

# Review for CSEC

---

By: Amjad Hasan

Via Dr.aseem's & Dr.Hafez's slides

# Cybersecurity definition and the modules

---

# Cybersecurity

---

- Cybersecurity:

mechanisms to protect the system and the sensitive information from digital attacks

And to achieve this goal there is many modules we could implement and the most popular two is CIA-AAA and Parkerian hexad

# CIA-AAA

- Confidentiality: Keeping information **secret from all**, but those who are authorized to see it.

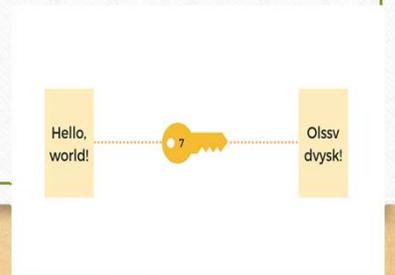
المحافظة على سرية الرسالة

- Attack on confidentiality: **interception** (الاعتراض)

- Example:

ارسال رسالة من شخص الى اخر و من ثم يقوم شخص غير مصرح له برؤية الرسالة

A way to strengthen the confidentiality is by using cryptography



# Integrity

---

- Ensuring that information **has not been altered** by unauthorized entities.

التأكد من أن المعلومات لم يتم تغييرها من قبل اشخاص غير مصرح لهم.

Attack on integrity: **Modification** (التعديل)

A way to defense against the modification is by using **hashing** algorithm

# Availability

---

- Assuring that system is **available** when needed

التأكد أن النظام متوفر و مثال: عند البحث عن موقع رتاج يكون متواجد

- Attack on availability ( **interruption** ) الانقطاع او التعطيل
- Example: DOS , DDOS Attacks
- A way to prevent the interruption is by **clustering** the server

# Authentication

---

- Process of **verifying** the identity of the user.
- Attack on authentication: **Fabrication**
- How to prevent the Fabrications?
- By implementing biometrical system
- **Note:** attack at the authentication leads to Authorization attack

# Authorization

---

- The mechanism of granting user a **privileges**
- Attack on authorization: getting privileges that is not allowed
- To prevent it we use ACLs mechanism

# Accounting

---

- the mechanism of making sure that an action of an entity in a system is **traceable**
- So every actions happen in you server should be tracible ( when the user entered it , for how long , what he did in the server )
- Attack on accounting: **Denying**
- so you should make a **log file as an evidence.**



# Test !!!

---

- An attacker saw a message, and he is authorized to see it so what type of CIA he attacked ??
- An attacker modified a message between two, and he is not authorized to do it so what type of CIA he attacked and is it Passive or Active ??
- An attacker saw a message between two, and he is not authorized to see it so what type of CIA he attacked and is it Passive or Active ??

# The Parkerian Hexad

---

- It contains of CIA and APU
- A: **Authenticity**: to know who you are talking/dealing with
- P :**Possession**: to **physically possess** the device that have the information on it

Ex: Laptop that have a sensitive Data on it Should be with you and safe.

**Utility**: **measurement** of usefulness,



# Some definitions

---

- **Cyber attack**: is an action that exploits a vulnerability in a system.
- **Threat**: is a constant danger that can harm the assets
- **Assets**: reputation , software , hardware , reputation , people
- **Vulnerability** is an identified weakness or flaw of an asset whose controls are not present, or are no longer effective

# The attackers

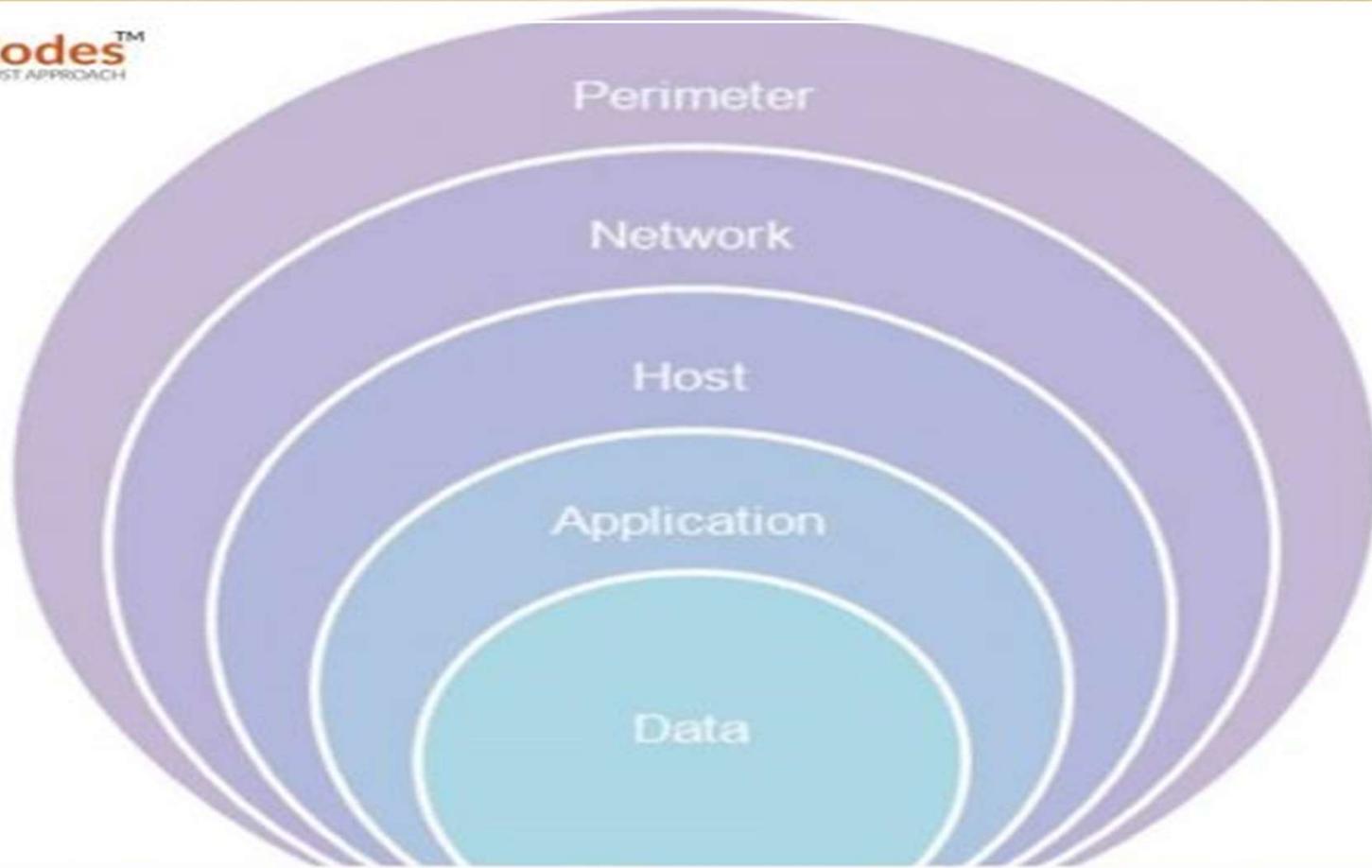
---

- Elite hackers: White, Black , Gray
- Script kiddies
- Insiders (the most dangerous)

# Types of Attacks

---

- *Passive Attacks:* do not require modifying
- *Active Attacks* require modifying
- *Brute Force Attack* the strongest but slower
- *Dictionary Attack* faster but brute force is stronger
- *Denial-of-service (DoS) Attack* 'smurf attack'
- *Distributed Denial-of-service (DDoS) Attack* Bots
- *Man-in-the-Middle (MITM) Attack* could be passive or active



## What Is Defense in Depth?



# Social Engineering

- Phishing
- Vishing (voice phishing)
- Smishing (SMS phishing)

People are the weakest link.



# Authentication

# Authentication , Identification

---

- Authentication verifying a claimed identity

اختبار هوية الفرد

- Identification: establishing an identity

انشاء هوية للفرد

# Authentication vs identification

---

- **Authentication** (also called verification)

Identity is provided

Is he really who he claims to be?

One-to-one verification

- **Identification**

No identity is provided

Who is he?

One-to-many

# Authentication Factors

---

Currently we have 3 authentication factors:

- Know: something only you remember    password, PIN
- Have: something only you possess        ID card , Passport
- Are: some biometric property            FingerPrint , Iris

# Parties involved

---

- The authenticator (or user).
- The verifier.
- The attacker.

# Biometrics

# Definition

---

- "Biometric Technologies" are
- automated methods

of verifying or recognizing the identity of a living person based on a **physiological** or **behavioral** characteristic

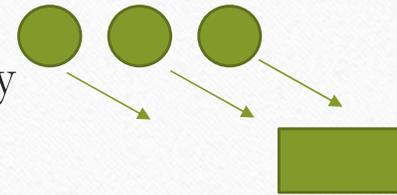
Physiological biometrics Examples: Fingerprint, Iris, Face, Hand

Behavioral biometrics Examples: Signature, Gait, Voice

# Positive / Negative

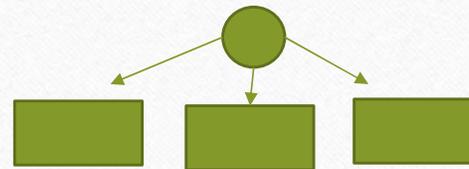
- Positive recognition:

To prevent **multiple** people from using the **same** identity



- Negative recognition:

To prevent **one** person from using **multiple** identities



Circle: user    rectangle: identity

# Physiological / Behavioural

---

- Physiological: Physical features "unchangeably" attached to a person  
Examples: fingerprint, DNA, and face

- Behavioural : Behaviour that is very specific to a person  
Example signature, gait, and voice

# Characteristics

- This Characteristics we use to determine weather the system is good and can be implemented or not ( PPCCUDA )
- 
- P : **Permanence**: الديمومة like fingerprint and DNA
  - P : **performance**: الأداء it should be accuracy and speed so is DNA good ?
  - C : **Collectability**: it should be easy to collect is DNA good here ?
  - C : **Circumvention**: How easy to fool the system
  - U : **Universality**: every human should have the characteristic
  - D : **Distinctiveness** : التميز Different persons should have different biometric properties
  - A : **Acceptability**: if we did the DNA test is everyone will accept it ??

# Errors

---

- Errors that may happen to any biometrical system
- False Non-Match Rate (FNMR)
- False Match Rate (FMR)
- Failure to Enroll Rate (FER)
- Failure to Capture Rate (FCR)

# FMR , FNMR

---

- False Match Rate (FMR)

false claimed identity is not recognized as false

Ex : شخص ليس من الجامعة و دخل من البوابة بدون أية مشاكل بسبب خطأ من جهاز البصمة

False Non-Match Rate (FNMR)

Probability that a correctly claimed identity is not recognized as true

Ex: شخص من الجامعة ولم يتم قبول دخوله لها بسبب خطأ في جهاز البصمة

# FER , FCR

- Failure to Enroll Rate (FER)

Probability that a person **cannot enroll** in the biometric system ( الإنخراط )

Ex: طالب بدون يد فلا يستطيع الفحص في جهاز البصمة

- Failure to Capture Rate (FCR)

Probability of **failure to capture** the biometric feature when trying to authenticate

يكون خطأ من الجهاز نفسه مثلاً لو كان لفحص البصمة قد يكون متسخ او جهاز التعرف على الصوت قد يكون هناك ضجيج حول المكان

## lets match the errors with the Characteristics

---

- Universality : FER
- Distinctiveness : FMR
- Permanence: FNMR
- Performance: FNMR, FCR
- Circumvention: FMR

**Note:** EER is the point where FMR and FNMR are equal

# Application Environments

---

- Overt vs. covert      Overt: user is *Aware* ,    Covert: user in *UnAware*
- Habituated vs. non-habituated      Habituated: used daily
- Attended vs. non-attended      Attended: guided by system
- Standard vs. non-standard      standard: how should the environment be
- Public vs. private      public: anybody can use the system
- Open vs. closed      Open: System can interact with other (biometric) system

# Biometrical Systems

---

- Enrollment module

Template created and stored in **database**

- Authentication module

**Checked** against stored template

So the enrollment is before the authentication

# Threshold

---

- $d \leq t$ : authentication OK
- $d > t$ : authentication NOT OK

	Templ 1	Templ 2	Templ 3	Templ 4	Templ 5
Test 1	<b>0,182</b>	0,588	0,435	0,208	0,909
Test 2	0,323	<b>0,213</b>	0,286	0,476	0,244
Test 3	0,909	0,625	<b>0,147</b>	0,476	1,111
Test 4	0,238	0,294	0,476	<b>0,256</b>	0,526
Test 5	0,588	0,454	1,250	0,526	<b>0,130</b>

- ( القيم الأكبر من ثري شولد في القطر تقسيم عددهم ) FNMR =
- ( القيم الأصغر من ثري شولد خارج القر على عددهم ) FMR =
- If the 'Threshold equals 0.213 'Then
- FNMR = 1/5
- FMR = 1/20

# Distance metrics

---

- Absolute Distance:  $d1(x,y) = \sum |x_i - y_i|$
- Euclidean Distance:  $d2(x,y) = \sqrt{[\sum (x_i - y_i)^2]}$
- Maximum Difference Distance:  $d3(x,y) = \max |x_i - y_i|$

# Example

---

- $X(3, 7, 4)$                        $Y(2, 5, 9)$
- Find (Absolute Distance Euclidean Distance Maximum Difference Distance)
- Abs distance =  $(1 + 2 + 3) = 6$
- Euc =  $(1 + 4 + 9) = \sqrt{14}$
- Max = 3

Password ):

# :Password

## Kinds of passwords

---

- Password
- A string of characters: A,B,C,...d,e,f,...1,2,3...!,",@,...
- 4-digit PIN codes:  $s = |S| = 10^4$
- 6 character passwords:
  - $s = 26^6$                       Small letters = 26
  - $s = 52^6$                       Capital letters = 26
  - $s = 62^6$                       digits = 10
  - $s = 94^6$                       special characters = 32

# Password:

## The art of counting

---

- Number of 5 letter combinations:  $26^5$
- Including capitals:  $52^5$
- Including numbers:  $62^5$
- All keyboard symbols:  $94^5$

Lets talk about some cases

---

# The position rule

---

- So the position is a VERY IMPORTANT rule
- It let us know how many spots gonna take in the box
- Ex: how many position of 3 out of 5
- The answer is  $(5!) / ((5-3)! * (3!)) = (5!) / (2!) * (3!)$
- $= (5*4*3*2*1) / (2*1) * (3*2*1) = 5 * 2 = 10$
- How we find it ??9

# The position rule

- So the rule is if we want to know how many position it will

$$= \frac{n!}{r! (n-r)!}$$

Take for r out of n the answer will be  $(n!) / ((r!) * (n-r)!)$

Another example lets say 4 out 7



$$(7!) / ((4!) * (7-4)!) = (7!) / ((4!) * (3)!) = (7*6*5*4!) / (4!) * (3*2*1)$$

$$= 7$$

All good ?

# The (Exact case)

- Now how to find the examples that asks for exact (digit/letter/special C)
- Ex: how many 5 characters passwords that have exactly 2 Capital letter ?

