UNIT 13: FUNDAMENTALS OF WEB APPLICATION SECURITY AUDIT

13.0 INTRODUCTION

In the previous units, we talked about web services in general, XML security, HTML5 security, and collecting and analysing logs for web apps. In the present unit, we will apprise you of the fundamentals of web application security auditing. It will explain the key concepts and techniques used in web application security audits, including threat modeling, vulnerability scanning, penetrating testing, and reporting. You will learn about the common types of vulnerabilities created in web applications and how to identify and mitigate them. Due to the continued increase in the number of websites every day, the security aspect of these sites becomes equally concerning. The unit concluded by highlighting the best practises for security web applications and protecting sensitive data.

13.1 LEARNING OUTCOMES

After studying this unit, you will be able to:

a) explain the fundamentals of web application security auditing.
b) apply modeling exercise to identify potential vulnerabilities in a web application
c) explain OWASP's top 10 attacks, and
d) understand the audit system process, and create a comprehensive report outling the findings and recommentrations for seucirting the web application

13.2 OVERVIEW OF THE WEB APPLICATION SECURITY AUDIT

Web application security is all about making sure that websites, web apps, and web services are safe. It will help us learn about the gaps and how to prevent these types of attacks in the future. Users will learn how to eradicate them and increase web application security. A hacker's attack might jeopardise your company's financial resources. Regular audits will reduce the danger of hackers attempting to exploit your software. Auditing prevents data breaches and increases customer loyalty. Customers that are disappointed are likely to express their opposing viewpoints with their peers. Web application audits can help you to avoid data breaches. As a consequence, you'll know how to fix web application vulnerabilities. There are several common types of web application security threats that web application security audits aim to address. These are:

- Injection attacks: Injection attacks occur when an attacker injects malicious code into an application, such as SQL injection or cross-site scripting (XSS) attacks.
- Broken authentication and session management: This occurs when authentication mechanisms are weak, allowing attackers to gain unauthorized access to the application or user data.
- Cross-site request forgery (CSRF): This type of attack occurs when a user is tricked into performing an action on a website without their knowledge or consent.
- Security misconfigurations: Security misconfigurations can occur when the application is not configured correctly, leaving it open to attacks.

13. 3 TYPES OF WEB APPLICATION SECURITY TESTING

The digitization revolution across the globe leads to hackers and cyberattacks. Hackers are developing more and more innovative techniques to evade already established security standards. Therefore, regular web security testing or auditing helps you avoid vulnerabilities.

There are different types of web application security testing methods that are used. For example,

13.3.1 Dynamic Application Security Testing (DAST*)*

This sort of testing is intended to determine the vulnerabilities an attacker targets and how they could get access to the system.

13.3.2 Static Application Security Testing (SAST)

SAST checks the source code of a web application for vulnerabilities in a more inside-out manner. A real-time snapshot of web application security can be beneficial.

13.3.3 Application Penetration Testing

Professionals play an important part in application penetration testing. He or she will attempt to find out how an attacker would get access to a web application.

(https://www.rapid7.com/fundamentals/web-application-security-testing).

Other methods for web application security testing are being used, such as threat modeling. It involves identifying potential threats to the web application and determining the level of risk associated with each threat. A comprehensive report is created outlining the findings and recommendations for securing the web application. In addition, users must (i) identify flaws and vulnerabilities in applications, (ii) abide by laws, (iii) analyse the present security system of the organisation, (iv) detect security breaches and anomalous behaviour, and (v) develop an effective security plan.

13.4 UNDERSTANDING THREAT MODELLING

Threat modeling is a structured approach to identifying potential threats to a system or application and evaluating the risks associated with those threats. The goal of threat modeling is to identify potential vulnerabilities and weaknesses in the system's design, architecture, and implementation, so they can be addressed proactively. It involves the following steps:

- Identify the assets: The first step in threat modeling is to identify the assets that need to be protected. This might include sensitive data, critical infrastructure, or intellectual property.
- Define the application's architecture: Once the assets have been identified, the application's architecture must be defined. This includes identifying the components and interactions within the application, such as the user interface, database, and network connections.

- Identify potential threats: With the application's architecture defined, potential threats can be identified. This might include threats such as injection attacks, denial-of-service attacks, or social engineering attacks.
- Analyze the threats: Once the threats have been identified, they must be analyzed to determine their potential impact on the system. This involves evaluating the likelihood of the threat occurring and the potential damage that could result.
- Mitigate the risks: Based on the analysis, steps can be taken to mitigate the risks associated with the identified threats. This might include implementing additional security controls, changing the application's design or architecture, or improving employee training and awareness.

