2019 wheel seminar

Security

lulu

보안의 3요소(C I A)

Confidentialiy
기밀성

인가된 사람만이 그 내용을 알아야 함

Integrity
무결성

비 인가된 방식으로 내용이 변경/파괴/훼손되면 안 됨

Availability
가용성

인가된 사용자는 요구 시 내용을 열람 가능

보안 공격

Passive attack

시스템에 영향 없이 정보를 취득하는 공격

Alice가 Bob에게 보낸 메일을
Trudy가 훔쳐본다

Active attack

해당 시스템을 바꾸면서 정보를 취득하는 공격

Alice가 Bob에게 보낸 메일을
Trudy가 빼돌려 내용을 고쳐 Bob에게 다시 보낸다

철저한 Prevent와 공격 발생 후 신속한 Recover 모두 중요!

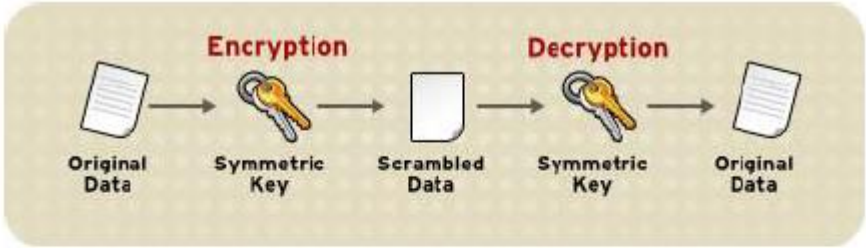
목차

- 암호화 방식
  - 실습
- SSL
  - 실습
- 보안 공격/방어
  - 실습

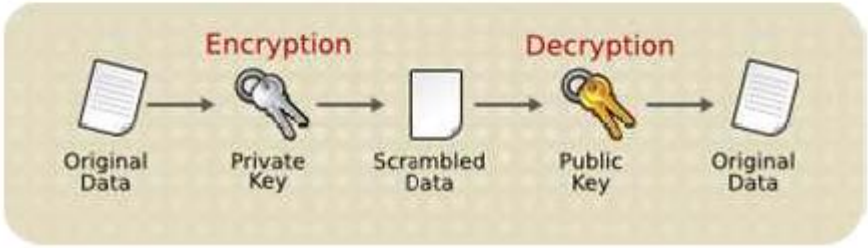
암호화 방식

암호화(Encryption)

- 대칭 키 암호
  - 암호화 키 == 복호화 키

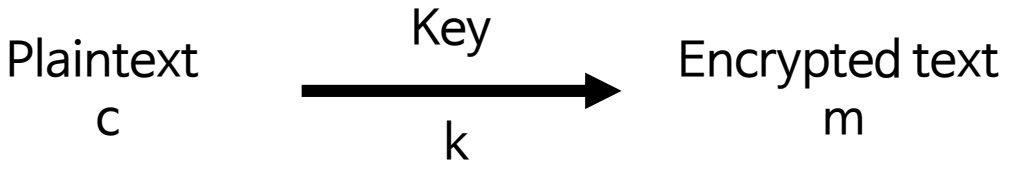


- 공개 키 암호(=비대칭 키)
  - 암호화 키 != 복호화 키



- (암호화는 아니지만) Hash

대칭키 암호



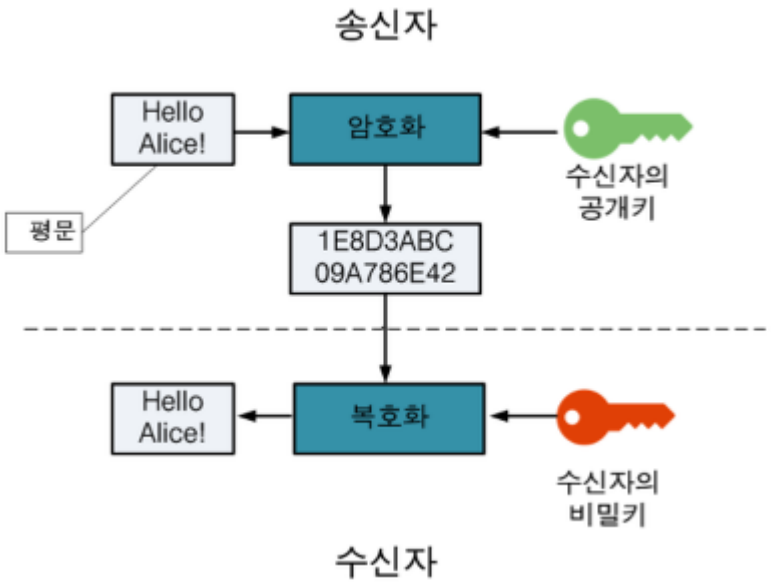
- 암호화하는 단위에 따라!
  - 스트림 암호(Stream Cipher)
    - c=m^G(k) ; G=pseudorandom generator
    - G(k)값으로 k를 얻어낼 수 없어야 함(One Time Pw가 활용되기도 함)
    - RC4, A5/1, A5/2
  - 블록 암호(Block Cipher)
    - Block 단위의 암호화
    - c=E(m,k), m=D(c,k)
    - https://youtu.be/NRWAySME3Co 참고
    - DES, 3DES, AES

공개 키 암호

- 특정한 종류의 정보(private key)없이는 풀기 어려운 문제(public key)

