

UNIT 1

Security:

Computer Security: The protection afforded to an automated information system in order to attain the applicable objectives of preserving the integrity, availability, and confidentiality of information system resources (includes hardware, software, firmware, information/data, and telecommunications).

This definition introduces three key objectives that are at the heart of computer security:

•**Confidentiality**: This term covers two related concepts:

- Data confidentiality: Assures that private or confidential information is not made available or disclosed to unauthorised individuals.
- Privacy: Assures that individuals control or influence what information related to them may be collected and stored and by whom and to whom that information may be disclosed.

•**Integrity**: This term covers two related concepts:

- Data integrity: Assures that information and programs are changed only in a specified and authorised manner.
- System integrity: Assures that a system performs its intended function in an unimpaired manner, free from deliberate or inadvertent unauthorised manipulation of the system.

•**Availability**: Assures that systems work promptly and service is not denied to authorised users.

These three concepts form what is often referred to as the **CIA triad**. The three concepts embody the fundamental security objectives for both data and for information and computing services.

The OSI Security Architecture:

Is a systematic way of defining the requirements for security and characterising the approaches to satisfying those requirements.

ITU-T3 Recommendation X.800, Security Architecture for OSI, defines such a systematic approach.⁴ The OSI security architecture is useful to managers as a way of organising the task of providing security. Furthermore, because this architecture was developed as an international standard, computer and communications vendors have developed security features for their products and services that relate to this structured definition of services and mechanisms. . The OSI security architecture focuses on security attacks, mechanisms, and services. These can be defined briefly as :

•**Security attack**: Any action that compromises the security of information owned by an organisation.

• **Security mechanism**: A process (or a device incorporating such a process) that is designed to detect, prevent, or recover from a security attack.

• **Security service**: A processing or communication service that enhances the security of the data processing systems and the information transfers of an organisation. The services are intended to counter security attacks, and they make use of one or more security mechanisms to provide the service.

The International Telecommunication Union (ITU) Telecommunication Standardization Sector (ITU-T) is a United Nations-sponsored agency that develops standards, called Recommendations, relating to telecommunications and to open systems interconnection (OSI).

Security Attacks:

Can be divided mainly into two types :

Passive Attacks : A passive attack attempts to learn or make use of information from the system but does not affect system resources.

Active Attacks: An active attack attempts to alter system resources or affect their operation.

Passive Attacks -

Passive attacks are in the nature of eavesdropping on, or monitoring of, transmissions. The goal of the opponent is to obtain information that is being transmitted. Two types of passive attacks are the release of message contents and traffic analysis.

The **release of message contents** is easily understood. A telephone conversation, an electronic mail message, and a transferred file may contain sensitive or confidential information. We would like to prevent an opponent from learning the contents of these transmissions. A second type of passive attack, **traffic analysis**, is subtler. Suppose that we had a way of masking the contents of messages or other information traffic so that opponents, even if they captured the message, could not extract the information from the message. The common technique for masking contents is encryption. If we had encryption protection in place, an opponent might still be able to observe the pattern of these messages. The opponent could determine the location and identity of communicating hosts and could observe the frequency and length of messages being exchanged. This information might be useful in guessing the nature of the communication that was taking place.

Passive attacks are very difficult to detect, because they do not involve any alteration of the data. Typically, the message traffic is sent and received in an apparently normal fashion, and neither the sender nor receiver is aware that a third party has read the messages or observed the traffic pattern. However, it is feasible to prevent the success of these attacks, usually by means of encryption. Thus, the emphasis in dealing with passive attacks is on prevention rather than detection.

Active Attacks -

Active attacks involve some modification of the data stream or the creation of a false stream and can be subdivided into four categories: masquerade, replay, modification of messages, and denial of service.

A **masquerade** takes place when one entity pretends to be a different entity. A masquerade attack usually includes one of the other forms of active attack. For example, authentication sequences can be captured and replayed after a valid authentication sequence has taken place, thus enabling an authorised entity with few privileges to obtain extra privileges by impersonating an entity that has those privileges.

Replay involves the passive capture of a data unit and its subsequent retransmission to produce an unauthorised effect

Modification of messages simply means that some portion of a legitimate message is altered, or that messages are delayed or reordered, to produce an unauthorised effect. The **denial of service** prevents or inhibits the normal use or management of communications facilities. This attack may have a specific target; for example, an entity may suppress all messages directed to a particular destination. Another form of service denial is the disruption of an entire network, either by disabling the network or by overloading it with messages so as to degrade performance.

Active attacks present the opposite characteristics of passive attacks. Whereas passive attacks are difficult to detect, measures are available to prevent their success. On the other hand, it is quite difficult to prevent active attacks because of the wide variety of potential physical, software, and network vulnerabilities. Instead, the goal is to detect active attacks and to recover from any disruption or delays caused by them. If the detection has a deterrent effect, it may also contribute to prevention.

Security Services : (For some reason has two defs from two organisations) **X.800** defines a security service as a service that is provided by a protocol layer of communicating open systems and that ensures adequate security of the systems or of data transfers **RFC 4949**, which provides the following definition: a processing or communication service that is provided by a system to give a specific kind of protection to system resources; security services implement security policies and are implemented by security mechanisms. **X.800 divides** these services into **five categories and fourteen specific services**

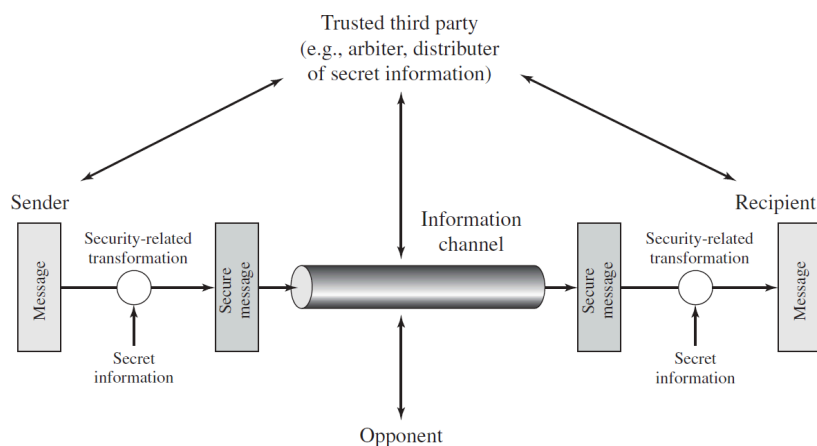
1. **AUTHENTICATION** The assurance that the communicating entity is the one that it claims to be. This further has two types :
 - **Peer Entity Authentication:** Used in association with a logical connection to provide confidence in the identity of the entities connected
 - **Data-Origin Authentication :** In a connectionless transfer, provides assurance that the source of received data is as claimed.
2. **ACCESS CONTROL** The prevention of unauthorised use of a resource (i.e., this service controls who can have access to a resource, under what conditions access can occur, and what those accessing the resource are allowed to do).
3. **DATA CONFIDENTIALITY** The protection of data from unauthorised disclosure. There are main four types :
 - **Connection Confidentiality** The protection of all user data on a connection.
 - **Connectionless Confidentiality** The protection of all user data in a single data block
 - **Selective-Field Confidentiality** The confidentiality of selected fields within the user data on a connection or in a single data block.
 - **Traffic-Flow Confidentiality** The protection of the information that might be derived from observation of traffic flows.
4. **DATA INTEGRITY** The assurance that data received are exactly as sent by an authorised entity (i.e., contain no modification, insertion, deletion, or replay). the types are :
 - **Connection Integrity with Recovery** Provides for the integrity of all user data on a connection and detects any modification, insertion, deletion, or replay of any data within an entire data sequence, with recovery attempted.
 - **Connection Integrity without Recovery** As above, but provides only detection without recovery.
 - **Selective-Field Connection Integrity** Provides for the integrity of selected fields within the user data of a data block transferred over a connection and takes the form of determination of whether the selected fields have been modified, inserted, deleted, or replayed.
 - **Connectionless Integrity** Provides for the integrity of a single connectionless data block and may take the form of detection of data modification. Additionally, a limited form of replay detection may be provided.
 - **Selective-Field Connectionless Integrity** Provides for the integrity of selected fields within a single connectionless data block; takes the form of determination of whether the selected fields have been modified.
5. **NONREPUDIATION** Provides protection against denial by one of the entities involved in a communication of having participated in all or part of the communication (Basically whatsapp single tick double tick and bluetick feature). There are two types :
 - **Nonrepudiation, Origin Proof** that the message was sent by the specified party.
 - **Nonrepudiation, Destination Proof** that the message was received by the specified party.
6. **AVAILABILITY SERVICE** Both X.800 and RFC 4949 define availability to be the property of a system or a system resource being accessible and usable upon demand by an authorised system entity, according to performance specifications for the system. A variety of attacks can result in the loss of or reduction in availability. Some of these attacks are amenable to automated countermeasures, such as authentication and encryption, whereas others require some sort of physical action to prevent or recover from loss of availability of elements of a distributed system. X.800 treats availability as a property to be associated with various security services. However, it makes sense to call out specifically an availability service. **An availability service is one that protects a system to ensure its availability. This service addresses the security concerns raised by denial-of-service attacks. It**

depends on proper management and control of system resources and thus depends on access control service and other security services.