It is an important component of a comprehensive security strategy, as it provides a structured approach to identifying and addressing potential security risks before they can be exploited by attackers. By proactively identifying vulnerabilities and weaknesses in the system's design and implementation, organisations can better protect their critical assets and data.

Check Your Progress 1

Note: a) Space is given below for writing your answer.

b) Check your answers with the one given at the end of this Unit.

(i) Why should we need web application security audit?

.....

.....

.....

.....

.....

(ii) Describe different types of web application security testing.

.....

.....

.....

.....

.....

13.5 WEB SERVICE SECURITY-RELATED STANDARDS

There are several security-related standards that are in different stages in special key areas. It has changed from time to time with the technological development and its uses. Some of them are presented below:

- XML encryption: The W3C group has defined an XML encryption specification.
- XML Digital Signature: The W3C/IETF group defined an XML digital signature specification.
- XML Key Management: To perform key management such as initial registration and revocation.
- OASIS Web Security TC: This group defines how to sign and encrypt a SOAP message in order to build a foundation for higher-level security services.
- Web Service Interoperability Organisation: This group has been defining a security profile to ensure basic interoperability among vendors (Aissi, S. et.al., 2006).
- Oracle considered interoperability of Web services platforms to be important than providing support for all edge cases of the Web service specifications.

For details, see Basic Security Profile 1.0 Specification at http://www.ws-i.org/Profiles/BasicSecuirtyProfile-1.0.html.

13.6 KEY AREAS FOR WEB APPLICATION SECURITY AUDIT

The key areas for web application security audit stages are presented in this section. There are various approaches are being adopted for security audits. It may take physical and non physical forms. For example, in December, 2004, security managers conducted a security audit at one of the corporate headquarters in Windsor, Connecticut. It was applied on twenty-one mail and each of the mail was having processing facilities. A security audit was designed to evaluate the controls that were implemented through the integrated security management system. Before starting the web application security audit, the user should discuss the scope of his work with his company's team and schedule the key areas for activities to be followed: (i) verification of vulnerabilities in web applications; (ii) verification of the detected vulnerabilities; and (iii) reporting on the work to be performed. New vulnerabilities may emerge at any time that were not previously successful. As mentioned in the previous courses, cyber threats are evolving but reliant on previous attacks. Hence, there is a need for continuous research on the latest threats trends and techniques that are being used by the hackers.

Recently, there has been significant work in the area of web service security technology. The industry plays a significant role in helping to standardise and highlight security issues. Hence, web application security audits have achieved great importance and are becoming popular among enterprises and software specialists. There are various reasons for any institution to carry out security audits at their respective organizations. The first audit performed by an organisation or company may become base for future company's audits. Therefore, every organisation asks the questions: Does Secure Sockets Layer (SSL) protect all communication? What about injection attacks and Cross-Site Scripting (XSS)? For web site security, you should have a software development life cycle (SDLC), host, monitor network, and application-based logs.

The frequency of hacker attacks is increasing day by day. As reported by the University of Maryland, that there is a hacker attack in every 39 seconds (https://eng.umd.edu/news). In addition, 43% target small businesses, and 64% of companies have experienced web-based attacks. Generally, we adopt measures to protect the system by adopting the following methods:

- Threat modelling: identify what to protect.
- Install a web application firewall.
- Segregate data based on the threat model.
- Fix critical applications.
- Monitor security.
- Plan for incident response.
- Improve your development process.

Monitoring of Security

- Maintain logs.
- Server, Apache, and security logs.
- Enable logging on the web server.
- Set up log rotation.
- Keep backups of logs read your logs daily.
- Alerts and summaries: swatch and log watch.

Check Your Progress 2

Note: a) Space is given below for writing your answer.

b) Check your answers with the one given at the end of this Unit.

