Securing Web Applications

Securing Web Applications

APTECH LIMITED
Contact E-mail: ov-support@onlinevarsity.com

Edition 1 - 2014

Dear Learner,

We congratulate you on your decision to pursue an Aptech course.

Aptech Ltd. designs its courses using a sound instructional design model – from conceptualization to execution, incorporating the following key aspects:

- Scanning the user system and needs assessment

Needs assessment is carried out to find the educational and training needs of the learner

Technology trends are regularly scanned and tracked by core teams at Aptech Ltd. TAG* analyzes these on a monthly basis to understand the emerging technology training needs for the Industry.

An annual Industry Recruitment Profile Survey# is conducted during August - October to understand the technologies that Industries would be adapting in the next 2 to 3 years. An analysis of these trends & recruitment needs is then carried out to understand the skill requirements for different roles & career opportunities.

The skill requirements are then mapped with the learner profile (user system) to derive the Learning objectives for the different roles.

- Needs analysis and design of curriculum

The Learning objectives are then analyzed and translated into learning tasks. Each learning task or activity is analyzed in terms of knowledge, skills and attitudes that are required to perform that task. Teachers and domain experts do this jointly. These are then grouped in clusters to form the subjects to be covered by the curriculum.

In addition, the society, the teachers, and the industry expect certain knowledge and skills that are related to abilities such as *learning-to-learn, thinking, adaptability, problem solving, positive attitude etc*. These competencies would cover both cognitive and affective domains.

A precedence diagram for the subjects is drawn where the prerequisites for each subject are graphically illustrated. The number of levels in this diagram is determined by the duration of the course in terms of number of semesters etc. Using the precedence diagram and the time duration for each subject, the curriculum is organized.

- Design & development of instructional materials

The content outlines are developed by including additional topics that are required for the completion of the domain and for the logical development of the competencies identified. Evaluation strategy and scheme is developed for the subject. The topics are arranged/organized in a meaningful sequence.

The detailed instructional material – Training aids, Learner material, reference material, project guidelines, etc.- are then developed. Rigorous quality checks are conducted at every stage.

- Strategies for delivery of instruction

Careful consideration is given for the integral development of abilities like thinking, problem solving, learning-to-learn etc. by selecting appropriate instructional strategies (training methodology), instructional activities and instructional materials.

The area of IT is fast changing and nebulous. Hence considerable flexibility is provided in the instructional process by specially including creative activities with group interaction between the students and the trainer. The positive aspects of web based learning –acquiring information, organizing information and acting on the basis of insufficient information are some of the aspects, which are incorporated, in the instructional process.

- Assessment of learning

The learning is assessed through different modes – tests, assignments & projects. The assessment system is designed to evaluate the level of knowledge & skills as defined by the learning objectives.

- Evaluation of instructional process and instructional materials

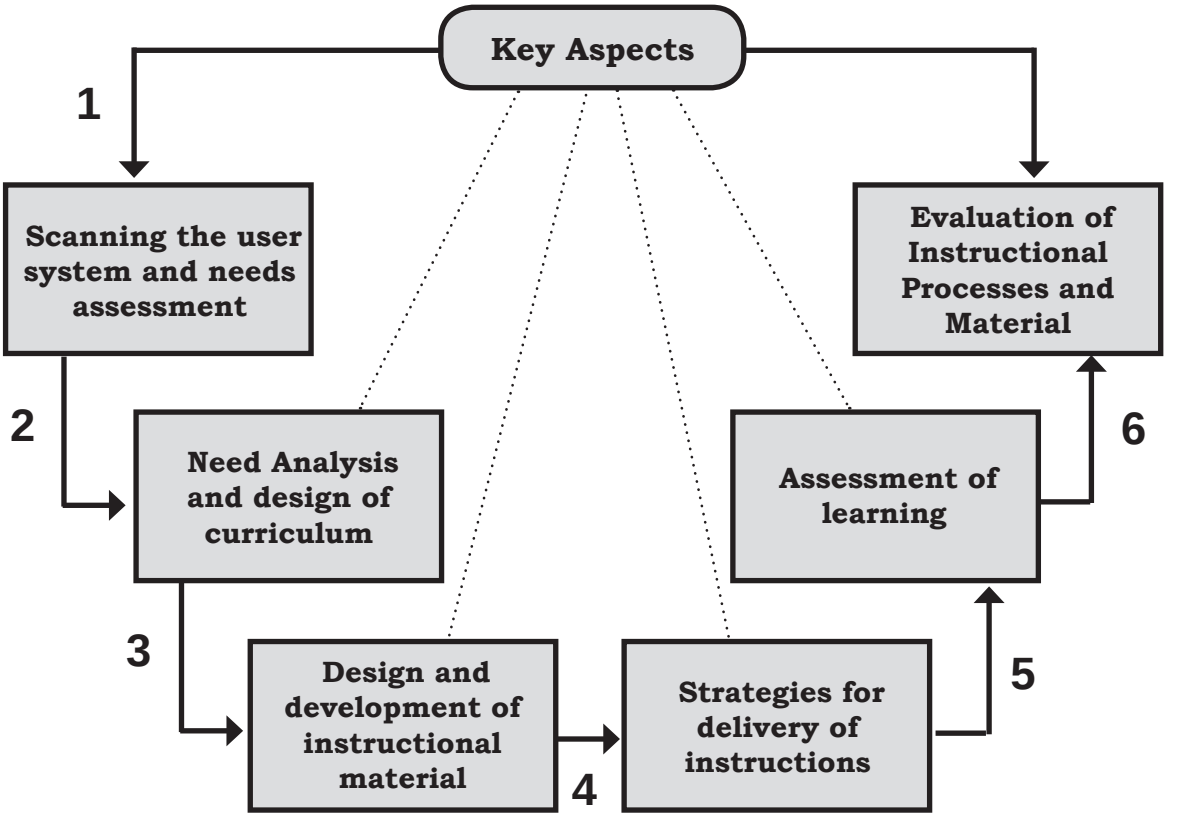
The instructional process is backed by an elaborate monitoring system to evaluate - on-time delivery, understanding of a subject module, ability of the instructor to impart learning. As an integral part of this process, we request you to kindly send us your feedback in the reply pre-paid form appended at the end of each module.

*TAG – Technology & Academics Group comprises of members from Aptech Ltd., professors from reputed Academic Institutions, Senior Managers from Industry, Technical gurus from Software Majors & representatives from regulatory organizations/forums.

Technology heads of Aptech Ltd. meet on a monthly basis to share and evaluate the technology trends. The group interfaces with the representatives of the TAG thrice a year to review and validate the technology and academic directions and endeavors of Aptech Ltd.

Industry Recruitment Profile Survey - The Industry Recruitment Profile Survey was conducted across 1581 companies in August/September 2000, representing the Software, Manufacturing, Process Industry, Insurance, Finance & Service Sectors.

Aptech New Products Design Model



Onlinevarsity
your e-way to learning

WRITE-UPS BY

EXPERTS AND **LEARNERS**

TO PROMOTE NEW AVENUES AND
ENHANCE THE LEARNING EXPERIENCE

FOR FURTHER READING, LOGIN TO

www.onlinevarsity.com

Preface

Web applications have become very popular today due to their efficiency and user-friendliness. They can be used for different types of transactions and online activities. However, use of Web applications comes with an additional responsibility of handling security of data and user information.

This book has been designed to equip you with the knowledge required to implement security while developing Web applications. After reading this book, you will be able to identify security issues in Web applications and perform security measures to deal with the vulnerabilities detected in the Web applications.

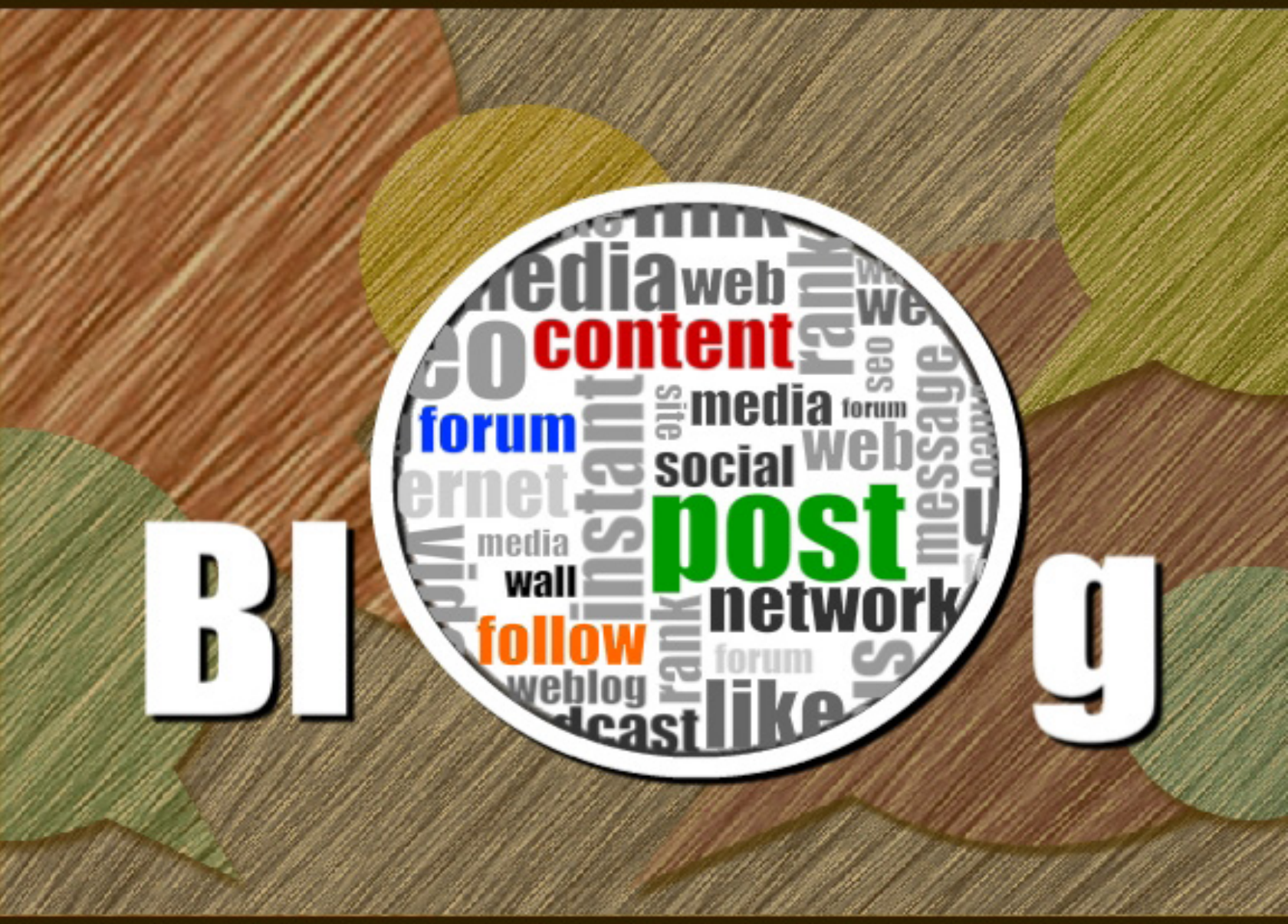
The knowledge and information in this book is the result of the concentrated effort of the Design Team, which is continuously striving to bring to you the latest, the best and the most relevant subject matter in Information Technology. As a part of Aptech's quality drive, this team does intensive research and curriculum enrichment to keep it in line with industry trends and learner requirements.

We will be glad to receive your suggestions. Please send us your feedback, addressed to the Design Centre at Aptech's corporate office.

Design Team

Onlinevarsity

your e-way to learning



Balanced Learner-Oriented Guide

for enriched learning available @

www.onlinevarsity.com

Sessions

ASK to LEARN

Onlinevarsity
your e-way to learning

Questions *in your* mind?

are here to HELP

Post your questions in the ASK to LEARN section for solutions.

Session 1
Introduction to Web Application Security

Welcome to the Session, **Introduction to Web Application Security**.

This session explains about security and impacts of security failure on the Web. This session also describes the need for security and different methods for authentication and session management in a Web application.

In this Session, you will learn to:

- Describe security
- Describe the impacts of security
- Lists and describe Web authentication methods
- Describe session and state management
- Describe Web technologies
- Explain the architecture of a Web application
- List the impacts of security failure on a Web application

1.1 Introduction to Security

Protecting the information assets by the use of processes, training, and technology is called as security in information technology.

Providing security to the transactions performed over the Internet is also included in Internet security. Usually, Internet security provides security to the browser, security to the data which is entered through a Web form, and protection and authentication to the data sent using Internet Protocol (IP).

To protect data which is sent over the Internet, Internet security is relies on certain standards and resources. Different kinds of encryptions are included such as Pretty Good Privacy (PGP).

Firewalls, which block malware, unwanted traffic, virus, and spyware programs are included in the other aspects of secure Web setup that work to monitor network traffic for hazardous attachments from specific devices or networks.

Since a lot of sensitive and confidential transactions are performed online, today the top priority for governments and businesses is Internet security. Financial details and other data handled by an agency's servers or business as well as network hardware are protected by good Internet security policies. The collapsing of an e-commerce business or any other operation where data is routed over the Web is a major threat that may surface due to insufficient Internet security.

1.2 Need of Internet Security

The major cause of intrusions is acquiring access or hacking into any computer network or business information system without access rights. A number of techniques are used by unauthorized persons/ software to take advantage of poorly protected credentials or records to exploit weaknesses of an operating system software or application.

Generally, a malware such as network worm and self-replicating email virus is installed remotely on a compromised system. The amount of data which can be stolen is increased by organized/professional cyber criminals/hackers by reducing the chances of getting detected. The key targets of cyber criminals are the remotely located large accessible stores of online data.

Seemingly harmless information can reveal a lot of data that can compromise a system. For example,

- ➔ Phone number
- ➔ Software and hardware used
- ➔ Types of network connections
- ➔ System configuration and authentication
- ➔ Access procedures

These are the types of information used by the intruders. Information related to security can enable individuals who are unauthorized to have access to important programs and files, thus compromising the security of the system. Passwords, personal information, access control keys and files, and encryption algorithms are examples of crucial information.

Concepts

Figure 1.1 depicts the reasons that lead to requirement of security over the Web.

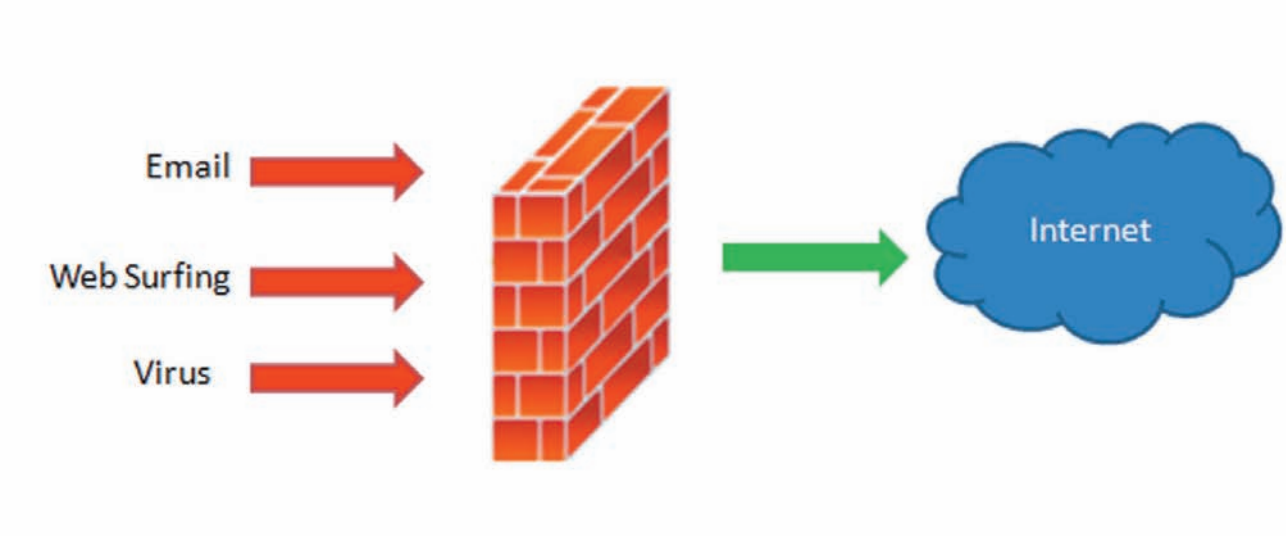


Figure 1.1: Need of Security

There are several situations wherein the system may become vulnerable to malicious attacks such as:

- Multiple workstations in an intranet may become infected when users click the malicious attachments in an email. The incident may affect support operations and even some sensitive data files may get deleted.
- Personal information of millions of people can be hacked by a hacker who acquires access to an unlatched computer.
- People tend to fall for phishing emails which lead to disclosure of user's credentials. The email account may contain private information which can be used for wrong purposes or harm an individual on the financial and/or personal level.

1.3 Impacts of Security Failure

Availability, confidentiality, and integrity are the three basic concepts of security crucial to information on the Internet or Web. Authentication, non-repudiation, and authorization are the concepts which are related to the people who utilize that information.

Loss of confidentiality is defined as the unauthorized reading or copying of information by someone. Confidentiality is considered as a very crucial attribute for certain types of information. Insurance and medical records, research data, specifications of new products, and strategies of corporate investment are some examples.

Corruption of information is done on an insecure network when it is easily available. Modification of information in unexpected ways is known as loss of integrity. This means that unauthorized modifications are done to information by intentional tampering or human error. Integrity is important for financial data and critical safety needs to be applied to activities such as air traffic control, electronic funds transfers, and also financial accounting.

To conduct business, including sales and marketing activities, customer services, producing financial statements, and also management of customer relationship, computer systems are required. The disruption in operations as a result of failure of computer systems due to any reason results in adverse impact on the business profitability.

The confidential information is retained by users on the computers, which includes information of the customer and also proprietary business. Any disclosure of personal information due to compromised security can damage the reputation of business, firms, and so on. It can also expose an organization to litigation and increase regulatory scrutiny. Thus, it may lead to legal, technical, and other expenses.

For example, during a congressional hearing, an account of a company of small wooden furniture which had been jacked successfully was given by a cyber-security official that had resulted in the loss of information containing designs of furniture. The alleged offender(s) was able to commandeer IP of the company so as to expose furniture designs to the market and sell it off at cheaper rates were the statements given by the witnesses.

1.4 Web Authentication and Session Management

The identification of an individual by a username and password is called as Authentication. Authentication is different than authorization in security systems, which gives access to system objects on the identity of the individual. Authentication ensures that the individual is who the user claims to be, but says nothing about the access rights. The access rights are verified by a process called as authorization.

The identity of a user is verified in authentication and the level of access granted to the user on the system/site is also determined using authorization. That is, whether the user is an administrator or a member, or some other privileged user. A number of Web authentication methods are available. The Web authentication selected largely depends on how confidential the information is on the Website and also how much control is needs to be exercises over members who view that information.

1.4.1 Types of Web Authentications

The different types of Web authentication are as follows:

➔ **HTTP Basic Authentication**

The simplest type of Web authentication available is HTTP basic. In this type of authentication, the user is asked to sign in with the use of his/her user name and password. However, the transmission of information is done using Base64 encoding. Here, the information sent is neither encrypted nor secure. Any third party/software can easily intercept the information. Figure 1.2 shows HTTP basic authentication flow.

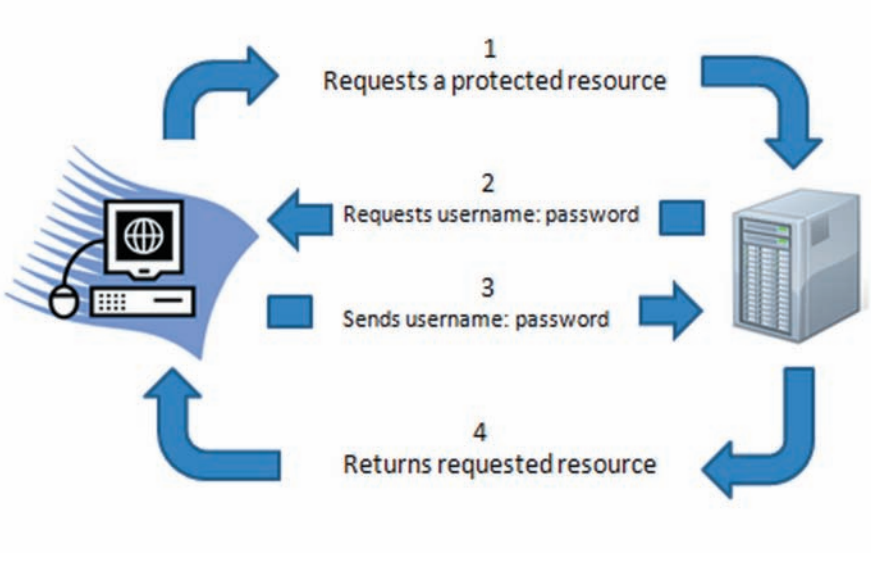


Figure 1.2: HTTP Basic Authentication

➔ **HTTP Digest Authentication**

Digest authentication protocols work in the same way as basic authentication. The identifying information is requested by the server, which is supplied by the user in the form of a user name and password. Then the credentials are compared to those stored in the file by the server and access is granted if the credentials match. It is a simple login scenario.

However, the primary difference in HTTP Digest Authentication is that the data is transferred in a secure manner. Here, the password is 'digested' and then stored in the user database in an encrypted form. Nobody, including the administrator can find out the password by looking at the sequence of encryption. Since only the Web server is capable of reading the password, the integrity of data is better maintained in this authentication mode.

➔ **HTTPS Client Authentication**

Secure Socket Layer (SSL) and HTTP are combined to get HTTPS. Everything here is operated on a closed circuit which is within the SSL, without any outside interference. As a result, the secure socket Public Key Certificate (PKC) can be read by the browser for verifying the legitimacy of every page encountered by it on a Website and also to compare it with the site's security certificate.

Since the information anywhere on the Web that is to be accessed is confidential in ecommerce, HTTPS is used. Thus, this authentication observes high security standards as data is sent through a secure channel between the browser and server including encryption.

➔ **Form Based Authentication**

The information which does not require greater security from visitors uses the form based authentication. This technique uses Web form to collect user information. Form based authentication is used for creating registration pages, surveys, contact forms, and so on.

The data to be validated, such as user name and password, is sent to the Web server. Figure 1.3 depicts Form Based authentication.

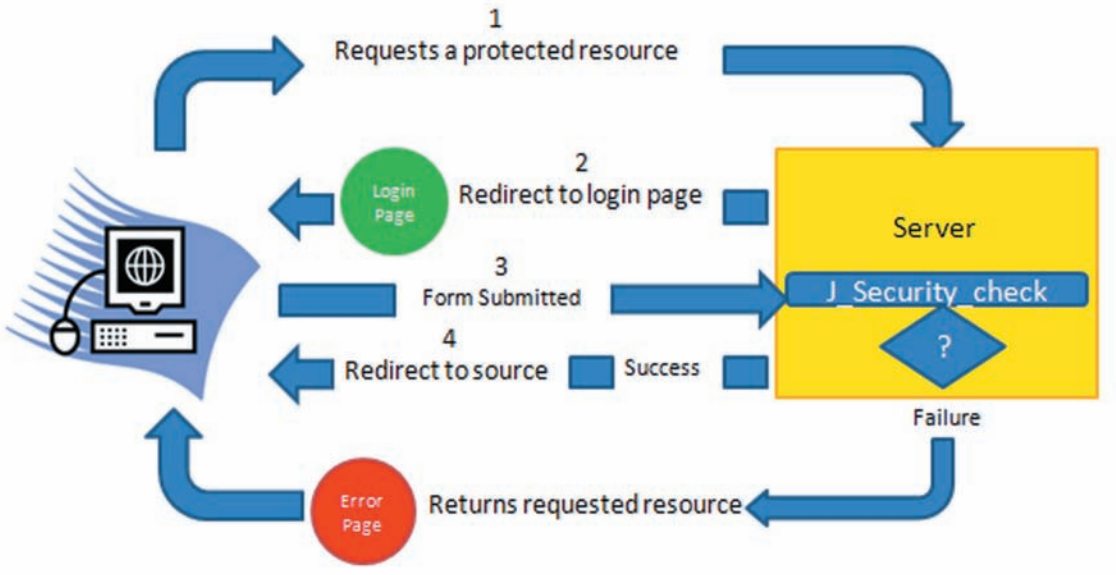


Figure 1.3: Form Based Authentication

1.5 Session Management

Generally, the users are required to provide their credentials at login to get connected to the Web or an application. They are not expected to provide these credentials again unless a privileged action is to be performed or login times out after successful authentication of the user. That is, user authentication is required to be performed only once as session management permits a Web application to keep track of the user and detect whether the user performing the given action is the same user who originally provided the credentials. The hard work done in the authentication process is easily bypassed by the attacker if there is any type of weakness in the layers of session management of the application.