Security Mechanisms :

1. **SPECIFIC SECURITY MECHANISMS** May be incorporated into the appropriate protocol layer in order to provide some of the OSI security services.
 - a. **Encipherment** The use of mathematical algorithms to transform data into a form that is not readily intelligible. The transformation and subsequent recovery of the data depend on an algorithm and zero or more encryption keys.
 - b. **Digital Signature** Data appended to, or a cryptographic transformation of, a data unit that allows a recipient of the data unit to prove the source and integrity of the data unit and protect against forgery (e.g., by the recipient)
 - c. **Access Control** A variety of mechanisms that enforce access rights to resources.
 - d. **Data Integrity** A variety of mechanisms used to assure the integrity of a data unit or stream of data units.
 - e. **Authentication Exchange** A mechanism intended to ensure the identity of an entity by means of information exchange.
 - f. **Traffic Padding** The insertion of bits into gaps in a data stream to frustrate traffic analysis attempts.
 - g. **Routing Control** Enables selection of particular physically secure routes for certain data and allows routing changes, especially when a breach of security is suspected.
 - h. **Notarization** The use of a trusted third party to assure certain properties of a data exchange.
2. **PERVASIVE SECURITY MECHANISMS** Mechanisms that are not specific to any particular OSI security service or protocol layer.
 - a. **Trusted Functionality** That which is perceived to be correct with respect to some criteria (e.g., as established by a security policy).
 - b. **Security Label** The marking bound to a resource (which may be a data unit) that names or designates the security attributes of that resource.
 - c. **Event Detection** Detection of security-relevant events.
 - d. **Security Audit Trail** Data collected and potentially used to facilitate a security audit, which is an independent review and examination of system records and activities.
 - e. **Security Recovery** Deals with requests from mechanisms, such as event handling and management functions, and takes recovery actions.

MODEL FOR NETWORK SEC



This general model shows that there are four basic tasks in designing a particular security service:

1. Design an algorithm for performing the security-related transformation. The algorithm should be such that an opponent cannot defeat its purpose.
2. Generate the secret information to be used with the algorithm.
3. Develop methods for the distribution and sharing of the secret information.
4. Specify a protocol to be used by the two principals that makes use of the security algorithm and the secret information to achieve a particular security service.

CRYPTOGRAPHY!!!!

referred as the science and art of transforming messages to make them secure and immune to attacks

can characterize by:

type of encryption operations used
substitution / transposition / product
number of keys used
single-key or private / two-key or public
way in which plaintext is processed
block / stream

The number of keys used. If both sender and receiver use the same key, the system is referred to as **symmetric, single-key, secret-key, or conventional encryption**. If the sender and receiver use different keys, the system is referred to as **asymmetric, two-key, or public-key encryption**.

The way in which the plaintext is processed. A **block cipher** processes the input one block of elements at a time, producing an output block for each input block. A **stream cipher** processes the input elements continuously, producing output one element at a time, as it goes along

ATTACKS ON CRYPTOGRAPHY :

Cryptanalysis: Cryptanalytic attacks rely on the nature of the algorithm plus perhaps some knowledge of the general characteristics of the plaintext or even some sample plaintext–ciphertext pairs. This type of attack exploits the characteristics of the algorithm to attempt to deduce a specific plaintext or to deduce the key being used.

•**Brute-force attack:** The attacker tries every possible key on a piece of ciphertext until an intelligible translation into plaintext is obtained. On average, half of all possible keys must be tried to achieve success. If either type of attack succeeds in deducing the key, the effect is catastrophic: All future and past messages encrypted with that key are compromised.

Symmetric Cipher Model:

Advantages :

- Algorithm used for decryption is reverse of encryption
- i.e if encryption uses a combination of addition and multiplication decryption is combination of division and subtraction
- Symmetric algorithms are efficient
- Take less time to encrypt than asymmetric

Disadvantages :

- Each pair must have a unique symmetric key
- If N people want to use there need **$n(n-1)/2$ keys**
- Distribution of keys between two parties can be difficult

Monoalphabetic : A character changes to the same character always regardless of its position in the text

CAESAR CIPHER (Easiest , earliest cipher by julius)

Formula

A shift may be of any amount, so that the general Caesar algorithm is

$$C = E(k, p) = (p + k) \bmod 26$$

where k takes on a value in the range 1 to 25. The decryption algorithm is simply

$$p = D(k, C) = (C - k) \bmod 26$$

Mono Alphabetic Cipher assigns random alphabet to plaintext for a char doesnt follow particular shift like caesar all stored in a dict and that is used to decrypt too

PLAYFAIR Cipher

Playfair Cipher The best-known multiple-letter encryption cipher is the Playfair, which treats digrams in the plaintext as single units and translates these units into ciphertext digrams

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

Rules

- 1.Repeating plaintext letters that are in the same pair are separated with a filler letter, such as x, so that balloon would be treated as ba lx lo on.
2. Two plaintext letters that fall in the same row of the matrix are each replaced by the letter to the right, with the first element of the row circularly following the last. For example, ar is encrypted as RM.
3. Two plaintext letters that fall in the same column are each replaced by the letter beneath, with the top element of the column circularly following the last. For example, mu is encrypted as CM.
- 4.Otherwise, each plaintext letter in a pair is replaced by the letter that lies in its own row and the column occupied by the other plaintext letter. Thus, hs becomes BP and ea becomes IM (or JM, as the encipherer wishes).

VIGNERE CIPHER -

		Plaintext																											
Key		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		
	A	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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	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		

Example: The plaintext is "JAVATPOINT", and the key is "BEST".

To generate a new key, the given key is repeated in a circular manner, as long as the length of the plain text does not equal to the new key.

J	A	V	A	T	P	O	I	N	T
B	E	S	T	B	E	S	T	B	E

Now to encrypt see where j and b intersect and that is your letter

To decrypt see in the key rows where cipher text appears for example ciphertext[1]=k and key[1]=b

Then see in b ka row where k appears and see the column to which k belongs in in this case "j"

Formula

$E_i = (P_i + K_i) \bmod 26$ -encryption

$D_i = (E_i - K_i) \bmod 26$ -decryption

VERNAM Cipher

Take the key len same as plain text then formula is

Result = ascii of plaintext-97 +Ascii of key -97mod 26

And message will be ascii of result +97

To decrypt :

Result = Ascii of Cipher -97 - Ascii of key - 97 mod 26

De_msg = Ascii of result +97

ABove all were substitution techniques

TRANSPOSITION TECHNIQUES -

Rail fence

Basically shifts down one space in diagonal and key is depth of array and cipher

text is array read row wise

```
m e m a t r h t g p r y
e t e f e t e o a a t
```

The encrypted message is

MEMATRHTGPRYETEFETEOAAT

Here depth is two and the decrypted message will be “meet me after the toga party “
Columnar

Here the matrix is filled normally row wise then the key is random order of columns
So that order forms the cipher text

```
Key:      4 3 1 2 5 6 7
Plaintext: a t t a c k p
           o s t p o n e
           d u n t i l t
           w o a m x y z
Ciphertext: TTNAAPTMTSUOAODWCOIXKNLYPETZ
```

Here 4312567 is the key

Size of matrix is fixed

Stenography

- Steganography is the art and science of writing hidden message in such a way that no one, apart from the sender and intended recipient, suspects the existence of the message. • Steganography works by replacing bits of useless or unused data in regular computer files (such as graphics, sound, text, html or even floppy disks) with bits of different, invisible information.
- This hidden information can be plain text, cipher text or even images.
- In modern steganography, data is first encrypted by the usual means and then inserted, using a special algorithm, into redundant data that is part of a particular file format such as a JPEG image, Bitmap image.
- Steganography process :
 - Cover-media + Hidden data + Stego-key = Stego-medium
- Cover media:
 - It is the file in which we will hide the hidden data
 - Cover-media can be image or audio file.
- stego-key:
 - Cover-media can be encrypted using stego-key:
- Stego-medium:
 - The resultant file is of above process called stego medium

Types of Stenography

- Text
 - plaintext=Since everyone can read, encoding text in neutral sentences is doubtfully effective
 - **Since Everyone Can Read, Encoding Text**
 - **In Neutral Sentences Is Doubtfully Effective**
 - Answer- ‘Secret inside’
- Image



- Audio

- It is a technique used to transmit hidden information by modifying an audio signal in an imperceptible manner. It is the science of hiding some secret text or audio information in a host message. The host message before steganography and stego message after steganography have the same characteristics.

