정보보호

사이버보안과 인공지능

보안을 위한 AI

AI 안티바이러스, AI보안관제

AI를 위한 보안

적대적 공격방어, 학습데이터 유출 방지. AI를 안전하게 할 방법도 필요하다.

AI를 이용한 악성코드 분석도 잘 고민해서 사용해야 한다. 패킹 등을 씌워 배포한 악성코드 등은 검출하지 못한다는 논문이 발표되어있음.

정보보호

악성코드 역사

- 스턱스넷

  망 분리로 모든 문제 해결 불가능

경계 보안으로 인해 웹 서버등 직접 공격은 매우 어렵다. 때문에 임직원들을 감염시킨 이후에 내부 장비를 감염시킨다. 안에서 밖으로 나가는것은 주로 막기 어렵기 때문. **C&C server** 기억해야 한다.

내부망에서 내부망 안으로 이동하는 것을 **Lateral movemen**t라고 한다. 내부에 들어온 악성 코드는 이렇게 내부에서 수평적으로 이동한다.

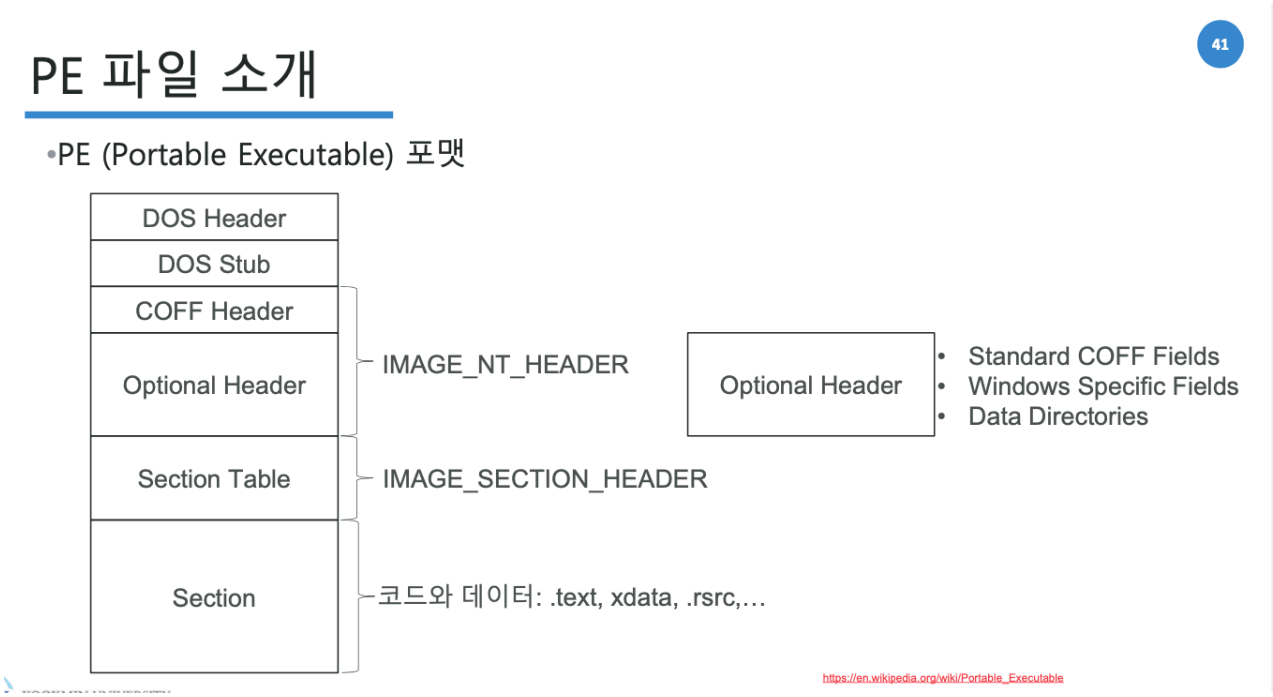
먼저 노트북 감염. CNC서버로 감염된 녀석들이 접근. 러터럴 무브먼트를 통해 이동. **Data Exfiltration.**

이러한 공격을 **APT공격이라고** 한다. 지능형 지속공격

- 제로 트러스트 (zero trust)

  누구도 믿으면 안된다는 것.