• First draw the Box

capital	capital	94-26=68	68	68
---------	---------	----------	----	----

- ولكن الترتيب ممكن يختلف يعني ممكن الكابيتال يكون في اخر خانه عشان هيك بنطبق قانون البوزيشن
- So for the position  $(5!) / ((2!) (3!)) = (5*4*3!) / (2*1) (3!) = 5*2 = 10$
- Now for the rest will be =  $68 * 68 * 68 * 26 * 26$
- So for position 10 and for capitals  $26*26$  and for the rest  $68*68*68$
- So the answer is  $86^3 * 26^2 * 10$

# Another example

- Exactly one number and one capital for 5 characters

• The box: 

number	capital	Rest	Rest	rest
--------	---------	------	------	------

- The position: 5 for number and 4 for capital
- the number = 10
- The capital = 26
- The rest =  $94 - 10 - 26 = 58 * 58 * 58$
- So the answer is =  $(5*10) * (4*26) * (58^3)$

## The (AT least) case first condition

---

- If it was (AT least one) there is a rule which is (all – those are wrong)
- So Ex: how many 5 characters there that contain at least 1 number
- Answer is = all (  $94^5$  ) those are wrong ( all without the digits ) =  $84^5$
- So the answer is =  $94^5 - 84^5$

# The (AT least) case second condition

---

- The ( AT least more than one ) here we will use the position rule and the EXACT case
- Ex: how many 5 characters contain at least 3 Capital letter
- Answer is = Exactly 3 + Exactly 4 + Exactly 5

# Exactly 3

- Exactly 3:

Capital	Capital	Capital	68	68
---------	---------	---------	----	----

- First the box
- Now the position :  $(5!) / ((3!) * (2!)) = 10$
- Now the capital =  $26 * 26 * 26$
- The rest =  $68 * 68$
- So the answer is  $(26^3) * (68^2) * 10$

# Exactly 4

- Exactly 4:

Capital	Capital	Capital	Capital	68
---------	---------	---------	---------	----

- First the box
- Now the position :  $(5!) / ((4!) * (1!)) = 5$
- Now the capital =  $26 * 26 * 26 * 26$
- The rest = 68
- So the answer is  $(26^4) * (68) * 5$

# Exactly 5

- Exactly 5:

Capital	Capital	Capital	capital	capital
---------	---------	---------	---------	---------

- First the box
- Now the position :  $(5!) / ((5!) * (0!) = 1$
- Now the capital =  $26 * 26 * 26 * 26 * 26$
- The rest = 1
- So the answer is  $(26^5) * 1$

# The answer

---

- The answer is
- $(26^3) * (68^2) * 10 + (26^4) * (68) * 5 + (26^5)$

# The (HARD AT least)

- Ex: at least 1 number and one capital = all – wrong + subtract twice

Number = 10 ✓

Capital = 26 ✗

Small = 26 ✗

Special = 32 ✗

number = 10 ✗

capital = 26 ✓

small = 26 ✗

special = 32 ✗

$$94^5 - 84^5 - 68^5 + 58^5$$

## Password: Combinatorics - 2

---

- At least 1 number?
  - Total number of 6 character passwords:  $94^6$
  - Number of 6 character passwords without numbers:  $84^6$
  - Answer:  $94^6 - 84^6 = 338.571.749.440$
- Trick: All – those that are wrong

# Password:

## Combinatorics - 3

---

94	93	92	91	90	89
----	----	----	----	----	----

- Have 6 different characters?
  - First character: 94 possibilities
    - Second character:  $(94-1)$  possibilities
    - Third character:  $(94-2)$  possibilities
    - Answer:  $94*93...*89 = 586.236.072.240 =$
  - Trick: Count every time what is still possible

# Password: Combinatorics - 4

- At least 1 capital and 1 number?
  - No restrictions:  $94^6$
  - No capitals:  $68^6$
  - No numbers:  $84^6$
  - No capitals and no numbers:  $58^6$
  - Answer:  $94^6 - 68^6 - 84^6 + 58^6 = 277.772.959.360 = 2^{38,02}$
- Trick: All – wrong ones + those subtracted twice!

# Password:

## Combinatorics – 5

---

- Exactly 1 number?
  - Choose position where the number will be: 6 possibilities
  - Number on that position: 10 possibilities
  - All other 5 positions:  $(94-10)$  possibilities
  - Answer:  $(6*10) * 84^5 = 250.927.165.440$   
Trick: Place number first.

oV

# Password:

## Combinatorics - 6

---

- Exactly 1 number and exactly 1 capital?
  - Choose position for the number: 6 possibilities
  - Number on that position: 10 possibilities
  - Choose position for the capital: (6-1) possibilities
  - Capital on that position: 26 possibilities
  - All other 4 positions:  $(9 \cdot 10 \cdot 26)$  possibilities
  - Answer:  $(6 \cdot 10) \cdot (5 \cdot 26) \cdot 58^4 = 88.268.668.800$
- Trick: Place number and capital first

# Password:

## Combinatorics - 7

---

- Exactly 2 numbers?
  - Choose 2 positions for the numbers:  $6 \cdot 5 / 2 = 15$  possibilities
  - Numbers on those position: 10 possibilities
  - All other 4 positions:  $(94-10)$  possibilities
  - Answer:  $15 \cdot 10^2 \cdot 84^4 = 74.680.704.000 =$

# Password:

## Combinatorics - 8

---

- Choose 2 positions for the numbers gives 15 possibilities. Why?
- "Choose m out of n":
  - $n! / (m! * (n-m)!)$
  - $k! = 1*2*...*(k-1)*k$
- "Choose 2 out of 6":  $6!/(2!*4!) = 15$

# Password: Probabilities

- What is the probability that a random password of 6 characters has no number in it?
  - Answer:  $84^6 / 94^6 = (84/94)^6 = 0,509$
  - So approximately have of the 6 character passwords does not have a number in it!
- In general is the probability equal to the size of set of correct answers divided by the total number of answers.

# Password: Good Properties

---

- **Hard to guess:** do not use names, dates, telephone numbers, etc.
- **Easy to remember:** no need to write it down or share with other persons
- **Private:** otherwise no authentication possible
- **Secret:** owner is the only one who knows it

# Password: The PROBLEM!

---

- We have limited memory
  - Can only remember  $7 \pm 2$  totally random symbols
- Even more problems when
  - We have multiple passwords
  - We need to change passwords regularly

# Password:

## What can we do – part 1?

---

- Pass phrase
  - Yesterday I watched a nice program on television.
  - YIwanpot or Y1wanp0t
- Use events on news or personal events when forced to change regularly

# Password: Pass faces and images

---

- It is easier to recognize than to remember.

# Cryptology

# Basic Terminology

---

- **plaintext** - original message
- **ciphertext** - coded message
- **cipher** - algorithm for transforming plaintext to ciphertext
- **key** - info used in cipher, known only to sender/receiver
- **encipher (encrypt)** - converting plaintext to ciphertext
- **decipher (decrypt)** - recovering ciphertext from plaintext
- **cryptography** - study of encryption and decryption principles/methods
- **cryptanalysis (codebreaking)** - study of principles/ methods of deciphering ciphertext *without* knowing key
- **cryptology** - field of both cryptography and cryptanalysis

# Cryptography Ciphers

---

- Plaintext can be encrypted through stream cipher or block cipher.
- **Stream** cipher: each plaintext bit transformed into ciphertext bit, one bit at a time (bit by bit) more secure but takes a lot of time
- **Block** cipher: message divided into blocks (e.g., sets of 8- or 16-bit blocks) and each is transformed into encrypted block.(number of bits together)

# Cryptography Techniques

---

- Symmetric Cryptography:

Encryption key = Decryption Key (same)      Example (DES , S-DES)

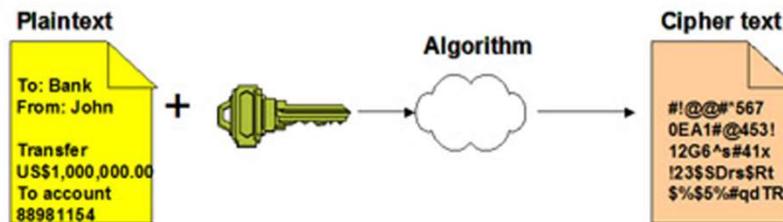
- Asymmetric Cryptography:

Encryption key = Decryption Key (different) also called public key algorithm

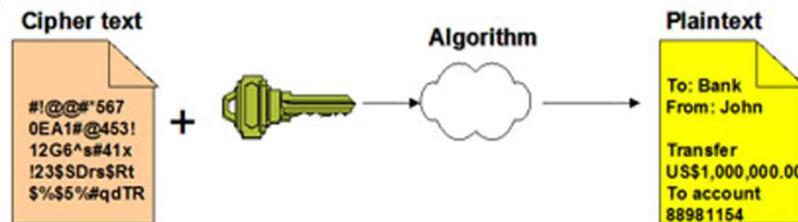
- Example (RSA)

# Test

## Encryption



## Decryption



# Block Cipher

---

## 1. Substitution Cipher:

- A technique in which the letters of plaintext are replaced by other letters or symbols.
- Position of a letter is fixed but its value will be changed.

## 2. Transposition Cipher:

- Value of a letter is fixed but its position is changed.

## 3. Product Cipher:

- Value and position of a letter are changed.

# Substitution Cipher

---

➤ **Mono-alphabetic cipher**

A cipher that uses fixed substitution over the entire message.

➤ **Poly-alphabetic cipher**

A cipher that uses a number of substitutions at different positions in the message.

# Cryptography Key Size

---

- When using ciphers, size of the cryptography key very important
- Strength of many encryption applications and cryptosystems measured by the key size

The security of a cipher should rely  
on the secrecy of the key only!

*Auguste Kerckhoffs, „La Cryptographie militaire“, 1883*

# Classical cryptology

Cryptology

```
graph TD; Cryptology --> Cryptography; Cryptology --> Cryptanalysis; Cryptography <--> Cryptanalysis;
```

Cryptography

*„Art and science of keeping messages secure“*

Cryptanalysis

*„Art and science of breaking ciphertext“*

# Cryptography & cryptoanalysis

---

- Cryptography: The art and science of keeping messages secure  
cryptographers
- Cryptanalysis: the art and science of breaking cipher text.
- Cryptanalysts

# Types of Attacks (cryptanalysis)

---

- Ciphertext-Only Attack:
- Known-Plaintext Attack:
- Chosen-Plaintext Attack
- Adaptive Chosen-Plaintext Attack

# Ciphertext-Only Attack:

- The attacker knows the encryption algorithm and have encrypted message, so he just captured a ciphertext
- Attacker knows cipher text of several messages
- So, he will try to continue sniffing so he can have a plaintext

```
mcx Epvvni Auipvo ngicxuo.  
v yvpa nv n bxfjocixnjku  
pxjxulo. Cmopl yvpa oc plb  
5 bnl nqvc iplpxnop n muli  
n muqp uv okp vnep nmopx  
EA5 knvk uv bcejcvpa cm 32  
Plopx n dcxa ul okp EA5 p  
cd okp bcxxpvjclauli EA5 k
```

# Known Plaintext

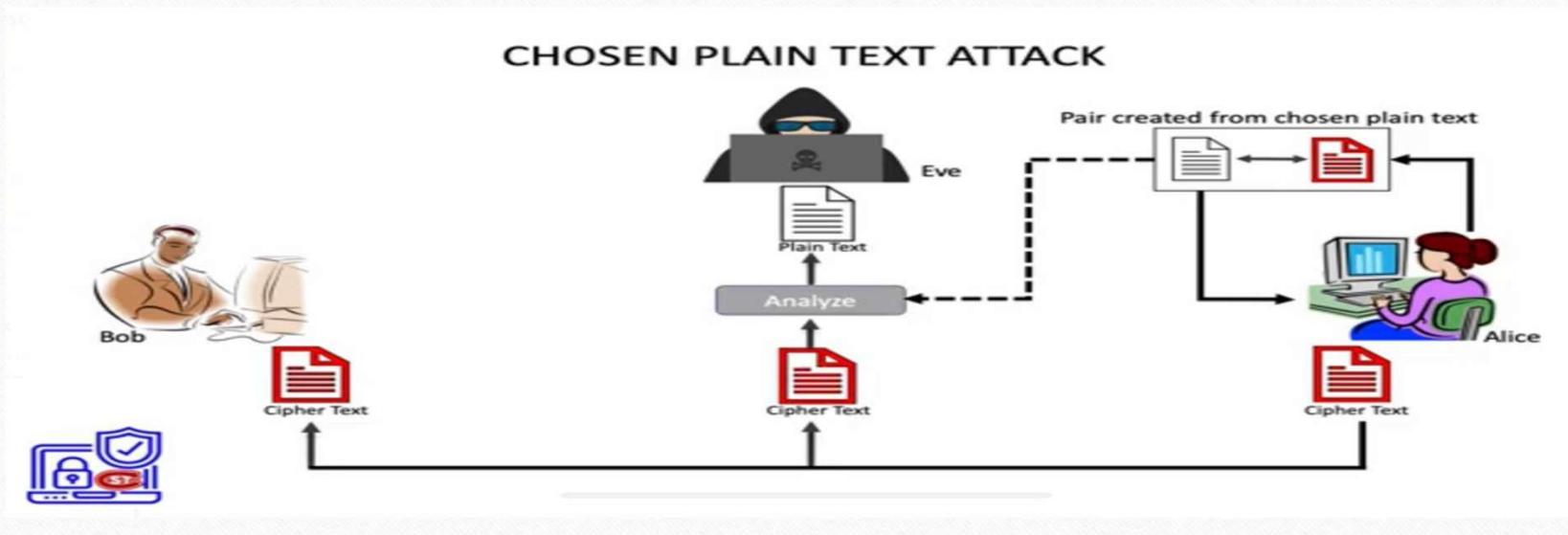
- the attacker have Known cipher text / plaintext pair of several messages.
- Now he will try to know the key of the encryption algorithm

```
mcx Epvvni Auipvo ngicxuo  
v yvpa nv n bxfjocixnjkub  
pxjxulo. Cmopl yvpa oc plb  
5 bnl ngvc iplpxnop n muli  
n muqp uv okp vnep nmopx  
EA5 knvk uv bcejcvpa cm 32  
Plop x n dcxa ul okp EA5 p  
cd okp bcxxpvjclauli EA5 k
```

```
r old boy who lives with his parents; John and  
and his little sister Sandy. They are from Engl  
yes are brown and his hair is black. His brothe  
l his hair is blond. They go everyday to school  
y much.
```

# Chosen-Plaintext Attack

- Attacker can choose the plaintext that gets encrypted thereby potentially getting more information about the key



# Adaptive Chosen-Plaintext Attack

- 
- Same as Chosen-Plaintext Attack but with more several messages

# Summary

Type of Attack	Known to cryptanalyst
Ciphertext Only	<ul style="list-style-type: none"><li>★ Encryption Algorithm</li><li>★ Ciphertext</li></ul>
Known Plaintext	<ul style="list-style-type: none"><li>★ Encryption Algorithm</li><li>★ Ciphertext</li><li>★ One or more PT-CT pairs formed with secret key</li></ul>
Chosen Plaintext	<ul style="list-style-type: none"><li>★ Encryption Algorithm</li><li>★ Ciphertext</li><li>★ PT message chosen by cryptanalyst, together with its CT generated with the secret key</li></ul>

# Caesar Monoalphabetic Substitution Cipher

MESSAGE FROM MARY STUART KILL THE QUEEN

Substitution Table - Caesar's Cipher	
ABCDEFGHIJKLMNOPQRSTUVWXYZ	
↓ ↓ ↓ ↓ ↓	
DEFGHIJKLMNOPQRSTUVWXYZABC	



key = 3 cyclic shifts

PHVVD JHIUR PPDUB VWXDU WNLOO WKHTX HHQ

General Substitution Table	
ABCDEFGHIJKLMNOPQRSTUVWXYZ	
EYUOBMDXVTHIJPRCNAKQLSGZFW	

26! possible keys

JBKKE DBMAR JJEAF KQLEA QHVII QXBNL BBP

# Monoalphabetic Substitution

---

- old and weak but not bad
- 26! Different keys it seem to be secure.
- Substitution using S-Boxes
- Easy to break by the Brute force Attack

# Vigenère Polyalphabetic Substitution Cipher

---

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	plaintext alphabet
A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	
B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	
C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	
D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	
E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	
F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	
G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	
H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	
I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	
J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	
K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	
L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	
M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	
N	O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	
O	P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	
P	Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	
Q	R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	
R	S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	
S	T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	
T	U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
U	V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	
V	W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	
W	X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	
X	Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	
Y	Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	
Z	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	

Vigenère square

Keyword: **WHITE**

**MESSAGE FROM ...**

**WHITEWH ITEW**

**ILALECL NKSI**

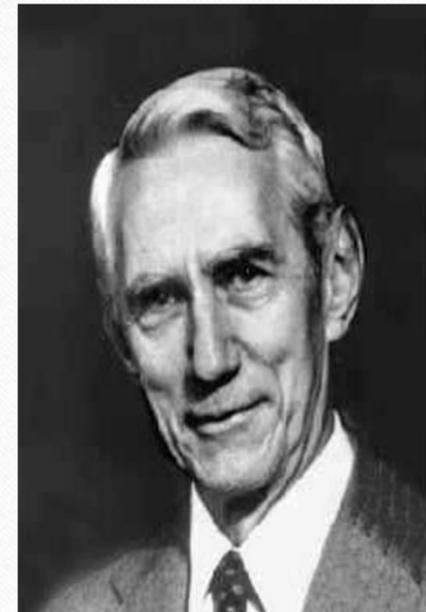
# Claude Shannon 1916 - 2001

- Basic Principles of „Confusion“ and „Diffusion“
- Shannon was the first to formulate these two principles explicitly, „**confusion**“ standing for **substitution** operations and „**diffusion**“ standing for **transposition** or **permutation** operations.

