 CPU 사용량이 올라가는 등 오작동 발생 가능

**SYN Flood 공격**

- 3way handshake 방식인 연결지향형 방식의 TCP 특성을 이용하여 서비스 거부하는 공격으로 포트의 최대 연결 허용 가능 대기 큐를 초과해 정상 연결을 하지 못하도록 만드는 공격이다.

- 대표 Flood 공격 중에 하나로 다량의 공격을 빠른 시간 내에 반복적으로 하는 공격

- TCP의 3 way handshake 방식을 이용, SYN 패킷을 보내 피해자 서버가 기다리게 하는 허점을 이용

**공격 원리**

-> 공격자가 출발 주소를 스푸핑하여 SYN패킷을 대상 시스템에게 전송

-> 대상 시스템은 해당 시스템의 응답을 대기한 상태로 다른 응답을 반복적으로 처리

-> 대기 큐에 연결 허용 횟수가 가득 차게 되어 정상 연결이 오는 것을 무시

**ACK Flood 공격**

- SYN Flood 공격과 유사한 형태로 연결 지향형의 특징을 이용한 서비스 거부 공격의 하나이며, 응답을 처리하는데 자원을 소비시켜 정상 행위를 할 수 없도록 만드는 공격이다.

- Flood 공격 중에 하나로 다량의 공격을 빠른 시간 내에 반복적으로 하는 공격

- TCP의 연결 지향형 특징을 이용한 방법으로 대상 시스템에 특정 패킷이 오면 해당 패킷을 처리하기 위해 자원을 소비하도록 유도하는 공격

**공격 원리**

- 공격자가 출발 주소를 스푸핑하여 ACK패킷을 대상 시스템에게 전송

- 대상 시스템은 ACK 플레그에 대응하는 응답을 처리함으로써 많은 리소스를 사용

**기타 TCP Flag(Push-ACK, FIN, RST, URG) Flood 공격**

­- TCP Flag(Push-ACK, FIN, RST, URG) Flood 공격? ACK Flood 공격과 거의 유사하며 TCP의 연결지향형 특징을 이용한 공격 방법

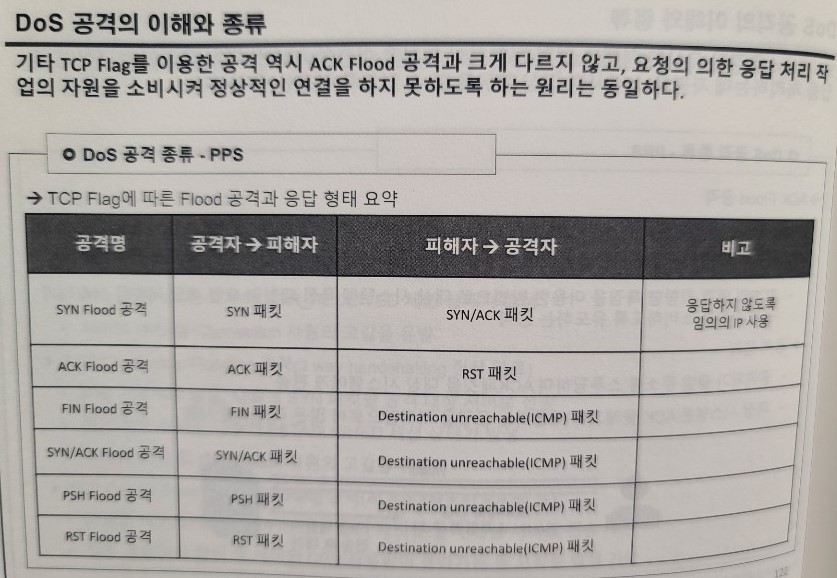
- Flag 설정에 따라 응답 처리 방식이 조금씩 다르지만 Flood 형태의 자원소모 시켜 서비스 거부하는 형태는 동일

**공격 원리**

- 공격자가 출발 주소를 스푸핑하여 ACK패킷을 대상 시스템에게 전송

- 대상 시스템은 ACK 플레그에 대응하는 응답을 처리함으로써 많은 리소스를 사용

- 서버는 대상 내용과 관계 없는 연결이기에 잘못된 연결을 알려주는 ICMP 프로토콜을 전송



**DoS 공격 종류 – RPS(RPS(Request Per Second)는 초당 요청에 따른 공격을 의미하며 보통 HTTP 등의 OSI 7 Layer 모델에서의 7계층인 어플리케이션 계층을 타켓으로 공격하는 것들이 대부분 이 공격에 해당한다.)**

**HTTP Flood 공격(OSI 7계층 영역에서 발생되는 공격이며 7계층의 특징으로 대역폭으로 차단하는 여타 보안 장비를 우회 할 수 있어 대응이 까다로운 공격으로 7계층의 내용을 확인할 수 있는 IPS와 같은 장비로 방어해야 한다.)**

- DDoS 공격의 분류로 HTTP의 메소드인 POST, GET의 메소드를 이용한 7계층 공격

- 다량의 HTTP 세션 연결하여 정상적인 세션이 연결될 수 없도록 하는 공격 기법을 활용

- 종종 트로이목마 악성코드를 이용한 방법들이 사용되며, 대역폭 자원을 고갈시키는 공격보다 적은 양으로 가용성 침해를 일으키는 것이 특징.

- GET방식은 이미지와 같은 표준 정적 컨텐츠를 이용하여 접근

- POST방식은 동적으로 생성된 자원을 이용하여 수행하는 공격 방법이 주로 사용

- 대역폭을 기준으로 차단하는 방어 장비를 우회 가능

**Slow-Rate 공격**

- Low and Slow(낮고 느린) 이라고도 불리우며 겉으로 보기엔 정상적이지만 느린 속도를 가진 트래픽으로 확인

- 대표적인 공격 도구는 Slowloris, Sockstress, R.U.D.Y(R-U-Dead-Yet)이 있음

- 트래픽을 양을 확인하여 방어하는 장비 혹은 OSI 7계층을 모두 보지 못하는 보안 장비를 우회 가능

**R.U.D.Y(R-U-Dead-Yet)(R.U.D.Y 류의 공격과 그의 변종이 많이 알려진 공격 중 하나로 알려지 있는 방법 중 하나이다. HTTP POST 메소드를 이용해 공격하는 방법을 이용하여 특히 그 중 Content-Length 속성 값을 변조하여 공격을 수행한다.)**

- Slow-rate 방식으로 실행되는 대표적 공격 도구이며, 오늘날까지도 이 형태로 공격이 다수 수행

- POST 메소드를 이용하고 ‘Content-Length’를 크게 할당하며, POST 영역 중 특정 변수를 지정하여 안에 값을 하나씩 대입하여 연결을 지속하는 것이 특징

- 프록시 기능을 추가 가능

**TorsHammer(Slow-Rate 공격을 이용한 도구로 POST 메소드에 Content-length를 크게 할당, 이후 content-length 값 만큼 조금씩만 데이터를 보내 연결을 계속 유지시키는 것과 동시에 다량의 세션을 유지하여 가용성 침해를 일으키는 공격이다.)**

- R.U.D.Y(R-U-Dead-Yet) 도구와 유사한 형태

- POST 메소드를 이용하여 Content-length의 길이를 확장한 후 해당 길이의 내용을 조금씩 보내 자원을 소모 시키는 도구

- Tor 서버를 이용하여 프록시 기능을 활용 가능

**Slowloris(OSI 7계층 중 7계층을 통한 공격으로 일정 대역폭을 탐지하여 방어하는 보안장비를 우회할 수 있다.)**

­- 로버트한슨에 의해 개발. 한대의 컴퓨터로 웹서버를 다운 시키는 공격도구로 Slow-Rate 공격 기법을 이용함

- HTTP Flood 유형의 공격으로 7계층에서 행하는 공격

- Apache 1.X, 2.X 버전에서 효과적인 공격 가능

- 다른 서비스 및 포트에 영향을 미치지 않으며, 웹 서버에만 영향을 미침

- 여러 세션을 길게 유지하여 연결 가능한 세션 수를 초과 시켜 정상 연결을 할 수 없도록 하는 공격 (HTTP에서 GET 또는 POST 헤더를 불완전 전송해 완전한 패킷이 올 때까지 세션 연결을 유지)

**Cache-Control(CC) Flood 공격(HTTP의 필드 값을 수정해 요청 시마다 새로운 응답 값을 받아 과부하 시키는 공격이다.)**

**공격 개요**

- OSI 7계층에서의 가장 상위 계층 중 HTTP를 타겟으로 하는 대표적인 공격 중 하나

- HTTP의 GET 메소드의 Cache Control 필드를 조작한 공격

**공격 원리**

- 일반적인 정상 웹 클라이언트는 한번 수신한 데이터는 웹 캐시에 저장하여 빠른 처리 진행하는 특징을 이용, 이 공격을 통해 웹으로부터 받아오는 데이터를 매번 새롭게 요청하여 과부하 시킴

- HTTP의 필드 값을 ‘no-store 또는 no-cache’와 ‘must-revalidate’로 변경하여 공격

**DoS 공격 종류 – 기타**

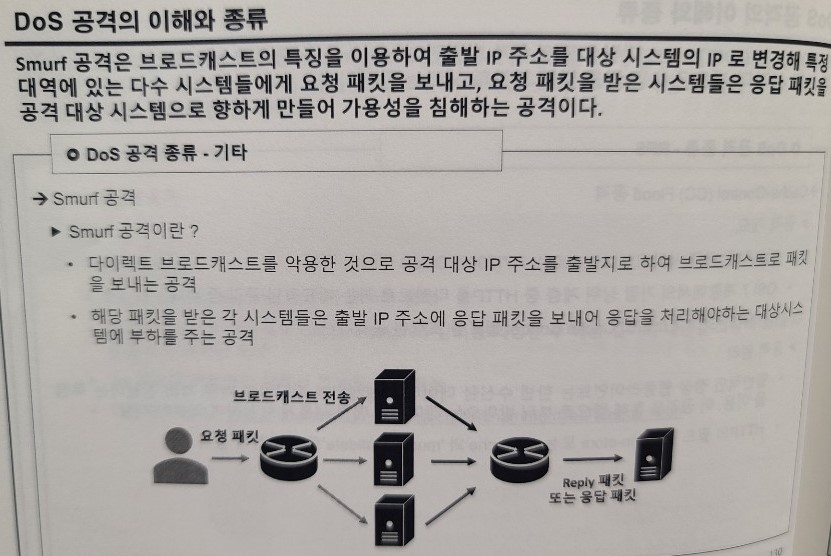
**LAND Attack(출발 IP와 도착 IP를 동일하게 하여 같은 파일을 대상시스템의 IP로 동일하게 보내어 작업을 반복되도록 하는 공격으로 현재 대부분의 os에서 패치가 되었기 때문에 적용이 제대로 이루어지지 않는 공격이다.)**