PE 파일



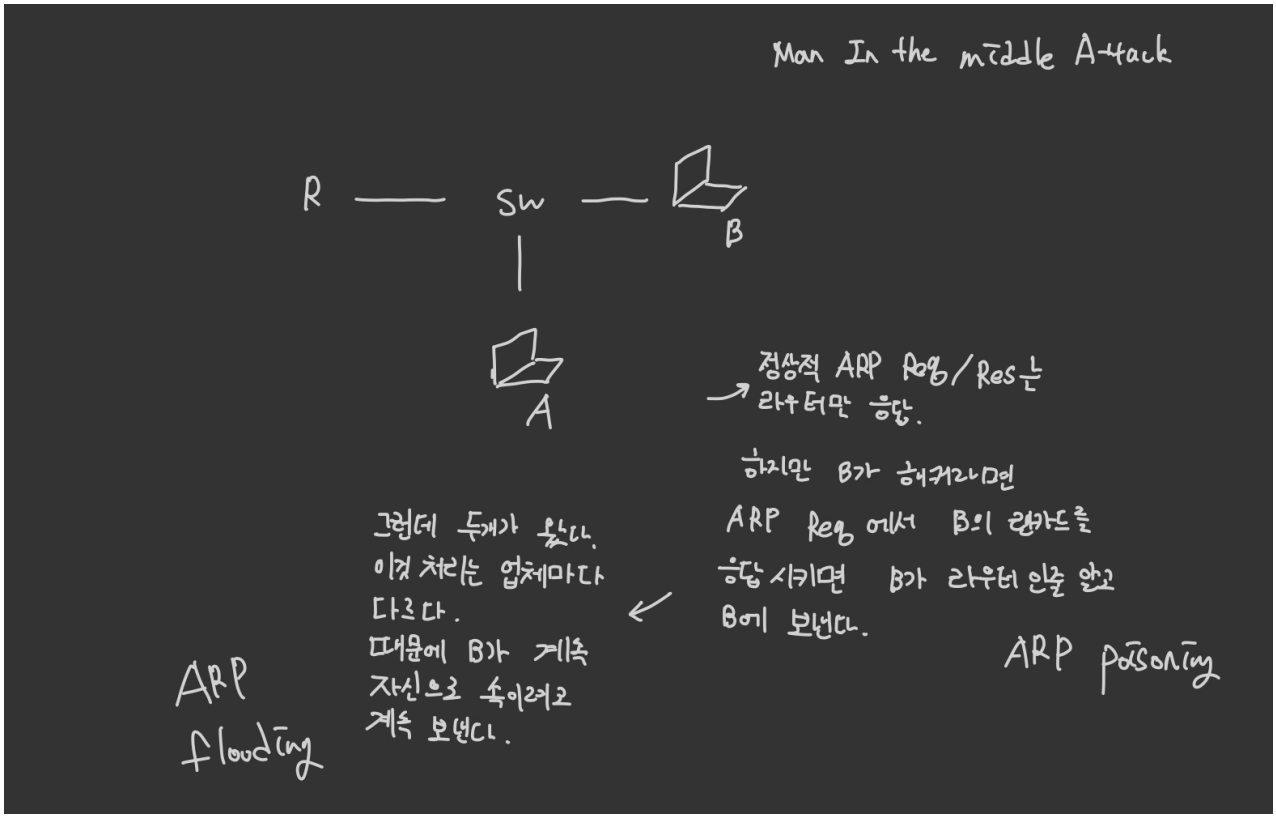
코프헤더부터 악성코드를 분별할 수 있는 좋은 정보가 나타난다.

정보보호

ARP(이더넷 프레임 레이어에서)

> Spoofing : 속이다. A가 아닌데 A인척 한다. 앞에 다른것이 붙을 수 있다, ex) ip spoofing, 등..
>
> sniff : 엿듣는것(도청)

랜에서 왔다갔다하는 이더넷 패킷을 엿듣게 하는것을 ARP spoofing이라고 한다. 트래픽을 멈추게 할수도, 의도적으로 패킷을 더 넣어 보낼수도 있다. ARP poisoning, ARP flooding, ARP poison routing 이라고도 한다.