(i) Discuss the web service security-Related Standards

……

……

……

……

……

……

……

(ii) List the k**ey** areas for Web Application Security Audit

……

……

……

……

……

……

13.7 WEB APPLICATION SECURITY AUDIT FOR ISO-27001 AUDITING

Web application Security Audit for ISO-27001: 2003 was designed to support standard specifications for web application security audit. A checklist for each control was available as a set of actions applied during the auditing. The purpose for the use of this approach was to allow the information security to be used in companies of different size, and with distinct technological developments. Antunes et.al. (2022) reported that it helps to the monitor the information security and cyber security risks. It was observed during the auditing process of fifty intervening SMEs. This type of information security mechanism was designed for the use of auditor so that he may use the application for the company to be audited. They also proposed an auditing process with evidence given by the consulting team, the automatic processing of the auditing report and the result analysis. The web application requires regular audit process.

It can be stored in a database which starts with auditing records, regular updates and delivery of reports along with result analysis. The application functions with three layers: web access through web browser; MySQL database and application layer.
In the knowledge society, the audit process plays a vital role in insuring a high level of information security quality.

Cyber attackers are always coming up with new ways to break through a company's security. Sometimes, following an industry is not the answer. study analyses how the use of machine learning approaches increases the discovery of vulnerable attacks in customer relationship management and finds that it has a significant influence on applications created in regulated environments (Stewart, 2022).

13.8 WEB SERVICE SECURITY AUDITS INITIATIVES

As mentioned in the previous units, that sometimes, organisation data becomes more useful for their future plans. Hence, it is important that this useful and sensitive information not be retrieved by other organisation, particularly competitors. Evolving laws on the protection of private data will be helpful for the consumer. Hence, there is always a need for security audits. The European Union Data Protection Act, US privacy law, and Canada's personal Information Protection law that have already been placed. There is no common approach adopted across the globe for web application security audit can be used information systems auditing. It has been varied from case to case. Some Web Application Security audits initiatives have been taken place globally and are presented below:

- Top down approach: Generally, user adopts this approach to tackle the problem from computer science perspective. Therefore, users have to identify particular problems for security audits as needed. For example, network providers would look at the auditing from the perspective of the network infrastructure, while operating system providers would look at it from the perspective of the operating system. In these cases, we have to adopt a top-down approach to auditing.
- Some organisation adopted an audit approach at the *outside of the network*. They started with a port scan from a computer connected outside of the organisation. The user worked their way inward i.e. technologically and organisationally.
- A *wireless networking audit* was used in another organisation. The organisation developed IEEE802.11b WLANs which consist of various digital access points, signal amplifiers and wireless NIC cards. Laptops were equipped with built-in wireless NIC

interfaces. The access points were connected computer to the internal network of the organisation.

- There were different approaches were being adopted to check on the *security of network passwords,* like telephone checks. This test was used to know how easy it would be for an outsider to get *user's password* by asking him directly. It was found that 75% provided their passwords. Hence, it becomes necessary for the awareness of the information security programmes (Lo and Marchand, 2004).
- The Web Services Interoperability Organisation (WS-1) is another organisation introduced the initial version of the WE-1 Basic Security profile. The main objective of WS-1 is to state what and how web security specifications should be applied to achieve interoperability degree.

13.9 FUNDAMENTAL OF WEB APPLICATION SECURITY AUDIT

Information systems are complex in nature and were developed to resolve problems in companies. It has resulted from business relationships with third parties like customers and partners. The audit can be done internally or externally.

A web application security audit includes all the auditing activities for specifications, projects, databases, hardware, and software, including input and output and life cycle programming etc. It was developed as a result of technological system penetration. In some of the organisation, achievement of technological systems, programming languages, programme techniques are being audited. The security of web applications is compromised when web applications are targeted to sensitive customer and business organisation data. It is quite possible; some organisation management has taken decision to purchase and installs software without the knowledge of users. Sometime, it didn't meet the security requirement. These organisations may expose to risk from insecure web application, if they download and install application from unknown web/internet sources. This type of web application security is the outcome from a deficiency of compliance with laws, regulations and policies applicable to any organisation and its information.

As mentioned in the previous courses that the security vulnerabilities of the web application are classified as per details given below:

- Cross Site Scripting (XSS)
- Injection Flaws
- Cross Site Request Forgery

- Malicious File Execution
- Failure to Restrict URL Access
- Information Leakage and Improper Error Handling
- Application Runtime Configuration
- Insecure Direct Object Reference
- Insecure Cryptographic Storage
- Insecure Commutation

To overcome from the above mentioned security vulnerabilities of the web application, the organisation risk can be reduced by analysing the source code of web applications for the common security vulnerabilities. Mostly web vulnerabilities are classified in coding errors and input validation. In addition, it may also be classified as unbounded parameters and encoding, design flaws etc. The audit process detect of the security vulnerabilities in an information system based on web application.

