

V SEMESTER DIPLOMA EXAMINATION JAN/FEB-2023
SCHEME OF VALUATION
CYBER SECURITY
SUB CODE : 20CS54I, V SEM

Q. no		DESCRIPTION	Marks distribution	Total marks
SECTION I				
1	a	Layer identification + any 4 types + advantage	2+4+4	10
	b	Device protection + Data protection	5+5	10
OR				
2	a	Explanation of five	5 x 2	10
	b	List of stages + Explanation of each stages	3 + 7	10
SECTION II				
3	a	Algorithm + example steps	6+8	14
	b	Hashing + Digital signature	3+3	6
OR				
4	a	Calculating decryption key + MSG encrypt and decrypt	8+6	14
	b	solving problem	3 + 3	6
SECTION III				
5	a	Diagram + explanation	4+6	10
	b	Responsibility (customer+cloud provider)+ Advantages	4+3+3	10
OR				
6	a	Diagram + explanation	6+4	10
	b	Blue team + Red team	5 +5	10
SECTION IV				
7	a	Steps to find vulnerability + common vulnerability(any six)	4 + 6	10
	b	Benefits + tools	5 + 5	10
OR				
8	a	Responsibility (customer + cloud provider)	5 + 5	10
	b	Need of DCA + Stage identification	8 + 2	10
SECTION V				
9	a	Diagram + explanation	4+6	10
	b	Importance + types +Features	3+3+4	10
OR				
10	a	Patch management + Antivirus management	5+5	10
	b	List + Explanation	4+6	10

V semester Diploma Examination Jan-2023
CYBER SECURITY 20CS54I
Model answers

NOTE: All model answers are general specific to subject, if any answers are relevant please give marks.

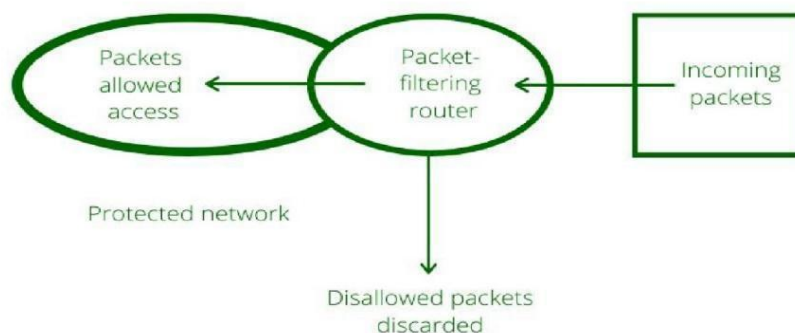
Q1. (a) At which layer of OSI stack firewall will be placed? Give the types and advantages of firewall in cyber security?

Ans: Network layer where firewall will be placed in OSI stack.

Different Types of Firewalls in Cyber Security

[Firewall in Cyber Security: Types, Advantages & Components \(knowledgehut.com\)](https://www.knowledgehut.com/cyber-security/types-of-firewalls/)

1. Packet-filtering Firewalls



A packet-filtering firewall is a type of firewall that can obstruct network traffic IP protocol, a port number, and an IP address. It applies a set of rules, which are based on the contents of IP and transport header fields on each packet. Upon receiving and analysing the outcome, the packet-filtering firewall decides whether to forward or discard the packet.

2. Proxy Service Firewalls

Proxy service firewalls are systems that filter messages at the application layer to improve network security. You can think of it as an intermediary between your internal network and outside servers. By analysing incoming traffic with stateful and deep packet inspection technology, they are more secure than traditional firewalls.

3. Stateful Multilayer Inspection (SMLI) Firewalls

Stateful Multilayer Inspection firewalls keep track of connections and provide standard firewall capabilities. Traffic is filtered based on state, port, and protocol, as well as administrator-defined rules and contexts. A prior connection and packets from a current connection are used in this process.

4. Unified Threat Management (UTM) Firewalls

SMLI firewalls work in conjunction with intrusion prevention and antivirus software to form a unified threat management firewall. UTM may include additional services such as cloud management.

5. Next-generation Firewalls (NGFW)

Compared to packet-filtering and stateful inspection firewalls, next-generation firewalls are more sophisticated. Unlike standard packet filters, they perform a more thorough inspection of packets, examining not just packet headers but also their contents and sources. As security, threats evolve and become more sophisticated, NGFWs are able to block them.

6. Network Address Translation (NAT) Firewalls

As a result, NAT firewalls are capable of assessing internet traffic and blocking unsolicited communications, so they only accept inbound web traffic from devices on your private network.

7. Virtual Firewalls

In cloud-based systems, both private and public, virtual firewalls serve as security appliances. Internet traffic is assessed and managed using this type of firewall, whether it is over a physical or virtual network.

Advantages of Using Firewalls

[What Is Firewall: Types, How Does It Work & Advantages | Simplilearn](#)

The 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.

Q 1.b) You have purchased a laptop for your business purpose then what measures shall you take to protect your device and data.

[Data Security Best Practices: 10 Methods to Protect Your Data | Ekran System](#)

1. Encryption — prevents unauthorized parties from reading your data.

2. Masking — suppresses or anonymizes high-value data by replacing sensitive information with random characters. You can also substitute data with a low-value representative token; this method is called **tokenization**.

3. Data erasure — involves cleaning your repository in case stored data is no longer used or active.

4. Data resilience — involves full, differential, and incremental **backups** of your critical data. Storing your valuable data in different locations helps to make it recoverable and resilient to different cybersecurity threats.