Attacks against session are generally done by taking advantage of weakness of the session management functionality or by obtaining a valid session by exploiting the user's existing session.

The client browser is uniquely identified by an application using session IDs while association level of access and IDs of session is done using the background processes also known as server-side processes. Therefore, once the authentication to the Web is successfully done by the client, with every new page request the client does not require to re-enter the login information as the stored authentication voucher is prepared using the session ID. Session ID information can be allocated and received by the following methods:

- Using URL for session IDs.
- The session ID information is stored in the hidden field and HTTP POST command can be used to submit it.
- Through the use of cookies.

Concepts

1.5.1 URL Based Session IDs

Session ID information can be embedded in the URL, which is received by the application through HTTP GET requests when the client clicks the links.

For example: http://www.example123.com/news.asp?article=37781;sessionid=**IE70012218**

Advantages:

- ➔ It is used even if the security setting of the client Web-browser is high and cookies are disabled.
- ➔ Some other user can also access the resource using the same URL thus, the resource can be shared.
- ➔ Saving the URL as favourite can be used to associate a Session ID permanently with a client browser.
- ➔ URL information is generally sent in the HTTP REFERER field as per the type of Web browser.

Disadvantages:

- ➔ The history of the browser or the stored favorites can be accessed by any person using same computer.
- ➔ The proxy servers and the firewalls may have the URL information as they are intermediary systems. Thus, this information can be accessed by the person having access to these systems and possibly may use the information wrongly.
- ➔ The URL and the associated session ID can be modified trivially by anyone in the standard Web browser. Thus, minimal skills are required to carry out this and these types of attacks are frequent.
- ➔ The HTTP REFERER field can be used to send the session information contained in the URL while navigating to a new Webpage which can be of other Website.

1.5.2 Hidden Fields

In this method, the session ID information is stored in the hidden field and HTTP POST command is used to submit it. For example,

Data can be embedded within the HTML of a page using hidden from fields as follows:

```
<FORM METHOD=POST ACTION='/agi-bin/news.pl'>
<INPUT TYPE='hidden' NAME='allowed' VALUE='true'>
<INPUT TYPE='hidden' NAME='sessionid' VALUE='IE60012219'>
<INPUT TYPE='submit' NAME='Read Article'>
```

Advantages:

- ➔ The attacker requires slightly higher skill level to retrieve or manipulate this session information as compared to URL embedding method.

- Without providing access to the session information, the client is allowed to store and transmit the URL information safely.
- Can be used even if the cookies are disabled and the browser has high security settings applied.

Disadvantages:

- Use of available tools such as Telnet or through personal proxy services, attacks can be carried out.
- The hidden fields embedded on a form make the Web page more complex since it already contains form information, active content such as Flash as well as client-side scripting such as VBScript or JavaScript. Thus, the page tends to become heavier than usual, thereby requiring more time to download on client machine. This makes the site work more slowly and appears unresponsive.
- Also, sometimes due to poor coding practices, the server side program may allow the POST content to be transformed into a URL as it fails to check the submission type and it is then submitted using the HTTP GET method.

1.5.3 Cookies

A cookie is a small, often encrypted file located in the browser directories and used by the Web developers for performing authentication function and also, helps the user to navigate Website(s). Thus, the knowledge of client browser is stored in cookies for a period of time and across many pages. Cookies can also last only for a single session and may also be used to store the expiry information. These cookies are also called as persistence cookies. If the expiration information is not present in a cookie, it is normally stored only in memory. If the browser is closed by the user, these 'session cookies' should be deleted.

For example, within the plain text of the HTTP server response, one can specify a cookie as follows:

```
Set-Cookie: sessionID='IE70012218'; path='/'; domain='www.example.com';
expires='2014-06-01 00:00:00GMT'; version=0
```

Advantages:

- Over a period of time, the session and persistence cookies can be used for regulating access to the Web application.
- The session Id timeouts can be controlled by using the options that are available in the programming language used for developing the page.
- The session information is not recorded on the intermediary devices.
- Most browsers have built-in cookie functionality. Thus, no extra efforts are required to ensure that the session ID information gets embedded in the pages served to the client.

Disadvantages:

- The disabling of cookies in a browser is the most common precaution for security. Therefore, the Web applications using cookies for session management do not work in browsers with cookies disabled.
- The client systems have persistent cookies as text files which can be copied and used on other systems with ease. Depending upon the permissions of the hosts file, the theft of this information can be done by other users of the host and user gets impersonated.
- The storage of complex arrays which contain state information is not possible as the size of cookies is limited.
- SET-COOKIE sends the cookies with each new page or file requested by Web browser within the defined domain.

1.5.4 Session ID

The strength of session ID is a very important aspect for state management in the Web application. The length and randomness are to two main characteristics of an ideal session ID. To protect session ID from being compromised through brute-force or predictive attack, the organization should see to it that a particular set of criteria are fulfilled by the session ID which is used to track the authenticated user.

1.6 Overview of Web Technologies

Nowadays, a wide range of Web technologies are available, from simple to complex, which can be used for creating a Web application. Some of the basic technologies include markup languages, such as HTML, for creating the look and feel of the Web page, programming languages such as C#, Java, PHP, and so on for writing the business logic, and SQL for creating and manipulating the database for storing the data. Several tools and software are available for creating Websites such as Visual Studio, Net Beans, Dreamweaver, and so on.

1.6.1 Markup Languages

Markup languages are used over the Internet in order to describe and confirm as to how the Web pages will be displayed in a browser and/or to define the data which the Web documents contain.

There is a wide range of markup languages. Rich Text Formatting (RFT) used by Word processors as markup language is one such example.

The most commonly used markup languages on the Internet are as follows:

- Hyper Text Markup Language (HTML) is the primary and most commonly used markup language for Web pages. HTML specifies what should be displayed on the page to the browser and how it should be displayed. For example, the text, images, and all other elements are specified by the markup as well as the appearance of text, such as italic or bold, color, position, and so on. Cascading Style Sheet (CSS) is a styling format that can be used to apply fonts, spacing, colors, and so on to the content on Web pages.

CSS allows setting position of elements, hiding some elements, or changing the appearance of the browser such as modifying the scroll bar color in Microsoft Internet Explorer.

- Extensible Markup Language (XML) is a common data interchange format. The structure of an XML document is similar to an HTML file. However, the difference between the two languages is that XML describes the data whereas HTML does the formatting. XML is used for the development of custom markup languages.
- Extensible Style Sheet Transformation (XSLT) is used to apply formatting to XML documents to define its appearance.

1.6.2 Programming Languages and Technologies

Custom applications can be created or functionality can be added to already existing applications using programming languages. By using programming language user can create visual animations, validate forms, respond to user actions, provide e-commerce solutions, and interact with databases on the Internet.

The programming languages are of two types compiled and interpreted. Additional steps are required by the compiled languages which translates them into the machine language code and then stores it in different file with extension either .dll or .exe. Interpreted languages are mostly scripting languages and thus, a browser or server understands it and responds to the code.

In general, more than one programming languages can be used to create the end-to-end solution for Web applications. Technologies are mostly server dependent so first the user needs to determine services to be provided by the hosting Web server before selecting a technology to be used for a Website.

There is no single specific language that is right for every Web project as the needs may differ from a person to person or organization to organization. Every technology and language has its own advantages and disadvantages. Listed here are some examples of most commonly used programming languages:

- PHP is a scripting language which is used on Unix-based servers instead of ASP. It is also an interpreted type of language. PHP is used commonly for providing server-side form, e-commerce processing, and accessing databases. Similar to ASP code, PHP can be embedded within the body of an HTML page.
- Java is an object object-oriented, compiled type of programming language which was designed for use on the Internet. Sun Microsystems designed Java in 1995 and then introduced it to Web developers for including dynamic elements and animations in Web pages. Java is more similar to C++ and is easier to learn.
- C# is an object-oriented, compiled type programming language that leverages on to the Microsoft .Net Framework and is used for creating Web applications for Windows platform. C# is used for server-side processing of ASP.NET Web application and is derived from C and Java.
- JavaScript is a scripting language used for client-side scripting on Web pages to respond to user actions such as button click or hovering of mouse pointer over an image. It is an interpreted language. Dynamic HTML pages can be created using JavaScript combined with CSS and HTML.

1.6.3 Databases and Servers

A collection of data that is well organized is known as a Database. Data is usually organized to model related features of reality in such a way that it supports the processes which require this information.

Specially designed applications used for interacting with the user and other applications and also the database itself for capturing and analyzing the data are known as Database Management Systems (DBMSs). An all-purpose DBMS is created for creation, definition, updating, querying, and administration of databases. The more popular DBMSs include Microsoft SQL Server, MySQL, and SQLite. A database cannot be generally ported across different DBMS, however different DBMSs can operate together by using models such as SQL and JDBC or ODBC for allowing a single application to work effortlessly with many databases.

A system responding to requests across a computer network for providing or helping in providing a network service is known as a server. In many cases, a computer can provide several services and have several servers running. They can run on a dedicated computer, mostly also known as 'The server', however many computers networked together can also host servers. The Web and enterprise applications are deployed on a server to be accessed globally.

1.7 Web Application Architecture

Generally, a Web application consists of a client as the end user and a Web application deployed on a server placed on a remote location that serves as a host. It is a simple setup which can be termed as 2-tier or Client-Server architecture. Figure 1.4 depicts Client-Server architecture.

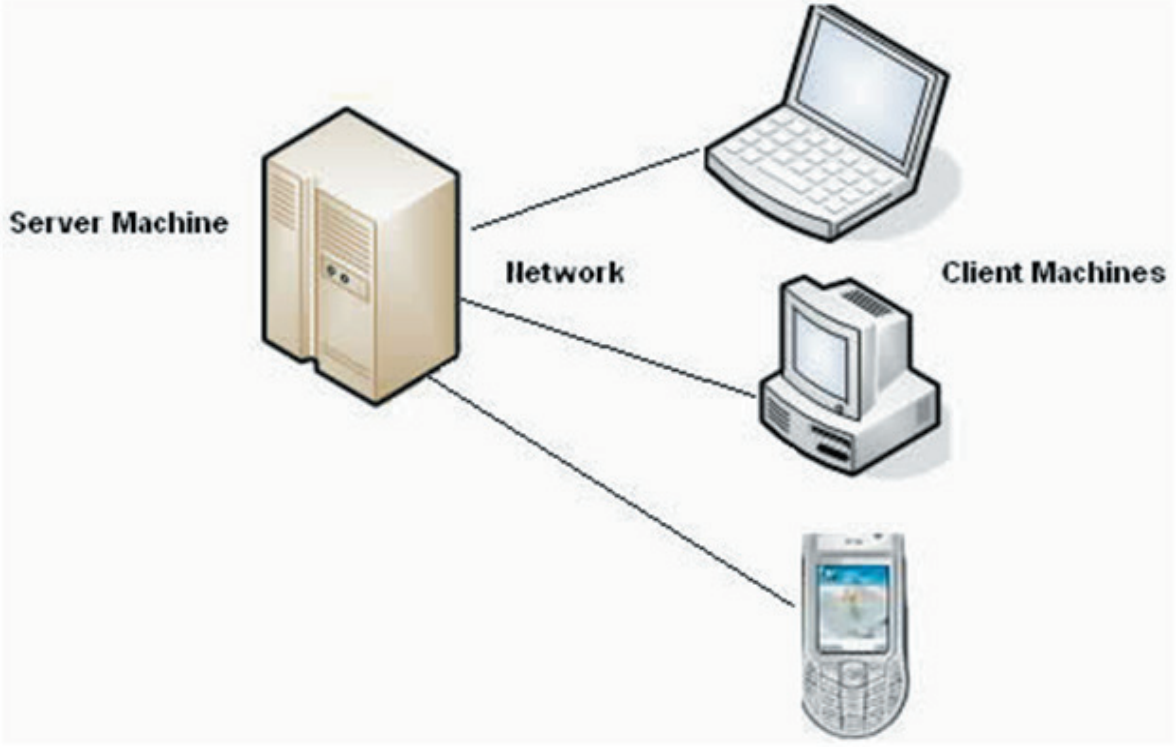


Figure 1.4: Client-Server Architecture

Concepts

With increasing complexity of a Web application, the components can be split across multiple tiers for reducing the management overhead.

A 3-tier architecture is an architectural style for deployment which describes the functionality into separate layers, each segment called as tier that can be located on the same or different machine. Component-oriented approach was basic for evolution of 3-tier architecture using platform specific communication methods instead of message-based approach.

Different applications can use this architecture differently depending on the requirements and situations. It can also be used in distributed applications. Figure 1.5 depicts 3-tier architecture.

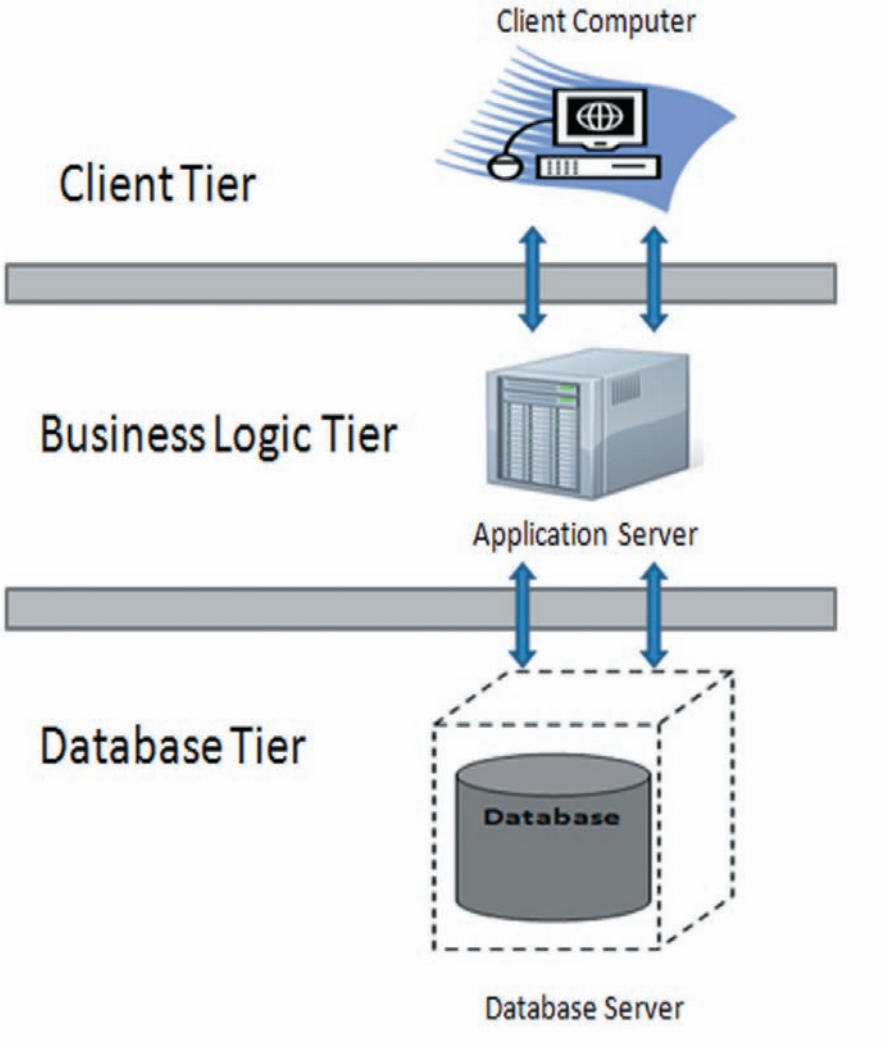


Figure 1.5: 3-Tier Architecture

3-tier architecture is composed of a Presentation tier, a Business Logic tier, and an Enterprise Information (EIS) Tier also called the Database tier.

1.7.1 Presentation Tier

It is the application's topmost layer. The User Interface (UI) for the application is provided by the presentation layer. For smart client interaction, it uses Graphical User Interface and for browser- based interaction it uses Web Based technologies. The information related to services such as browsing merchandise, shopping cart contents, purchasing, and so on is displayed in the presentation tier. Communication with other tiers takes place by displaying the result in the form of output to the browser/client tier as well as the other tiers within the network.

The Presentation tier includes the Web browser, client-side application downloaded components such as .Net assemblies or Java Applets. Through simple HTML the client tier interacts with the Web server over HTTP or the client can act as Web service entity in case of rich client and use SOAP over HTTP interactions with the Web server. Further, the client can use security token such smart cards. These tokens can be used for authenticating users and also to protect request.

1.7.2 Business Logic Tier

From the presentation tier the logic tier is pulled out and just like its own layer an application's functionality is controlled by performing detailed processing. Solving of mission-critical business problems is done at logic tier. The components of this layer can be present on a server machine to help assisting in resource sharing. These mechanisms can be used to apply business rules, like legal or governmental regulations and business algorithms, and data rules, that are designed to maintain the consistency of data structures within either specific databases or multiple databases. These middle-tier components can be utilized by all applications and also can be moved to various different locations as they are not bound to a specific client. For example, to minimize the network round-trips the simple edits may be placed at the client side, or in the stored procedures the data rules can be placed.

1.7.3 Database Tier (Enterprise Information System (EIS) Tier)

This tier is the real DBMS access layer, it consists of database servers. It can be accessed via the business services layer and occasionally by the user services layer. Information is stored as well as retrieved here.

Data is kept neutral as well as independent from business logic or application servers in this tier. Scalability and performance improves by providing data its own tier. Rather than consisting of raw DBMS connections, this layer consists data access components which helps in resource sharing and also allowing configuration of clients without the installation of DBMS libraries and also ODBC drivers on each client separately. Example of this is a computer system hosting database management system (DBMS), like Microsoft SQL Server database.

A Web application with different technologies used in the different tiers is shown in figure 1.6.

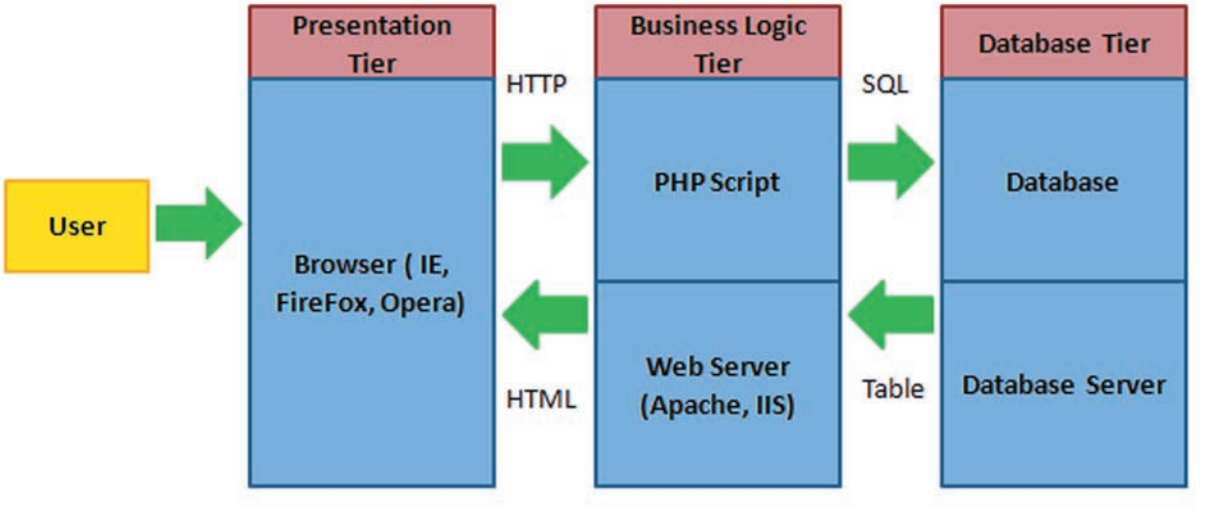


Figure 1.6: Example of 3-tier Architecture

In the given example, the presentation layer consists of a browser such as Internet Explorer, Opera, or Firefox. The browser sends HTTP request to the Web server such as IIS or Apache server which returns an HTML page to the client as a response. Consider a Website wherein login page is sent back to the client. The credentials are entered by the user on the browser and the data is sent back to the Web server. The Web server validates the data by communicating with the database on the Data tier to verify if the credentials match any of the records in the database. The record, if exists, is sent back to the Web server. Based on the response from the Database server, the next page is shown on the client browser that may contain some messages if the credentials do not match and home page of the Website if the credentials match with a record in the database.

1.8 Impacts of Security Failure on the Web Application

A breach in security can affect the Web application and the organization at large in several ways as follows:

- **Damage to customer confidence**: When it comes to Web application, customer may be the first one to notice the result of attack. Customer strongly believes that security breach can affect them economically as well as personally. This may undermine the customer's confidence. For example, if a security failure leads to a wrong transaction which involves a huge sum of money, it will affect the severely affect the customer's trust in the Web application and in case the same problem occurs again in the future, the user will simply stop using the application.

- **Loss of revenue**: The security failure may also lead to loss of revenue. For example, if security of an online shopping site is affected by some malicious program, the customers will stop visiting the site and the owner of the site will suffer a severe setback in his/her business due to loss of revenue.

- **Damage to the reputation**: Security failure also strongly damage the reputation of an organization due to which the business may get affected.

- **Legal Consequences**: Security breach may result in legal consequences. An organization may be put to risk due to security breach that may lead to disclosure of sensitive information related to the business or of the clients.

- **Interruption of business processes**: There can be interruption in the business process due to failure in security. For Example, if the central server of a bank is attacked by some threat, the whole banking process gets interrupted and the transactions may not be performed accurately.

It can also lead to other far reaching effects. Immediate action must be taken against a security weakness so as to remove it and minimize the resulting damage.

1.9 Check Your Progress

1. The processed information which is returned to the client is formatted by the _____.

(A)	Web tier	(C)	Presentation tier
(B)	Database tier	(D)	Business tier

2. Match the following authentication modes with the corresponding description.

	Authentication Mode		Description
a.	HTTP Basic Authentication	1.	Everything here is operated on a closed circuit which is within the SSL, without any outside interference
b.	HTTP Digest Authentication	2.	The transmission of information is done using Base64 encoding
c.	HTTPS Client Authentication	3.	Used when information does not require greater security from visitors
d.	Form Based Authentication	4.	The password is 'digested' and then stored in the user database in an encrypted form

(A)	a-2, b-4, c-1, d-3	(C)	a-3, b-4, c-1, d-2
(B)	a-4, b-3, c-2, d-1	(D)	a-2, b-3, c-4, d-1

3. Which of the following statements about 3-tier architecture are true?

a.	The business logic is actually implemented on the database server.
b.	Processed information is usually formatted using the mark up languages such as XML in case of rich client or HTML in case of client browser.
c.	To request the database systems, mostly Structured Query Language (SQL) query language is used.
d.	The business tier can only communicate with the database tier and not with the presentation tier.

(A)	a, b	(C)	b, c
(B)	b, d	(D)	a, d

4. JavaScript is a _____ language used for creating Web pages which responds to user actions such as button click or hovering of mouse pointer over an image.

(A)	Compiled	(C)	markup
(B)	Scripting	(D)	object-oriented

5. Modification of information in unexpected ways is known as _____.

(A)	loss of integrity	(C)	loss of confidential information
(B)	loss of availability	(D)	loss of data

6. Which of the following modes can be used for session management in a Web application?

(A)	URL based Session Id	(C)	Session ID in the hidden fields of forms
(B)	Through the use of cookies	(D)	All of these

1.9.1 Answers

1.	B
2.	A
3.	C
4.	C
5.	A
6.	D

Summary

- Protecting the information assets by the use of processes, training, and technology is called as security in information technology.
- The major cause of intrusions is acquiring access or hacking into any computer network or business information system without access rights.
- Availability, confidentiality, and integrity are the three basic concepts of security crucial to information on the Internet.
- The client browser is uniquely identified by an application using session IDs while association level of access and IDs of session is done using the background processes also known as server-side processes.
- Markup languages are used over the Internet in order to describe and confirm as to how the Web pages will be displayed in a browser and/or to define the data which the Web documents contain. Custom applications can be created or functionality can be added to already existing applications using programming languages.
- 3-tier architecture is composed of a Presentation tier, a Business Logic tier, and an Enterprise Information Tier (EIS).
- The business tier includes the application server and the business logic is implemented on the application server.