- Local Area Network Denial의 약어

- 출발지와 도착지 IP 주소를 대상 시스템의 주소로 동일하게 만들어 보내는 공격

- 자기 자신에서 출발하 패킷은 자기 자신에게 패킷을 다시 보내는 것을 반복(Loop)

- 현재 대부분의 OS 패치가 적용되어 공격이 적용되지 않음

****

**Ping of Death(DoS 공격의 대표적인 예로 특정 시스템의 데이터를 오버플로우 발생시켜 가용성 침해를 일으키는 공격 분류 중 하나다.)**

- DoS의 대표적 공격 예시

- 일반적으로 ping 패킷은 ICMP을 이용하여 패킷을 전송하는 형태로 쓰이는 명령

- 전형적인 ping 전송 패킷의 크기는 56 Byte이거나 헤더를 고려할 때 64Byte 크기를 가짐

- 과거 1981년 RFC791 문서에 정형화된 ping 패킷은 최대크기의 데이터를 처리 못하는 시스템이 설계되었으며, 이를 타겟으로 공격을 수행

- 현재는 OS 단계에서 패치가 된 상태로 최신 버전에서는 시스템오류를 발생하지 않음

**공격 원리**

- RFC791 표준에 따르면 IPv4에서 패킷의 최대 크기는 65,535(2의 16승 - 1)Byte이며, 16bit 폭의 헤더 필드 제한

- 2계층의 프레임 크기는 최대 1500 byte로 패킷을 쪼개 전송

- IP 헤더의 오프셋 필드는 13bit까지 표현이 가능하며 오프셋 단위는 8Byte 단위로 나뉨

- 오프셋은 최대로 처리할 수 있는 크기는 65,528((2의 13승 - 1) \* 8) Byte이나 3계층의 IP 헤더(20Byte)를 추가한 크기까지 처리하게 됨(총 65,548Byte)

- 이는 최대 오프셋을 가진 IP 프레그먼트가 모였을 때 패킷이 처리할 수 있는 최대 크기보다 7Byte 초과

- 이 데이터 차이로 인해 버퍼오버플로우 할 수 있으며 이를 통한 문제가 발생할 수 있음

**Wordpress DoS(현재까지 많이 활용되는 오픈 소스이며, CVE-2018-6389는 워드프레스에 해당하는 취약점으로 여러 모듈을 하나의 요청으로 여러 개의 모듈을 호출해 과부하를 발생시키는 공격이다.)**

- 워드프레스는 전세계 도메인에서 29% 활용되는 오픈소스

- 템플릿 시스템을 사용하여 PHP, HTML 코드 수정 없이도 다시 정리할 수 있는 위젯이 포함됨

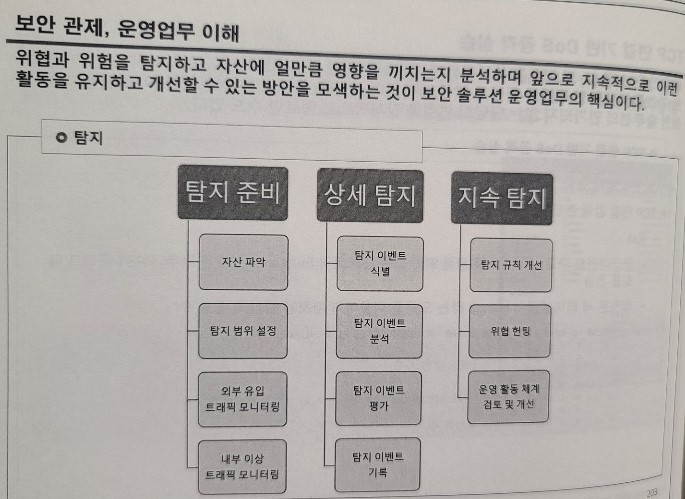
**CVE-2018-6389 원리**

- 관리자 페이지와 wp-login.php 페이지에서도 인증 없이 동작

- 하나의 요청으로 브라우저를 통해 워드프레스 접근 시 다량의 JS/CSS 파일을 여러 개 로드 가능

- 서버 요청으로 모든 JS 모듈을 로드시키도록 요청(약 181개의 I/O 액션을 수행하고 파일 컨텐츠를 제공 받는 형태)

- 응답 패킷은 대부분 4MB로 약 2.2초가 소요되며, 하나가 아닌 여러 요청을 다중으로 보내 서버에게 과부하 발생 시킴



**보안 관제, 운영업무 이해(보안 솔루션 운영의 가장 기본이 되는 활동은 탐지다. 다양한 솔루션에서 발생하는 이벤트를 탐지하고 어떻게 대응할 것인지 고민하는 것이 중요하다.)**

**탐지 준비**

- 자산 파악

-> 회사 내부에 어떤 IT 자산이 있는지 파악

-> 자산의 유형에 따라 적절한 모니터링 및 대응 솔루션 구축

- 탐지 범위 설정

-> 탐지 범위와 대상을 명확히 정의

-- 자산 파악이 우선

-- 파악 후 어느 곳을 어떻게 적절히 탐지할지 고려

--- 탐지 대상을 좁게 잡으면 위협 징후를 놓칠 우려가 있음

--- 탐지 대상을 넓게 잡으면 효율성이 떨어짐

**외부 유입 트래픽 모니터링**

- 네트워크 상태 변화에 대한 인식

-> 트래픽 양 증폭 > DDoS 이상 유무 확인 및 대응

-**내부 이상 트래픽 모니터링**

- 내부에서 외부로 정보가 유출되는 상황 인식

-> 목적지가 외부인데 주민등록번호 형식의 데이터가 트래픽에서 확인되는 경우

- 내부 네트워크에 이상상황이 발생하는 경우

-> 개발자 망에서 인사팀 망으로 트래픽이 유입되는 경우

- 유출 관련 솔루션에서 탐지 알림이 발생한 경우

**상세 탐지**

**탐지 이벤트 식별**

- 어떤 보안 솔루션에서 발생한 이벤트인지 파악해서 적절한 조치 가능

-> WAF에서 발생한 이벤트 > 웹 해킹 대응

- 탐지 이벤트 출발지 및 도착지 파악

-> 아이피, 포트 등 정보를 파악해 어떤 자산에 대한 탐지 이벤트인지 식별

**탐지 이벤트 분석**

- 보안 솔루션에서 발생한 이벤트의 상태를 분석

-> WAF에서 발생한 /etc/passwd 요청 이벤트

> 웹 서버 로그에서 200 OK 확인 > 파일 다운로드 취약점 파악

- 위협과 관련된 이벤트에서 출발지와 목적지 아이피와 포트 파악

-> IDS/IPS 등 보안 솔루션에 패턴 등록하여 대응

**탐지 이벤트 평가**

- 발생한 탐지 이벤트를 평가하여 위험 수준을 산출

-> 위험도 평가 기준을 마련

-> 상대적으로 위험도가 높게 평가되었을 경우 빠른 조치를 수행

-> 시대에 따라 공격 위험도와 신규 공격 기술이 발견되기에 주기적으로 위험도 평가 기준을 재정립

**탐지 이벤트 기록**

- 발생한 탐지 이벤트를 지속적으로 기록

-> 최근에 발생한 이벤트가 기존에 기록된 이벤트와 유사한 내용이 있는지 분석

--> 과거에 놓친 위험 이벤트가 있는지 파악

**지속 탐지(안정된 보안 솔루션 운영업무가 지속되는 것과 회사의 안정성은 비례된다. 내부로 유입되는 위협과 내부에서 발생한 위험뿐만 아니라 타사에서 발생한 사고들까지 분석하여 지속적으로 보안 솔루션에 적용하여 운영한다.)**

**탐지 규칙 개선**

- 탐지 이벤트 기록과 새로 발생한 탐지 이벤트를 토대로 탐지 규칙을 개선

**위협 헌팅**

- 공격으로 밝혀진 다양한 정보를 수집

-> 수집한 데이터를 토대로 자산에 유사한 위협이 있었는지 재평가

-> 재평가에서 놓친 위협을 찾게 되면 다양한 이벤트에서 연관분석을 수행

-> 실제 자산에 영향을 미쳤다면 대응 방안을 수립

**운영 활동 체계 검토 및 개선**

- 탐지에 대한 모든 활동 검토

-> 탐지 정책, 절차 기술 등

-> 개선점 도출, 실무에 적용

**방화벽(네트워크 보안 장비 중 가장 대표적인 장비이며, 내.외부 망을 분리시키고 그 사이에 배치 시켜 정보의 악의적인 흐름, 침투 등을 방지하는 역할을 하는 시스템이다. 네트워크 보안을 위한 방화벽의 선구자는 1980년대 후반에 사용된 라우터로 점차 보안장비로 별도 독립되었다.)**

- 어원 : 건물에 화재가 발생했을 때 더 이상 주변으로 피해가지 않기 위해 연결 경로를 차단하는 벽

- 네트워크를 외부망과 내부망으로 분리시키고 그 사이에 배치 시켜 정보의 악의적인 흐름, 침투 등을 방지하는 시스템

- 보안 관리자가 미리 정해 놓은 보안 정책에 따라 차단하거나 허용하는 기능을 수행하는 소프트웨어 또는 하드웨어 기반 시스템

- 허가된 접근, 서비스, 사용자만을 통과 시키는 정책을 수립하여 외부망으로부터 내부망 보호

**주요 기능**

- 접근 통제(Access Control) : 송신자의 IP 및 포트 번호를 바탕으로 패킷 필터링 수행

- 주소변환(Network Address Translation) : NAT 기능을 이용하여 사설망과 공인망 구분

- 인증(Authentication) : 트래픽에 대한 사용자 신분 증명

- 감사기록/추적기능(Logging/Auditing) : 접속 정보 기록, 네트워크 사용에 따른 통계 정보

- 프록시(Aproxy) : 어플리케이션 계층 필터링, 실제 IP 주소를 감춤

**방화벽**

**1세대 방화벽(패킷 필터(Packet Filter))**

- 첫 번째 네트워크 방화벽의 유형은 “패킷 필터”로 불림.