[Basic Security Measures You have to Take to Protect Your Digital Assets and Devices - InfosecTrain](#)

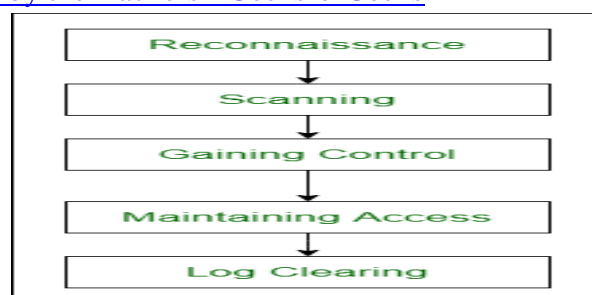
- 1. Secure the company Wi-Fi and stay up to date:** Firstly, you must update your software frequently because the old versions cannot prevent new hacking methods. You must adhere to Wi-Fi security best practices to protect your digital assets. Ensure your router is kept in a physically safe position, change the network name, ensure the firewall is enabled, and regularly update the firmware and software.
- 2. Maintain a Data Backup:** Data Backup is a process of copying the data files and storing them in a different location. We can restore the files whenever we want. Data Backup is very important: assume your system were crashed, or assume your system was locked by some ransomware that is when you can use the backed up data.
- 3. Two-step verification:** In the world of cyber-crimes, hoping that a password can save your data is nothing but your innocence. There are many social engineering techniques where a cybercriminal can easily get the credentials of your social media accounts, your bank accounts, or any other private accounts. So the only solution to escape the attacks is to have two-step verification. This is nothing but using an OTP to login into your accounts. This method is helpful because you will receive the OTP to your mobile (physical device), which an attacker cannot access. Even if the attacker knows your credentials, he cannot log in to your account until they have the OTP.
- 4. Limit Access:** Restricting access to digital assets and systems reduces the risk of loss or theft. Limiting access is a crucial step in protecting digital assets. Make sure that only those workers who need to use digital assets and systems have access to them. Authorized users of these systems should follow data security best practices, including password protection and authentication while utilizing personal devices and other risk factors.
- 5. Cyber Insurance:** Cyber insurance is kind of similar to regular insurance. For example, take health insurance. This health insurance can compensate for the money when you are ill. You can pay your medical bills with health insurance. Similarly, Cyber Insurance can compensate for the costs of your data loss and investigations of cyberattacks.

- 6. Document Protocols:** Create a calendar and set timelines for activities like backups, upgrades, and software reviews using an ongoing schedule and calendar. Make a list of your company's current protocols, and make sure to update them as required in the future. Guidelines for digital asset management should be viewed as a dynamic document that changes and evolves as content, applications, and programs mature and evolve.

Q2. (a) Describe the way hackers collect information from intended users/organization.

Methodology followed by the Hackers

[Methodology followed by the Hackers - GeeksforGeeks](#)



Different types of methodologies:

1. Reconnaissance

Reconnaissance is the process of gathering information about the target system. Finding vulnerabilities in the computer system or the methods that are left vulnerable is part of the process. If the hacker is able to get access to the system, he or she will continue the hacking procedure. The hacker has a lot of knowledge at the end of the reconnaissance phase, which he can use to build a promising attack on the target system.

2. Scanning

Before launching an attack, the hacker wants to determine whether the system is operational, which apps are in use, and what versions of those programs are in use. Scanning entails looking for all open and closed ports in order to locate a backdoor into the system. It entails getting the target's IP address, user accounts, and other information. The information acquired during the reconnaissance phase is utilized to inspect the network using tools such as port scanners. N-map is a popular, powerful, and freely available scanning tool.

3. Gaining Control

The information obtained in the previous two phases is utilized to enter and take control of the target system over the network or physically in this phase of the hacking method. This stage is often referred to as "Owning the System."

4. Maintaining Access

After acquiring access to the system in the previous stage, the hacker keeps the access for

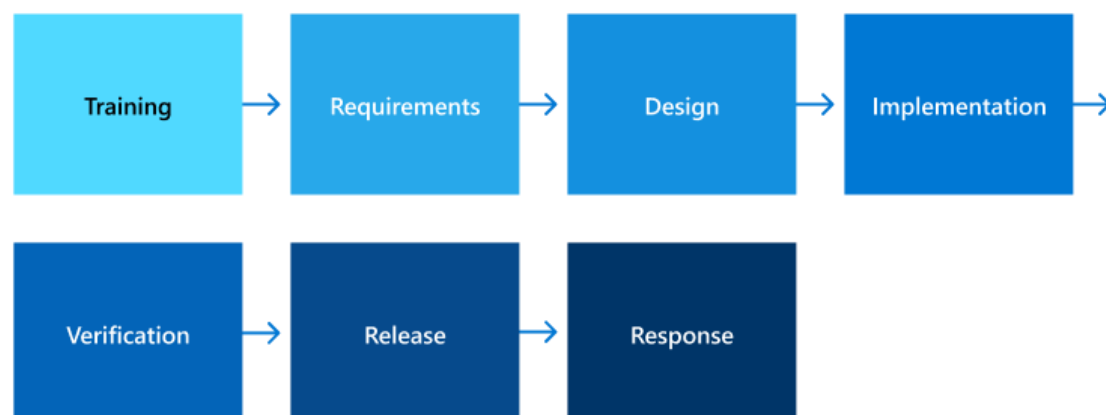
future attacks and makes changes to the system so that no other security personnel or hacker can acquire access to the compromised system. The attacked system is referred to as the “Zombie System” in this case.

5. Log Clearing

It is the method of erasing any remaining log files or other sorts of evidence on the hacked system that could lead to the hacker’s capture. Penetration testing is one of the instruments in ethical hacking approaches that can be used to catch a hacker.

Q2. (b) Think that you are the security manager for your project team, how do you apply secure SDLC in each stage of project development.

Microsoft SDL consists of seven components including five core phases and two supporting security activities. The five core phases are requirements, design, implementation, verification, and release. Each of these phases contains mandatory checks and approvals to ensure all security and privacy requirements and best practices are properly addressed. The two supporting security activities, training and response are conducted before and after the core phases respectively to ensure they're properly implemented, and software remains secure after deployment.



Training

All Microsoft employees are required to complete general security awareness training and specific training appropriate to their role.