ASK to LEARN

Onlinevarsity
your e-way to learning

Questions *in your* mind?

are here to HELP

Post your questions in the ASK to LEARN section for solutions.

Session 2

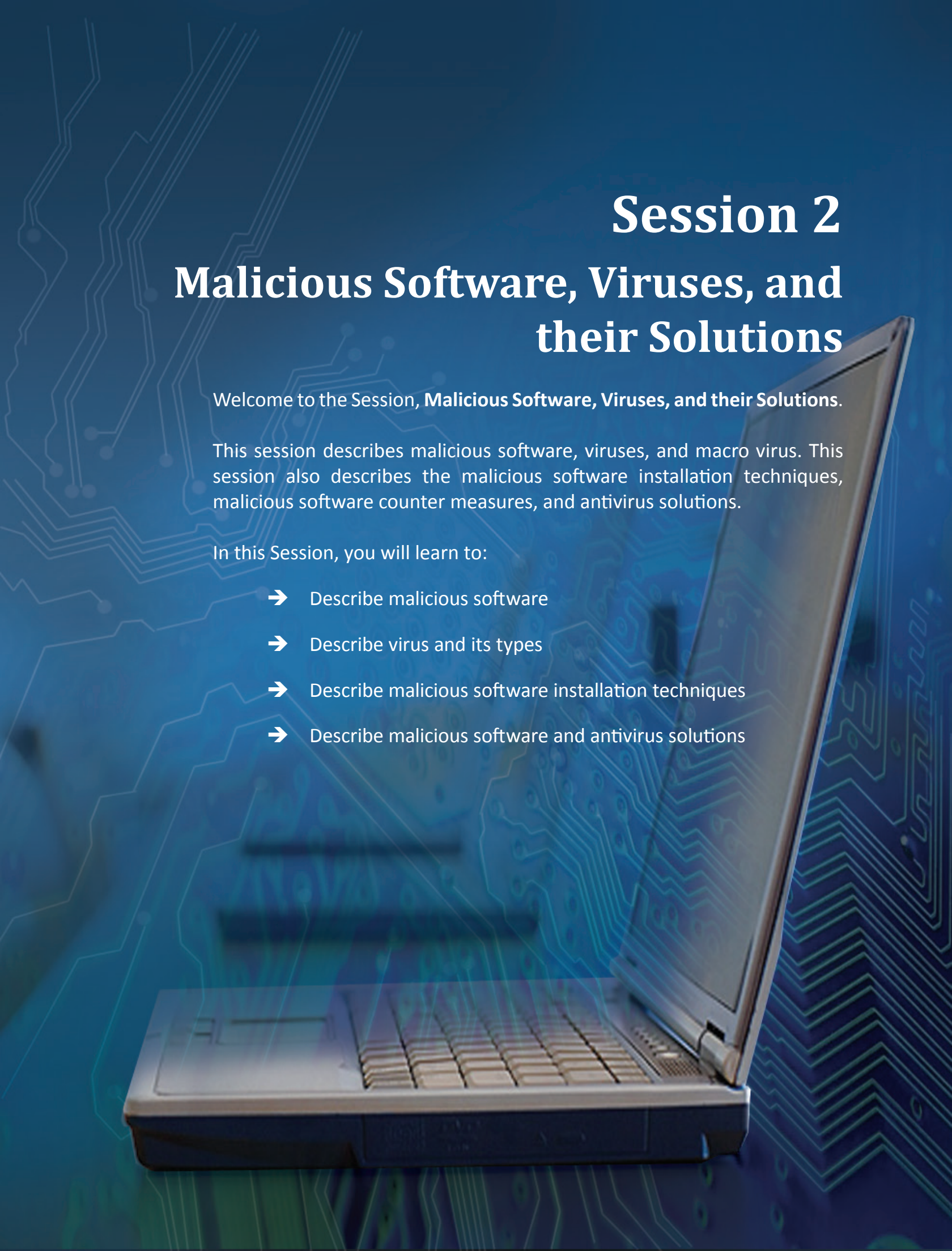
Malicious Software, Viruses, and their Solutions

Welcome to the Session, **Malicious Software, Viruses, and their Solutions**.

This session describes malicious software, viruses, and macro virus. This session also describes the malicious software installation techniques, malicious software counter measures, and antivirus solutions.

In this Session, you will learn to:

- Describe malicious software
- Describe virus and its types
- Describe malicious software installation techniques
- Describe malicious software and antivirus solutions



2.1 Malicious Software

Malicious software is software used to gather sensitive information, disrupt computer operation, or gain access to unauthorized computer systems. It can be in the form of active content, code, scripts, and other software. Malicious software is also referred to as malware.

The different types of malicious software such as adware, browser hijacking software, spyware, viruses, and also fake security software are included in the term malware.

The security of the computer and privacy is critically affected as soon as a malware is installed in the computer. For example, personal information may get relayed to any third party machines and these advertisers are known as malware. The programs which contain viruses and worms also lead to damage of data and/or functioning of the computer.

The different types of malware are as follows:

- **Virus**: The most destructive and commonly-known form of malware is virus. Deletion of data, hijacking the device for attacking the other system, sending of spam messages, or hosting and sharing illegal content is done by viruses.
- **Spyware**: The collection and distribution of personal information to the interested unauthorized third party without the knowledge of the user is done by Spyware. The installation of Trojan viruses is also done by spyware.
- **Adware**: Adware is short for Advertising-supported software. The displaying of unwanted advertisements when the user is online is done by Adware. Its purpose is to render advertisements automatically to generate revenue for its owner.
- **Fake security software**: A legitimate software that tricks user into providing personal information, opening your system to further infection, or even damaging 'clean ups 'is called Fake security software.
- **Browser hijacking software**: The changing in browser settings, displaying of pop-up ads, and creation of new desktop shortcuts is done by browser hijacking software. It is also able to relay the personal information of user to unauthorized third party.
- **Worm**: A worm is known as a self-replicating virus as it performs duplication of files rather than alteration of files. The worms get noticed only when there is an uncontrolled replication which consumes a lot of system resources, thereby halting or slowing down other tasks.
- **Logic bomb**: A programming code inserted intentionally and designed to execute under situations such as failure of a program to respond or lapse of certain amount of time is called a logic bomb. A logic bomb, when 'exploded ' or executed may be designed to print or display a fake message, corrupt or delete data, or have other unwanted or harmful effects.
- **Trapdoor**: Trapdoor is used for gaining access as part of the system by some procedures other than the usual procedure. Trapdoors may be inserted by the hackers who have penetrated in the system to allow them to again gain access at a later date. Sometimes, even the system developers leave a debug trapdoor in software which is then used by hackers to exploit the system.

- **Trojan**: Trojan (Trojan Horse) is a malicious program with harmful code contained inside a harmless appearing program or data. This program can carry out chosen damage such as deleting or modifying data on hard disk by gaining control of certain area on the system.
- **Rootkits**: Rootkits is a set of tools used for gaining and maintaining access to a system by the intruders without user's knowledge. It can modify the running processes. It intrudes in several operating systems such as Solaris, Linux, and some versions of Windows. A computer affected by a rootkit is known as rooted computer. There are three types of rootkits as follows:
  - **Kernel Rootkits** - It modifies the code and replaces the portion of existing kernel code of the system and hides on the computer system. It can be very dangerous because it is difficult to detect without appropriate software.
  - **Library Rootkits** - The manipulation of system calls is done using hooks, patches, or replacements to hide the information of the intruder.
  - **Application Rootkits** - Replacement or modification of regular application binaries with hooks, injected code, camouflaged fakes, or patches.

2.2 Viruses

A virus is a malware but it cannot execute independently. It attaches itself to a program and executes when the host program is running. While executing in such a way, virus can perform any function such as deleting files or programs for which the current user has permission.

The three features of computer viruses are as follows:

- **Infection mechanism**: The mechanism of spreading of virus in such a way that it is enabled to replicate itself is termed as infection mechanism or infection vector.
- **Trigger**: The activation or delivery of payload is determined by this condition or event.
- **Payload**: Besides spreading, activities such as damage to files and code in the computer are done by the virus.

Figure 2.1 depicts the phases in the life cycle of a virus.

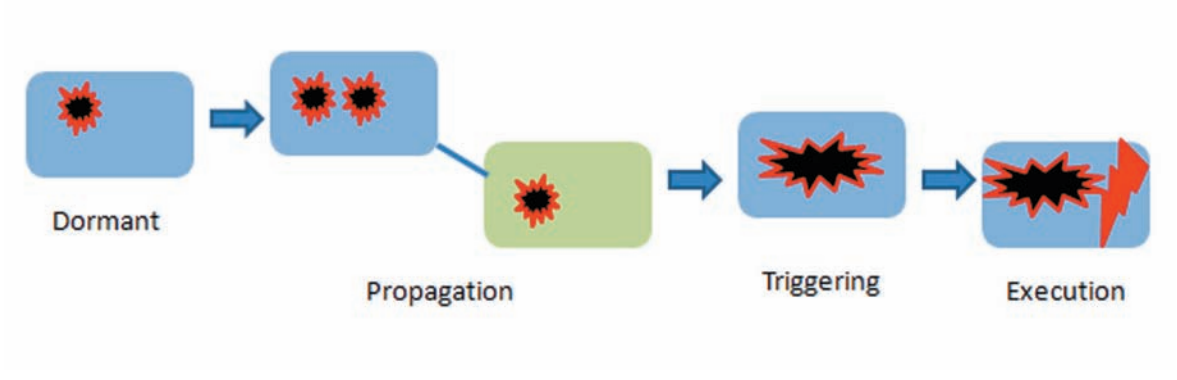


Figure 2.1: Phases of Life Cycle of a Virus

Concepts

A typical virus has following four phases in its life cycle:

- **Dormant phase**: This virus is idle in this phase. The activation of the virus happens eventually due to some events such as change of system date, the presence of other file or programs, or the exceeding limit of disk's capacity. This stage is not present in all the viruses.
- **Propagation phase**: The virus gets copied into other programs or some system areas on disk gets infected by the virus. The propagating version may not be similar to actual virus and thus, detection is often morphed by it. The clone of virus is now contained by each and every infected program, which on itself enters the phase of propagation.
- **Triggering phase**: The virus is activated to perform the function for which it was intended. As with the dormant phase, the triggering phase can be caused by a variety of system events, including a count of the number of times that this copy of the virus has made copies of itself.
- **Execution phase**: The function is done. The function can be harmless such as a message on screen, or a damaging one, such as the destruction of data and files of a program or the system.

2.2.1 Types of Viruses

In virus classification by the target are as follows:

- **Boot sector infector**: The master boot record or boot record is infected and when the system containing the virus is booted, the virus gets spread.
- **File infector**: The files which are considered as executable by the shell or operating system are infected by this type of virus.

Following are the types of viruses according to it spreading technique:

- **Encrypted virus**: A random encryption key is created by a portion of virus and the remaining virus is encrypted by it. The virus is then decrypted by a random key when an infected program is executed. A different random key is selected when the virus replicates itself. As different virus has different encrypted keys, no constant bit pattern can be found.
- **Stealth virus**: It is a form of virus which is designed explicitly to hide itself from being detected by antivirus software. Thus, the whole virus is hidden rather than just the payload.
- **Polymorphic virus**: It is a virus which mutates with each infection, makes detection by 'signature' of virus impossible.
- **Metamorphic virus**: The mutation is done with every infection in the same manner as in the case of polymorphic virus. The only difference between them is that a metamorphic virus is rewritten by itself completely with each iteration; causing difficulty in detection. The appearance as well as the behavior may be changed by metamorphic viruses.

Common symptoms of a malware affected system are as follows:

➔ **Browser crashes and instabilities**

- Browser is closed unexpectedly or becomes unresponsive.
- The home page is changed to another Website and cannot be reset.
- The browser gets a new toolbar added to it.
- Clicking a link does not give any response or the user is redirected to a Website which is unrelated.

➔ **Poor system performance**

- Internet connection stops unexpectedly.
- Computers do not give response and take longer time to get started.
- Applications fail to open.
- Applications get blocked from downloading updates and are not able to open (mostly security programs).
- Addition of new icons on the desktop or installation of suspicious programs.
- Specific system settings and options of configuration become unavailable.

➔ **Advertising**

- Popping up of ads even if the browser is closed.
- Browser gets launched automatically for displaying ads.
- New pages opened automatically for displaying ads.
- Only ads are displayed in search results.

2.3 Malicious Software Installation Techniques

The infection-installation life cycle is comparatively easy to understand. For example, a computer gets infected through infection vector such as removable media. A virus would then be placed within the computer, start-up settings and registry would be modified, and harmful things may happen with the launch of virus. After sometime, the virus would be detected by the antivirus vendors, detection signatures and clean-up script would be created by them, and the updated product would be distributed. Thus, the end result is reassuring but it becomes an endless game.

Later, new methods are unveiled by the virus authors by adding additional steps to the installation lifecycle for facilitating more scalable and robust malicious goal.

The important steps incorporated are dropper and downloader strategies described as follows:

- **Dropper**: A distributable software package which has multiple malware components in it is termed as Dropper. A dropper installs its payload of malicious agents automatically, disables the monitoring software and victim's security, obfuscates its activities, and seeks to hide core components. Once completed with these tasks, the core malware agent would be started by it.

- **Downloader**: Downloader would perform the same activities as Dropper, that is, disable the monitoring software and victim's security, obfuscate its activities and seek to hide core components, and so on. However, Downloader is smaller than Dropper because the core malicious library components are not contained in it. The Downloader connects to a remote file repository and downloads the core components as it is without the core malicious library components.

The incorporation of droppers and downloaders into the installation lifecycle has increased the number cyber crimes as it facilitates a federated solution ecosystem and offers new potential for evasion. Figure 2.2 depicts the life cycle of a Dropper.

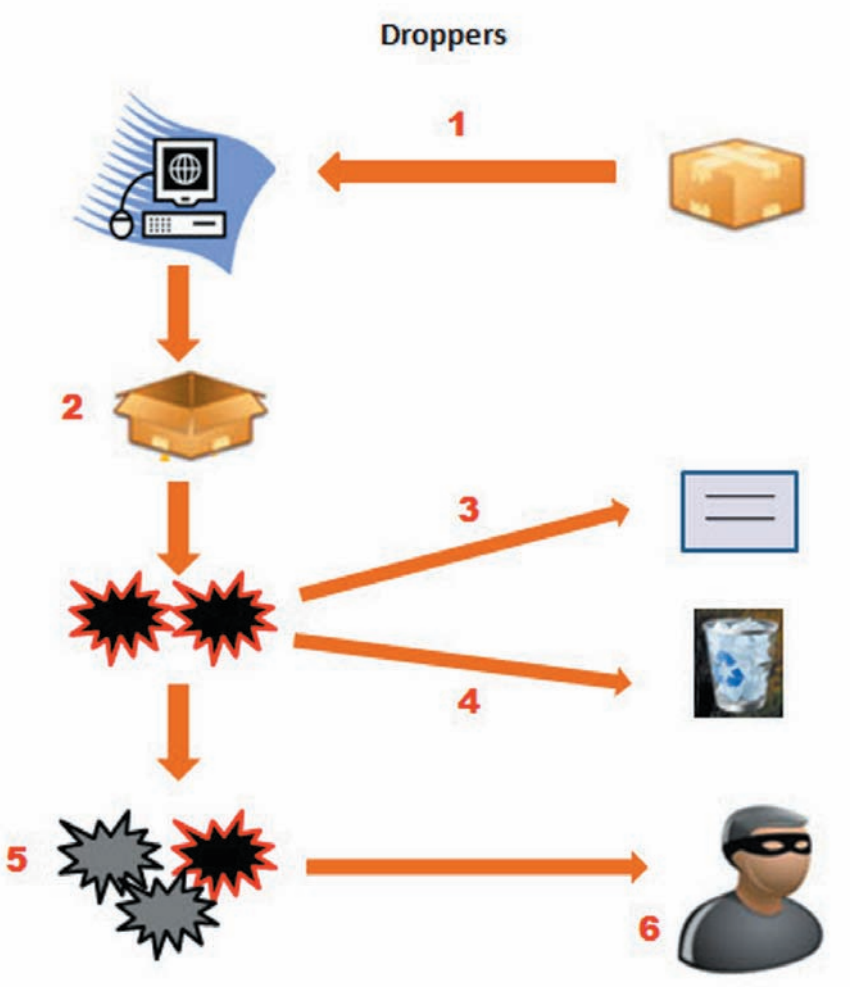


Figure 2.2: Dropper Life Cycle

The activities performed by a Dropper during its life span are as follows:

1. After succumbing to social engineering or falling victim to an exploit, the dropper binary which is self-contained is downloaded by the computer.
2. After the dropper is executed, it unpacks and installs the components that it contains by executing embedded commands.
3. Components of the dropper try to modify configuration settings and disable security settings, and after the system restarts, it also ensures execution of the main malware components automatically.
4. Any proof of system modifications and the infection vector is obfuscated or deleted.
5. The core malware contained in the dropper is ready for use while the original package which was having the non-essential components extracted from the dropper is deleted from the computer.
6. The core malware component then waits for new commands by contacting the attacker's sites.

Figure 2.3 depicts the life cycle of a Downloader.

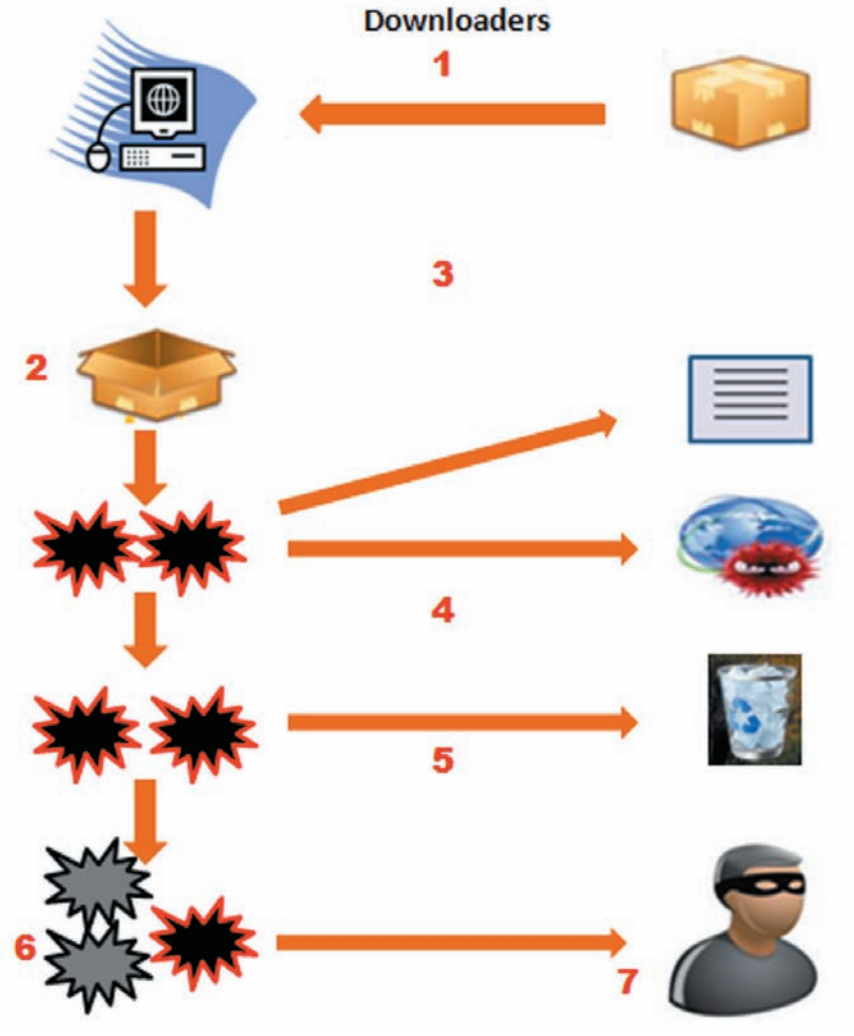


Figure 2.3: Downloader Life Cycle

The activities performed by a Downloader during its life span are as follows:

1. After succumbing to social engineering or falling victim to an exploit the downloader binary is downloaded by the computer.
2. After the downloader is executed it unpacks and installs the temporary components by executing embedded commands.
3. Components of the downloader try to modify configuration settings and disable security settings, and after the system restarts, it also ensures execution of the main malware components automatically.
4. By connecting to a remote file repository, the downloader downloads the core malware component.
5. Any proof of system modifications and the infection vector is obfuscated or deleted.
6. The core malware contained in the downloader is ready for use while the original package which was having the non-essential components extracted from the downloader is deleted from the computer.
7. The core malware component then waits for new commands by contacting the attacker's sites.

2.4 Solutions for Malicious Software

The elimination of malware and prevention from malware is done effectively by good quality anti-malware software. If the infected system is used only by a single user, then all the user needs is anti-malware software for vigorous and dependable Internet security. The anti-malware software contains all the primary and important features of security which are found in Internet security suites except the elements such as password managers and parental control.

Note - Parental control is a feature which can be included in computers, digital television services, mobile devices, and software. It is mostly used to keep track of activities of children on the system such as the sites visited by them, time they spend on the PC, and so on.

The ideal solution for non-intrusive safeguards of Internet is anti-malware software. Now-a-days the Internet is fully littered with malware which is programmed to look for weaknesses in the end user's system. The computer is defended from malware tricksters such as Trojans, rootkits, viruses, spyware, pop-ups, scripts, key loggers, and adware by the best malware protection software.

Kaspersky Antivirus is known for leading online and offline protection from malware. F-Secure Antivirus is considered as a reliable shield for scanning of low-resource malware. Several other anti-malware software are available and can be used to secure the system from malicious attacks.

2.4.1 Anti-malware Software Features

The two main functions namely detect malware and remove it should be effectively performed by an application to be counted among the best anti-malware software. If these two functions are not performed by the software, then its other functions provide very low or no value.

The user usually looks for anti-malware software which has a proven ability of recognizing and eradicating malware and also does not need much resource. The technical and the management support provided by anti-malware software developers is also an important feature in selecting anti-malware software.

Some important features of the anti-malware software are as follows:

- **Malware Detection**: The malware which cannot be detected cannot be removed by the anti-malware software. Hence, it is an important factor for selecting anti-malware software. It should not only detect the known threats, but also the newly emerging threats and the scripts which behave suspiciously.
- **Malware Removal**: Removal of malware which has infected the system is a challenging task rather than the detection of it. The anti-malware which is capable of removing the malware safely from an infected system and restoring it to its original state that is pre-infected state is mostly rated as best by a user.
- **Malware Management**: A malware removal tool should be light to run and can be customized for handling malware in the way a user wants. Only the different types of scan are not important, the user should also be able to look for the resource that the scan utilizes and can adjust it accordingly. Some features the user can consider to maximize the PC performance during scans include full screen modes, the ability to create white lists, and low-use detection.
- **Help and Support**: The type of technical support and customer assistance provided by the anti-malware manufacturers is an important aspect for selecting the anti-malware software. Some anti-malware software providers also offer email chat and telephone support, online documentation, and user manuals.

2.4.2 Malicious Software Counter Measures

Malicious software attacks are increasing day by day due to which Anti-Spyware, Antivirus, and such products have to play a catch-up game continuously. Introduction of new and increasingly complex malicious software continues. Sometimes it can widely spread and damage the system before the protection software companies deliver the solution to detect and protect from it.

Hence, there is need to develop a malicious software strategy that consists of procedures for malicious software detection, control and recovery of damaged files, and also should clearly outline the objectives for it. Risk is involved in introducing security measures, so this should be done under the advice of qualified network management dealer. Following are some counter measures for malicious software:

- **Security Awareness**: Awareness of user is the most important counter measure for controlling virus infection and spread. Only trusted sources should be used for accepting any type of data. Users should be warned while visiting 'hostile' Websites and for opening suspicious email.
- **Patch Management**: The process of updating PCs or servers with the latest service packs and security patches is called as Patch Management. Developers of spyware, viruses, and other malicious software exploit the existing flaws in software already existing in the PC for spreading and damaging the system. Patches are issued by software companies to fix the flaws which have been discovered.

To detect the patches available, use automatic updates for vulnerabilities which is important for maintaining proper system functionalities. Sometimes, installing an update or patch may interfere with the currently running processes.

- **Antivirus Scanners**: These products are used for scanning instant messages, emails, and files to detect malicious software by matching the signature patterns. These products may work effectively only if they are regularly updated for latest virus signatures as new viruses are continuously evolving. For activating these products, a user just needs to follow the steps given in the manual of the product. These scanners can be provided on network hosts and/or on the gateways to a network. The important criteria for selection of an antivirus scanner are regularity and method of update as it needs to be frequently updated for being effective.