**confusion**

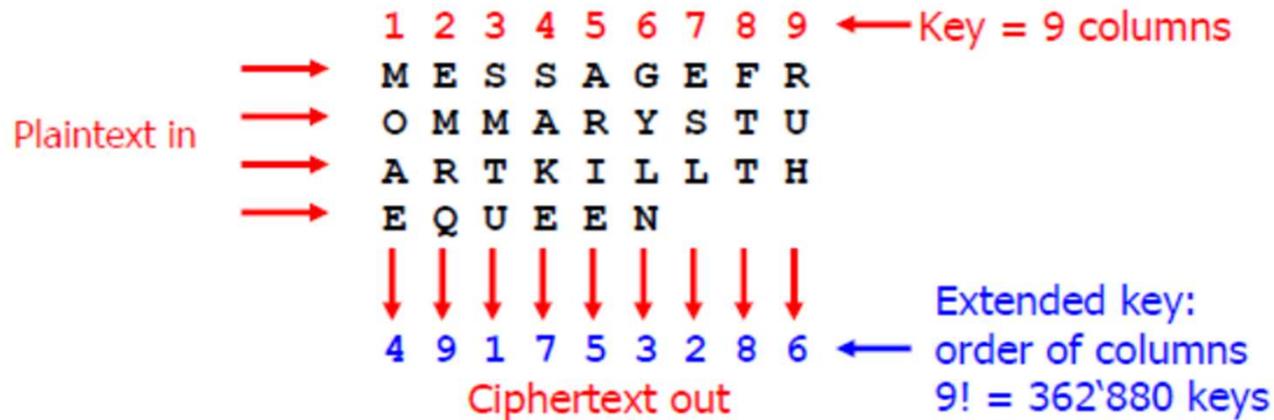
Caesar and Vigenère cipher.

S-Box in modern cipher.



# Transposition Cipher-1

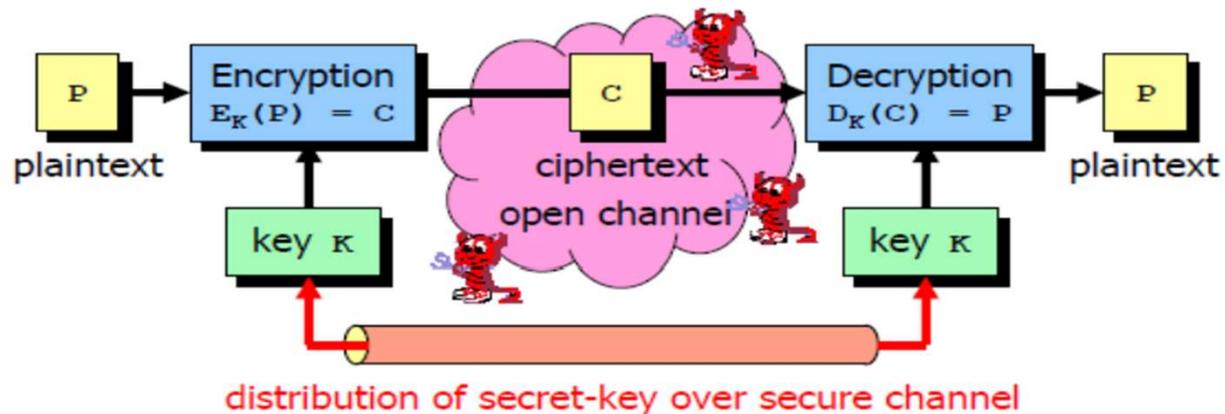
MESSAGE FROM MARY STUART KILL THE QUEEN



MOAEE MRQSM TUSAK EARIE GYLNE SLFTT RUH  
SMTUE SLGYL NMOAE ARIER UHSAK EFTTE MRQ

Diffusion means permutation of bit or byte positions !

# Shannon's Model of a Secrecy System



- Same key used for encryption and decryption
- Key must be kept absolutely secret
- Same key can be used for several messages, but should be changed periodically → **secure key distribution problem!**

# Data Encryption Standard (DES)

# Overview of DES

---

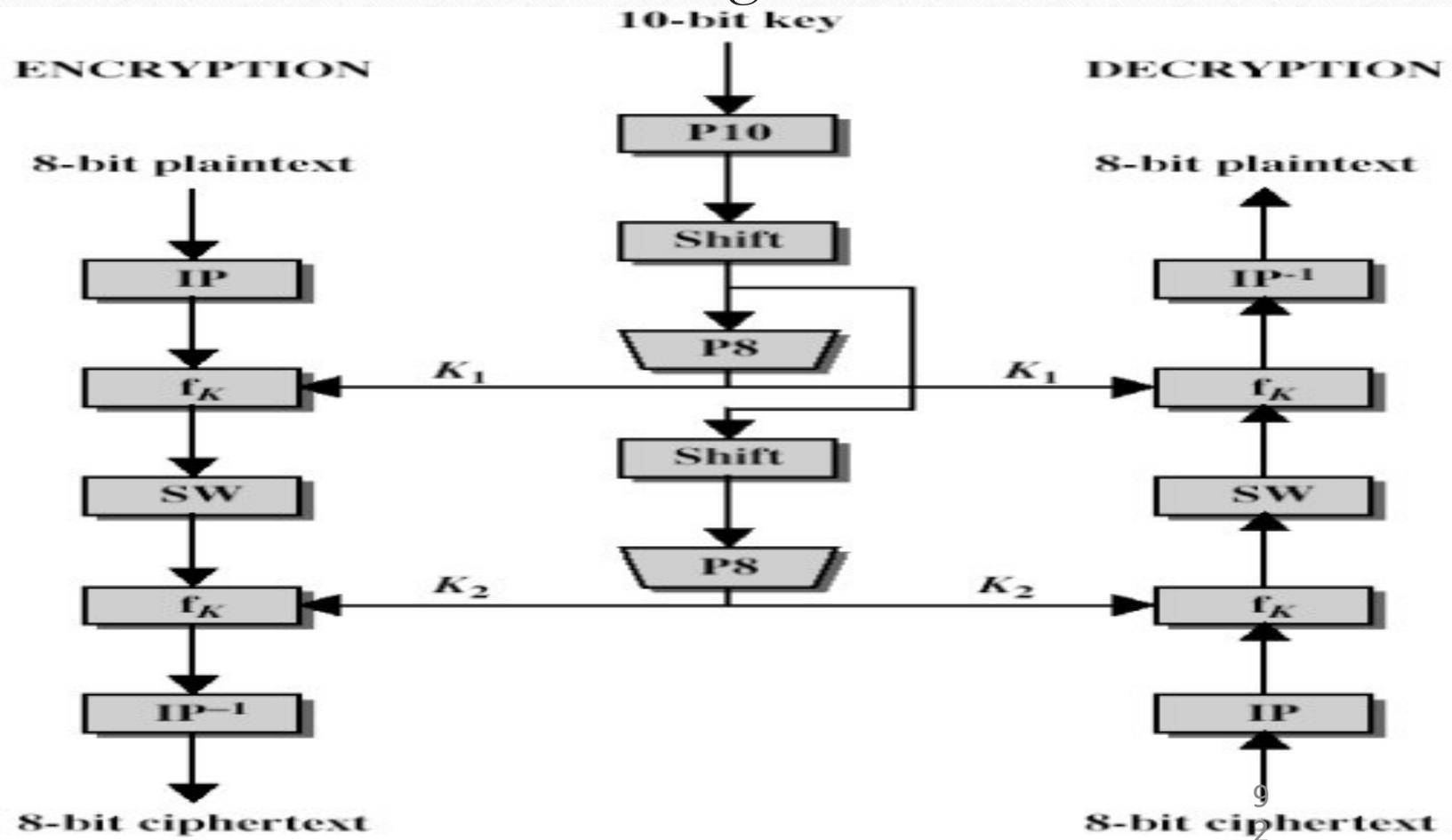
- Symmetric block cipher.
- 56-bit key.
- 64-bit input block, 64-bit output block.
- Developed in 1977 by National Institute of Standards and Technology (NIST); and designed by IBM.

# Simplified DES (S-DES)

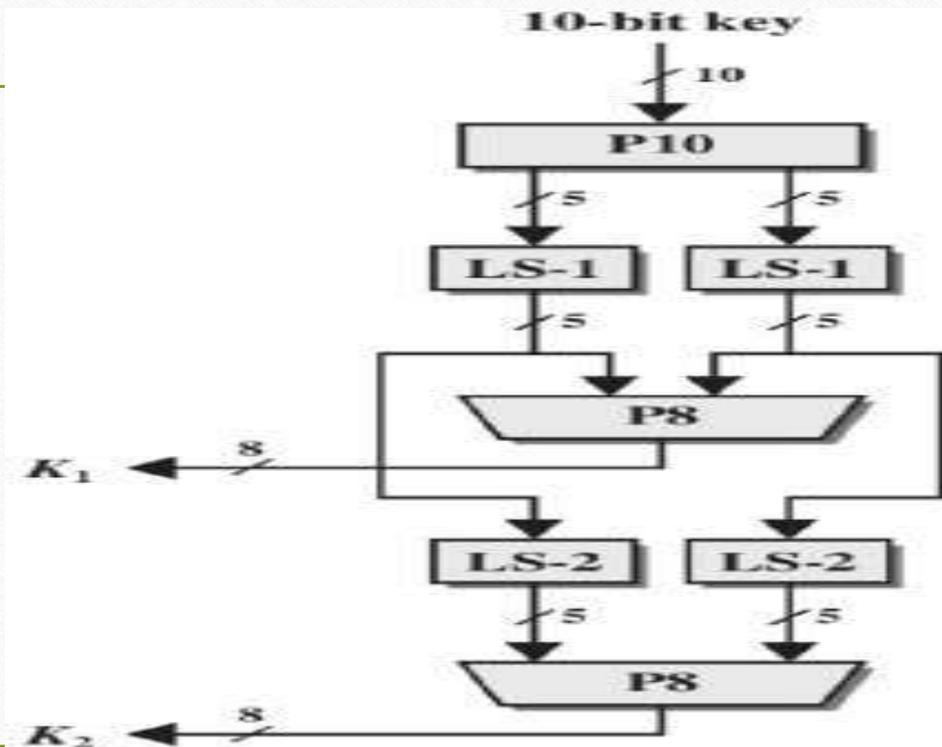
---

- Input (plaintext) block: 8-bits
- Output (ciphertext) block: 8-bits
- Key: 10-bits
- Rounds: 2
- Round keys generated using permutations and left shifts
- Encryption: initial permutation, round function, switch halves
- Decryption: Same as encryption, except round keys used in opposite order