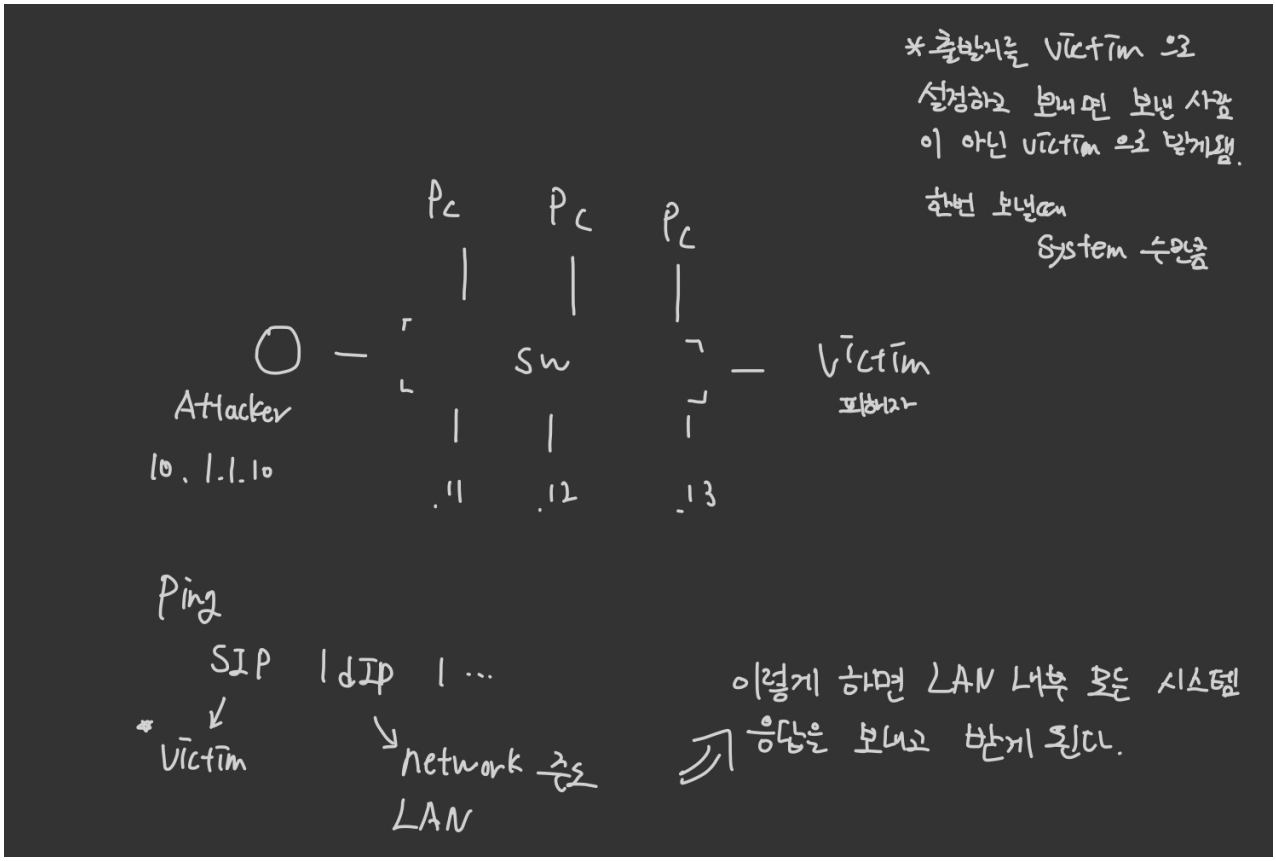


Smurf attack

패킷을 많이 보내면 해가 될 수 있음을 의도.

> ICMP(internet control message protocol) 인터넷 컨트롤 메세지인데, 죽었는지 살았는지 확인하는 그런 것.

만일 해커가 패킷 보낼때 출발 주소를 속여서 보내면 서버가 응답할때 도착 주소를 다른곳에 보내게 되어 낭비시킬 수 있다.



랜에서만 가능한 것은 아님. 안쪽으로 패킷을 넣을수만 있다면 외부에서도 충분히 가능하다.때문에 일반적으로 외부에서 들어오는 패킷은 ICMP를 막게 하기도 하고 ping에 대해 모두 응답하지는 않는 등 조치를 취한다.

MAC Flooding

이전 시대 hub사용 당시는 모든 패킷을 복사해 모든곳에 보내는 형식이라 공격자가 패킷을 보기만 하면 됐었지만, 현재는 스위치를 사용하기 때문에 목적지에만 보낸다.

이때 스위치는 맥주소와 아이피주소를 캐시로 가진다. 만약에 스위치에 있는 캐시 메모리가 쓰레기 값으로 가득 채운다면 ? 공격자가 의미없는 패킷을 생성해서 마구 보낸다. 이렇게 해서 캐시를 더미값으로 꽉 채우면 스위치는 옛날 방식인 허브방식으로 동작한다. 이때는 브로드캐스팅으로 다시 보내기 때문에 공격자는 패킷을 전부 볼 수 있다.

단 이것은 치명적 단점이 있는데 인터넷이 느려진다.

해결 방법은 IP:MAC쌍을 지정된 것만 허가하는 것이다. ARP watch도 좋은 방식.

IP spoofing

중요. 먼저 소켓프로그래밍 부터. `sock_raw` 값을 넣으면 tcp/udp,ip등 변경할 수 있다. 이것을 해커가 노린 것인데, 출발지 주소를 바꿔서 보냈다.

Dos

공격자가 한명. SYN flooding.

> SYN flooding : 리눅스 서버에 웹을 운영한다고 하자. 특수한 공격도구들이 있다. 매우 많은 패킷을 계속해서 보낸다.그런데 이때 출발지 ip에 대해 서버가 너무 많이 받으면 그것을 막을 수 있다. 이때문에 공격자는 출발 ip를 랜덤하게 row-socket 프로그래밍 하는 것이다. 이렇게 보내면 막기가 어려움.

DDoS

공격하는 호스트가 여러대. Smurf.

정보보호

IP spoofing

Ingress/egress filtering

Ingress

밖에서 안으로 들어오는 것을 막는 것. 예를들어 밖에서 들어오는 패킷의 ip가 내부망 주소이다? 이것은 해킹이므로 막는다.

egress

안에서 밖으로 나가는 것을 막는 것. 내부 패킷이 나갈때 확인을 하는데 출발지 ip가 내부망 주소가 아니고 이상한 주소라면 이상한 패킷이므로 막는다.

라우터에 이 ingress/egress 검열을 한다면 부하가 간다. 때문에 ingress 먼저 적용한다.

TCP session hijacking

> 비행기 납치 하는것을 hijacking.

TCP 3 handshake 에서 세션이 맺어졌는데 이 세션을 통째로 가져가 버리는 것. TCP sequence number가 있다. A가 서버 B는 클라이언트, 공격자 E가 있을때 E가 먼저 ip spoofing을 한다. B가 사용하는 ip주소로 바꾼다. 이제 E가 A에게 접속한다. 패킷이 왔다갔다 할때 출발 주소가 B이므로 B인줄 안다. 받은 패킷에 대해 응답할때 B로 가는데 E는 B에게 디도스 공격 등으로 죽여놓는다. 그러면 B는 메시지를 받지 못한다. E는 이 메세지를 추측하거나 가져오거나 한다.

