

JUNE 17 REPORT

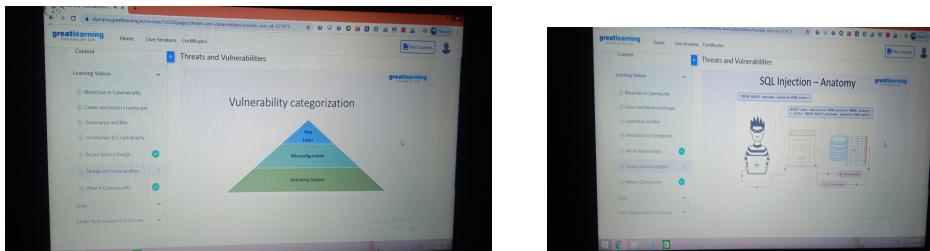
Date:	17/06/20	Name:	ANKITHA C C
Course:	Cyber security	USN:	4AL16EC004
Topic:	1. Vulnerabilities & Password Security 2. What is Cryptography? 3. Message integrity	Semester & Section:	8TH & A
Github Repository:	ankitha-course		

FORENOON SESSION DETAILS

Image of session

Report - Report can be typed or hand written for up to two pages.

1. Vulnerabilities & Password Security



In computer security, a vulnerability is a weakness which can be exploited by a threat actor, such as an attacker, to perform unauthorized actions within a computer system. To exploit a vulnerability, an attacker must have at least one applicable tool or technique that can connect to a system weakness. In this frame, vulnerabilities are also known as the attack surface.

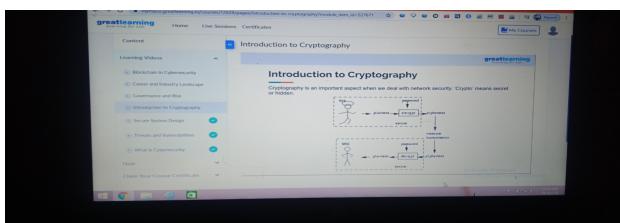
Vulnerability management is the cyclical practice of identifying, classifying, remediating, and mitigating vulnerabilities.^[1] This practice generally refers to software vulnerabilities in computing systems.

A security risk is often incorrectly classified as a vulnerability. The use of vulnerability with the same meaning of risk can lead to confusion. The risk is the potential of a significant impact resulting from the exploit of a vulnerability. Then there are vulnerabilities without risk: for example when the affected asset has no value. A vulnerability with one or more known instances of working and fully implemented attacks is classified as an exploitable vulnerability—a vulnerability for which an exploit exists. The window of vulnerability is the time from when the security hole was introduced or manifested in deployed software, to when access was removed, a security fix was available/deployed, or the attacker was disabled—see zero-day attack.

Security bug (security defect) is a narrower concept: there are vulnerabilities that are not related to software: hardware, site, personnel vulnerabilities are examples of vulnerabilities that are not software security bugs.

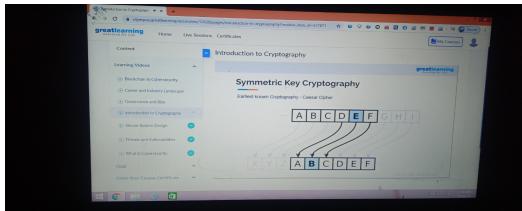
Constructs in programming languages that are difficult to use properly can be a large source of vulnerabilities. Your ongoing security program. Equally, significant changes to your organisational structure may require another evaluation of testing strategy to ensure it meets your evolving business needs.

2. Cryptography



Cryptography, or cryptology (from Ancient Greek: κρυπτός, romanized: kryptós "hidden, secret"; and γράφειν graphein, "to write", or -λογία -logia, "study", respectively), is the practice and study of techniques for secure communication in the presence of third parties called adversaries. More generally, cryptography is about constructing and analyzing protocols that prevent third parties or the public from reading private messages; various aspects in information security such as data confidentiality, data integrity, authentication, and non-repudiation are central to modern cryptography. Modern

cryptography exists at the intersection of the disciplines of mathematics, computer science, electrical engineering, communication science, and physics. Applications of cryptography include electronic commerce, chip-based payment cards, digital currencies, computer passwords, and military communications.