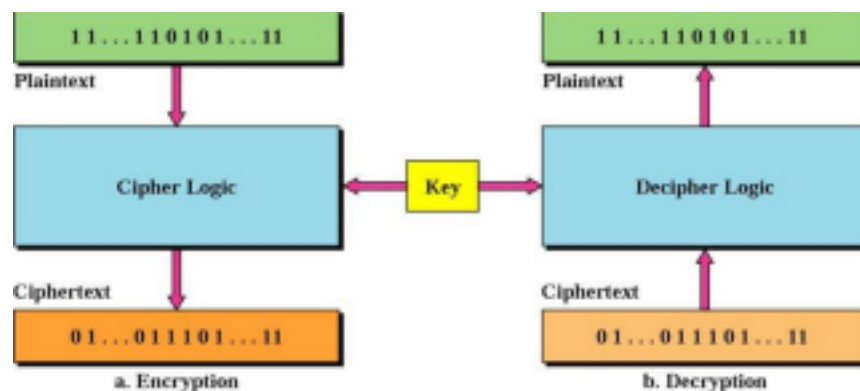
Steganography	Cryptography
Unknown message passing	Known message passing
Steganography prevents discovery of the very existence of communication	Encryption prevents an unauthorized party from discovering the contents of a communication
Little known technology	Common technology
Technology still being develop for certain formats	Most of algorithm known by all
Once detected message is known	Strong current algorithm are resistant to attacks ,larger expensive computing power is required for cracking
Steganography does not alter the structure of the secret message	Cryptography alter the structure of the secret message

P Box

- Permutation box
- Performs transposition at bit level
- Transposes bits
- The key and the encryption/decryption algo are embedded in the hardware
- Plain text and cipher text have the same number of 1s and 0s

Block Cipher

- Traditional ciphers used character or symbols as he unit of encryption/decryption
- Modern ciphers use a block of bits as a unit of encryption and decryption



Block Cipher Principles

- most symmetric block ciphers are based on a Feistel Cipher Structure
- needed since must be able to decrypt ciphertext to recover messages efficiently
- block ciphers look like an extremely large substitution
- would need table of 264 entries for a 64-bit block
- instead create from smaller building blocks
- using idea of a product cipher

Confusion vs Diffusion

- cipher needs to completely obscure statistical properties of original message
- a one-time pad does this
- more practically Shannon suggested combining S & P elements to obtain:
 - diffusion – dissipates statistical structure of plaintext over bulk of ciphertext
 - confusion – makes relationship between ciphertext and key as complex as possible

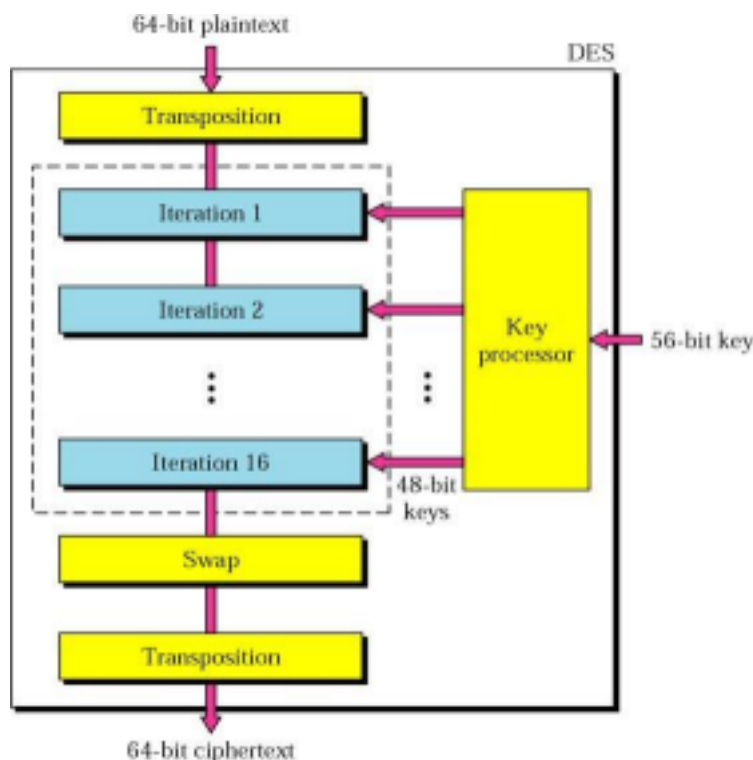
S Box

- Substitution box
- Performs substitution at bit level
- Transposes the permuted bits
- substitutes one decimal digit with another
- 3 components
 - Encoder
 - Decoder
 - P box

Data Encryption Standard (DES)

- most widely used block cipher in world
- encrypts 64-bit data using 56-bit key
- has widespread use
- has been considerable controversy over its security

DES has 2 transposition blocks one swapping block 16 complex blocks called the iteration blocks



The DES cipher uses the same concept as the Caesar cipher, but the encryption/decryption algorithm is much more complex due to the sixteen 48-bit keys derived from a 56-bit key.

Advanced Encryption System(AES)

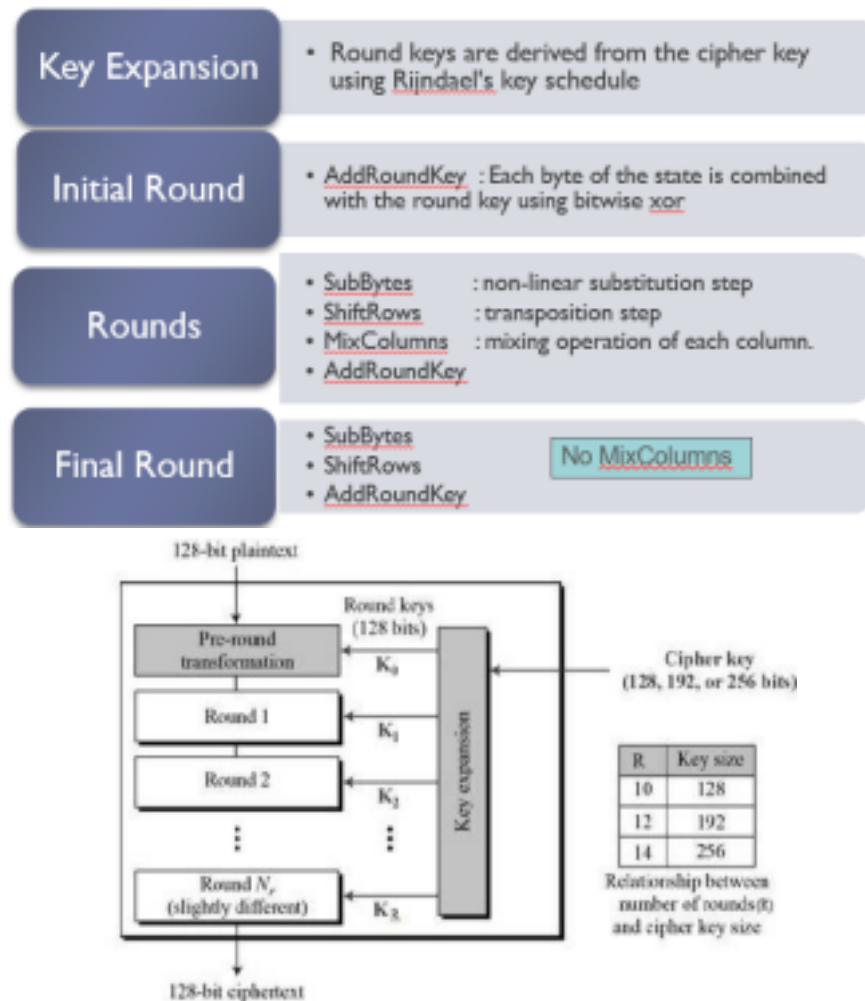
AES Competition Requirements

Private key symmetric block cipher

- 128-bit data, 128/192/256-bit keys
- Stronger & faster than Triple-DES
- Provide full specification & design details
- Both C & Java implementations

Features of AES

- SP Network: It works on an SP network structure rather than a Feistel cipher structure, as seen in the case of the DES algorithm.
- Key Expansion: It takes a single key up during the first stage, which is later expanded to multiple keys used in individual rounds.
- Byte Data: The AES encryption algorithm does operations on byte data instead of bit data. So it treats the 128-bit block size as 16 bytes during the encryption procedure.
- Key Length: The number of rounds to be carried out depends on the length of the key being used to encrypt data. The 128-bit key size has ten rounds, the 192-bit key size has 12 rounds, and the 256-bit key size has 14 rounds.



AES Security

- AES was designed after DES.
- Most of the known attacks on DES were already tested on AES.
- Brute-Force Attack
 - AES is definitely more secure than DES due to the larger-size key.
- Statistical Attacks
 - Numerous tests have failed to do statistical analysis of the ciphertext
- Differential and Linear Attacks
 - There are no differential and linear attacks on AES as yet.