- 패킷 필터는 네트워크의 패킷 중 IP 주소와 포트를 보며 허용(Allow), 드랍(Dropped), 거부(rejected) 세가지 단계로 구분

- 1세대 방화벽(패킷 필터)은 인터넷상 컴퓨터 간의 패킷을 검사하여 등록된 정보와 매칭 여부를 확인하여 허용과 거부 여부를 판단

- 패킷 필터. IP 주소와 포트를 가지고

- 허용, 드랍, 거부 3가지 단계로 구분

- 드랍, 거부의 차이점 : 드랍은 패킷을 떨구는 것, 아예 차단. 거부는 reject패킷을 보낸다

**2세대 방화벽(상태 필터(Statusful Filter))**

- OSI 7계층의 4계층(전송 계층)까지 해석 가능

- 상태의 판단을 내리기까지 정보를 보유하고 있다가 처리하는 방식

-> 연결의 시작 또는 동일 연결인지를 결정하고 판단

- 4계층에 해당

- 회선 수준 게이트웨이(Circuit level Gateway) 방화벽

-> 구성 가능한 규칙에 따라 양쪽 끝 간의 연결이 유효한지 판단, 이후 허용된 범위와 시간에서만 트래픽을 허용

-> 사설망 데이터를 숨기고 구현하기 쉽도록 별도 프록시가 필요하지 안음

-> 개별 패킷을 필터링 하지 않아 한번 성공하면 공격자가 이점을 취할 수가 있음

- 단점 : 개별 패킷을 필터링 하지 않아 한번 성공하면 공격자가 이점을 취한다.

**3세대 방화벽(어플리케이션 계층(Application Layer))**

- 어플리케이션 계층으로 FTP, DNS, DNS 등의 내용 해석 가능

-> 허용된 포트의 프로토콜 내 악성 행위들도 탐지 가능의 이점 존재

**차세대 방화벽(NGFW(Next-Generation Firewall))**

- 2012년대 이후로 차세대방화벽 등장

- 어플리케이션 스택의 깊고 넓은 기능으로 확장

-> IPS(Intrusion Prevention System: 침입 차단 시스템) 기능

-> 계정 관리 통합 기능

-> 웹 어플리케이션 방화벽(WAF)

**방화벽 유형별 분류**

**네트워크 기반 방식**

- LAN, WAN, 인터넷 등의 환경의 게이트웨이 위치에 존재

- 범용 하드웨어 내 소프트웨어 어플라이언스 장비 또는 하드웨어 기반 어플라이언스 장비의 형태를 가짐

- DHCP, VPN 등의 네트워크 보호 기능 제공

**호스트 기반 방식**

- 시스템 안팎에서 네트워크 트래픽을 제어

- 유닉스의 데몬이나 윈도우의 서비스 등 엔드포인트 보안이거나 에이전트 등의 응용프로그램 형태를 가짐

**분류 형태**

- 네트워크 계층 또는 패킷 필터(Packet filters) :

-> TCP/IP 프로토콜 스택의 상대적으로 낮은 수준에서 패킷이 설정된 규칙과 비교하여 역할 수행

-> 크게 Stateful 형태와 Stateless 형태로 나뉨

-> Stateful 방화벽 : 연결 상태를 저장하여 상태를 포함한 정책 반영을 수행

-> Stateless : 연결 상태를 저장하지 않고 정책을 반영

-> 메모리 용량이 덜 필요함

-> 통신 관계를 기반한 처리를 하지 못하여 세밀한 부분을 놓칠 수 있음

- 어플리케이션 계층 :

-> TCP/IP 단계의 응용프로그램 단계에서 작동하며 응용프로그램 정보를 확인

-> 응용프로그램 내부에 숨어 있는 악성행위 패턴을 감지 차단이 가능(예. 웜, 트로이목마 등)

-> 소켓 콜의 후킹으로부터 어플리케이션 방화벽 역할을 수행

-> 패킷 필터 방식과 유사하지만 포트별로 연결을 필터링하는 대신 프로세스별로 필터링 적용

-> 취약한 서비스를 보호하기 위해 샌드박싱(Sandboxing)이라고 하는 강제적 접근 제어(MAC : Mandatory Access Control)에 의존하는 차세대 응용 방화벽으로 대체되기 시작

- 프록시(Proxy) :

-> 내부 사용자와 외부 사용자 중간에서 중개자 역할을 하는 방화벽

-> 방화벽이 사용자의 요청을 수신하면 내부에서 서버 목적지 주소를 판별하여 프록시가 주소를 변환하고 필요한 경우 추가 액세스 제어 점검 및 로그를 수행한 다음 사용자를 대신해 서버에 연결

-> 내부 사용자가 인터넷 사용 시 웹 브라우저상에 특정 포트를 지정하지 않아도 되는 트랜스 패어런트(Transparent) 프록시와 별도 브라우저 설정이 필요한 정상(Normal) 프록시 존재

-> 응용프로그램 계층 조사 가능하나 지원 응용프로그램이 제한적

-> 호스트의 직접 접근을 막아주는 장점이 존재

-> 나쁜 확장성가 트래픽 성능 저하의 원인인 단점이 존재

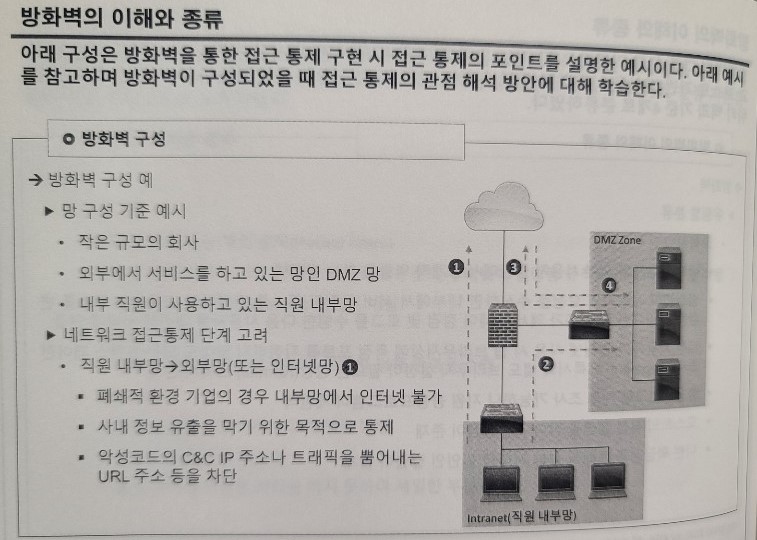
- NAT(Network Address Translation) :

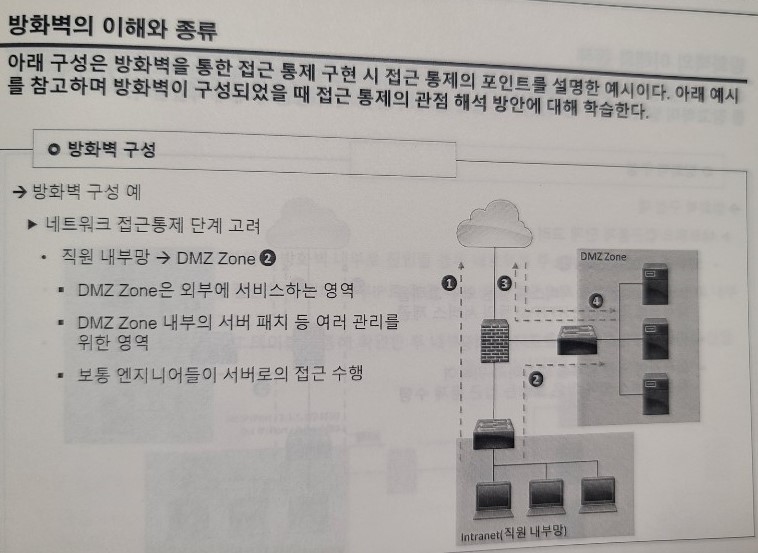
-> 종종 NAT 기능을 포함한 방화벽이 있으며, 호스트들은 방화벽 뒤에 위치해 보호함