Requirements

Every product, service, and feature Microsoft develops starts with clearly defined security and privacy requirements; they're the foundation of secure applications and inform their design. Development teams define these requirements based on factors such as the type of data the product will handle, known threats, best practices, regulations and industry requirements, and lessons learned from previous incidents. Once defined, the requirements are clearly defined, documented, and tracked.

Design

Once the security, privacy, and functional requirements have been defined, the design of the software can begin. As a part of the design process, threat models are created to help identify, categorize, and rate potential threats according to risk. Threat models must be maintained and updated throughout the lifecycle of each product as changes are made to the software.

Implementation

Implementation begins with developers writing code according to the plan they created in the previous two phases. Microsoft provides developers with a suite of secure development tools to effectively implement all the security, privacy, and function requirements of the software they design. These tools include compilers, secure development environments, and built-in security checks.

Verification Testing

Before any written code can be released, several checks and approvals are required to verify that the code conforms to SDL, meets design requirements, and is free of coding errors. Various automated checks are also required and are built into the commit pipeline to analyse code during check-in and when builds are compiled.

Release

After passing all required security tests and reviews, builds aren't immediately released to all customers. Builds are systematically and gradually released to larger and larger groups, referred to as rings, in what is called a safe deployment process (SDP).

Response

All Microsoft services are extensively logged and monitored after release, identifying potential security incidents using a centralized proprietary near-real-time monitoring system

Q3. (a) Let $p=23$ and $q=5$ Alice picks $x=4$ and Bob picks $y=3$, compute the shared secrete key between Alice and Bob using Daffy Hellman key exchange protocol Algorithm

ALGORITHM:

1. Key = $(Y_A)^{XB \bmod q}$ -> this is the same as calculated by B
2. Global Public Elements

- q : q is a prime number
- a : $a < q$ and α is the primitive root of q

3. Key generation for user A

- Select a Private key X_A Here, $X_A < q$

Now, Calculation of Public key Y_A $Y_A = a^{X_A} \bmod q$

4. Key generation for user B

- Select a Private key X_B Here, $X_B < q$
- Now, Calculation of Public key Y_B $Y_B = a^{X_B} \bmod q$

5. Calculation of Secret Key by A

- $\text{key} = (Y_B)^{X_A} \bmod q$

6. Calculation of Secret Key by B

- $\text{key} = (Y_A)^{X_B} \bmod q$

Example

1. Alice and Bob both use public numbers $P = 23$, $G = 5$
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: $x = (5^4 \bmod 23) = 4$
- Bob: $y = (5^3 \bmod 23) = 10$

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

5. Alice and Bob now calculate the symmetric keys

- Alice: $k_a = y^a \bmod p = 10^4 \bmod 23 = 18$
 - Bob: $k_b = x^b \bmod p = 4^3 \bmod 23 = 18$
6. 18 is the shared secret key.

Q3. (b) Give the importance of the following

I. Hashing

II. Digital signature

I) Importance of Hashing

Hashing gives a more secure and adjustable method of retrieving data compared to any other data structure. It is quicker than searching for lists and arrays. In the very range, Hashing can recover data in 1.5 probes, anything that is saved in a tree. Hashing, unlike other data structures, doesn't define the speed. A balance between time and space has to be maintained while hashing.

II) Importance of a Digital Signature

A digital signature is required to ensure that the data or message being sent is legitimate. It is more trustworthy than receiving plaintext. Data integrity, message authentication, and message non-repudiation are all provided by digital signature. When the user verifies the digital signature using a public key that has been supplied by the originator, it helps to offer message authentication and ensures that the message is authentic and does not contain malware.

Q4. (a) Given the implementation of RSA $P=53$ $Q=59$ if encryption key is 3, what is the decryption key? Encrypt and decrypt the message '6' using above keys.

Generating encryption key (Public Key):

1. Select two prime no's. Suppose $P = 53$ and $Q = 59$.
2. Now First part of the Public key: $n = P \times Q = 3127$.
3. We also need a small exponent say e : But e Must be an integer. Not be a factor of n . $1 < e < \Phi(n)$, Given it to be equal to 3.
4. Our Public Key is made of n and e ($n = 3127$ and $e = 3$)

Generating decryption key (Private Key):

5. We need to calculate $\Phi(n)$: Such that $\Phi(n) = (P-1)(Q-1)$ so, $\Phi(n) = 3016$
6. Now calculate Private Key, d :

$$d = (k \times \Phi(n) + 1) / e \text{ for some integer } k$$
For $k = 2$,
value of decryption key d is 2011.

Now we are ready with our – Public Key ($n = 3127$ and $e = 3$) and Private Key($d = 2011$)

Now we will encrypt '6' :

1. Given the message: 6
2. Encrypted Data $c = 6^{e \bmod n}$.

$$C = 6^{3 \bmod 3127}$$

Thus, our Encrypted Data comes out to be 216

3. Now we will decrypt 216 :
4. Decrypted Data $= c^{d \bmod n}$.

$$E = 216^{2011 \bmod 3127}$$

Thus our Decrypted Data comes out to be 6.

Q.4.b) Find the GCD for the following:

(i)gcd(108,144)

(ii)gcd(360,210)

(i)gcd(108,144)

1. Find the prime factorization of 108

$$108 = 2 \times 2 \times 3 \times 3 \times 3$$
2. Find the prime factorization of 144

$$144 = 2 \times 2 \times 2 \times 2 \times 3 \times 3$$
3. To find the GCD, multiply all the prime factors common to both numbers:
Therefore, $GCD = 2 \times 2 \times 3 \times 3$
 $GCD = 36$

(ii) gcd(360,210)