# S-DES Algorithm

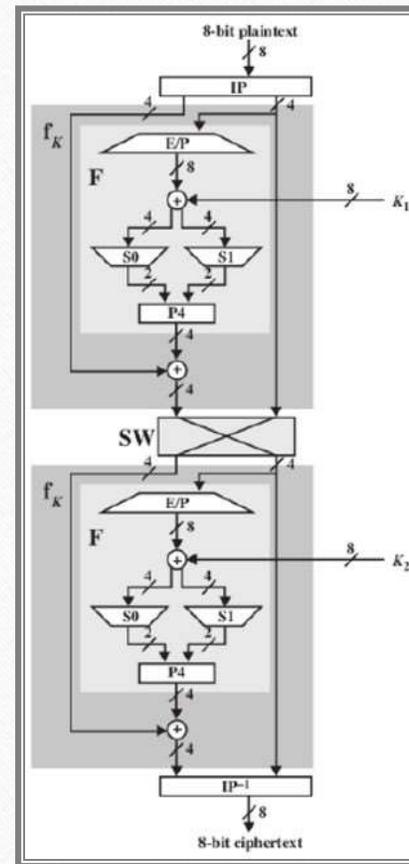


## S-DES Round Keys Generation

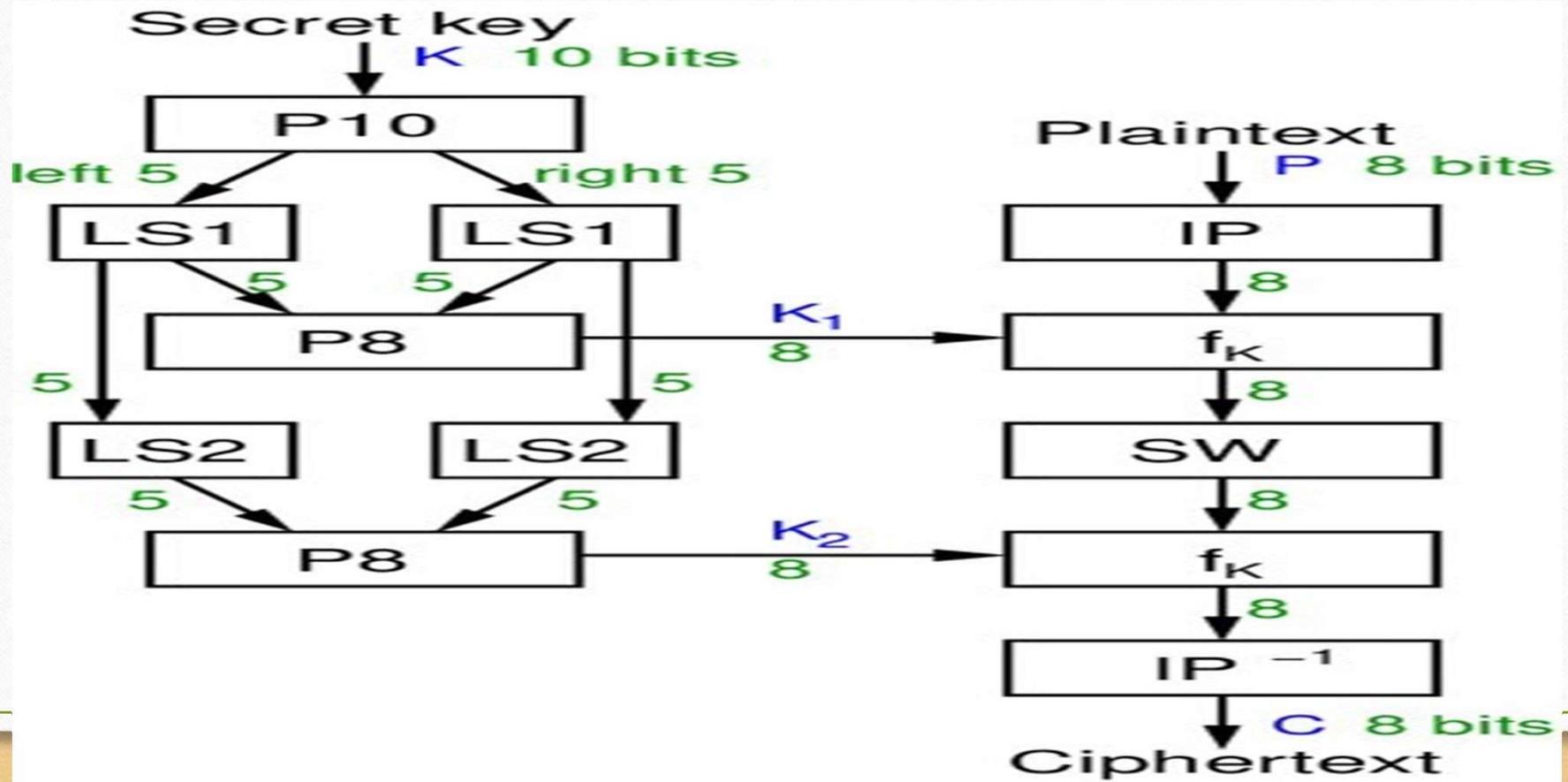


# S-DES

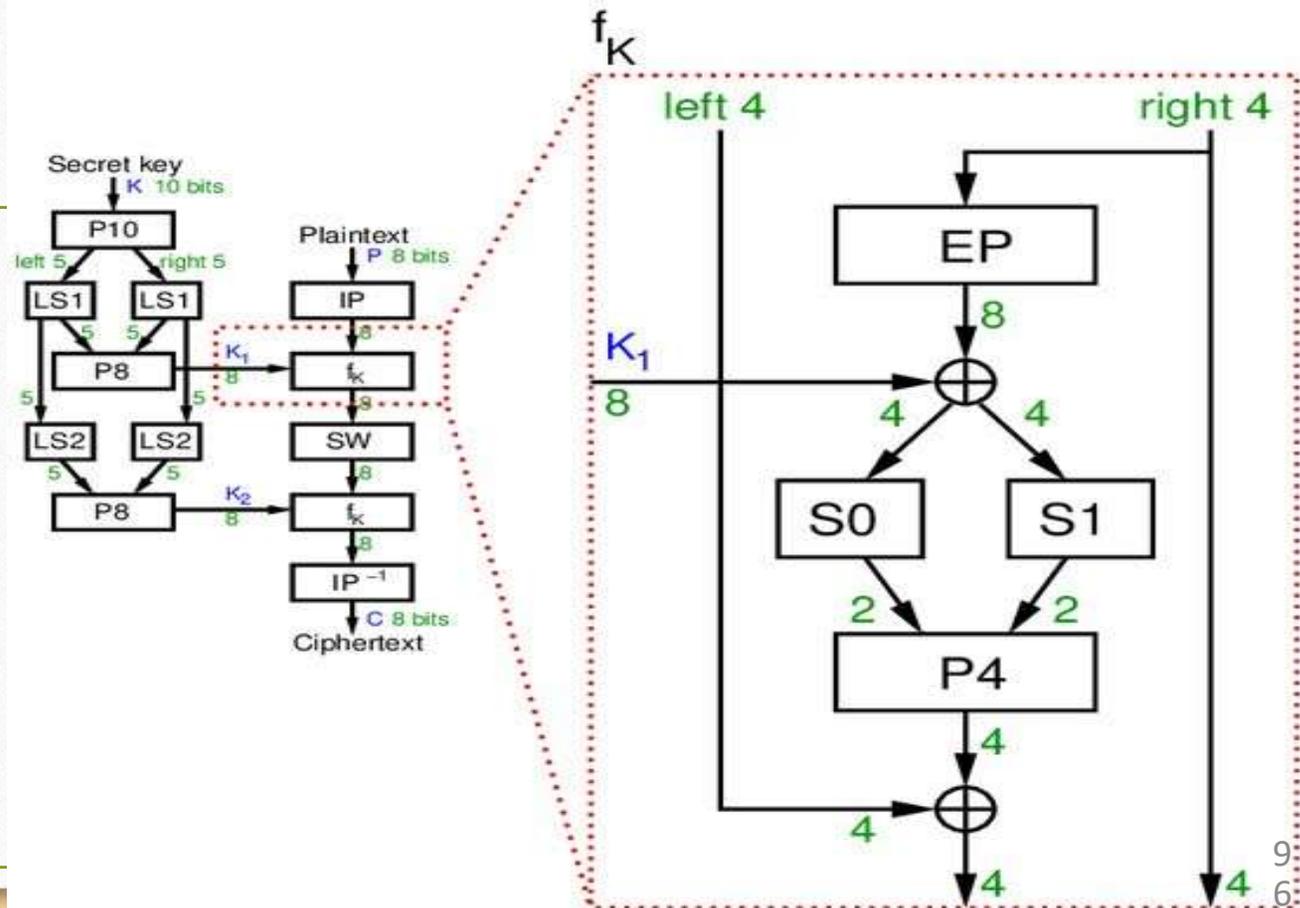
•94



# S-DES Key Generation and Encryption



# S-DES Round Function



# S-DES Permutations

- Permutation means transposition or rearrangement of bits.

## ➤ P10 (permutation)

Input	1	2	3	4	5	6	7	8	9	10
Output	3	5	2	7	4	10	1	9	8	6

## ➤ P8 (selection and permutation)

Input	1	2	3	4	5	6	7	8	9	10
Output	6	3	7	4	8	5	10	9		

## ➤ P4 (permutation)

Input	1	2	3	4
Output	2	4	3	1

# S-DES Operations

## ➤ EP (Expansion and Permutation)

Input	1	2	3	4				
Output	4	1	2	3	2	3	4	1

## ➤ IP (Initial Permutation)

Input	1	2	3	4	5	6	7	8
Output	2	6	3	1	4	8	5	7

## ➤ $IP^{-1}$ (Inverse of Initial Permutation)

Input	1	2	3	4	5	6	7	8
Output	4	1	3	5	7	2	8	6

# S-DES Operations

---

- LS-1: left shift by 1 position
- LS-2: left shift by 2 positions
- $IP^{-1}$ : inverse of IP, such that  $X = IP^{-1} (IP(X))$
- SW: swap the halves (Switching Function)
- $f_K$ : round function using round key K
- F: internal function in each round

# XOR Table

---

- If the bits are similar, the output is 0
- If the bits are different, the output is 1

A	B	A XOR B
0	0	0
0	1	1
1	0	1
1	1	0

# XOR Example

---

10100010

00101101

---

10001111



## S-Boxes of S-DES

---

- S-Box considered as a matrix: input used to select row/column; selected element is output
- 4-bit input:  $\text{bit}_1, \text{bit}_2, \text{bit}_3, \text{bit}_4$
- $\text{bit}_1\text{bit}_4$  specifies row (0, 1, 2 or 3 in decimal)
- $\text{bit}_2\text{bit}_3$  specifies column
- 2-bit output
- Indexing of S-Boxes starts from 0 to 3 for rows and columns.

## S-Boxes of S-DES

---

$$S_0 = \begin{bmatrix} 01 & 00 & 11 & 10 \\ 11 & 10 & 01 & 00 \\ 00 & 10 & 01 & 11 \\ 11 & 01 & 11 & 10 \end{bmatrix} \quad S_1 = \begin{bmatrix} 00 & 01 & 10 & 11 \\ 10 & 00 & 01 & 11 \\ 11 & 00 & 01 & 00 \\ 10 & 01 & 00 & 11 \end{bmatrix}$$

## S-Boxes of S-DES

---

$$S_0 = \begin{bmatrix} 1 & 0 & 3 & 2 \\ 3 & 2 & 1 & 0 \\ 0 & 2 & 1 & 3 \\ 3 & 1 & 3 & 2 \end{bmatrix} \quad S_1 = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 2 & 0 & 1 & 3 \\ 3 & 0 & 1 & 0 \\ 2 & 1 & 0 & 3 \end{bmatrix}$$

## S-DES vs. DES

---

	S-DES	DES
Block size	8 bits	64 bits
Key size	10 bits	56 bits
Rounds	2	16
IP	8 bits	64 bits
S-Boxes	2	8
Round keys	2	16
Round key size	8 bits	48 bits

# DES Example

---

Deploying S-DES cipher, encrypt the plaintext •  
(01110010) using the key (1010000010).

# S-DES summary

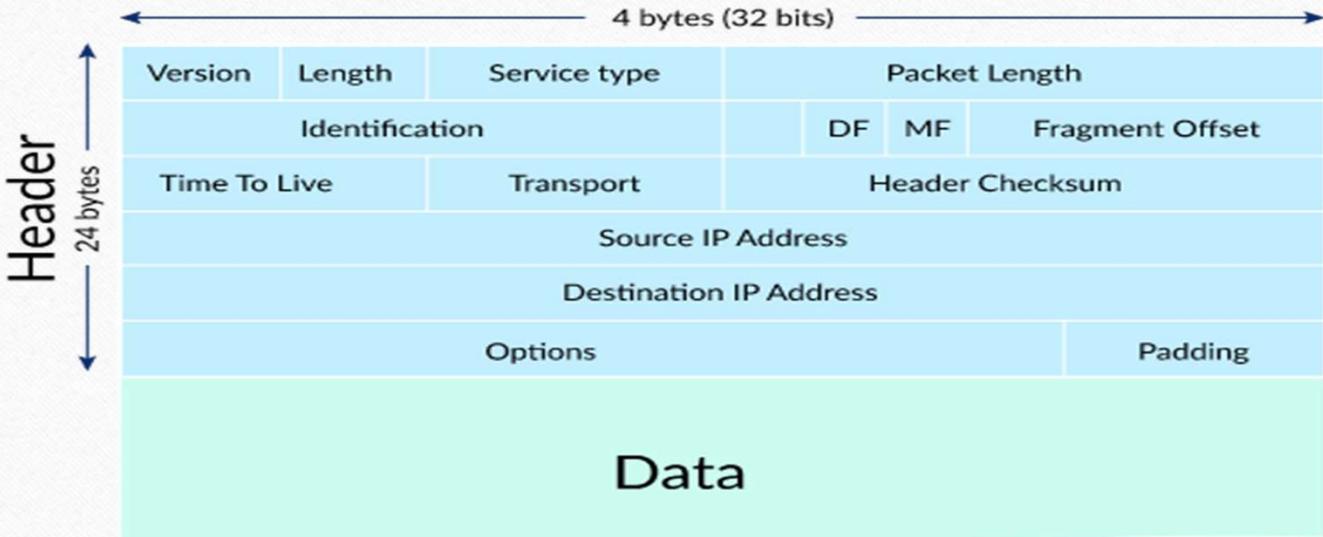
---

- Educational encryption algorithm
- Brute force attack on S-DES is easy since only 10-bit key
- If we know plaintext and corresponding ciphertext, can we determine key? Very hard

# Network Attacks



# IP PACKET



# Network Traffic Basics

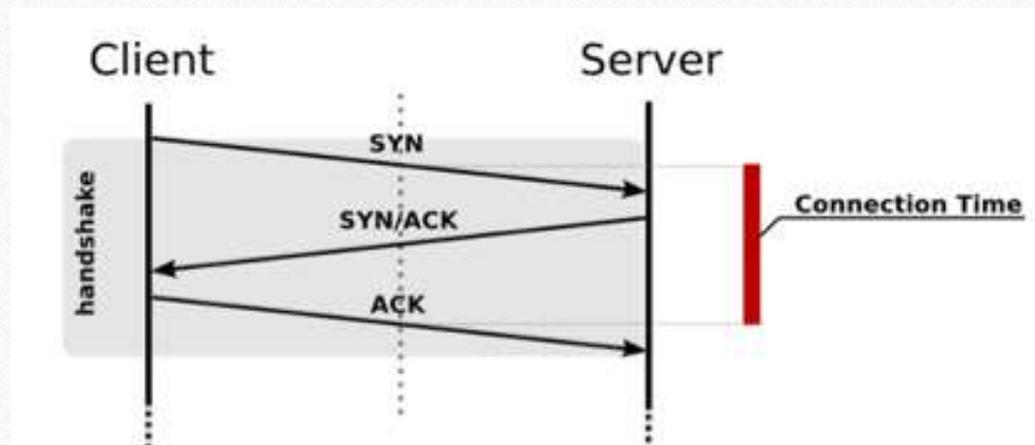
---

- The Internet Protocol (IP) and the Transmission Control Protocol (TCP) are the most commonly used protocols in network attacks.
- The IP protocol defines the rules for getting a packet from one point to another and the
- TCP protocol defines the rules ensuring that the data received at the destination is accurate and in the correct sequence.



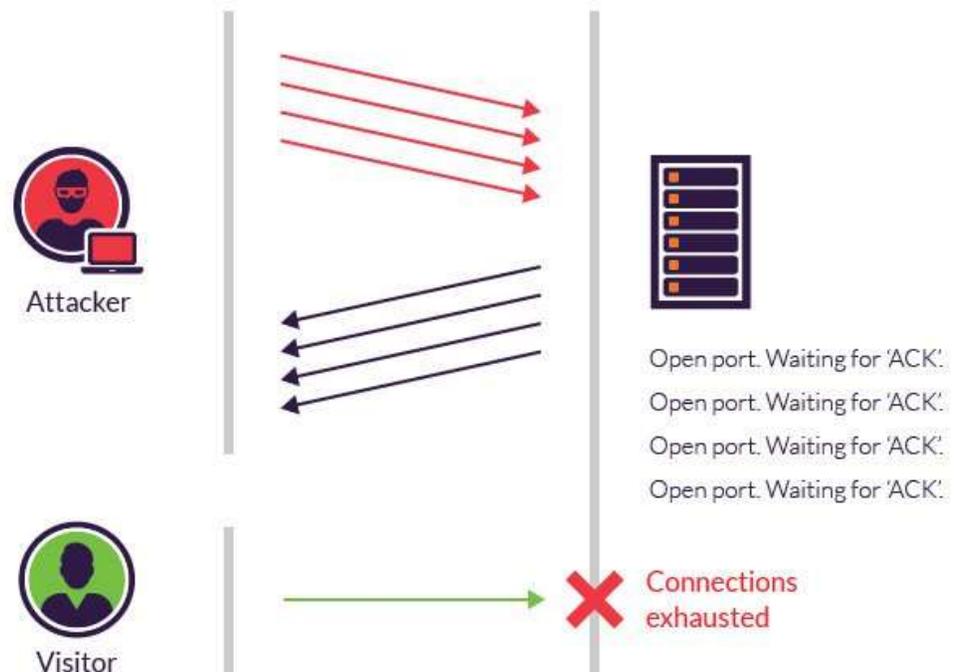
- To achieve these capabilities, both the TCP and IP protocols attach headers to the data given by the application,

# TCP Protocol



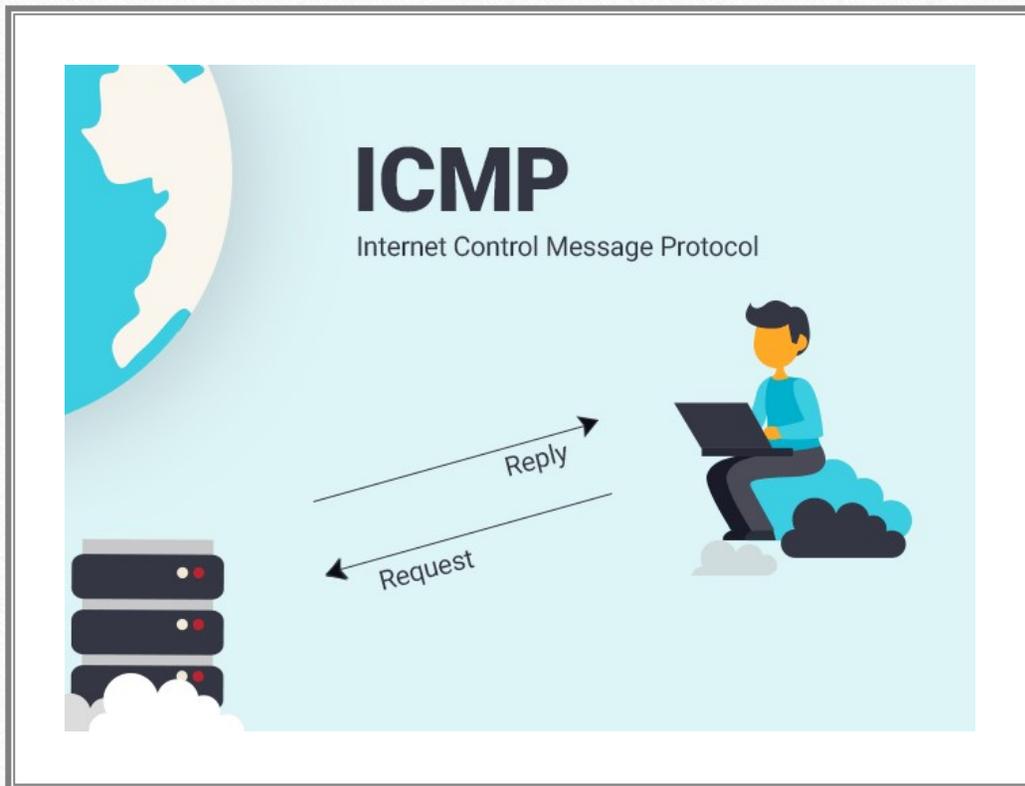
# SYN flood

- A classic DOS attack was the SYN flood
  - The attacker computer sends a stream of TCP SYN messages to the victim's computer.
  - The victim computer responds to all of the SYN messages, starting up a connection for each one.
  - The attacker does not respond to the victim's ACK/SYN messages with ACKs.
  - The overhead from maintaining all of these open connections slows down the victim computer, disabling it or perhaps even causing it to crash.



---

# The Ping of Death



# ICMP

- The *Internet Control Message Protocol (ICMP)* allows routers to send error and control messages to other computers, especially routers, on the network.
- ICMP operates at the network (routing) layer of the TCP/IP stack.

# Ping

---

- The most widely used ICMP message is the *ping*.
- Basically, ping is used to see if packets are reaching a particular computer.
- The client sends a ping request, and when it receives it, the server responds with a reply.
- A ping is normally 32 bytes in size.

