Network Security 개념 및 중요성

-데이터 통신 경로상에서 정보 보안을 유지하기 위한 기술(아키텍쳐/솔루션)

-통신 가능한 경로를 구성 할 것인지? 경로를 만든다면 어떻게 보안 수준을 유지할 것인지?

-인증(Authentification), 무결성(integrity), 암호화(confidentiality)

Internet/intranet/DMZ 아키텍쳐, VPN (Virtual Private Network) 기술, 네트워크 보안 솔루션

데이터 보안 유지한다라 할 때 돈이 필요하다. 악의적인 목적이 있는 사람에게서 데이터 보호하기 위해. 전용망 private network , 거리나 Band /width에 따라 돈이 나감

전세계 모든 사용자들이 인터넷을 share하고있다. 아무것도 보장이 되지 않는다.

인증은 상호간 통신하는데 누구랑 통신하는지 명확히 알고있느냐, 전용선은 내가 누구랑 연결하는지 정해져있지만 인터넷은 중간에 어떤 놈이 내 데이터를 다 쳐다볼 수 있다.

무결성: A로 데이터를 보냈는데 B로 바뀐게아니냐, 암호화, 별도의 장비를 쓴다 IPSec 프로토콜?

지리, B/W 비용. MPLS VPM? 테크닉 사업자들이 망을 미리 구성해놓고. CPE, VPN , MPLS VPM

2.Network Security Architecture

Data를 생성/처리하는 Device, Network 상에서 전달되는 Data의 보안 레벨이 다양함.

(기업내 임직원들만 이용해야만 하는 Data, 파트너사 까지 이용 되어야하는지? 외부에도 이용 되어야 하는지) Internet, Intranet, Extranet. 예전에는 데이터센터가 별도의 공간을 가지고 인프라 부터 올려서 서버 설치하고 이 데이터들을 잘 저장한다거나, 활용 할 수 있게하였다.

임직원들이 존재하는 사업장내에 디바이스들, 응용 서버나 스토리지들이 위치하고있는 이 경로를 전용으로 구축. Extranet이란? 관문존? 보안관문 설치해서 통신이 원하는 애들 하고만 한다

Firewall, IPS, 외부로 유출되는지 DLP, 메일보안 이런 것들. 웹키퍼. 다음 강의에는 보안솔루션들,

VPN들, 기술들 IPSEC, MPLS망 등 학습

3. 방화벽 개념 및 동작원리

Firewall. 서로 다른 보안레벨의 네트웍의 경로 사이에 위치하며, 지나 다니는 트래픽을 보고 통과시킬지 말지 결정. Proxy방식, Packet Filtering, Stateful inspection – UTM, NGFW

보안 기능 외 네트웍 장비로서의 기능과 관리(모니터링 )기능. 이중화 방식, 세션 테이블 관리(TCP UDP, Timeout), 로깅/관제. L3 = IP, L4 = PORT. 포트 정보를 보고 해당 패킷 보낼지 말지 결정한다.

초창기에는 TRAFFIC이 별로없을때는 그냥 모든 데이터를 Telnet, FTP 정도 HTTP 등 Proxy 방식으로 중간 방화벽이 게이트웨이 역할. 네트웤 ACL 네트워크 장비에서 엑세스 컨트롤도 한다.

A와 B가 가는데 양방향으로 응답 돌아오는 세션(IP,포트정보)를 적용해서 또 열어놔야된다.

Statful Inspection 이란? 지나가는 상태 정보를 보는데. 방화벽의 성능 뿐만이 아니라 Rule관리의 편의성, 등을 위해. 세션 테이블을 가지게 되는데 방화벽 정책은 일단 다막고 받아들일거만 받아들임 TCP 시스템이 보안 레벨이 높은곳에서 낮은곳으로는 올 퍼밋, 보안 레벨이 낮은곳에서는 ALL deny 부분 퍼밋. 물론 요즘은 아웃바운드도 다 컨트롤하는 곳도 있다.

Log를 남겨야 되기에 rule 넣고 그 위에

생활코딩 HTTPS와 SSL 인증서.

HTTP란 HyperText Transfer Protocol의 약자이다. Html 통신규약 의미한다. S는 secure이고.

HTTP는 암호화 되지 않는 방법으로 데이터를 전송하기에 서버와 클라이언트가 주고받는 메시지메 감청하는 것이 매우 쉽다. 예를들어 로그인을 위해서 서버로 비밀번호를 전송하거나, 또는 기밀문서를 열람하는 과정에서 악의적인 감청이나 데이터의 변조등이 일어날 수 있다는 것이다.

이를 보안한 것이 HTTPS이다. 사람들이 아이디나 비밀번호를 비슷하게 만드는데 로그인 페이지가 HTTPS가 아닌걸로 이루어져서 해커들이 해킹에 성공하게되면 그 사이트하나만 해킹당하는게아니라 다른 사이트들의 정보들도 전부다 해킹당하게 되는 것.