시퀀스 넘버를 함께 받는데 이 값을 알아야 가능. 예전에는 이 시퀀스 넘버가 쉬웠다. 그래서 TCP통신 초기 클라이언트가 서버에게 보내는 ISN(이니셜 시퀀스 넘버)를 국제표준으로 복잡하게 만들자고 정함.

M+F(sip, sport,dip,dport | 여기까지 랜덤성,<some secret> | 여기는 안정성) : M = monotonically increasing clock/counter

F는 해시값을 준다. MD5말고 SHA256같은거.

Injecting false routing information

주입한다 잘못된 라우팅 정보를. 라우터를 찾을때 라우터중 하나가 만일 해커에 의해 점령당한 라우터라고 한다면 잘못된 정보를 쭉뿌린다. 이것을 막기 위해 라우터 주변에서 서로를 인증한다. 해시 맥을 붙여서 라우터들간 인증시킴.

Port scan

공격자가 서버 특정 포트에 패킷 보내봄. Syn/ack 가 오면 포트 존재. rst/ack가 오면 포트가 존재하지 않음. 이것을 이용해 포트가 열린 것을 파악한다. 포트에 따라 제공하는 서비스가 조금씩 다르므로 대략적인 제공 서비스를 파악할 수 있다. 취약한 ip도 찾고 취약한 포트번호도 찾는것이 목적.

fin패킷이나 ack패킷만 보내는 방법도 존재한다. 어쨋든 공격자는 되돌아오는 반응을 보고 정보를 추측해내는 것이다. 때문에 요즘에는 어떠한 정보도 내보내지 않도록 주의하며 보안을 구축한다.

Firewall

일반적으로 방화벽을 라우터와 스위치 사이에 구성. 웹서버와 디비서버가 구성되 있다고 한다면 디비서버에는 웹서버만 접근할 것임. 때문에 방화벽이 외부에서의 접근을 웹서버로의 접근만 허용하도록 하고싶다.

정보보호

Firewall

방화벽이 내부망에 대한 패킷을 검사하고 올바르지 않으면 버림.

0~1023까지는 주로 서버. (최근에는 더 늘어나긴 했음.) 이후 ~65535까지 클라이언트. 방화벽 테이블 볼줄 알아야. 위에서부터 우선순위다. 위에서부터 검열함.

안으로 들어오는 규칙에 대해 하나, 그것에 대한 응답을 위해 하나 더 해줘야 한다. (response때문에 규칙을 또 만들어야 함)

no	S_ip	S_port	D_ip	D_port	Prot.	action
1	*	*	210.123.36.100/32	80	TCP	accept
2	210.123.36.100/32	80	*	*	TCP	accept
3	210.123.37.0/24	1024:65535	*	*	*	accept
4	*	*	210.123.37.0/24	1024:65535	*	accept
5	*	*	*	*	*	deny

• First-rule first-first

–(210.123.37.2, 2000, 164.210.293.17, 443, TCP) à accepted by the 3rd rule

이렇게 표현되고 이것이 몇번 규칙에 적용 되는가? 이런식으로 시험 가능. 풀이 해보자면 1번규칙은 목적지 규칙에 대해 탈락. 2번 규칙은 출발지 주소에 의해 탈락. 3번 규칙은 ip는 만족. 포트도 만족. 3번째 규칙에서 만족하여 방화벽 통과.

문제 stateless

현재 이것은 옛버전. stateless firewall. 안밖 규칙을 따로 적용. 이 방식의 단점은 패킷을 다 따로 처리해야한다는것. 두번째는 공격상의 문제인데, •(His IP, His port, 210.123.37.10, 12345) 로 구성해서 보낸다고 해보자. 이것은 네번째 규칙에 의해 통과된다. 이러한 문제로 인해 statefull firewall. 이문제는 안에서 밖으로, 밖에서 안으로 방향성에 문제에 의해 만들어지는 문제이다.