13.10 OWASP TOP WEB APPLICATION SECURITY RISKS

To specialise in web application security assessments, it is important to update yourself with new software, tools, languages, or platforms. It can be implemented with industry experts through professional training and continuous interaction or networking with professionals. For example, the following resources will be useful:

The Open Web Application Security (OWASP) project foundation is an independent non-profit body. It has been supported by volunteers and developing application security programmes (www.owasp.org).There are many ways to compromise and breach applications. Security threat is frequently changing with time as presented in the Figure 4.1.

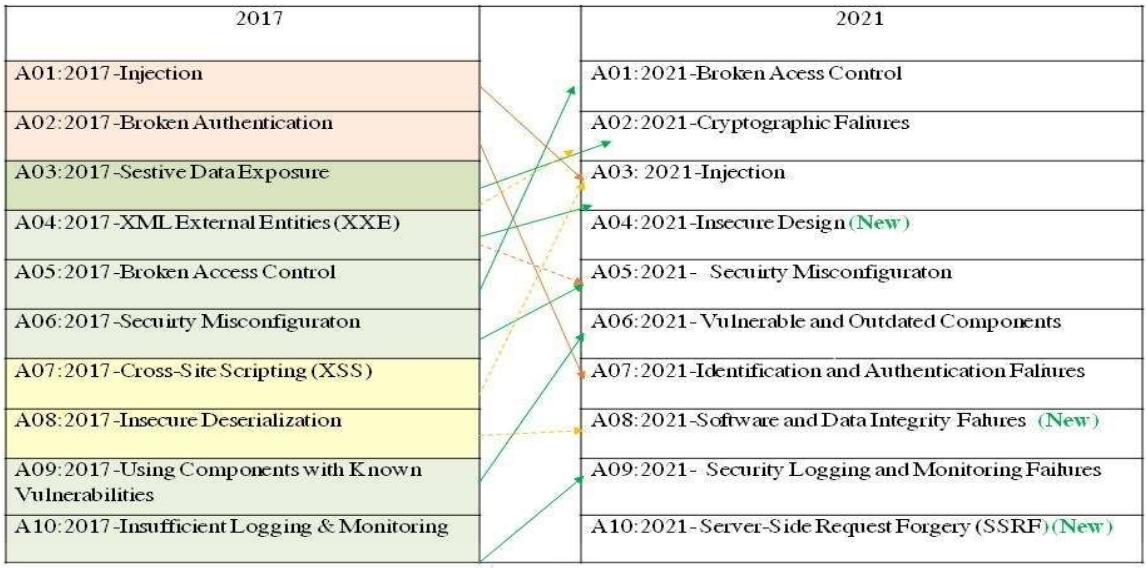

Figure 4.1: OWASP top Ten 2021
Source: Compiled from https://owasp.org/www-project-top-ten/

Various vulnerability scanners have been set up to identify the above-mentioned vulnerabilities. The vulnerability scanner is not very effective because it creates sometimes false information. Therefore, information security experts have to check the attack to see what flaws need to be corrected to make the system more secure. It can be achieved by following the above-mentioned OWASP Top 10, which will help you know how the attackers attack and how to defend against it, etc. But it takes more time and requires more resources. In this method, vulnerability scanners only detect the vulnerabilities, which is considerable, but their automatic mitigation is not possible.

Popat et al. (2021) designed an algorithm and developed a method to detect the vulnerability using Python nodules. They used proof-based vulnerability scanning to identify the vulnerabilities in the system and confirm the identified issues. An open-source web application vulnerability scanner that established proof of the vulnerability was used. For instance, if the user detects a SQL injection vulnerability, it will explain the database name as the proof of the exploit. Major security risks as prioritised by OSASP's Top Ten in 2021 are presented below:

A01:2021-Broken Access Control

Figure 4.1 reveals that the top ten places shifted from 2017 to 2021. Broken Access Control shifted from fifth place to first place in 2021 with a 55.97% maximum incidence rate. During 2021, more than 90% of applications were assessed for some type of broken access control. It appears the most in the contributed dataset, with over 318k instances.