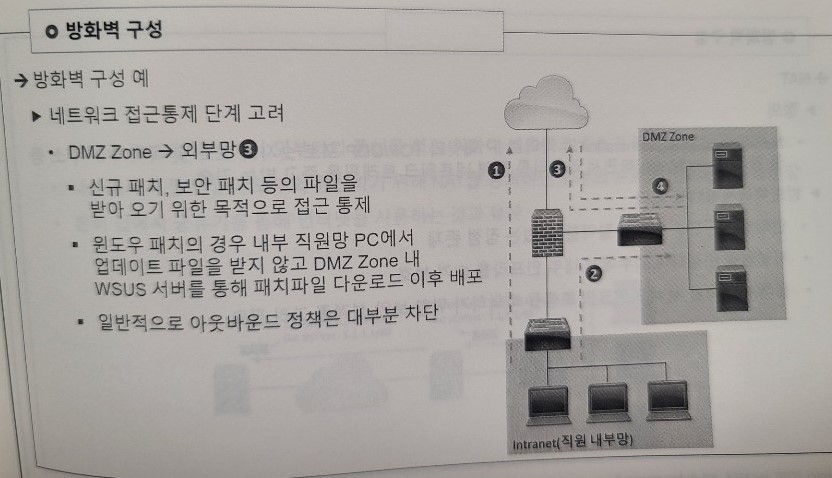
-> 실제 연결된 네트워크르 숨겨주며 안전하게 보호

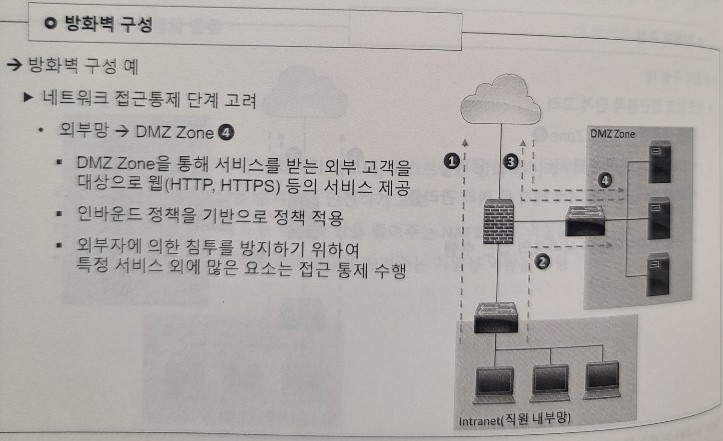
-> 공개 IP 주소를 얻는데 드는 비용과 양을 줄여주는 NAT로써의 역할을 수행

-> 방화벽의 역할 보다는 네트워크 연결을 숨겨주는 기밀성 역할을 수행









**NAT(NAT의 기능의 활용에 따라 효율적인 보안 수준 강화를 수행할 수 있으며, NAT의 특징으로 인해 추적이 어려워 악성 행위에 대한 정확한 대응이 어려울 수도 있다.)**

- Network Address Translation의 약어로 IP 패킷의 TCP/UDP 포트 숫자와 소스 및 목적지 IP 주소 등을 재기록(readdress)하면서 라우터를 통해 네트워크 트래픽을 주고 받는 기술

- 방화벽에서 NAT의 의미 :

-> 인터넷의 공인 IP 주소를 절약할 수 있는 장점 존재

-> 사설망을 구성하여 외부에서 내부 인프라를 숨겨 보호

-> 침해사고 대응 시 네트워크의 흐름을 파악하기 위한 보안 설계를 고려 필요

**DNAT**

- Destination NAT로 외부에서 방화벽 내부로 진입할 경우 네트워크 주소가 변환되는 것을 의미

- 외부에서 내부 서버를 접근하기 위해 외부 IP로 접근하는 것을 기준으로 포트와 IP를 매핑하여 내부 사설 서버로 전달

- 매핑 정보는 방화벽 NAT 테이블을 통해 확인한 후 내부 서버로 재기록(readdress)하여 패킷을 전달

**SNAT**

- Source NAT로 내부에서 외부와의 통신을 하기 위하여 네트워크 주소가 변환되는 것을 의미

- 내부에서 특정 단말이 외부와 통신하기 위해 NAT를 통과하면 외부의 IP로 변경되어 외부에 전달

- 흔히 집에서 공유기를 통해 인터넷을 사용하는 것과 유사

**NAT의 취약성**

- 추적의 어려움

-> 침해사고 발생 시 악성코드가 감염된 단말의 위치를 정확히 찾기가 어려움

-> 코드 밸런싱이 적용된 스위치 환경에서 추가적으로 SNAT가 적용되어 있는 경우 여러 대의 서버가 특정 장비로 트래픽이 흘러갈 때 하나의 IP로 비춰 추적이 어려움

**HA(방화벽의 장애로 인한 서비스 중단으로 내부 인프라의 영향을 받을 수 있기 때문에 HA 개념을 도입되어 서비스 영향도에 미치치 않도록 이중화 기술을 도입한다.)**

- High Availability의 약어로 고가용성 또는 이중화로 알려짐

- 방화벽이 서비스 중단 되었을 때 연결된 다른 방화벽이 활성화 되어 서비스를 백업할 수 있도록 하는 기술

- 고가용성을 위한 방법 :

대칭형 HA : AS(Active-Standby)방식과 AA(Active-Active) 방식 존재. L4 스위치를 기반하여 구성 가능. AS 방식(하나의 방화벽을 메인으로 작업 수행하고 나머지 방화벽 하나를 대기. Active 상태의 방화벽이 페일오버(특정 이벤트로 인해 한 대가 Standby에서 Active 상태로 자동 변환되는 과정)가 되면 Standby 되어 있는 방화벽이 동작). AA 방식(두 개의 방화벽이 로드벨런싱을 수행하면서 모두 활성화. 트래픽이나 리소스를 감안하여 동시에 이용하는 구성)

Fallback : 방화벽의 장비가 언제나 중단 될 수 있다고 가정. L2 스위치를 기반하여 방화벽이 중단되었을 경우에도 통신이 계속 될 수 있도록 구성. 방화벽을 두 대 구매를 해야 하는 비용 부담을 줄일 수 있음. 차단의 기능보다는 가용성에 대해 우선순위 비중을 높인 구조

Full Mesh : L2 또는 L4 스위치의 서비스 중단도 고려한 디자인. 방화벽과 각 스위치를 모두 이중화. 중요도에 따라 장비의 이중화 설계가 다르며 많은 비용이 발생

**Untangle(잘 알려진 오픈 소스 방화벽 종류 중에 하나로 다양한 필터링을 할 수 있는 기능들을 제공하고 있으며, 일부 소프트웨어는 유료로 사용하고 있다.)**

- 네트워크 관리 소프트웨어를 제공.

- 전세계적으로 4만개 조직에서 사용되고 있는 오픈소스 소프트웨어

- 어플리케이션은 아래와 같은 서비스를 제공함(안티 스팸, 컨텐트 필터, 안티바이러스, 안티 피싱, 안티 스파이웨어, 방화벽, OpenVPN, 라우터, 웹 캐시 소프트웨어 등)

- GNU 프로젝트를 기반으로 데비안 계열의 운영체제를 사용

- Untangle에서 제공하는 N.G Firewall의 경우 단순 방화벽 기능의 이상으로 여러 기능을 제공함으로 단순 방화벽이라고 하기보다는 종합적 역할을 수행하는 UTM에 더 가까운 솔루션

**IPS(Suricata) 구축**

- IPS는 외부망과 내부방을 구분하여 바깥을 통신하는 외부를 NAT로 연결하고 내부 통신을 Host-Only 방식을 이용해 구성한다. 이러한 구조가 되었을 때 네트워크 환경 내 하위 단말의 통신할 때 차단 또는 경고 메시지를 전달 할 수 있다.

**포워딩 설정**

- 네트워크 구성 중 외부 네트워크와 내부 네트워크가 분할되는 구성이 되는 가운데 포워딩은 하위 네트워크 환경에서 통신이 이루어질 수 있도록 하는 기능이다. 때문에 포워딩 기능 없이는 하위 단말의 통신이 불가하므로 꼭 수행해주어야 하는 단계이다.

- Suricata는 NFQUEUE 라이브러리를 활용하여 패킷을 확인

**NFQUEUE**

- 커널 패킷 필터로 패킷 큐(대기열)에 대한 접근 권한을 주는 라이브러리

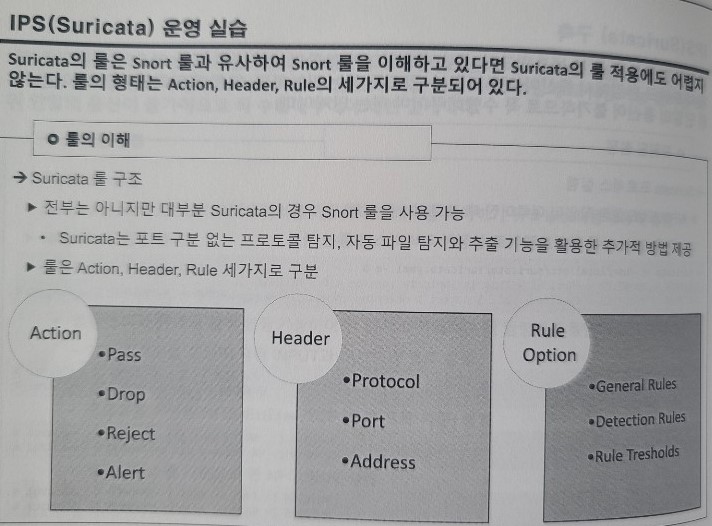
- Suricata는 NFQUEUE 기능을 이용하여 흘러오는 패킷을 확인하고 차단할지 허용할지 정보를 남길지 등의 행동을 판단