1. Find the prime factorization of 360

$$360 = 2 \times 2 \times 2 \times 3 \times 3 \times 5$$

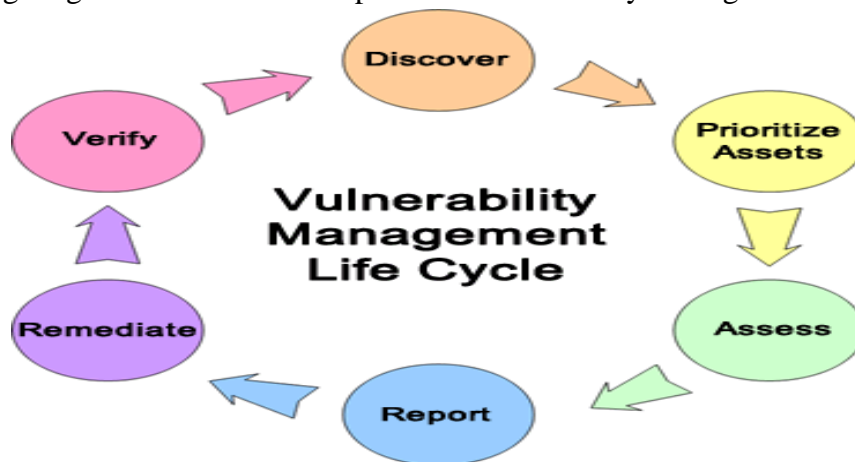
2. Find the prime factorization of 210
 $210 = 2 \times 3 \times 5 \times 7$
3. To find the GCD, multiply all the prime factors common to both numbers:
Therefore, $\text{GCD} = 2 \times 3 \times 5$
 $\text{GCD} = 30$

Q5. (a) Describe the life cycle of Vulnerability Management.

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.

Steps in the Vulnerability Management Life Cycle

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.

[What is the Shared Responsibility Model? | CrowdStrike](#)

Q5. (b) Describe the shared Responsibility Model in cloud with advantages.

The **Shared Responsibility Model** is a security and compliance framework that outlines the responsibilities of **cloud service providers (CSPs)** and **customers** for securing every aspect

of the cloud environment, including hardware, infrastructure, endpoints, data, configurations, settings, operating system (OS), network controls and access rights

Direct Control

While the Shared Responsibility Model is based on the idea that two or more parties play a role in ensuring security of distinct elements within the public cloud environment, it is important to note that the customer and CSP do not share responsibility for the same asset.

Rather, the CSP or the customer has full and complete responsibility for the security of all assets under their direct control, regardless of the service model type.

Customers are typically also responsible for:

- Identity Access and Management (IAM)
- User security and credentials
- Endpoint security
- Network security
- Security of workloads and containers
- Configurations
- APIs and middleware
- Code

Cloud Service Providers (CSPs)—such as Amazon, Microsoft or Google—are responsible for areas for which they possess direct control. This typically includes security of:

- The physical layer and all associated hardware and infrastructure
- The virtualization layer
- Network controls and provider services
- Facilities that run cloud resources

Shared Responsibility Model Advantages

While a shared security model is complex and requires careful consideration and coordination between the CSP and customer, the approach offers several important benefits to users. These include:

- **Efficiency:** Though the customer bears significant levels of responsibility under the Shared Responsibility Model, some key aspects of security – such as security of hardware, infrastructure and the virtualization layer – are almost always managed by the CSP. In a traditional on-premises model, these aspects were managed by the customer. The shift to the cloud frees up IT staff to refocus efforts on other tasks and

needs, as well as dedicate available resources and investments to those areas for which they bear responsibility.

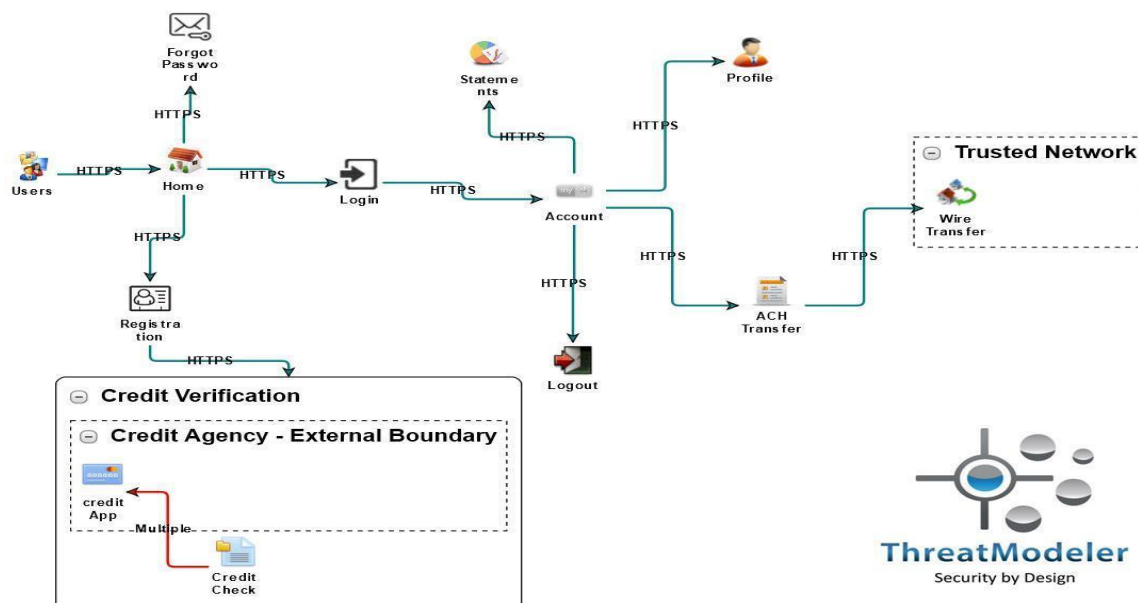
- **Enhanced protection:** Cloud service providers are hyper focused on the security of their cloud environment and typically dedicate significant resources to ensuring their customers are fully protected. As part of the service agreement, CSPs conduct robust monitoring and testing, as well as timely patching and updating.
- **Expertise:** CSPs often have a higher level of knowledge and expertise when it comes to the emerging field of cloud security. When customers engage a cloud vendor, they benefit from the partner organization's experience, assets and resources.

