|  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Makeup Examination-Sept.2023****V SEMESTER DIPLOMA EXAMINATION**  **SCHEME OF VALUATION** | | | | | | | | | | |
| **CYBER SECURITY** | | | | | | | | | | |
| **SUB CODE : 20CS54I, V SEM** | | | | | | | | | | |
| **Q. no** | | **DESCRIPTION** | | | | | | | **Marks distribution** | **Total marks** |
| **SECTION I** | | | | | | | | | | |
| 1 | a | Firewall+IPS | | | | | | | 5+5 | 10 |
| b | Different methods+Explanation(Minimum 5 methods) | | | | | | | 5+5 | 10 |
| **OR** | | | | | | | | | | |
| 2 | a | Explanation of any 5 | | | | | | | 5\*2 | 10 |
| b | Layers + Vulnerabilities | | | | | | | 5+5 | 10 |
| **SECTION II** | | | | | | | | | | |
| 3 | a | Calculating decryption key + Calculating n(φ) | | | | | | | 5+5 | 10 |
| b | Euler Totient of each carries 2.5+ Finding GCD of each carries 2.5 | | | | | | | 5+5 | 10 |
| **OR** | | | | | | | | | | |
| 4 | a | Using Algorithm +Finding secret Key | | | | | | | 3+7 | 10 |
| b | MAC+Digital Signature | | | | | | | 5+5 | 10 |
| **SECTION III** | | | | | | | | | | |
| 5 | a | List types of Hardening+ Explanation | | | | | | | 5+5 | 10 |
| b | Third party patch Management+Autopilot | | | | | | | 5+5 | 10 |
| **OR** | | | | | | | | | | |
| 6 | a | Explanation of any 5 types of hardnening | | | | | | | 5\*2 | 10 |
| b | Difinition+Any four Models | | | | | | | 2+8 | 10 |
| **SECTION IV** | | | | | | | | | | |
| 7 | a | Any 5 Differentiations | | | | | | | 5\*2 | 10 |
| b | Web Vulnerabilities | | | | | | | 1\*10 | 10 |
| **OR** | | | | | | | | | | |
| **8** | a | Explanation of any 5 best Practices | | | | | | | 5\*2 | 10 |
| b | Cloud Vulnerabilities | | | | | | | 5\*2 | 10 |
| **SECTION V** | | | | | | | | | | |
| 9 | a | Vulnerability Life cycle+Explain two types of Vulnerabilities | | | | | | | 6+4 | 10 |
| b | Diagram + explanation | | | | | | | 4+6 | 10 |
| **OR** | | | | | | | | | | |
| 10 | a | GRC Explanation+Benefits | | | | | | | 5+5 | 10 |
| b | Definition +explanation+Challenges | | | | | | | 2+3+5 | 10 |
|  |  |  |  |  |  |  |  |  |  |  | |  |
|  |  |  |  |  |  |  |  |  |  |  | |  |
|  |  |  |  |  |  |  |  |  | | |
|  |  |  |  |  |  |  |  |  |  |  | |  |
|  |  |  |  |  |  |  |  |  |  |  | |  |

## **Makeup Examination-Sept.2023**

## **V semester Diploma Examination**

**CYBER SECURITY 20CS54I**

**Model answers**

**NOTE: All model answers are general specific to subject, if any answers are relevant**

**please give marks.**

**SECTION-I**

**1** a. Describe how the following concepts secure our Network a). Firewall b). IPS. 10M

**a). Firewall**

A firewall is a network security system that monitors and controls incoming and outgoing network traffic based on predetermined security rules. It acts as a barrier between a trusted internal network and untrusted external network, such as the Internet.

A firewall is one of the first lines of defence in preventing cyber-attacks. Naturally, this presents an opportunity for penetration testers and threat actors alike, to attempt exploits that would compromise a network’s security.

The firewall checks each incoming and outgoing network packet and compares it against its set of security rules to determine whether to block or allow the packet to pass through to its intended destination. The main purpose of a firewall is to prevent unauthorized access to or from a private network while allowing authorized communications to pass through.



**Advantages of Using Firewalls**

● It provides enhanced security and privacy from vulnerable services. It prevents unauthorized users from accessing a private network that is connected to the internet.

● Firewalls provide faster response time and can handle more traffic loads.

● A firewall allows you to easily handle and update the security protocols from a single authorized device.

● It safeguards your network from phishing attacks

**b)IPS**

An Intrusion Prevention System (IPS) is a type of network security system that monitors network traffic and actively blocks or stops malicious activity. Unlike an Intrusion Detection System (IDS), which simply alerts administrators of a security breach, an IPS takes proactive measures to prevent intrusions from occurring in the first place.

An IPS typically operates by analysing network traffic in real-time and comparing it against a set of predefined security rules or signatures. If the system detects a potential threat, it will take immediate action to prevent it from causing harm to the network or its systems. This can involve blocking the offending traffic, resetting the connection, or even isolating the affected system from the network. They must be powerful enough to scan a high volume of traffic without slowing down network performance.

In computer science, IPS is considered a crucial component of an overall security strategy, as it provides an additional layer of protection against threats that traditional firewalls may not detect or be able to prevent.



**1b** Write about different methods of infiltration. 10M

Cybercriminals are constantly finding new and innovative ways to infiltrate company networks and steal sensitive data.

**1.Social engineering:** Social engineering refers to the manipulation of individuals to steal sensitive information or perform actions that compromise a company's security.

* **Pretexting** This is when an attacker calls an individual and lies to them in an attempt to gain access to privileged data. For example, pretending to need a person ‘s personal or financial data in order to confirm their identity.
* **Tailgating** This is when an attacker quickly follows an authorized person into a secure, physical location. Where an attacker follows an unaware user to gain access to an area without authorization.
* **Something for something (quid pro quo)** This is when an attacker requests personal information from a person in exchange for something, like a free gift.
* **Denial-of-Service** Denial-of-Service (DoS) attacks are a type of network attack that is relatively simple to carry out, even by an unskilled attacker. A DoS attack results in some sort of interruption of network service to users, devices or applications.
* **Distributed DoS** A Distributed DoS (DDoS) attack is similar to a DoS attack but originates from multiple, coordinated sources
* **Man in The Middle:** A MitM attack happens when a cybercriminal takes control of a device without the user‘s knowledge. With this level of access, an attacker can intercept and capture user information before it is sent to its intended destination. These types of attacks are often used to steal financial information. There are many types of malware that possess MitM attack capabilities.