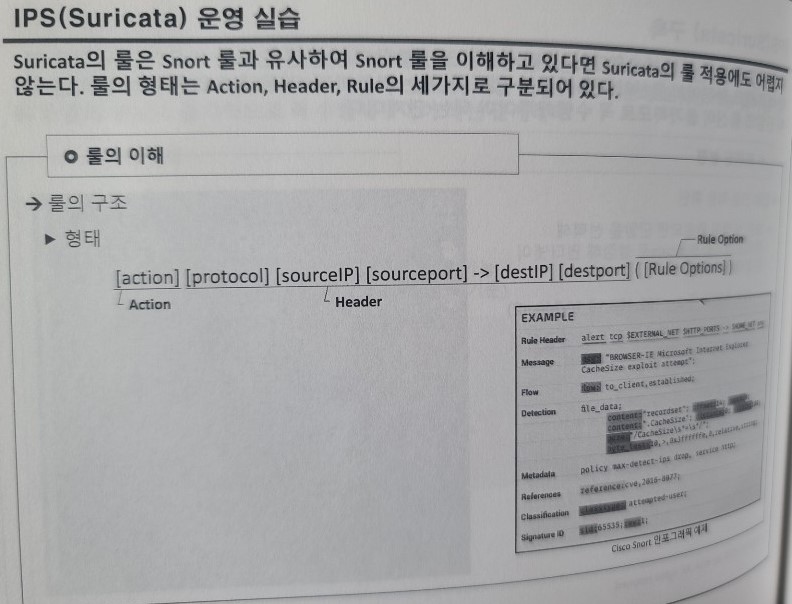
- 즉, Suricata가 활성화되어 있지 않은 경우 NFQUEUE가 패킷 큐를 잡고 있는(홀딩) 상태이기 때문에 내부에서 인터넷 등 패킷이 전달되지 않음(인터넷 사용 불가)

**Masquerade**

- NAT 기반으로 외부망과 내부망을 구분한 후 패킷을 설정되어 있는 경로에 따라 전달하는 역할 수행

- 외부를 통해 전달될 때 내부 IP는 외부 IP로 변경되어 전달





**룰의 구조**

- Action :

-> Pass : 패킷 검사를 하지 않고 통과

-> Drop : 시그너처가 일치하는 패킷을 바로 차단

-> Reject : 시그너처가 일치하는 경우 거부 패킷을 보내 Drop과 동일한 역할 수행

-> Alert : 시그너처가 일치하는 경우 경고 알림을 기록

- Header :

-> Protocol : TCP, UDP, ICMP 등의 프로토콜을 정의하며, 2.0 버전부터 HTTP, TLS, SMB 등 다양한 프로토콜 추가 지원하는 등의 표현을 나타냄

-> Port : 출발지, 도착지 포트를 가리킴

-> Address : IP 주소에 대한 것으로 출발지, 도착지 IP 주소를 가리킴

- Rule Options :

-> 탐지 또는 차단을 하기 위한 세부 옵션들을 설정하고 각 룰에 대한 정보를 제공

-> :으로 값을 정의하고 ;으로 각 룰 옵션을 구분

-> 룰의 분류

--> 일반 룰 옵션(General Rule Options)

---> 규칙에 대한 정보를 제공하지만 탐지 중에 영향을 미치지 않는 옵션

---> 메시지, 트래픽 흐름, 참조, 클래스타입, sid/rev 등 포함

--> 탐지 옵션(Detection Options)

---> 탐지를 위한 옵션으로 Payload, Non-Payload, Post-Detection으로 구분

---> 컨텐츠, 정규 표현식, Byte 테스트(binary 값 탐지)

--> Rule Thresholds

---> 룰 규칙의 일부로 포함되거나 독립형태로 활용되며 임계 값 규칙을 적용할 때 사용

--> 옵션 활용 예

---> Meta information : 탐지되는 룰의 정보를 주거나 우선순위를 부여(일반 룰 옵션)

- 예) msg, reference, sid, rev, classtype, priority 등

---> Payloads : 네트워크 패킷에서 전달하려는 데이터를 검증하는 것으로 시그니처와 비교(탐지 옵션)

- 예) Content, Nocase, Offset, Depth, Distance, within, pcre, Byte Test 등

---> Event Processing : 트래픽의 발생 빈도, 임계치 등의 다양한 이벤트 흐름을 탐지(Rule Thresholds)

- 예) thresholds, track, count, seconds, timeout 등

**로그 종류**

- Drop 또는 Reject 행동을 수행한 로그, 알림 로그, 상태 로그(발생 패킷 수 등), 시스템 로그(로그인, 동작 오류) 등으로 구분

- 로그의 기본 경로 : /usr/local/var/log/Suricata

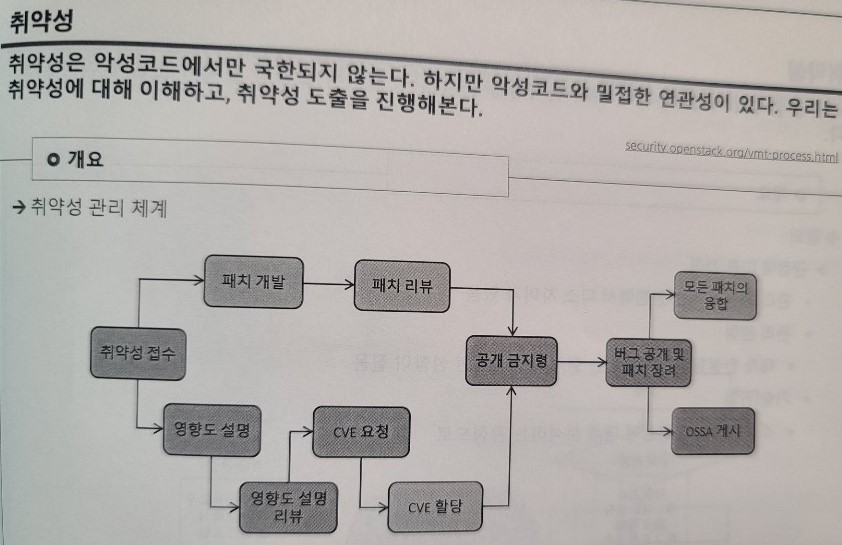
- 로그 파일 :

-> Drop로그 : drop.log

-> 이벤트 로그 : fast.log

-> 상태 로그 : stats.log

-> 시스템 로그 : suricata.log



- ISO 27005 : 하나 이상의 위협에서 악용 될 수 있는 자산 또는 자산의 그룹(회사의 업무를 지원하는 정보 자원을 포함하여 비즈니스 운영과 연속성을 위한 모든 가치) 약점

- 위키피디아 : 보안 취약점 또는 단순히 취약점은 좁은 의미로 컴퓨터의 하드웨어 또는 소프트웨어의 결함이나 체계 설계 상의 허점으로 인해 사용자(특히, 악의를 가진 공격자)에게 혀용된 권한 이상의 동작이나 허용된 범위 이상의 정보 열람을 가능하게 하는 약점. 취약성 = 공격 행위 + 의도 된 행위

**취약성(어떤 관점에서 바로보는가에 따라 취약성은 분리된다. 관리자 관점과 분석가 관점에 따라 다르다.)**

**관점에 따른 차이**

- 관리 점검과 기술관점에서 다소 차이가 있음

- 관리 점검 : 대개 컨설팅에서 사용하는 형태로 정책적인 성향이 짙음

- 기술 관점 : 소프트웨어 취약성에 대해 분석하는 관점으로

**취약점 vs 취약성(분석가 관점에서의 취약점은 다르게 사용될 수 있으며, 관리자 관점에서는 취약성보단 취약점으로 주로 언급되어 진다.)**

- 엄밀히 구분하면 단어 구성의 의미적 차이가 있지만, 대부분 취약점으로 사용하고 있음

- 수직적 / 수평적 구분

-> 수직적 구분 : 특정 기업의 자산이나 환경에서 발생하는 보안 위협 -> 취약점

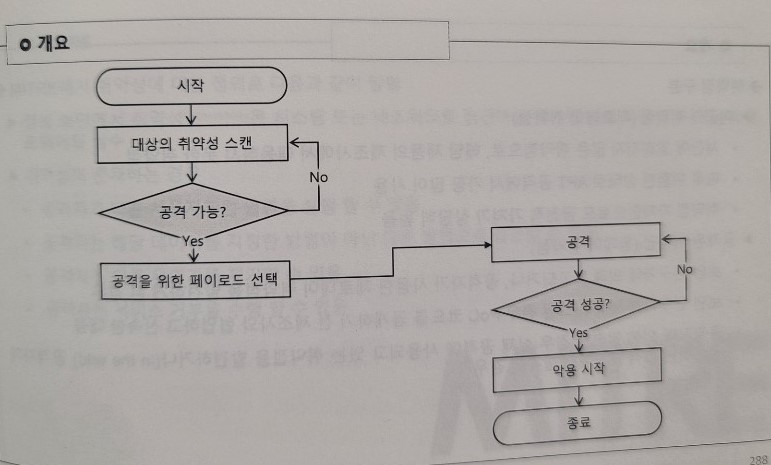
-> 수평적 구분 : 여러 기업에서 사용하는 공통적인 요소에서 발생하는 보안 위협 -> 취약성

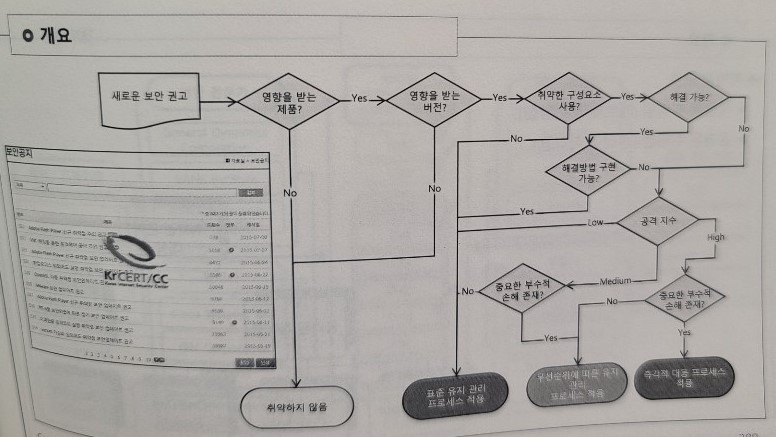
- As-ls / To-Be 구분

-> As-ls : 현재 공격에 악용될 수 있는 상태 -> 취약점

-> To-Be : 언젠가 공격에 악용 될 수 있는 상태 -> 취약성

예) 데이터베이스에 소프트웨어 보안적 취약점이 발견되었으나 보안 패치가 진행되지 않아 취약성으로 분류된다.