DES Algorithm	AES Algorithm
Key Length - 56 bits	Key Length - 128, 192, 256 bits
Block Size - 64 bits	Block size - 128 bits
Fixed no. of rounds	No. of rounds dependent on key length
Slower and less secure	Faster and more secure

Modes of Operation

- block ciphers encrypt fixed size blocks
- eg. DES encrypts 64-bit blocks, with 56-bit key
- need way to use in practise, given usually have arbitrary amount of information to encrypt
- four were defined for DES in ANSI standard ANSI X3.106-1983 Modes of Use •
- subsequently now have 5 for DES and AES
- have block and stream modes

Electronic Codebook Book (ECB)

- message is broken into independent blocks which are encrypted
- each block is a value which is substituted, like a codebook, hence name
- each block is encoded independently of the other blocks
- $C_i = DES_{K_1}(P_i)$
- uses: secure transmission of single values

Advantages of using ECB –

- Parallel encryption of blocks of bits is possible, thus it is a faster way of encryption.
- Simple way of the block cipher.

Disadvantages of using ECB –

- Prone to cryptanalysis since there is a direct relationship between plaintext and ciphertext.

Cipher Block Chaining(CBC)

- message is broken into blocks
- but these are linked together in the encryption operation
 - each previous cipher blocks is chained with current plaintext block, hence name
- use Initial Vector (IV) to start process
 - $C_i = DES_{K_1}(P_i \text{ XOR } C_{i-1})$
 - $C_{-1} = IV$
- uses: bulk data encryption, authentication

Advantages of CBC –

- CBC works well for input greater than b bits.
- CBC is a good authentication mechanism.

- Better resistive nature towards cryptanalysis than ECB.

Disadvantages of CBC –

- Parallel encryption is not possible since every encryption requires a previous cipher.

Cipher Feedback(CFB)

- message is treated as a stream of bits
- added to the output of the block cipher
- result is feed back for next stage (hence name)
- standard allows any number of bit (1,8 or 64 or whatever) to be feed back
- denoted CFB-1, CFB-8, CFB-64 etc
- is most efficient to use all 64 bits (CFB-64)
 - $C_i = P_i \text{ XOR } \text{DESK}_1(C_{i-1})$
 - $C_{-1} = IV$
- uses: stream data encryption, authentication

Advantages of CFB –

- Since, there is some data loss due to the use of shift register, thus it is difficult for applying cryptanalysis.

Disadvantages of using CFB –

- The drawbacks of CFB are the same as those of CBC mode. Both block losses and concurrent encryption of several blocks are not supported by the encryption. Decryption, however, is parallelizable and loss-tolerant.

Output Feedback(OFB)

- message is treated as a stream of bits
- output of cipher is added to message
- output is then feed back (hence name)
- feedback is independent of message
- can be computed in advance
 - $C_i = P_i \text{ XOR } O_i$
 - $O_i = \text{DESK}_1(O_{i-1})$
 - $O_{-1} = IV$
- uses: stream encryption over noisy channels

Advantages of OFB –

- In the case of CFB, a single bit error in a block is propagated to all subsequent blocks. This problem is solved by OFB as it is free from bit errors in the plaintext block.

Disadvantages of OFB-

- The drawback of OFB is that, because to its operational modes, it is more susceptible to a message stream modification attack than CFB.

Counter (CTR)

Counter Mode –

- The Counter Mode or CTR is a simple counter-based block cipher implementation. • Every time a counter-initiated value is encrypted and given as input to XOR with plaintext which results in ciphertext block.
- The CTR mode is independent of feedback use and thus can be implemented in parallel.

Advantages of Counter –

- Since there is a different counter value for each block, the direct plaintext and ciphertext relationship is avoided. This means that the same plain text can map to different ciphertext. • Parallel execution of encryption is possible as outputs from previous stages are not chained as in the case of CBC.

Disadvantages of Counter-

- The fact that CTR mode requires a synchronous counter at both the transmitter and the

receiver is a severe drawback. The recovery of plaintext is erroneous when synchronization is lost.

Application of Block Ciphers

- **Data Encryption:** Block Ciphers are widely used for the encryption of private and sensitive data such as passwords, credit card details and other information that is transmitted or stored for a communication. This encryption process converts a plain data into non-readable and complex form. Encrypted data can be decrypted only by the authorised person with the private keys.
- **File and Disk Encryption:** Block Ciphers are used for encryption of entire files and disks in order to protect their contents and restrict from unauthorised users. The disk encryption softwares such as BitLocker, TrueCrypt also uses block cipher to encrypt data and make it secure.
- **Virtual Private Networks (VPN):** Virtual Private Networks (VPN) use block cipher for the encryption of data that is being transmitted between the two communicating devices over the internet. This process makes sure that data is not accessed by unauthorised person when it is being transmitted to another user.
- **Secure Sockets Layer (SSL) and Transport Layer Security (TLS):** SSL and TLS protocols use block ciphers for encryption of data that is transmitted between web browsers and servers over the internet. This encryption process provides security to confidential data such as login credentials, card information etc.
- **Digital Signatures:** Block ciphers are used in the digital signature algorithms, to provide authenticity and integrity to the digital documents. This encryption process generates the unique signature for each document that is used for verifying the authenticity and detecting if any malicious activity is detected.

Public Key Cryptography

- Two keys
- Public and private key
- Public key is announced to the public

Advantages

- Removes the restriction of a shared symmetric key between two entities
- Number of keys needed is reduced
- For 10 users require 20 keys

Disadvantages

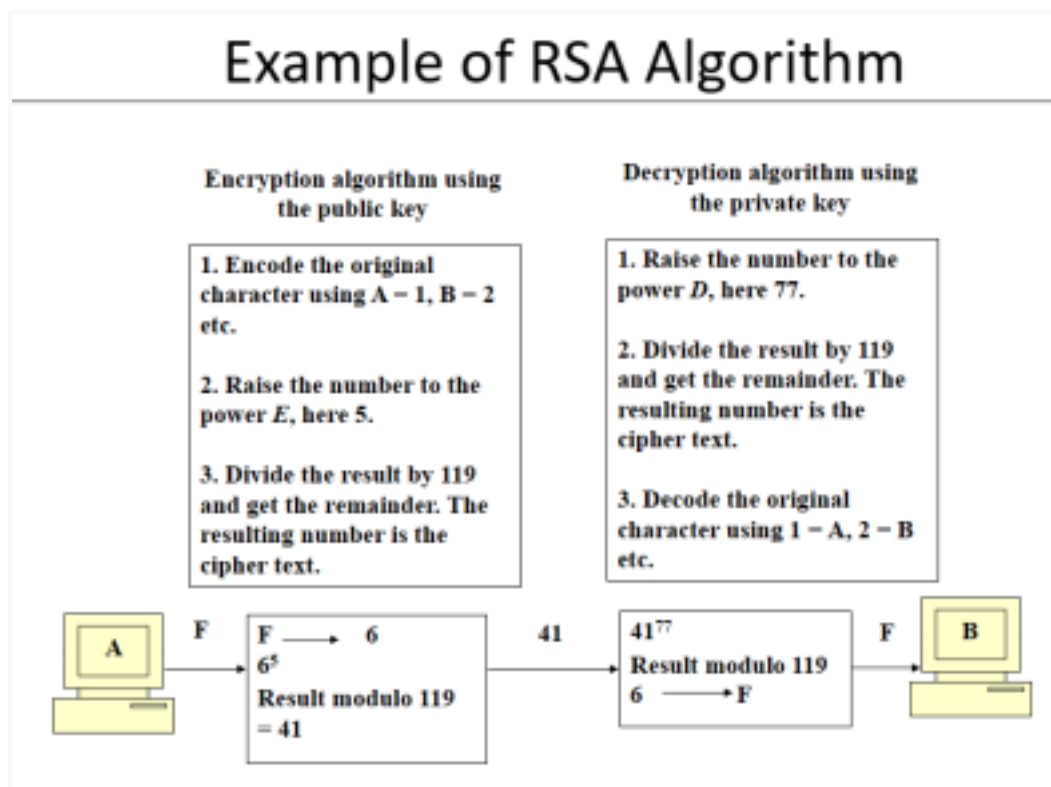
- Complex algorithms
- Association between the entity and the public key must be verified

Public-key algorithms are more efficient for short messages. RSA (Rivest, Shamir, Adleman)

- The most popular one.
- Assumption/theoretical basis:
- Factoring a big number is hard.
- Variable key length (usually 512 bits).
- Variable plaintext block size.
- Plaintext must be “smaller” than the key.
- Ciphertext block size is the same as the key length.
- Based on the theory of Prime Numbers

