The following scenario relates to **Questions** **1.**

A colleague of yours is designing an application that will enable distributed teams to collaborate remotely on authoring documents online using a documents management and sharing system. They have heard of secure design principles such as: Least Privileges, Separation of Duties, and Defense in Depth. The colleague turns to you for help with incorporating these design principles into their design.

**Question 1**

Describe the following secure design principles:

(a) Least privileges (**5 marks**)

*A security principle in which a person or process is given only the minimum level of access rights (privileges) that is necessary for that person or process to complete an assigned operation.*

*This right must be given for a minimum amount of time that is necessary to complete the operation*

*Limits the damage in case of exploited vulnerability*

*There is need for proper granularity of privileges and permissions to apply this principle*

(b) Separation of duties (**5 marks**)

*Also known as the compartmentalization principle, or separation of privilege. Separation of duties is a security principle which states that the successful completion of a single task is dependent upon two or more conditions that need to be met and just one of the conditions will be insufficient in completing the task by itself.*

(c) Defense in depth (**5 marks**)

*Also known as layered defense, Defense in depth is a security principle where single points of complete compromise are eliminated or mitigated by the incorporation of a series or multiple layers of security safeguards and risk-mitigation countermeasures.*

*Have diverse strategies, so that if one layer of defense turns out to be inadequate, another layer of defense will hopefully prevent a full breach.*

*Example*

*For SQL injection mitigation a layered security needs to be followed:*

*Validate all the tainted inputs for bad/danger characters*

*Use of Parameterised prepared statements through out the application*

*Use low privileged database account for all DB interactions*

*Can be further enhanced by role based access accounts etc.*

(d) Choose two design principles (from a, b and c, above) and describe what sort of attacks they would help mitigate in the scenario above.

(**5 marks, 5 marks**)

*Least privileges mitigates: session hijacking, misuse of privileges, insider threats, accidental deletion of information, elevation of privilege attacks, etc*

*Separation of duties mitigates: session hijacking, non-repudiation, etc*

*Defense in depth mitigates: multi stage attacks, can prevent total breach, can buy time for security personnel to apply patches by slowing attackers, etc*

**Question 2**

Using suitable examples, describe the following secure development principles.

(a) Canonicalisation (**5 marks**)

*Canonicalisation (sometimes called standardisation or normalisation) is a process for converting data that has more than one possible representation into a “standard”, “normal”, or canonical form.*

*The best way to describe canonicalization is to remember that it stems from canon, meaning an authentic piece of writing. What they're talking about is taking untrusted data and formatting it as an unambiguous representation, such that any software process can never misrepresent it*.

*This could be achieved using encoding schemes or some sanitisation techniques.*

*Plus suitable example(s)*

(b) Output encoding (**5 marks**)

*Encoding, closely related to escaping is a powerful mechanism to help protect against many types of attacks, especially injection attacks and Cross-site Scripting (XSS).*

*Essentially, encoding involves translating special characters into some equivalent that is no longer significant in the target interpreter.*

*To encode properly, you need to know what interpreters the data might end up in*

*You want to make sure that you encode all the characters that might cause a problem.*

*One of the best approaches is to use a positive encoding scheme, where all characters except a minimal known good set are encoded*

*Plus suitable example(s)*

(c) Error encoding (**5 marks**)

*Error messages are one of the first sources an attacker will look to determine information about an application.*

*Without proper handling of input and the response generated from that input in the form of an error message, sensitive information can be leaked.*

*Validate all input to prevent an attacker from forcing an error by using an input (type, value, range, length, etc.) that the software is not expecting.*

*The error messages must be non-verbose and explicitly specified in the software*

*For example, when a user tries to access a file that does not exist, the error message typically indicates, “file not found”*

*When accessing a file that the user is not authorised for, it indicates, “access denied”.*

*The user is not supposed to know the file even exists, but such inconsistencies will readily reveal the presence or absence of inaccessible files or the site’s directory structure*

