CMR
INSTITUTE OF
TECHNOLOGY

USN										

Improvement Assesment Test

Sub:	NETWORK SECURITY		Sec	A & B				Code:	10EC832
Date:	21 / 05 / 2018	Duration:	90 mins	Max Marks:	50	Sem:	VIII	Branch:	TCE

1. **In RSA system it is given p=7, q=11, e=17, M=8. Find the cipher text C. Also find M from decryption. [10 marks]**

Step: 1: $pq = 77$

Step: 2: $\emptyset(pq) = (p-1)(q-1) = 6 \times 10 = 60$

Step: 3: $e = 17$

Step: 4: $d = e^{-1} mod\ 60 = 17^{-1} mod\ 60 = -7 mod 60 = 53$

q	r_1	r_2	r	t_1	t_2	t
3	60	17	9	0	1	-3
1	17	9	8	1	-3	4
1	9	8	1	-3	4	-7
8	8	1	0	4	-7	60
	1	0		-7	60	

Step: 5: $C = M^e mod77 = 8^{17} mod77 = 57$

$(17)_{10} = (10001)_2$

$1: 1 \times 8\ mod\ 77 = 8$

$0: 8 \times 8\ mod\ 77 = 64$

$0: 64 \times 64\ mod\ 77 = 15$

$0: 15 \times 15\ mod\ 77 = 71$

$1: 71 \times 71 \times 8\ mod\ 77 = 57$

Step: 6: $M = C^d mod77 = 57^{53} mod77$

$(53)_{10} = (110101)_2$

$1: 1 \times 57\ mod\ 77 = 57$

$1: 57 \times 57 \times 57\ mod\ 77 = 8$

$0: 8 \times 8\ mod\ 77 = 64$

$1: 64 \times 64 \times 57\ mod\ 77 = 8$

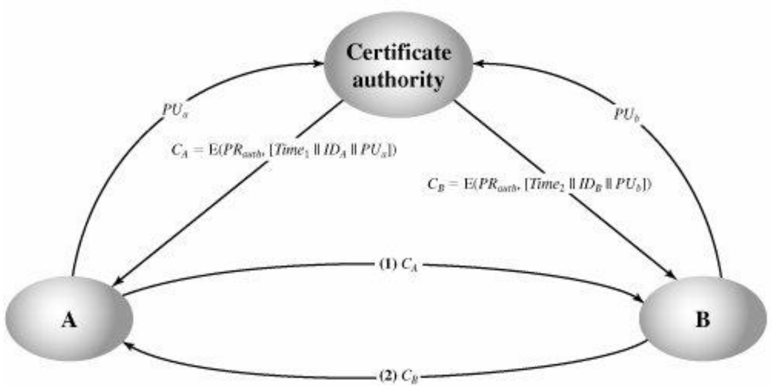
$0: 8 \times 8 \ mod \ 77 = 64$

$1: 64 \times 64 \times 57 \ mod \ 77 = 8$

2(a) Distinguish between conventional and public key encryption methods. [5 marks]

Conventional Encryption	Public key Encryption
Needed to Work:	Needed to Work:
1. The same algorithm with the same key is used for encryption and decryption	1. One algorithm is used for encryption and decryption with a pair of keys, one for encryption and one for decryption.
2. The sender and receiver must share the algorithm and the key.	2. The sender and receiver must each have one of the matched pair of keys (not the same one).
Needed for Security:	**Needed for Security:**
1. The key must be kept secret.	1. One of the two keys must be kept secret.
2. It must be impossible or at least impractical to decipher a message if no other information is available.	2. It must be impossible or at least impractical to decipher a message if no other information is available.
3. Knowledge of the algorithm plus samples of ciphertext must be insufficient to determine the key.	3. Knowledge of the algorithm plus one of the keys plus samples of ciphertext must be insufficient to determine the other key

2 (b) With the help of a block diagram, explain the process of public key exchange with the help of certificate authority. [5 marks]


Certificate authority
PU_a
PU_b
$C_A = E(PR_{auth}, [Time_1 \| ID_A \| PU_a])$
$C_B = E(PR_{auth}, [Time_2 \| ID_B \| PU_b])$
(1) C_A
(2) C_B
A
B