Algorithm

1. Choose two large prime numbers P and Q .
2. Calculate $N = P \times Q$.
3. Select the public key (i.e. the encryption key) E such that it is not a factor of $(P - 1)$ and $(Q - 1)$.
4. Select the private key (i.e. the decryption key) D such that the following equation is true: $(D \times E) \bmod (P - 1) \times (Q - 1) = 1$
5. For encryption, calculate the cipher text CT from the plain text PT as follows:
 $CT = PTE \bmod N$
6. Send CT as the cipher text to the receiver.
7. For decryption, calculate the plain text PT from the cipher text CT as follows:
8. $PT = CTD \bmod N$



Diffie Hellman Key Exchange Algorithm

- Discovered by Whitfield Diffie and Martin Hellman
- “New Directions in Cryptography”
- Diffie-Hellman key agreement protocol
 - Exponential key agreement
 - Allows two users to exchange a secret key
 - Requires no prior secrets
 - Real-time over an untrusted network
- Based on the difficulty of computing discrete logarithms of large numbers.
- No known successful attack strategies*
- Requires two large numbers, one prime (P), and (G), a primitive root of P
 1. P and G are both publicly available numbers
 2. P is at least 512 bits
 3. Users pick private values a and b

4. Compute public values
5. $x = ga \bmod p$
6. $y = gb \bmod p$
7. Public values x and y are exchanged

Compute shared, private key

$$ka = ya \bmod p$$

$$kb = xb \bmod p$$

Algebraically it can be shown that $ka = kb$

Users now have a symmetric secret key to encrypt

Example

Alice and Bob get public numbers

$$P = 23, G = 9$$

Alice and Bob compute public values

$$X = 94 \bmod 23 = 6561 \bmod 23 = 6$$

$$Y = 93 \bmod 23 = 729 \bmod 23 = 16$$

Alice and Bob exchange public numbers

Alice and Bob compute symmetric keys

$$ka = ya \bmod p = 164 \bmod 23 = 9$$

$$kb = xb \bmod p = 63 \bmod 23 = 9$$

Alice and Bob now can talk securely!

Diffie-Hellman is currently used in many protocols, namely:

- Secure Sockets Layer (SSL)/Transport Layer Security (TLS)
- Secure Shell (SSH)
- Internet Protocol Security (IPSec)
- Public Key Infrastructure (PKI)

UNIT 2 !!

Program Security

We know that security implies some degree of trust that the program enforces expected confidentiality, integrity, and availability.

Early work in computer security was based on the paradigm of "**penetrate and patch**," in which analysts searched for and repaired faults. Often, a top-quality "tiger team" would be convened to test a system's security by attempting to cause it to fail. The test was considered to be a "proof" of security; if the system withstood the attacks, it was considered secure. Unfortunately, far too often the proof became a counterexample, in which not just one but several serious security problems were uncovered. The problem discovery in turn led to a rapid effort to "patch" the system to repair or restore the security. However, the patch efforts were largely useless, making the system less secure rather than more secure because they frequently introduced new faults. There are at least four reasons why.

- The pressure to repair a specific problem encouraged a narrow focus on the fault itself and not on its context. In particular, the analysts paid attention to the immediate cause of the failure and not to the underlying design or requirements faults.
- The fault often had non obvious side effects in places other than the immediate area of the fault.
- Fixing one problem often caused a failure somewhere else, or the patch addressed the problem in only one place, not in other related places.
- The fault could not be fixed properly because system functionality or performance would suffer as a consequence.

The inadequacies of penetrate-and-patch led researchers to seek a better way to be confident that code meets its security requirements. One way to do that is to compare the requirements with the behavior. That is, **to understand program security, we can examine programs to see**

whether they behave as their designers intended or users expected. We call such unexpected behavior a program security flaw; it is inappropriate program behavior caused by a program vulnerability.

Program security flaws can derive from any kind of software fault. That is, they cover everything from a misunderstanding of program requirements to a one-character error in coding or even typing. The flaws can result from problems in a single code component or from the failure of several programs or program pieces to interact compatible through a shared interface. The security flaws can reflect code that was intentionally designed or coded to be malicious or code that was simply developed in a sloppy or misguided way. Thus, it makes sense to divide program flaws into two separate logical categories: inadvertent human errors versus malicious, intentionally induced flaws.

Types of Flaws:

To aid our understanding of the problems and their prevention or correction, we can define categories that distinguish one kind of problem from another. For example, Landwehr et al. present a taxonomy of program flaws, dividing them first into intentional and inadvertent flaws. They further divide intentional flaws into malicious and nonmalicious ones.

In the taxonomy, the **inadvertent** flaws fall into six categories:

- validation error (incomplete or inconsistent): permission checks
- domain error: controlled access to data
- serialisation and aliasing: program flow order
- inadequate identification and authentication: basis for authorization
- boundary condition violation: failure on first or last case
- other exploitable logic errors

NON MALICIOUS PROGRAM ERRORS

programmers and other developers make many mistakes, most of which are unintentional and non malicious. Many such errors cause program malfunctions but do not lead to more serious security vulnerabilities. However, a few classes of errors have plagued programmers and security professionals for decades, and there is no reason to believe they will disappear. There are main three types of errors :

Incomplete Mediation :

Incomplete mediation in non-malicious faults in network security refers to a security vulnerability that occurs when an application does not properly validate all user input before processing it. This can allow an attacker to bypass security controls and gain unauthorized access to sensitive data or resources.

One common example of incomplete mediation is a buffer overflow attack. In this type of attack, the attacker injects more data into a buffer than it can hold, causing the excess data to overwrite adjacent memory. This can overwrite the application's stack frame, which can allow the attacker to execute arbitrary code.

Buffer Overflows :

A buffer overflow is the computing equivalent of trying to pour two liters of water into a one- liter pitcher: Some water is going to spill out and make a mess.

Define :

A buffer (or array or string) is a space in which data can be held. A buffer resides in memory. Because memory is finite, a buffer's capacity is finite. For this reason, in many programming languages the programmer must declare the buffer's maximum size so that the compiler can set aside that amount of space.

Example : Suppose a C language program contains the declaration:
`char sample[10];`

The compiler sets aside 10 bytes to store this buffer, one byte for each of the ten elements of the array, `sample[0]` through `sample[9]`. Now we execute the statement:

```
sample[10] = 'A';
```

The subscript is out of bounds (that is, it does not fall between 0 and 9), so we have a problem. The nicest outcome (from a security perspective) is for the compiler to detect the problem and mark the error during compilation. However, if the statement were

```
sample[i] = 'A';
```

we could not identify the problem until `i` was set during execution to a too-big subscript. It would be useful if, during execution, the system produced an error message warning of a subscript out of bounds. Unfortunately, in some languages, buffer sizes do not have to be predefined, so there is no way to detect an out-of-bounds error.

VIRUS AND OTHER MALICIOUS CODE

Since computer data is not usually seen directly by users, malicious people can make programs serve as vehicles to access and change data and other programs.

Why Worry about Malicious Code :

- Malicious code can do much (harm):

Malicious code can do anything any other program can, such as writing a message on a computer screen, stopping a running program, generating a sound, or erasing a stored file. Or malicious code can do nothing at all right now; it can be planted to lie dormant, undetected, until some event triggers the code to act.

- Malicious Code Has Been Around a Long Time

Kinds of Malicious Code

Malicious code or a **rogue program** is the general name for unanticipated or undesired effects in programs or program parts, caused by an agent intent on damage. This definition eliminates unintentional errors, although they can also have a serious negative effect.

The **agent** is the writer of the program or the person who causes its distribution.

Types :

1. A **virus** is a program that can pass on malicious code to other nonmalicious programs by modifying them.
 - a. A virus can be either transient or resident. A transient virus has a life that depends on the life of its host; the virus runs when its attached program executes and terminates when its attached program ends. (During its execution, the transient virus may have spread its infection to other programs.) A resident virus locates itself in memory; then it can remain active or be activated as a stand-alone program, even after its attached program ends.
2. A **Trojan horse** is malicious code that, in addition to its primary effect, has a second, nonobvious malicious effect.
3. A **logic bomb** is a class of malicious code that "detonates" or goes off when a specified condition occurs. A **time bomb** is a logic bomb whose trigger is a time or date.
4. A **trapdoor** or **backdoor** is a feature in a program by which someone can access the program other than by the obvious, direct call, perhaps with special privileges.
5. A **worm** is a program that spreads copies of itself through a network.
 - a. The primary difference between a worm and a virus is that a worm operates through networks, and a virus can spread through any medium (but usually uses copied program or data files). Additionally, the worm spreads copies of itself as a stand-alone program, whereas the virus spreads copies of itself as a program that attaches to or embeds in other programs.
6. A **rabbit** is a virus or worm that self-replicates without bound, with the intention of exhausting some computing resource.

How Virus Attach

1. Appended Viruses

- a. A program virus attaches itself to a program; then, whenever the program is run, the virus is activated. This kind of attachment is usually easy to program.
- b. In the simplest case, a virus inserts a copy of itself into the executable program file before the first executable instruction. Then, all the virus instructions execute first; after the last virus instruction, control flows naturally to what used to be the first program instruction.