**2.Password attacks:** Weak passwords are a common vulnerability exploited by hackers.

Entering a username and password is one of the most popular forms of authenticating to a web site. Therefore, uncovering your password is an easy way for cybercriminals to gain access to your most valuable information. Some of the common password security attacks:

* Password Spraying: This technique attempts to gain access to a system by spraying‘ a few commonly used passwords across a large number of accounts.
* Dictionary Attacks: A hacker systematically tries every word in a dictionary or a list of commonly used words as a password in an attempt to break into a password protected account.
* Brute Force Attacks: The simplest and most commonly used way of gaining access to a password-protected site, brute-force attacks see an attacker using all possible combinations of letters, numbers and symbols in the password space until they get it right.

**3.Software vulnerabilities:** Hackers can exploit vulnerabilities in software and applications to gain unauthorized access to a system or to install malware.

**4.Physical attacks:** Physical attacks refer to the hacking of company systems through physical means like stealing a laptop or connecting to a network port

**5.Phishing attacks:** Phishing attacks are a common tactic used by hackers to gain access to an organization's network. They usually send emails or messages that appear to come from a trusted source, like a bank, to trick employees into revealing sensitive information.

**6.Malware:** Malware is another common way hackers infiltrate organizations. Malware can come in various forms, including viruses, Trojans, and ransomware.

**OR**

2 a. List and Explain different types of Malwares 10M

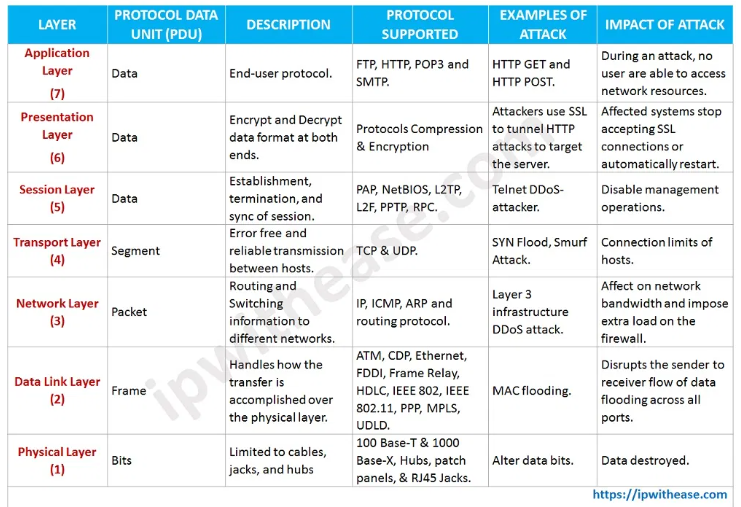
Malware, or malicious software, is any program or file that is intentionally harmful to a computer, network or server. Malware can infect networks and devices and is designed to harm those devices, networks and/or their users in some way.

Types of malware include the following:

* A [virus](https://www.techtarget.com/searchsecurity/definition/virus) is the most common type of malware that can execute itself and spread by infecting other programs or files.
* A [worm](https://www.techtarget.com/searchsecurity/definition/worm) can self-replicate without a host program and typically spreads without any interaction from the malware authors.
* A [Trojan horse](https://www.techtarget.com/searchsecurity/definition/Trojan-horse) is designed to appear as a legitimate software program to gain access to a system. Once activated following installation, Trojans can execute their malicious functions.
* [Spyware](https://www.techtarget.com/searchsecurity/definition/spyware) collects information and data on the device and user, as well as observes the user's activity without their knowledge.
* Ransomware infects a user's system and encrypts its data. Cybercriminals then demand a ransom payment from the victim in exchange for decrypting the system's data.
* A [rootkit](https://www.techtarget.com/searchsecurity/definition/rootkit) obtains administrator-level access to the victim's system. Once installed, the program gives threat actors root or privileged access to the system.
* A [backdoor](https://www.techtarget.com/searchsecurity/definition/back-door) virus or remote access Trojan ([RAT](https://www.techtarget.com/searchsecurity/definition/RAT-remote-access-Trojan)) secretly creates a backdoor into an infected computer system that enables threat actors to remotely access it without alerting the user or the system's security programs.
* [Adware](https://www.techtarget.com/searchsecurity/definition/adware) tracks a user's browser and download history with the intent to display pop-up or banner advertisements that lure the user into making a purchase. For example, an advertiser might use cookies to track the webpages a user visits to better target advertising.
* [Keyloggers](https://www.techtarget.com/searchsecurity/definition/keylogger), also called system monitors, track nearly everything a user does on their computer. This includes emails, opened webpages, programs and keystrokes.

**2b** .Illustrate different Vulnerabilities in 7 layers of OSI model. 10M

Attacks can happen at different levels of the network models with different protocols.



**1. Sniffing (physical)**

Back in the day when there was no remote home phone, you had to have multiple phones at home anyone can remember the problems that had been occurred when two people wants to speak on the phone someone else could grab another phone and hear the speeches. in the context of network security when packets are not encrypted someone else could intrude to your network and steal those packets with some sniffer applications that work on the **physical**layer of **OSI model** like Wireshark, Tcpdump, WinDump. some protocols that work on this layer and can be sniffed are:

Examples of protocols that use physical layers include:

* Digital Subscriber Line.
* Integrated Services Digital Network.
* Infrared Data Association.
* Universal Serial Bus (USB.)
* Bluetooth.
* Ethernet.

**2.SPOOFING (Data Link)**

Spoofing is the act of a person or a program that successfully identifies itself which is from an unknown source as being from a known, trusted source. Spoofing can apply to emails, phone calls, and websites, or can be more technical, such as a computer spoofing an IP address, Address Resolution Protocol (ARP), or Domain Name System (DNS) server.