# Ping

---

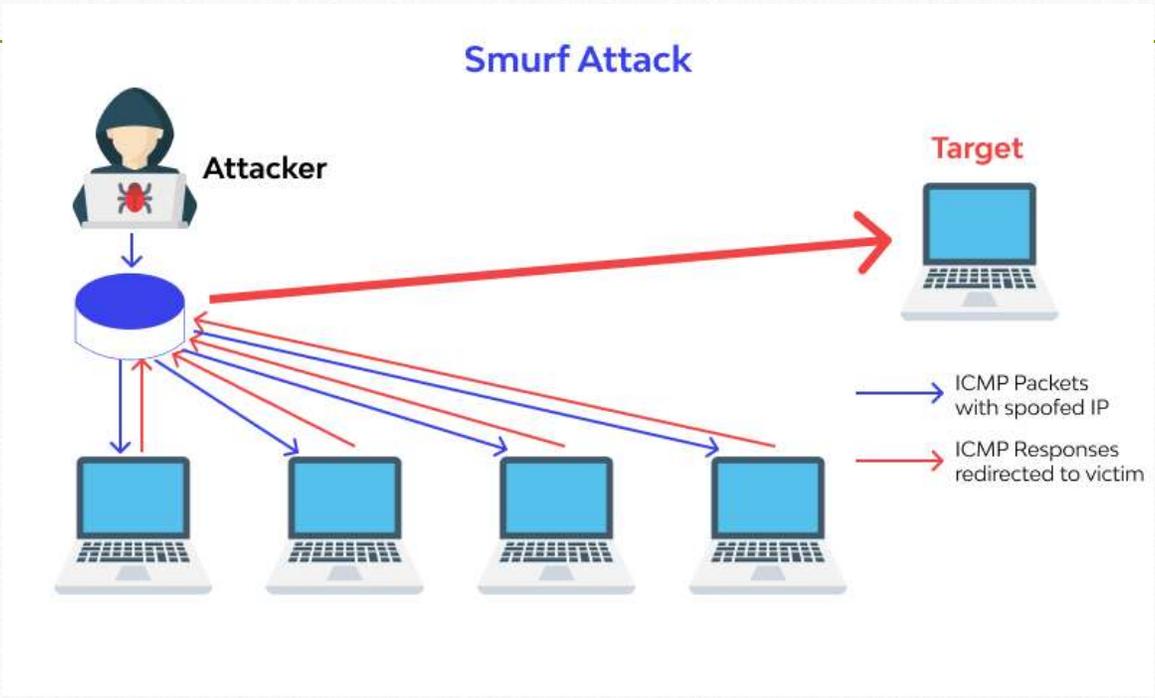
- Maximum IPv4 packet size is 65,535 bytes.
- Ping of death attack indicates sending 65,536 bytes or more.
- A ping packet of this size is illegal to IP protocol.
- So: if we send a packet that is more than 65535 bytes it will cause B.O. which causes the system to crash

# Ping

---

- The ping of death uses the ICMP ping to DOS a computer by crashing it.
- It does this by sending an illegally large ping packet.
  - In this case, more than 65,536 bytes.
- The packet causes a buffer overflow that crashes the computer.
- اذا سبب الانهيار هو حجم البينغ الكبير

# The Smurf Attack



سبب الانهيار هو  
عدد الـ ping ريكويست  
الكبير

# Smurf

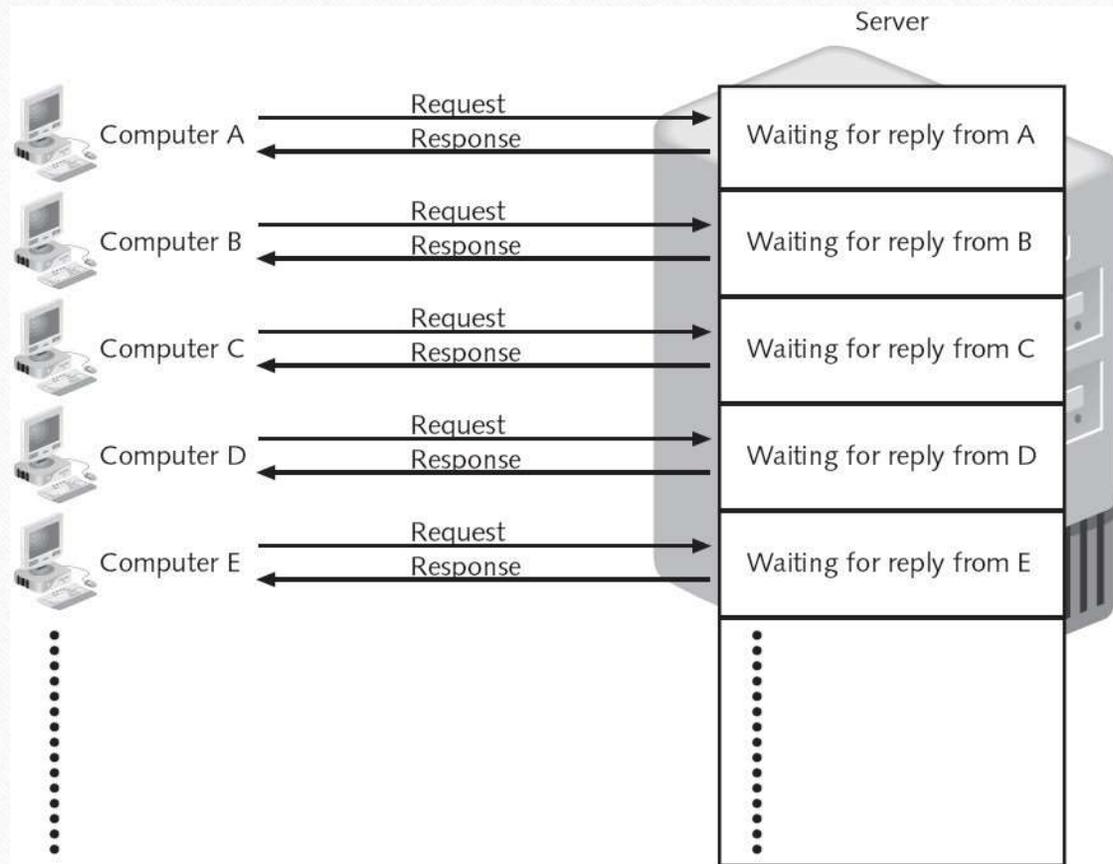
---

- So the smurf attack depends on sending a lot of Ping request then **Spoofing** the IP address to the victim's
- If enough computers (possibly thousands) receive the forged ping request, the reply packets can crash the victim computer,
- To prevent this kind of attack is
  - Computers do not reply to broadcast pings.
  - Block broadcast packets at the router.

# Denial of Service (DoS)

---

- Attempts to consume network resources so that the network or its devices cannot respond to legitimate requests
- Distributed denial of service (DDoS) attack
  - A variant of the DoS
  - May use hundreds or thousands of **zombie** computers in a botnet to flood a device with requests

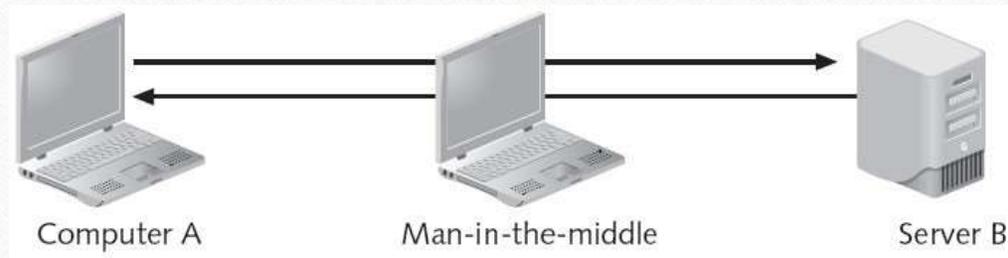


**Figure 4-4** DoS attack

# Man-in-the-Middle Attack

---

- Passive--attacker reads traffic
- Active--attacker changes traffic
- Common on networks



# Replay Attack

---

- Attacker captures data
- Resends the same data later
  - A simple attack: capture passwords and save them
  
- Note: same MITM but resend it later

# Sidejacking

---

- This attack based on stealing the cookies
- Almost all social networking sites are vulnerable to this attack
  - Facebook, MySpace, Yahoo, etc.

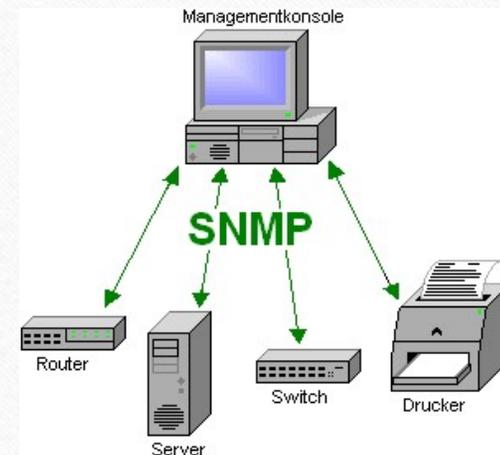


# بعض المصطلحات في الشبكات

---

# SNMP (Simple Network Management Protocol)

- Used to manage switches, routers, and other network devices
- Early versions did not encrypt passwords, and had other security flaws
- But the old versions are still commonly used



# DNS (Domain Name System)

- DNS is used to resolve domain names like [www.ccsf.edu](http://www.ccsf.edu) to IP addresses like 147.144.1.254
- DNS has many vulnerabilities
  - It was never designed to be secure

Name → IP address



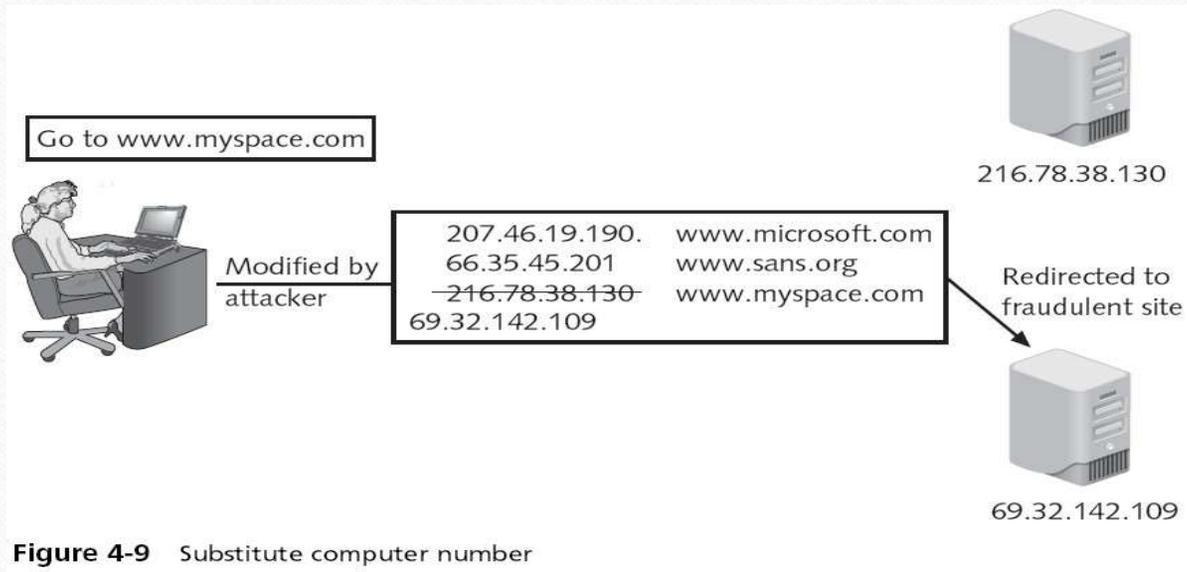
Where is [www.ccsf.edu](http://www.ccsf.edu)?



[www.ccsf.edu](http://www.ccsf.edu) is at 147.144.1.254

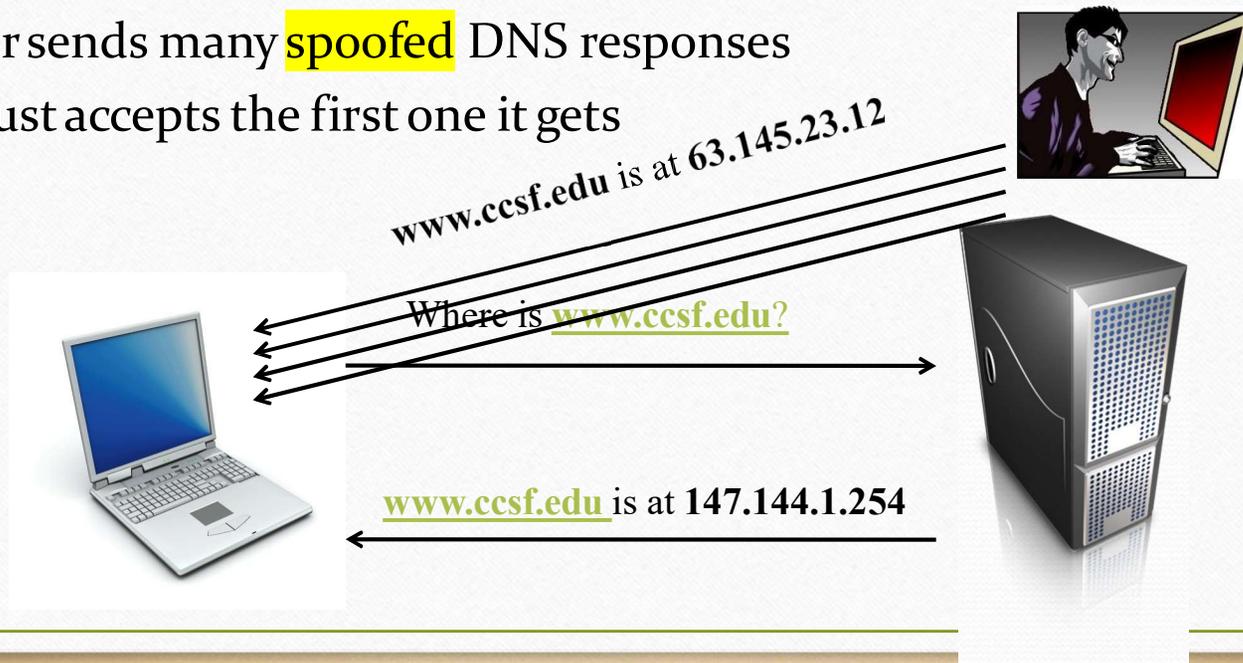


# DNS Poisoning



# DNS Cache Poisoning

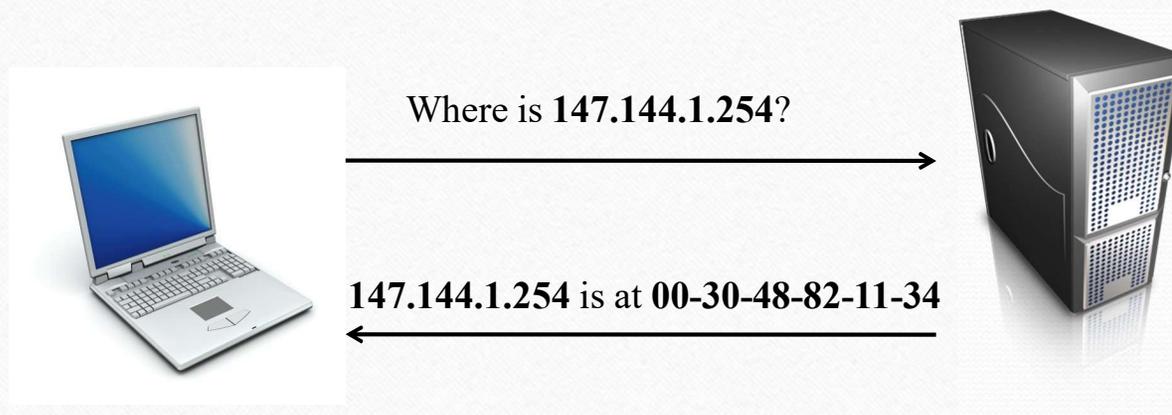
- Attacker sends many **spoofed** DNS responses
- Target just accepts the first one it gets