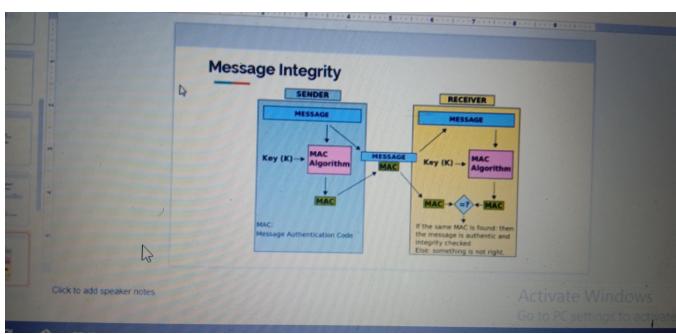


Cryptography prior to the modern age was effectively synonymous with encryption, the conversion of information from a readable state to apparent nonsense. The originator of an encrypted message shares the decoding technique only with intended recipients to preclude access from adversaries. The cryptography literature often uses the names Alice ("A") for the sender, Bob ("B") for the intended recipient, and Eve ("eavesdropper") for the adversary. Since the development of rotor cipher machines in World War I and the advent of computers in World War II, the methods used to carry out cryptology have become increasingly complex and its application more widespread.

Modern cryptography is heavily based on mathematical theory and computer science practice; cryptographic algorithms are designed around computational hardness assumptions, making such algorithms hard to break in practice by any adversary. It is theoretically possible to break such a system, but it is infeasible to do so by any known practical means. These schemes are therefore termed computationally secure; theoretical advances, e.g., improvements in integer factorization algorithms, and faster computing technology require these solutions to be continually adapted. There exist information-theoretically secure schemes that provably cannot be broken even with unlimited computing power—an example is the one-time pad—but these schemes are more difficult to use in practice than the best theoretically breakable but computationally secure mechanisms.

The growth of cryptographic technology has raised a number of legal issues in the information age. Cryptography's potential for use as a tool for espionage and sedition has led many governments to classify it as a weapon and to limit or even prohibit its use and export. In some jurisdictions where the use of cryptography is legal, laws permit investigators to compel the disclosure of encryption keys for documents relevant to an investigation. Cryptography also plays a major role in digital rights management and copyright infringement of digital media.

3.message integrity



The screenshot shows a course interface for 'Introduction to Cryptography'. On the left, there's a sidebar with navigation links like 'Home', 'Live Sessions', 'Certificates', 'Content', 'Learning Videos', and 'Quiz'. The main area displays a conversation between Alice and Bob. Alice sends a message 'We are under quarantine' and its MAC. Bob receives the message and MAC, and the system shows the decrypted message 'We are still working under quarantine'.

In cryptography, a message authentication code (MAC), sometimes known as a tag, is a short piece of information used to authenticate a message—in other words, to confirm that the message came from the stated sender (its authenticity) and has not been changed. The MAC value protects both a message's data integrity as well as its authenticity, by allowing verifiers (who also possess the secret key) to detect any changes to the message content.

Informally, a message authentication code system consists of three algorithms:

A key generation algorithm selects a key from the key space uniformly at random.

A signing algorithm efficiently returns a tag given the key and the message.

A verifying algorithm efficiently verifies the authenticity of the message given the key and the tag. That is, return accepted when the message and tag are not tampered with or forged, and otherwise return rejected.

For a secure unforgeable message authentication code, it should be computationally infeasible to compute a valid tag of the given message without knowledge of the key, even if for the worst case, we assume the adversary can forge the tag of any message except the given one.

Formally, a message authentication code (MAC) system is a triple of efficient Algorithms (G, S, V) satisfying:

G (key-generator) gives the key k on input 1^n , where n is the security parameter.

S (signing) outputs a tag t on the key k and the input string x .

V (verifying) outputs accepted or rejected on inputs: the key k , the string x and the tag t .

S and V must satisfy the following:

$$\Pr[k \leftarrow G(1^n), V(k, x, S(k, x)) = \text{accepted}] = 1.$$

A MAC is unforgeable if for every efficient adversary A

$$\Pr[k \leftarrow G(1^n), (x, t) \leftarrow AS(k, \cdot)(1^n), x \notin \text{Query}(AS(k, \cdot), 1^n), V(k, x, t) = \text{accepted}] < \text{negl}(n),$$

where $AS(k, \cdot)$ denotes that A has access to the oracle $S(k, \cdot)$, and $\text{Query}(AS(k, \cdot), 1^n)$ denotes the set of the queries on S made by A , which knows n . Clearly we require that any adversary cannot directly query the string x on S , since otherwise a valid tag can be easily obtained by that adversary.