**취약성(제로데이(미공개)와 원데이(공개된)으로 구성되며, 최근 다양한 형태의 공격 중에 가장 많이 사용하는 형태이다.)**

- 미공개 취약점(제로데이 취약점)

-> 세간에 알려지지 않은 취약점으로, 해당 제품의 제조사에서 대응하지 못한 취약점

-> 매우 위험한 상태로 APT 공격에서 가장 많이 사용

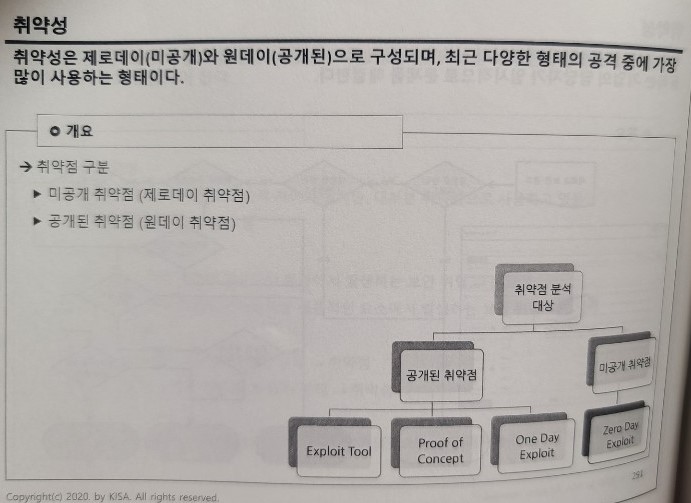
-> 취약점 자체만으로도 금전적 가치가 상당히 높음

- 공개된 취약점(원데이 취약점)

-> 보안 연구가에 의해 보고되거나, 공격자가 사용한 제로데이 취약점을 발견하게 된 경우

-> 보안 연구가에 의해 보고된 경우 PoC 코드를 공개하기 전 제조사와 협업하고 신속한 대응

-> 공격자에 의해 알려진 경우 실제 공격에 사용되고 있는 취약점을 발견하거나(In the wild) 공격자가 사용하는 공격 도구에서 발견된 경우



**MITRE(마이트레)(산,학,연을 구성하여 다양한 사이버 공격과 방어에 대해 정의하고 새로운 패러다임을 만들어 냈다.)**

- 정보 보안에서 취약성(Vulnerability)은 시스템 또는 네트워크로 접근하는 해커에 의해 이용 될 수 있는 소프트웨어의 실수

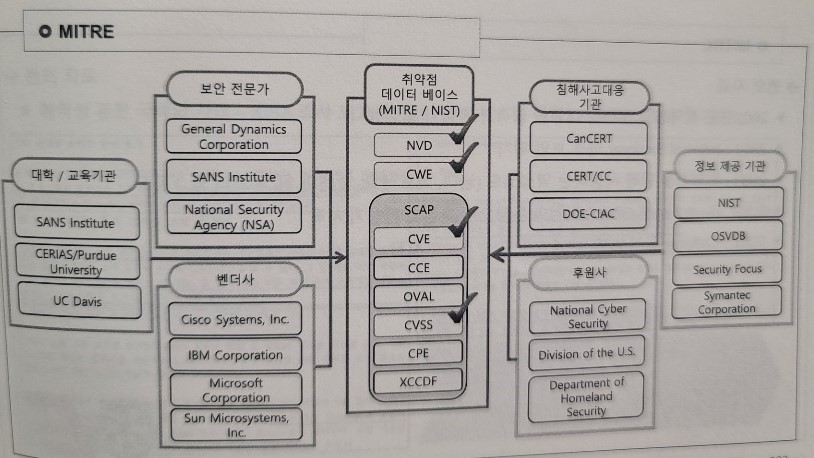
- 취약성이 존재하는 경우

-> 공격자가 다른 사용자처럼 명령을 실행 할 수 있음

-> 공격자는 해당 데이터를 지정한 방법이 아닌 다른 방법으로 접근할 수 있음

-> 공격자는 다른 무언가를 제기할 수 있음

-> 공격자는 서비스 거부를 수행 할 수 있음



**취약성에 고유 식별자를 붙임으로써 분석가, 관리자 등 정보를 쉽게 검색하여 관리할 수 있다.**

**CVE**

- CVE는 Common Vulnerabilities & Exposure의 약자로 공통 취약성 식별자라 부른다. 편하게 CVE 스키마, CVE 이름, CVE 숫자, CVE 엔트리 등으로 불린다.

- CVE는 MITRE에서 만든 것으로 1999년도 설계했다. 본격적으로 사용하기 시작한 시기는 2002년으로 미국 국립 표준 기술 연구소(NIST)에서 국가 취약성 데이터베이스(NVD)를 구축하고 협력체계를 구축한 시기이다.

- CVE 코드가 붙을 수 있는 요소는 “소프트웨어 취약성”이다. 소프트웨어는 하드웨어의 반대 개념으로 “컴퓨터가 읽고 쓸 수 있는 모든 것”으로 요약할 수 있다.

- 대표적인 예로 펌웨어, 미들웨어, 운영체제, 응용 프로그램 등 이며, 그렇기에 펌웨어 취약성, 미들웨어 취약성, 운영체제 취약성, 응용 프로그램 취약성 등 모두 CVE 코드를 받을 수 있다.

- CVE 스키마는 고유하기에 검색하여 관련 정보들을 쉽게 찾아 볼 수 있다.

**CVSS**

- Common Vulnerability Scoring System 약자로 취약성의 위험성을 정량화된 수치화 시키기 위해 만들어졌다.

- CVE는 1999년에 개발되었고, CVSS는 2005년에 개발되었다.

- 개발된 이유는 발견된 취약성에서 우선순위를 두어 대응의 순서를 두기 위함이다.

- CVSS는 1989년에 설립된 국제 침해사고대응 팀(FIRST)에서 개발하고 운영하고 있다.

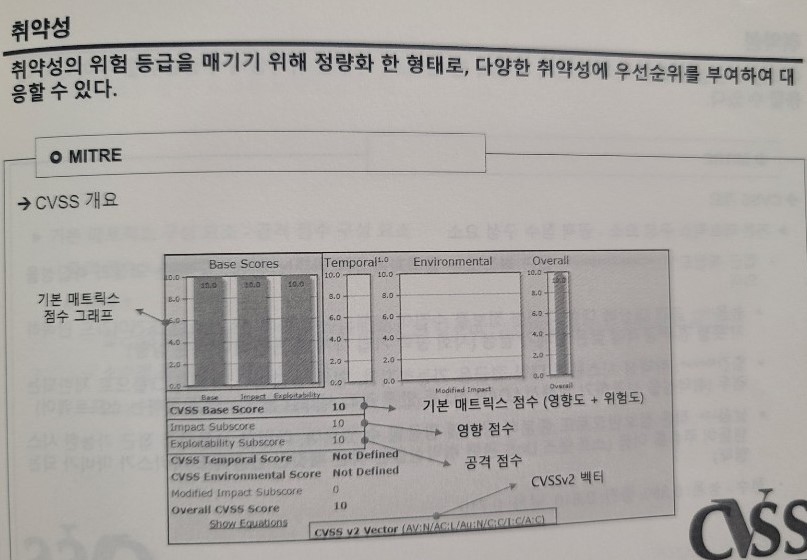
- 가장 많이 쓰는 버전 2

- 세 가지 매트릭스를 사용하는데, 가장 중요한 점수는 기본 매트릭스이다.

- 기본 매트릭스 : 접근 경로, 접근 복잡도, 인증 필요 여부, 기밀성, 무결성, 가용성

- 임시 매트릭스 : 공격코드 공개 여부, 취약성 패치 단계, 취약성 신뢰도

- 환경 매트릭스 : 피해, 취약성 영향 시스템 분포도, 기밀성 요구사항, 무결성 요구사항, 가용성 요구 사항



**CVSS**

- 기본 매트릭스 구성 요소 – 공격 점수 구성 요소

-> 접근 경로(AV-Access Vector) : 접근 경로는 공격 수행 위치로도 해석, 공격자가 호스트에서 멀어질수록 더 높은 취약성 점수를 얻음

--> 로컬 : 로컬접근으로 취약성 공격이 가능한 것으로, 일반적으로 권한상승 취약성이 여기에 해당

--> 인접 네트워크(Adjacent Network) : 데이터링크 계층에서 공격이 가능한 것으로, 도메인 스푸핑, ARP 스푸핑, 블루투스 공격 등 여기에 해당

--> 네트워크 : 네트워크 계층에서 공격이 가능한 취약성으로, 일반적으로 원격 코드 실행 취약성, DDoS 취약성 등이 여기에 해당

- 점수 – 로컬: 0.395, 인접 네트워크 : 0.646, 네트워크 : 1.000

- 기본 매트릭스 구성 요소 – 공격 점수 구성 요소

-> 인증 : 인증은 말 그대로 취약성을 발생시키기 위해 접근하는 과정에서 대상이 인증하는 형태를 의미

--> 복수 : 인증하는 과정이 2회 이상인 경우에 해당한다.

--> 단수 : 인증하는 과정이 1회인 경우에 해당한다.

--> 없음 : 인증 과정이 없는 경우에 해당한다.