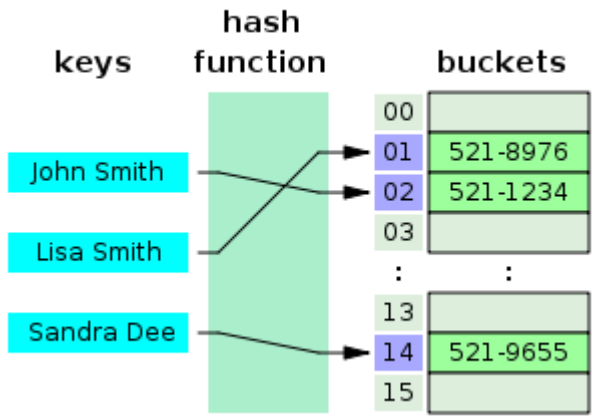
- 오늘날 널리 이용되는 RSA 암호(Rivest Shamir Adleman)
  - NP 문제인 소인수분해 문제를 이용한다

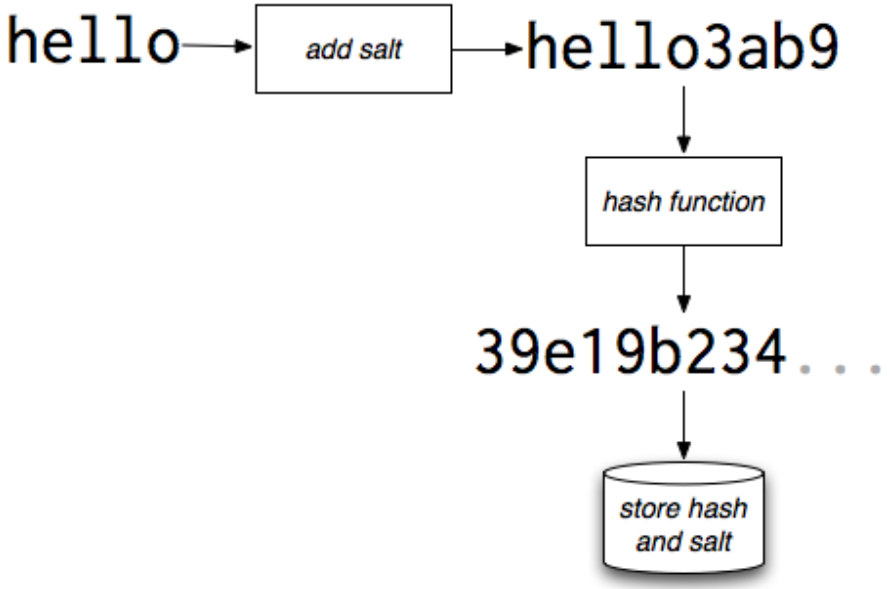
- 두 종류로 나뉨
  - 공개 키 암호
    - 공개키로 암호화, 비밀키로 복호화
    - 비밀키를 갖고있는 사용자만 해독 가능
  - 공개 키 서명
    - 비밀키로 암호화, 공개키로 복호화
    - 누구나 해독 가능하나, 비밀키를 갖는 사용자만 암호화 가능



해시(Hash)

- 데이터를 특정길이의 다른 데이터로 전환하는 것
  - 자료구조 Hash table으로 생각하면 됨
  - 다시 복호화할 수 없다는 점에서 암호화는 아님
  - 비밀번호 보관, 데이터 위/변조 검증, 데이터 식별자
  - MD5, SHA1
- Salt





암호화 실습

openssl로 암호화 해보기_대칭키방식

```
wheelseminar@tong:~/lulu/security_test$ echo 'test sentence' > plaintext.txt
wheelseminar@tong:~/lulu/security_test$ openssl enc -e -des3 -salt -in plaintext.txt -out ciphertext.txt
enter des-ede3-cbc encryption password:
Verifying - enter des-ede3-cbc encryption password:
wheelseminar@tong:~/lulu/security_test$ openssl enc -d -des3 -in ciphertext.txt -out plaintext2.txt
enter des-ede3-cbc decryption password:
```

openssl enc -e - des3 - salt - in [입력파일] - out [출력파일] : des3방식으로 암호화하기
openssl enc - d - des3 - in [입력파일] - out [출력파일] : des3방식으로 복호화하기

openssl로 암호화 해보기_공개키방식

```
wheelseminar@tong:~/lulu/security_test$ echo 'test sentence' > plaintext.txt
wheelseminar@tong:~/lulu/security_test$ openssl genrsa -out private.pem 1024
Generating RSA private key, 1024 bit long modulus
.+++++
.....................+++++
e is 65537 (0x010001)
wheelseminar@tong:~/lulu/security_test$ openssl rsa -in private.pem -out public.pem -outform PEM -pubout
writing RSA key
```

```
wheelseminar@tong:~/lulu/security_test$ openssl rsautl -encrypt -inkey public.pem -pubin -in plaintext.txt -out ciphertext.txt
wheelseminar@tong:~/lulu/security_test$ openssl rsautl -decrypt -inkey private.pem -in ciphertext.txt -out plaintext.txt
```

openssl genrsa -out private.pem 1024 : private key 생성
openssl rsa -in private.pem -out public.pem -outform PEM -pubout : private key에 맞는 public key 생성

openssl rsautl -encrypt -inkey public.pem -pubin -in [입력파일] -out [출력파일] : public key, RSA 방식으로 암호화
openssl rsautl -decrypt -inkey private.pem -in [입력파일] -out [출력파일] : private key, RSA 방식으로 복호화

SSL

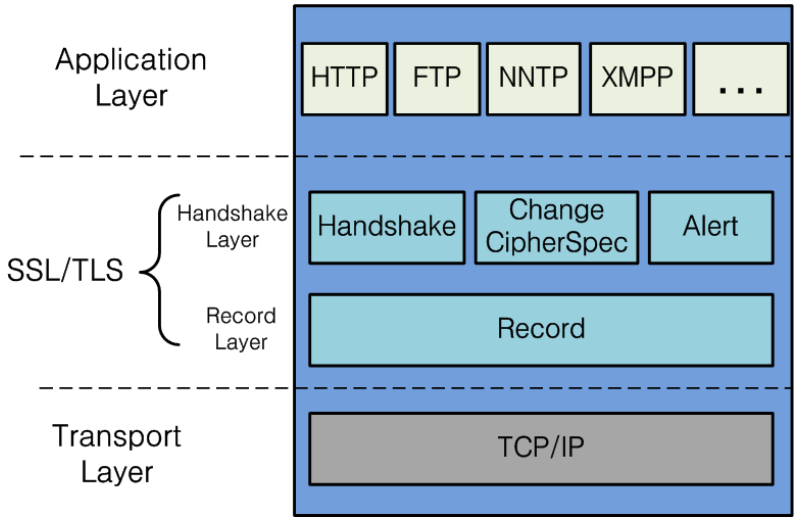
HTTPS(HTTP over SSL)