대책 statefull

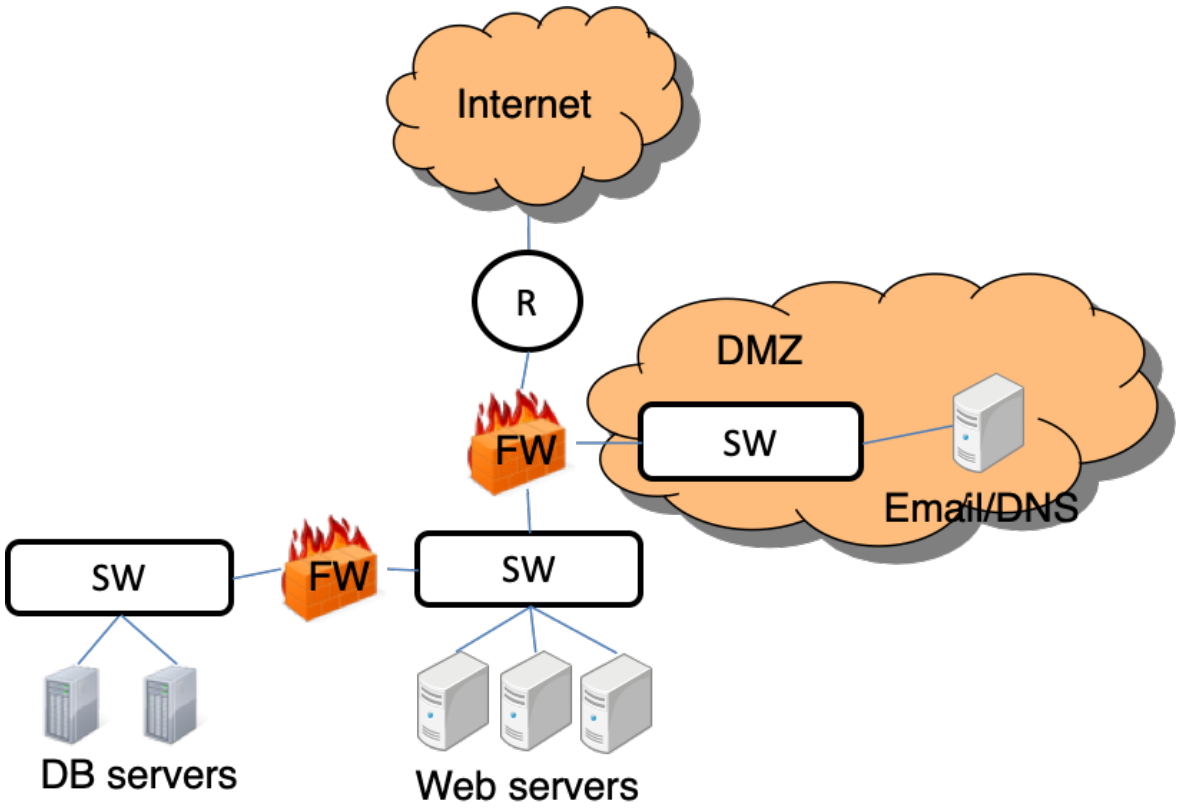
no	S_ip	S_port	D_ip	D_port	Prot.	action
1	*	*	210.123.36.100/32	80	TCP	accept
2	210.123.37.0/24	1024:65535	*	*	*	accept
3	*	*	*	*	*	deny

방화벽이 패킷을 메모리에 기억하고 있다. 규칙에 의해 기억된 패킷을 기억하고 이것에 대한 응답을 내보낼때 기억한 메모리을 참조해 내보낸다.

DMZ(demilitarized zone)

방화벽에 랜카드를 꽂아서 방화벽으로 흘러들어오는 정보를 dmz로 보내 독립성을 추구한다. 이렇게 하면 밖에서 dmz로 연결되는 규칙, dmz에서 내부 db로 접근하는 규칙.

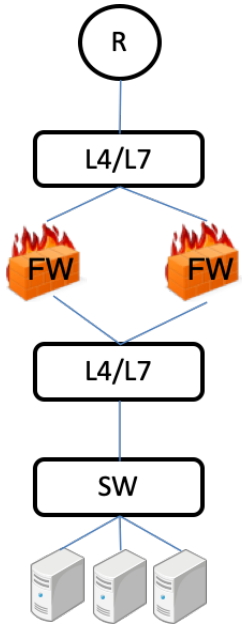
내부 스위치를 통해 접근이 아닌 방화벽을 통해서 접근하게 되므로 강력하다. (방화벽 규칙에 의해서만 접근 가능하므로.)



이런식으로 구현하는 곳이 많음.

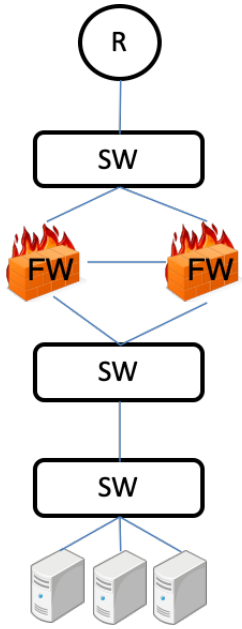
방화벽 고장 문제

방화벽을 설치했더니 방화벽이 고장났을때 전체가 먹통되는 문제가 발생한다. 때문에 방화벽을 하나 더 두어서 가용성을 높혔다. 그런데 이때 문제가 발생한다. 방화벽이 두개이기 때문에 statefull을 위해 메모리에 패킷을 기억해놓은 것이 만일 다른쪽으로 갈 경우 메모리에 없기 때문에 거부한다. 이때문에 방화벽 윗,아랫단에 L4/L7 을 단다.



패킷이 들어오면 »H(s_ip XOR d_ip) mod (# of firewall) 돌린다. 이렇게 돌리면 0,1 둘중에 하나가 나온다. 이렇게 구성해서 0이면 왼쪽, 1이면 오른쪽 이렇게 보낸다. XOR연산은 s_ip,d_ip값이 바뀌어도 값이 동일하므로 항상 같은 방화벽으로 들어갈 수 있게 됨.

근데 이게 돈이 많이 든다 그래서 방화벽끼리 랜으로 연결하고 방화벽이 가진 정보를 동기화하도록 한다.