2. Virus Appended to a Program.

- a. This kind of attachment is simple and usually effective. The virus writer does not need to know anything about the program to which the virus will attach, and often the attached program simply serves as a carrier for the virus. Most viruses attach in this manner.

3. Viruses That Surround a Program

- a. An alternative to the attachment is a virus that runs the original program but has control before and after its execution.
- b. The virus writer might arrange for the virus to attach itself to the program that constructs the listing of files on the disk. If the virus regains control after the listing program has generated the listing but before the listing is displayed or printed, the virus could eliminate its entry from the listing and falsify space counts so that it appears not to exist.

4. Document Viruses

- a. Currently, the most popular virus type is what we call the document virus, which is implemented within a formatted document, such as a written document, a database, a slide presentation, or a spreadsheet.

5. Virus Integrated into a Program.

- a. Finally, the virus can replace the entire target, either mimicking the effect of the target or ignoring the expected effect of the target and performing only the virus effect.

How Viruses Gain Control:

The virus (V) has to be invoked instead of the target (T). Essentially, the virus either has to seem to be T, saying effectively "I am T"

Virus Completely Replacing a Program.

The virus can supplant T by altering the sequence that would have invoked T to now invoke the virus V; this invocation can be used to replace parts of the resident operating system by modifying pointers to those resident parts, such as the table of handlers for different kinds of interrupts.

Homes for Viruses

The virus writer may find these qualities appealing in a virus:

- It is hard to detect.
- It is not easily destroyed or deactivated.
- It spreads infection widely.
- It can reinfect its home program or other programs.
- It is easy to create.
- It is machine independent and operating system independent.

Few viruses meet all these criteria. The virus writer chooses from these objectives when deciding what the virus will do and where it will reside.

One-Time Execution

The majority of viruses today execute only once, spreading their infection and causing their effect in that one execution. A virus often arrives as an e-mail attachment of a document virus. It is executed just by being opened.

Boot Sector Viruses

A special case of virus attachment, but formerly a fairly popular one, is the so-called boot sector virus. The boot sector is an especially appealing place to house a virus. The virus gains control very early in the boot process, before most detection tools are active, so that it can avoid, or at least complicate, detection. The files in the boot area are crucial parts of the operating system.

Consequently, to keep users from accidentally modifying or deleting them with disastrous results, the operating system makes them "invisible" by not showing them as part of a normal listing of

stored files, preventing their deletion. Thus, the virus code is not readily noticed by users.

Memory-Resident Viruses

Some parts of the operating system and most user programs execute, terminate, and disappear, with their space in memory being available for anything executed later. For very frequently used parts of the operating system and for a few specialised user programs, it would take too long to reload the program each time it was needed. Such code remains in memory and is called **"resident" code**.

Virus writers also like to attach viruses to resident code because the resident code is activated many times while the machine is running. Each time the resident code runs, the virus does too. Once activated, the virus can look for and infect uninfected carriers.

Macro Virus

One popular home for a virus is an application program. Many applications, such as word processors and spreadsheets, have a "macro" feature, by which a user can record a series of commands and repeat them with one invocation. Such programs also provide a "startup macro" that is executed every time the application is executed. A virus writer can create a virus macro that adds itself to the startup directives for the application. It also then embeds a copy of itself in data files so that the infection spreads to anyone receiving one or more of those files.

Email Virus

Many viruses are transmitted by e-mail, using either of two routes. In the first case, some virus writers generate a new e-mail message to all addresses in the victim's address book. These new messages contain a copy of the virus so that it propagates widely. Often the message is a brief, chatty, non-specific message that would encourage the new recipient to open the attachment from a friend (the first recipient). For example, the subject line or message body may read "I thought you might enjoy this picture from our vacation." In the second case, the virus writer can leave the infected file for the victim to forward unknowingly. If the virus's effect is not immediately obvious, the victim may pass the infected file unwittingly to other victims.

Polymorphic Viruses

The virus signature may be the most reliable way for a virus scanner to identify a virus. If a particular virus always begins with the string 47FoFooEo8 (in hexadecimal) and has string 00113FFF located at word 12, it is unlikely that other programs or data files will have these exact characteristics. For longer signatures, the probability of a correct match increases.

If the virus scanner will always look for those strings, then the clever virus writer can cause something other than those strings to be in those positions. For example, the virus could have two alternative but equivalent beginning words; after being installed, the virus will choose one of the two words for its initial word. Then, a virus scanner would have to look for both patterns. A virus that can change its appearance is called a **polymorphic virus**

TABLE 3-2 Virus Effects and Causes.

Virus Effect	How It Is Caused
Attach to executable program	<ul style="list-style-type: none">• Modify file directory• Write to executable program file
Attach to data or control file	<ul style="list-style-type: none">• Modify directory• Rewrite data• Append to data• Append data to self
Remain in memory handler address table	<ul style="list-style-type: none">• Intercept interrupt by modifying interrupt• Load self in nontransient memory area

Conceal self falsify result	<ul style="list-style-type: none"> • Intercept system calls that would reveal self and • Classify self as "hidden" file
Infect disks	<ul style="list-style-type: none"> • Intercept interrupt • Intercept operating system call (to format disk, for example) • Modify system file • Modify ordinary executable program
Spread infection	<ul style="list-style-type: none"> • Infect boot sector • Infect systems program • Infect ordinary program • Infect data ordinary program reads to control its execution
Prevent deactivation deactivation	<ul style="list-style-type: none"> • Activate before deactivating program and block • Store copy to reinfect after deactivation

Prevention of Virus Infection

The only way to prevent the infection of a virus is not to share executable code with an infected source. This philosophy used to be easy to follow because it was easy to tell if a file was executable or not.

Nevertheless, there are several techniques for building a reasonably safe community for electronic contact, including the following:

- Use only commercial software acquired from reliable, well-established vendors.
- Test all new software on an isolated computer.
 - If you must use software from a questionable source, test the software first on a computer with no hard disk, not connected to a network, and with the boot disk removed. Run the software and look for unexpected behavior, even simple behavior such as unexplained figures on the screen. Test the computer with a copy of an up-to-date virus scanner, created before running the suspect program. Only if the program passes these tests should it be installed on a less isolated machine.
- Open attachments only when you know them to be safe.
- Make a recoverable system image and store it safely.
 - If your system does become infected, this clean version will let you reboot securely because it overwrites the corrupted system files with clean copies.
- Make and retain backup copies of executable system files.
 - This way, in the event of a virus infection, you can remove infected files and reinstall from the clean backup copies (stored in a secure, offline location, of course).
- Use virus detectors (often called virus scanners) regularly and update them daily. Many of the virus detectors available can both detect and eliminate infection from viruses.

TARGETED MALICIOUS PROGRAM

Another class of malicious code is written for a particular system, for a particular application, and for a particular purpose. Many of the virus writers' techniques apply, but there are also some new ones.

Trapdoors

A trapdoor is an undocumented entry point to a module. The trapdoor is inserted during code development, perhaps to test the module, to provide "hooks" by which to connect future modifications or enhancements or to allow access if the module should fail in the future. In addition to these legitimate uses, trapdoors can allow a programmer access to a program once it is placed in production.

Salami Attack

An attack known as a salami attack. This approach gets its name from the way odd bits of meat and

fat are fused together in a sausage or salami. In the same way, a salami attack merges bits of seemingly inconsequential data to yield powerful results. For example, programs often disregard small amounts of money in their computations, as when there are fractional pennies as interest or tax is calculated.

Such programs may be subject to a salami attack, because the small amounts are shaved from each computation and accumulated elsewhere—such as the programmer's bank account! The shaved amount is so small that an individual case is unlikely to be noticed, and the accumulation can be done so that the books still balance overall.

Development Controls are a set of security practices that are integrated into the software development process. These controls can help to prevent or mitigate security vulnerabilities in software. Some examples of development controls include:

- **Secure coding practices:** These practices include using secure coding techniques, such as input validation, data sanitization, and error handling.
- **Security testing:** This involves using a variety of testing techniques to identify and fix security vulnerabilities in software.
- **Static analysis:** This involves using tools to analyze the source code of software for potential security vulnerabilities.
- **Configuration management:** This involves controlling and tracking changes to software and its configuration.

Peer reviews are a process in which developers review each other's code for potential security vulnerabilities. Peer reviews can be an effective way to identify security vulnerabilities that may be missed by individual developers.

Hazard analysis is a systematic process for identifying and assessing potential hazards in a system. Hazard analysis can be used to identify security hazards in software and to develop controls to mitigate those hazards. Some examples of hazard analysis techniques include:

- **Hazard and Operability Studies (HAZOP):** This technique involves brainstorming potential hazards and their consequences.
- **Failure Modes and Effects Analysis (FMEA):** This technique involves identifying and analyzing potential failures in a system and their effects.
- **Fault Tree Analysis (FTA):** This technique involves modeling the logical relationships between failures in a system.