# ARP (Address Resolution Protocol)

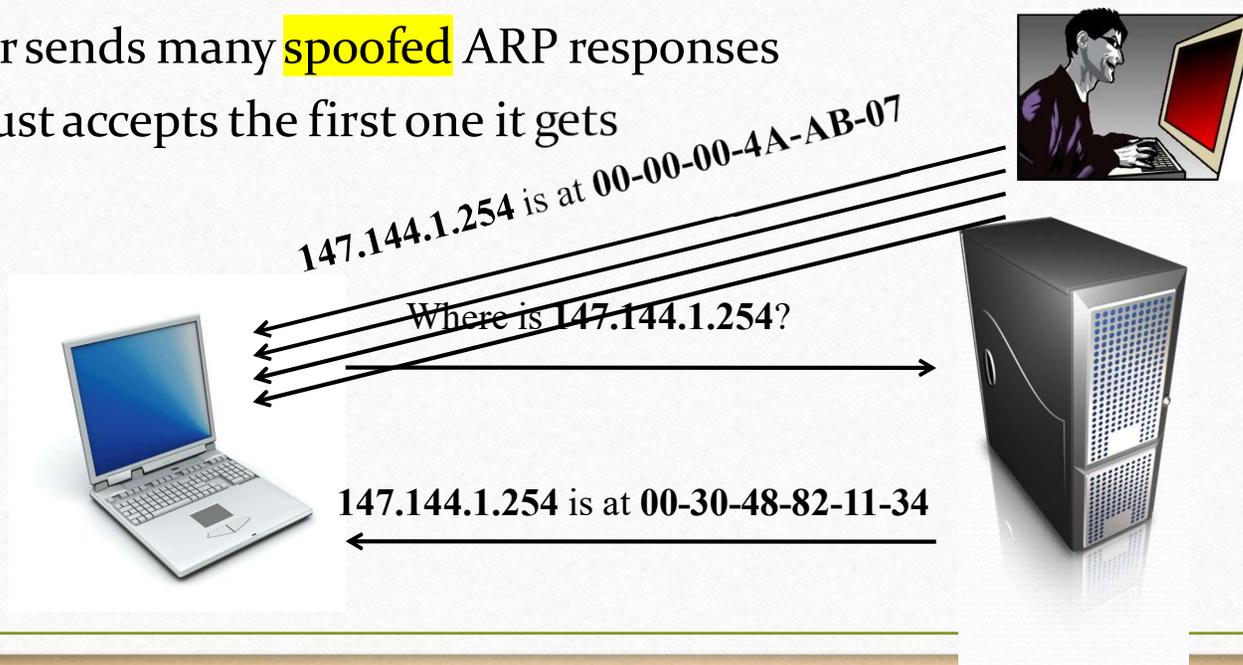
---

- ARP is used to convert **IP addresses** like 147.144.1.254 into **MAC addresses** like 00-30-48-82-11-34



# ARP Cache Poisoning

- Attacker sends many **spoofed** ARP responses
- Target just accepts the first one it gets



# Results of ARP Poisoning Attacks

---

- Steal of data
- MITM
- Prevent user to Access Internet

# Firewall

# What is a Firewall?

---

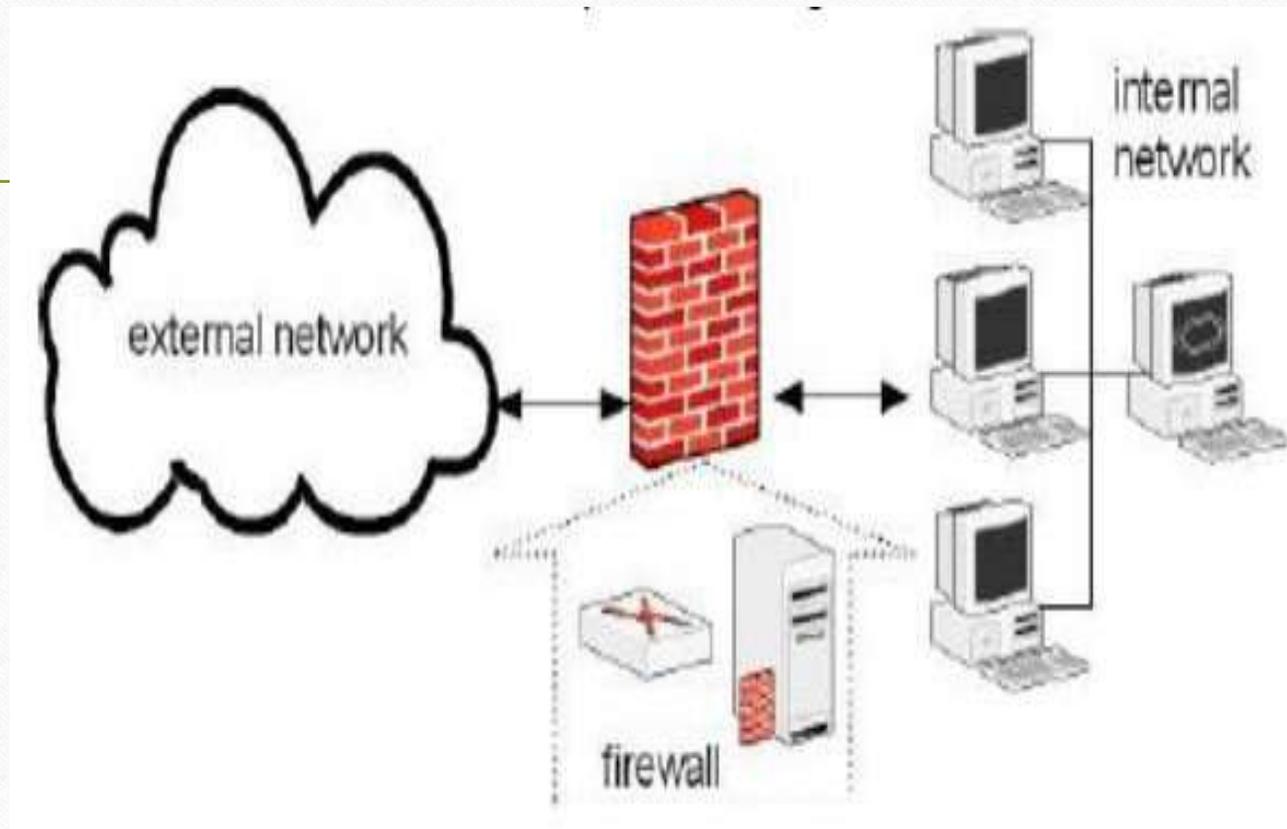
- A firewall is a system that enforces access policy between two (or more) networks.
- A firewall makes the decision on what to do with connection packets based on rules/policies.
  - Allow
  - Reject
  - Reject and inform

# Firewall Mechanism

---

- Two main approaches to setup a firewall:
  - Block all that is not explicitly authorized.
    - Allow all that is not specifically blocked.
- Firewall Mechanism:
  - Firewall examines all traffic packets between the networks.
  - Packets are evaluated against a list of “rules/policies” and conditions.

# Firewall Architecture



# Firewall Architecture

---

- The previous example shows a firewall architecture made of two blocks. The external network (left side) and the internal network composed of four computers (right side) are two entities separated physically by a firewall, whose goal is filter the inbound traffic and outbound traffic.
- **Inbound** traffic: the traffic that comes from the external network to the internal network.
- **Outbound** traffic: the traffic that goes from the internal network towards the external network.

# Default Firewall Policies

---

- Default to block all
  - Most secure implementation
- Default to allow all
  - Least secure implementation
  - Can you really trust it???

# Types of Firewalls

---

- Stateless Packet Filtering – Network Layer
- Stateful Packet Filtering – Network Layer
- Circuit Proxy – Transport Layer
- Application Proxy – Application Layer

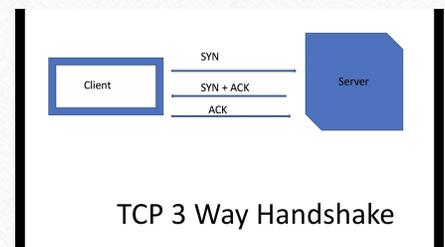
# Stateless Packet & stateful packet

- **Stateless** :Control the forwarding or dropping of the **data based on the IP header information**, not the payloads.

Example IP destination address, IP source address

**Stateful** : controls also by the IP header but it **also keeps track** on the TCP Connection

In other words: the stateful packet filter will keep track of all conversations and **ensure that all packets transiting comply with proper protocol rules and operations.**

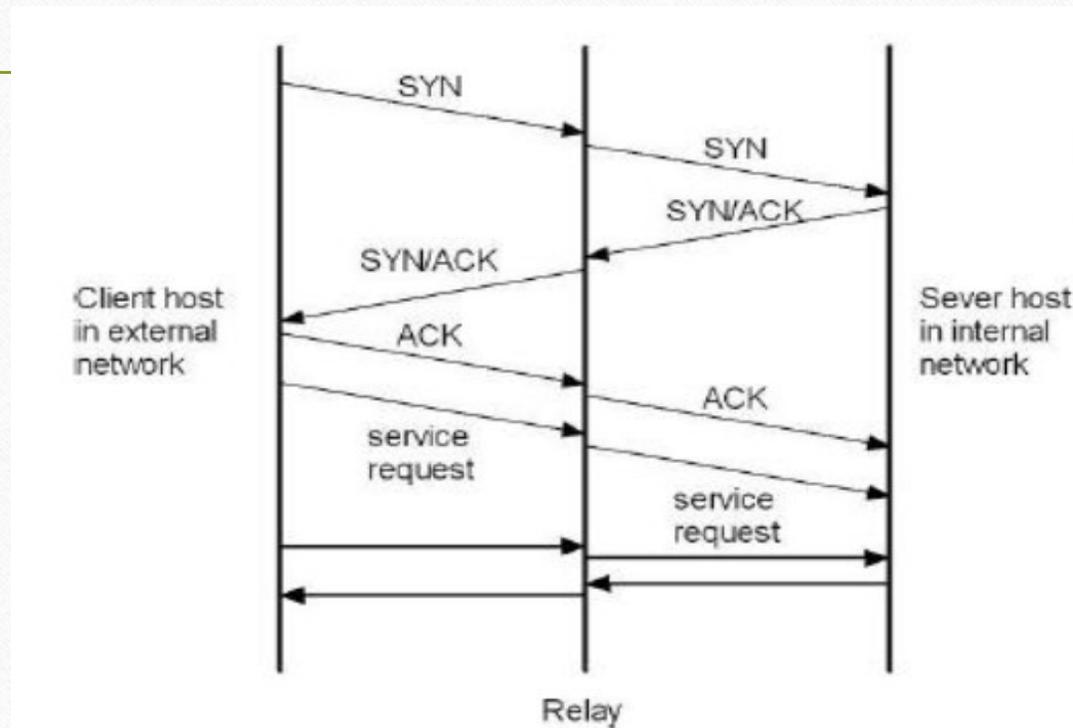


# Circuit Proxy

---

- Circuit proxy firewall acts at layer 4 (Transport Layer).
- They act as **intermediate** that relay a TCP connection between an internal and external host.
- They **disallow the direct connection** between the external and the internal networks.

# Circuit Proxy



# Application Proxy

---

- The same as Circuit Proxy but acts on the application layer

# Demilitarized Zones (DMZ)

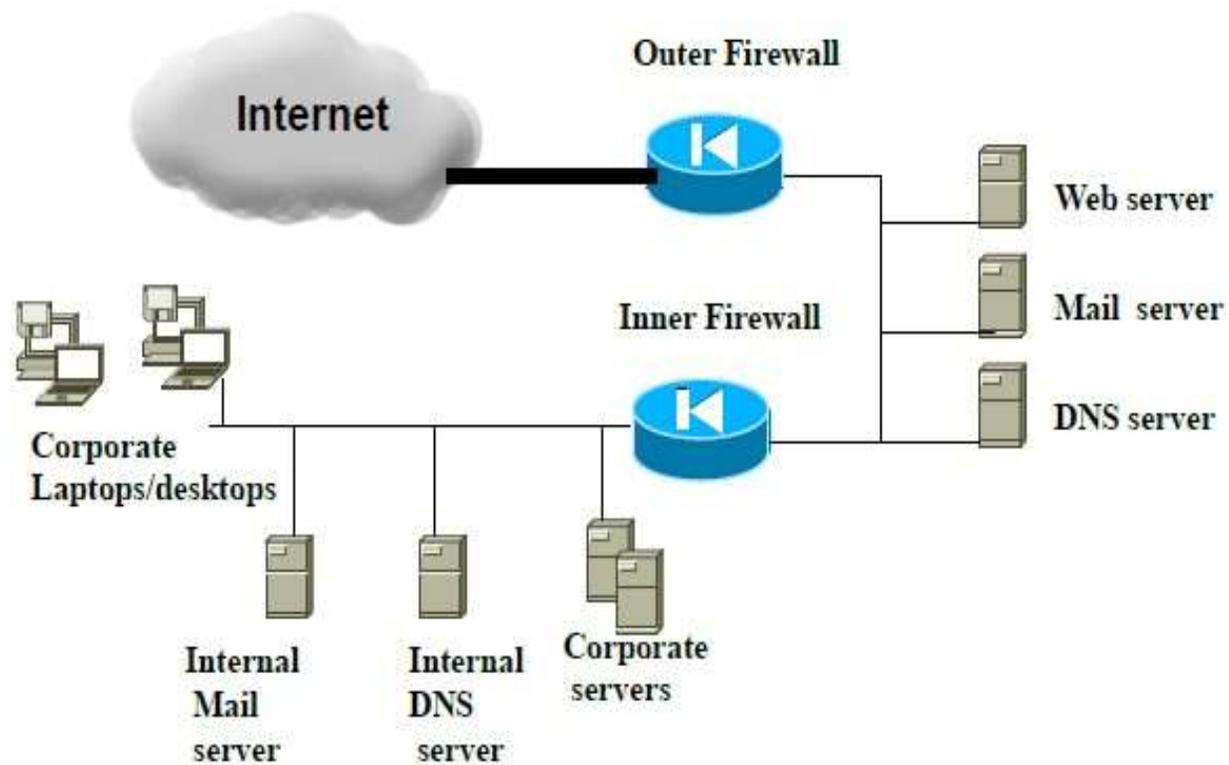
---

DMZ is a subnet between two firewalls in an internal network

- External firewall protects DMZ from external threats
- Internal firewall protects internal network from DMZ
- The role of the DMZ is to provide **strong separation** between the external and internal networks.

# DMZ

## Example 1



# Benefits of Firewall

---

- **Control access** based on sender or receiver addresses.
- **Hiding the internal network** (e.g., addresses, traffic, etc.
- **Reduce attacks by hackers.**

# Intrusion Detection System

---

IDS

# Introduction

---

- it is very important to have additional protection mechanisms on the internal hosts and network.
- Intrusion detection systems fulfill such purpose by monitoring computing systems and reporting intrusive behaviors.

# Intrusion

---

- **Intrusion**: attempt to compromise the confidentiality, integrity, availability, or to bypass the security mechanisms of a system. (attempt to breaking into a system).
- Intrusions have many causes:
  - ❖ **Malware** (viruses, worms, trojan horses, etc...).
  - ❖ Attackers gaining unauthorized access.
  - ❖ Authorized users who misuse their privileges.
  - ❖ Authorized users who attempt to gain additional privileges.
- Although many intrusions are malicious in nature, many others are not; for **example**: a person might mistype the address of a computer and accidentally attempt to connect to a different system without authorization.

# Intrusion Detection

---

- Intrusion detection: is the process of **monitoring** the network traffic in order to identify unauthorized activities. (كأنه ماسح ضوئي)
- Intrusion detection system (IDS): is a system that automates the intrusion detection process. The primary responsibility of an IDS is to detect unwanted and malicious activities.
- Intrusion prevention system (IPS): is a system that has all the capabilities of an intrusion detection system, **in addition to the ability of stopping possible incidents.**
- Intruders may be from outside the network or legitimate users of the network.

## Why IDS should be used?

---

- Identifying incidents, logging information about them, attempting to stop them, and reporting them to security administrators.
- Identifying problems with security policies
- Documenting existing threats
- Deterring individuals from violating security policies.