IP spoofing and ARP spoofing, in particular, may be used to leverage man-in-the-middle attacks against hosts on a computer network. Spoofing attacks that take advantage of TCP/IP suite protocols may be mitigated with the use of firewalls capable of deep packet inspection or by taking measures to verify the identity of the sender or recipient of a message.

**3.man-in-the-middle (Network)**

Many of the protocols in the TCP/IP suite do not provide mechanisms for authenticating the source or destination of a message, leaving them vulnerable cause an attacker secretly relays and possibly alters the communications between two parties who believe that they are directly communicating with each other.

**4.Reconnaissance (Transport)**

In the context of cybersecurity, reconnaissance is the practice of discovering and collecting information about a system. One of the most common techniques involved with reconnaissance is port scanning, which sends data to various TCP and UDP (user datagram protocol) ports on a device and evaluates the response. Some common examples of reconnaissance attacks include [packet sniffing](https://www.techopedia.com/definition/4113/sniffer), [ping sweeping, port scanning](https://www.tripwire.com/state-of-security/featured/common-basic-port-scanning-techniques/), [phishing](https://www.tripwire.com/state-of-security/security-awareness/6-common-phishing-attacks-and-how-to-protect-against-them/), [social engineering](https://www.tripwire.com/state-of-security/security-awareness/psychological-tricks-malware-trade/), and internet information queries.

**5.Hijacking (Session)**

Sometimes also known as **cookie hijacking** is the exploitation of a valid computer session to gain unauthorized access to information or services in a computer system. In particular, it is used to refer to the theft of a [magic cookie](https://en.wikipedia.org/wiki/Magic_cookie) used to authenticate a user to a remote server. It has particular relevance to web developers, as the HTTP cookies used to maintain a session on many websites can be easily stolen by an attacker using an intermediary computer or with access to the saved cookies on the victim’s computer. these explosions can be carried out by these attacks

* 1. **Cross-site scripting: XSS attacks**enable attackers to inject client-side scripts into web pages. It causes running codes, which is treated as trustworthy because it appears to belong to the server, on the victim computer. It allows the attacker to obtain a copy of the cookie or perform other operations.
  2. **Session side jacking:**where the attacker uses packet sniffing to read network traffic between two parties to steal the session cookie.
  3. **Malware** and unwanted programs can use browser hijacking to steal a browser’s cookie files without a user’s knowledge.

**6.Phishing (presentation)**

Phishing attacks are the practice of sending fraudulent messages that appear to come from a trusted source. It is usually performed through email. The goal is to steal sensitive data like credit card and login information or install malware on the victim’s machine. Phishing is a common type of cyber-attack that everyone should learn about in order to protect themselves.

**7.Exploit (Application)**

An exploit is a program that takes advantage of a bug or vulnerability in other systems. the cause vulnerability may be due to bad system configuration or a bug in a specific version of software installed on the victim system. Many exploits are designed to provide super user-level access to a victim system or are designed to cause DoS (denial of service) or [DDoS](https://www.avast.com/c-ddos" \t "_blank) (distributed denial of service) attacks, in which attackers can bring down a website or critical system without even using an exploit.

**SECTION-II**

**3a** Explain RSA algorithm. Using RSA solve p=13, q=17, e=35, d=? 10M

Find the product of two number

N=p\*q

N=13\*17

N=221

CALULATE n(φ)=(p-1)(q-1)

n(φ)= (13-1)(17-1)

n(φ)= 12\*16

n(φ)=192

Encryption Key e=35

Substitute the value for i till we get whole number for d from i=1,2,3…..etc

d = ((φ (n)\*i)+1)/e for i=2 we got whole number

d=((192\*2)+1)/35

d=11

Hence the decryption key is =11

**3b** .(i).Euler Totient Function. φ(360), φ(777) 5M

φ(360) φ(777)

=φ (23\*32\*5) =φ (3\*7\*37)

=φ(4\*6\*4) =2\*6\*36

=24\*4 =432

=96

(ii). GCD of two numbers .gcd(108,144), gcd(161,28) 5M

gcd(108,144)

108=22\*32\*3

144=22\*22\*32

Common factors on both the numbers are 22\*32,hence the gcd(108,144)=36

gcd(161,28)

161=7\*23

144=2\*2\*7

Common factors on both the numbers are 7.hence the gcd(161,28)=7

**OR**

4 a. Let p=23 and q=5 Alice picks x=4 and Bob picks y=3, Compute the shared secret between Alice and Bob using Diffie-Hellman key exchange protocol. 10M

**Formulas used in Diffie-Hellman key exchange protocol:**

1. Key = (YA) XBmod p -> this is the same as calculated by B

2. Global Public Elements

p: p is a prime number. here in our example p=23

α: α < p and α is the primitive root of p. here in our example α =5

3. Key generation for user A

* Select a Private key XA   Here, XA<p

Now, Calculation of Public key YAYA = aXA mod p

4. Key generation for user B

* Select a Private key XB    Here, XB<q
* Now, Calculation of Public key YBYB = aXb mod q

5. Calculation of Secret Key by A

* key =(YB)XAmod p

6. Calculation of Secret Key by B

* key =(YA)XBmod p

**Example**

1. Alice and Bob both use public numbers P = 23, q = 5

2. Alice selected private key a = 4, and Bob selected b = 3 as the private key

3. Both Alice and bob now calculate the value of x and y as follows:

* Alice:    x = (54 mod 23) = 4
* Bob:    y = (53 mod 23) = 10

4. Now, both Alice and Bob exchange public numbers with each other.

5. Alice and Bob now calculate the symmetric keys

* Alice: ka = ya mod p = 104 mod 23 = 18
* Bob: kb = xb mod p = 43 mod 23 = 18

18 is the shared secret key.