- 점수 – 복수 : 0.450, 단수 : 0.560, 없음 : 0.704

- 기본 매트릭스 구성 요소 – 공격 점수 구성 요소

-> 공격 점수 = 20 x 접근 경로 x 접근 복잡도 x 인증

- 기본 매트릭스 구성 요소 – 영향 점수 구성 요소

-> 기밀성 영향도(C-Confidentiality Impact)

--> 악용되는 취약성이 기밀성에 얼만큼의 영향을 미치는가를 측정

--> 기밀성은 권한이 있는 사용자에게만 정보 접근 및 공개를 하고, 권한이 없는 사용자에게는 접근을 제한함으로써 정보의 비밀을 얼만큼 유지할 수 있는가를 의미

-> 무결성 영향도(I-Integrity Impact)

--> 악용되는 취약성이 무결성에 얼만큼의 영향을 미치는가를 측정

--> 무결성은 정보의 정확성, 신뢰성, 일관성 등을 보장하는 것을 의미

-> 가용성 영향도(A-Availability Impact)

--> 악용되는 취약성이 시스템의 가용성에 얼만큼의 영향을 미치는가를 측정

--> 가용성은 시스템이 사용 가능한 정도를 의미하며 쉽게 표현하면 가동률로 이해

- 기본 매트릭스 구성 요소 – 영향 점수 구성 요소

-> 점수

--> 없음 – 0 : 기밀성, 무결성, 가용성에 어떠한 영향도 없는 취약성을 의미

--> 부분 – 0.275 : 일부분 기밀성, 무결성, 가용성에 손상을 주는 취약성을 의미

--> 완벽 – 0.660 : 기밀성, 무결성, 가용성에 심각한 손상을 주는 취약성을 의미

**최종 점수**

­- 기본 점수 = (공격 부분 점수 x 0.4) + (영향 부분 점수 x 0.6)

-> C.I.A에 의해 생성되는 점수가 접근 경로, 접근 복잡성, 인증 보다 20% 더 가중

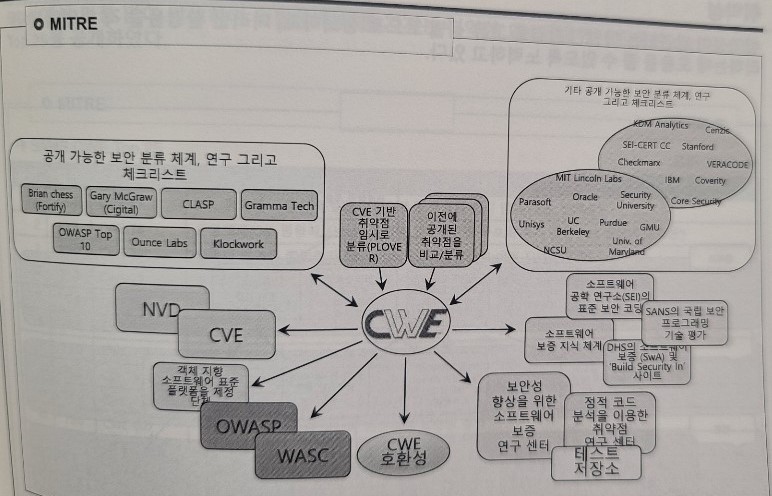
-> 6개의 개별 구성요소 점수는 소수점 3자리로 사용, 계산 결과 점수는 소수점 2자리에서 반올림

- 위험도 범위

-> 버전 2에서는 명확한 범위가 정해져 있지 않음

-> 10점에 가까울수록 위험한 취약성

-> 버전 3에서는 [취약하지 않음(0.0), 조금 취약함(0.1 – 3.9), 취약함(Medium 4.0 – 6.9)]



**CWE**

- CWE는 계층형 구조를 가짐

- 상위 조건이 만족해야 하위 조건이 발현 가능하다는 형태를 가지고 있음

- 구조

-> 각각의 CWE에 아이콘으로 표기하여 정보를 제공

-> 뷰 : 개발자, 연구자, 개발 언어(C, Java 등)

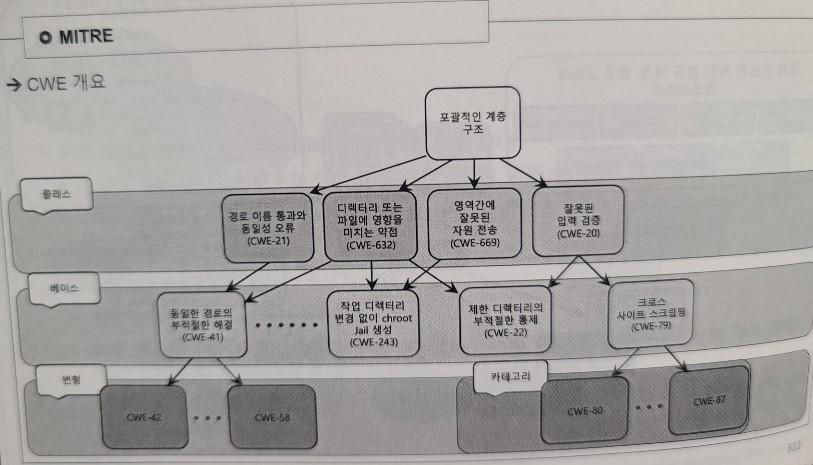
-> 카테고리 : 공통적 특성을 가지는 약점 유형들을 그룹화

-> 약점 : 개별 취약점을 의미

--> 클래스 : 매우 취약점을 의미

--> 베이스 : 추상적인 보안 약점

--> 변형 : 구체적인 보안 약점





**TTPs 방법론**

- 여러 해킹 기법들의 진화로 사이버 보안 팀 내에서 새로운 전략, 방법론이 제시됨

- TTPs는 Tactics, Techniques and Procedures로 공격 기법에 대한 사이버 위협 인텔리전스로 활용

- 특정 위협 행위자 또는 집단과 관련된 활동, 방법, 패턴을 의미함

- TTPs 구분의 예

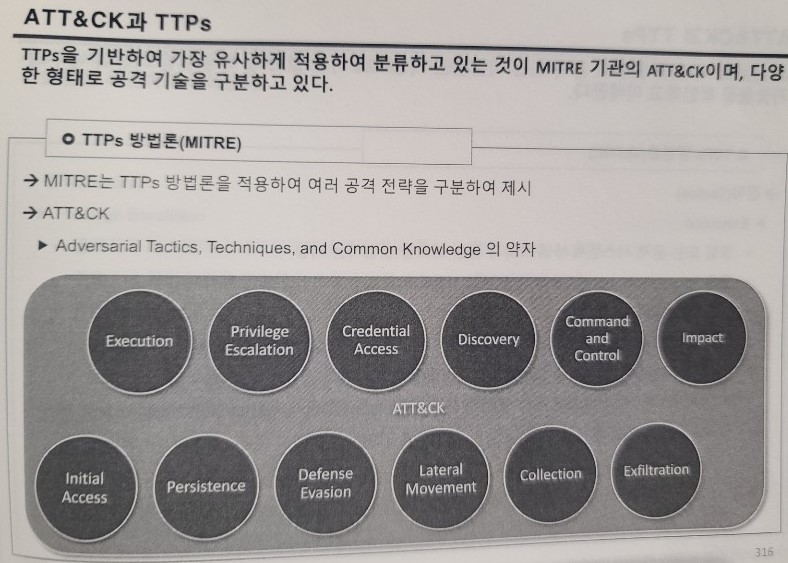
-> 특정 악성코드 집단의 특성 분석

-> 특정 악성코드 변종 예

-> 특정 공격 패턴

-> 공격자가 사용하는 인프라

-> 피해자 공격 대상



**전략(Tactics)**

- Initial Access

-> 네트워크 내에서 초기 발판을 얻기 위해 공격자들이 사용하는 공격 벡터를 의미

-> 전술(Techniques)

--> Valid Accounts : 공격자는 Credential Access(신뢰도 접근)을 이용하여 서비스 계정의 증명을 훔처 초기 접근 권한을 획득하는 기술

--> Spearphishing link : 스피어 피싱의 변종으로 이메일 내 첨부파일이 자체 차단하는 것을 대신하여 악의적인 링크를 포함하여 보낸다는 점에서 차이가 존재하는 기술

--> Supply Chain Compromise : 데이터 또는 시스템을 손상시키기 위해 최종 소비자가 제품을 수령하기 전에 제품 또는 전달 메커니즘을 조작하는 기술

- Execution

-> 로컬 또는 원격 시스템에서 공격자가 만든 코드를 실행하는 기술

-> 종종 lateral movement 기술과 혼용하여 사용되기도 함

-> 전술(Techniques)

--> Execution through API : 시스템 API를 통하여 바이너리 파일을 실행하는 공격

--> Command-Line Interface : 운영체제 플랫폼의 많은 명령어를 이용한 공격 기술

--> PowerShall : 윈도우 운영체제 내 존재하는 Command-Line Interface와 스크립트를 이용한 공격

--> Source : 명령 쉘 또는 실행 파일을 이용하여 공격하는 기술

- Persistence

-> 시스템에 대한 접근, 조치, 구성 변경을 이용하여 지속적으로 존재하는 공격

-> 시스템 재시작, 접근 실패 등으로 인한 시스템 재접근이 가능하도록 재실행 또는 백도어로 대체

-> 전술(Techniques)

--> External Remote Services : VPN과 같이 외부로부터의 연결 서비스를 허용하는 기술

--> Hypervisor : 운영시스템 내 하이퍼바이저 기능을 활용하여 루트킷의 기능으로 사용하는 기술

--> Modify Existing Service : VPN과 같이 외부로부터의 연결 서비스를 허용하는 기술

--> Registry Run Keys / Start Folder : 시작프로그램을 등록하여 재시작시 자동 실행하는 기술

--> Web Shell : 웹서버 내 웹 쉘을 등록하여 원하는 시점에 웹 쉘을 통해 시스템에 접근하는 기술

- Privilege Escalation

-> 공격자가 시스템이나 네트워크에서 더 높은 수준의 사용 권한을 얻는 공격

-> 주로 시스템의 약점을 이용하여 로컬 관리자 또는 시스템의 권한을 얻음

-> 전술(Techniques)

--> Bypass User Account Control : 사용자 접근 통제를 우회하여 상위 권한을 획득하는 기술

--> DLL Injection : 실행되는 프로그램이 불러오는 DLL의 값을 수정하여 상위 권한을 획득하는 기술

--> Exploitation of Vulnerability : 취약점을 이용한 침투를 통해 상위 권한을 획득하는 기술

--> Setuid and Setgid : 유닉스 계열 운영체제 중 Setuid, Setgid 권한을 이용 상위 권한 획득하는 기술

- Defense Evasion

-> 시스템을 방어하기 위해 설치된 시스템의 탐지를 회피하거나 방어를 우회하는 공격

-> 방어 우회로 이득이 생기는 다른 범주들과 유사함

-> 전술(Techniques)