- **Audit Information**: An abnormal activity may be detected using audit logs such as firewall logs. For example, malicious software trying to read or write to unauthorised area and Trojans trying to send data from some site may be detected using audit logs.

- **System Hardening**: The damage caused by the malicious software can be minimized by setting the policy of a running application as least privileged and carefully implementing the system access control.

- **Active Content Blocking**: It protects the network from malicious content and also blocks the unwanted Internet traffic. It also helps to make sure that the business resources are being utilized for business purpose only. The components that a user should look for Internet blocking or filtering system are:
  - Centralized Administration
  - Automatic Updates
  - Reporting Capabilities
  - Category Based Products

- **Firewalls**: Some of the remote programs can be blocked if they depend on a particular port to execute with the help of firewalls. A firewall can be server based or PC based.

2.5 Antivirus Solutions

Antivirus software can be defined as a category of programs which detects, remedies, and prevents malware infections on computing devices and systems.

The name antivirus was first given to the program which removed a type of malware known as virus. However, today the antivirus program are used for preventing and detecting infections of almost all types such as adware, Trojan horses, spyware, worms, and so on.

Antivirus software detects malware by using two methods namely, signature-based detection and heuristics-based detection.

2.5.1 Signature-based Detection

Malicious software can be detected by knowing and then observing the patterns. These patterns are defined in data definition files or signature files. If the antivirus software is signature based, then it will scan the signature files for determining if a pattern found is malicious or not. Then, the antivirus, depending on the configuration settings will quarantine or delete the malware which is being detected due to signature match. The viruses or malware which are quarantined cannot damage the computer while they are quarantined. These quarantined viruses can be used by information security professionals for further analysis. The signature files must be kept up to date to protect the system from recent malicious threats. The Automatic update setting will ensure the download and installation of latest signature files when available.

2.5.2 Heuristic-based Detection

The heuristic-based detection method is used in the advanced antivirus software in addition with the signature-based detection. The major difference of this method with signature-based detection is that it is capable of detecting the malware which are previously unknown. Therefore, for this particular malware, signature does not exist. By performing dynamic scanning of the potential files, heuristic -based scanner looks for a sequence and sequences of set of instructions which differentiate the malicious software from normal program. Additionally, some of the heuristic-based scanners may decompile the malware by reading the source code of malware and by reverse engineering the infected program.

Some important features or functions commonly found in antivirus software are as follows:

- **Real-time Scanner**: Network data that enters into the computer is scanned by real-time scanner of the antivirus program for detecting malware as it enters the system.
- **On-access Scanner**: Whenever files are opened or accessed by the user, the on-access scanner scans the file for detecting virus or other malware.
- **On-Demand Scanner**: It provides ability to perform a custom scan of drives, folders, and files.
- **Heuristic Scanner**: Heuristic scanner is also present in the antivirus software. It uses existing information of the malware and the past experience to detect new threats even before they are found by the vendors of antivirus software for creating updates to detect it.
- **Compressed File Scanner**: A compressed file such as zip file can be a source of malware. These compressed files can also be scanned by almost all antivirus programs. An efficient program can scan many levels deep for detecting malware when it is buried within multiple compressed files.
- **Script Blocking**: Mostly the malicious code is executed by using scripting languages used in making the Websites. Many antivirus programs can also monitor Visual Basic, Java, ActiveX, and other script files to detect and block activity if found malicious.
- **Scheduled Scans**: Creating a schedule is possible in most of the antivirus software, for performing an automatic scan. Some antivirus programs are more flexible programs which may allow scheduling any type of scan or even customize a scan, while some may restrict based on type of scans which can be scheduled.

- **POP3 Email Scanning**: The antivirus software can also monitor incoming and outgoing POP3 email traffic and file attachments associated with them for detecting and alerting about virus or other threats.
- **Webmail Protection**: The Web-based email traffic such as Hotmail or Yahoo! Mail can be monitored by antivirus programs to detect and block virus and other malware which are sent as file attachments in the mail.
- **Instant Messaging Protection**: Many types of malware can now be spread through instant messaging programs such as Yahoo! Messenger or AOL Instant Messenger (AIM). To detect and block malicious threats, instant messaging traffic can be monitored by some antivirus software.
- **Automatic Virus Updates**: The major issue faced by users having antivirus software is to keep it up to date. Configuration of many antivirus software programs is possible for automatically getting connected to the vendor site and regularly downloads the new updates.
- **Automatic Program Updates**: To add functionality for detecting new threats, the scan program and engine(s) itself may be updated periodically. Configuration of many antivirus software programs is possible for automatically checking for new updates. Also, if updates are available the software can instantly download and install them.

2.6 Check Your Progress

1. A programming code inserted intentionally and designed to execute under situations such as the failure of a program to respond or the lapse of a certain amount of time is called _____.

(A)	Logic bomb	(C)	Trapdoors
(B)	Worms	(D)	Virus

2. Match the following authentication modes with the corresponding description.

	Authentication Mode		Description
a.	Encrypted virus	1.	It is a form of virus which is designed explicitly to hide itself from being detected by antivirus software.
b.	Stealth virus	2.	It is a virus which mutates with each infection, makes detection by 'signature' of virus impossible.
c.	Polymorphic virus	3.	This virus is rewritten by itself completely with each iteration, causing difficulty in detection.
d.	Metamorphic virus	4.	A random encryption key is created by a portion of this virus and the remaining virus is encrypted by it.

(A)	a-2, b-4, c-1, d-3	(C)	a-3, b-4, c-1, d-2
(B)	a-4, b-1, c-2, d-3	(D)	a-2, b-3, c-4, d-1

3. Which of the following statements about heuristic-based detection are true?

a.	The heuristic-based detection method is used in the advanced antivirus software in addition with the signature-based detection.
b.	The heuristic-based detection is capable of detecting the malware which are previously unknown is the major difference between heuristic.

(A)	Statement b	(C)	Statements a and b
(B)	Statement a	(D)	None of these

4. The process of updating PCs or servers with the latest service packs and security patches is called _____.

(A)	Patch Management	(C)	System hardening
(B)	Active content blocking	(D)	Firewalls

5. A distributable software package which has multiple malware components in it is termed as _____.

(A)	Downloader	(C)	Trojan
(B)	Dropper	(D)	None of these

6. Which of the following are features of antivirus software?

(A)	Real-time scanner	(C)	Scheduled scan
(B)	Webmail Protection	(D)	All of these

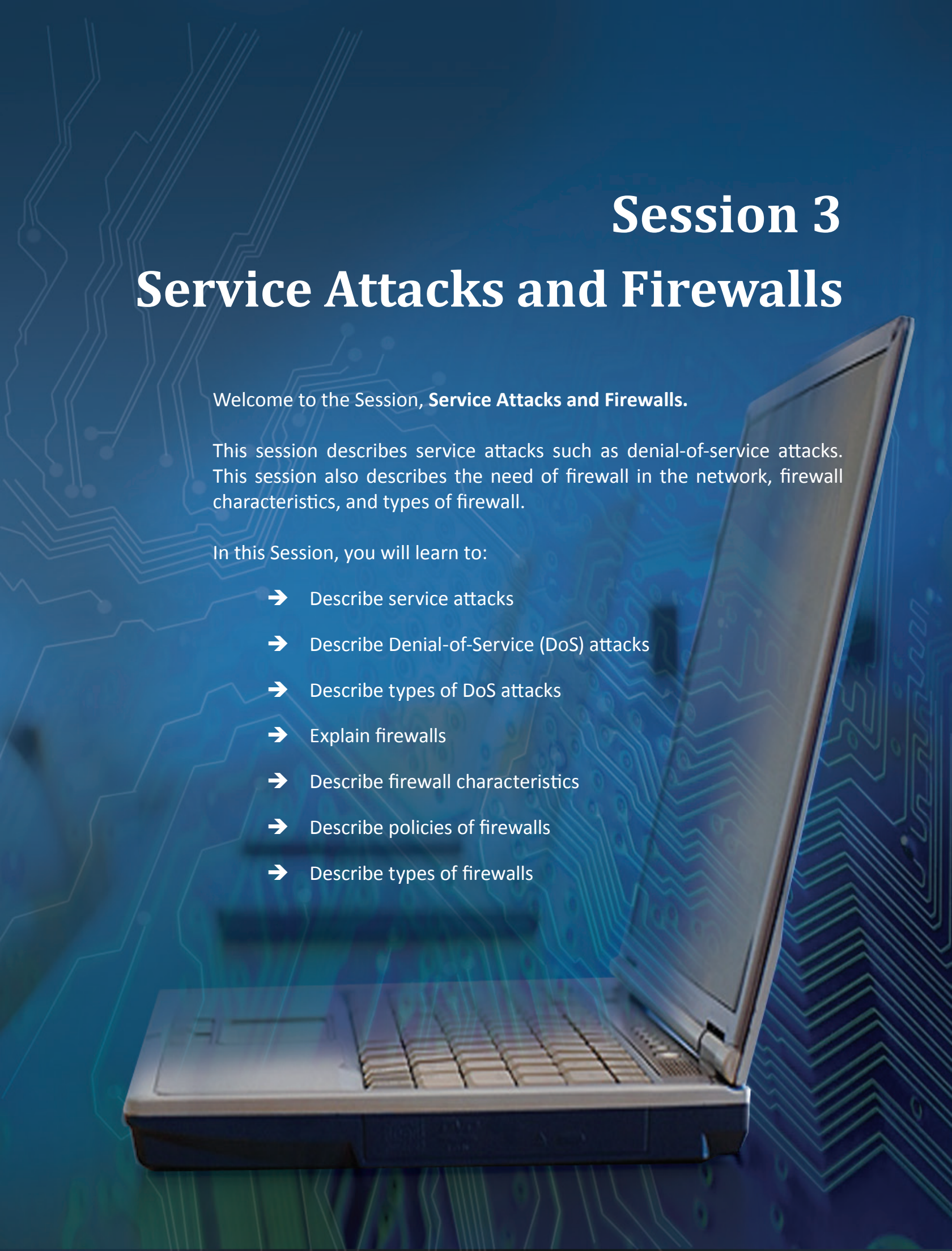
2.6.1 Answers

1.	A
2.	B
3.	C
4.	A
5.	B
6.	D

Summary

- Malicious software is software used to gather sensitive information, disrupt computer operation, or gain access to unauthorized computer systems.
- The additional installation steps incorporated in malicious software installation are dropper and downloader.
- The elimination of malware and prevention from malware is done effectively by a good anti-malware software.
- The process of updating PCs or servers with the latest service packs and security patches is termed as Patch Management.
- Antivirus software can be defined as a category of programs which detects, remedies, and prevents malware infections on computing devices and systems.
- Malicious software can be detected by knowing and then observing the patterns in a signature-based scanner.
- Heuristic-based scanner is capable of detecting the malware which are previously unknown.

Session 3
Service Attacks and Firewalls



Welcome to the Session, **Service Attacks and Firewalls.**

This session describes service attacks such as denial-of-service attacks. This session also describes the need of firewall in the network, firewall characteristics, and types of firewall.

In this Session, you will learn to:

- Describe service attacks
- Describe Denial-of-Service (DoS) attacks
- Describe types of DoS attacks
- Explain firewalls
- Describe firewall characteristics
- Describe policies of firewalls
- Describe types of firewalls

3.1 Service Attacks

The attacks done on a service or via a service can be termed as service attacks. The service attacks may lead to infection in the system or also can lead to denial of service to the legitimate users by blocking the services.

The two main types of service attacks are:

- Denial-of-Service Attacks
- Distributed Denial-of-Service (DDoS) Attacks

3.1.1 Denial-of-Service (DoS) Attack

A Denial-of-Service (DoS) attack is an attack where legitimate users are not permitted to access the service as all the resources are utilized by the attacker. In this type of attack, the attacker usually sends requests to the server for authentication or for establishing connection with invalid return addresses. The return address is not found by the server or network for sending the authentication approval, which causes server to wait before closing the connection. When the server closes the connection, more number of messages with invalid return addresses are sent to server for authentication by the attacker. Thus, after the authentication process completes, the server again waits and begins the same task again and thereby, it is kept busy by the attacker. This leads to denial-of-service to valid users trying to access the server.

3.1.2 Types of DoS Attacks

The different types of denial-of-service attacks are as follows:

- **Ping of Death**

  Generally, every message which needs a response from the server can be used for denial-of-service attack. The goal of the attacker is to force the server to spend more time on the requests and try to respond to the request till it crashes.

  A relatively harmless Internet Control Message Protocol (ICMP) echo request or the ping message can be a powerful tool for an attack. Thus, ping flooding technique is called the Ping of Death where attacker sends very large ping packets which make the system vulnerable and it finally crashes.

- **SYN Flood**

  In a TCP SYN flooding attack, all the system resources are utilized and no new TCP connection can be made. The attacker uses three-way handshaking protocol along with invalid IP addresses for establishing connection.

Note - In the process, to establish connection between TCP client and server three steps are involved. They are:

1. A SYN message is sent by client.
2. Server sends a message which contains ACK (acknowledgement) for the SYN message of client and its own SYN message.
3. Then, ACK message is sent by the client for server's SYN message.

Therefore, this process of establishing connection is called as three-way handshake.

The request is sent by the attacker using a SYN message with a false address to begin connection establishment with the server. The server is unaware of the invalid address and responds back by sending SYN-ACK message, and waits for the ACK message which never arrives.

Till the timeout time, the connection remains open partially and in the wait state. Thus, the system resources are unnecessarily kept busy till the connection closes. In this manner, the attacker floods the server by SYN packets or connection requests, all with invalid addresses.

Each request is treated equally by the server and it waits till time out for the ACK before closing the connection. Since the false request comes before the timeout, the system resources are not freed early. Thus, all the system resources get busy leaving no space for new connections.

➔ **UDP Flood**

There is no flow control mechanism, datagram sequence number, and connection state tracking in a UDP connection. There is no tab maintained on the expected packets in the UDP protocol. Therefore, it is very easy to flood UDP ports so that they become busy responding to packets with false addresses and no resource or bandwidth is left for network connection with the server.

➔ **ICMP Redirect Floods**

The ICMP redirect message type is used for shorter router acknowledgement to update the routing tables. The acknowledged redirect messages sent by server can be tampered and used for sending wrong data. This could make the server send the traffic to the computer desired by the attacker as if it is forwarding the traffic to another remote host.

➔ **Reflected Attack**

False packets may be sent to many computers. When the false packets are received by the computers, they reply back to the spoofed address as the packet was spoofed. Thus, instead of responding to real sender, the machines will try to communicate with the machine address in the false packet. Thus, the attack is so strong that the server shuts down.

3.1.3 Distributed DoS Attack

When an attack occurs from a single computer or from a network of computers, it is termed as Denial-of-Service Attack. When the attack is designed and planned from a distributed network system, it is termed as Distributed Denial-of-Service Attack.

While it is easy to shut down a small Website server by any denial-of-service attacks, it is very difficult to crash the server of large Websites by using only denial-of-service attack. This is because, multiple servers are used by large Websites for hosting their system and also more secure and complicated protocols are used by them.

As shown in figure 3.1, to attack large Websites, the attackers use a distributed setup in which they use several machines to target a single Website. These machines could be already infected ones or part of the attacker's network. These malicious programs having a target to bring down a particular Website are called bots. Thousands or hundreds of system flooding requests sent to the Website server bring the server down as all the resources are utilized and the server cannot respond to a new request.

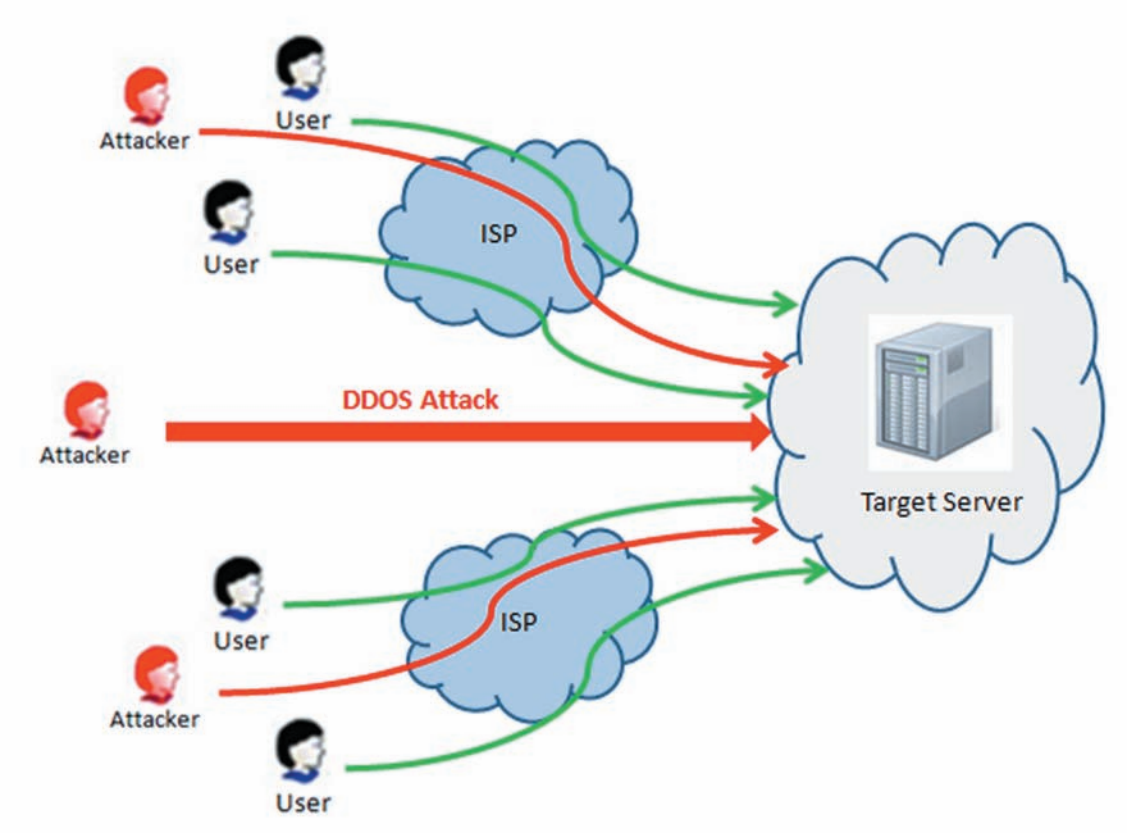


Figure 3.1: Distributed Denial-of-Service Attack

Concepts

3.1.4 Types of Distributed DoS Attacks

The different types of distributed DoS attacks are as follows:

- **Peer-to-Peer Attacks**: In this type of attack, the legitimate visitor is redirected to the site or the server, the attacker wants to attack by making use of peer-to-peer network. If attacker is able to pull in many users in this way, the Website shuts down due to DDoS.

- **Degradation of Service Attacks**: This type of attack is done to overload the server until it becomes extremely slow to the point of no use. Thus, this Website will not be used by anyone as it continues to give a slow response for a long time and more visitors will try to find other Websites with similar functionality.

  The main issue here is that, these types of attacks are difficult to detect because it will be hard to tell if it is a boost of real traffic or the DDoS attack. An important thing that can be done by the Website owner in such situation is to analyze the visitor's actions and benchmark that with historical data. Thus, it would help to know if it is an attack or not.

- **Application Level Attacks**: These are also called as Layer 7 DDoS attacks. In this attack, the weakest point of the Website will be targeted. These attacks are very difficult to stop. These attacks can be stopped with only with appropriate software, infrastructure, and knowledge about the attacks.

- **Multi-Vector Attacks**: It is the most complex form of DDoS. In this attack, attackers not only make use of various attacking strategies, but also often use several tools for attacking. While performing this type of attack, the attacker pinpoints the application on the server and at the same time, floods the site with unwanted traffic.

3.2 Counter Measures for Service Attacks

It is very difficult or almost impossible to stop the denial-of-service attacks. So the best thing is to prevent it, by taking all necessary steps to make it harder for the attacker to perform such attacks.

It is impossible to guess correctly as to who may be attacking the Website. Potentially, the attacker could be anyone doing it for fun or intentionally to bring the server down.

So instead of focusing on the attacker, the following precautions can be taken:

- **Setting Up a Filter**

  Before a request reaches Web server, a filter can check the information in the request. Thus, by analyzing the information, it can determine if the request is a fake or genuine message, and can prevent the attacks.

- **Installing Firewalls**

  Make sure firewall is installed so that all the messages pass through the firewall. The requests can be allowed or discarded according to the policies configured in the firewall settings.

➔ **Server Configuration**

A server configuration may prevent situations which results in all the resources getting consumed. The following instructions may help to restrict the usage of server resources:

a. Limit the services which can be executed by the server at a time. This could include number of same services or different services.

b. Close all the network ports which are unused.

c. Based on the IP addresses restrict the access.

d. For emails and error messages, create a separate partition which may prevent the whole system from being blocked.

➔ **Disable UDP Ports**

Since the TCP protocol is more secure as compared to UDP protocol, try to use TCP instead of UDP, as far as possible. All the inherently insecure UDP ports and UDP services should be disabled.

➔ **Deny Ping Requests**

Denial-of-service attacks make use of ping command. Hence, consider restricting or limiting the acceptance of ping from the remote machines. To cut down the risk of attacks, it is better to reject external ping requests.

➔ **System Updates**

Updates of the services and products are constantly researched and then, updates are released by the software companies to add features, and make them more secure for better management and security. The software used by the user should be updated as per the official release of the product by the vendor.

➔ **Antivirus**

Users of a computer system must ensure the security of their system by checking if their system is prone to virus attacks, so that unknowingly they do not become a participant in the distributed denial-of-service attack. User must install and update the antivirus software regularly.

3.3 Introduction to Firewalls

A collection of integrated security measures designed for preventing unauthorized access to a computer system in a network is termed as firewall. It is same as firewalls of buildings, as it is also intended to isolate one network from other.

As depicted in figure 3.2, firewalls can protect a network of systems or local systems from security threats of network effectively, while at the same time allow access to outside world through Internet and Wide Area Network.

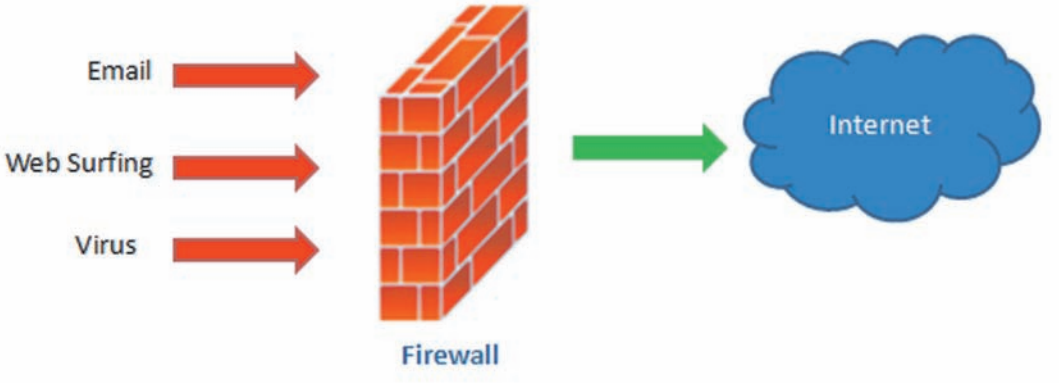


Figure 3.2: Firewall

Figure 3.2 shows a firewall placed between the system and the Internet, and protects it from the Internet threats such as intruders, viruses, and so on.

3.3.1 Need for Firewalls

An organization now is no longer without Internet connectivity. The services and information available that are necessary to the organization are available on the Internet which can be accessed by using LAN or dial-up connection. Though Internet access benefits an organization in many ways, it also allows the outside world to interact with the local network assets. This may prove as threat to the organization. While it is possible to equip each server and workstation with strong security features such as protection against intrusion on the network, it may sometimes not be sufficient or cost-effective. Therefore, a firewall is a widely accepted alternative or at least complementary to the services of host-based security.

To erect a perimeter or an outer security wall and to establish a control link, the firewall is positioned between the Internet and the premises network. The goal of the security wall is to protect the internal network from Internet-based threats and imposing auditing and security at a single choke point. The firewall may be a set of systems which will cooperate with each other to perform the function of firewall or just one system. Thus, firewalls provide an additional layer of defense and insulate the internal systems from the external world.

3.3.2 Firewall Characteristics

The different firewall characteristics are as follows:

- All traffic from outside to inside, and vice versa, must pass through the firewall. This can be achieved by physically blocking all access to the local network except through the firewall.
- Only authorized traffic defined in the local security policy can pass through firewall. Different types of firewalls have implementation of different types of security policies.
- Penetration is not possible in the firewall itself.