그렇기에 HTTPS로 프로토콜 변경을한다. SSL,TLS 프로토콜 레이어. HandShake, Alert 등.

SSL의 동작 방식을 알게된다. SSL 디지털 인증서. 암호화 KEY, 복호화 도 할 수 있다.

SSL 암호화의 종류?

1. 대칭키. 암호화와 복호화 양쪽다 동일한 방법으로 할 수 있는 암호화 기법이다.

OPENSSL이라는 것을 이용해서 어떤 텍스트를 암호화해보자.

openssl enc -e -des3 -salt -in plaintext.txt -out ciphertext.bin;

openssl 이용한다 enc -e -des3는 des3방법으로 암호화한다.

-in plaintext가져오고 -out ciphertext.bin으로 암호화한다.

이상한 정보가 나온다 cat ciphertext.bin 해보니까. 암호를 멀리있는사람이 복호화하려면 키값을 알아야됨. 멀리있는 사람에게 키값을 전송해야됨. 그 키값은 암호화가 되어있지않음. 의미가없다?..

치명적인 단점.

1. 공개키 방식

공개키 방식은 두개의 키를 갖게되는데 A키로 암호화를 하면 b키로 복호화 할수 있고, B키로 암호화하면 A키로 복호화할 수 있는 방식이다. 이 방식을 착안해서 두개의 키중 하나의 키를 비공개키(PRIVATE KEY)로하고, 나머지 공개키(public key)로 지정한다. 비공개키는 자신만이 가지고 있고, 공개키를 타인에게 제공한다. 공개키를 제공 받은 타인은 공개키를 이용해서 정보를 암호화한다. 암호화한 정보를 비공개키를 가지고있는 사람에게 전송, 비공개 키의 소유자는 이 키를 이용해서 암호화된 정보를 복호화한다. 이 과정에서 공개키가 유출된다고 해도, 비공개키를 모르면

정보를 복호화 할 수 없기 때문에 안전하다. 공개키로는 암호화는 할 수 있지만 복호화는 할 수 없기 때문이다. 인증서의 원리.

공개키가 데이터를 제공한 사람의 신원을 보장해주게 되는것읻아.

openssl genrsa -out private.pem 1024;

rsa 키 생성.

Cat private.pem

openssl rsa -in private.pem -out public.pem -outform PEM -pubout;

pem 키 가져와서 public.pem 키 만든다

openssl rsautl -encrypt -inkey public.pem -pubin -in file.txt -out file.s니;

이건 encrypt 암호화하고 public.pem이라는 키를사용한다.

File.txt를 암호화한다. 암호화된파일은 file.ssl이라고 export한다.

암호화되어있다. 비공개키를 가지고있는사람은 자신에게 도착한 정보 복호화한닥

비공개키를쓰고 자신외에는 그 내용을 복호화할수없다.

이번엔

openssl rsautl -decrypt -inkey private.pem -in file.ssl -out decrypted.txt;

복호화 코드

공개키 방식으로 암호화 및 복호화했다. 매우 중요함.

인증서용어.

1. 클라이언트가 접속한 서버가 신뢰할 수 있는 서버임을 보장한다.
2. SSL 통신에 사용할 공개 키를 클라이언트에게 제공한다.

CA? : 인증서의 역할은 클라이언트가 접속한 서버가 클라이언트가 의도한 서버가 맞는지 보장한다. 이 역할하는 민간기업을 CA(Certificate authority) 혹은 Root certificate라고 부른다.

Comodo나 godaddy, globalsign, Symantec이라고 한다. 공인된 인증서가 아니면 구글크롬이나

이런데서 보증되는 사이트가 아니라는 표시가 나타나게된다.

ssl인증서의 내용. 1. 서비스의 정보, 2. 서버측의 공개키(공개키의 내용, 암호화방법)

인증서안에 여러정보들이 포함되어있다.

1. SSL 통신과정.

Handshake => 전송 => 세션 종료

실제 데이터: 대칭키 대칭키의 키: 공개키.

악수가 제일 중요하다. 사람과 사람이 소통할때를 생각해보자. 우선 인사를 한다.

인사를 통해서 상대의 기분과 상황을 상호탐색을 하는 것이다. 이 과정이 잘되야 소통이 원할해진다. 클라이언트와 서버 사이도 마찬가지다. 실제 데이터를 주고 받기전에 클라이언트와 서버는 일종의 인사인 Handshake를 한다. 이 과정을 통해서 서로 상대방이 존재하는지, 또 상대방과 데이터를 주고 받기 위해서는 어떤 방법을 사용해야하는 지를 파악한다.

SSL 방식을 이용해서 통신을 하는 브라우저와 서버 역시 핸드쉐이크를하는데. 이때 SSL 인증서를 주고받는 다.