An alternative approach is to use **certificates** that can be used by participants to exchange keys without contacting a public-key authority, in a way that is as reliable as if the keys were obtained directly from a public-key authority. In essence, a certificate consists of a public key plus an identifier of the key owner, with the whole block signed by a trusted third party. Typically, the third party is a certificate authority, such as a government agency or a financial institution that is trusted by the user community. A user can present his or her public key to the authority in a secure manner, and obtain a certificate. The user can then publish the certificate. Anyone needed this user's public key can obtain the certificate and verify that it is valid by way of the attached trusted signature. A participant can also convey its key information to another by transmitting its certificate. Other participants can verify that the certificate was created by the authority. We can place the following requirements on this scheme:

1. Any participant can read a certificate to determine the name and public key of the certificate's owner.

2. Any participant can verify that the certificate originated from the certificate authority and is not counterfeit.
3. Only the certificate authority can create and update certificates.
4. Any participant can verify the currency of the certificate.

Application must be in person or by some form of secure authenticated communication. For participant A, the authority provides a certificate of the form **C_A = E(PRauth, [T||ID_A||PU_a])**

where PRauth is the private key used by the authority and T is a timestamp. A may then pass this certificate on to any other participant, who reads and verifies the certificate as follows:
D(PUauth, C_A) = D(PUauth, E(PRauth, [T||ID_A||PU_a])) = (T||ID_A||PU_a)

The recipient uses the authority's public key, PUauth to decrypt the certificate. Because the certificate is readable only using the authority's public key, this verifies that the certificate came from the certificate authority. The elements ID_A and PUa provide the recipient with the name and public key of the certificate's holder. The timestamp T validates the currency of the certificate. The timestamp counters the following scenario. A's private key is learned by an adversary. A generates a new private/public key pair and applies to the certificate authority for a new certificate. Meanwhile, the adversary replays the old certificate to B. If B then encrypts messages using the compromised old public key, the adversary can read those messages.

3(a) What requirement must a public key cryptosystem fulfill to be a secure algorithm?

Requirements for Public-Key Cryptography: [5 marks]

Public key cryptography depends on a cryptographic algorithm based on two related keys. Diffie and Hellman postulated this system without demonstrating that such algorithms exist. However, they did lay out the conditions that such algorithms must fulfill:

i) It is computationally easy for a party B to generate a pair (public key *PUb*, private key *PRb*).
ii) It is computationally easy for a sender A, knowing the public key and the message to be encrypted, *M*, to generate the corresponding ciphertext:
C = E(PUb, M)
iii) It is computationally easy for the receiver B to decrypt the resulting ciphertext using the private key to recover the original message:
M = D(PRb, C) = D[PRb, E(PUb, M)]
iv) It is computationally infeasible for an adversary, knowing the public key, *PUb*, to determine the private key, *PRb*.
v) It is computationally infeasible for an adversary, knowing the public key, *PUb*, and a ciphertext, *C*, to recover the original message, *M*.
vi) The two keys can be applied in either order:
M = D[PUb, E(PRb, M)] = D[PRb, E(PUb, M)]

These are formidable requirements, as evidenced by the fact that only a few algorithms (RSA, elliptic curve cryptography, Diffie-Hellman, DSS) have received widespread acceptance in the several decades since the concept of public-key cryptography was proposed.

(b) Users A and B use the Diffie-Hellman key exchange technique with a common prime q=11 and a primitive root α=5. [5 marks]

i. If user A has private key X_A=3, what is A's public key Y_A?
ii. If user B has private key X_B =2, what is B's public key Y_B?

What is the shared secret key K_A and K_B?

$\alpha = 5 \quad and \;\; q = 11$

i) If user A has private key $X_A = 3$ then the public key of A i.e. Y_A will be
$Y_A = (\alpha)^{X_A} mod\ q = (5)^3 mod\ 11 = 4$
ii) If user B has private key $X_B = 2$ then the public key of B i.e. Y_B will be

$Y_B = (\alpha)^{X_B} mod\ q = (5)^2 mod\ 11 = 3$

The secret key of A i.e K_A and the secret key of B is K_B will be same

$K_A = (Y_B)^{X_A} mod\ q = (3)^3 mod\ 11 = 5$

$Similarly\quad K_B = (Y_A)^{X_B} mod\ q = (4)^2 mod\ 11 = 5$

4.Discuss Deffie-Hellman key exchange algorithm. Explain how the algorithm is used to exchange secret key. [10 marks]

