

USN

--	--	--	--	--	--	--	--	--	--



Internal Assessment Test 3 – June 2022

Sub:	Network Security				Sub Code:	18EC821	Branch:	ECE
Date:	18-06-22	Duration:	90 min's	Max Marks:	50	Sem / Sec:	8 – A, B, C, D	OBE

Answer any FIVE FULL Questions							MARKS	CO	RBT
1.	With neat diagram explain Architecture for Distributed Intrusion Detection.					[10]	CO4	L1	

Sol Distributed Intrusion Detection

Until recently, work on intrusion detection systems focused on single-system standalone facilities. The typical organization, however, needs to defend a distributed collection of hosts supported by a LAN or internetwork. Although it is possible to mount a defense by using stand-alone intrusion detection systems on each host, a more effective defense can be achieved by coordination and cooperation among intrusion detection systems across the network.

Porras points out the following major issues in the design of a distributed intrusion detection system [PORR92]:

- A distributed intrusion detection system may need to deal with different audit record formats. In a heterogeneous environment, different systems will employ different native audit collection systems and, if using intrusion detection, may employ different formats for security-related audit records.
- One or more nodes in the network will serve as collection and analysis points for the data from the systems on the network. Thus, either raw audit data or summary data must be transmitted across the network. Therefore, there is a requirement to assure the integrity and confidentiality of these data. Integrity is required to prevent an intruder from masking his or her activities by altering the transmitted audit information. Confidentiality is required because the transmitted audit information could be valuable.
- Either a centralized or decentralized architecture can be used. With a centralized architecture, there is a single central point of collection and analysis of all audit data. This eases the task of correlating incoming reports but creates a potential bottleneck and single point of failure. With a decentralized architecture, there are more than one analysis centers, but these must coordinate their activities and exchange information.

Architecture for Distributed Intrusion Detection

2.	Compare statistical anomaly detection and rule-based intrusion detection.					[10]	CO4	L1
----	---	--	--	--	--	------	-----	----

Sol Statistical Anomaly Detection

As was mentioned, statistical anomaly detection techniques fall into two broad categories: threshold detection and profile-based systems. Threshold detection involves counting the number of occurrences of a specific event

type over an interval of time. If the count surpasses what is considered a reasonable number that one might expect to occur, then intrusion is assumed. Threshold analysis, by itself, is a crude and ineffective detector of even moderately sophisticated attacks. Both the threshold and the time interval must be determined.

Because of the variability across users, such thresholds are likely to generate either a lot of false positives or a lot of false negatives. However, simple threshold detectors may be useful in conjunction with more sophisticated techniques. Profile-based anomaly detection focuses on characterizing the past behavior of individual users or related groups of users and then detecting significant deviations.

A profile may consist of a set of parameters, so that deviation on just a single parameter may not be sufficient in itself to signal an alert. The foundation of this approach is an analysis of audit records. The audit records provide input to the intrusion detection function in two ways. First, the designer must decide on a number of quantitative metrics that can be used to measure user behavior. An analysis of audit records over a period of time can be used to determine the activity profile of the average user. Thus, the audit records serve to define typical behavior. Second, current audit records are the input used to detect intrusion. That is, the intrusion detection model analyzes incoming audit records to determine deviation from average behavior.

Examples of metrics that are useful for profile-based intrusion detection are the following:

- Counter: A nonnegative integer that may be incremented but not decremented until it is reset by management action. Typically, a count of certain event types is kept over a particular period of time. Examples include the number of logins by a single user during an hour, the number of times a given command is executed during a single user session, and the number of password failures during a minute.
- Gauge: A nonnegative integer that may be incremented or decremented. Typically, a gauge is used to measure the current value of some entity. Examples include the number of logical connections assigned to a user application and the number of outgoing messages queued for a user process.
- Interval timer: The length of time between two related events. An example is the length of time between successive logins to an account.
- Resource utilization: Quantity of resources consumed during a specified period. Examples include the number of pages printed during a user session and total time consumed by a program execution.

Given these general metrics, various tests can be performed to determine whether current activity fits within acceptable limits. [DENN87] lists the following approaches that may be taken:

- Mean and standard deviation
- Multivariate
- Markov process
- Time series
- Operational

Rule-Based Intrusion Detection

Rule-based techniques detect intrusion by observing events in the system and applying a set of rules that lead to a decision regarding whether a given pattern of activity is or is not suspicious. In very general terms, we can characterize all approaches as focusing on either anomaly detection or penetration identification, although there is some overlap in these approaches.