3.3.3 Security Policies of Firewalls

The rules laid out by an organization are guidelines for implementing the firewall security policy technically. The rules are created based on the security policy of organization or user, while implementing the firewall. The primary function of the firewall is to allow/disallow the different types of connections and traffic according to the security policy.

In the case of a packet filter firewall, which packet is to be forwarded and discarded through firewall is decided by the firewall policy. If the firewall is of gateway or application proxy type, then which type of service is to be allowed to be accessed through firewall is decided by the security policy.

The following are the two policies to be defined in the firewall settings:

➔ **Allow By Default**

In this policy, all the services and packets are allowed through firewalls and the packet or service which is not required is denied explicitly. It is insecure by nature as anything will be passed through firewall. It is mostly used for research and development purposes.

➔ **Deny By Default**

In this type of policy, passing through firewall will be denied to all the services and packets. Anything which is to be allowed to pass is done explicitly. This policy is secure by nature as all the forthcoming unknown threats are already denied by default. This policy is widely used. The only threat that remains unchecked by using this policy is the threat of packets or services which are being allowed to pass through firewall.

The security requirement of an organization is used for making the firewall policy.

Following are some common rules to be implemented in the firewalls:

➔ Telnet access is not allowed through firewall as telnet is insecure by nature and data is passed in the plain text format.

➔ The FTP connection to and from the network should not be allowed by default, except in special cases such as uploading error logs to the vendor ftp sites.

➔ Unsecure email access should be prohibited by default.

➔ Direct connection between outside service and internal client should not be allowed. If no other alternative is available, then use proxy server instead.

3.4 Types of Firewalls

The different types of firewalls are as follows:

- **Packet Filtering Firewall**: A firewall can act as a packet filter by allowing packets which meets the criteria to pass and reject the packet based on criteria mentioned in the policy of firewall. Thus, a firewall which works as a packet filter applies a set of rules to the outgoing and incoming packets and accordingly forwards or discards the packet. The firewall is configured for filtering the packets both to and from the internal network. These packet filters are easy to implement and are transparent to the end users compared to other types of firewalls. However, they are difficult to configure, particularly when large number of rules for handling a wide variety of application users and traffic are to be generated.

- **Stateful Inspection Firewalls**: A directory of outbound TCP connections is created by stateful inspection packet firewall for tightening up the rules for TCP traffic. The packet information is reviewed by it in the same manner as packet filtering firewall. However, it also records TCP connection information. Some of these firewalls also keep records of the TCP sequence numbers for preventing the attacks which are based on sequence number such as session hijacking. Some also look at the application data in limited amounts for some protocols such as FTP for identifying and tracking related connections.

- **Application-Level Gateway**: An application-level gateway is also known as application proxy and acts such as a relay for application-level traffic. By using a TCP/IP application such as FTP or Telnet, the user contacts the gateway and then, gateway asks for the name of the remote host from the user which is to be accessed. If valid user ID and authentication information is provided, the gateway relays TCP segments having application data between the two endpoints by contacting the application which is on the remote host. The service cannot be forwarded and supported across the firewall if the proxy code for a specific application is not implemented on the gateway.

- **Circuit-Level Gateway**: The circuit-level gateway is for standalone systems. In comparison with application gateways, circuit-level gateway does not allow end-to-end TCP connection, instead sets two TCP connections, one between a TCP user on an outside host and itself and one between a TCP user on an inner host and itself. The gateway relays TCP segments from one connection to the other, once the two connections are established without examining the contents. Connections will be allowed based on the security settings.

3.5 Role of Firewalls in Web Applications

Standard firewalls are designed for restricting access to certain services or ports which an administrator desires to protect from unauthorized access.

The form of firewall that controls input, output, access, and service is known as an application firewall. It operates by potentially blocking the output, input, or system service call according to the firewall configured policy and monitors them. All the network traffic of any OSI layer is controlled by the application firewall.

There are two main types of application firewalls, host-based application firewalls and network-based application firewalls.

Web Application Firewalls (WAFs) are also called as 'Deep Packet Inspection Firewalls' as they inspect every response and request within HTTPS/HTTP/XML-RPC/SOAP/Web Service layer. Some of the firewalls look for specific 'attack signatures' for identifying the attack which is sent by an intruder, while others look for abnormal behavior which does not fit in the normal traffic patterns. These firewalls can be either hardware or software appliance-based and can be positioned in front of the Web server for shielding it from the incoming attacks. Often the WAFs are with negative signature detection and HTTPS/HTTP protocol enforcement.

Other protection techniques include URL scanning and normalization, positive security functionality which enforces proper page logic flow and application operation, and adaptive learning sections which can update policies of security on the fly. The attacks masked by HTTPS encryption are also blocked by WAFs by using the detection policy violations, Web server's private key, and also by resetting the offending connections. These sessions can be either actively terminated and re-encrypted or passively decrypted and inspected.

WAFs can configure policy according to the usage of specific Web functions and elements such as form fields and application session logic. The session awareness is the major differentiator possessed by Web application firewall. They can block attacks specific to a wide range of databases, Web servers, and programming platforms.

WAFs can also be configured to rewrite and mask outbound and inbound server responses which may help to protect against leakage of sensitive information such as credit card numbers. Thus, this security can be used by Payment Card Industry Data Security Standard (PCI DSS) policy. They can be deployed between Web servers and perimeter defenses to protect them or can be installed on Web server platforms directly such as host-based WAFs. Figure 3.3 shows a simplified enterprise network and position of WAF in it.

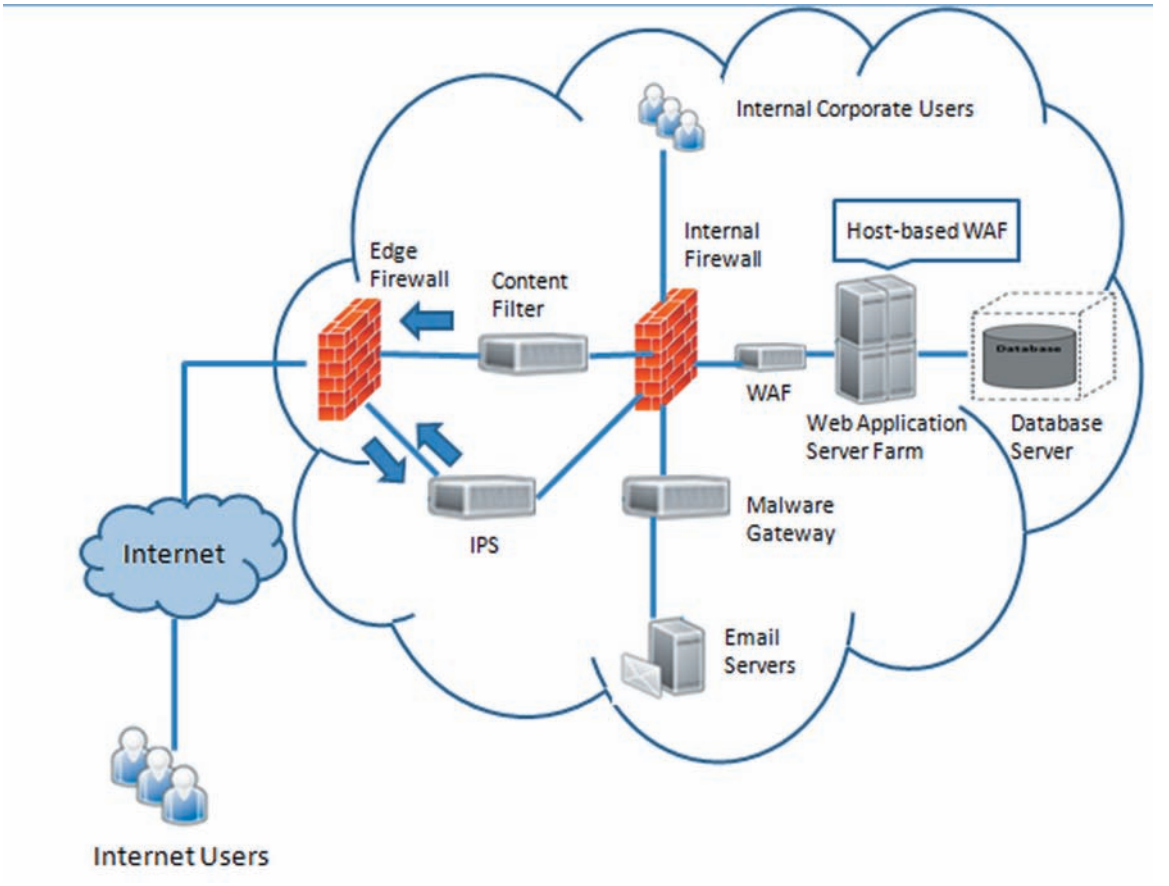


Figure 3.3: Simplified Enterprise Network with WAF

The figure indicates that the WAF protects the Web application and the related server from the external traffic. Also, the firewall used in the internal network of an organization is displayed.

Concepts

3.6 Check Your Progress

1. A _____ attack is an attack where legitimate users are not permitted to access the service as all the resources are utilized by the attacker.

(A)	Denial-of-Service (DoS)	(C)	Ping of Death
(B)	SYN flood	(D)	UDP flood

2. Match the following DoS attacks with the corresponding description.

	DoS Attacks		Description
a.	SYN flood	1.	The acknowledged redirect messages sent by server can be tampered and used for sending wrong data.
b.	Ping of Death	2.	Attacker sends very large ping packets which make system vulnerable and the server crashes.
c.	UDP flood	3.	The request is sent by the attacker by sending a SYN message with a false address to start the connection establishment with the server.
d.	ICMP Redirect Floods	4.	It is very simple to flood UDP ports, so that they become busy in responding to the packets with false addresses.

(A)	a-2, b-4, c-1, d-3	(C)	a-3, b-4, c-1, d-2
(B)	a-4, b-3, c-2, d-1	(D)	a-2, b-3, c-4, d-1

3. While configuring server to prevent the DoS and DDoS, what points are to be considered?

a.	Limit the services which can be executed by the server at a time.
b.	Make sure firewall is installed, so that all the messages are passed through firewall.
c.	Close all the network ports which are unused.
d.	As the TCP protocol is more secure as compared to UDP protocol, try to use TCP instead of UDP if possible.

(A)	a, b	(C)	a, c
(B)	b, d	(D)	a, d

4. Which of the following options are types of firewalls?

(A)	Circuit-Level Gateway	(C)	Stateful Inspection
(B)	Application-Level Gateway	(D)	All of these

5. In _____ attack, attackers not only make use of various attacking strategies, but also often use several tools for attacking.

(A)	Degradation of Service	(C)	Multi-Vector
(B)	Application Level	(D)	Peer-to-Peer

6. Web Application Firewalls (WAFs) are also called as _____ because they inspect every response and request within HTTPS/HTTP/XML-RPC/SOAP/Web Service layer.

(A)	Defense in Depth Firewall	(C)	In-depth Inspection Firewall
(B)	Deep Packet Inspection Firewall	(D)	None of these

3.6.1 Answers

1.	A
2.	D
3.	C
4.	D
5.	C
6.	B

Summary

- A Denial-of-Service (DoS) attack is an attack where legitimate users are not permitted to access the service as all the resources are utilized by the attacker.
- The types of Denial-of-Service (DoS) attacks are: Ping to Death, SYN flood, UDP flood, ICMP Redirect Floods, and Reflected Attack.
- When the attack is designed and planned from a distributed network system, it is called as Distributed Denial-of-Service (DDoS) attack.
- The types of Distributed Denial-of-Service (DDoS) attacks are: Peer-to-Peer Attacks, Degradation of Service Attacks, Application Level Attacks, and Multi-Vector Attacks.
- A collection of integrated security measure designed for preventing unauthorized access to a computer in a network is called as firewall.
- In case of a packet filter firewall, which packet is to be forwarded and discarded through firewall is decided by the firewall policy.
- The form of firewall that controls input, output, access, and service is known as an application firewall.
- Web Application Firewalls (WAFs) are also called as 'Deep Packet Inspection Firewalls' as they inspect every response and request within HTTPS/HTTP/XML-RPC/SOAP/Web Service layer.

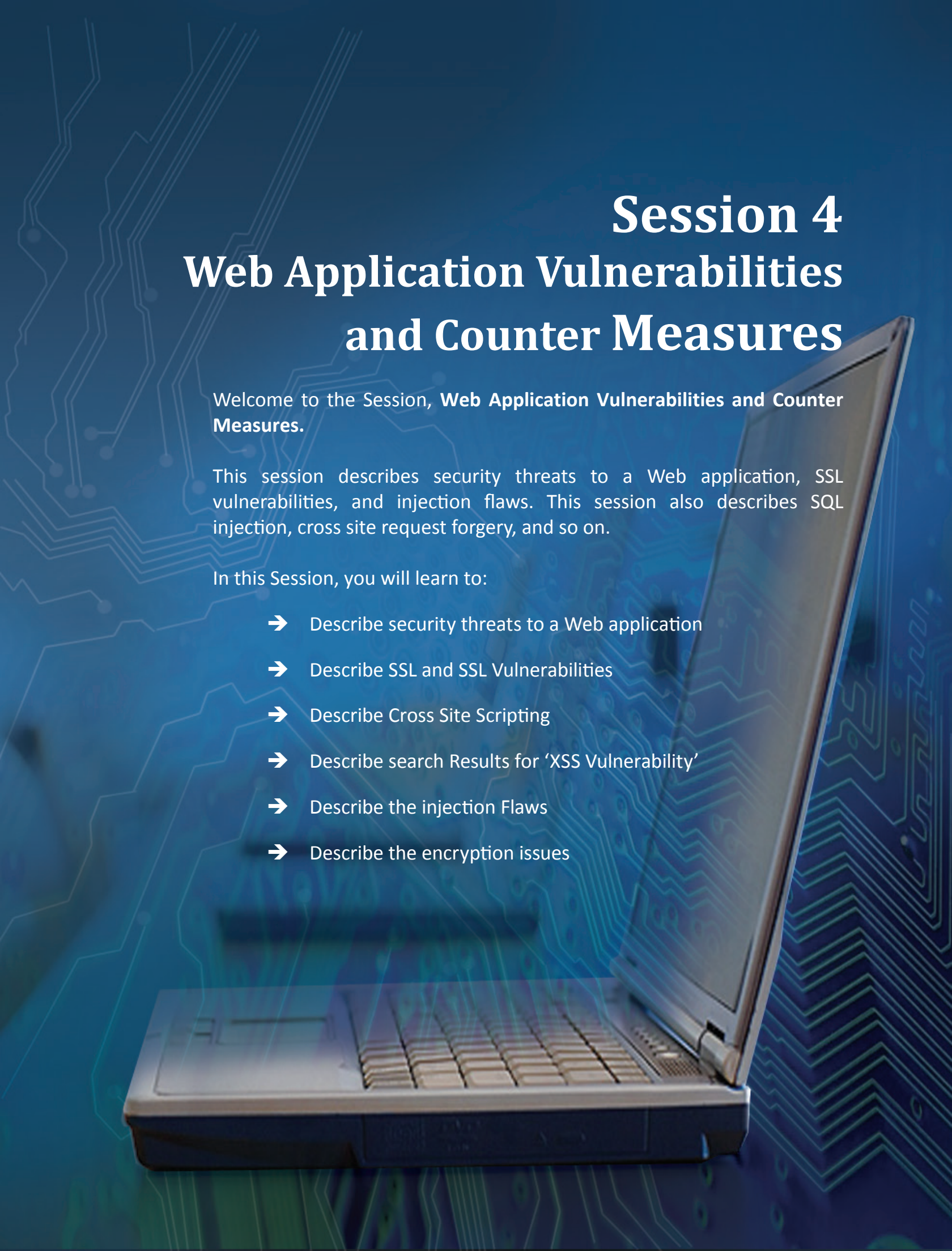
Session 4
Web Application Vulnerabilities and Counter Measures

Welcome to the Session, **Web Application Vulnerabilities and Counter Measures.**

This session describes security threats to a Web application, SSL vulnerabilities, and injection flaws. This session also describes SQL injection, cross site request forgery, and so on.

In this Session, you will learn to:

- Describe security threats to a Web application
- Describe SSL and SSL Vulnerabilities
- Describe Cross Site Scripting
- Describe search Results for 'XSS Vulnerability'
- Describe the injection Flaws
- Describe the encryption issues



4.1 Vulnerabilities and Threats of a Web Application

The acceptance of Web-based technologies involves great risk as e-commerce and other businesses are conducted through it. This is where the role of Web application security comes in. Although, Web applications have enabled the organizations to connect effortlessly with customers, suppliers, and stakeholders; a variety of previously unknown risks of security have been exposed by the Web application vulnerabilities.

If care is not taken for the Web application security, then not only the user's entire database which contains sensitive information is at risk, but also the user's Website may become the launch site for criminal activities such as transfer of illegal content or hosting other sites.

Hackers take advantage of this lack of security in a Web application and the vulnerabilities to perform Cross Site Scripting or SQL Injection. The attacker may also inject code maliciously within the vulnerable applications for tricking the users and also redirecting them to the other sites.

The following are some vulnerabilities and threats of a Web application:

- SSL Vulnerabilities
- Cross Site Scripting
- Injection Flaws
- Encryption issues
- Session vulnerabilities
- Malicious File Execution
- Cross Site Request Forgery

4.1.1 SSL Vulnerabilities

A Secure Socket Layer (SSL) can be defined as a standard security technology which is used to establish an encrypted link between the server and a client.

By using SSL, the transmission of sensitive information such as login credentials, social security number and, credit card number is done safely. Usually, the data which is sent between the Web server and the client is in plain text which makes it vulnerable to eavesdropping. If the data gets intercepted by the attacker while it is being sent to the Web server, the information may become visible and get misused.

It is necessary to protect the sensitive data by encrypting it, but applications fail to do so frequently. Encryption of information is needed for all the authenticated connections, not only for the Internet Web pages but also for the backend connections. An application which fails to do so and the one which can be forced out by the attackers from the encrypting mode, falls victim to the SSL vulnerabilities.

4.1.2 Cross Site Scripting

The most commonly used technique for hacking of Application layer is Cross Site Scripting (XSS). The XSS hacking technique refers to the vulnerabilities in the Web application code which allows the attacker to send the malicious content from end-users and also collect some data from the victim.

Using XSS, an attacker is allowed to embed malicious content to JavaScript, ActiveX, VBScript, Flash or, HTML for fooling the user and thus, the code is executed on the user's machine and the information required by the attacker is gathered. The use of XSS may result in security compromises such as exposing private information, stealing or manipulating cookies, executing malicious code on users system, or to create requests on behalf of a valid user. The data containing the malicious code is usually in the hyperlink format and is distributed on the Internet via every possible means.

The XSS URL can be formulated and distributed by the attacker as a hacking tool just by using the browser for testing the response of a dynamic Website. The Web page which passes parameters to the database such as forgot password forms, login forms, and so on can be vulnerable to this hacking.

In general, the XSS attack uses the malicious client-side script for infecting a legitimate Web page. The script is downloaded and executed when user visits the Web page. Figure 4.1 shows the basic pattern of XSS attack.

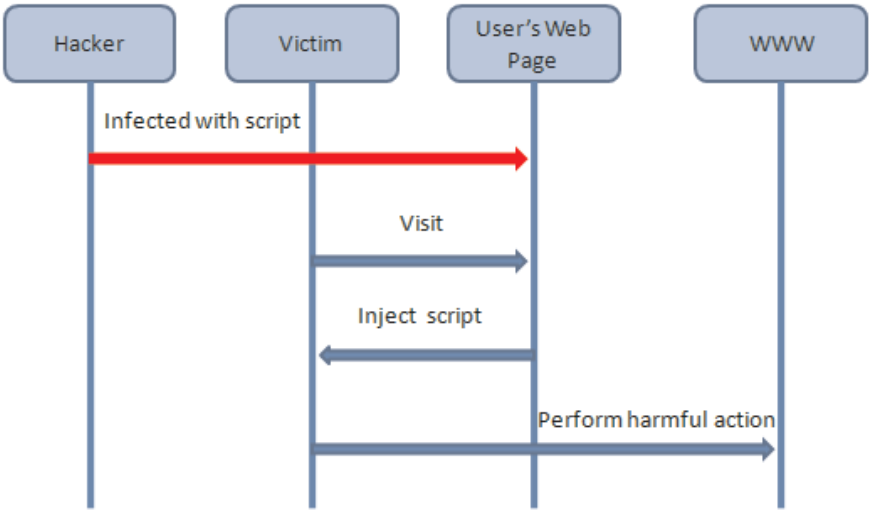


Figure 4.1: Basic Pattern of XSS Attack

The image depicts how a hacker may inject script in a Webpage which may later infect the target. Search engines often leave the entered text value in the URL address to be able to bookmark pages. For example, while searching content on XSS Vulnerability, the URL would be as follows:

```
http://test.searchengine.com/search.php?q=XSS%20Vulnerability
```

The following script is inserted by the attackers:

```
<script type="text/JavaScript">
alert ('This is an XSS Vulnerability')
</script>
```

Concepts

The code is a JavaScript code used for displaying a message box but the attackers uses it in such a way that the parameters in the URL are replaced by this script and the message will be displayed to the user instead of the result page.

After the submission of query to search.php, it will be encoded and the resulting URL will be similar to the following link:

```
http://test.searchengine.com/search.php?q=%3Cscript%3Ealert%28%91This%20
is%20an%20XSS%20Vulnerability%92%29%3C%2Fscript%3E
```

Thus, due to this query, while loading the result page of search engine no result would be displayed; instead the alert of JavaScript which is injected by using XSS vulnerability into the page would be seen.

4.1.3 Injection Flaws

Injection Flaws are the serious vulnerabilities to Web application wherein an attacker passes the malicious code from a Web application to some other system. Though any Web application which passes the data through an HTTP request to any external system can be vulnerable to a particular injection attack, but the Web applications which use Structured Query Language for interacting with the direct operating system calls, database back-end, or shell commands are more vulnerable to these attacks.

For prevention of the possible injection vulnerabilities, any data which is passed over an HTTP request to an external system should be scanned carefully to remove characters which the attacker may use for embedding executable code in a request. If a HTTP request is passed by a Web application without scanning to an external system, an attacker may gain access to the database backend, or may even execute the malicious shell commands or direct operating system calls with the permissions of that Web server.

SQL Injection is the most prominent form of injection flaws in Web applications which exploits the backend database. In order to exploit an SQL injection flaw, the attackers try to find specific parameters which a Web application passes across to a backend database and then insert the malicious SQL statements in those parameters. The attacker can corrupt the database or even reveal sensitive information present in the database if the Web application forwards the malicious SQL statements set by the attacker unknowingly to the backend database.

SQL Injection can happen with concatenated dynamic database queries where user supplies input. For example, in the following query, the parameter variable is a user specified value:

```
“select * from MYTABLE where name=” + parameter
```

In the query, parameter is the dynamic value passed which is a vulnerable injection flaws and can be changed by attackers.

JDBC example:

```
// input parameter
String empId = req.getParameter(“empId”)
String query = “SELECT * FROM Employee WHERE id = ‘” + empId +”’”;
```

The `getParameter()`method is used to get the dynamic parameter **`empId`** from the user input and then passed to the query in the where clause. Thus, **`empId`** can be changed by the attacker which can result in corruption of data, execution of malicious code on the database server, and so on.

4.1.4 Encryption Issues

The key part of most of the Web applications is to protect the sensitive data using cryptography. Failing to encrypt the sensitive data is a frequent mistake made during application development. Applications used for encryption are mostly with poorly designed cryptography which uses weak ciphers or use inappropriate ciphers.

General vulnerabilities in encryption are as follows:

➔ **Brute Force Cracking**

Brute Force is nothing but trial and error method. In this method, every possible key is tried by the attacker until the correct key is found. There is no encryption program which is completely safe from the Brute Force method, but if the number of keys is too large, then it can make it difficult for the attacker to use Brute Force. For example, there are 256 keys possible with a 56 bit key. Thus, it will be difficult for the attacker to try 72,057,594,037,927,936 times to find the correct key.

➔ **Back Doors**

A security hole in software is termed as back door. A back door may be present in the software by accident or the attacker may have intentionally created it. If the back door is discovered by the attacker, it can be used by them to find the key or password.

4.1.5 Session Vulnerabilities

Sometimes the session tokens and account credentials are not properly maintained and protected by the Web applications. This can lead to hacking of keys, passwords, or authentication tokens. For example, a cookie is received from the Website which contains session information. This cookie can be reused by the attacker later, if the session is not destroyed by the user.