Q6. (a) Create a Threat Model for Secure Online Banking.

Secure Online Banking [Secure Online Banking - ThreatModeler](#)

Threat model is for an application, which is intended to allow authenticated customers to check their statements, update their profile information, or to transfer funds between financial institutions. The application also includes a credit check for new users.

The threat model diagram shown below maps how users may navigate from the application do icons on the diagram (represent home page through various use cases). Of particular importance regarding secure online banking, we should note that several of the identified use cases require data attackers consider high-value.



In particular:

- The **Forgot Password** use case requires the user's username and email address;
- The **Registration** use case requests username, password, and certain personally identifying information;
- The **Login** use case, of course, asks for the user's username and password;

- The **Profile** use case, in addition to other information, allows users to update their name and address;
- The **ACH Transfer** use case may allow users to transfer funds to vendors or other individuals; thus it asks for account and routing numbers, as well as the name and address of the recipient;
- The **Credit Verification** process will require the users' social security number, date of birth, and personal identifying information.

In other words, this application processes all the data types financial institution hackers seek. Providing secure online banking will make securing applications such as this, a priority for the organization.

Q6. (b) Describe the responsibility of Red team and Blue team.

I Red team

[Red Team vs. Blue Team in Cybersecurity | Coursera](#)

The National Institute of Standards and Technology (NIST) defines a red team as “a group of people authorized and organized to emulate a potential adversary’s attack or exploitation capabilities against an enterprise’s security posture.” The red team plays the part of the attacker or competitor with the intention of identifying vulnerabilities in a system.

Red team activities

- Social engineering
- Penetration testing
- Intercepting communication
- Card cloning
- Making recommendations to blue team for security improvements

Red team skills

- **Software development:** When you know how applications are built, you’re better able to identify their possible weaknesses (as well as write your own programs to automate the attack process).
- **Penetration testing:** Much of a red team’s job is to identify and try to exploit known vulnerabilities on a network. This includes familiarity with vulnerability scanners.
- **Social engineering:** An organization’s biggest vulnerability is often its people rather than its computer network. Social engineering tactics like phishing, baiting, and tailgating can sometimes be the easiest way past security defenses.
- **Threat intelligence and reverse engineering:** Knowing what threats are out there—and how to emulate them—can make you a more effective attacker.
- **Creativity:** Finding ways to beat a blue team’s defenses often requires creating new and innovative forms of attack.

II Blue team

NIST defines a **blue team** as “the group responsible for defending an enterprise’s use of information systems by maintaining its security posture against a group of mock attackers.” If the red team is playing offense, the blue team is playing defense to protect an organization’s critical assets.

Blue team activities

As a blue team member, it’s your job to analyze the current security posture of your organization and take measures to address flaws and vulnerabilities. Playing for the blue team also means monitoring for breaches and responding to them when they do occur. Some of these tasks include:

- Digital footprint analysis
- DNS audits
- Installing and configuring firewalls and endpoint security software
- Monitoring network activity
- Using least-privilege access

Blue team skills

Defending a company against attack involves understanding what assets need to be protected and how to best protect them. Here are some skills that could serve you well in a blue team role:

- **Risk assessment:** Risk assessment helps you identify key assets that are most at risk for exploitation so you can prioritize your resources to protect them.
- **Threat intelligence:** You’ll want to know what threats are out there so you can plan appropriate defenses. Blue teams have to stay a step ahead of attackers.
- **Hardening techniques:** Recognizing weaknesses in your organization's security is only helpful if you know the techniques for fixing them.
- **Monitoring and detection systems:** As a blue team professional, you’ll need to know how to use packet sniffers, security and information event management (SIEM) software, intrusion detection systems (IDS), and intrusion prevention systems (IPS).

Q7. (a) How do you find vulnerability in your android application? What are the common vulnerabilities?

[What Is Mobile Application Security and How Does It Work? | Synopsys](#)

- Interacting with the application and understanding how it stores, receives, and transmits data.
- Decrypting encrypted parts of the application.
- Decompiling the application and analysing the resulting code.
- Using static analysis to pinpoint security weaknesses in the decompiled code.
- Applying the understanding gained from reverse engineering and static analysis to drive dynamic analysis and penetration testing.
- Utilizing dynamic analysis and penetration testing to evaluate the effectiveness of security controls (e.g., authentication and authorization controls) that are used within the application.

The OWASP Mobile Top 10 at a Glance

[OWASP Mobile Top 10 - More Security for Mobile Applications | turingpoint](#)

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.

Q7. (b) What are the essential benefits we can realize with the adoption of DevOps principals, describe the sample tools used at various DevOps life cycle stages.

Benefits of DevOps

DevOps proponents describe several business and technical benefits, many of which can result in happier customers. Some benefits of DevOps include:

- Faster, better product delivery
- Faster issue resolution and reduced complexity
- Greater scalability and availability
- More stable operating environments
- Better resource utilization
- Greater automation
- Greater visibility into system outcomes
- Greater innovation

The following shows a sample of tools used at various DevOps lifecycle stages.

- **Plan.** This phase helps define business value and requirements. Sample tools include Jira or Git to help track known issues and perform project management.
- **Code.** This phase involves software design and the creation of software code. Sample tools include GitHub, GitLab, Bitbucket, or Stash.
- **Build.** In this phase, you manage software builds and versions, and use automated tools to help compile and package code for future release to production. You use source code repositories or package repositories that also “package” infrastructure needed for product release. Sample tools include Docker, Ansible, Puppet, Chef, Gradle, Maven, or JFrog Artifactory.
- **Test.** This phase involves continuous testing (manual or automated) to ensure optimal code quality. Sample tools include JUnit, Codeception, Selenium, Vagrant, TestNG, or BlazeMeter.
- **Deploy.** This phase can include tools that help manage, coordinate, schedule, and automate product releases into production. Sample tools include Puppet, Chef, Ansible, Jenkins, Kubernetes, OpenShift, OpenStack, Docker, or Jira.
- **Operate.** This phase manages software during production. Sample tools include Ansible, Puppet, PowerShell, Chef, Salt, or Otter.
- **Monitor.** This phase involves identifying and collecting information about issues from a specific software release in production. Sample tools include New Relic, Datadog, Grafana, Wireshark, Splunk, Nagios, or Slack.