- 보안이 강화된 HTTP(443번 포트)

용어 정리

- SSL(Secure Socket Layer protocol)
  - HTTPS는 SSL 프로토콜 위에서 돌아가는 프로토콜

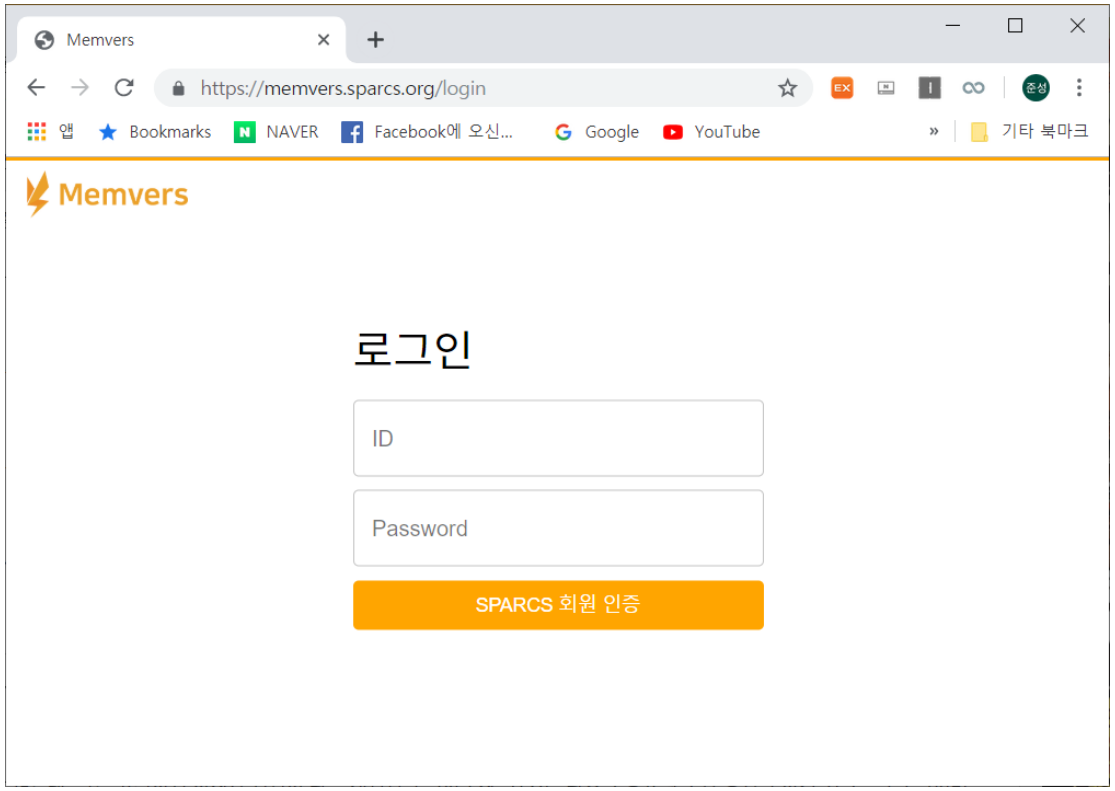
- TLS(Transport Layer Security protocol)
  - SSL과 같다고 보면 됨
  - 표준화 기구 IEFT에게 관리되면서 변경된 이름
  - TLS 1.0 == SSL 3.0



SSL을 만든 Netscape

https://www.ssllabs.com/ssltest/ SSL test 사이트

SSL의 역할



- 통신내용의 감청/변조를 막아준다
- Client가 접속하려는 Server가 신뢰할 수 있는지 확인한다

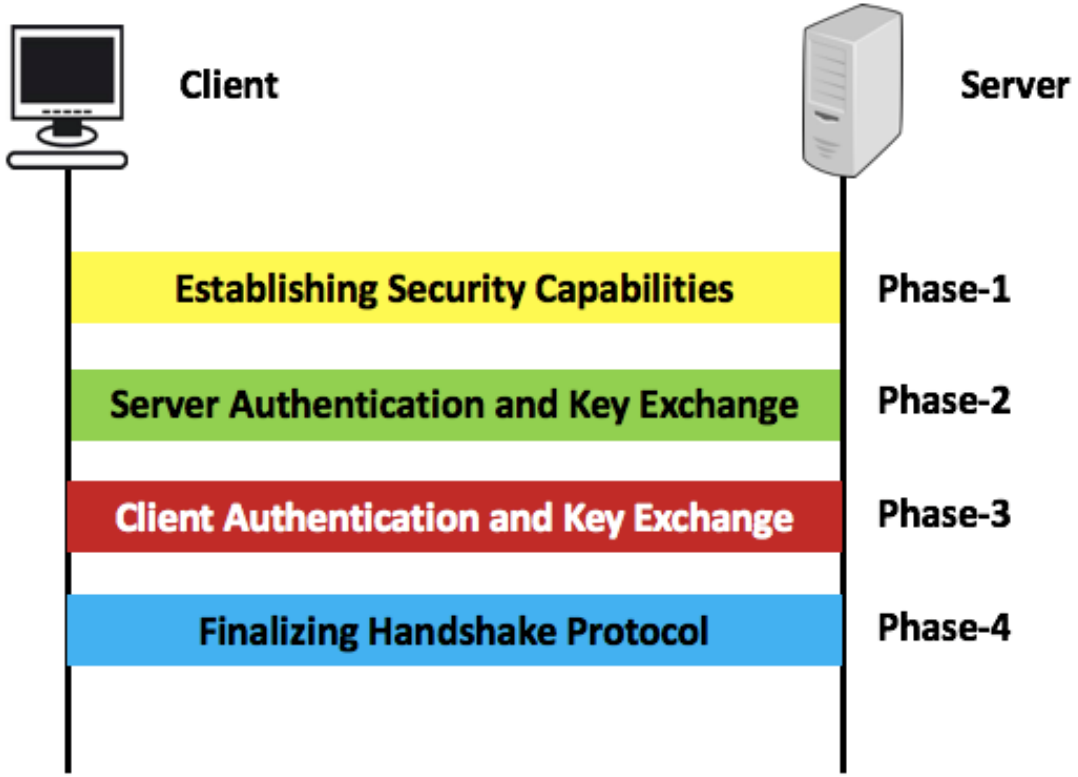
https://www.mysoftkey.com/security/ssl-protocol-overview/ SSL 개념 정리