1. 클라이언트가 서버에 접속한다. 이 단계를 CLIENT HELLO 라고한다. 이 단계에서 주고받는 정보는. 아래와같다

클라이언트 측에서 생성한 랜덤 데이터, 클라이언트가 지원하는 암호화방식들

클라이언트와 서버가 지원하는 암호화 방식이 서로 다를수 있기 때문에 상호간에 어떤 암호화 방식을 사용할 것인지에 대한 협상을 해야한다

. 이 협상을 위해서 클라이언트 측에서는 자신이 사용할 수 있는 암호화 방식을 전송한다

세션 아이디: 이미 SSL 핸드쉐이킹을 했다면 비용과 시간을 절약하기 위해서 기존의 세션을 재활용하게 되는데 이 때 사용할 연결에 대한 식별자를 서버 측으로 전송한다.

1. 서버는 CLIENT HELLO에 대한 응답으로 SERVER HELLO를 하게된다.

서버 측에서 생성한 랜덤 데이터: 아래 3번 과정 참조.

서버가 선택한 클라이언트의 암호화 방식: 클라이언트가 전달한 암호화 방식 중에서 서버쪾에서도 사용할 수 있는 암호화 방식을 선택해서 클라이언트로 전달한다. 이로써 암호화 방식에 대한 협상이 종료되고 서버와 클라이언트는 이 암호화 방식을 이용해서 정보를 교환하게 된다.

1. 클라이언트는 서버의 인증서가 CA에 의해서 발급된 것인지를 확인하기 위해서

클라이언트에 내장된 CA리스트를 확인한다. CA리스트에 인증서가 없다면 사용자에게 경고 메시지를 출력한다.

클라이언트는 상기 1번을 통해서 받은 서버의 랜덤데이터와 클라이언트가 생성한 랜덤데이터를 조합해서 pre master secret라는 키를 생성한다. 이 퀴는 뒤에서 살펴볼 세션단계에서 데이터를 주고받을 때 암호화하기 위해 사용될 것이다. 이때 사용할 암호화 기법은 대칭 키 이기 떄문에

Pre master secret 값은 제 3자에게 절대로 노출되어서는 안된다.

그럼 문제는 이 pre master secret 값을 어떻게 서버에게 전달할 것인가 이다.

이 때 사용하는 방법이 바로 공개키 방식이다. 서버의 공개키로 pre master secret 값을 ㅏ암호화해서 서버로 전송하면 서버는 자신의 비공개키로 안전하게 복호화 할 수 있다.

그럼 서버의 공개키는 어떻게 구할 수 있을까? 서버로부터 받은 인증서안에 들있다. 이 ㅅㅂ

서버의 공개키를 이용해서 pre master secret 값을 암호화 한 후에 서버로 전송하면 안

안전하게 전송할 수 있다.

CA를 통해서 인증서 구입 및 서버에 세팅하는 방법.

인증서가 은근 비싸네.

서브도메인 몇 개지운하냐, 비용이들지않지만 Cacert는

다만 브라우저가 이 인증서를 신뢰할수 있는 인증서가 아니라고 함.

서버에 쓰기에는 좀 무리가 있다.

Starcom꺼 StarSSL Free 좋다. StarSSL은 없고 Certbot써야된다.

웹 서버 세팅. APACHE기준.

Starcom.client.certificate.p12

Sub.class1.server.ca.pem

Readme.txt 파일등이있고

Sudo apt-get install apache2

아파치의 s니 모듈활성화한다

Sudo a2enmod ssl

Sudo service apache2 restart

재시작

SSL 인증서 관련된 파일을 위치시킬 디렉토리 만든다.

Sudo mkdir /etc/apache2/s니

인증서 파일들을 이동시킨다.

Sudo mv ~/s니/ .

Sudo !!

보안을 위해서 디렉토리와 파일의 권한을 조정한다

Sudo chown -R root :root /etc/apache2/s니;

파일의 권한을 600 ( 소유자만 읽기, 쓰기 허용)

Sudo chmod 600 /etc/apache2/s니/\*.\*

디렉토리의 권한을 700 (소유자만 읽기, 쓰기, 실행 허용)

Sudo chmod 700 /etc/apache2/s니 ;

Virtualhost를 설정한다. 하나의 웹서버에서 여러 개의 서비스를 도메인별로 운영할수 있도록 돕는다. 편집기는 nano를 사용한다. Sudo nano /etc/apache2/sites-available/default-ssl

8.파일의 내용에서 지시자의 값을 아래와 같이 변경한다.

SSLCertificateFile /etc/apache2/ssl/ssl.crt

SSLCetificateKeyFile /etc/ssl/private/ssl-cert-snakeoil.key

9.버추얼 호스트 default-ssl

서티피케이트 체이닝 ? 여러 인증기관을 hierarchy하게

SSLcertificatechainfile

Sslcacertificatefile도 같이 바꿔준다

Sudo a2ensite default-ssl

버추얼 호스트 default-ssl을 활성화된 서비스로 등록한다.

Apache를 재시작한다.

Sudo service apache2 restart