Q8. (a) Describe the shared Responsibility Model in cloud.

The **Shared Responsibility Model** is a security and compliance framework that outlines the responsibilities of **cloud service providers (CSPs)** and **customers** for securing every aspect of the cloud environment, including hardware, infrastructure, endpoints, data, configurations, settings, operating system (OS), network controls and access rights

Direct Control

While the Shared Responsibility Model is based on the idea that two or more parties play a role in ensuring security of distinct elements within the public cloud environment, it is important to note that the customer and CSP do not share responsibility for the same asset.

Rather, the CSP or the customer has full and complete responsibility for the security of all assets under their direct control, regardless of the service model type.

Customers are typically also responsible for:

- Identity Access and Management (IAM)
- User security and credentials
- Endpoint security
- Network security
- Security of workloads and containers
- Configurations
- APIs and middleware
- Code

Cloud Service Providers (CSPs)—such as Amazon, Microsoft or Google—are responsible for areas for which they possess direct control. This typically includes security of:

- The physical layer and all associated hardware and infrastructure
- The virtualization layer
- Network controls and provider services
- Facilities that run cloud resources

Q8. (b) What is the need of Dynamic Code Analysis? Which stage of secure SDLC it is applied?

This form of code analysis is essential, as it tests the code in real-life scenarios. Unexpected errors caused by interaction with multiple application functions are hard, or even impossible to find using static analysis. These errors only become obvious during the integration of various components or interaction with the whole system on deployment. Therefore, a dynamic analysis should be performed once the software is functionally complete. Additionally, doing dynamic analysis will:

- Traditional pre-release security testing is no longer enough for modern web application development
- Modern DAST automates application security testing and integrates it into agile software development workflows
- Shifting left with accurate dynamic testing is the only way to build scalable web application security and move towards DevSecOps

- Allow testers to perform application analysis without having access to the actual code.
- Reveal errors that can crash the program.
- Help testers ensure that the product/software works well.
- Help quality enhancement by taking into consideration any drawbacks.
- Require less expertise to perform; therefore, it is less expensive than static code analysis. Static code analysis requires an expert in the language in which the application has been developed.

Although security tests are carried out at every step, the fourth phase of the SDLC is the testing-only phase where rigorous assessments and analyses, such as the Dynamic Code Analysis, a kind of an Application Security Testing, also called the Open Web Application Security Project, are carried out.

Q9. (a) Describe Android Application Security architecture.

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 –

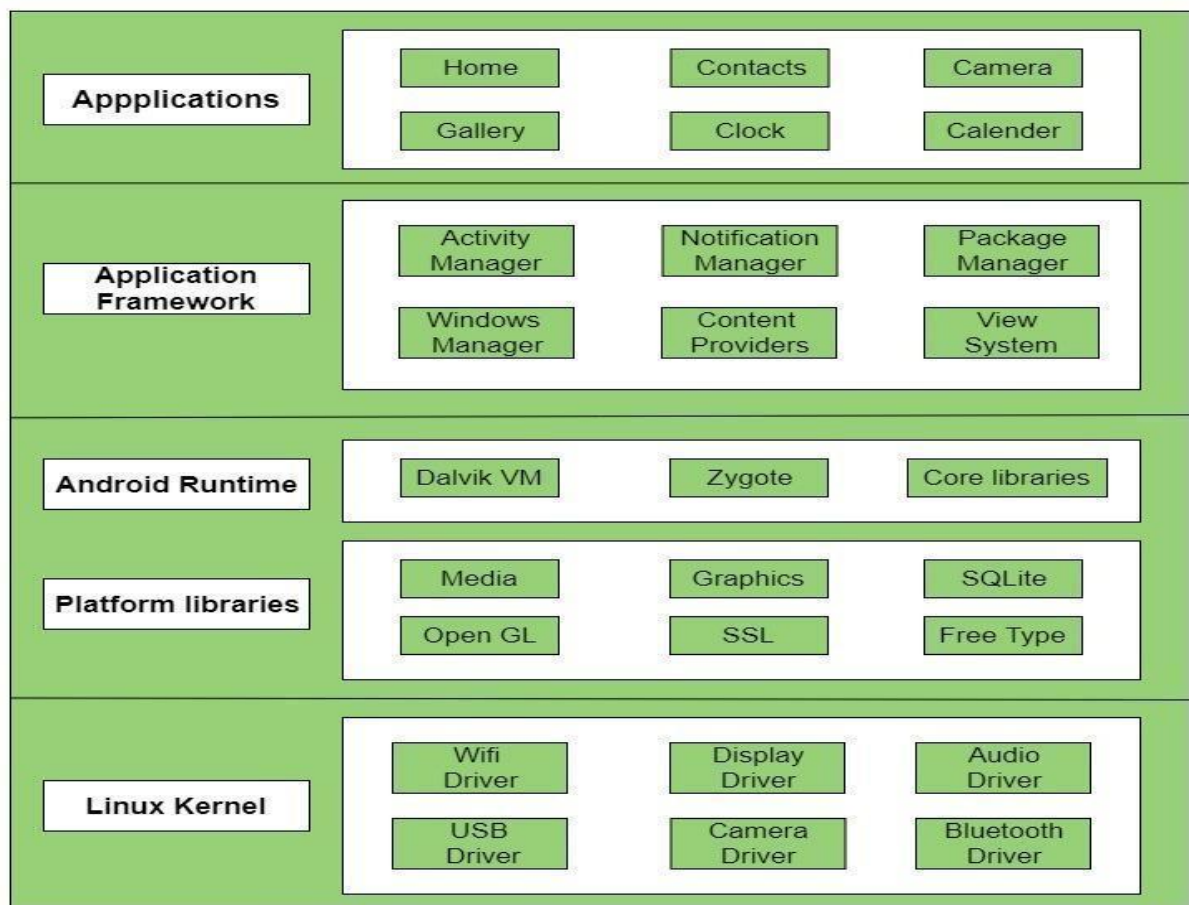
Linux Kernel is heart of the android architecture. It manages all the available drivers such as display drivers, camera drivers, Bluetooth drivers, audio drivers, memory drivers, etc. which are required during the runtime.