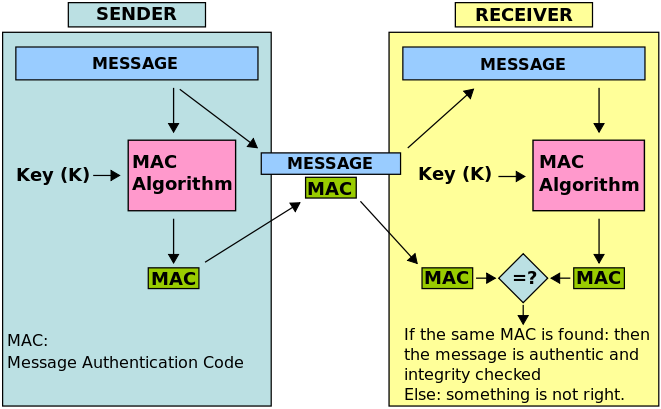
4b.Write a note on

a). MAC b). Digital Signature 10M

### **MAC**

MAC stands for Message Authentication Code. MAC algorithm is a symmetric key cryptographic technique to provide message authentication. For establishing MAC process, the sender and receiver share a symmetric key K. Essentially, a MAC is an encrypted checksum generated on the underlying message that is sent along with a message to ensure message authentication. Here in MAC, sender and receiver share same key where sender generates a fixed size output called **Cryptographic checksum or Message Authentication code** and appends it to the original message. On receiver’s side, receiver also generates the code and compares it with what he/she received thus ensuring the originality of the message. These are components:

* Message
* Key
* MAC algorithm
* MAC value



b) Digital Signature

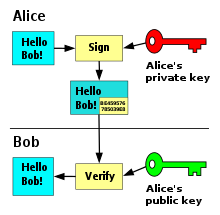
* A digital signature is **a cryptographic output used to verify the authenticity of data**.A digital signature algorithm consists of two operations:

**a signing operation**, which uses a signing key to produce a signature over raw data.

**a verification operation,** where the signature can be validated by a party who has no knowledge of the signing key.

The main purposes of a digital signature are:

* verification of the integrity of the signed data.
* non-repudiation if the signer claims the signature is not authentic
* Digital signatures rely on asymmetric cryptography, also known as public key cryptography. **An asymmetric key consists of a public/private key pair.**The private key is used to create a signature, and the corresponding public key is used to verify the signature.



**SECTION III**

**5 a** Explain Hardening in Linux. 10M

Linux hardening refers to the process of securing a Linux operating system and reducing its potential attack surface. This is achieved by implementing various security measures, such as updating and patching the system, configuring secure permissions and users, disabling unnecessary services, using firewalls and intrusion detection systems, and more. The goal of Linux hardening is to minimize the risk of unauthorized access, theft of data, or other security incidents by making the system less vulnerable to attacks.

Linux security hardening is a specialized procedure which must contain following guidelines:

1) Remove Unneeded Functionality: Uninstall unnecessary software and disable system services that are not needed.

2) Minimize Open Ports and Other Network Vulnerabilities: Network-based attacks are among the most common threats. To reduce your risk, identify open networkaccessible ports and remove any corresponding processes that are not needed.

3) Review User Accounts and Authentication: User accounts and authentication should be governed using a centralized control system such as Active Directory.

4) Review Service Accounts: System and service accounts should be reviewed and any that are no longer required should be removed.

5) SSH Hardening for Linux and Unix

SSH (Secure Shell) hardening refers to the process of improving the security of an SSH server in a Linux system. This involves implementing various measures to reduce the risk of unauthorized access, theft of data, or other security incidents. Some common SSH hardening practices include:

* Updating and patching the SSH server software to the latest version
* Disabling root login through SSH
* Using strong authentication methods, such as key-based authentication instead of password authentication
* Configuring SSH to listen on a non-standard port, to reduce the risk of automated attacks
* Limiting the number of login attempts to prevent brute force attacks
* Disabling legacy protocols, such as Telnet, and using only secure protocols, such as SSH
* Using public key infrastructure (PKI) to secure SSH communications
* Enabling logging and auditing to monitor and track SSH activity.

6) Kernel Hardening for Linux: SELinux modifies the Linux kernel to enforce mandatory access controls, restricting how Linux processes can access files and programs. This additional layer of restriction provides a fundamental protection mechanism against root kit malware.

7) Embedded Linux Hardening: Embedded Linux provides a stripped-down operating system for embedded devices or embedded systems, such as smart phones, smart TVs, set-top boxes and broadband internet routers. For example, the open-source Android platform developed by Google is optimized for smart phones and TVs. The functionality included in the operating system varies widely depending upon the intended application.

8) Kali Linux Hardening: Kali Linux has been optimized to be the pen testers’ platform of choice, so it has a wide range of security auditing and pen testing utilities.

5 b. Explain the following in Windows: a). Third Patch management b). Windows Autopilot. 10M

## Third patch management:-

## Third-party patching (patch management) is the process of installing patches to third-party applications, that are installed on your company's endpoints, to address bugs or vulnerabilities in the software. Third-party patching is critical for the security of your organization that prevents data breaches.

* Third-party patch management: Third-party patch management refers to the process of managing and updating software applications from non-Microsoft sources. This includes updating software applications such as web browsers, media players, and productivity tools. Third-party patch management is important for keeping systems secure and up-to-date, as many security vulnerabilities are often found in third-party software.

Windows autopilot:-

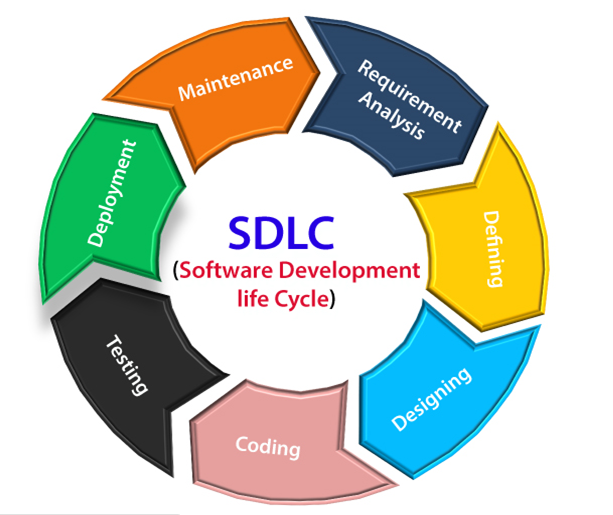
Windows Autopilot is **a collection of technologies used to set up and pre-configure new devices, getting them ready for productive use**.