SSL의 작동 과정

악수(Handshake) -> 전송(Session) -> 통신 종료

https://www.mysoftkey.com/security/4-phases-of-ssl-protocol/
https://opentutorials.org/course/228/4894 SSL 동작방법에 대한 설명

SSL Handshake의 작동 과정



목표:

데이터 전송(Session) 때 사용할 대칭 키를 안전하게 확보하자!

Handshake_phase 1

Client

Server

Client_Hello

- Client Random
- 암호화 방식 (저는 A,B,C 방식 사용할 수 있어요!)
- 세션 id(server가 client를 구분할 수 있도록 하는 임시 이름)

Server_Hello

- Server Random
- 암호화 방식 (그 중에서 가장 안전한 B로 가죠)
- (세션 id)

Handshake_phase 2

Client

Server

Server_Hello

- 인증서

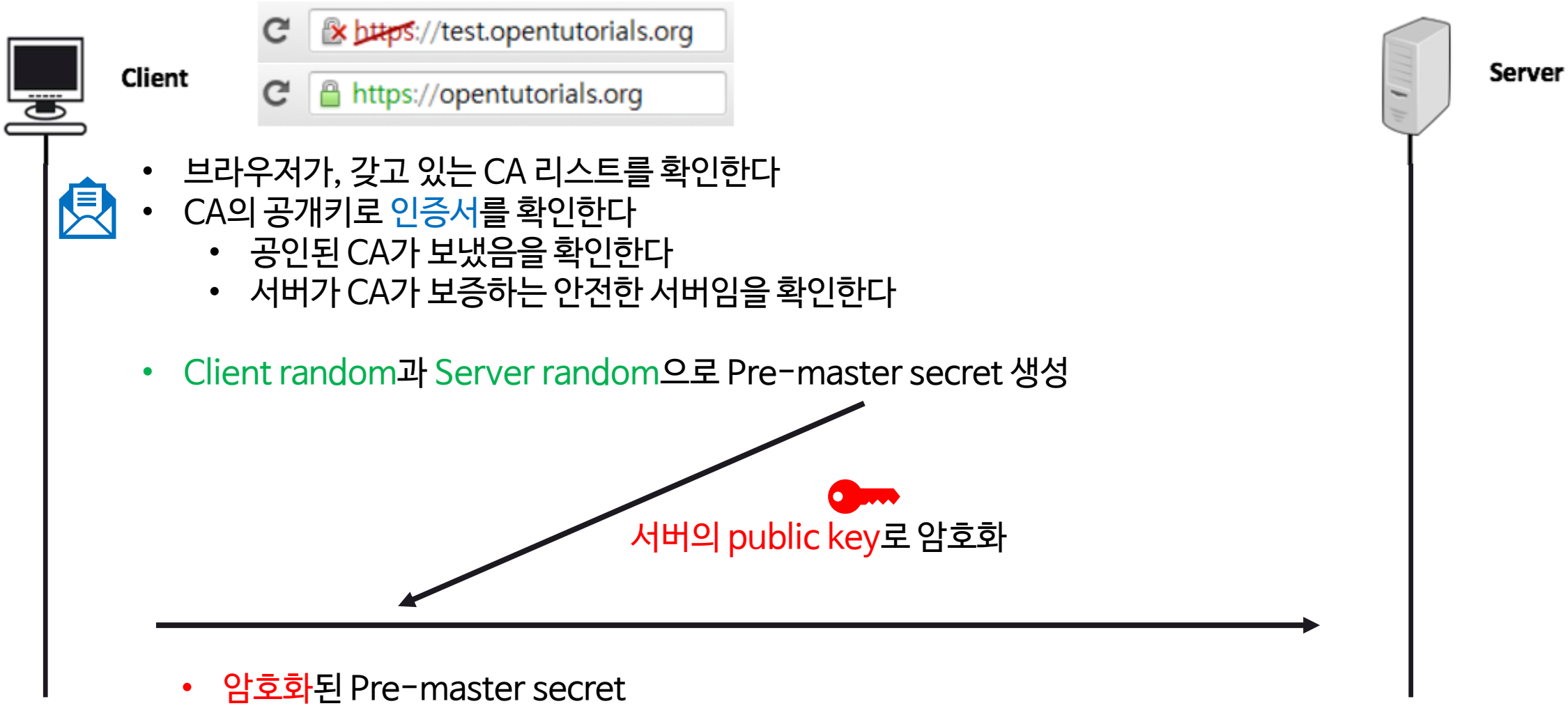
- 누가 발급해줬는가(CA: Certificate Authority == Root Certificate)
- 누구에게 발급해줬는가(도메인)
- 서버의 public key