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.

The features of Linux kernel are:

- **Security:** The Linux kernel handles the security between the application and the system.
- **Memory Management:** It efficiently handles the memory management thereby providing the freedom to develop our apps.
- **Process Management:** It manages the process well, allocates resources to processes whenever they need them.
- **Network Stack:** It effectively handles the network communication.
- **Driver Model:** It ensures that the application works properly on the device and hardware manufacturers responsible for building their drivers into the Linux build.

Pictorial representation of android architecture with several main components and their sub components –



Q9. (b) Why Is WAF (Wireless Application Firewall) Security Important? Give its types and features.

[What is WAF | Types, Security & Features Explained | Imperva](#)

1. WAFs are important for a growing number of organizations that offer products or services online—this includes mobile app developers, social media providers, and digital bankers.
2. A WAF can help you protect sensitive data, such as customer records and payment card data, and prevent leakage.
3. WAF can help you meet compliance requirements such as PCI DSS (the Payment Card Industry Data Security Standard), which applies to any organization handling cardholder data and requires the installation of a firewall.
4. A WAF is thus an essential component of an organization's security model.
5. It is important to have a WAF, but it is recommended you combine it with other security measures, such as intrusion detection systems (IDS), intrusion prevention systems (IPS), and traditional firewalls, to achieve a defense-in-depth security model.

Types of Web Application Firewalls

- **Network-based WAF**—usually hardware-based, it is installed locally to minimize latency. However, this is the most expensive type of WAF and necessitates storing and maintaining physical equipment.
- **Host-based WAF**—can be fully integrated into the software of an application. This option is cheaper than network-based WAFs and is more customizable, but it consumes extensive local server resources, is complex to implement, and can be expensive to maintain. The machine used to run a host-based WAF often needs to be hardened and customized, which can take time and be costly.
- **Cloud-based WAF**—an affordable, easily implemented solution, which typically does not require an upfront investment, with users paying a monthly or annual security-as-a-service subscription. A cloud-based WAF can be regularly updated at no extra cost, and without any effort on the part of the user.

Web application firewalls typically offer the following features and capabilities:



Attack signature databases

Attack signatures are patterns that may indicate malicious traffic, including request types, anomalous server responses, and known malicious IP addresses. WAFs used to rely predominantly on attack pattern databases that were less effective against new or unknown attacks.



AI-powered traffic pattern analysis

Artificial intelligence algorithms enable behavioural analysis of traffic patterns, using behavioural baselines for various types of traffic to detect anomalies that indicate an attack. This allows you to detect attacks that don't match known malicious patterns.



Application profiling

This involves analyzing the structure of an application, including the typical requests, URLs, values, and permitted data types. This allows the WAF to identify and block potentially malicious requests.



Customization

Operators can define the security rules applied to application traffic. This allows organizations to customize WAF behaviour according to their needs and prevent the blocking of legitimate traffic.



Correlation engines

These analyze incoming traffic and triage it with known attack signatures, application profiling, AI analysis, and custom rules to determine whether it should be blocked.



DDoS protection platforms

You can integrate a cloud-based platform that protects against distributed denial of service (DDoS) attacks. If the WAF detects a [DDoS attack](#), it can transfer the traffic to the DDoS protection platform, which can handle a large volume of attacks.



Content delivery networks (CDNs)

WAFs are deployed at the network edge, so a cloud-hosted WAF can provide a CDN to cache the website and improve its load time. The WAF deploys the [CDN](#) on several points of presence (PoPs) that are distributed globally, so users are served from the closest PoP.

Q10. (a) Give two KRI examples each for the following domains:

a. Patch Management

b. Anti-virus management

a. Patch Management

Here are five Key Risk Indicators (KRIs) for patch management:

Time to Patch: This measures the time elapsed between the release of a security patch and the completion of its installation across all systems. A long time to patch can indicate a lack of efficiency in the patch management process and increase the risk of exploitation of known vulnerabilities.

Percentage of Patched Systems: This measures the proportion of systems that have been updated with the latest security patches. A low percentage of patched systems increases the risk of exploitation of known vulnerabilities and can have serious consequences for the security of an organization's assets.

Patch Failure Rate: This measures the rate at which patches fail to install correctly. High patch failure rates can indicate issues with the patch management process, such as compatibility problems, and can increase the risk of exploitation of known vulnerabilities.

Patch Compliance: This measures the extent to which the organization's patch management policies and procedures are being followed. Low patch compliance can indicate a lack of discipline in the patch management process and increase the risk of exploitation of known vulnerabilities.

Number of Critical Patches: This measures the number of critical security patches that have been released but not yet installed. A high number of uninstalled critical patches increases the risk of exploitation of known vulnerabilities and can have serious consequences for the security of an organization's assets.

b. Anti-virus management

Here are five Key Risk Indicators (KRIs) for Anti-virus management:

Time to Detection: This measures the time elapsed between the appearance of a new malware threat and the detection of that threat by the anti-virus software. A long time to detection can indicate a lack of efficiency in the anti-virus management process and increase the risk of successful malware attacks.

Percentage of Malware-Infected Systems: This measures the proportion of systems that have been infected with malware. A high percentage of infected systems can indicate a failure of the anti-virus management process and increase the risk of data loss or theft.