# IDS Models

---

- Anomaly detection

Based on **behavioral** after analyzes a set of characteristics of the system

- Misuse detection

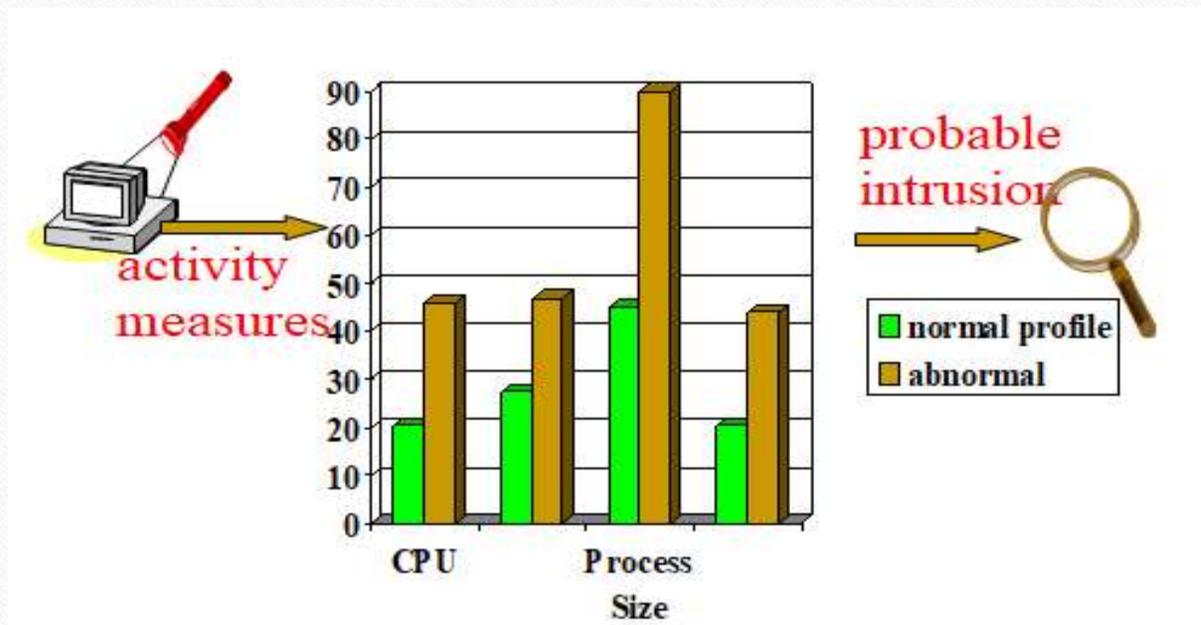
Also known as signature-based detection or Pattern Matching—Matches pattern of malicious activity. (**Based on Signature**)

- Specification-based detection

Examines the protocol and/or payload content to determine the validity of the specifications.  
(**Based on protocol and payload**)

# Anomaly Detection IDS

- Relatively high false positive rate - anomalies can just be new normal activities.



# Anomaly Detection IDS

---

- This type of IDS models the normal usage of a network as a genuine behavior.
- Anything distinct from the genuine behavior is assumed to be an intrusion activity.
  - For instance, flooding a host with lots of packet.
- The primary strength is its ability to recognize novel (zero-day) attacks.

## Example: Network Anomalies

- Normal traffic flowing to 142.104.112.106, the protected system

14345	03/23/2004	11:55:10	00:00:03	ftp	33291	20	<u>142.104.112.115</u>	142.104.112.106
14346	03/23/2004	11:55:13	00:00:04	smtp	32267	25	142.104.112.113	142.104.112.106
14347	03/23/2004	11:55:17	00:00:05	ftp	33547	20	<u>142.104.113.10</u>	142.104.112.106
14348	03/23/2004	11:55:22	00:00:03	http	32523	80	<u>142.104.115.21</u>	142.104.112.106
14349	03/23/2004	11:55:25	00:00:01	http	33035	20	<u>142.104.124.110</u>	142.104.112.106
14350	03/23/2004	11:55:26	00:00:02	http	32779	20	142.104.124.39	142.104.112.106
14351	03/23/2004	11:55:27	00:00:01	http	32011	20	142.104.124.78	142.104.112.106
14352	03/23/2004	11:55:27	00:00:03	http	34315	20	142.104.124.71	142.104.112.106
14353	03/23/2004	11:55:30	00:00:07	http	36107	20	142.104.124.80	142.104.112.106
14354	03/23/2004	11:55:37	00:00:01	ftp	35851	20	142.104.112.115	142.104.112.106

- Anomalous traffic flowing to 142.104.112.106

14355	03/23/2004	11:55:38	00:00:01	http	26891	80	<u>009.009.009.009</u>	142.104.112.106
14356	03/23/2004	11:55:38	00:00:01	http	26635	80	<u>009.009.009.009</u>	142.104.112.106
14357	03/23/2004	11:55:38	00:00:01	http	27403	80	<u>009.009.009.009</u>	142.104.112.106
14358	03/23/2004	11:55:38	00:00:01	http	26427	80	<u>009.009.009.009</u>	142.104.112.106
14359	03/23/2004	11:55:38	00:00:01	http	27659	80	<u>009.009.009.009</u>	142.104.112.106
14360	03/23/2004	11:55:38	00:00:01	http	27147	80	<u>009.009.009.009</u>	142.104.112.106
14361	03/23/2004	11:55:38	00:00:01	http	27915	80	009.009.009.009	142.104.112.106
14362	03/23/2004	11:55:38	00:00:01	http	28171	80	009.009.009.009	142.104.112.106
14364	03/23/2004	11:55:39	00:00:01	http	28959	80	009.009.009.009	142.104.112.106
14365	03/23/2004	11:55:39	00:00:01	http	31499	80	009.009.009.009	142.104.112.106
14366	03/23/2004	11:55:39	00:00:01	http	30319	80	009.009.009.009	142.104.112.106
14367	03/23/2004	11:55:39	00:00:01	http	29963	80	009.009.009.009	142.104.112.106
14368	03/23/2004	11:55:39	00:00:01	http	30475	80	009.009.009.009	142.104.112.106
14369	03/23/2004	11:55:39	00:00:01	http	29195	80	009.009.009.009	142.104.112.106
14370	03/23/2004	11:55:39	00:00:01	http	29451	80	009.009.009.009	142.104.112.106
14371	03/23/2004	11:55:39	00:00:01	http	30731	80	009.009.009.009	142.104.112.106
14372	03/23/2004	11:55:39	00:00:01	http	29707	80	009.009.009.009	142.104.112.106
14373	03/23/2004	11:55:39	00:00:01	http	28683	80	009.009.009.009	142.104.112.106
14374	03/23/2004	11:55:39	00:00:01	http	31243	80	009.009.009.009	142.104.112.106
14375	03/23/2004	11:55:39	00:00:01	http	30987	80	009.009.009.009	142.104.112.106

# Drawbacks of Anomaly Detection IDS

*else*

---

- Relatively high false positive rate.
- Anomalies can just be new normal activities.

# Misuse Detection IDS

---

- Misuse detection IDSs rely on pattern matching algorithms. For example, an IDS that watches web servers might be programmed to look for the string “phf” in (“GET /cgi-bin/phf?”), as an indicator of a CGI program attack.
- Can't detect new attacks

# Drawbacks of Misuse Detection IDS

---

- They are unable to detect novel attacks (zero-day attacks).
- Have to programmed again for every new pattern to be detected.

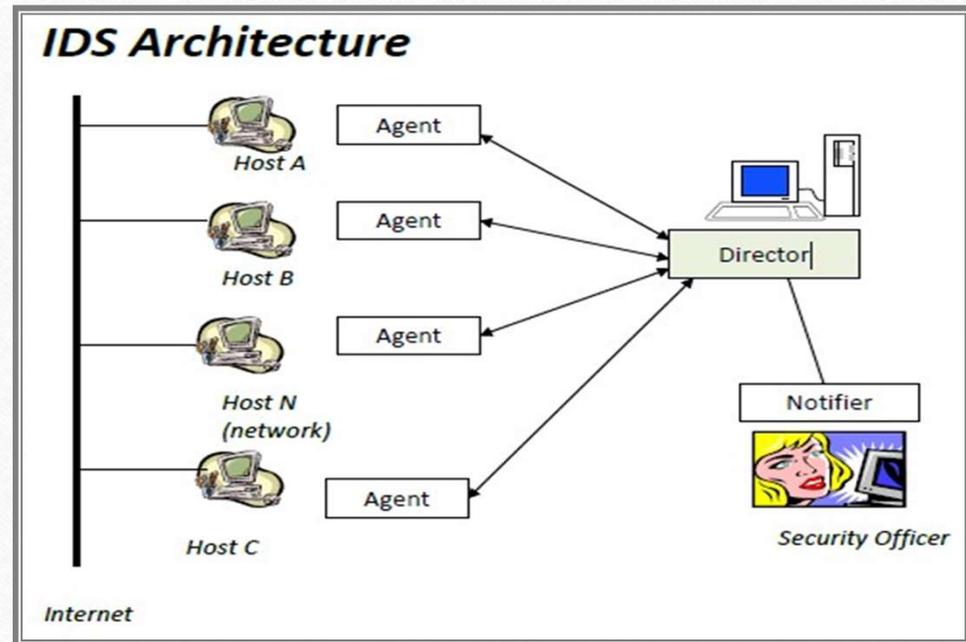
## IDS Architecture

- Basic architecture of an intrusion detection system involves 3 components: *Agent*, *Director*, and *Notifier*.

**Agent:** جمع المعلومات وارسالها للدائير كتر

**Director:** ترتيب و تنظيم المعلومات و حذف المكرر و ارسالها للنوتيفايير

**Notifier:** تحديد اذا ما كان هنالك هجوم و اخبار صاحب الشأن



# Host & Network based IDS

---

- Host based on single computer
- Network based in multiple computers

# Honeypot

---

- Honeypots are decoy systems that designed to redirect a potential attacker away from critical systems.
- a honeypot is a system designed to teach how intruders probe for and exploit a system. By learning their tools and methods, you can then better protect your network and systems.

# Honeypots are Designed To

---

1. you can know how the attacker is thinking and the techniques he used , so you can improve the security of the network
2. you can know the details of the attacker like IP address so you can replay to him with an attack
3. you can distract the attacker of the main value information that you are hiding

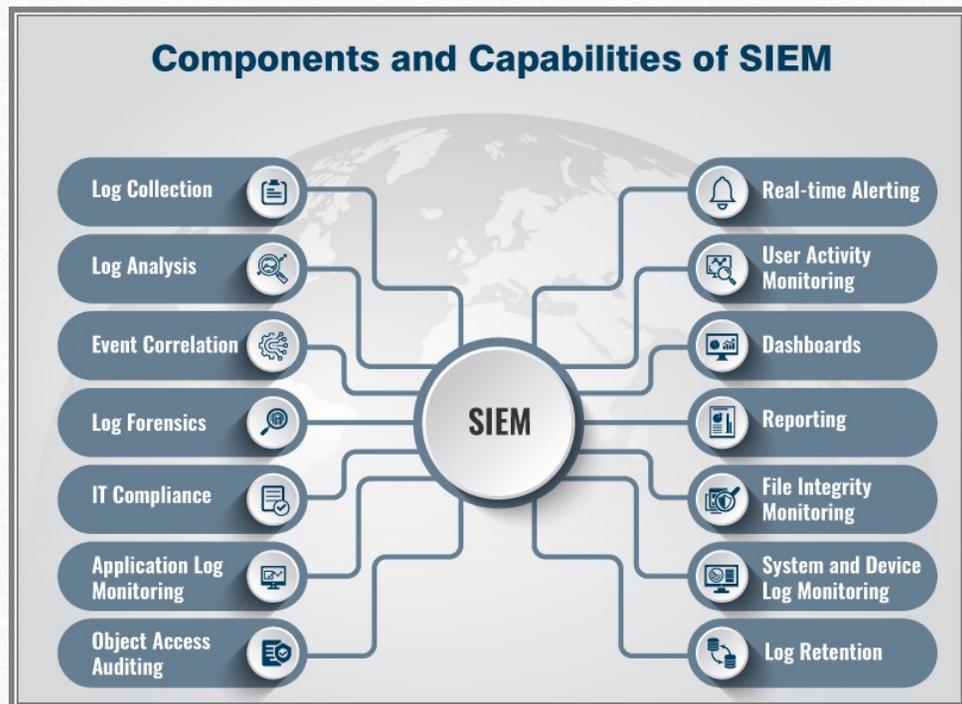
# The plan

---

- The simple plan is to build a box I wanted to learn about, put it on the network, and then wait.
  - How do I track the intruders moves?
  - How do I alert myself when the system is probed or compromised?
  - how do I stop the intruder from compromising other systems?
- The solution to this was simple, put the honeypot on its own network behind a firewall.

## Security Information Management (SIM)

- SIM provides a simple mechanism that allows security teams to collect and analyze vast amounts of security alert data.
  - More specifically, SIM solutions collect, analyze and correlate – in real-time – all security device information across an entire enterprise.
- Correlated results are then displayed on a centralized real-time console that is part of an intuitive graphical user interface.



# Application Security

---

# Malicious Software (Malware)

---

- Malware:

Malicious software programmed to damage other people's computer systems.

A malware could lead to:

- Gaining unauthorized access
- Revealing private information
- Modifying contents
- Denial of services    DOS,DDOS

# Malware

---

- There are three main reasons that facilitate the mechanism of malware installation and infection: (يضعف)
  - Software loopholes and flaws
  - Improper system configurations
  - Luring users to download malicious scripts

# Types of Malware

---

- ❖ Trojan Horses
- ❖ Viruses
- ❖ Worms
- ❖ Rootkits

# Trojan Horses

---

- *A Trojan horse is a program with an overt (documented or known) effect and a covert (undocumented or unexpected) effect.*
- In other words, Trojan horses are software programs that appear to do one particular thing, but secretly also do other malicious things.

# Viruses

---

- A virus is a malicious program that can *insert* a copy of itself into other files or programs, and then *performs some malicious actions*.
- A virus has two modes of operation:
  - **Insertion phase**: during which the virus inserts itself in a program.
  - **Execution phase**: during which the virus performs some malicious actions.

## Common categories of viruses

---

1. **Boot sector infector** : inserts itself into the **boot sector** of a disk
2. **Executable infector**: targets executable programs (e.g., .exe files).
3. **Multipartite virus**: affects both applications and boot sectors.
4. **TSR Virus**: remains active in memory even after operation. TSR stands for terminate and stay resident.
5. **Stealth virus**: conceals the infection of files to readers.
6. **Encrypted virus**: most of the virus code is encrypted.
7. **Polymorphic virus**: changes its form each time it infects another file.
8. **Macro virus**: is interpreted rather than executed directly.

# Worms

---

- A worm is self-replicating software designed to spread through the network, and it has the capability of propagation by copying itself from computer to computer.
- Typically, exploit security flaws in widely used services
- Can cause enormous damage:
  - Launch DDOS attacks.
  - Access sensitive information.
  - Cause confusion by corrupting the sensitive information.

# Worms

---

## ❖ Worm Structure:

A typical worm consists of:

- *Target locator subroutine*: used to find new targets
- *Infection propagator subroutine*: used to transfer the infection to a new computer

## ❖ Worm Types:

*Mass mailers* and *rabbits* are the two most common types of worms:

- *Mass mailers*: reproduce themselves to other computers through emails.
- *Rabbits*: can massively replicate to take over the entire memory, crashing the system.

## Example: Worm Pseudocode

Avoid multiple infection

Infection

Privilege escalation

Activation

Self replication

Manipulation

```
void main(){ // worm

    check_if_already_infected();
    if(already_infected){
        return;
    }

    infect(); // make sure of successive executions

    if(!admin_privileges){
        get_admin_privileges();
    }

    for(;;){
        block_until_some_condition();

        send_copies_of_me_over_internet();

        do_some_damage();
    }
}
```

## Example: The Melissa Worm

---

- ❖ Created in 1999 by David L. Smith
- ❖ First widely publicized worm targeted at Microsoft products.
- ❖ Replicate itself through emails:
  - Target Microsoft Outlook programs.
  - When the user opens an infected email attachment, the viral code will search 50 email addresses stored within Outlook and send an email to each of these addresses with a worm attachment.
- ❖ Email message template:

**From :** <the infected sender>

**Subject:** Important message from <the infected sender>

**To:** <The 50 chosen recipients>

**Attachment:** LIST.DOC

**Body:**

Here is that document you asked for... Don't show anyone else.

## Example: The Nimda Worm

---

- ❑ Released September 18, 2001.
- ❑ Multi-mode spreading:
  - Attack IIS servers via infected clients
  - هو خادم ويب من شركة مايكروسوفت
  - Email itself to address book
  - Copy itself across open network shares
  - Modifying web pages on infected servers



# Rootkits

---

- A rootkit is software designed to gain root-level privileges or administrator-level control over a computer system.
- Rootkits can evade normal security measures, by modifying the core components of an operating system, such as:
  - Modifying the kernel of an operating system.
  - Installing drivers to subvert security mechanisms.
- Rootkits are commonly used as a method for:
  - Hiding files from the operating system, such as hiding running processes services, registry keys, and open TCP/UDP ports.
  - Stealing sensitive information from the system.