Windows Autopilot is a service provided by Microsoft that allows for the deployment, reset, and repurposing of Windows 10 devices. Autopilot simplifies the device provisioning process, enabling IT administrators to quickly and easily set up new devices for use by their organizations. This includes personalizing the device, installing apps, and applying policies and settings. With Autopilot, IT administrators can also manage the device's life cycle, from initial deployment to end of life, ensuring that devices are always secure, up-to-date, and fully functional.

**OR**

6 a. Explain the Software Development Life Cycle. 10M

SDLC is the acronym for software development life cycle. It is also called the software development process. All the tasks required for developing and maintaining software. It consists of a plan describing how to develop, maintain, replace and alter the specific software. It is a process for planning, creating, testing, and information system. It is a framework of describes the activity performed at each stage of software development. It is a process used by a system analyst to develop an information system including requirements, validation, training, and ownership

. 

**Stage-1:Requirement gathering**   
The feasibility report is positive towards the project and next phase start with gathering requirement from the user. Engineer communicates with the client and end-users to know their Idea and which features they want to software to include.

**Stage-2: Software design**   
It is a process to transform user requirements into a suitable form. It helps programmers in software coding. There is a need for more specific and detailed requirements in software. The output of the process can directly be used in implementation in a programming language. There are three design levels as follows.

1. **Architectural design**   
   It is the highest abstract version of the system. In a software system, many components interact with each other.
2. **High-level design**   
   It focuses on how the system along with all its components and its can be implemented in form of modules.
3. **Detailed design**   
   It defines the logical structure of each module and its interface to communicate with each module.
4. **Stage-3:** **Developing Product**   
   In this phase of SDLC, you will see how the product will be developed. It is one of the crucial parts of SDLC, It is also called the Implementation phase.

**Stage-4:** **Product Testing and Integration**   
In this phase, we will integrate the modules and will test the overall product by using different testing techniques.

**Stage-5 Deployment maintenance**

Linux hardening refers to the process of securing a Linux operating system and reducing its potential attack surface. This is achieved by implementing various security measures, such as updating and patching the system, configuring secure permissions and users, disabling unnecessary services, using firewalls and intrusion detection systems, and more. The goal of Linux hardening is to minimize the risk of unauthorized access, theft of data, or other security.  
In this phase, the actual deployment of the product, or you can say the final product will be deployed, and also we will do maintenance of product for any future update and release of new features.

6b. What is threat modelling? Explain different threat modelling methodologies. 10M

Threat modeling is a method of optimizing network security by locating vulnerabilities, identifying objectives, and developing countermeasures to either prevent or mitigate the -attacks against the system.

Threat Modeling effects of cyber Methodologies

1. STRIDE A methodology developed by Microsoft for threat modeling, it offers a mnemonic for identifying security threats in six categories:

* **Spoofing**: An intruder posing as another user, component, or other system feature that contains an identity in the modeled system.
* **Tampering**: The altering of data within a system to achieve a malicious goal.
* **Repudiation**: The ability of an intruder to deny that they performed some malicious activity, due to the absence of enough proof.
* **Information Disclosure**: Exposing protected data to a user that isn't authorized to see it.
* **Denial of Service**: An adversary uses illegitimate means to exhaust services needed to provide service to users.
* **Elevation of Privilege**: Allowing an intruder to execute commands and functions that they aren't allowed to.

2. P.A.S.T.A This stands for Process for Attack Simulation and Threat Analysis, a seven-step, risk-centric methodology. It offers a dynamic threat identification, enumeration, and scoring process. Once experts create a detailed analysis of identified threats, developers can develop an asset-centric mitigation strategy by analyzing the application through an attacker-centric view.

### 3. OCTAVE The Operationally Critical Threat, Asset, and Vulnerability Evaluation (OCTAVE) process is a risk-based strategic assessment and planning method. OCTAVE focuses on assessing organizational risks only and does not address technological risks. OCTAVE has three phases:

* Building asset-based threat profiles. (Organizational evaluation)
* Identifying infrastructure vulnerabilities. (Information infrastructure evaluation)
* Developing and planning a security strategy. (Evaluation of risks to the company's critical assets and decision making.)

4. Trike Trike focuses on using threat models as a risk management tool. Threat models, based on requirement models, establish the stakeholder-defined "acceptable" level of risk assigned to each asset class. Requirements model analysis yields a threat model where threats are identified and given risk values. The completed threat model is then used to build a risk model, factoring in actions, assets, roles, and calculated risk exposure.

5. VAST Standing for Visual, Agile, and Simple Threat modeling, it provides actionable outputs for the specific needs of various stakeholders such as application architects and developers, cybersecurity personnel, etc. VAST offers a unique application and infrastructure visualization plan so that the creation and use of threat models don't require any specialized expertise in security subject matters.

**SECTION III**

7. a. Illustrate static code analysis vs dynamic code analysis. 10M

|  |  |
| --- | --- |
| **SAST** | **DAST** |
| **White box security testing**  The tester has access to the underlying framework, design, and implementation. The application is tested from the inside out. This type of testing represents the developer approach. | **Black box security testing**  The tester has no knowledge of the technologies or frameworks that the application is built on. The application is tested from the outside in. This type of testing represents the hacker approach. |
| **Requires source code**  SAST doesn’t require a deployed application. It analyzes the sources code or binary without executing the application. | **Requires a running application**  DAST doesn’t require source code or binaries. It analyzes by executing the application. |
| **Finds vulnerabilities earlier in the SDLC**  The scan can be executed as soon as code is deemed feature-complete. | **Finds vulnerabilities toward the end of the SDLC** Vulnerabilities can be discovered after the development cycle is complete. |
| **Less expensive to fix vulnerabilities**  Since vulnerabilities are found earlier in the SDLC, it’s easier and faster to remediate them. Findings can often be fixed before the code enters the QA cycle. | **More expensive to fix vulnerabilities**  Since vulnerabilities are found toward the end of the SDLC, remediation often gets pushed into the next cycle. Critical vulnerabilities may be fixed as an emergency release. |
| **Can’t discover run-time and environment-related issues**  Since the tool scans static code, it can’t discover run- time vulnerabilities. | **Can discover run-time and environment-related issues**  Since the tool uses dynamic analysis on an application, it is able to find run-time vulnerabilities. |
| **Typically supports all kinds of software** Examples include web applications, web services, and thick clients. | **Typically scans only apps like web applications and web services**  DAST is not useful for other types of software. |

7 b. Explain top 10 OWSAP issues in Web Applications 10M

The OWASP Mobile Top 10 give you an overview of the ten most critical security risks to your apps and web applications.