OWASP stated ‘CWE-200: Exposure of Sensitive Information to Unauthorized Actor, CWE-201: Insertion of Sensitive Information Into Sent Data, and CWE-352: Cross-Stie Request Forgery' are among the Common Weakness Enumerations (CWEs) featured.’ Only trustworthy server-side or server-less APIs, where the attacker cannot alter the access control check or metadata, are effective for access control (https://owasp.org/Top10/A01_2021-Broken_Access_Control).

A02:2021-Crptographic Failures

It has been shifted up from third to second position with 46.44% of the maximum incidence rate (previously known as "A03: 2017-Sensitive Data Exposure"), which was a broad system rather than a root cause as stated in the OSASP project. It needs data protection, i.e., passwords for online transactions, personal information, business protection, etc., generally if the data falls under privacy laws. (https://owasp.org/Top10/A02_2021-Cryptographic_Failures).

A03:2021-Injection

A03:2021—Injection has been ranked third with a highest incidence rate of 19.09 percent. Over ninety percent of the applications were screened for injection. The 33 CWEs mapped into this category have the second most occurrences in applications with 274k occurrences. One of the reasons applications become vulnerable to attack is when an application does not check, clean, or validate user supply data before using it, in addition to the other reasons as mentioned on the OSASP project page (https://owasp.org/Top10/A03_2021-Injection).

A04:2021-Insecure Design

This kind of category emerged in 2021 and focuses on risks related to design flaws with a 24.19% maximum incidence rate. For industry, more threat modelling, secure design patterns and principles, and conference architectures are needed. It represented different weaknesses, expressed as “missing or ineffective control design” (https://owasp.org/Top10/A04_2021-Insecure_Design).

A05:2021-Security Misconfiguration

A05:2021-Security Misconfiguration shifted from fifth to sixth place during the said duration, with a 19.84% maximum incidence rate. Ninety percent of applications were tested for some form of misconfiguration. A4:2017-XML External Entities (XXE) is part of this risk category. If any part of the application stack, lacks adequate security hardening, the application may be vulnerable (https://owasp.org/Top10/A05_2021-Security_Misconfiguration).

A06:2021-Vulnerable and Outdated Components

This category moves up from ninth place in 2017 to sixth place in 2021 with issues that need to be tested and assessed for risk. It has a maximum incidence rate of 27.96%. A user may become a victim if they have unsupported or out-of-date software. Don't scan for vulnerabilities regularly. For more reasons, you can visit https://owasp.org/Top10/A06_2021-Vulnerable_and_Outdated_Components.

A07:2021- Identification and Authentication Failures

It was previously known as Broken Authentication. Its position decreased from second to seventh. This is an integral part of the Top 10 with maximum incidence rate (14.84%). Authentication weaknesses possibility were there, if the application allows automated attacks such as credential stuffing, in which the attackers have a list of valid usernames and passwords(https://owasp.org/Top10/A07_2021-Identification_and_Authentication_Failures).

A08:2021-Software and Data Integrity Failures

It was a new category for 2021 of OWASP top ten. A08:2017 -Insecure Deserialization in 2017 is now a part of this category. It has maximum incidence rate (16.67%) and concentrate on establishing assumptions about software updates, essential data, and CI/CD pipelines without testing their integrity. The attacker has the ability to upload their own updates, which will then be disseminated and run on all systems.. For more examples and details see (https://owasp.org/Top10/A08_2021-Software_and_Data_Integrity_Failures)

A09-Security Logging and Monitoring Failures

It was earlier known as A10:2017-Insufficient Logging & Monitoring and include more types of failures with maximum incidence rate (19.23%) and known as A09-Suecurity Logging and Monitoring failures. Its position shifted from 10th in 2017 to 9th place in 2021. It can directly

impact visibility, incident alerting, and forensics. It cannot be possible to detect Breaches without logging and monitoring (https://owasp.org/Top10/A09_2021-Security_Logging_and_Monitoring_Failures).

A10:2021-Server-Side Request Forgery

This has been rated as having a relatively low incidence rate (2.72%) and represents the scenario where the security members are informing. It occurs When a web application tries to get a remote resource without checking the URL given by the user. (https://owasp.org/Top10/A10_2021-Server-Side_Request_Forgery_%28SSRF%29).

For the latest updates, you should study The Web Application Hacker's Handbook: Finding and Exploiting Security Flaws by Dafydd Stuttard and Marcus Pinto (https://www.wiley.com). This resource update web application is the primary source for most of the organization, in addition to previous courses in your programme. It will apprise you of the latest step-by-step techniques for attacking and defending the range of constantly changing web applications.