False Positive Rate: This measures the rate at which the anti-virus software identifies benign files as malware. High false positive rates can indicate that the anti-virus software is overly aggressive and can cause productivity losses by blocking legitimate files.

Signature Update Compliance: This measures the extent to which anti-virus software is updated with the latest malware definitions. Low update compliance can indicate a lack of discipline in the anti-virus management process and increase the risk of successful malware attacks.

Number of Undetected Threats: This measures the number of malware threats that have been discovered but not yet detected by the anti-virus software. A high number of undetected threats can indicate that the anti-virus management process is not effective and increase the risk of successful malware attacks.

[ITIL incident management process: 8 steps with examples \(manageengine.com\)](https://manageengine.com/blog/itil-incident-management-process-8-steps-with-examples/)

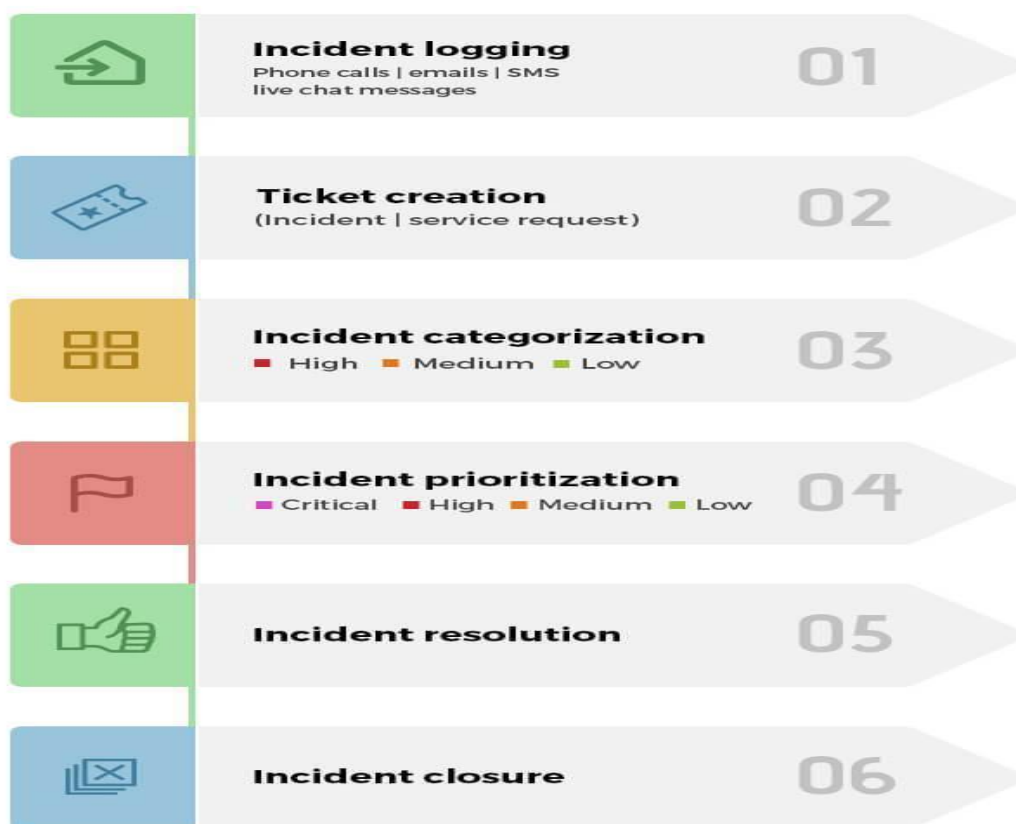
Q10. (b) Describe the IT Incident management life cycle/ Process flow.

The incident management process can be summarized as follows:

- **Step 1:** Incident logging.

- **Step 2** : Incident categorization.
- **Step 3** : Incident prioritization.
- **Step 4** : Incident assignment.
- **Step 5** : Task creation and management.
- **Step 6** : SLA management and escalation.
- **Step 7** : Incident resolution.
- **Step 8** : Incident closure.

These processes may be simple or complex based on the type of incident; they also may include several workflows and tasks in addition to the basic process described above.



Incident logging

An incident can be logged through phone calls, emails, SMS, web forms published on the self-service portal or via live chat messages.

Incident categorization

Incidents can be categorized and sub-categorized based on the area of IT or business that the incident causes a disruption in like network, hardware etc.

Incident prioritization

The priority of an incident can be determined as a function of its impact and urgency using a priority matrix. The impact of an incident denotes the degree of damage the issue will cause

to the user or business. The urgency of an incident indicates the time within which the incident should be resolved. Based on the priority, incidents can be categorized as:

Critical High Medium Low

Incident routing and assignment

Once the incident is categorized and prioritized, it gets automatically routed to a technician with the relevant expertise.

Creating and managing tasks

Based on the complexity of the incident, it can be broken down into sub-activities or tasks. Tasks are typically created when an incident resolution requires the contribution of multiple technicians from various departments.

SLA management and escalation

While the incident is being processed, the technician needs to ensure the SLA isn't breached. An SLA is the acceptable time within which an incident needs response (response SLA) or resolution (resolution SLA). SLAs can be assigned to incidents based on their parameters like category, requester, impact, urgency etc.

Incident resolution

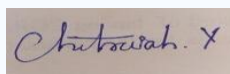
An incident is considered resolved when the technician has come up with a temporary workaround or a permanent solution for the issue.

Incident closure

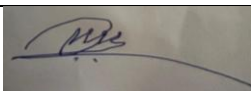
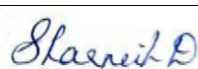
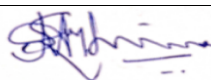
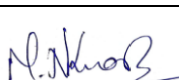
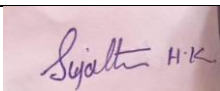
An incident can be closed once the issue is resolved and the user acknowledges the resolution and is satisfied with it.

CERTIFICATE

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



CHITRASHEKHARAIAH. Y
LECTURER CSE,
GPT RAICHUR-117

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