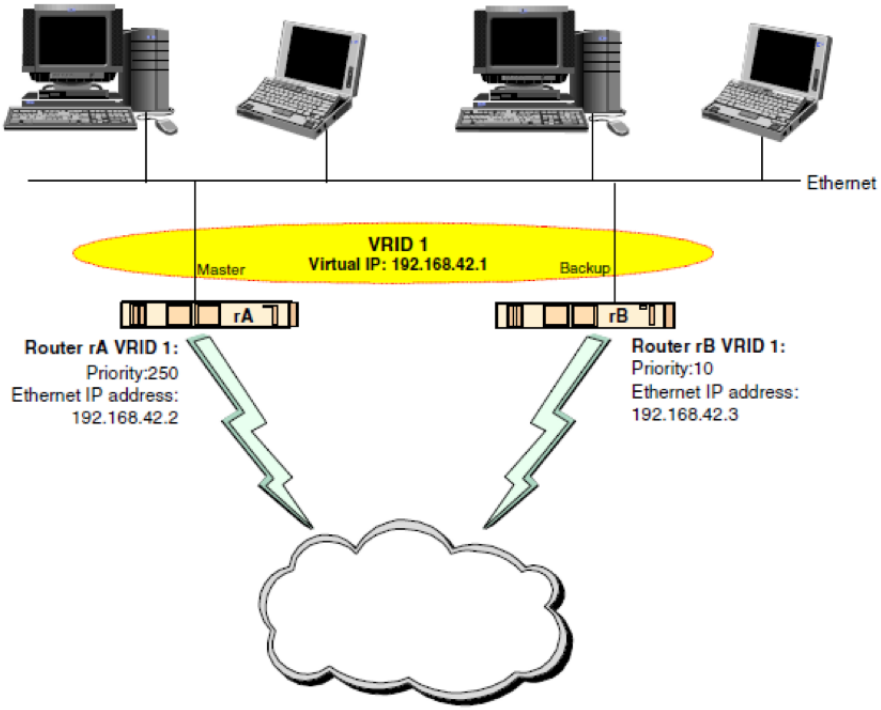


방화벽이 죽었는지 판별하는 것은 스위치가 지속해서 ping요청을 해당 경로로 보낸다. 때문에 장애에 대해 대처가 가능하다.

고 가용성

고 가용성 문제는 때문에 방화벽 뿐만 아니라 라우터에도 적용 된다.

VRRP를 잘 이해할것.



두 박스가 라우터. 두 대의 라우터는 각각 자신의 고유한 피지컬 ip를 가진다. 그리고 가상 ip 를 가지게 한다. 노트북들은 이 가상의 ip로 패킷을 토스. 만일 왼쪽이 주 라우터라면 가상 ip 주소를 왼쪽이 추가로 가지도록 한다. 그래서 일반적 통신을 왼쪽에게 시킨다. 그러면 오른쪽은 계속 듣고있는다. A가 나 살아있다는 브로드캐스트를 보내는데 이것을 잘 확인. 하다가 A의 브로드캐스트가 안오면 가상 ip를 자신이 가져옴. 이렇게 하면 서버들은 아이피를 바꾸지 않고도 계속 통신 할 수 있음.

마스터와 슬레이브. 마스터가 처리하는 애고 슬레이브는 평소에 놀고있음. 마스터가 안되면 슬레이브가 처리. 해커가 자신이 마스터 행세를 하면 안되기 때문에 이런 통신은

1.No authentication

2.Simple clear-text passwords

3.Strong authentication (using IP authentication with MD5 HMAC)

등 인증해서 처리.

근데 이게 하나를 놀게 하기가 너무 아까움. 그러면 가상 ip를 하나 더 두게 한다. 그래서 처리 하다가 하나 죽으면 그 죽은 아이피도 가져오게 함.