Diffie-Hellman Key Exchange Algorithm:

The purpose of the algorithm is to enable two users to securely exchange a key that can then be used for subsequent encryption of messages. The algorithm itself is limited to the exchange of secret values. The Diffie-Hellman algorithm depends for its effectiveness on the difficulty of computing discrete logarithms. Briefly, we can define the discrete logarithm in the following way. First, we define a primitive root of a prime number *p* as one whose powers modulo *p* generate all the integers from 1 to *p* 1. That is, if *a* is a primitive root of the prime number *p*, then the numbers *a* mod *p*, *a*2 mod *p*,..., *a*p1 mod *p* are distinct and consist of the integers from 1 through *p*- 1 in some permutation.

For any integer *b* and a primitive root *a* of prime number *p*, we can find a unique exponent *i* such that

$b \equiv a^i mod\ p\quad where\ 0 \leq i \leq (p-1)$

The exponent *i* is referred to as the discrete logarithm of *b* for the base *a*, mod *p*.

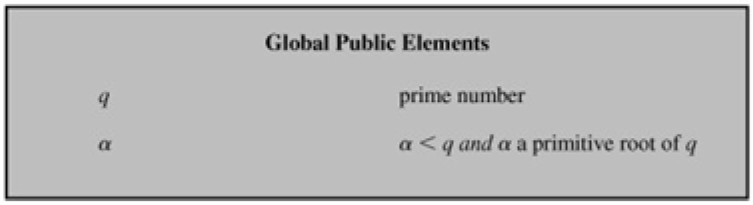
Algorithm:

there are two publicly known numbers: a prime number *q* and an integer that is a primitive root of *q* i.e. α

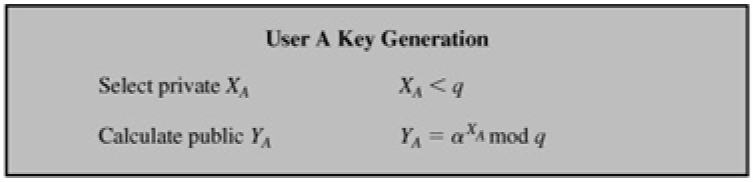
Suppose the users A and B wish to exchange a key. User A selects a random integer $X_A < q$ and computes $Y_A = \alpha^{XA}$ mod *q*. Similarly, user B independently selects a random integer $X_A < q$ and computes $Y_B = \alpha^{XB}$ mod *q*. Each side keeps the *X* value private and makes the *Y* value available publicly to the other side. User A computes the key as $K = (Y_B)^{XA}$ mod *q* and user B computes the key as $K = (Y_A)^{XB}$ mod *q*. These two calculations produce identical results:

$K = (Y_B)^{X_A} mod\ q\quad as\ \ Y_B = \alpha^{XB}\ Hence\ \ K = (\alpha^{X_B})^{X_A} mod\ q\qquad (User\ \ A)$

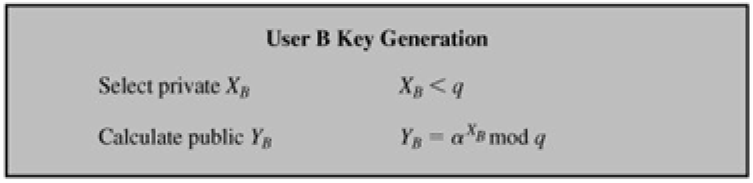
$K = (Y_A)^{X_B} mod\ q\quad as\ \ Y_A = \alpha^{XA}\ Hence\ \ K = (\alpha^{X_A})^{X_B} mod\ q\qquad (User\ \ B)$



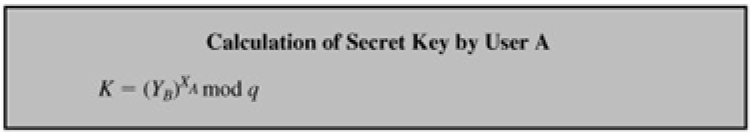
Global Public Elements	
q	prime number
α	$\alpha < q$ *and* α a primitive root of q



User A Key Generation	
Select private X_A	$X_A < q$
Calculate public Y_A	$Y_A = \alpha^{X_A} \bmod q$