**1. Improper Platform Usage** The first item among the OWASP top 10 is improper platform usage. Platforms such as iOS, Android, or Windows Phone provide different capabilities and features that you can use. If the app does not use an existing function or even uses it incorrectly, this is called improper use. This can be, for example, a violation of published guidelines that affects the security of the app.

**2. Insecure Data Storage** Insecure data storage as well as unintentional data leaks also fall under the OWASP Mobile Top Ten. Mobile application penetration testing tools help uncover such grievances. However, it does not necessarily have to be your SQL database. Manifest and log files, cookie storage or cloud synchronization can also be affected.

**3. Insecure Communication** Your app transports data from point A to point B. If this transport is insecure, the risk increases. Here, too, the main mobile application penetration testing tools will help you. They support you in detecting faulty app-to-server or mobile-to-mobile communication.

**4. Insecure Authentication** Secure authentication adds another key security aspect to your OWASP Mobile Security Checklist. In fact, there are many different ways that the app can provide insecure authentication. A classic example is a back-end API service request that the mobile app executes anonymously without relying on an access token.

**5. Lack of Cryptography** The insecure use of cryptography can be observed in most app applications. This is usually one of two problems: a fundamentally flawed process behind the encryption mechanisms or the implementation of a weak algorithm.

**6. Insecure Authorization** Unlike authentication, authorization deals with the verification of an identified person. It verifies that the necessary authorizations are in place to perform certain actions. Of course, the two are closely related - yet both items belong separately on the OWASP Top 10 list

**7. Poor Client Code Quality** This item of the OWASP Top 10 refers to an explicit programming language. All vulnerabilities from code-level errors can provide attackers with a way inside. The main risk lies in the need to make localized changes to the code. In particular, insecure API usage or insecure language constructs are common problems that you need to fix directly at the code level.

**8. Code Manipulation** From a technical perspective, any code on a mobile device is vulnerable to tampering. This is because the mobile code is running in a foreign environment. It is no longer under the control of your organization. Therefore, there are numerous ways to modify it at will.

**9. Reverse Engineering** Attackers who want to understand how your app works can use reverse-engineering to access all the information they need. Especially metadata, which is supposed to be a relief for your programmers, is a high risk. Basically, if you can clearly understand the string table of the binary or cross-functional analysis is possible, the app is considered at risk.

**10. Extraneous Functionality** Hidden backdoor functionality or internal security controls are a common problem in mobile applications. The problem with them is that they are not only useful for developers, but also for hackers. This allows them, for example, to disable 2-factor authentication or change basic functionality.

**OR**

8. a. Explain IAM best practices. 10M

* IAM stands for Identity Access Management.
* IAM allows you to manage users and their level of access to the aws console.
* It is used to set users, permissions and roles. It allows you to grant access to the different parts of the aws platform.
* AWS Identity and Access Management is a web service that enables Amazon Web Services (AWS) customers to manage users and user permissions in AWS.
* With IAM, Organizations can centrally manage users, security credentials such as access keys, and permissions that control which AWS resources users can access.
* Without IAM, Organizations with multiple users must either create multiple user accounts, each with its own billing and subscriptions to AWS products or share an account with a single security credential. Without IAM, you also don't have control about the tasks that the users can do.
* IAM enables the organization to create multiple users, each with its own security credentials, controlled and billed to a single aws account. IAM allows the user to do only what they need to do as a part of the user's job.

**IAM best practices**

1. Adopt a Zero Trust Approach to Security

2. Identify and Protect High-Value Data

3. Enforce a Strong Password Policy

4. Use Multi-Factor Authentication (MFA)

5. Automate Workflows

6. Adopt The Principle of Least Privilege

7. Enforce Just-in-Time Access Where Appropriate

8. Leverage Both Role-Based Access Control and Attribute-Based Access Control Policies

9. Regularly Audit Access to Resources

10. Centralize Log Collection

.b. Elaborate on Cloud Vulnerabilities 10M

Common cloud vulnerabilities are

1. Cloud Misconfiguration: Cloud misconfiguration is the the most common vulnerability in organizations. Misconfigurations are often caused by a lack of knowledge of good practices or lack of peer review from DevOps/infra team.

2. Identity and Access Management Unsecure identity and access management (IAM) is a common vulnerability in cloud systems. it occurs when a user or service of your infrastructure has access to resources they should not be able to access and/or do not need.

3. Public Data Storage This vulnerability occurs when a given data blob, like an S3 bucket or, an SQL database, is partly or completely opened to the public to access via either read-only or both read and write. DevOps team, sysadmins, and managers should follow some basic principles to minimize the risk of public data storage misconfiguration.

4.Insecure APIs APIs are used in modern software development as microservices, application and website backends. They must handle requests received from mobile devices, applications, webpages and third parties. This is why having a secure API is critical to ensuring cyber threat mitigation and to protect against unwanted traffic.

Some of the most malicious requests are:

• Code and query injection (SQL injection, command injection)

• Taking advantage of a bad access control

• Targeting a vulnerability due to an outdated component (software libraries, database engine, runtime environments, etc.)

5.Lack of Visibility When companies are using thousands of instances of cloud services, it can be easy to get lost in them or forget about some of those running instances. Lack of visibility of cloud infrastructure is a major issue that can delay action on a threat and result in a data breach. Managers, sysadmins and DevOps teams must take a proactive security approach.