By implementing a combination of development controls, peer reviews, and hazard analysis, organizations can help to improve the security of their software.

e. Conclusions (for Controls for Security)

- Developmental / OS / administrative controls help produce/maintain higher-quality (also more secure) s/w
- *Art* and science - no „silver bullet“ solutions
- „A good developer who truly understands security will incorporate security into all phases of development.“

[textbook, p. 172]

- Summary: [cf. B. Endicott-Popovsky]

Control	Purpose	Benefit
Develop- mental	Limit mistakes Make malicious code difficult	Produce better software
Operating System	Limit access to system	Promotes safe sharing of info
Adminis- trative	Limit actions of people	Improve usability, reusability and maintainability

Certainly, let's explain these terms in the context of prevention controls in network security:

Brain Virus (Prevention Control Example):

Brain Virus: The "Brain" virus, also known as the "Pakistani Brain," is one of the earliest computer viruses, dating back to 1986. It was designed to infect IBM PC-compatible computers.

Prevention Control Example: To prevent the spread of computer viruses like the Brain virus, organizations implement preventive measures such as antivirus software, email filtering, intrusion detection systems, and user education. Antivirus software is designed to detect and remove known viruses, including the Brain virus, from infected computers. Email filtering helps in identifying and quarantining malicious attachments, and intrusion detection systems can monitor network traffic for unusual behavior associated with virus activity.

Internet Worm (Prevention Control Example):

Internet Worm: An internet worm is a self-replicating program that spreads over the internet,

typically exploiting vulnerabilities in software to infect other devices. An example is the "ILOVEYOU" worm from 2000.

Prevention Control Example: To prevent internet worms, network administrators can use multiple preventive controls, including patch management to keep software up to date, intrusion prevention systems to detect and block worm-like behavior, and network segmentation to limit the worm's ability to propagate through the network. Firewalls and access controls can also restrict unauthorized access to network resources.

Web Bugs with Content (Prevention Control Example):

Web Bugs: Web bugs, also known as web beacons or pixel tags, are small pieces of code embedded in web pages or email content. They are used for tracking user activity and gathering information about user behavior.

Prevention Control Example: Preventing web bugs and their potential privacy and security implications involves using privacy-focused web browser extensions or settings that block tracking scripts, controlling email settings to prevent the automatic loading of images, and implementing strict content security policies on web servers to restrict the use of web bugs. These measures can help users and organizations protect their privacy and data from being collected by web bugs without their consent.

In network security, prevention controls aim to stop security threats before they can cause harm. They include a combination of technical solutions, policies, and user education to safeguard data, systems, and networks. The specific controls and measures implemented can vary based on the nature of the threat and the security requirements of the organization or individual.

UNIT 3

Authentication

- Process of reliability verifying the identity of someone
- **Can be defined as - Determining an identity to the required level of assurance**
- First step towards an cryptographic solution
- Idea is based on secrets
- Entity and authenticator share the same secret

Authentication Mechanisms

- Passwords
- Message digests of passwords
- Authentication Tokens
- Certificate-based Authentication
- Biometrics

Password Based Authentication

- Most common form of authentication
 - String of alphabets, numbers and special characters, which is supposed to be known only to the entity that is being authenticated
 - Supposed to be the simplest and the least expensive authentication mechanism ●
- Does not require any special hardware or software support

Problems

- Database contains passwords in clear text
- Password travels in clear text from the users computer to the server

Cryptographic Authentication Protocols

● Authentication Tokens

- Extremely useful alternative to a password
- Small device that generates a new random value every time
- This randomness is the basis for authentication
- Size of the devices are small key chains, calculators or credit cards

● Certificate-based Authentication

- User's certificate details need to be stored on the server-side
- CA distributes the certificates to the users also
- Validation between the two takes place at the time of authentication

● Biometrics

- Works on the basis of some human characteristics : finger prints, voice, iris etc.
- User database has a sample of user's biometric characteristics
- During authentication the user is required to provide another sample
- This is matched and the user validity is proved

Message Authentication

message authentication is concerned with:

- protecting the integrity of a message
- validating identity of originator
- non-repudiation of origin (dispute resolution)

then three alternative functions used:

- hash function
- message encryption
- message authentication code (MAC)

Message Security Requirements

- disclosure
- traffic analysis
- masquerade
- content modification
- sequence modification
- timing modification
- source repudiation
- destination repudiation

Authentication Functions

1. Symmetric Message Encryption

- a. encryption can also provides authentication
- b. if symmetric encryption is used then:
 - i. receiver know sender must have created it
 - ii. since only sender and receiver know key used
 - iii. know content cannot have been altered...
 - iv. ... if message has suitable structure, redundancy or a suitable checksum to detect any changes

2. Public-Key Message Encryption

- a. if public-key encryption is used:
 - i. encryption provides no confidence of sender
 - ii. since anyone potentially knows public-key
- b. however if
 - i. sender signs message using their private-key
 - ii. then encrypts with recipients public key
 - iii. have both secrecy and authentication
- c. again need to recognize corrupted messages
- d. but at cost of two public-key uses on message
- e. Dirty little detail on PKCS
 - i. Every time you encrypt, size expands
 - ii. Due to protections in PKCS#1
- f. So signing (by encryption) then encrypting, the size is more than doubled!

3. Message Authentication Code (MAC)

- a. generated by an algorithm that creates a small fixed-sized block
 - i. depending on both message and secret key
 - ii. like encryption though need not be reversible
- b. appended to message as a “signature”
- c. receiver performs same computation on message and checks it matches the MAC
- d. provides assurance that message is unaltered and comes from sender e. a small fixed-sized block of data
 - i. generated from message + secret key
 - ii. $MAC = C(K, M)$
- f. appended to message when sent

why use a MAC?

- sometimes only authentication is needed
- sometimes need authentication to persist longer than the encryption (e.g. archival use)
- note that a MAC is not a digital signature
- Does NOT provide non-repudiation
- as shown the MAC provides authentication
- can also use encryption for secrecy:
 - generally use separate keys for each
 - can compute MAC either before or after encryption
 - is generally regarded as better done before, but see Generic Composition

MAC Properties

- A MAC is a cryptographic checksum
- $MAC = CK(M)$
 - Condenses a variable-length message M
 - Using a secret key K
 - To a fixed-sized authenticator
- Is a many-to-one function
 - Potentially many messages have same MAC
 - But finding these needs to be very difficult

Security of MACs

like block ciphers have:

- brute-force attacks exploiting

- strong collision resistance hash have cost $2^{m/2}$

128-bit hash looks vulnerable, 160-bits better

MACs with known message-MAC pairs

- can either attack keyspace (cf. key search) or MAC
- at least 128-bit MAC is needed for security

cryptanalytic attacks exploit structure

- like block ciphers want brute-force attacks to be the best alternative
- more variety of MACs so harder to generalize about cryptanalysis

Hash Function

The hash value represents concisely the longer message may called the message digest

A message digest is as a "digital fingerprint" of the original document

Motivation for Hash Algorithms

Intuition

- Limitation on non-cryptographic checksum
- Very possible to construct a message that matches the checksum

Goal

- Design a code where the original message can not be inferred based on its checksum •

such that an accidental or intentional change to the message will change the hash value **Hash**

Function Applications

Used Alone

- Fingerprint -- file integrity verification, public key fingerprint
- Password storage (one-way encryption)

Combined with encryption functions

- Hash based Message Authentication Code (HMAC)
- protects both a message's integrity and confidentiality
- Digital signature
- Ensuring Non-repudiation
- Encrypt hash with private (signing) key and verify with public (verification) key

Hash and MAC Algorithms

Hash Functions

- condense arbitrary size message to fixed size
- by processing message in blocks
- through some compression function
- either custom or block cipher based

Message Authentication Code (MAC)

- fixed sized authenticator for some message
- to provide authentication for message
- by using block cipher mode or hash function

SHA-512

Step 1: Append padding bits

Step 2: Append length

Step 3: Initialize hash buffer

Step 4: Process the message in 1024-bit (128-word) blocks, which forms the heart of the algorithm

Step 5: Output the final state value as the resulting hash

SHA-512 is a complex algorithm, but it can be summarized in the following steps:

- The input message is padded to a multiple of 1024 bits.
- The padded message is broken down into 64-bit blocks.
- Each block is processed through a series of mathematical operations.
- The output of the mathematical operations is a 512-bit hash value.