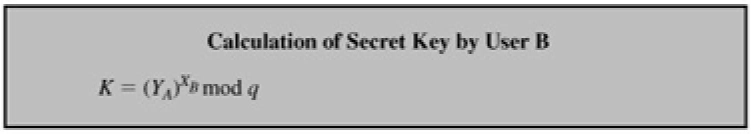


User B Key Generation	
Select private X_B	$X_B < q$
Calculate public Y_B	$Y_B = \alpha^{X_B} \bmod q$



Calculation of Secret Key by User A

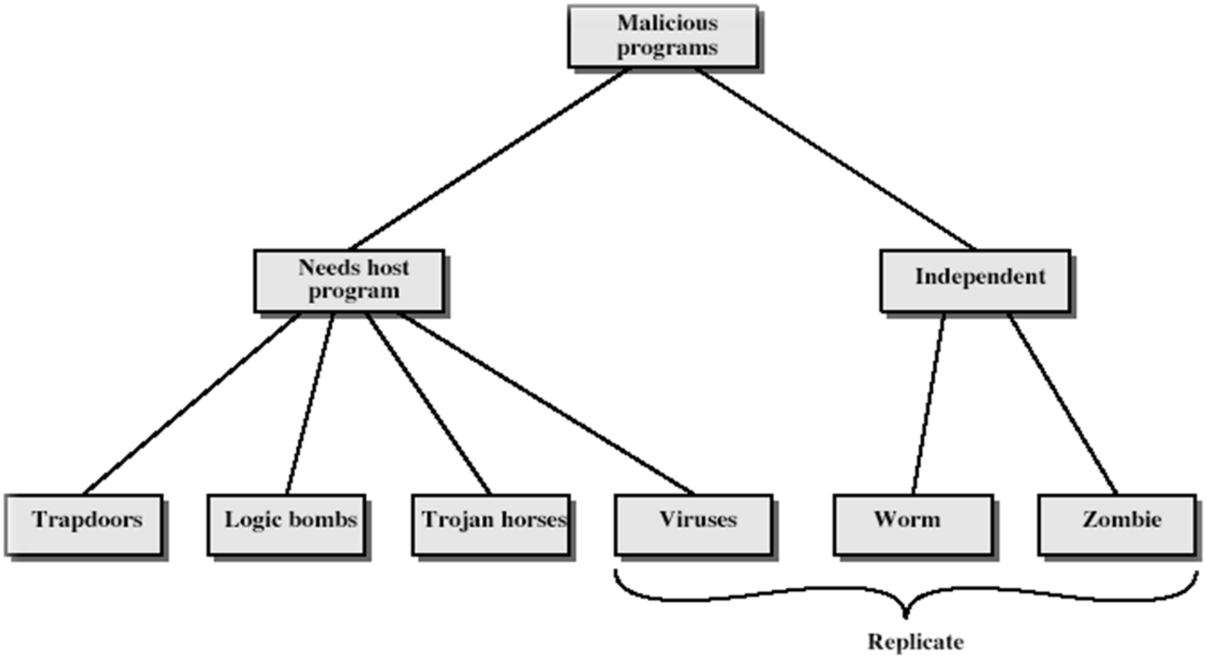
$K = (Y_B)^{X_A} \bmod q$



Calculation of Secret Key by User B

$K = (Y_A)^{X_B} \bmod q$

5. Give the taxonomy of malicious programs and explain them in brief. [10 marks]



Backdoor

A backdoor, also known as a trapdoor, is a secret entry point into a program that allows someone that is aware of the backdoor to gain access without going through the usual security access procedures. Programmers have used backdoors legitimately for many years to debug and test programs. This usually is done when the programmer is developing an application that has an authentication procedure, or a long setup, requiring the user to enter many different values to run the application. To debug the program, the developer may wish to gain special privileges or to avoid all the necessary setup and authentication. The programmer may also want to ensure that there is a method of activating the program should something be wrong with the authentication procedure that is being built into the application. The backdoor is code that recognizes some special sequence of input or is triggered by being run from a certain user ID or by an unlikely sequence of events. Backdoors become threats when unscrupulous programmers use them to gain unauthorized access.