The identification of an individual by a username and password is called as Authentication. Thus, if an identity is claimed by a user, the user should provide some proof such as username and password. If the password and username provided matches a record known to the application, then the user is an authenticated user. Authentication is an important aspect of security because it is the starting point in any further activity of the user within an application.

4.1.6 Malicious File Execution

Malicious file executions have different meanings for different people. The discussion here is limited to execution, either delayed or immediate, of files or file handles which can be modified in some way by the user input. The malicious file execution may occur in some of the following situations:

1. The file name is completely or partially determined by the dynamic input.
2. The file is written to the disk or uploaded to the Website without any validation performed on it.
3. If a command or data file is uploaded.

Thus, an attacker can manipulate the process in these three situations and gain privileges for accessing data and even execute unauthorized code.

Since these are not the only situations wherein the issue exists, let the user understand the common problems that may be present in the listed scenarios:

- **Missing or insufficient input validation** – This issue is most commonly observed during file processing. It deals with filename, file contents, and path. While uploading a file the issues caused are overwriting the password file and/or overwriting files in the file system. By this method, the attacker cannot only break the site but also own the site and even the server that is used for hosting the site.

- **No virus scanning** – It is similar to the input validation, but here, while uploading a file which may be executed anytime later is not scanned. As a result, there may be a possibility of infection to the computer if that file has been corrupted by an attacker.

- **No size checks** – The size limit is an important factor to be checked while uploading or downloading a file which is ignored by many users. There should be a limit to the size of file which is to be uploaded. For example, a spreadsheet can be of few MBs. However, if this size limit is not set, the attacker may try to upload a file of large size and may upload it many times to fill the file system due to which the server is blocked for the other valid users.

- **Invalid file type processing** – This is an important and difficult task in a file processing process and is required in many cases. If the Website is created for image hosting, a Word document should not be accepted if an attacker tries to upload it. Thus, there should be a whitelist created for verifying the types of files which are uploaded. This does not only mean checking the extension of the file. Checking extension can be first step in the process. This check often involves encoding as well.

  Note - A generic list of IP addresses or email address which can be considered as spam free is termed as whitelist. Whitelists are mostly used with emails applications for allowing the users to compile the sender's list from which they wish to receive emails from.

- **Direct Object Reference (DOR) problems** – This problem is encountered when the references to an internal implementation such as directory, file, key, or database record is exposed by the developer. These direct object references can be manipulated by an attacker for accessing other objects without authorization if the access control check for the application is not in place. Thus, DOR problems may result in broken authentication, elevation of privileges, or many other serious problems.

- **No output encoding** – This is not a very common issue seen every time. The basic problem in this is the file which is being written is not being validated properly. There can be many files with extra information in them and of different types. However, there are few files which can be read and then written to; ensuring that the unwanted data is not in the respective file. This does not mean checking for all unwanted data nor it is applicable to all file types but can be useful for some applications.

➔ **Unauthorized access** – This issue exists because the application has a proper authentication mechanism implemented but no appropriate authorization process available. The user once authenticated, can perform any function in the application. For example, in many of the applications, the link to visit the admin page may not be seen on the home page but if the URL is typed in the Address Bar nothing can stop the user from visiting the admin page. Also, depending on type of site, the harm may vary. Generally, only the trusted people can upload a file, but broken authentication may allow uploading of file from any user.

4.1.7 Cross Site Request Forgery (CSRF)

A victim's browser is forced to send a request in CSRF attack to a vulnerable Web application, which performs the action selected by the victim. The attacked site does not contain any malicious code. Therefore, this attack is known as 'Cross Site'.

Thus, if user is not protected against CSRF, he is vulnerable to it. This type of attack can be very harmful to several Web applications.

Following are the series of steps involved in the CSRF attack:

1. User logs in or is logged in already to 'Vulnerable Website Y'. This Website is used for some sort of e-commerce functionality. The site also allows saving the user's credit card details for future purchases.
2. An attacker realizes that CSRF is a vulnerability of the Website Y while using it.
3. This attacker crafts an html and/or JavaScript code and posts on the message board or forum that user reads which can be any site on the Internet.

   The code might look something like the following code:

   ```
   <imgsrc="http://www.siteX.com/completePurchase.do?itemId=ABC123" />
   ```

4. The user views the html and/or JavaScript code when the forum is visited by the user and the code is executed. User has just purchased an item of the attacker's choice without being aware of it.

4.2 Measures against the Security Threats

The security vulnerabilities and threats to a Web application must be addressed at the earliest in order to prevent further harm. The counter measures for the mentioned vulnerabilities are as follows:

➔ **Counter Measures for SSL Vulnerabilities**

SSL/TLS is used to encrypt the data when it is transmitted between server and client and vice versa. Protection may also be provided to data at the endpoints, but here the concentration is on protecting the data while it is being transmitted.

There are several ways to implement SSL/TLS correctly. Some of them are as follows:

1. Setup the application or Web server to only have an SSL/TLS enabled service listening. No http services should be running or at least the port should be blocked via firewall.

2. The security configuration settings of configuration file of a specific programming language/ technology can be used to guarantee safe transmission of data in applications.

Code Snippet 1 shows an example of applying security-constraint in the web.xml file of a Java Web application.

Code Snippet 1:

```
<security-constraint>
<web-resource-collection>
<url-pattern>/admin/*</url-pattern>
</web-resource-collection>
<user-data-constraint>
<transport-guarantee>CONFIDENTIAL</transport-guarantee>
</user-data-constraint>
</security-constraint>
```

The code requires URL with pattern '**`/admin/*`**'. The request, if sent using http, will be redirected to https due to the statement '`<transport-guarantee> CONFIDENTIAL</transport-guarantee>`' code. This method can be useful if the Web application is structured in such a manner that it can protect appropriate URLs by use of simple matching pattern of URL.

3. The Enterprise Security API (ESAPI) is an open source library and can make the task simpler. The `assertSecureChannel()` method of this API can be used for secure transmission of data. The syntax to use the method is shown in Code Snippet 2.

Code Snippet 2:

```
//this method can throw an AccessControlException
//uses the current http request
//there is an overloaded method that accepts a specified //http request
object as a parameter
ESAPI.httpUtilities().assertSecureChannel();
```

If SSL/TLS is not used in the transmission of data in this method, it does not redirect but instead an exception, `AccessControlException` is thrown. This exception can be caught and dealt appropriately. This code should be added in the J2EE filter configuration, if a page requires SSL and it can then execute on all requests ('`/*`').

4. There should be SSL/TLS for all login pages and also for any authenticated pages which are used after login.

5. Use http POST method instead of http GET method for transmitting sensitive information. In GET method the sensitive data can be seen as the part of requested URL so the data is protected by SSL/TLS in POST method.

6. All the cookies should be marked as secure to prevent attack on them which may cause a user to request a non-SSL/TLS URL from the application.

7. By staying aware and educated about the patches and technologies may help. SSL/TLS is not a new technology but a reasonably large numbers of such issues have been observed recently which have also been solved. The bugs are generally in implementation and not in the design.

8. Follow the legal regulations and policies for Web application development. For example, there are some legal requirements imposed for using SSL/TLS when using the SSN or credit card number to protect the data in transmission mode.

➔ **Counter Measures for Cross Scripting**

The basic technology solutions for XSS are as follows:

- **Canonicalize Input:**
  The input data should be provided in the simplest form for validation so that the user is confident that the validations are not avoided.

- **Validate Input Using a Whitelist:**
  Whitelist validation is the key for better security. So apply whitelist and also check the length, the content of the data, data type, and so on.

- **Contextual Output Encoding/Escaping:**
  Output encoding should be done properly as it is the last step. Developer must understand where the data is sent and how the location may be interpreted by the browser.

➔ **Counter Measures for Injection Flaws**

The simplest method for protecting against the injection flaws is by avoiding access to external interpreters as much as possible. For some system calls and many of the shell commands for performing same functions language, specific libraries are available. Large numbers of issues with the shell commands are avoided because using such libraries does not include the shell interpreter of the operating system.

For the cases where one cannot avoid the use of parameters such as the calls to backend database, the data provided must be validated carefully to checking for any malicious content. Also, many requests can be structured in a way such that all the parameters supplied may be treated as data, instead of the potential executable content. The use of prepared statements or stored procedures can be useful as they ensure that the supplied input is treated as data. Using the mentioned method may reduce the risk but cannot eliminate it.

Another strong method used to protect injection attacks is by setting proper privileges. Ensure that the Web application which is running is only with privileges that are absolutely needed for performing functions. Thus, the Web server should not access the database as DBA or run as root, otherwise these administrative privileges can be misused by the attacker.

If the use of external commands cannot be avoided, then any information from user which is to be inserted in a command should be checked properly. There should be mechanisms in place to handle any possible timeouts, errors, and blockages during calls. All the error codes, output, and return codes from the calls should be checked to ensure that the processing which occurred was appropriate and as expected. Also, this will help to determine if something has gone wrong otherwise the attack which happened could never have been detected.

➔ **Counter Measures for Encryption Issues**

In encryption method, software creates a password so the information can be read only by the intended recipients. Hence, it is important to select software for encryption which creates a strong password or uses a good algorithm so that the password cannot be guessed by the hackers or other malicious third party software.

The common approaches for solving these issues are as follows:

- **Stay educated**: As a result of increased computing power and new cryptanalytic attacks, good algorithms are somewhat depleting. Reading books about crypto would be of great help and may help in some of the cases.
- **Use a good algorithm**: Creation of crypto algorithm is a difficult task unless user is an expert in it. The algorithms used in the software for encryption is created by a team and is analyzed by peers for number of years before accepting it. Avoid using old and weak algorithms as they are more vulnerable to attack and can be broken in minutes or seconds now-a-days. So making the right choice of the algorithm is important.
- **Do good key management**: The key management is an important aspect. Depending on the requirement, the key management techniques can be selected. For example, there are options such as hard-coding a key, random key generation, and so on.
- **Encrypt credentials**: Every application connects to some database, Web service, or some other repository by using credentials. Thus, the credentials used to make these connections should be encrypted to protect data of storage systems from easy access.
- **Simplify**: Follow the old 'Keep it Stupid and Simple (KISS)' principle. When it comes to encryption, the least required settings should be done to store sensitive data safely and securely. The complexity may make it difficult to manage the security also.
- **Store the key/data/password separately**: Key and password are the pieces of information which are used for data protection mechanism. Storing all of these in one place makes is simple for the attacker to access them all together. If stored separately, it is more difficult for the attacker to access them.
- **Use filesystem controls**: The filesystem access control of the operating system can be used to protect data as a defence in depth task. However, protecting the key or password by using filesystem access control can be the best practice to secure data.

Note - The multiple security counter measures used in co-ordination to protect the information assets integrity in an enterprise is called as defense in depth. It is based on the military principle where it is difficult for the enemy to penetrate through multi-layer and complex defense system rather than a single barrier.

➔ **Counter Measures for Session Vulnerabilities**

The following are some measures to be considered to protect the Web application from session vulnerabilities:

- **Authenticate over a secure channel**: Authentication should use encryption mechanism to protect sensitive credentials when transmitted. This can help to protect the session vulnerability to a great extent.
- **Provide logout feature**: Logout option should be provided to the user. Placing a logout button or link can make it easy for the users to simply terminate the authenticated session as desired by them.
- **Provide related functions**: If the application has authentication mechanism, there should also be a functionality to update the credentials, specifically key or password, when required. The functions such as reset or change password should be implemented. In addition, if an attacker tries to brute-force the authentication, an account lockout process should prevent the application from being accessed. These functions may also go wrong so implementation should be done with great care.
- **Handle session management properly**: Unless there is a good reason, the built-in session management capabilities can be used in Java, .Net, or other applications. Thus, there is no need to implement a session handling concept and then use it.
- **Consider single sign-on**: Single sign-on is a very helpful option. The user is authenticated by each application or being redirected to a common authentication function by accepting the credentials and using the credentials for authenticating with the single user repository. So the actual authentication process is to be built only once and can be reused. The most important thing in this is to create the process correctly.
- **Use multi-factor authentication:** This method can increase the security in a greater way but is a costly measure for implementation. Many online systems are using this process in which a code is being sent to the phone that needs to be entered during the authentication process. Thus, an attacker needs the user's password as well as phone to be authenticated. This decreases the chances of vulnerability and is an excellent option.
- **Re-authenticate high value transactions**: The re-authentication technique should be implemented in the applications which are used for important transactions by an authenticated user such as wire funds transfer, e-commerce purchase, and so on. This ensures proper authentication of the user currently visiting the site.

➔ **Counter Measures for Malicious File Execution**

The following are the general best practices to deal with the issues of malicious file execution:

- **Missing or insufficient input validation** – ESAPI can be easily used to validate the user input through the framework. Code Snippet 3 checks the filename of the uploaded file to verify if it is a valid file.

Code Snippet 3:

```
if (!ESAPI.validator().isValidFileName("upload", filename,
allowedExtensions, false)) {

throw new ValidationUploadException("Upload only files with simple
names and with the following extensions:" + allowedExtensions,
"Upload failed. Is the file valid? Please check.");

}
```

- **No virus scanning** – Direct virus scan is not supported in ESAPI but several antivirus vendors support the API access. Users can scan files using the antivirus software. The file should be deleted if the scan fails and the incident should be logged for any further reference.
- **No size checks** – There are many Web frameworks which supports checking of file size. When there is no support provided from built-in libraries, it is important to perform this check manually using custom code. Just refer the file to a file object and call the `length()` method which will usually return the size in bytes.
- **Invalid file type processing** – The input validation portion of the ESAPI performs some of this task. The application can validate the file name. The content of the file is then processed and validated by the application.
- **Direct Object Reference (DOR) problems** – To solve this issue, the interface `AccessReferenceMap` of ESAPI can be used. This interface has methods that allows a user to add a direct reference and generate an indirect reference for it, and then by using indirect reference retrieve the direct reference, and vice versa. Also, the `RandomAccessReferenceMap` class of ESAPI can be used to generate a random reference for a direct reference.
- **No output encoding** – The encoding of output can be done in many ways but user has to consider the different file types. There are many third party libraries available which can be helpful for processing the file types.
- **No authorizing access** – Authorization process should be performed for all pages so that the admin rights or privileges are not exploited by any other user. The solution for authorization is done by restricting access to the site URL.

➔ **Counter Measures for Cross Site Request Forgery**

ESAPI can be used to solve the CSRF issue. The steps to protect the Web application from CSRF attacks are as follows:

1. A new CSRF token should be generated which should be added once the user login and it should be stored in the http session. Code Snippet 4 is to be executed when the users login to the application. In ESAPI, this is done by default and the token is stored as a member variable of the `User` object and thus, stored in the session as well.

Code Snippet 4:

```
//this code is DefaultUser implementation of ESAPI

/** This user's CSRF token. */

private String Tokencsrf = resettheCSRFToken();

...

public String resettheCSRFToken() {

     Tokencsrf = ESAPI.randomizer().getRandomString(8,
DefaultEncoder.CHAR_ALPHANUMERICS);

     return Tokencsrf;

}
```

The `ESAPI.randomizer().getRandomString()` is used for generating this random token which will be used later for authorizing the session.

2. Add token as a hidden field or parameter on the other form or the URL that is to be protected. Code Snippet 5 is used for passing the parameter in the URL.

Code Snippet 5:

```
//from HTTPUtilitiles interface

final static String CSRF_TOKEN_NAME = "ctoken";


//this code is from the DefaultHTTPUtilities implementation in //
ESAPI

public String addCSRFToken(String href) {

     User user = ESAPI.authenticator().getCurrentUser();

     if (user.isAnonymous()) {

          return href;

     }
```

Concepts

```
// if there are already parameters append with &, otherwise
//append with ?
      String token = CSRF_TOKEN_NAME + "=" + user.getCSRFToken();
      returnhref.indexOf( '?') != -1 ? href + "&" + token : href +
"?" + token;
}
...
public String getCSRFToken() {
      User user = ESAPI.authenticator().getCurrentUser();
      if (user == null) return null;
      return user.getCSRFToken();
}
```

The method addCSRFToken( ) should be called on redirection to a page which needs CSRF protection and the CSRF token is passed in the URL . To access the token which is created in the first step, the getCSRFToken( ) method is used and the token is used for verifying the session.

3. Check on the server side if the submitted token is same as that of the token stored in session.
The method should be called from struts action or servlet in Java or any server side mechanism used in the application to handle request. This should be done for all requests for validating it and protecting it against CSRF. If the token does not match the request it is consider to be forged.
Code Snippet 6 shows the implementation of the process for authenticating the user based on the token generated earlier.

Code Snippet 6:

```
//this code is from the DefaultHTTPUtilities implementation in ESAPI
public void verifyCSRFToken(HttpServletRequest request) throws
IntrusionException {
      User user = ESAPI.authenticator().getCurrentUser();

// check if user authenticated is with this request - no CSRF //
protection required
```

```
     if(request.getAttribute(user.getCSRFToken()) != null ) {
          return;
     }
     String token = request.getParameter(CSRF_TOKEN_NAME);
     if ( !user.getCSRFToken().equals( token ) ) {
          throw new IntrusionException("Authentication failed",
"Possibly forged HTTP request without proper CSRF token detected");
     }
}
```

The code uses the `getCSRFToken()` to retrieve the token for the current session and then, it is verified in the `verifyCSRFToken()` method. Thus, if the token matches, the user is redirected successfully to the requested page else, the `IntrusionException` is thrown.

4. By removing the `User` object from session on session timeout and logout, the session is destroyed. Code Snippet 7 shows the implementation of `logout()` method. The session is invalidated and the current user object is reset by setting it to anonymous user.

Code Snippet 7:

```
//this code is in the DefaultUser implementation of ESAPI
public void logout() {
     ESAPI.httpUtilities().killCookie(ESAPI.currentRequest(),
ESAPI.currentResponse(),  HTTPUtilities.REMEMBER_TOKEN_COOKIE_NAME
);
     HttpSession session=ESAPI.currentRequest().getSession(false);
     if (session != null) {
          removeSession(session);
          session.invalidate();
     }
     ESAPI.httpUtilities().killCookie(ESAPI.currentRequest(),
ESAPI.currentResponse(), "JSESSIONID");
     loggedIn = false;
     logger.info(Logger.SECURITY_SUCCESS, "Logout successful" );
     ESAPI.authenticator().setCurrentUser(User.ANONYMOUS);
}
```

The `logout()` method is to be called on session timeout or logout to delete the data related to the current session and thereby, invalidate the session so that it cannot be used by an attacker.

In this manner, a solid and efficient protection against CSRF is established.

4.3 Check Your Progress

1. _____ can be defined as a standard security technology which is used to establish encrypted link between the server and a client.

(A)	Secure Socket Layer	(C)	Cross site Scripting
(B)	Injection	(D)	CSRF

2. Match the following malicious file issues with the corresponding description/example.

	Malicious File Issue		Description/Example
a.	No size checks	1.	This issue is most common in the file processing process which deals with filename, file contents, and path.
b.	Missing or insufficient input validation	2.	The direct object references can be manipulated by an attacker for accessing other objects without authorization.
c.	Direct Object Reference (DOR) problems	3.	The file which is being written is not being validated properly.
d.	No output encoding	4.	The attacker may try to upload a file of large size and may upload it many times to full the file system due to which the server is blocked for the other valid users.

(A)	a-2, b-4, c-1, d-3	(C)	a-4, b-1, c-2, d-3
(B)	a-4, b-1, c-2, d-3	(D)	a-2, b-3, c-4, d-1

3. Which of the following statements about injection flaws are true?

a.	The use of prepared statements or stored procedures can be useful as they ensure that the supplied input is treated as data.
b.	Ensure that the Web application which is running is only with privileges that are absolutely needed for performing functions.

(A)	Statement b	(C)	Statements a and b
(B)	Statement a	(D)	None of these

4. Which of the following are basic technological solutions for Cross Site Scripting?

(A)	Canonicalize Input	(C)	Validate Input Using a Whitelist
(B)	Contextual Output Encoding/ Escaping	(D)	All of these

5. A security hole in software is termed as _____.

(A)	Trojan	(C)	Brute force
(B)	Back door	(D)	None of these

6. Which of the following are approaches for handling encryption issues?

(A)	Do good key management	(C)	Stay Educated
(B)	Use filesystem controls	(D)	All of these

4.3.1 Answers

1.	A
2.	C
3.	C
4.	D
5.	B
6.	D

Summary

- Secure Socket Layer (SSL) can be defined as a standard security technology which is used to establish encrypted link between the server and a client.
- The XSS hacking technique refers to the vulnerabilities in the Web application code which allows the attacker to send the malicious content from end-users and also collect some data from the victim.
- If user supplied data is directly used to generate queries and used in the interpreter, then the respective Web application becomes vulnerable to the Injection flaws.
- The identification of an individual by a username and password is called as Authentication.
- A victim's browser is forced to send a request in Cross Site Request Forgery (CSRF) attack to a vulnerable Web application, which performs the action selected by the victim. The attacked site does not contain any malicious code. Therefore, this attack is known as 'Cross Site'.
- The simplest method for protecting against the injection flaws is by avoiding access to external interpreters as much as possible.
- In encryption method, software creates a password so the information can be read only by the intended recipients. Hence, the key must be strong and created using a strong algorithm.
- ESAPI, which is open source and free, can be used to solve most of the Web application security vulnerabilities.

Onlinevarsity

your e-way to learning

Technowise

Are you a

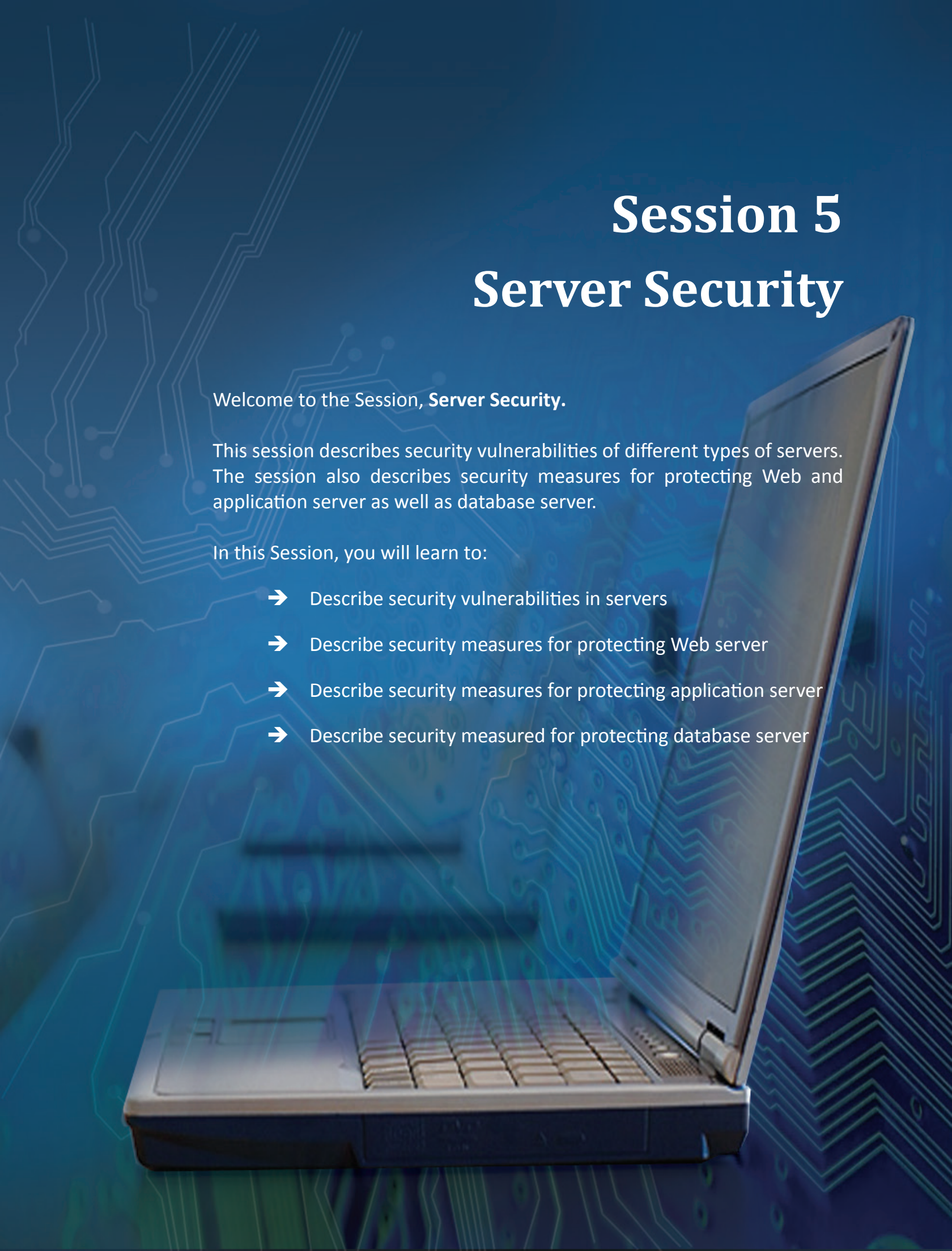
TECHNO GEEK

looking for updates?