- CA만의 비밀키로 서명되어 전송

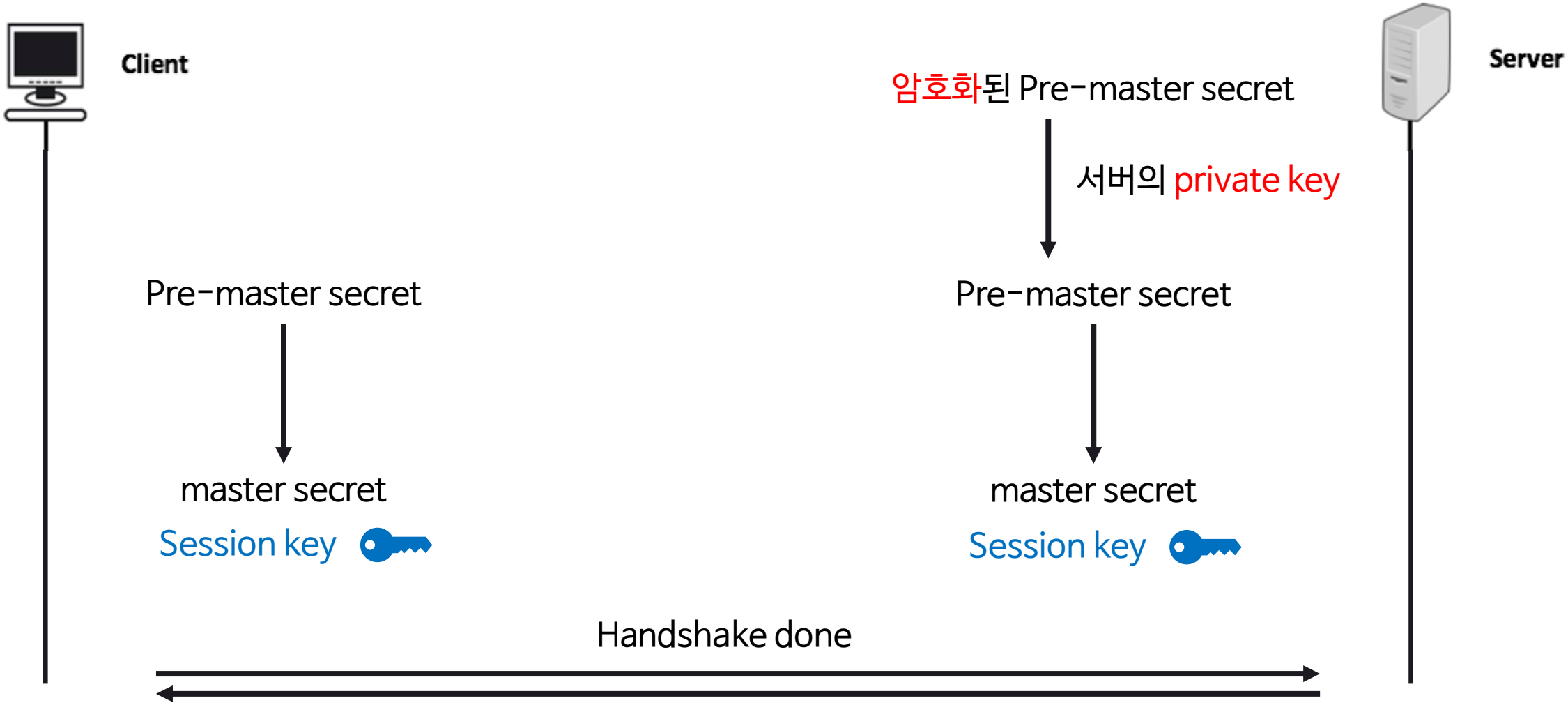
Server_Hello _Done

https://en.wikipedia.org/wiki/Certificate_authority CA 목록 볼 수 있음

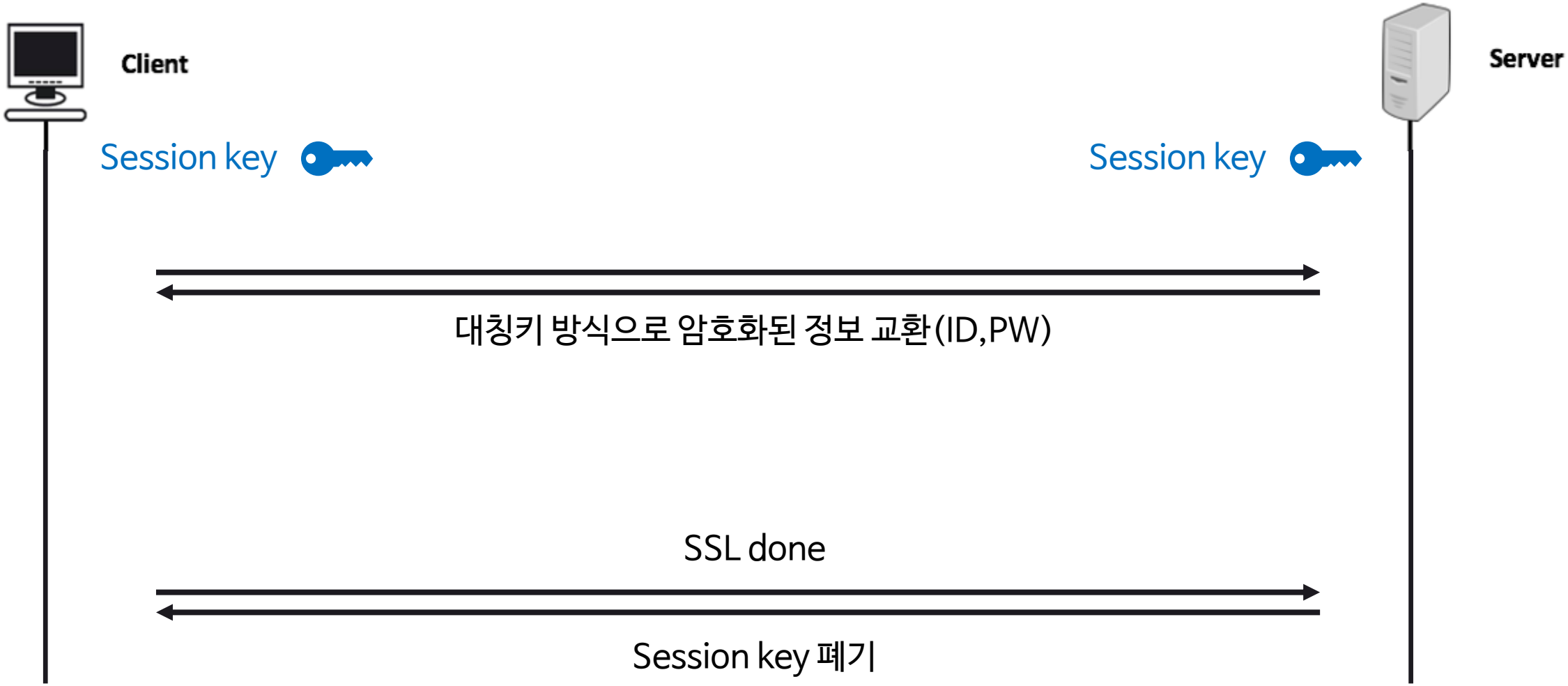
Handshake_phase 3



Handshake_phase 4



전송(Session)및 통신 종료



SSL 정리

- Client와 Server의 정보교환시 : Session key를 통한 대칭키 방식
- 대칭키를 나눠 가질 때: CA가 발급한 인증서를 통한 공개키 방식

- 대칭키의 문제점: key가 유출되었을 때 보안에 치명적
- 공개키의 문제점: 성능의 저하/서버 부하

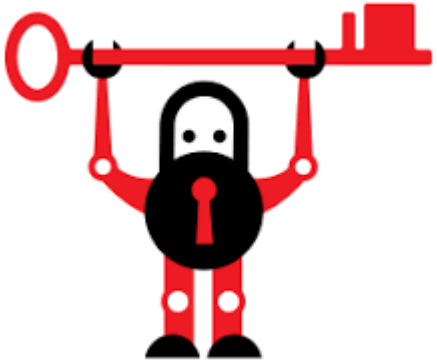
SSL은 대칭키와 공개키의 장점만을 취한 이상적인 통신 프로토콜이다

물론 SSL도 보안공격을 받을 수 있다
더 찾아보기: MiTM, SSL strip, HTTP/2

SSL 실습

tong.sparcs.org에 https를 달아보자!

- Nginx를 설치하고 서버 start
- Certbot을 이용한 인증서 발급
  - Let's Encrpyt: HTTPS의 확산을 늘리기 위해 시작된 비영리 프로젝트



tong.sparcs.org에 https를 달아보자!

1. Certbot install
   - 최신 버전으로 받는 것이 매우 중요하다! => PPA(Personal Package Archive)

   - Sudo apt-get install software-properties-common
   - Sudo add-apt-repository ppa:certbot/certbot
   - Sudo apt-get update

   - Sudo apt-get install python-certbot-nginx

tong.sparcs.org에 https를 달아보자!

2. Certbot에게서 인증서 받기
   - Sudo certbot --nginx -d [받고싶은 도메인]

```
Please choose whether or not to redirect HTTP traffic to HTTPS, removing HTTP access.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
1: No redirect - Make no further changes to the webserver configuration.
2: Redirect - Make all requests redirect to secure HTTPS access. Choose this for
new sites, or if you're confident your site works on HTTPS. You can undo this
change by editing your web server's configuration.
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
Select the appropriate number [1-2] then [enter] (press 'c' to cancel): 1
```

   - HTTP로 들어왔을 때 HTTPS로 redirect 할 것인가? (YES:1/NO:2)

3. https://tong.sparcs.org로 접속하여 인증서를 확인한다!

공격

보안을 위협하는 공격들

- BFA
- Dictionary attack
- Dos/DDos
  - SYN flooding
- Web based attack
  - SQL injection
  - XSS
  - CSRF

BFA(Brute Force Attack)

- 가능한 모든 방법을 다해보는 공격이다
  - 비밀번호를 0000부터 9999까지 다 넣어보기

- 방어법
  - 시간이 오래걸리게 한다
  - 암호 길이를 길게 한다
  - 가능한 key space를 늘린다
  - 시간 당 대입 횟수를 제한한다