6. Lack of Multi-factor Authentication Multi-factor authentication (MFA) is an authentication method in which a user must present at least two forms of identification validation to access an account or data.

7. Malicious Insiders Unauthorized access occurs when a user obtains access to some or all of the company’s cloud resources. Malicious insiders can also access cloud resources via account hijacking due to a successful phishing attack and/or weak credential security.

8.Distributed Denial-of-Service Attacks Distributed denial-of-service (DDoS) attacks are malicious efforts to take down a web service such as a website. It works by flooding the server with requests from different sources. The goal is to make the server unresponsive to requests from legitimate users.

**SECTION V**

9. a. Explain Vulnerability life cycle. Elaborate the types of Vulnerabilities. 10M

The Vulnerability Management Life Cycle is intended to allow organizations to identify computer system security weaknesses; prioritize assets; assess, report, and remediate the weaknesses; and verify that they have been eliminated.

The following diagram illustrates the steps in the Vulnerability Management Life Cycle.



The steps in the Vulnerability Management Life Cycle are described below.

1. **Discover:** Inventory all assets across the network and identify host details including operating system and open services to identify vulnerabilities. Develop a network baseline. Identify security vulnerabilities on a regular automated schedule.

2. **Prioritize Assets:** Categorize assets into groups or business units, and assign a business value to asset groups based on their criticality to your business operation.

3. **Assess:** Determine a baseline risk profile so you can eliminate risks based on asset criticality, vulnerability threat, and asset classification.

4. **Report:** Measure the level of business risk associated with your assets according to your security policies. Document a security plan, monitor suspicious activity, and describe known vulnerabilities.

5. **Remediate:** Prioritize and fix vulnerabilities in order according to business risk. Establish controls and demonstrate progress.

6. **Verify:** Verify that threats have been eliminated through follow-up audits.

Types of Vulnerabilities are

**1. Hardware Vulnerability**: A hardware vulnerability is a weakness which can used to attack the system hardware through physically or remotely. For examples:

* + Old version of systems or devices
  + Unprotected storage
  + Unencrypted devices, etc.

**2. Network Vulnerability** A weakness happen in network which can be hardware or software.

For examples:

* + Unprotected communication
  + Malware or malicious software (e.g.:Viruses, Keyloggers, Worms, etc)
  + Misconfigured firewalls

**3. Operating System Vulnerability**

An operating system vulnerability is a loophole or flaw in your operating system that makes it easier for cybercriminals to break in. An operating system is the main software that runs your computer or device – common examples include Windows, MacOS, Android, and Linux

**4.Application Vulnerability**

Application vulnerabilities are weaknesses in an application that an attacker could exploit to harm the security of the application. Vulnerabilities can be introduced into an application in various ways, such as failures in the design, implementation, or configuration of an application.

**5.Human Vulnerabilities**

People represent one of the main weaknesses of cybersecurity. In fact, human vulnerabilities can cause much more damage and be more costly than any of the other vulnerability types on this list. Even though advanced hacking skills and powerful malware bolster the capabilities of a cyber attacker, it is, in the end, humans that represent the only un-patchable risk in cybersecurity. Ex: Social engineering. Defined as the art of gaining access to data, systems, or buildings by exploiting human psychology.

**6.Process Vulnerabilities**

Process vulnerabilities are unique as they are created by specific process controls or the lack of them. Process controls can look different depending on the industry of the organization. This type of vulnerability is certainly the hardest to define. Ex: If an employee followed a process in place which allowed them to create and use a weak password.

9 b. Explain Android Application Security architecture. 10M

Android architecture contains different number of components to support any android device needs.

The main components of android architecture are following:-

* Applications
* Application Framework
* Android Runtime
* Platform Libraries
* Linux Kernel

## **Applications** –

Applications is the top layer of android architecture. The pre-installed applications like home, contacts, camera, gallery etc and third party applications downloaded from the play store like chat applications, games etc. will be installed on this layer only.  
It runs within the Android run time with the help of the classes and services provided by the application framework.

## **Application framework** –

Application Framework provides several important classes, which are used to create an Android application. It provides a generic abstraction for hardware access and helps in managing the user interface with application resources. It includes different types of services activity manager, notification manager, view system, package manager etc. which are helpful for the development of our application according to the prerequisite.

## **Application runtime** –

Android Runtime environment is one of the most important part of Android. It contains components like core libraries and the Dalvik virtual machine (DVM). Mainly, it provides the base for the application framework and powers our application with the help of the core libraries.

## **Platform libraries** –

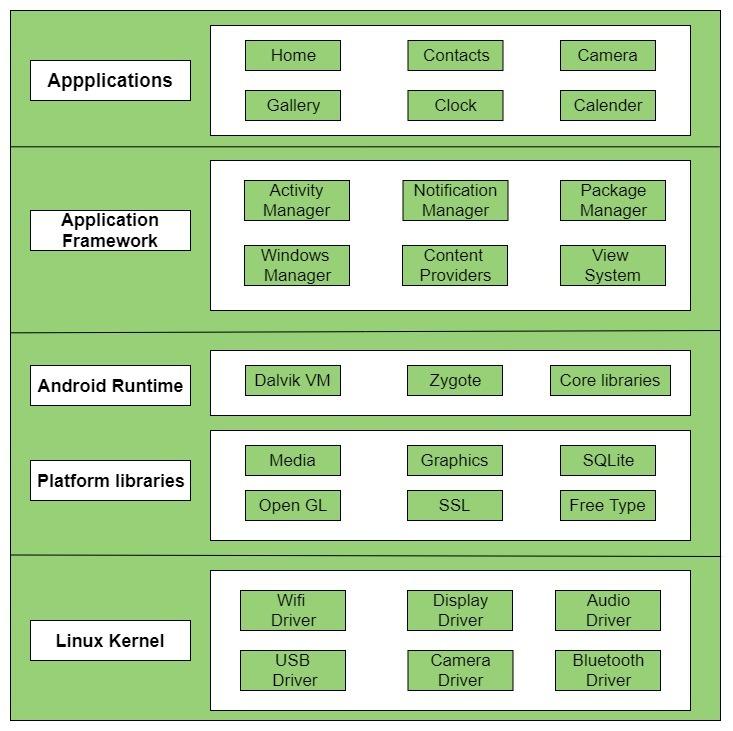
The Platform Libraries includes various C/C++ core libraries and Java based libraries such as Media, Graphics, Surface Manager, OpenGL etc. to provide a support for android development.