Check Your Progress 3

Note: a) For writing answers space is given below.

b) Check your answers with the one given at the end of this Unit.

(i) List the OWASP top 10 attackers

……

……

……

……

……

……

……

13.11AUDIT SYSTEM PROCESS AND PROTECTION OF WEB APPLICATION

As we know, systems cannot be totally secure. Hence, it becomes necessary to audit the system's processes. It will help prevent other users from entering the system for unauthorised logging. While analysing the log files, we can know the method used by the attacker for hacking information. It can be used to design more resistant systems. It is rightly said that if you “know your enemies and know yourself," you will not be imperilled in a hundred battles (https://en.wikiquote.org/wiki/Sun_Tzu). The auditing process's goal is to provide the ability to determine:

(a) Whether were you attacked or not?

(b) How were you attacked?

(c) When were you attacked?

(d) What was attached?

In addition to the above mentioned purposes, auditing has also been done for traffic analysis, statistics and weather prediction.

Normally logging includes information such as time of event, initiating process or owner of process. What to audit might be exceptionally critical to the overall security process. Analyse of the web server log files to get right scrutiny system usage may give better results.

What to audit and how to audit?

A web application is a software programme that runs in a browser on the user’s machine. It is convenient to provide access to other users without installing it on their systems. It leads to the server side, which doesn’t control the processes. Hence, they become victims of injection, inadequate authentication, exposure of data, and poor security configuration. Therefore, the following is needed for the audit:

- Audit of Web Apps
- Audit of Internet Infrastructure
- Audit of APIs and Web Services
- Audit of Internet
- Audit in Redteam mode as real hacker (create a real attack in order to test the global security level)

How to protect Web Application?

To protect the web application, the following steps are suggested:

- As a web developer, you should make more of an effort to protect and secure your web application, as suggested by OWASP.

- XXE cannot be permitted by classifying and identifying sensitive data as per privacy laws, or business requirement, etc.
- Another flaw in web applications' security is *SQL injection*. Attackers used it to have direct access to the database.
- If you use JSON instead of XML, the risk of an SML External Entities (XXE) attack is reduced.
- Applications are vulnerable to attack when they run with known vulnerable components and with insufficient logging and monitoring. It requires the removal of unused dependencies and effective monitoring, alerting, and log generation.
- Broken access control, security misconfiguration, cross-site scripting (XSS) and insecure deserialization are solutions for common attacks.
- Logging and continuous monitoring provide visibility into events and behaviours to help detect and respond to malicious activities quickly.
- The user should use layered security mechanisms, which increase the security of the system as a whole. For instance, a web server is compromised by a formerly undisclosed vulnerability.
- The user should choose an open design that ensures that security is well built. Open design will help your system's security, which shouldn't rely on the secrecy of its implementation.
- It should have established monitoring procedures for logging all security events. For example, an application that requires authentication has failed 1000 times in two minutes from a single Internet Protocol address. This may be an attack from the hackers. It is also called a "misconfigured device.
- The web server should not trust the Real-Time Transport Protocol server (the minimum privilege used).
- While using a positive security model (what is allowed and rejects everything else), it should be ensured security.
- While using third-party code in open-source projects, it needs to be monitored. Suppose, if users are using package tools, they use their warning system to take the necessary precautions.

- It should be ensured that appropriate access is given to the web application. Authentication and authorization steps will ensure the right user has access to the web application.
- To ensure the security of the user's application, it is preferable to seek help from a specialised person at the time of urgency so that an essential issue can be taken care of by him.
- You need to plan for handling security issues with your application, data encryption, and compliance requirements.
- Web developer is the first person who can catch application security vulnerabilities and their training need to be plan regularity for latest developments. It will help them to update latest ways to catch and resolve application security issues.

Each of these steps has its strengths and weaknesses, and the best approach will depend on the specific needs and requirements of the organization. A comprehensive security testing strategy will often incorporate multiple methodologies to provide a more complete assessment of the application's security location. Securing web applications is critical to protecting sensitive data and ensuring business continuity. There are some best practises for securing web applications. For example, keep software up-to-date, use strong authentication, implement access controls, encrypt sensitive data, implement security testing, use secure coding practices, and monitor for suspicious activity regularly. As a user, you follow these best practices, and your organisation can reduce the risk of security breaches and protect sensitive data and systems from being compromised.