--> Bypass User Account Control : 사용자 계정 접근 통제 우회를 통해 시스템에 접근하는 기술

--> Binary Padding : 실행 파일의 변경을 이용해 변경된 파일 해쉬 값 탐지 방어를 우회하는 기술

--> Rootkit : 시스템의 API를 수정하여 악성코드가 보이지 않도록 하는 기술

--> Hidden Files and Directiories / Users / Window : 파일, 디렉토리, 사용자, 윈도우 등 숨기는 기술

- Credential Access

-> 상업 환경에서 사용되는 시스템, 도메인, 서비스 자격 증명에 접근, 통제하는 기술

-> 공격자는 관리자 또는 사용자 계정으로부터 정상적인 자격 증명을 얻으려고 시도함

-> 정상적인 자격증명 과정으로 탐지가 어려움이 존재

-> 전술(Techniques)

--> Brute Force : 무작위로 값을 대입하여 패스워드 정보를 탈취하는 공격 기술

--> Create Account : 시스템에 충분한 권한이 있는 계정을 생성하여 공격자가 쉽게 접근하는 기술

--> Exploitation of Vulnerability : 취약점을 이용하여 권한이 있는 정상적인 계정에 접근하는 기술

--> Network Sniffing : 네트워크 정보를 모니터하여 패킷 정보 내 계정 정보를 탈취 후 접근하는 기술

--> Private Keys : 시스템 내 계정 정보 또는 키로깅을 통하여 계정 정보 탈취 후 접근하는 기술

- Discovery

-> 공격자가 시스템에 대한 지식을 얻을 수 있는 기술

-> 시스템 내에 있는 자산의 가치를 파악하고 목표 방향을 잡을 수 있는 도움을 제공

-> 전술(Techniques)

--> Account Discovery : 시스템 또는 도메인 계정의 리스트를 추출하는 공격 기술

--> File and Directory Discovery : 시스템 내 파일과 디렉토리 리스트를 추출하는 공격 기술

--> Network Service Scanning : 시스템 내 어떤 네트워크 서비스를 제공하는지 추출하는 공격 기술

--> Remote System Discovery : 원격 시스템 정보를 추출하는 공격 기술

--> System Time Discovery : 산업 환경 내 동기화 시간 정보들을 추출하는 공격 기술

- Lateral Movement

-> 원격 접속 도구 없이 원격에 위치한 시스템의 네트워크를 이동하여 공격 또는 탐색 하는 기술

-> 공격자와의 실시간 연결이 아니기 때문에 네트워크 전역의 불필요 행위를 할 수도 있음

-> 전술(Techniques)

--> Logon Scripts : 시스템에 로그인 시 실행하는 스크립트를 이용한 기술

--> Remote Service : 공격자는 터미널 서비스에 로그인하는 사용자를 통해 공격하는 기술

--> Windows Admin Share : 관리자만 접근 가능한 숨겨진 네트워크(C$, ADMIN$)를 통한 공격 기술

--> Windows Remote Management : winrm와 같은 명령어를 윈도우 서비스 프로토콜을 이용한 기술

- Collection

-> 민감 데이터, 파일 정보를 식별하고 수집하는데 사용되는 기술

-> 공격자가 정보를 검색할 수 있는 시스템 또는 네트워크 인프라를 정보 수집을 포함

-> 전술(Techniques)

--> Audio Capture : 컴퓨터 장치 또는 어플리케이션 내부의 오디오 녹음 기능을 이용한 도청 기술

--> Clipboard Data : 클립보드 내에 있는 정보를 수집

--> Data from Local System / Network Shared Drive / Removeable Media : 로컬 시스템, 네트워크 공유 드라이버, 삭제된 미디어 데이터를 식별 및 수집하는 기술

--> Email Collection : 이메일 내 데이터를 식별 및 수집하는 기술

--> Input Capture / Screen Capture : 입력 값(Ex. 키로깅) / 스크린 샷 정보들을 수집하는 기술

- Command and Control

-> 대상 네트워크 내에서 상대방이 공격자 통제 내에 있는 시스템과 통신하는 기술

-> 시스템 구성 및 네트워크 방식에 따라 다양하고 은밀한 수준의 명령, 제어 설정 기술은 다수 존재

-> 전술(Techniques)

--> Commonly Used Port : 공격자는 커뮤니케이션을 위해 방화벽, IDS 등의 탐지 또는 차단을 우회하는 포트를 사용하여 공격 하는 기술

--> Connection Proxy : 시스템 간에 네트워크 연결하는 중개기를 통해 명령을 전달하는 기술

--> Remote File Copy : 한 시스템에서 다른 시스템으로 복사되어 도구나 다른 파일을 준비하는 기술

- Exfiltration

-> 공격자가 네트워크에서 데이터를 훔치기 위해 사용하는 기술로 구성

-> 탐지를 피해 데이터 압축 또는 암호화 등의 기술을 사용하여 탈취

-> 전술(Techniques)

--> Data Compressed : 데이터가 유입 차단이 되기 전 데이터를 분할 하거나 명령을 압축하는 기술

--> Data Encrypted : 탐지 우회 또는 노출을 막기 위해 데이터를 암호화 하는 기술

--> Exfiltration Over Alternative Protocol : 메인 프로토콜이 아닌 다른 프로토콜을 이용하는 기술

- Impact

-> 비즈니스 및 운영프로세스를 조작하여 공격자가 가용성을 방해하거나 무결성을 손상 시키는 전략

-> 공격자들이 최종 목표를 위해 수행하거나 기밀성 침해 공격을 덮기 위해 사용할 수 있음

-> 전술(Techniques)

--> Network Denial of Service : DoS 공격을 수행하여 사용자의 대상 자원의 가용성을 저하시키거나 차단하는 공격을 수행

--> Data Encrypted for Impact : 특성 대상 또는 많은 양의 대상 시스템들의 데이터를 암호화하고 복호화 키에 대한 접근을 보류하여 데이터를 접근할 수 없도록 수행

--> System Shutdown/Reboot : 공격자가 임의로 시스템을 종료 시키거나 재부팅

**어플리케이션 보안(운영체제 내에서 범용적으로 사용되는 프로그램)**

- NCS에 따르면 어플리케이션으로부터 오는 위협을 보호하기 위해 보안 솔루션을 도입 또는 운영하여 보안 수준을 강화하는 목적으로 의미

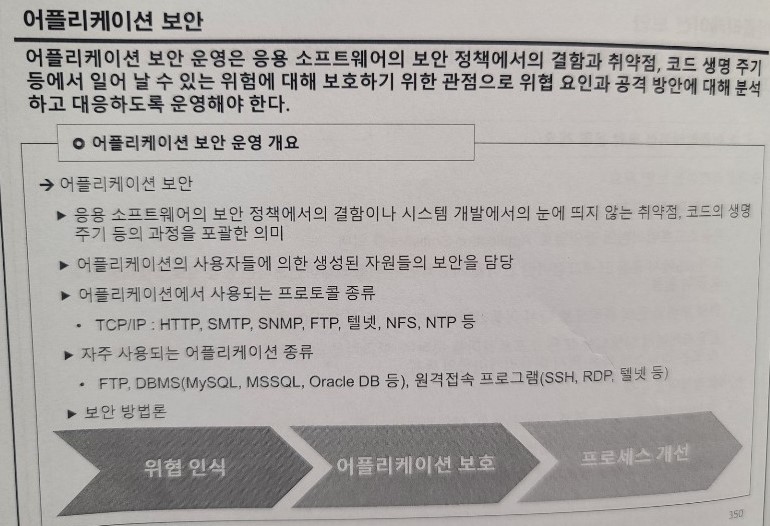
- 어플리케이션을 보호하는 방법은 솔루션 외에 2가지 형태의 방법으로 보호가 가능

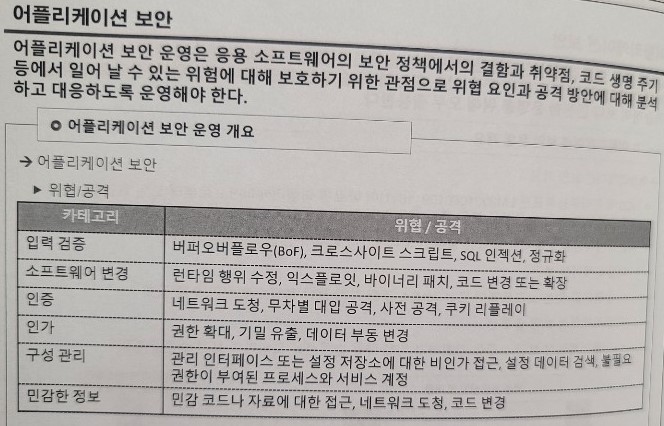
-> 어플리케이션 보안

--> 운영되고 있는 소프트웨어 또는 서비스 등의 자체 보안 설정으로 보안 수준 강화

-> 어플리케이션 보안 솔루션

--> 어플리케이션 보안 수준을 강화하기 위해 다른 솔루션을 도입하여 보안 수준을 강화







**어플리케이션 보안 운영 개요**

- 가이드 내용에는 아래와 같은 범주로 크게 분류

- 계정 관리 : 원격 접속이 가능한 계정에 대한 보안 설정

- 파일 및 디렉토리 관리, DB 관리

- 접근 통제

- 서비스 관리 : 불필요한 서비스의 비활성화. 서비스의 보안 설정 관리

- 패치 관리 : 노후화된 버전의 서비스 패치

- 로그 관리 : 비정상적인 접근 등의 이상 접근을 추적하고 추후 공격에 대비

- 보안 관리, 옵션 관리