* **Media** library provides support to play and record audio and video formats.
* **Surface manager** responsible for managing access to the display subsystem.
* **SGL** and **OpenGL** both cross-language, cross-platform application program interface (API) are used for 2D and 3D computer graphics.
* **SQLite** provides database support and **FreeType** provides font support.
* **Web-Kit** This open source web browser engine provides all the functionality to display web content and to simplify page loading.
* **SSL (Secure Sockets Layer)** is security technology to establish an encrypted link between a web server and a web browser.

## **Linux Kernel** –

Linux Kernel is heart of the android architecture. It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime.

The Linux Kernel will provide an abstraction layer between the device hardware and the other components of android architecture. It is responsible for management of memory, power, devices etc.



**OR**

10 a. Explain GRC in an company and its benefits. 10M

GRC stands for Governance, Risk Management, and Compliance. It refers to a set of processes and technologies used by organizations to manage and monitor compliance with legal and regulatory requirements, manage risks and ensure that the company is operating in an ethical and transparent manner.

Core GRC principles:

Governance: Ensuring that organizational activities support the organization's business goals.

Risk: The identification, classification and addressing of any risk associated with organizational activities.

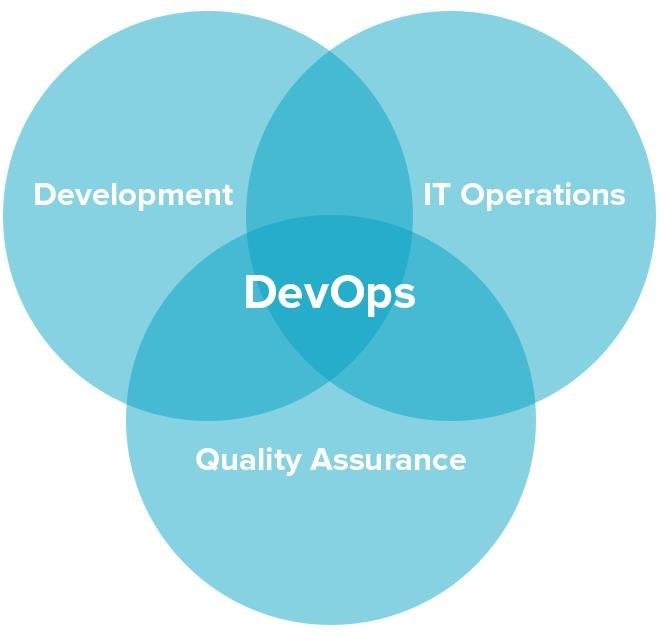
Compliance: Ensuring that an organization is meeting compliance with all legal and regulatory requirements

**The benefits of a strong GRC program in a company include:**

* **Improved Risk Management:** By having a clear understanding of the risks facing the company and implementing strategies to manage those risks, companies can minimize potential harm and maximize opportunities.
* **Increased Compliance**: GRC helps ensure that the company is following all relevant laws, regulations, and ethical standards, reducing the risk of fines, legal action and reputational damage.
* **Enhanced Transparency**: A strong GRC program promotes transparency in the company's operations and decision-making processes, which can increase trust among stakeholders and enhance the company's reputation.
* **Efficient Use of Resources**: By streamlining processes and utilizing technology to automate GRC activities, companies can reduce the time and cost associated with managing risks and ensuring compliance.
* **Improved Decision Making**: By having a comprehensive understanding of the company's risks and compliance obligations, companies can make better-informed decisions that promote long-term success.

10.b. Illustrate the need of DevOps in Companies and its Security Challenges. 10M DevOps is a software development practice that emphasizes collaboration and communication between development and operations teams. The goal of DevOps is to improve the speed and reliability of software delivery, as well as to increase the efficiency and responsiveness of the overall development process.

* DevOps security (DevSecOps) is an approach to cybersecurity that focuses on application development and development operations (DevOps). It combines three phrases:
  1. Development 2.Operations 3.Security



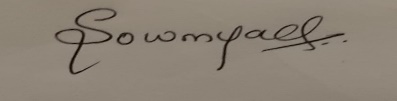
* The goal of DevOps security is to remove barriers between an organization’s software development and its IT operations. Namely, it calls for speed and intense, fast communication and collaboration. It is essentially a philosophy that covers developers’ code and its subsequent need to work (and grow) properly with the organization’s employees and customers.
* core of the DevOps security philosophy is **continuous deployment.** Development and IT teams work closely and rapidly to add to and fix software. This means, for example, adding new features and troubleshooting bugs so that they can be continuously released in fast cycles without causing disruptions. It enables teams, other employees, and customers to continue interacting with software without interruption

**Security challenges in DevOps include:**

* **Configuration management**: ensuring the consistency and security of configurations across multiple environments, from development to production.
* **Continuous Integration and Continuous Deployment (CI/CD):** securing the pipeline and ensuring that only authorized changes are deployed to production.
* **Infrastructure as Code (IaC):** managing the security of infrastructure components defined as code, such as cloud resources or containerized applications.
* **Container security**: securing the lifecycle of containers, from build to deployment and runtime.
* **Secrets management**: securely storing and managing sensitive information, such as passwords and access keys, in the DevOps environment.
* **Compliance:** ensuring that DevOps practices align with regulations and industry standards, such as those related to data privacy and security

**CERTIFICATE**

This is to certify that all the model answers prepared by me for subject **CYBER SECURITY (20CS54I )** are as per the syllabus.



SOWMYA C J

LECTURER

COMPUTER SCIENCE AND ENGINEERING

GOVERNMENT POLYTECHNIC ARAKERE-157