Logic Bombs:

One of the oldest types of program threat, predating viruses and worms, is the logic bomb. The logic bomb is code embedded in some legitimate program that is set to "explode" when certain conditions are met. Examples of conditions that can be used as triggers for a logic bomb are the presence or absence of certain files, a particular day of the week or date, or a particular user running the application. Once triggered, a bomb may alter or delete data or entire files, cause a machine halt, or do some other damage.

Trojan horse

A Trojan horse is a useful, or apparently useful, program or command procedure containing hidden code that, when invoked, performs some unwanted or harmful function. Trojan horse programs can be used to accomplish functions indirectly that an unauthorized user could not accomplish directly. For example, to gain access to the files of another user on a shared system, a user could create a Trojan horse program that, when executed, changed the invoking user's file permissions so that the files are readable by any user. The author could then induce users to run the program by placing it in a common directory and naming it such that it appears to be a useful utility.

Viruses

A computer virus is a type of malware that propagates by inserting a copy of itself into and becoming part of another program. It spreads from one computer to another, leaving infections as it travels. Viruses can range in severity from causing mildly annoying effects to damaging data or software and causing denial-of-service (DoS) conditions. Almost all viruses are attached to an executable file, which means the virus may exist on a system but will not be active or able to spread until a user runs or opens the malicious host file or program. When the host code is executed, the viral code is executed as well. Normally, the host program keeps functioning after it is infected by the virus. However, some viruses overwrite other programs with copies of themselves, which destroys the host program altogether. Viruses spread when the software or document they are attached to is transferred from one computer to another using the network, a disk, file sharing, or infected e-mail attachments.

Worm

Computer worms are similar to viruses in that they replicate functional copies of themselves and can cause the same type of damage. In contrast to viruses, which require the spreading of an infected host file, worms are standalone software and do not require a host program or human help to propagate. To spread, worms either exploit a vulnerability on the target system or use some kind of social engineering to trick users into executing them. A worm enters a computer through a vulnerability in the system and takes advantage of file-transport or information-transport features on the system, allowing it to travel unaided.

Zombies

A zombie is a program that secretly takes over another Internet-attached computer and then uses that computer to launch attacks that are difficult to trace to the zombie's creator. Zombies are used in denialof- service attacks, typically against targeted Web sites. The zombie is planted on hundreds of computers belonging to unsuspecting third parties, and then used to overwhelm the target Web site by launching an overwhelming onslaught of Internet traffic.

6. Briefly describe, advanced antivirus techniques. [10 marks]

Advanced Antivirus Techniques

There are two advance antivirus Techniques:

i) **Generic Decryption**
ii) **Digital Immune System**

Generic Decryption

Generic decryption (GD) technology enables the antivirus program to easily detect even the most complex polymorphic viruses, while maintaining fast scanning speeds. Recall that when a file containing a polymorphic virus is executed, the virus must decrypt itself to activate. In order to detect such a structure, executable files are run through a GD scanner, which contains the following elements:

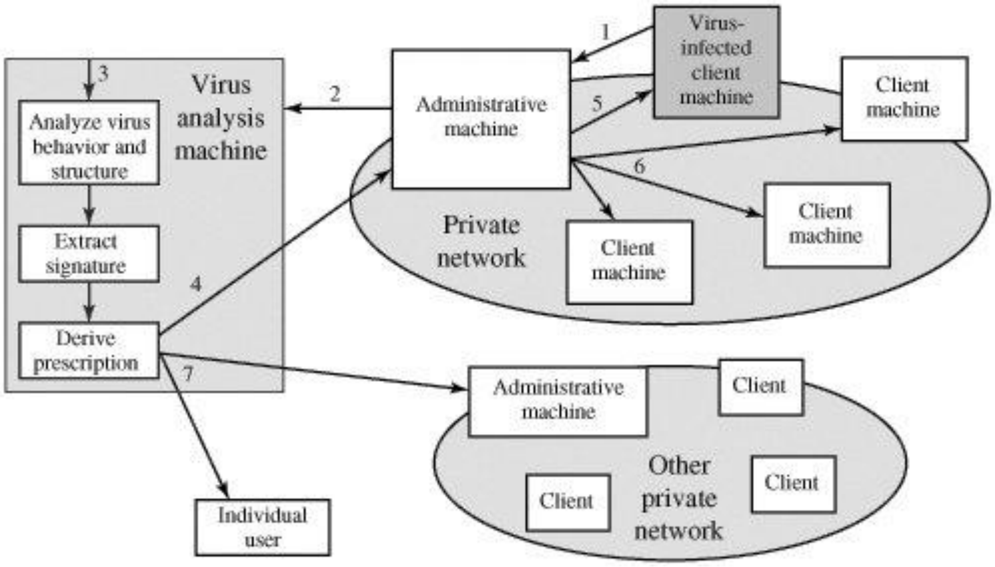
● **CPU emulator:** A software-based virtual computer. Instructions in an executable file are interpreted by the emulator rather than executed on the underlying processor. The emulator includes software versions of all registers and other processor hardware, so that the underlying processor is unaffected by programs interpreted on the emulator.