SHA-512 Compression Function

- heart of the algorithm
- processing message in 1024-bit blocks
- consists of 80 rounds
- updating a 512-bit buffer
- using a 64-bit value W_t derived from the current message block

- and a round constant based on cube root of first 80 prime numbers

HMAC

- specified as Internet standard RFC2104
- uses hash function on the message:
- $HMAC = Hash[(K+ XOR opad) || Hash[(K+ XOR ipad)||M]]$
- where $K+$ is the key padded out to size
 - and opad, ipad are specified padding constants
 - overhead is just 3 more hash calculations than the message needs alone
 - any hash function can be used

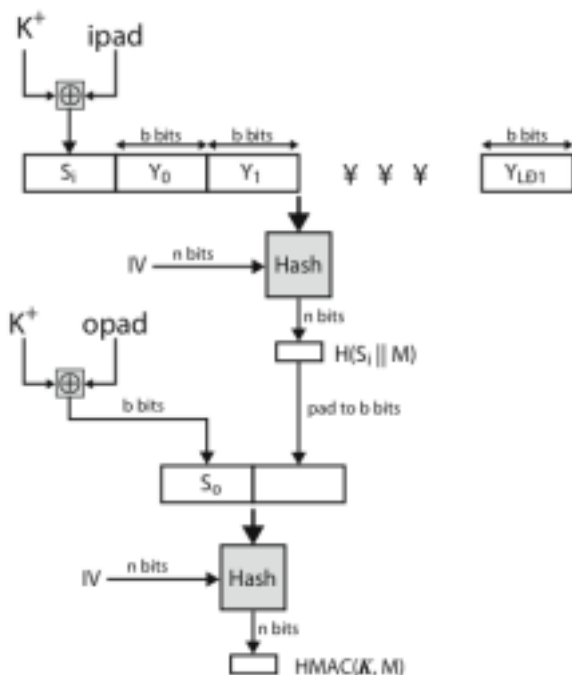


Figure 12.10 HMAC Structure

Authentication Applications

Developed to support application-level authentication

and digital signatures Most widely used services:

- Kerberos
- X.509

Kerberos – a private-key authentication service

- Developed as part of Project Athena at MIT
- Symmetric encryption
- using no public keys
- Provides centralised private-key third-party authentication in a distributed network

Kerberos Features

- Provide security in a distributed architecture consisting of dedicated user workstations (clients), and distributed or centralised servers
- Require the user to prove his identity for each service invoked
- Require that servers prove their identity to clients
- Secure, Reliable, Transparent, and Scalable

Working of kerberos

a Kerberos environment consists of:

- a Kerberos server
- a number of clients, all registered with server
- application servers, sharing keys with server

A Kerberos Realm

- Set of managed nodes that share the same Kerberos database

Multiple Kerber

- Kerberos server in each realm shares a secret key with one another
- There must be trust between the servers
- i.e. each server are registered with one another
- Does not scale well



X.509 – a public-key directory authentication service
Digital Certificate

- Digital certificates (DC) similar to passport
- It is simply a small computer file
- Establishes the relation between a user and their public key
- DC must contain the user name and the user's public key to prove that a particular value belongs to a particular user.
- Problem of man-in-the-middle attack was solved by using digital certificates

Digital Certificate Contents

- Main contents are the subject name (user), validity and public key
- Signed by a Certification Authority (CA)
- Provides guarantees about a user's identity
- No two digital certificates issued by the same user can have the same serial number

Working of dc

Digital signatures are a cryptographic technique that allow the sender of a digital message to prove their identity and the integrity of the message. Digital signatures work by using a pair of cryptographic keys: a private key and a public key. The private key is kept secret by the sender, and the public key is shared with the receiver.

To create a digital signature, the sender uses their private key to encrypt a hash of the message. The hash is a unique identifier for the message, and it is calculated using a cryptographic hash function. The encrypted hash is then appended to the message and sent to the receiver. To verify the digital signature, the receiver uses the sender's public key to decrypt the encrypted hash. If the decryption is successful, the receiver knows that the message has not been tampered with and that it was sent by the sender.

Authentication protocols are protocols that allow two or more parties to verify each other's identity. Authentication protocols are used in a wide variety of applications, such as secure communication, network access control, and electronic transactions.

There are many different types of authentication protocols, but they all work by exchanging some form of credentials between the parties.



unit 2 like 16 pages of notes left

unit 3 200 slides yayay

Unit 4 47 slides +40 slides +31 slides +10 slides





Wireless Security - wireless nets and wireless devices - host of problems

Problems

1. Channel: broadcast comm, eavesdropping + jamming, active attacks -> comm protocols vulnerabilities
2. Mobility: portable + mobile -> more risks
3. Resources -> sophisticated OS, limited memory + processing -> counter threats, DOS and malware
4. Accessibility: sensors + remotes -> hostile envs, physical attacks

THREATS

1. Accidental assoc. - wireless access points, connect to wrong one, breach is accidental, exposes resources of LAN
2. Malicious assoc- appear access point -> steal info
3. Ad hoc networks: peer-to-peer nets -> no access points between them
4. Non-trad net -> Bluetooth, barcode reader -> eavesdropping and spoofing
5. MAC spoofing - ID theft
6. Man in middle attack
7. DOS
8. Network injections

Security mechanisms

1. Secure wireless transmissions : signal hiding + encryption
2. Secure wireless access points - IEEE 802.1X network access control
3. Secure wireless networks: encryption, antivirus, anti-spyware, spoofing, router PS, check computer access

MOBILE DEVICE SEC

The provided text discusses the evolving landscape of network security in organizations, particularly in the context of mobile device use and the associated security threats. It also outlines the major security concerns related to mobile devices and the strategies to address these concerns. Let's break down the key points:

Changes in Network Security:

Traditionally, organizations had tightly controlled IT environments with Windows PCs, on-premises servers, and clear network perimeters.

Today, significant changes have occurred, including the proliferation of mobile devices, the use of cloud-based applications, the de-perimeterization of networks, and the need to accommodate external parties like guests and contractors.

Mobile Device Security Threats:

The text identifies several major security concerns related to mobile devices:

Lack of Physical Security Controls: Mobile devices are often under the user's control, and they can be moved between secure and non-secure locations. Theft and tampering are realistic threats.

Use of Untrusted Mobile Devices: Employees often use personal smartphones and tablets, which may not be fully trustworthy in terms of security.

Use of Untrusted Networks: Mobile devices may connect to untrusted networks, making them susceptible to eavesdropping and attacks.

Use of Applications Created by Unknown Parties: Third-party applications can pose security risks if not properly vetted.

Interaction with Other Systems: Data synchronization between devices and cloud-based storage can lead to unsecured data storage and malware risks.

Use of Untrusted Content: Mobile devices may access untrusted content, which could lead to security vulnerabilities.

Use of Location Services: GPS capabilities can disclose a user's physical location, which may be exploited by attackers.

Mobile Device Security Strategy:

The text outlines a strategy for addressing mobile device security concerns, categorized into three areas:

Device Security: This involves configuring and securing the mobile device itself, whether it's provided by the organization or is part of a bring-your-own-device (BYOD) policy. Measures include enabling PIN or password protection, avoiding auto-complete features, enabling remote wipe, and more.

Traffic Security: To protect data in transit, all traffic should be encrypted using secure protocols like SSL. Virtual private networks (VPNs) can be used to ensure secure communication. Strong authentication mechanisms should be used to control access.

Barrier Security: Network-level security includes firewall policies, intrusion detection, and intrusion prevention systems specifically tailored to mobile device traffic. These security mechanisms can restrict the scope of data and application access.

In summary, the text highlights the need for organizations to adapt their security strategies to account for the growing use of mobile devices and changing network dynamics. Security measures should address device security, traffic security, and network-level barrier security to mitigate the specific threats associated with mobile devices.

WIRELESS LAN SEC

Wireless local area network security (WLAN security) is a security system designed to protect networks from the security breaches to which wireless transmissions are susceptible. This type of security is necessary because WLAN signals have no physical boundary limitations, and are prone to illegitimate access over network resources, resulting in the vulnerability of private and confidential data. Network operations and availability can also be compromised in case of a WLAN security breach. To address these issues, various authentication, encryption, invisibility and other administrative controlling techniques are used in WLANs. Business and corporate WLANs in particular require adequate security measures to detect, prevent and block piggybackers, eavesdroppers and other intruders.

Some of the technologies employed in WLAN security include:

- **Wired Equivalent Privacy (WEP):** An old encryption standard used to overcome security threats. WEP provides security to WLAN by encrypting the information transmitted over the air so that only the receivers with the correct encryption key can decrypt the information.
- **WPA/WPA2 (Wi-Fi Protected Access):** Improved on WEP by introducing Temporal Key Integrity Protocol (TKIP). While still using RC4 encryption, TKIP uses a temporal encryption key that is regularly renewed, making it more difficult to steal. In addition, data integrity was improved through the use of a more robust hashing mechanism.

- Wireless Intrusion Prevention Systems/Intrusion Detection Systems: Intrusion detection and prevention focuses on radio frequency (RF) levels. This involves radio scanning to detect rogue access points or ad hoc networks to regulate network access. Advanced implementations are able to visually represent the network area along with potential threats, and have automatic classification capabilities so that threats can be easily identified.