Rule-based anomaly detection is similar in terms of its approach and strengths to statistical anomaly detection. With the rule-based approach, historical audit records are analyzed to identify usage patterns and to generate automatically rules that describe those patterns. Rules may represent past behavior patterns of users, programs,

privileges, time slots, terminals, and so on. Current behavior is then observed, and each transaction is matched against the set of rules to determine if it conforms to any historically observed pattern of behavior.

As with statistical anomaly detection, rule-based anomaly detection does not require knowledge of security vulnerabilities within the system. Rather, the scheme is based on observing past behavior and, in effect, assuming that the future will be like the past. In order for this approach to be effective, a rather large database of rules will be needed. For example, a scheme described in [VACC89] contains anywhere from 10^4 to 10^6 rules.

	3. Explain in detail the software threats and list out types of virus in detail.	[10]	CO4	L2
Sol	<p>Malicious software can be divided into two categories: those that need a host program, and those that are independent. The former, referred to as parasitic, are essentially fragments of programs that cannot exist independently of some actual application program, utility, or system program. Viruses, logic bombs, and backdoors are examples. Independent malware is a self-contained program that can be scheduled and run by the operating system. Worms and bot programs are examples.</p> <p>Virus: Malware that, when executed, tries to replicate itself into other executable code; when it succeeds the code is said to be infected. When the infected code is executed, the virus also executes.</p> <p>Worm A computer program that can run independently and can propagate a complete working version of itself onto other hosts on a network.</p> <p>Logic bomb A program inserted into software by an intruder. A logic bomb lies dormant until a predefined condition is met; the program then triggers an unauthorized act.</p> <p>Trojan horse A computer program that appears to have a useful function, but also has a hidden and potentially malicious function that evades security mechanisms, sometimes by exploiting legitimate authorizations of a system entity that invokes the Trojan horse program.</p> <p>Backdoor (trapdoor): Any mechanism that bypasses a normal security check; it may allow unauthorized access to functionality. Mobile code Software (e.g., script, macro, or other portable instruction) that can be shipped unchanged to a heterogeneous collection of platforms and execute with identical semantics. Exploits Code specific to a single vulnerability or set of vulnerabilities.</p> <p>Downloaders Program that installs other items on a machine that is under attack. Usually, a downloader is sent in an e-mail. Auto-rooter Malicious hacker tools used to break into new machines remotely.</p> <p>Kit (virus generator): Set of tools for generating new viruses automatically.</p> <p>Spammer programs: Used to send large volumes of unwanted e-mail.</p> <p>Flooders: Used to attack networked computer systems with a large volume of traffic to carry out a denial-of-service (DoS) attack.</p> <p>Keyloggers Captures keystrokes on a compromised system.</p> <p>Rootkit Set of hacker tools used after attacker has broken into a computer system and gained root-level access.</p> <p>Zombie, bot Program activated on an infected machine that is activated to launch attacks on other machines.</p> <p>Spyware Software that collects information from a computer and transmits it to another system.</p>			

	Adware Advertising that is integrated into software. It can result in pop-up ads or redirection of a browser to a commercial site.			
4.	List the techniques used by firewalls to control access and enforce security policy in detail.	[10]	CO5	L1
Sol	<p>Originally, firewalls focused primarily on service control, but they have since evolved to provide all four:</p> <ul style="list-style-type: none"> • Service control: Determines the types of Internet services that can be accessed, inbound or outbound. The firewall may filter traffic on the basis of IP address, protocol, or port number; may provide proxy software that receives and interprets each service request before passing it on; or may host the server software itself, such as a Web or mail service. • Direction control: Determines the direction in which particular service requests may be initiated and allowed to flow through the firewall. User control: Controls access to a service according to which user is attempting to access it. This feature is typically applied to users inside the firewall perimeter (local users). It may also be applied to incoming traffic from external users; the latter requires some form of secure authentication technology, such as is provided in IPsec. Behavior control: Controls how particular services are used. For example, the firewall may filter e-mail to eliminate spam, or it may enable external access to only a portion of the information on a local Web server. 			
5.	Explain the phases of malicious programs in detail.	[10]	CO5	L2
Sol	<p>Infection mechanism: The means by which a virus spreads, enabling it to replicate. The mechanism is also referred to as the infection vector.</p> <ul style="list-style-type: none"> • Trigger: The event or condition that determines when the payload is activated or delivered. • Payload: What the virus does, besides spreading. The payload may involve damage or may involve benign but noticeable activity. During its lifetime, a typical virus goes through the following four phases: <ul style="list-style-type: none"> • Dormant phase: The virus is idle. The virus will eventually be activated by some event, such as a date, the presence of another program or file, or the capacity of the disk exceeding some limit. Not all viruses have this stage. • Propagation phase: The virus places a copy of itself into other programs or into certain system areas on the disk. The copy may not be identical to the propagating version; viruses often morph to evade detection. Each infected program will now contain a clone of the virus, which will itself enter a propagation phase. • 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. • 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. Most viruses carry out their work in a manner that is specific to a particular operating system and, in some cases, specific to a particular hardware platform. Thus, they are designed to take advantage of the details and weaknesses of particular systems. 			
6.	Explain what a firewall is, highlighting the capabilities and limitations.	[10]	CO5	L2
Sol	<p>The following capabilities are within the scope of a firewall:</p> <ol style="list-style-type: none"> 1. A firewall defines a single choke point that keeps unauthorized users out of the protected network, prohibits potentially vulnerable services from entering or leaving the network, and provides protection from various kinds of IP spoofing and routing attacks. The use of a single choke point simplifies security management because security capabilities are consolidated on a single system or set of systems. 2. A firewall provides a location for monitoring security-related events. Audits and alarms can be implemented on the firewall system. 3. A firewall is a convenient platform for several Internet functions that are not security related. These include a network address translator, which maps local addresses to Internet addresses, and a network management function that audits or logs Internet usage. 			