## Types of Rootkits

---

- **Firmware rootkits** is rarely checked for integrity. Rootkits installed here can survive reboots and operating system reinstallations.
- **Hypervisor rootkits** modify the boot sequence of the target system and take advantage of virtualization aspects of modern CPUs. They load the original operating system as a virtual machine and are therefore able to intercept all hardware calls.
- **Bootloader rootkits** occur when an attacker can replace the original bootloader with another that he controls. These bootloaders are generally used to subvert full disk encryption solutions
- برنامج كمبيوتر مسؤول عن تمهيد الكمبيوتر. عند إيقاف تشغيل الكمبيوتر ، فإن برامجه - بما في ذلك أنظمة التشغيل ورمز التطبيق والبيانات - تظل مخزنة على الذاكرة.



## Types of Rootkits, Cont'd

---

- **Kernel mode rootkits** are the **most common type of rootkit**. They add additional code or replace portions of the operating system itself through the loading of device drivers or loadable kernel modules. This allows them to execute with the **same privileges as the operating system** and are therefore very hard to detect and remove.
- **Library rootkits** replace, patch, or hook system calls to hide attacker information.

# Malware Defense Practices

## ❖ Prevention Practices:

Block malware from getting into genuine systems using the following measures:

1. Install software patch in time. هي برنامج مصمم لإصلاح المشاكل أو لتحديث برنامج كمبيوتر أو البيانات الداعمة له. وهذا يشمل تحديد نقاط الضعف الأمنية والأخطاء الأخرى، وتحسين قابليتها للاستخدام أو أدائها.
2. Avoid downloading software from untrusted sites.
3. Avoid opening risky email attachments.

## ❖ Restoration استعادة Practices:

Disinfect infected systems using the following measures:

1. Scan files using a virus scanner; quarantine or remove infected files.
2. keep a backup of the system files and user files, which can be used to restore the system.

# Software Security Exploits

---

- Buffer Overflow (BO)
- Cross Site Scripting (XSS)
- SQL injection (SQLi)

# Buffer Overflow (BO)

---

- Buffer overflow is a flaw in a program that accepts an input value larger than the size of memory location in the buffer.
- Buffer overflow could exploit the internal memory structure of an operating system, which could lead to privilege escalation, access of computer's resources, and system crashing.
- Two main reasons that cause buffer overflow:
  - Ineffective or lacking of input validation.
  - Running programs with high privileges.

## Example: BO in C program

Suppose the following program which written in C, the program has two variables which are adjacent in memory: an 8-byte-long string buffer (A), and a two-byte integer (B).

```
char A[8] = " ";
```

```
unsigned short B = 1979;
```

Initially, variable (A) contains nothing but zero bytes, and variable (B) contains the number 1979.

variable name	A								B	
value	[null string]								1979	
hex value	00	00	00	00	00	00	00	00	07	BB

## Example: BO in C program, Cont'd

Now, the program attempts to store the “excessive” string terminated by null character (null-terminated string) in ASCII encoding in the (A) buffer, by using:

```
strcpy(A, "excessive");
```

“excessive” is 9 characters long and encodes to 10 bytes including the null terminator, but (A) can take only 8 bytes. By failing to check the length of the string, it also overwrites the value of (B):

variable name	A								B	
value	'e'	'x'	'c'	'e'	's'	's'	'i'	'v'	25856	
hex	65	78	63	65	73	73	69	76	65	00

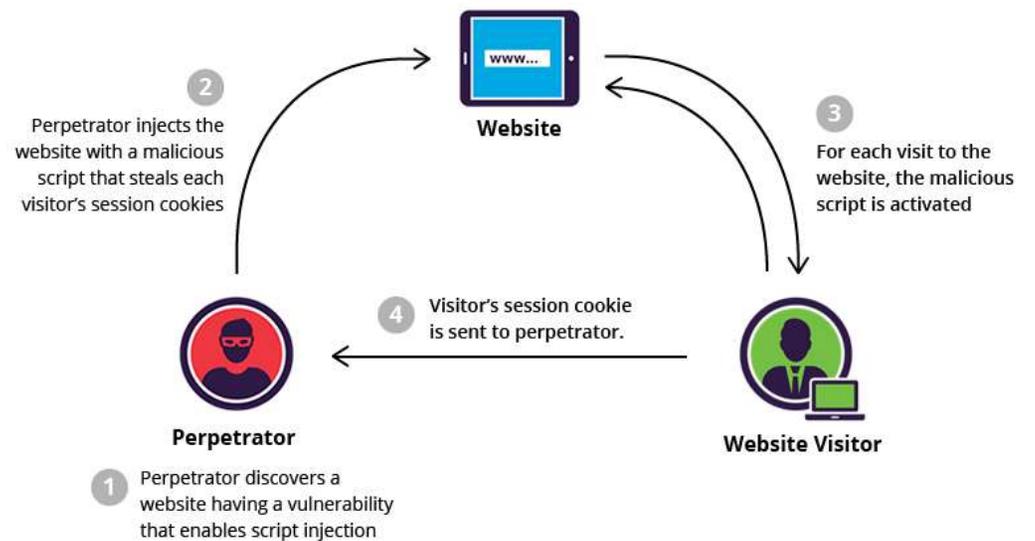
## Buffer overflow countermeasures

---

- Deploying input validation mechanisms
- Running applications with the **least privileges**.
- Patching and updating applications.

# Cross Site Scripting (XSS)

- In this attack, the attacker insert malicious script, usually JavaScript code or HTML tag to a web server (i.e. website), and when a user sends a request to this website and receives the response, the hidden malicious script of the attacker can be executed on the web browser of the client and do many malicious actions, such as sending session cookies and credential information, in addition to privilege escalation.
- XSS is a flaw in web applications, due to improper sanitization of user input in the output that it generates.



## Example of XSS, Cont'd

---

1. The attacker injects a payload into the website's database by submitting a vulnerable form with malicious JavaScript content.
2. The victim requests the web page from the web server.
3. The web server serves the victim's browser the page with attacker's payload as part of the HTML body.
4. The victim's browser executes the malicious script contained in the HTML body. In this case, it sends the victim's cookie to the attacker's server.
5. The attacker now simply extracts the victim's cookie, and can use the victim's stolen cookie for impersonation.

## XSS countermeasures

---

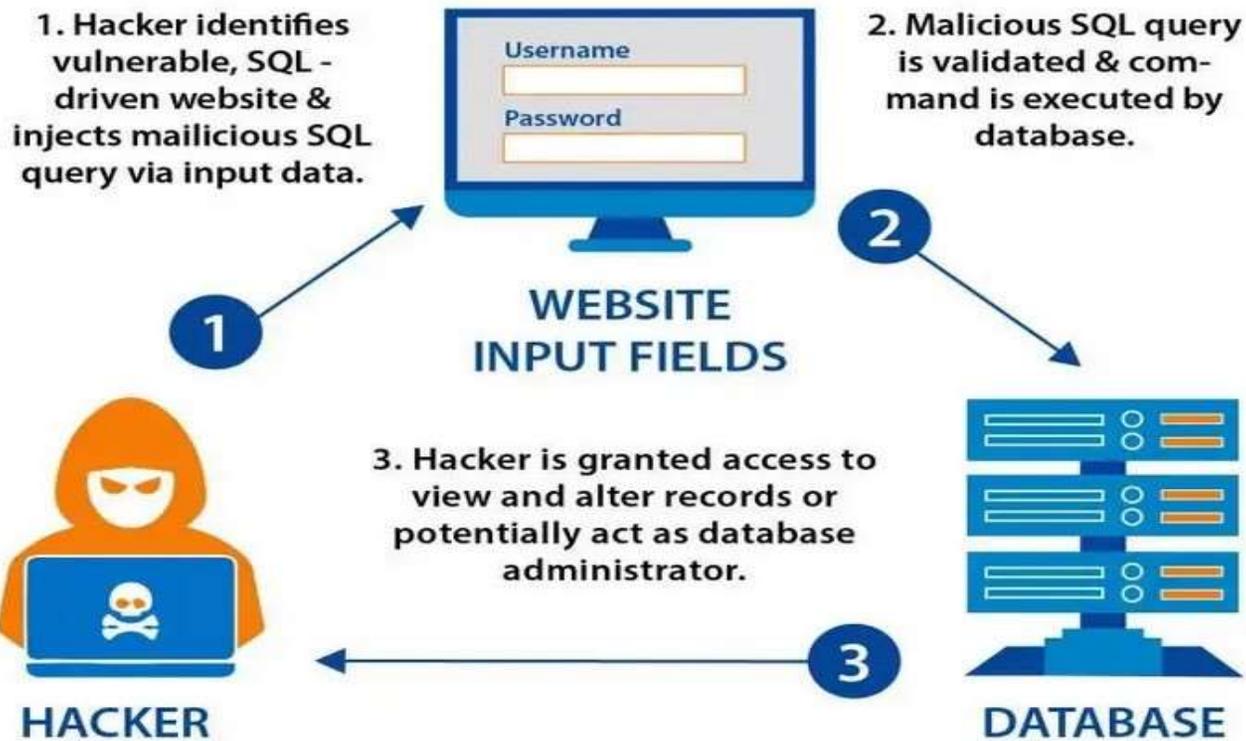
- Deploying input validation mechanisms.
- Using content security policies.
- Regular scanning of web applications.

## SQL injection (SQLi)

---

- SQLi is an attack that allows an attacker to execute malicious SQL statements, which grant the attacker control over a SQL database of a web application.
- As a result of deploying SQLi, the attacker might be able to access portion or entire SQL database of a web page. In addition to adding, modifying, or deleting records in the database.

## Example of SQLi



## Example2 of SQLi

---

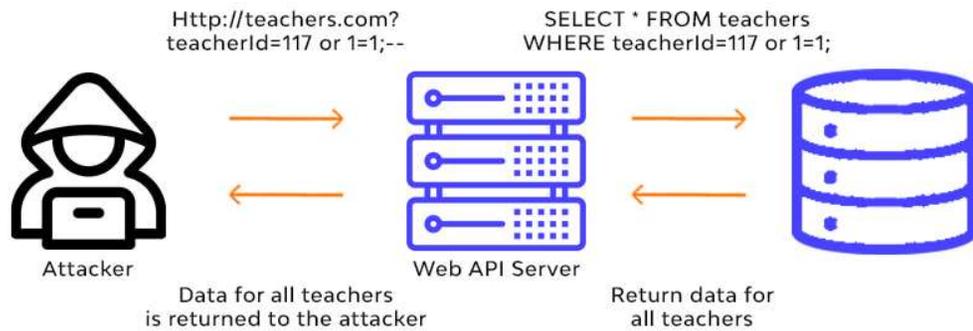
Enter Customer Number 385762

Customer	Acct #	Balance	Payments
385762	90021	3451.32	87,239

Enter Customer Number 385762 or ' 1=1 --

Customer	Acct#	Balance	Payments
1	1	400.23	1,413.00
58	5460	132.00	56,212.31
700	324	90.0	21.00
703	64421	42,000	940,310.98
903	21443	103.00	12.10
...			

## SQL Injection



Another  
example



```
SQL
SELECT *
FROM accounts
WHERE username = 'Mohammed'
AND password = 'password123'
```



id	username	password
1	admin	admin
2	derek32	mozDzs3
3	hunterx	asdfg123
4	Mohammed	password123
5	walter	danger12
6	yasmine	yooS2002

## Example2 of SQLi, Cont'd

---

- ❖ `string SelectedCustomer = UserInput.Text;`
- ❖ `string SQL = "Select * from Customers  
where CustomerID = " + SelectedCustomer;`
- ❖ `Command.Execute SQL;`

## SQLi causes and countermeasures

---

### ❖ Causes:

- Ineffective or lacking of input validation
- Using dynamic SQL.
- Running applications with high privileges.

### ❖ Countermeasures:

- Using white-list input validation.
- Using parameter SQL statement (so user can read characters instead of letting the browser execute the script).
- Using stored procedures with no dynamic SQL.
- Running applications with least privileges.

# Usable and Security

---

# Usability

---

- The mechanism of employing a system to achieve a set of goals, by taking in the consideration effectiveness, efficiency, and satisfaction.
- Usability is deployed to improve user experience and interaction with systems.

# Usability Components

---

- **Effectiveness:** الفعالية

The ability of a system to provide facilities/features to users to reach their goals.

- **Efficiency:** الكفاءة

The amount of available resources (e.g. time, effort, actions) that can be utilized by users to reach their goals.

- **Satisfaction:** الرضى

The measurement of how pleasant the user is when using a system.

# Usability Components

---

- Effectiveness:

- Can users achieve their goals with the system?
- Can users do what the system says it should be able to do?

- Efficiency:

- How much effort is required from users in order to achieve their goals?

- Satisfaction:

- Is the system pleasant to use?

# Security vs. Usability

---

- Security is a process, rather than a product.
- In security, humans are the weakest link.
- Therefore, hackers only need one error from this weakest link (humans) in the security process, in order to conduct a successful attack.
- Social engineering attacks work pretty good in this context.

# Security vs. Usability

---

Confidentiality  
Integrity  
Availability

**VS.**

Effectiveness  
Efficiency  
Satisfaction

# Security vs. Usability

HOW DO WE FIND THE  
PERFECT BALANCE?

**USABILITY**

**SECURITY**



# Security vs. Usability

**THERE IS NO  
ONE-SIZE-FITS-ALL  
SOLUTION.**

# Security vs. Usability

PEOPLE HAVE DIFFERENT **EXPECTATIONS.**



PEOPLE HAVE DIFFERENT **NEEDS.**

# Security vs. Usability

PEOPLE WILL ALWAYS USE YOUR  
APPLICATION IN **UNEXPECTED WAYS.**

THEY WILL DO WHAT YOU ARE  
**LEAST PREPARED FOR.**

# Security vs. Usability

IF YOUR USER EXPERIENCE IS SO BAD  
THAT YOUR PRODUCT HAS NO USERS...

**DOES IT MATTER THAT IT'S TERRIBLY  
INSECURE?**

# Security vs. Usability

IF YOUR PRODUCT HAS ALL THE USERS,  
BUT THEY LOSE THEIR MONEY BECAUSE  
YOUR PRODUCT IS INSECURE...

**IS THE USER EXPERIENCE STILL GOOD?**

# Security-Usability dilemma

---

- Usually the user looks for the effectiveness, efficiency, and satisfaction of a system, rather than the confidentiality, integrity, and availability of that system.
- In other words, users look for the ease of use, rather than the security of a system.

# Example: Passwords

---

- If a password is *very* strong (secure), then it is not usable (hard to remember).
- If a password is usable (easy to remember), then it is very weak (insecure).
- If a strong password should be used, but the user can not remember it, then the user will write it down.

# Usable Security

---

Passwords Security-Usability dilemma solutions:

- Passphrases
- Frequently changed passwords
- Dynamic passwords
- Graphical passwords
- Hardware-based solutions (e.g. Tokens)

# Graphical Passwords

---

- Graphical passwords could be a good solution for the security-usability dilemma:
  - Larger password space
  - More difficult to build dictionary
  - Easier to remember and harder to forget
  - Better balance between security and usability

## Example2: CAPTCHA

- Completely Automated Public Turing test to tell Computers and Humans Apart
- Represents a form of challenge-response test used in systems to determine whether the user is human.



Type the characters above:

# Usable Security

---

- CAPTCHA security-usability dilemma:
  - If a captcha is very strong, then it is hard for machines, and also hard to be solved by users.
  - If a captcha is easy for users to solve, then it is often weak (easy for machine to recognize).

# Usable Security

---

- Can we find a better CAPTCHA scheme that provides a good balance between security and usability?
  - CAPTCHA + Behavioral Biometrics
  - CAPTCHA + BMI (Brain-Machine Interface)

# Usable Security

Usable Security is the study of how we can best *balance* the needs of security with how the users of that system wish to use it.

# Usable Security

---

- Good Practices:
  - Deploy strong cryptography algorithms in data communications.
  - Assure the user involvement in the system design process.
  - Conduct user modeling for new security features.

And that's it  
Any questions?

---

Thank You All And Good Luck

---