13.12 LET US SUM UP

This unit discusses the fundamentals of web application security auditing. Due to the continued increase in the number of websites every day, the security aspect of these sites becomes equally concerning. In view of this threat, the unit highlights the basic conceptual framework for web application security audits, types and fundamentals of web application security audits, the OWASP top 10 attacks, and the audit system process. Later issues related to web application auditing as well as its measure were presented for your basic understanding.

13.13 CHECK YOUR PROGRESS: THE KEY

1. (i) Auditing process of web services helps us to verify the threat of your application that emerges by hacker's attack. As a user, you will learn how to eliminate them and

improve web application security level. The auditing process of web services helps us verify the threat to your application that emerges from a hacker's attack.. As a user, you will learn how to eliminate them and improve web application security. The auditing process of web services helps us verify the threat to your application that emerges from a hacker's attack. Another benefit of auditing is gaining the customer's loyalty by preventing data leakage. The web application audit process will help the business avoid such unnecessary information leakages.

(ii) **Ref. section 13.3**

2. (i) -XML encryption: The W3C group has defined an XML encryption specification.

-XML Digital Signature: The W3C/IETF group defined an XML digital signature specification..

-XML Key Management: To perform key management such as initial registration and revocation.

- OASIS Web Security TC: This group defines how to sign and encrypt a SOAP message in order to build a foundation for higher-level security services.

- Web Service Interoperability Organisation: This group has been defining a security profile to ensure basic interoperability among vendors (Aissi, S. et al., 2006).

(ii) (a) Threat modelling: Identify what to protect

a) Install a Web Application Firewall
b) Segregate data based on threat model
c) Fix critical applications
d) Monitor Security
e) Plan for incident response
f) Improve your development process

Monitoring of Security

a) Maintain logs
b) Server, Apache, and Security logs
c) Enable logging in web server
d) Set up log rotation
e) Keep backups of logs read your logs daily
f) Alerts and summaries: swatch and log watch (Source: James Walden (2009) Web security essentials for universities)

3. Refer to **section 1.9**

13.14 REFERENCES AND FURTHER READINGS

Aissi, S., Dabbous, N., and Prasad, A. (2006). Security for Mobile Networks and Platforms, Boston/London: Artech House

Antunes, M., Maximiano, M. and Gomes, R. (2022). A customizable web platform to manage standards compliance of information security and cybersecurity auditing, Procedia Computer Science, Science Direct (www.sciencedirect.com)

Information Systems Security Association (ISSA). (2017). Best practices for web application security. Retrieved from https://www.issa.org/resource/best-practices-for-web-application-security/

National Institute of Standards and Technology (NIST). (2017). NIST Special Publication 800-115: Technical Guide to Information Security Testing and Assessment. Retrieved from https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-115.pdf

Popa, M. (2009). Detection of the security vulnerabilities in Web Applications, *Informatica Economica*, 13 (1).

Popat, J.M., Nawab, M.A., Kumar, A.R., Prakash, N. and Manjula M. (2021). Web application security-Automating the manual exploitation methods and eliminating false positives, International Journal of Advance Research, Ideas and Innovation in Technology, 7 (4).

Open Web Application Security Project (OWASP). (2021). OWASP Top Ten. Retrieved from https://owasp.org/Top10/

Stewart, H. (2022) Security versus compliance: An empirical study of the impact of industry standards compliance on application security, *International Journal of Software Engineering and Knowledge Engineering*, 32 (3), p-363-393.https://doi.org/10.1142/S0218194022500152

Lo and Marchand (2004). Security Audit: A case study CCECE (2004- CCGEI). Retrieved from https://ieeexplore.ieee.org/stamp/stamp.jsp?arnumber=1349612

Microsoft. (2021). Threat modeling: A structured approach to security. Retrieved from https://docs.microsoft.com/en-us/azure/security/develop/threat-modeling

SANS Institute. (2019). Best practices for securing web applications. Retrieved from https://www.sans.org/white-papers/40725.

Websites References

https://www.softwaretestinghelp.com/penetration-testing-guide
https://www.dfs.ny.gov/consumers/alerts/equifax_data_breach
https://dea.gov.in/sites/default/files/Press-CERT-Fin%20Report.pdf
https://openai.com/blog/chatgpt
https://scholar.google.com/
https://www.bing.com/
https://www.java.com/en/
www.owasp.org
http://www.ws-i.org/Profiles/BasicSecuirtyProfile-1.0.html
https://eng.umd.edu/news