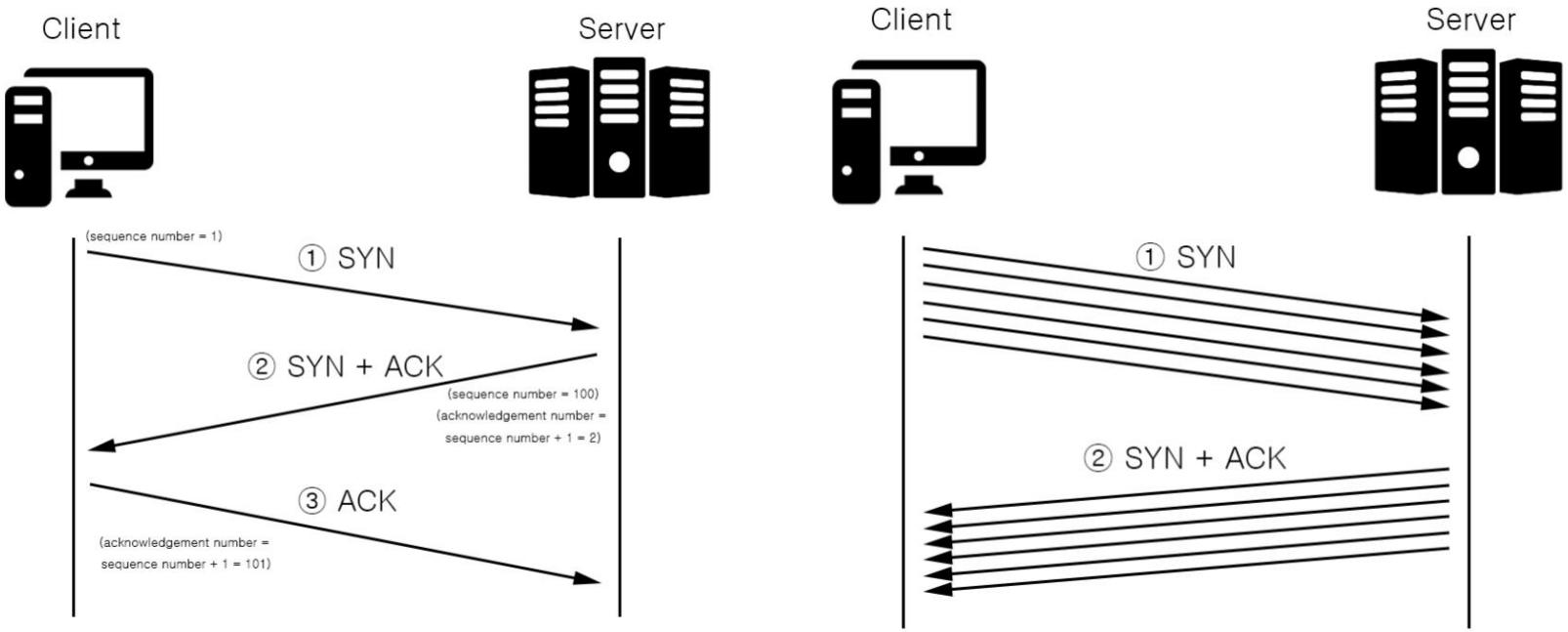
Dictionary attack

- 사용자 이름, 생일, 흔히 쓰는 영단어 들을 조합하여 공격한다
  - Junsung0510, dbswnstjd, thisispassword

- 방어법
  - 쉬운 비밀번호를 쓰지 않는다

DoS/DDoS

- (Distributed) Denial of Service
- 한정된 네트워크 자원을 모두 소모시켜 정상적인 사용자들의 접근을 제한하는 공격



https://sata.kr/22 SYN Flooding

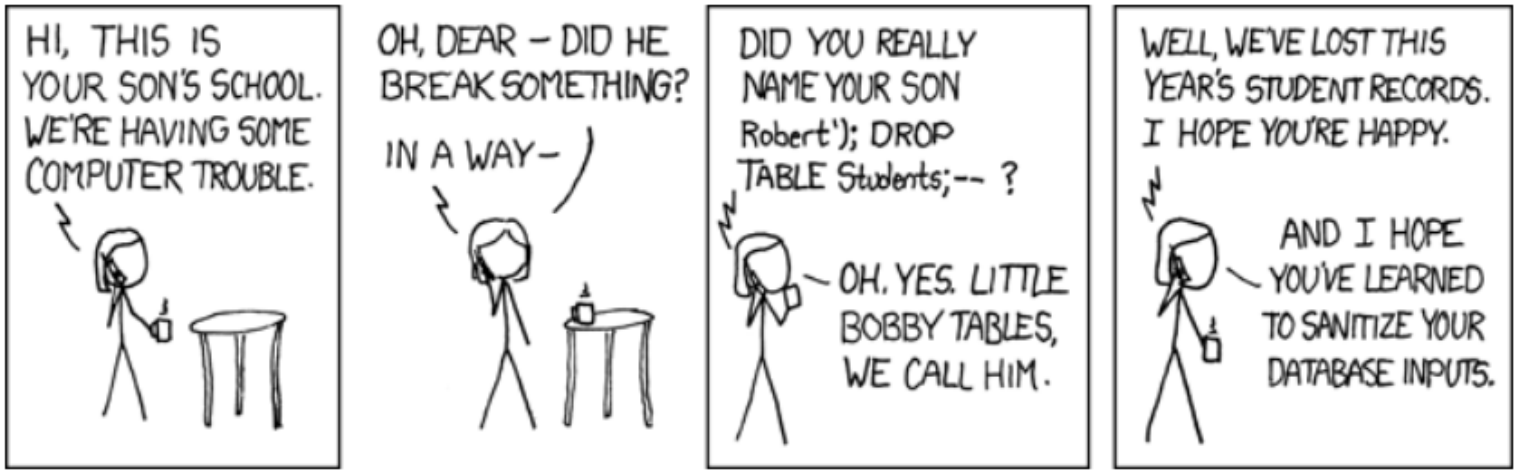
Web-based attack

- 웹페이지의 입력공간에 공격 문자열을 삽입한다

- SQL injection : 서버에서 본인이 원하는 SQL 구문을 실행
- XSS(Cross Site Scripting): 원하는 〈script〉를 사용자가 실행
- CSRF(Cross Site Request Forgery): 사용자가 의도치 않게 서버에 특정 행위를 요청시킴

SQL injection 예시

- Pw에 ‘OR ‘1’=‘1을 입력하면?
  - SELECT id FROM user WHERE id=‘ lulu ’ AND pw=‘ ’OR ‘1’=‘1 ’



https://xkcd.com/327/

  - INSERT INTO students (name) values (‘ Robert’);DROP TABLE Students;-- ’);