정보보호

가용률

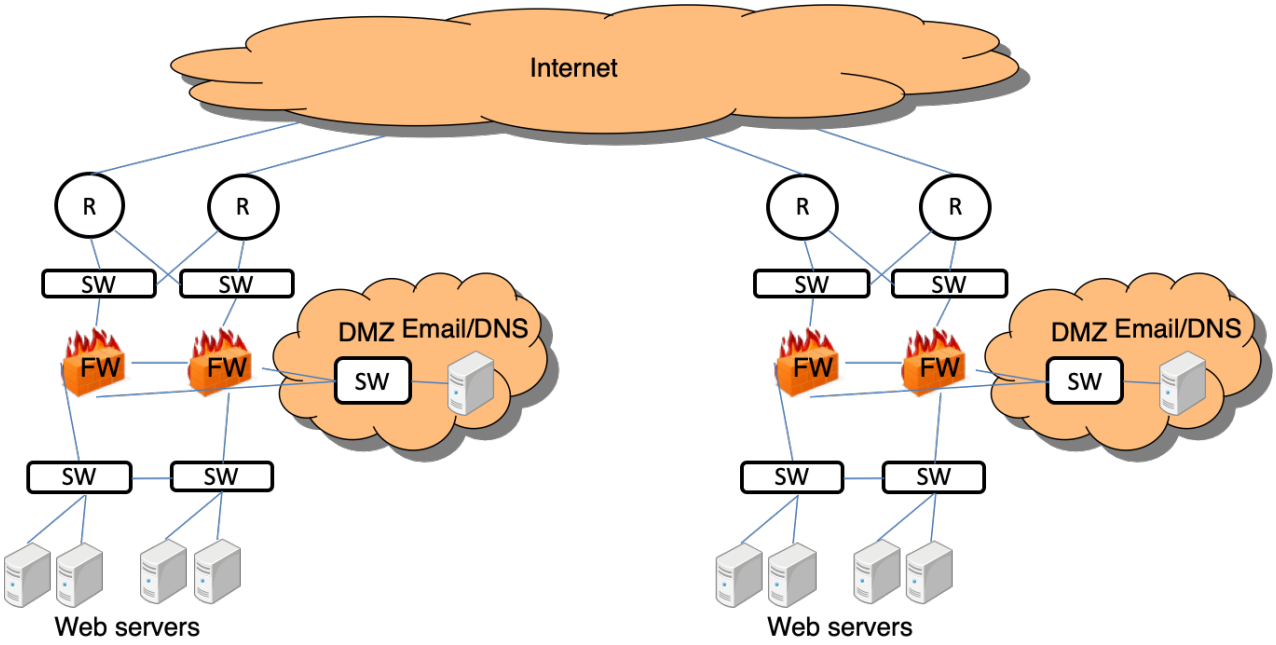
서버가 2대면 동작할 확률

> 1 - (1-p)(1-p) -> 1-두대 동시에 고장날 확률.

p = 0.9999 의 의미는 1년에 53분 고장나있는 것이다. P2 = 1-(1-p)^2

Single point of failure : 한 점을 끊었을때 시스템 장애가 생김.

Disaster Recovery(DR) Center



Global Server Load Balancing(GSLB)

정보보호

VRIP, firewall 등 더 볼 것

DDoS

최근엔 랜섬디도스 공격이 이슈임.

> Bontnet, bot master, bots, C&C(command & control) server 용어 알아놓을 것.

최근 디도스는 많은 분량이 아니고 일반적인 분량의 트래픽을 많은 대수로 보내게 되어 어렵게 한다. **Flash crowd**

Intrusion Prevention Sysyem

디도스는 완벽하게 없앨 수 있는 것이 아님. 완화시키는 기술이 이 IPS. IDS의 한계를 극복한 것. IDS는 탐지는 하는데 방어를 하지는 않는다. IPS는 이런 단점을 극복. IDS + Firewall = IPS. IDS가 탐지한 것에 대해 차단.

이것만 가지고는 기술이 약해 DDoS mitigation기능을 추가함.

> IP-spoofing test, Traffic management 기능을 수행할 수 있도록 함. Traffic management는 과도하게 트래픽을 사용하는 것을 막도록 함.
>
> 출발지 IP가 위조했는지 아닌지를 감별해낸다.

IP-spoofing test : Using SYN-cookie

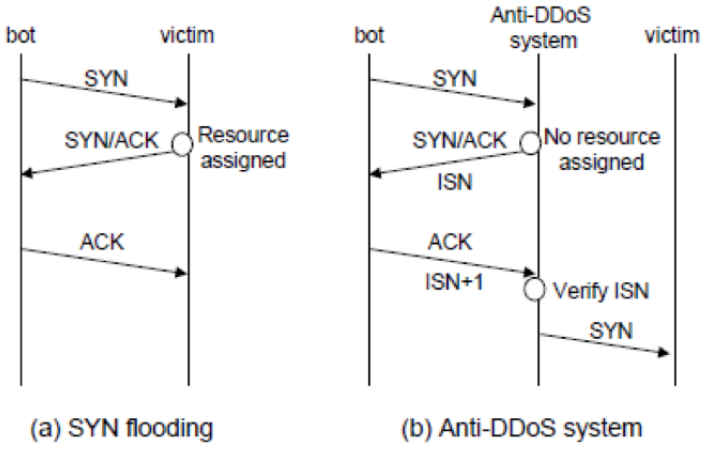


Fig. 2. SYN flooding v.s. Anti-DDoS system's SYN cookies. ISN is computed from $h(s\_IP, s\_port, d\_IP, d\_port, key, time)$, where h is a hash function.

안티 디도스 시스템을 서버 앞에 설치. 본래는 SYN을 받으면 그 정보를 메모리에 올려놓는데, 안티 디도스는 기록하지 않음. 대신 SYN/ACK 보낼때 ISN을 같이 보낸다. TCP 특성상 ISN을 받으면 +1을 해서 보내므로 받아와서 -1 하면 본래 ISN값. 받은 패킷을 분석해 다시 ISN을 만들어서 되돌아온 ISN값을 비교해 같으면 서버에게 전달한다.

메모리를 할당하지 않기 때문에 리소스를 매우 적게 사용한다. 이 기능이 매우 좋아서 디도스 대응으로 많이 이용되고 있다.

웹 보안

> WAF : Web Application Firewall, 웹 보안을 잘 할 수있는 기능을 모아놓은 것. 80번 포트에서 발생할 수 있는 해킹을 막는 기술.
>
> CSRF, SQL Injection, OWASP, Vulnerability

Web shell

MITRE Common Weakness Enumeration - CWE-434 : 취약점들의 이름을 통일해서 부르자. MITRE 에서 공통의 이름을 부여하자고 해서 CWE(CVE)를 만듬.

카페에서 이미지를 업로드 하는 것과 같이 서버가 업로드 가능한 상황. 어디에 업로드 되는지 공격자가 알아야 한다. 그리고 이 위치에서 실행 가능한 권한이 존재해야 한다.

막는것은 쉽다. 공격자가 파일을 아무거나 올릴 수 없도록 하자. 파일의 확장자를 검열해 올리지 못하도록, 또는 파일의 크기도 제한을 시키면 된다. 불필요한 디렉토리에서는 권한을 제한한다. 웹 어플리케이션은 루트 권한으로 돌리지 않도록 한다.