4. A firewall can serve as the platform for IPsec. Using the tunnel mode capability described in Chapter 19, the firewall can be used to implement virtual private networks.

Firewalls have their limitations, including the following:

1. The firewall cannot protect against attacks that bypass the firewall. Internal systems may have dial-out capability to connect to an ISP. An internal LAN may support a modem pool that provides dial-in capability for traveling employees and telecommuters.

2. The firewall may not protect fully against internal threats, such as a disgruntled employee or an employee who unwittingly cooperates with an external attacker.

3. An improperly secured wireless LAN may be accessed from outside the organization. An internal firewall that separates portions of an enterprise network cannot guard against wireless communications between local systems on different sides of the internal firewall.

4. A laptop, PDA, or portable storage device may be used and infected outside the corporate network, and then attached and used internally.

7. Explain with a necessary diagrams the digital immune system

[10]

CO4

L2

DIGITAL IMMUNE SYSTEM The digital immune system is a comprehensive approach to virus protection developed by IBM and subsequently refined by Symantec [SYMAOI]. The motivation for this development has been the rising threat of Internet-based virus propagation. We first say a few words about this threat and then summarize IBM's approach.

Traditionally, the virus threat was characterized by the relatively slow spread of new viruses and new mutations. Antivirus software was typically updated on a monthly basis, and this was sufficient to control the problem. Also traditionally, the Internet played a comparatively small role in the spread of viruses. But as [CHES97] points out, 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. In response to the threat posed by these Internet-based capabilities, IBM has developed a prototype digital immune system. This system expands on the use of program emulation discussed in the preceding subsection and provides a general purpose emulation and virus-detection 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 illustrates 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.

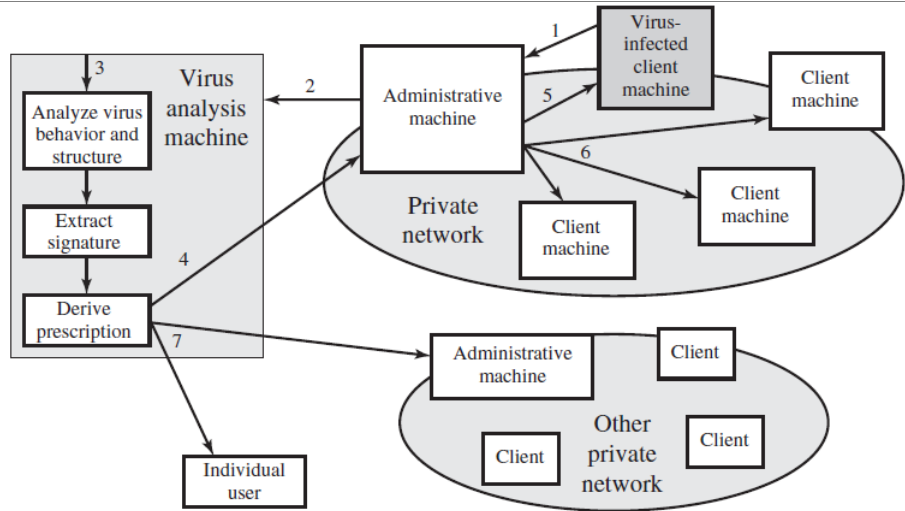
3. This machine creates an environment in which the infected program can be safely run for analysis. Techniques used for this purpose include emulation, or the creation of a protected environment within which the suspect program can be executed and monitored. The virus analysis machine then produces 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.



Digital Immune System