XSS 예시와 방어법

- ⟨script⟩alert(‘lulu is handsome’)⟨/script⟩

- SQL injection과 XSS 방어법
  - 기본적으로 입력을 필터링해야 한다
  - Ex) ⟨,⟩를 막는다

CSRF의 예시

- http://test.com/MyAccount?newPW=luluishandsome
- http://bank.com/Withdraw?money=99999999&sendTo=lulu
- 방어법
  - GET,POST 등을 적절히 활용한다
  - 서버쪽에서 random 토큰을 보내고, 사용자에게서 요청을 받을때마다 토큰을 확인한다 : csrf-token

Linux의 보안

Fail2ban

- Linux 사용자 로그인에서 비밀번호를 n번 이상 틀리면 m초 동안 로그인 할 수 없도록 하는 프로그램
  - SSH, FTP 로그인

```
29 # "ignoreip" can be an IP address, a CIDR mask or a DNS host. Fail2ban will not
30 # ban a host which matches an address in this list. Several addresses can be
31 # defined using space separator.
32 ignoreip = 127.0.0.1/8   <- 입력하신 IP로 접근시 아무리 실패해도 차단되지않습니다
33
34 # External command that will take an tagged arguments to ignore, e.g. <ip>,
35 # and return true if the IP is to be ignored. False otherwise.
36 #
37 # ignorecommand = /path/to/command <ip>
38 ignorecommand =
39
40 # "bantime" is the number of seconds that a host is banned.
41 bantime  = 600   <-일정횟수 초과시도시 접근거부 시간입니다. (단위 : 초)
                        *tip : -1(영구차단)로 설정가능
42
43 # A host is banned if it has generated "maxretry" during the last "findtime"
44 # seconds.
45 findtime  = 600 <- 입력하신 시간간격 사이에 지정횟수를 초과시 차단합니다.(단위 : 초)
46
47 # "maxretry" is the number of failures before a host get banned.
48 maxretry = 3 <- 입력하신 횟수초과시 접근거부합니다.
```

위설정대로라면 127.0.0.1/8 ip는 PW 가 틀려도 거부가 되질 않고, 600초 동안 3회 틀릴시 600초 동안 해당 ip는 접속이 거부됩니다.

https://skibis.tistory.com/50

iptables

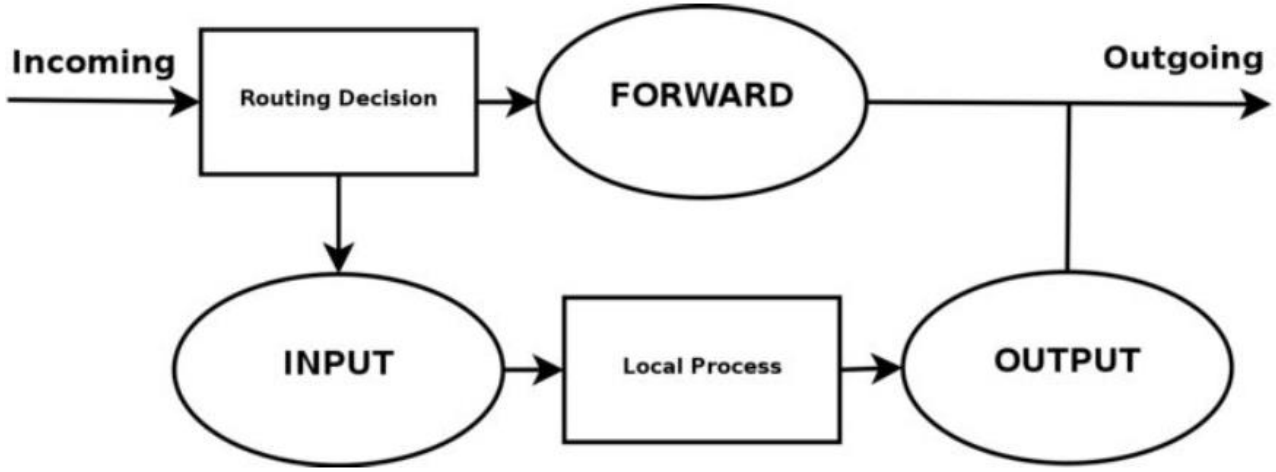
- 리눅스 상의 방화벽 설정 도구

- 패킷필터링을 사용자 공간에서 제어할 수 있다
  - 패킷 = 헤더 + 데이터
  - 헤더에 들어있는 정보: 출발지 IP:PORT, 도착지 IP:PORT, 프로토콜 옵션 등
  - 이 정보들로 패킷을 필터링 하는것이 패킷필터링

iptables

- Iptables를 통과하는 패킷은 아래 3가지로 나뉜다
  - Chain INPUT : 서버로 들어오는 패킷
  - Chain OUTPUT : 서버에서 나가는 패킷
  - Chain FORWARD : 서버에서 forwarding되는 패킷

- Iptables는 이들을 각각
  - ACCEPT : 통과
  - DENY: 메시지와 함께 거절
  - DROP : 패킷 무시
  - 의 정책을 매길 수 있다



Iptables 사용하기

- Sudo apt-get install iptables-persistent
- /sbin/iptables -L : 현재 규칙 보기
- /sbin/iptables -P [chain] [정책] : 기본정책을 변경
- /sbin/iptables -F [chain] : 해당 정책의 세부 규칙 삭제
- 그 밖에도 많은 명령어들이 있다

https://linuxstory1.tistory.com/entry/iptables-기본-명령어-및-옵션-명령어
iptables의 개념과 옵션 명령어들이 있으니 참고

Iptables 사용하기

- 활용예시
  - 소스 ip가 192.168.0.111인 접속을 모두 막아라
  - /sbin/iptables -A INPUT -s 192.168.0.111 -j DROP

- 순서가 중요하다
  - 192.168.100.0/24에서 오는 패킷을 드랍하도록 지정
  - 192.168.100.13으로 들어오는 패킷을 모두 허용하도록 지정

  => 두번째 규칙은 무시된다!

감사합니다