Cookie

유저에 웹 브라우저에 저장되는 작은 텍스트. 이 기록은 서버가 남기는 것. 마지막에 접속했던 시간이나 페이지 등을 남긴다.

쿠키값은 사용자가 변경 가능하기에 서버는 이 값을 맹신하면 안된다.

정보보호

웹보안

Cookie

프라이버시를 해칠 수 있다. 간단한 사용자를 식별 할 수 있는 태그 등을 남긴다. SSL을 쓰는 것이 좋다. 보안상.

클라이언트에 저장되는 요소는 SSL로 보안 할 수 없다. (쿠키 등)

쿠키는 사용자가 변경 할 수 있다. 옛날에 결제 정보를 쿠키에 두도록 했다가 문제가 발생한 예시가 존재함.

Countermeasure

쿠키는 서버가 내려줌. 클라이언트가 쿠키를 바꾸지 못하도록 바꾸고 싶음. 서버가 키와 값을 해쉬 돌려서 태그로 붙인다.

Session management

서버쪽에서 클라이언트의 상태를 확인하기 위한 기술.

- 세션토큰을 이용하는 방법이 세가지.

  브라우저 쿠키를 이용하는 방법, URL링크에 담는 방법. Hidden form을 이용하는 방법.

Session hijacking

값을 유출하거나 뺏어올 수 있을 때. 예측 가능한 id나 값을 설정하지 말것.

> cross site criptiong(XSS)

Strong session token

> SID = [userID, exp. time,data]
>
> 위 값만 가지면 위조 가능하기에 여기다가 HMAC(k, SID) = Session Token
>
> 추가로 ip정보를 넣어 더 강력하게 할 수도 있고 로그아웃에 대한 스테이터스도 두어서 세션 하이재킹을 막는다.
>
> 클라이언트 ip주소를 넣으면 왜 좋으냐 하면 클라이언트가 바뀌었나 안바뀌었나 파악할 수 있기 때문임. 하지만 이것에 의존하면 안된다.

XSS(Cross-site scripting)

CVE(CWE) Improper Neutralization of Input During Web Page Generation

적절하지 못한 중성화(위험한 태그를 없애는 것을 의미), 웹페이지가 생성되는 동안. 인풋값을 적절하게 뉴트럴라이제이션 하지 못한것을 의미.

보는 글에 자바 스크립트 등으로 실행 가능 하도록 만듦. 보는 사람의 입장에서 코드가 동작 하도록 함.

- Persistent XSS

  공격자가 공격 코드를 포함해 글을 올림. 클라이언트가 읽음. 읽는 과정에서 공격 코드를 읽고 구동시킴. 공격자에게 세션 토큰이나 쿠키 값 등을 전송하도록 함.

  요즘 유행하는 것을 여기에 비트코인 채굴 코드를 넣어서 채굴하도록 함. 채굴에 성공하면 공격자에게 보냄.

  persistent라고 붙인 이유는 계속 남아있기 때문.

- Non Persistent XSS (or reflected XSS)

  검색엔진에서 검색어를 입력하게 되면 검색 결과 페이지 맨위에 결과로써 올라왔다.

중요한 것은 공격에 해당하는 스크립트를 만들어놓고 공격 대상자가 이 스크립트를 실행되도록 만듬.

> Sanitization - 방역

바람직하지 못한 캐릭터(스크립트성 언어의 캐릭터)를 제거하도록 함. 예를들어 JSP에서 <를 &lt 등으로 변경하면 동작하지 않음.

클라리언트가 서버를 너무 믿어서 당함.

Cross-Site Request Forgery(CSRF)

거짓된 요청을 해서 공격. Aka one-click attack or session riding

성공하면 서버를 공격자가 가져갈 수 있음. (성공만 하면 대박) 서버가 클라이언트를 너무 믿어서 당함.

- 패스워드를 바꿀 때 예전 패스워드를 한번 더 물어보는 이유.

  먼저 XSS로 공격. 이메일 등을 통해서 전송하면 클라이언트가 서버에 로그인한 상태에서 공격 XSS를 누르면 변경 페이지로 자동으로 가서 변경되게끔 시도하도록 함.

  현재 비밀번호를 한번 더 입력하라고 한다면 이 시도를 막을 수있다.

SQL Injection

데이터베이스를 공격하는 기술. CWE-89. Failure to Sanitize Data into SQL Queries.

id나 패스워드 등 입력창에 sql구문을 집어넣는 것.

때문에 '"\/;와 같은 특수문자를 사용하지 못하도록 해야함. 데이터베이스의 계정 권한을 낮춤.

Insecure Direct Object Reference

직접 URL을 참조하여 들어가는 것. URL에 내용 일부를 살짝 변경하여 다른 정보에 접근 하는 것.

Access Control for Administration

관리자용 url이 따로 존재할 수 있는데 공격자들이 이것을 추측해서 시도해 볼 수 있으므로 주의해야 한다. 웹서버 자체에서 접근 제어를 제한할 수 있다.

2factor authentication

두가지 요소로 인증할 것. 패스워드 OTP 전화인증 등...

보안 도구들 : Vulnerability Scanner

nmap, hping3, Nessus, Paros ...

black box, white box(소스코드가 공개된 종류들)