*Plus suitable example(s)*

(d) Session management (**5 marks**)

*Just because someone is authenticated and authorised to access system resources does not mean that security controls can be lax after authenticated session is established, because a session can be hijacked.*

*Session hijacking attacks happen when an attacker impersonates the identity of a valid user and interjects themselves into the middle of an existing session, routing information from the user to the system and from the system to the user through them.*

*Plus suitable example(s)*

(e) Auditing and logging (**5 marks**)

*Logging usually means recording various events that happen within an application.*

*Focus should not only be on the interests of the developer*

*Auditing is about recoding specific events usually with specific parameters.*

*Sometimes there is a legal obligation to record events*

*Note that these events can become of interest for information disclosure if too much information is being logged.*

*Have automated systems that scan the logs for suspicious activities. It is pointless to accumulate lots of logs without anyone looking at them.*

*Plus suitable example(s)*

**Question 3**

Injection flaws are a major security risk to web applications.

(a) Describe what injection flaws are in general from a web application security point of view. (**5 marks**)

*Injection flaws occur when an attacker sends specific crafted data, usually malicious codes, to a trusted web application and results in unintended actions returned by the trusted web application. The impact of injection flaws could vary depending on the malicious codes. If the malicious codes were past to an operating system as system calls or shell scripts, they can result in executing unauthorized commands. If the malicious codes were past to a database as SQL commands, they can result in the access of information that is not intended to be shown. For example, on a customer directory search page, an attacker can input specific data in the html form that asks for customer name and trick a web server to reveal the entire content of the customer database.*

(b) With the aid of examples, describe what the following injection flaws are:

(i) Cross-Site Scripting (XSS) (**5 marks**)

*Cross Site Scripting (XSS) is a type of injection flaw that an attacker injects malicious codes through either the URL or the user inputs of a web page, resulting in the redirection of other naive users’ web pages to a malicious web site, usually without the awareness of the naive users.*

*There are many forms of XSS. An attacker injects malicious contents into a trusted web site as user inputs and the malicious contents are stored in the trusted web site’s database. Once a naive user accessed the malicious contents posted on the trusted web site, such as by clicking a link posted on the trusted web site or by reading the user messages posted on a guest book, the trusted web site would either redirect the user to the attacker’s web site or to run a malicious script program on the naive user’s computer. In additional to injecting malicious contents into a trusted web site, an attacker can also send a naive user mails with an URL pointing to a trusted web site but embedded with malicious scripts.*

*The goal of XSS attacks is usually compromising users‟ privacy or information confidentiality. XSS is often used as a technique for phishing or cookie stealing. Once naive users tricked to access the attacker’s site or to run a malicious script, they are usually asked to reveal their confidential information, such as user name, password, or credit card information. Since the naive users may think that they are communicating with the trusted web site, they would inevitably give their information to the malicious site.*

(ii) SQL Injection (**5 marks**)

*SQL injection is an attack that exploits the vulnerability of the invalidated user input and reveals the content of the data that are not supposed to be seen by an attacker. For example, in order to search for the information of a customer “Smith” , the SQL command past from the web application to the database is below:*

*SELECT \* FROM user\_data WHERE name=‟Smith‟*

*In this example, the table name in the database is user\_data and the attribute name for searching is name. A user Smith can type in his name and search for his account information. However, an attacker may type in Smith’ OR ‘1’=’1, which will result in the following SQL past to the database:*

*SELECT \* FROM user\_data WHERE name = „Smith’ OR ‘1’=’1‟*

*Since the logical statement, 1=1, is always true, the database will return will all user data in the table user\_data. The ramification for this attack is to examine user inputs for invalid inputs, such as the one in the example. User input validation can be done at either the web application layer or the database layer.*

(c) Describe some of the countermeasures that could be taken to mitigate the security risks caused by:

(i) Cross-Site Scripting (XSS) (**5 marks**)