Login to

www.onlinevarsity.com

Session 5
Server Security



Welcome to the Session, **Server Security.**

This session describes security vulnerabilities of different types of servers. The session also describes security measures for protecting Web and application server as well as database server.

In this Session, you will learn to:

- ➔ Describe security vulnerabilities in servers
- ➔ Describe security measures for protecting Web server
- ➔ Describe security measures for protecting application server
- ➔ Describe security measured for protecting database server

5.1 Introduction to Server

A server is defined as a system which responds to the requests sent by a computer in a network to provide data, help, or some network service. A server can also be a program and thus, the computer on which the program runs is usually referred to as a server. There may be many other different programs running on it.

Servers are usually a part of the client-server architecture. The client is served by the server by providing requested service or data through a response. That is, the requested tasks are performed by the servers on behalf of clients. In the Internet Protocol networking, a socket listener program is termed as server.

Essential services are provided by the server across a network, either to public users through Internet or to private users within an organization. There are several types of servers such as Web server, application server, database server, file server, mail server, gaming server, print server, and so on.

5.2 Types of Servers

The following are types of servers:

- **Web Server**: A Web server is a computer that hosts the Web components related to a Web application. It helps to deliver the Web contents which are accessed via the Internet. Thus, the use of Web server is to host Websites and data storage.

- **Application Server**: An application server is a computer that can host Web as well as enterprise applications. It can either be a specific implementation instance server portion or a software framework which provides application-server implementation as a general approach without considering the function. Thus, the server is dedicated for efficient execution of programs, scripts, routines, and so on to support the different types of applications.

- **Database Server**: The database server is a computer that hosts the data store of a Web and/or enterprise application. The Structured Query Language (SQL) requests are sent to the server via the application and the result of the query is returned as a response. The data resides on the server within a database. The server can use its own processing capabilities when a SQL request received instead of sending all the data with it to the client as in case of the file server.

- **Mail Server**: A computer which serves as the electronic post office for emails is called as mail server. The mail servers run specifically designed software and the mails which are exchanged across the network are passed between them. The software used by the mail servers for handling emails and attachments in emails are based on the standardized protocols.

- **Proxy Server**: Proxy server lies between client and a server (another server on Web) for filtering requests, sharing connections, and improving performance.

- **Telnet Server**: The user is able to logon to a host computer and perform the tasks as if the user is working on the remote computer itself using telnet server.

➔ There are many other types of servers such as FTP server, virtual server, chat server, and so on.

5.2.1 Security Vulnerabilities in Servers

Web server may be exploited or hacked by hackers due to security vulnerabilities. The most common vulnerabilities in Web servers are:

➔ **Misconfiguration of the Web Server Software**

The default settings of the server can leave the Websites open for an attack. It is important to restrict and edit permissions of the users using the Web Server. Also, the users having access to the default Website can access the default folder and will have privileges such as execute and full control of the files.

➔ **Flaws in the Operating System or Programming Code**

All programs should be patched and updated including Web server applications and OS on a regular basis. The patches can be applied to the systems manually or automated after testing them.

➔ **Vulnerable Default Installation**

The Web server and OS should not be left with default configuration for a long time when installed as it may lead to security breaches.

Application servers should be isolated from direct Internet access for preventing security breaches. The main threats to an application server are:

➔ Unauthorized access

➔ Network eavesdropping

➔ Viruses, Trojan horses, and worms

Figure 5.1 shows the threats to an application server.

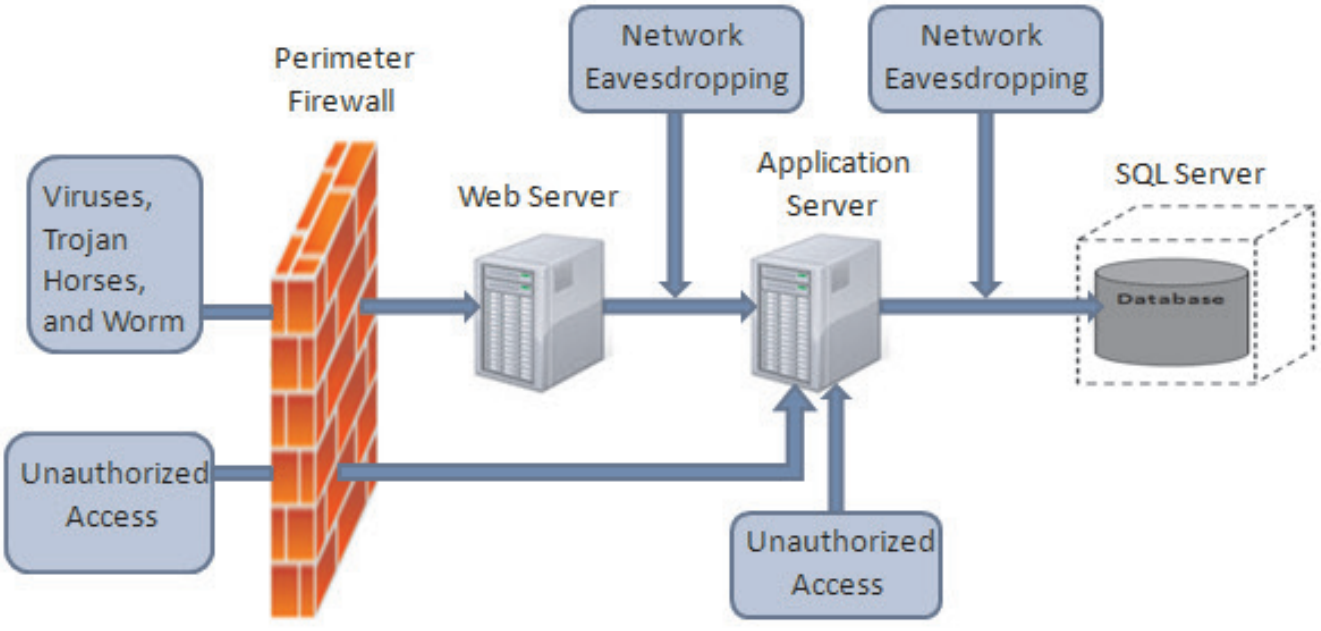


Figure 5.1: Threats to an Application Server

➔ **Unauthorized Access**

The external attacker can communicate with the application server directly if the ports used by an application running on the server are not blocked at the perimeter firewall. If computers other than front-end Web servers are allowed to connect to the application server, the chances of attack on the server increase considerably.

Vulnerabilities which may result in unauthorized access are:

- Firewall configurations and weak perimeter network
- Open superfluous ports on the firewall
- There may be no IPSec policies for restricting host connectivity
- Unwanted active services
- Unwanted protocols
- Weak password and account policies

Common attacks used to gain unauthorized access are:

- Port scanning which detects listening services
- Malicious application input
- Banner grabbing which gives away software versions and available services
- Password attacks against weak passwords of default accounts

Concepts

Counter measures for preventing unauthorized access are:

- Block all traffic except communication ports which are necessary using firewall
- IPSec policies or TCP/IP filtering for preventing unauthorized hosts to establish connections
- Static DCOM endpoint mapping which allows access only to hosts that are authorized
- Disable unused services

➔ Network Eavesdropping

The network data can be intercepted by attackers using network monitoring tools which intercept data moving between application server and Web server. The data can be viewed as well as modified by the attackers.

Vulnerabilities which can make application server fall to eavesdropping technique are:

- Sensitive data transmitted by the application in clear text
- Lack of application or transport layer encryption
- Interfaces of network-hardware administration are insecure
- Using .NET Remoting TCP Channel to connect remotely

The packet-sniffing tool can be placed by the attacker for capturing messages transmitted on the network.

Following are the counter measures to prevent packet sniffing:

- Use secure authentication methods where passwords are not sent without encrypting over the network
- With Enterprise Services applications, use Remote Procedure Call (RPC) encryption
- With .NET Remoting, use the Http Channel and SSL
- Secure communication channels using Internet Protocol Security (IPSec) or Secure Sockets Layer (SSL)
- Use a segmented network that can protect the data from eavesdropping

➔ Viruses, Worms, and Trojan Horses

The attacks from Viruses, Trojan Horse, and Worms are noticed when they begin to consume and utilize the system resources leading to system slow down and halting execution of other applications.

The following are vulnerabilities due to Viruses, Worms, and Trojan Horses:

- Unpatched servers
- Running unnecessary services

- Unnecessary ISAPI filters and ISAPI extensions

Counter measures that help mitigate the risk posed by viruses, Trojan horses, and worms include:

- Apply the latest software patches regularly.
- Unused functionalities should be disabled.
- Run processes with least privileged accounts for reducing the scope of damage in case of a security compromise.

5.3 Security Vulnerabilities in Database Servers

Data is the core part of Web applications which is stored on the database server. Hence, it is very important for all resources involved in the development of a Web application to be aware of the security vulnerabilities in a database server.

The following are some common security vulnerabilities in database server:

➔ **SQL injections**

SQL injection, the most commonly observed vulnerability of database servers, is a critical problem encountered in the database server. The database administrator is then left to handle the application after it is being attacked by the injection. The malicious code and variables are inserted into strings which are passed to the instance of SQL server later for parsing and execution. Firewall protection and input validation can be two important methods to prevent SQL injection.

➔ **Deployment failure**

A database server is vulnerable to attacks mostly due to inappropriate deployment of the database on the server. Perform some functionality test on the server and ensure that it is working as designed and as per the expectations. Few checks should be done to ensure proper working of the server such as checking the server configuration, device connection sensitivity settings for the devices which will be connected to the server, and so on.

➔ **Data leaks**

Database is the back-end part of a Web application and may be considered as secured from Internet-based threats, but it is not so. Database can also be hacked by hackers for exploiting it since it also contains networking interfaces. Administrators should use TLS or SSL encryption to avoid this.

➔ **Stolen database backups**

External attackers who try to steal data are a threat , but even insiders from the organization can prove to be a threat and steal archives such as database backups for money, retribution, or profit. The archives should be encrypted to reduce the risk of data theft.

➔ **The abuse of database features**

It has been observed during research that majority database attacks in the past few years have been performed simply by misusing the features of a database. Access to the server can be gained by manipulating simple flaws and any service or code could be run by the attacker by using standard features and tools of database server. Future risk can be reduced by removal of unwanted tools.

➔ **A lack of segregation**

Segregation of user and administrator power can make it difficult for the attacker to attack. Also, limiting the power of user accounts can make it harder for the attacker to completely gain control over the database.

➔ **Hopscotch**

Sometimes rather than attacking directly, the attacker waits to find a weakness in the system that can be taken advantage of and can be used for more serious attacks until the backend system is reached. Thus, attackers play a game of hopscotch wherein they do not take advantage of issues such as buffer overflow to gain access to the database, but they try to find infrastructural weakness which can be used for an effective and serious attack.

Note - Hopscotch is a game that can be played alone or with several players. In the game, the players toss a small object into numbered blocks drawn on the ground in the shape of rectangles and then jump or hop through the blocks to retrieve the object.

5.4 Security Measures for Servers

The common security measures for all the servers are strong passwords, patch or update software, backups, scanning, and so on. The detailed description for security measures of Web server, application server, and database server are mentioned here.

5.4.1 Security Measures for Web Server

Ensuring Web server security is one of the most important tasks as the legitimate public users should be allowed to access the Web resources. At the same time, it is also necessary to keep away the unauthorized users trying to access the server. The following are the techniques to protect the Web server:

➔ **Use separate servers for internal and external applications**

There should always be separate classes of Web application for internal and external users so that they can be placed on two different servers. This reduces the risk of external users to penetrate and gain access to internal sensitive information of an organization. Technical controls can be used to separate the external and internal application if no resource is available.

➔ **Use a separate development server for debugging and testing apps**

Testing should be done on standalone servers. However, all organizations do not follow this; rather developers develop new applications or are allowed to change code directly on the production server. There is extreme security and reliability risk involved in such practices. Testing code or application on the production server may cause interruptions in the sessions of users. Also, security vulnerability may be introduced as the code is not tested by the developer and it could be vulnerable to attack. Also, modern version control systems can be used to automate the coding, debugging, and testing processes.

➔ **Store logs in a secure location and audit Website activity**

Maintaining activity logs is an important part of security as the Web servers are used for Internet-based services. Detection of vulnerabilities may be possible due to these audit trials and will also help to define measures to be taken in future against such attacks. Sometimes audit trial may enable troubleshooting of server performance issues. Ensure the safety of the logs by storing them physically in secure location. Log modification or snooping can be prevented by implementing encryption on the host used for storing the logs.

➔ **Educate developers on sound security coding practices**

Generally, a developer focuses more on the business requirements while creating apps but security of information is also one of the critical requirements which is often overlooked. Developers should be educated about the security issues which affect the Web servers. The developers should be aware of the security mechanism on the network to ensure the applications created by them do not evade or sidestep those mechanisms.

➔ **Patch the Web server and operating system**

This is usually overlooked by the administrator when he/she is overburdened with other tasks. The Web server should be regularly patched with recent security fixes. This task can be automated using tools such as Red Hat's up-to-date service and Microsoft's Software Update Service (SUS). In a similar manner, the Operating system should also be patched on regular basis.

➔ **Use application scanners**

Consider using application scanner if affordable for validating internally developed code. Many tools can be used to detect the exploitable code and thus prevent it from entering the production server undetected.

5.4.2 Security Measures for Application Server

The following are the measures for protecting application server:

➔ **Strong passwords**: Default or poor passwords such as 12345 are the most vulnerable aspects that attackers look for while attacking. The most important measure is to use strong passwords.

Login access should be filtered (for example, based on IP origin in the firewall). Users can use their own controlled password dictionary for setting the passwords.

➔ **Patch/update software**: Services and applications such as Customer Relationship Management (CRM) applications which are not updated regularly have vulnerabilities which hackers look for and try to exploit. Update the systems and software regularly. Run external or vulnerability assessment periodically.

➔ **Remove unwanted services**: The services which are not needed or used should be removed as these can be a mode for the intruders to penetrate in the system because the services are exposed. Sometimes the organization is not even aware about the unwanted services as they were installed by default during installation of the server in the past. So remove the unwanted software packages and periodically run scanners to scan ports. The services such as DNS or email can be outsourced to experienced vendors.

Other protection and hardening measures:

The following methods can be used:

- Use a firewall to block the default ports which are unused, implement basic safety measures such as limiting number of connections at a time to prevent the denial-of service attacks and spoofing.
- By using source-based IP filtering or port-knocking, protect control panel, or remote management access points.
- Follow the vendor's recommendation to harden a specific service which is made public.
- Do not disclose the information such as version number and name in the public application banners, signatures, or pages.

➔ **Logs and Monitoring tools**: Logs can be a very useful tool for the system administrator to detect any intrusion or warnings. Auditing tools or software can be used to archive/manage logs. Sometimes there may be a sudden increase in the CPU or bandwidth utilization which may be a warning for a security issue.

Thus, server resources utilization can be monitored using monitoring tools.

➔ **Rootkit detectors**: The software can be used to detect an exploit in the system by scanning system files. Antimalware software or anti-virus software can be used for detecting rootkits.

➔ **Recovery**: Recovery of a server is possible mostly by having a good backup strategy such as automated backups, several levels of backups, and frequent backups.

5.4.3 Security Measures for Database Server

Data security is an important aspect for every business. Following are the measures for protecting the database server:

➔ **Remove the blank, default, and weak password/username**

It might be hard task for an organization to keep record of hundreds or even more databases. However, removing the blank or default and weak log-in username and password is the first and most important step to secure the database. The attackers may be keeping track on the default accounts and may try to attack when possible.

➔ **Put up a strong firewall**

Firewalls are a must in order to protect the network adequately. The firewalls protect by controlling the ingoing and outgoing traffic from the system.

➔ **Install antivirus protection**

Anti-malware or antivirus software is a must for protecting from the threats such as virus, worms, and so on which can infect the system. They can be a last line of defense as they will protect the system in case the infected elements enter the system.

➔ **Update programs regularly**

The entire program in the computer should be patched and updated on a regular basis. There is no point in installing software if they are not proper and regularly updated. Since all the security features used are not hundred percent fool proof, it is recommended to regularly update the tools to keep the system safe. Also, it can help to stay up-to-date on the recent loop holes or issues which have been fixed by the programmers.

➔ **Backup regularly**

Regular backups should be scheduled to an external hard drive for ensuring that the data is safely stored. Servers should have a complete backup weekly and an incremental backup every night. Data getting compromised can prove to be a great damage but having a backup of it can help to minimize the damage.

➔ **Monitor diligently**

Technology proves to be good after it is being utilized. Appropriate monitoring tools can be used for monitoring the data as well as server activities. The tool should be configured for looking at any malicious code or any information relevant to business which may indicate breach. Even user can audit logs, error messages, and warnings. If monitoring is not done regularly or properly, the compromise in security may be detected very late or at times may not be detected at all.

➔ **Be careful with email, Instant Messaging, and surfing the Web**

The emails received from an unknown source having strange attachments can possess malicious content. Downloading the content or clicking the link of such emails may result in a nasty infection to the computer due to virus or other threats. Users should act smartly while surfing on the Web as well. Read all the warnings shown and take them seriously and also, understand that all the software comes with its own set of vulnerabilities.

5.5 Check Your Progress

1. _____ is a system which responds to the requests sent by a computer in a network to provide data, help, or some network service.

(A)	Server	(C)	Router
(B)	System	(D)	None of these

2. Match the following servers with the corresponding description.

	Servers		Description
a.	Proxy Servers	1.	A computer which serves as the electronic post office for emails.
b.	Mail Server	2.	The user is able to logon to a host computer and perform the tasks as if the user is working on the remote computer itself.
c.	Telnet Server	3.	The computer or software that helps to deliver the Web contents which are accessed via Internet.
d.	Web server	4.	Lies between client and a server for filtering requests, sharing connections, and improving performance.

(A)	a-2, b-4, c-1, d-3	(C)	a-3, b-4, c-1, d-2
(B)	a-4, b-1, c-2, d-3	(D)	a-2, b-3, c-4, d-1

3. Which of the following statements about SQL injection are true?

a.	Input validation and firewall are the basic protection methods which prevent SQL injection.
b.	SQL injection can be a helpful method for protecting the database server.

(A)	Statement b	(C)	Statements a and b
(B)	Statement a	(D)	None of these

4. The _____ tool is placed by the attacker to capture traffic on the network.

(A)	Patch	(C)	Firewall
(B)	Scanner	(D)	Packet-sniffing

5. Which of the following are common security measures for servers?

(A)	Scanning	(C)	Strong password
(B)	Patch or update software	(D)	All of these

6. _____ can be a very useful tool for system administration to detect any intrusion or warnings.

(A)	Scanner	(C)	Backup
(B)	Logs	(D)	None of these

5.5.1 Answers

1.	A
2.	B
3.	B
4.	D
5.	D
6.	B

Summary

- A server is a system which responds to the requests sent by a computer in a network to provide data, help, or some network service.
- The different types of servers include Web server, application server, database server, FTP server, virtual server, chat server, and so on.
- The default settings of the server can leave the Websites open for an attack.
- It is important to restrict and edit permissions of the users for all types of servers.
- The common security measures for all the servers are strong password, patch or update software, backups, scanning, and so on.
- Logs and monitoring tools can be used to detect the intrusions, errors or warnings, and so on.
- Educating developers and resources operating the servers regarding server security vulnerabilities is important to prevent security risk.

Onlinevarsity

your e-way to learning

GROWTH

RESEARCH

OBSERVATION

UPDATES

PARTICIPATION

www.onlinevarsity.com

Session 6

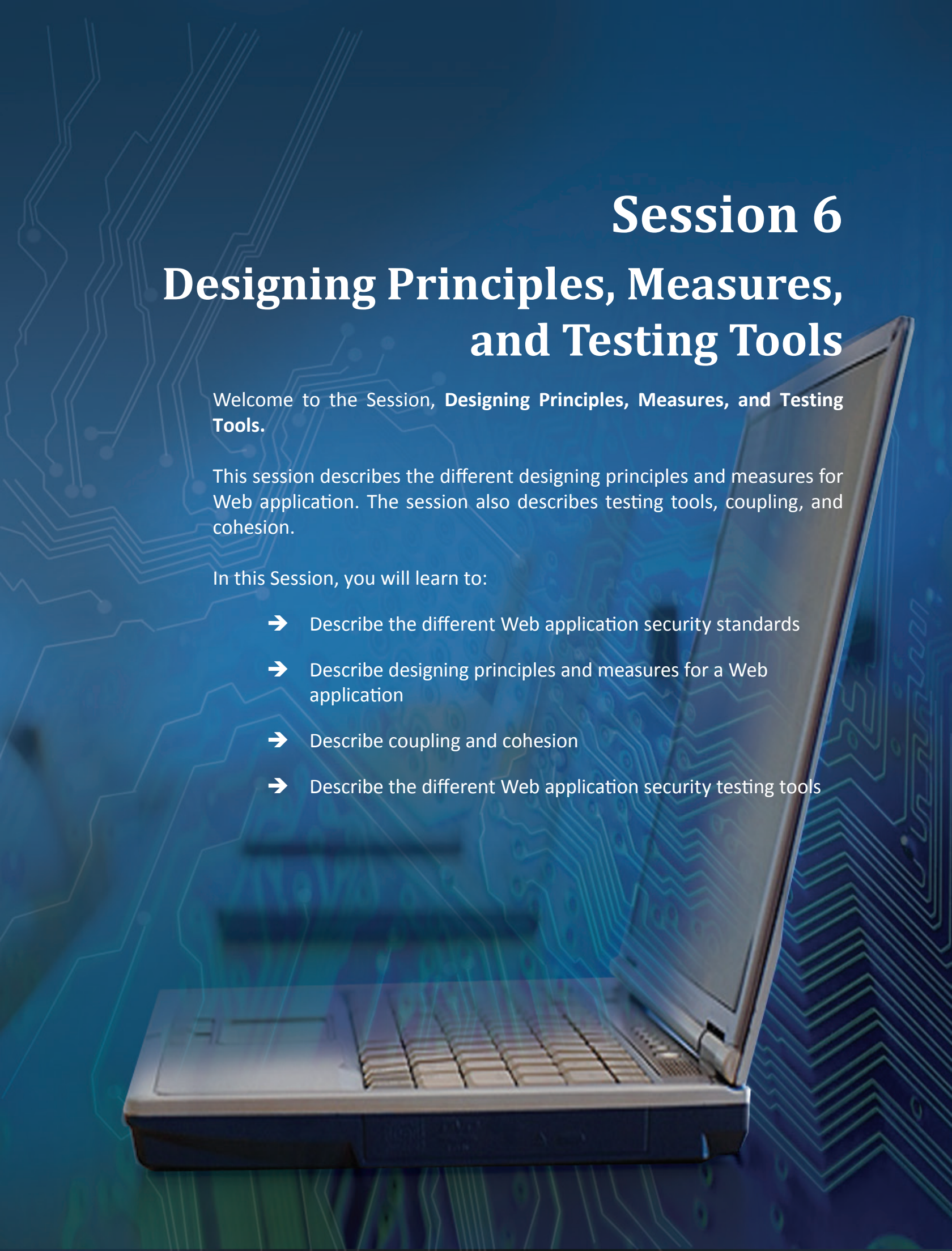
Designing Principles, Measures, and Testing Tools

Welcome to the Session, **Designing Principles, Measures, and Testing Tools.**

This session describes the different designing principles and measures for Web application. The session also describes testing tools, coupling, and cohesion.

In this Session, you will learn to:

- Describe the different Web application security standards
- Describe designing principles and measures for a Web application
- Describe coupling and cohesion
- Describe the different Web application security testing tools



6.1 Web Application Security Standards

Web applications are to be designed using standardized protocols. The following list of standards will help to build a more secure Web application:

6.1.1 Server Configuration and Access Control

- A production Web server should not contain any unused files.
- The production Web server should not be used for development activities. Even testing should be done on different environment.
- The least permission principle must be used in file permissions.
- The account on the Web server should be with minimum privileges.
- The users should not have write access within the Web root.
- Only execution permissions should be granted on the files if necessary.
- The execute permissions should not be permitted on static HTML pages.
- The Directory viewing should be disabled on the Web server.
- For Web application which needs authentication, make sure to recheck the credentials for all the resources.
- The server name should not be used for user access; instead DNS aliases should be used for accessing the Web application.

