# What is a vulnerability and how is it related to a bug?

In cyber security, a vulnerability is a weakness can be exploited by a cyber attack to gain unauthorized access to or perform unauthorized actions on a computer system. Vulnerabilities can allow attackers to run code, access a systems’ memory, install malware, and steal, destroy or modify sensitive data.

**National Institute of Standards and Technology (NIST):** Weakness is an information system, system security procedures, internal controls, or implementation that could be exploited or triggered by a threat source.

**Vulnerability is a subset of bug**

A bug is any defect in a product.

A vulnerability is bug that manifests as an opportunity for malicious use of the product. Vulnerabilities generally are not that clearly evident, but require ingenuity to be exploited.

# Discuss the scope/guidelines of the secure software development.

The Security Development Lifecycle (SDL) is a software development security assurance process consisting of security practices grouped by six phases: training, requirements & design, construction, testing, release, and response.

The basic task of security requirement engineering is to identify and document actions needed for developing secure software systems. Security elements of the SDLC must include:

• Planning to meet security requirements and goals.

• Threat modelling.

• Design to include security and privacy concerns.

• System architecture (e.g., web, applications, user interfaces, programmatic interfaces, file import/export, reports, databases).

• Documentation.

• Change management (See IS-3, III).

• Testing, including creating test plans, reviewing test results and confirming fixes and patches.

• Secure deployment practices and separation of duties.

Regarding code review, Units developing software that will process, store or transmit Institutional Information classified at Protection Level 3 or higher or Availability Level 3 or higher must:

● Perform code reviews to reduce cyber risk.

● Consider and check for common security mistakes.

● Include in the review process a senior software developer and, if possible, choose an independent one.

● Include in the review process an IT Workforce Member with specific security experience.

● Use automated secure code testing/checking tools:

o Perform static code analysis.

o Perform dynamic code analysis.

# Discuss some of the factors that lead to software insecurity.

Complexity, inadequacy, and changes in the software’s processing model (e.g., a web – or server – oriented architecture model)

Incorrect assumptions by the engineer, including assumptions about the capabilities, outputs, and behavioural states of the software’s execution environment or about expected inputs from external entitles (users, software processes).

Flawed specification or design, or defective implementation of

* The software’s interfaces with external entitles. Development mistakes of this type include inadequate (or non-existent input validation, error handing, and exception handling.
* The components of the software’s execution environment (from middleware – level and operating – system – level to firmware – and hardware – level components)

Poor implementation of software interfaces (input validation, error and exception handling.)

Inadequate knowledge of secure coding practices

Unintended, unexpected interactions

* With other components
* With the software’s execution environment

Absent or minimal consideration of security during all lifecycle phase

# Describe what the perimeter security is and why it is considered as a myth

Perimeter security is the philosophy of setting up functional apparatus or techniques at the perimeter of the network to secure data and resources. It is part of the greater security field and has its own role in active system protection.

As the cloud becomes more utilized, perimeter security will become less effective. And due to more advanced threats, we must make sure that our data, software, and networks are protected by more than just perimeter security measure. While perimeter security acts as an organization’s first line of defence, it is not the one true answer to security.

# Why cannot we rely only on the security-related automated tools for achieving secure software?

The security – related automated tools will just identify the common coding problems (implementation bugs) automatically, before a problems is released. So if a rule has not yet been written in the tool, it will never find that problem.

Also tools have limited impact:

* Black box security testing only works for web applications because http protocol is stateless and simple.
* Code review tool only look for bugs in code written in certain programming languages.
* Code review tools are unable to identify architectural flaws

# What are the main limitations of the penetration testing technique?

* The ‘reformed hackers’ could be malicious and hide some of the bugs (finding) they found during the penetration testing.
* Security issues often involve unexpected misuse of an application discovered by an attacker
* It is difficult to show whether or not a system is secure enough under malicious attack. How many tests do we need to do before giving up and declaring ‘secure enough’?

# Describe the myth that says Software security is a cryptography problem.

Security is a system property, not a thing. Cryptography can be a useful tool when it comes to securing data and communications but it is no silver bullet.

# Are the developers the only people responsible for securing the software systems they develop? Why?

No. Software security is the responsibility of everyone but it must be coordinated by a central software security group (SSG).

# Give an example where using JavaScript carelessly can give rise to a security vulnerability.

<FORM ACTION="submit\_order" METHOD="GET" NAME="f">

How many pizzas would you like to order?

<INPUT TYPE="text" NAME="qty" VALUE="1" onKeyUp="computePrice();">

<INPUT TYPE="hidden" NAME="price" VALUE="5.50"><BR>

<INPUT TYPE="submit" NAME ="Order" VALUE="Pay">

<INPUT TYPE="submit" NAME ="Cancel" VALUE="Cancel">

</FORM>

<SCRIPT>

function computePrice() {

f.price.value = 5.50 \* f.qty.value;

f.Order.value = "Pay £" + f.price.value

}

</SCRIPT>

# What is the definition and purpose of CVE, CVSS and CWE? Give some examples of the CWE/SANS top 25 most dangerous software errors.

CWE: Common weakness enumeration

CWE-119 Improper Restriction of Operations within the Bounds of a Memory Buffer

CWE-79 Improper Neutralization of Input During Web Page Generation ('Cross-site Scripting')

CWE-20 Improper Input Validation

CWE-200 Information Exposure

CWE-125 Out-of-bounds Read

# What is the difference between POST and GET?

Both GET and POST method is used to transfer data from client to server in HTTP protocol but Main difference between POST and GET method is that GET carries request parameter appended in URL string while POST carries request parameter in message body which makes it more secure way of transferring data from client to server in http protocol.

# HTTP protocol is a stateless protocol. Describe the different options and their drawbacks to keep the client state registered in the server side.

* the use of HTTP cookies.
* server side sessions,
* hidden variables (when the current page contains a form)
* URL-rewriting using URI-encoded parameters, e.g., /index.php?session\_id=some\_unique\_session\_code.