*Countermeasures to XSS can be categorized as server-side solutions and client-side solutions. Server- side solutions are implemented on the web applications to detect malicious codes injected by attackers and client-side solutions are used to prevent naive users from running malicious URL or scripts on the user site.*

(ii) SQL Injection (**5 marks**)

*User input validation can be done at either the web application layer or the database layer.*

*At the web application layer, input validation is done by comparing user inputs with either an acceptable list of inputs (whitelisting) or a list of potential malicious values (blacklisting). Whitelisting is more restricted on user inputs but blacklisting can create loopholes.*

*At the database layer, stored procedures can be used to validate inputs. Stored procedures are subroutines that are called upon by the database once associated data attributes are accessed. To validate user inputs, application developers have to write stored procedures for the data attributes that are related to the inputs and set up constraints for accessing these data attributes.*

**Question 4**

Describe how the following factors have exacerbated the security issues surrounding web applications:

(a) Frameworks **(5 marks)**

*Frameworks*

*Use of several third party packages in most of our web applications means that there are many parts of the applications that we do not fully understand and which we are not fully in control of.*

*Developers assume a lot about frameworks’ security*

*Frameworks abstract too much which means anyone can build a web application even people with no knowledge of security.*

(b) The web being dynamic and not static **(5 marks)**

*This is one of the major issues*

*Users can send arbitrary input, which exposes applications to injection attacks*

*Because the web is dynamic there are many active components that can execute code which means there are many parts for the hackers to attack*

(c) Internet of Things **(5 marks)**

*Connectivity from IoT means it is almost impossible for anyone to have an inventory of processes or systems that interface with their web applications or online systems.*

*Widens attack surface*

*Internet of the unpatchables made of gadgets made with no security in mind is a big concern.*

*There are currently no international standards on IoT components, a lot of them use weak protocols, weak encryption and have no security in them*

*A lot of IoT gadgets are manufactured in places and by people where security is not a priority*

(d) Use of passwords for authentication **(5 marks)**

*Users use weak passwords because they are easy to remember*

*Users reuse and recycle passwords on many online applications*

*Users write passwords down for easy of remembering*

*Users can be convinced to share passwords with strangers unknowingly*

(e) Resources e.g. time and money **(5 marks)**

*Shortage of trained security personnel is a big issue*

*Lack of developers with adequate security training*

*When money or time runs out on an IT project security is usually deprioritized because it is usually not one of the features on the requirements spec.*

*Highly trained security professionals are expensive to hire or contract*

*A lot of smaller companies lack the resources to procure or setup secure systems*

**Question 5**

(a) With the aid of a diagram, describe the basic layout and elements of a symmetric cryptographic system. **(8 marks)**



*5 marks for diagram and 3 marks for description*

*On the left side, generates 2 keys and shares one with intended receivers. The sender uses an encryption algorithm together with a key to encrypt some text*

*In the middle the text travels through a public network*

*On the right the receive uses a decryption algorithm and the key shared by the sender to decrypt the message*

(b) Describe the main differences between symmetric and asymmetric cryptographic systems. **(8 marks)**

*For symmetric systems the key is considered to be the ‘same’ for both the encryption and decryption processes –*

*The understanding is that the keys are so closely related that knowledge of one implies knowledge of the other.*

*In contrast, asymmetric cryptography uses two keys, which while related, are such that knowledge of the encipherment key does not give knowledge of the decipherment key.*

*For asymmetric systems the keys are different but clearly they need to be related in some way for the system to work*

*But it is the case that the relationship between the two keys makes it simply too difficult to determine one from the other.*

(c) A lot of people assume that using SSL guarantees security. However, describe any three weaknesses with SSL. **(9 marks)**

*There are lots of attacks which target encryption*

*Unfortunately it does not stop attacks that directly target the server nor client components of an application as most successful applications do.*

*It does not prevent any of vulnerabilities which matter,*

*it just makes sure transmitted messages are encrypted.*

*Or many others that can render an application critically exposed to attack.*

*It can be fooled to use null cipher or reduced encryption*