● **Virus signature scanner:** A module that scans the target code looking for known virus signatures.
● **Emulation control module:** Controls the execution of the target code.

Digital Immune System

The digital immune system is a comprehensive approach to virus protection developed by IBM. The motivation for this development has been the rising threat of Internet-based virus propagation. Traditionally, the virus threat was characterized by the relatively slow spread of new viruses. Two major trends in Internet technology have had an increasing impact on the rate of virus propagation in recent years:
● **Integrated mail systems:** Systems such as Lotus Notes and Microsoft Outlook make it very simple to send anything to anyone and to work with objects that are received.
● **Mobile-program systems:** Capabilities such as Java and ActiveX allow programs to move on their own from one system to another.

IBM has developed a prototype digital immune system. The objective of this system is to provide rapid response time so that viruses can be stamped out almost as soon as they are introduced. When a new virus enters an organization, the immune system automatically captures it, analyzes it, adds detection and shielding for it, removes it, and passes information about that virus to systems running IBM AntiVirus so that it can be detected before it is allowed to run elsewhere.



[Figure: Digital Immune System]

The typical steps in digital immune system operation:
1. A monitoring program on each PC uses a variety of heuristics based on system behavior, suspicious changes to programs, or family signature to infer that a virus may be present. The monitoring program forwards a copy of any program thought to be infected to an administrative machine within the organization.
2. The administrative machine encrypts the sample and sends it to a central virus analysis machine.
3. This machine creates an environment in which the infected program can be safely run for analysis. Techniques used for this purpose include emulation, or the creation of a protected environment within which the suspect program can be executed and monitored. The virus analysis machine then produces a prescription for identifying and removing the virus.
4. The resulting prescription is sent back to the administrative machine.
5. The administrative machine forwards the prescription to the infected client.
6. The prescription is also forwarded to other clients in the organization.

7. Subscribers around the world receive regular antivirus updates that protect them from the new virus.

7(a) Explain the typical phases of operation of a virus or worm. [5 marks]

Typical virus goes through the following four phases:

i) **Dormant phase:** The virus is idle. The virus will eventually be activated by some event, such asa date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage.

ii) **Propagation phase:** The virus places an identical copy of itself into other programs or into certain system areas on the disk. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase.

iii) **Triggering phase:** The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.

iv) **Execution phase:** The function is performed. The function may be harmless, such as a message on the screen, or damaging, such as the destruction of programs and data files.

7 (b) in general terms, how does a worm propagate? [5 marks]

A worm is a program that can replicate itself and send copies from computer to computer across network connections. Upon arrival, the worm may be activated to replicate and propagate again. In addition to propagation, the worm usually performs some unwanted function. A worm actively seeks out more machines to infect and each machine that is infected serves as an automated launching pad for attacks on other machines. The new copy of the worm program is then run on the remote system where, in addition to any functions that it performs at that system, it continues to spread in the same fashion. A network worm exhibits the same characteristics as a computer virus: a dormant phase, a propagation phase, a triggering phase, and an execution phase. The propagation phase generally performs the following functions:

1. Search for other systems to infect by examining host tables or similar repositories of remote system addresses.

2. Establish a connection with a remote system.

3. Copy itself to the remote system and cause the copy to be run.

The network worm may also attempt to determine whether a system has previously been infected before copying itself to the system. In a multiprogramming system, it may also disguise its presence by naming itself as a system process or using some other name that may not be noticed by a system operator. As with viruses, network worms are difficult to counter.