6.1.2 Authentication

- All authentication must use encryption while transmitting data on network.
- The authentication or logon cookies should not be persistent.
- Depending on the legal requirement of the application, log the instance of users who are logged in.
- Input validation must be performed on the login form.
- The username, hostname, or database name should not be hard coded in the application.
- Directory service can be used for authentication where ever possible.
- User accounts should be locked after 60 days of inactivity where applications do not use Lightweight Directory Access Protocol (LDAP) authentication.
- The database server should verify the identity of the Web server which sends the requests.

Note - A software system which stores, provides access, and organizes information in a directory is called a directory service. In software engineering, a map between names and values is called as a directory. The lookup of values, given a name, are allowed in directory services, like a dictionary. For example, Domain Name System (DNS) server maps computer hostnames to IP addresses.

An application protocol for maintaining and accessing distributed directory information services over an Internet Protocol network is called as Lightweight Directory Access Protocol (LDAP).

6.1.3 Data Validation

On server side, all data input should be validated. Client side validation cannot be used as a sole control for data input validation. The only strategy which is sustainable is Accepting Known Good Data for data validation.

6.1.4 Database Interfaces

- Database name, hostnames, passwords, and usernames should not be stored within the code base.
- The data which is classified as public can be hosted on a single tier Web application. Confidential, internal, and sensitive data should be hosted on the Web application with multi-tier architecture.
- Database engines should be used which allow security permissions to an individual database object only.
- The Web accounts must not be default accounts and must not have administrative privileges.

6.1.5 Authenticated Session Management

- Always a new session ID should be generated for each login.
- The session ID should be assigned in the Web applications where authentication is necessary after the login process is completed successfully.
- The session IDs generated by the application should be random and with minimum of 10 alphanumeric characters in it.
- The authenticated session ID can be protected by using encryption while submitting the data between the server and the client.
- The session data is not used as identification for the user.
- At logout, the session related information should be destroyed by the server.

Open Web Application Security Project (OWASP)

The Open Web Application Security Project (OWASP) is a devoted open community for finding and fighting the reasons of insecure software. All the OWASP documents, forums, tolls, and chapters are open and free to anyone interested in improving the application security. It is a new type of entity in the security market. As there is no commercial pressure in OWASP, it allows providing cost-effective, practical, and unbiased information about application security. OWASP is not officially connected or associated to any technology company. The OWASP supports approaching application security as a people, technology, and process problem. Additional information of OWASP can be obtained from its official site, http://www.owasp.org/.

6.2 Web Application Designing Principles and Measures

The following are the design principles taken from architectures which have scaled and performed well over time:

- **Design coarse-grained services**: The client-server interactions are minimized and thus, a design of cohesive units of work is developed by using coarse-grained services. It provides loose coupling between the service and client by helping to abstract service internals from client. The ability to encapsulate change is increased by loose coupling. If fine-grained services are ready, they can be wrapped in the facade layer for helping to achieve benefit of a coarse-grained service.

- **Minimize round trips by using batch**: To reduce the call latency, the round trips should be minimized. This principle should be applied to threads, processors, processes, or servers for reducing communication across boundaries. While making remote server calls, this principle can be particularly helpful.

- **Acquire late and release early**: The duration of limited and shared resources which are occupied should be minimized such as database connections and network. It can be expensive to release and again re-acquire the resources from the operating system, hence, the recycling plan can be considered for supporting 'acquire late and release early'. In this manner, the use of shared resources across requests is optimized.

- **Evaluate affinity with processing resources**: There is an affinity between the resource and processor or server when certain resources are available only from certain processors or servers. The affinity not only results in improved performance, but also can impacts the scalability. Thus, the scalability needs should be carefully evaluated. The application's ability to scale can be inhibited if application requests are bounded to a particular processor or server by affinity. The ability to distribute processing across the servers and processors influences the capacity of the application as an when the load on it increases.

- **Put the processing closer to the resources it needs**: If a lot of client-server interaction is involved in the processing, the processing code should be pushed closer to the client. If the processing interacts more with the data store, the processing should be closer to the data.

- **Pool shared resources**: Pool shares the resources which are expensive or scarce such as creating network connections or using database. The performance overhead can be eliminated by using pooling technique. In this, connection to resource is established by sharing limited resources with a large number of clients for improving scalability.

- **Avoid unnecessary work**: To reduce unnecessary processing, use techniques such as validating input early, avoiding round trips, and caching.

- **Reduce contention**: The common sources of contention are hotspots and blocking. Blocking may be caused due to long running tasks such as expensive I/O operations. Concentration of access to certain data which is needed by everyone results in hotspots. While accessing resources, blocking should be avoided because requests are queued due to resource contention. In a database scenario, large tables must be indexed properly to avoid blocking due to Read/Write operations. However, different parts of the table can be accessed by many clients without any difficulty. While the small tables may be used frequently by the clients. Minimizing the amount of time code retains locks and efficient use of shared threads are the techniques which can be used to reduce contention.

  Note - Contention is competition for resources. The term is used in networks especially for describing the situation where two or more nodes attempt to transfer a message at the same time and across the same wire.

- **Use progressive processing**: Data changes must be handled with efficient practices. When a portion of data changes, all the data should not be processed, only process the changed data. Also, rendering output progressively should be considered. Entire result should be unblocked when partial result can be given to the user.

- **Process independent tasks concurrently**: Multiple tasks that are to be processed which are independent of each other can be executed asynchronously to complete them concurrently. The I/O bound tasks are benefited by asynchronous processes, but when the tasks are CPU-bounded; they have limited benefits and are restricted to a single-processor. Additional threads can be used for context switching if a single CPU server is used and there would be limited gains as it is not real multithreading. However, due to the overhead of thread switching the single CPU-bound multithreaded tasks are performed relatively slowly.

6.2.1 Coupling and Cohesion

For increasing scalability of an application the two principles used are increasing cohesion and reducing coupling. Cohesion can be measured as the number of different components taking advantage of data and shared processing. The degree of dependency between different parts of a system is called Coupling. Thus, an application designed in modular fashion contains set of loosely coupled components that are highly cohesive.

The following are the recommendations to ensure appropriate degrees of coupling and cohesion in design:

- **Design for high cohesion**: Logically related entities should be grouped together like classes and methods. For example, logically related set of methods should be in a class. Weak cohesion among components may result in more number of round trips because the components or classes are not logically grouped and may be residing in different tiers of the architecture.

- **Design for loose coupling**: Try to reduce coupling within and across application components. If a user needs to make changes and has tight coupling, the changes are to be rippled across the tightly coupled components. Changes are limited in loosely coupled components. In addition, scalability for different components and greater flexibility for choosing optimized strategies for performance is provided independently by loose coupling in the system.

- **Partition application functionality into logical layers**: Ensures that presentation logic, data access logic, and business logic are separated using logical layers to partition application. This logical organization results in a cohesive design wherein the related data and classes are located close to each other, within a single boundary. The use of expensive resources can be optimized in this way.

- **Evaluate resource affinity**: Contrast and compare the advantages and disadvantages of resource affinity. In some scenarios, affinity to a particular resource can improve performance. However, affinity may satisfy performance goals for now, but scalability of the application is affected by resource affinity.

- **If possible use early binding**: Early binding minimizes runtime overhead, so prefer it wherever possible. It is the most efficient way to call a method.

6.3 Web Application Testing Tools

Web applications have become popular since the year 2000 on the Internet due to the interactive experience provided by it to the users. Earlier, static HTML Web pages could only be viewed and users were not able to interact with them for creating personal accounts, querying database, add content, or perform a transaction. A Web application provides an interactive experience but for doing so, it needs to frequently store, use, and collect sensitive personal data for delivering services. Customers use these applications for their convenience and also, take risk by providing sensitive information to the Web application which can be vulnerable.

Web Application Security Consortium

The Web Application Security Consortium (WASC) is an international organization dedicated to the refinement, promotion, and establishment of Internet security standards. The consortium was founded in January 2004 and it includes independent members and also, those associated with government agencies, corporations, and academic institutions.

The WASC is authorized to research, publish, and discuss information about Web application security issues. The organization thus educates enterprises and individuals about such issues and the preventive measures against specific threats. Although members of WASC may belong to corporations which are involved in the development, research, design, and distribution of Web security related products but it is vendor-neutral.

Following is the list of products and tools for scanning the Web applications:

- NetSparker by Mavituna Security
- NTOSpider by NTObjectives
- WebInspect by HP
- WebKing by Parasoft
- Acunetix WVS by Acunetix
- Burp Suite Professional by PortSwigger
- N-Stalker by N-Stalker
- Nessus by Tenable Network Security
- ParosPro by MileSCAN Technologies
- NeXpose by Rapid7
- Retina Web Security Scanner by eEye Digital Security
- WebApp360 by nCircle
- Websecurify by GNUCITIZEN

6.3.1 Web Application Scanner

A program which communicates with a Web application via the Web front-end to detect the potential security threats and vulnerabilities in the Web application architecture used to implement it is termed as Web application security scanner or Web application scanner. The scanner performs a black box test on the Web application and detects vulnerabilities by actually attacking the Web application.

Following is the list of open source tools available:

- Arachni by Tasos Laskos
- Grendel-Scan by David Byrne and Eric Duprey
- Grabber by Romain Gaucher
- W3AF by Andres Riancho
- Wapiti by Nicolas Surribas

- Paros by Chinotec
- Skipfish by Michal Zalewski
- Watcher by Casaba Security

6.3.2 Vulnerability Scanners

The automated tool which scans Web application and looks for known security vulnerabilities such as SQL injection, cross-site scripting, and so on is called as Vulnerability Scanner. Many open source and commercial tools are available having their own strengths and weakness.

Table 6.1 detailed list of some scanning tools.

Name	License	Owner	Platforms
Hailstorm	Commercial	Cenzic	Windows
Windows	Commercial/Free (Limited Capability)	Contrast Security (Aspect)	SaaS
AppScan	Commercial	IBM	Windows
Vega	Open Source	Subgraph	Windows, Linux, and Macintosh
SOATest	Commercial	Parasoft	Windows, Linux, and Solaris
WebScanService	Commercial	German Web Security	N/A
Zed Attack Proxy	Open Source	OWASP	Windows, Unix/Linux, and Macintosh

Table 6.1: Scanning Tools

Figure 6.1 shows the Zed Attack proxy tool used for capturing session started in a browser.

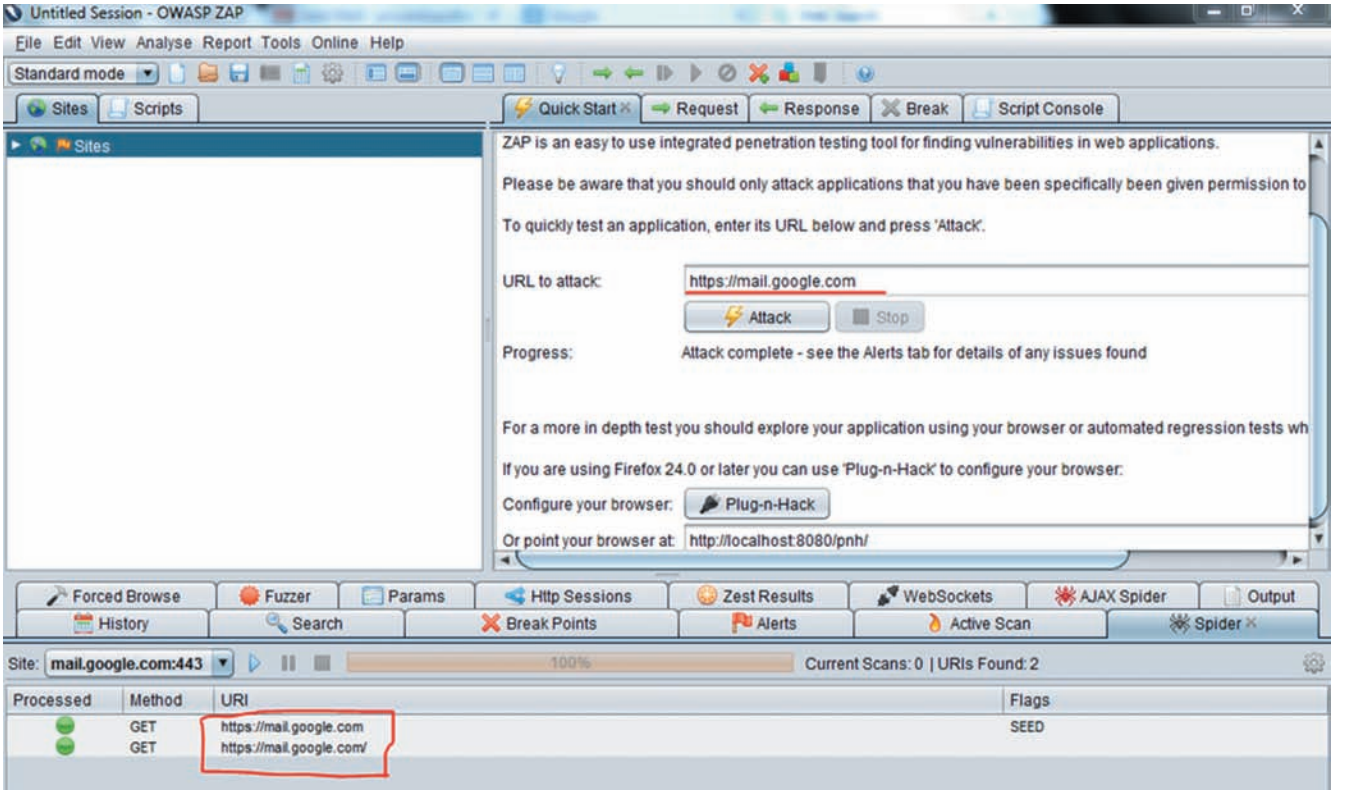


Figure 6.1: Zed Attack Proxy Tool

6.3.3 Fiddler

A proxy server application for HTTP debugging developed by Microsoft team is called Fiddler. Fiddler captures and logs HTTPS and HTTP traffic for the user to review. It is also used to modify HTTP traffic as it is received or sent for troubleshooting purposes. Traffic from Microsoft's WinInet HTTP(S) stack is directed automatically to the proxy server by default at runtime, but any Web application or browser can be configured to route its traffic via Fiddler.

Following are the key features of Fiddler:

➔ **HTTP/HTTPS Traffic Recording**

Fiddler is a free HTTP debugging proxy server tool which captures traffic between the Internet and computer. It can be used to capture and debug traffic from any application which supports proxy such as Chrome, Safari, IE, Firefox, and so on.

➔ **Web Session Manipulation**

Fiddler can be used to easily edit and manipulate Web sessions. Only a breakpoint is to be set to pause the processing of the session and alteration of request/response should be permitted. Any HTTP request can run via Fiddler and the contents of the request such as header, type encoding, and so on can be observed.

➔ **Web Debugging**

Fiddler can be used to debug traffic from Windows, Linux, or MAC system and mobile devices. It can also be used to ensure proper header, cookie, and cache directives are transferred between server and client. It supports different frameworks such as Java, .Net, and so on.

➔ **Security Testing**

Fiddler can also be used for security testing of Web applications such as to modifying or displaying the requests, decrypting HTTPs traffic using a man-in-middle technique, and so on. Fiddler can be configured to decrypt all the traffic or only a specific session as per requirement.

➔ **Performance Testing**

Fiddler can be used for tracking the page weight, also HTTP compression and caching can be glanced. The performance issues can be bottlenecked by using rules such as 'Flag any uncompressed responses which are larger than 25 kb'.

6.3.4 Customizing Fiddler

Ranging from simple Fiddler script to powerful extensions can be developed using .Net language. Also, benefit is gained from a rich extensibility model.

Using Fiddler

Following are steps for using Fiddler:

1. Download fiddler from **www.fiddler2.com** and install it. Figure 6.2 shows the screen which can be seen when fiddler is opened.

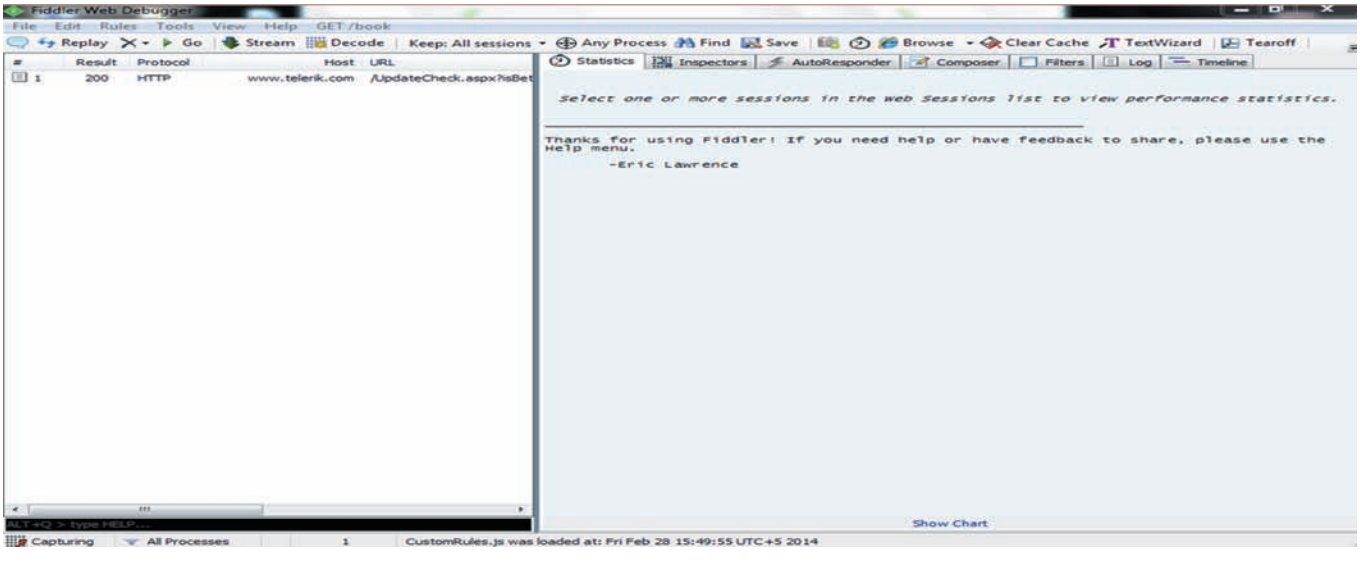


Figure 6.2: Fiddler Home Screen

Figure 6.3 shows the fiddler capturing the sessions running in the browser. HTTP sessions are been logged.

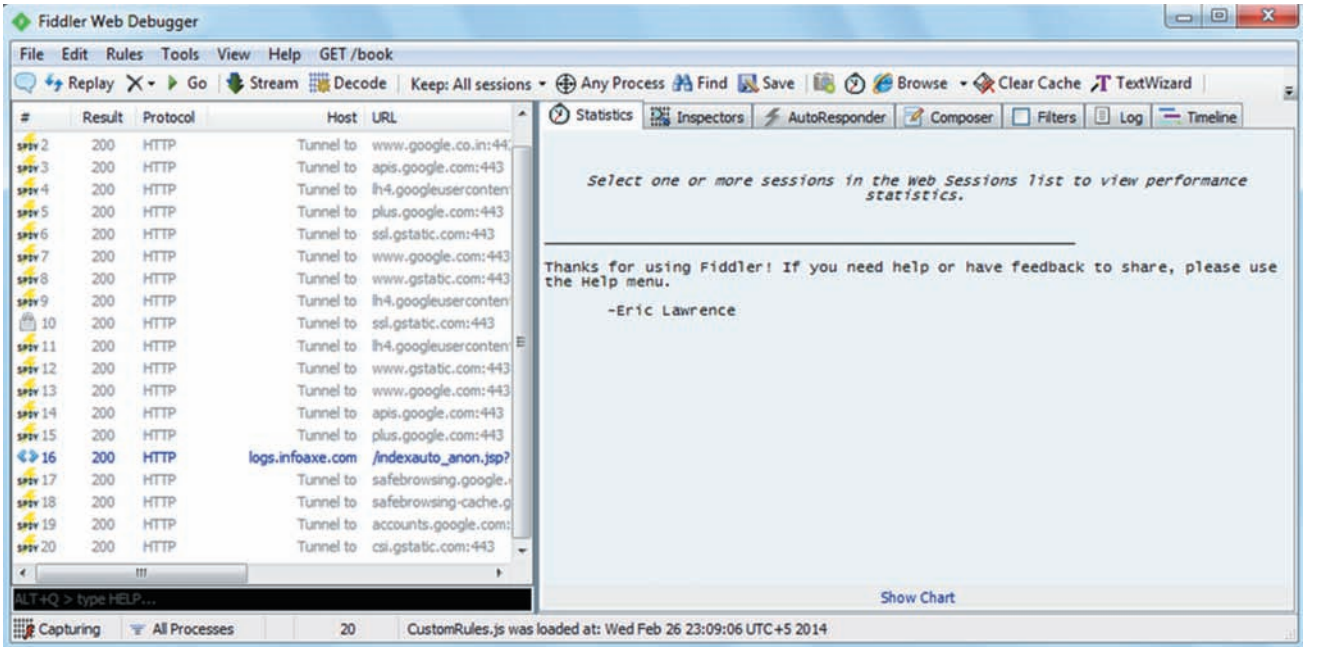


Figure 6.3: Fiddler Capturing Session Information

2. Turn on HTTPS options in fiddler by clicking **Tools** → **Fiddler Options** and select **HTTPS** as shown in figure 6.4 to log the HTTPs sessions.

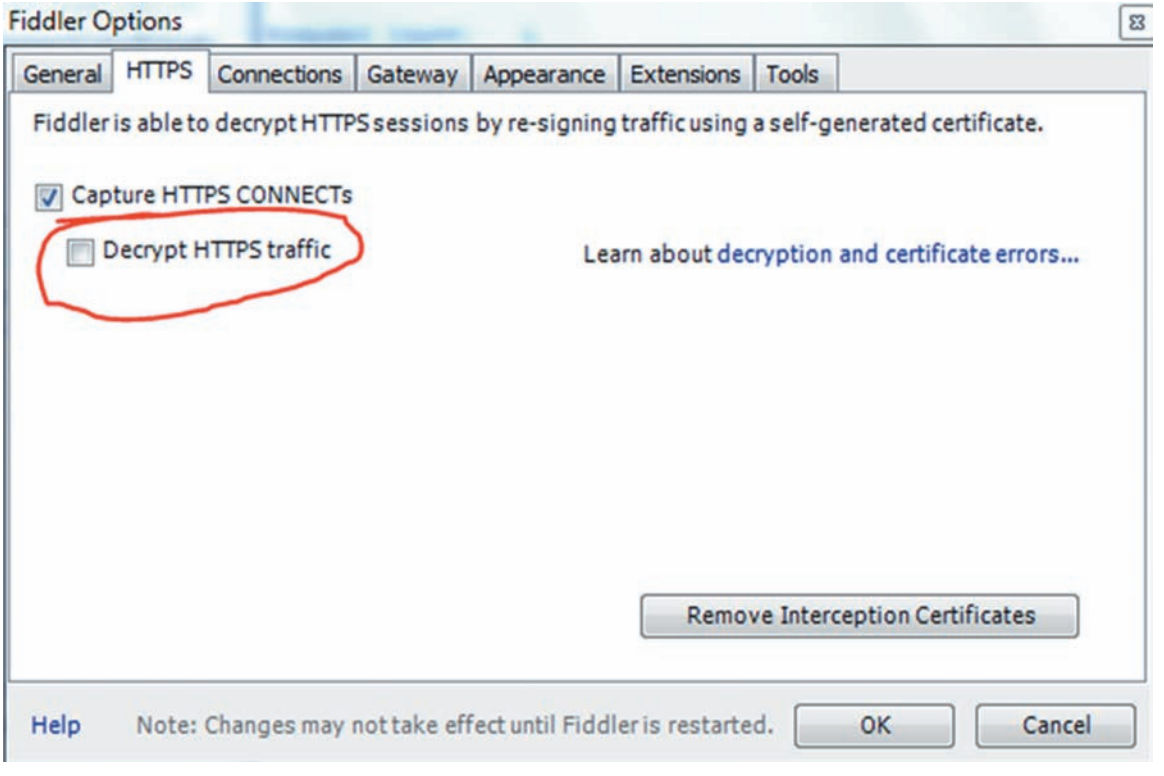


Figure 6.4: Fiddler Options Dialog Box

Concepts

3. Check the checkbox '**Decrypt HTTPs traffic**' to capture the HTTPs traffic.

4. Now, start HTTPS sessions from the browser.

5. Select the sessions which are to be recorded as Web Test as shown in figure 6.5.

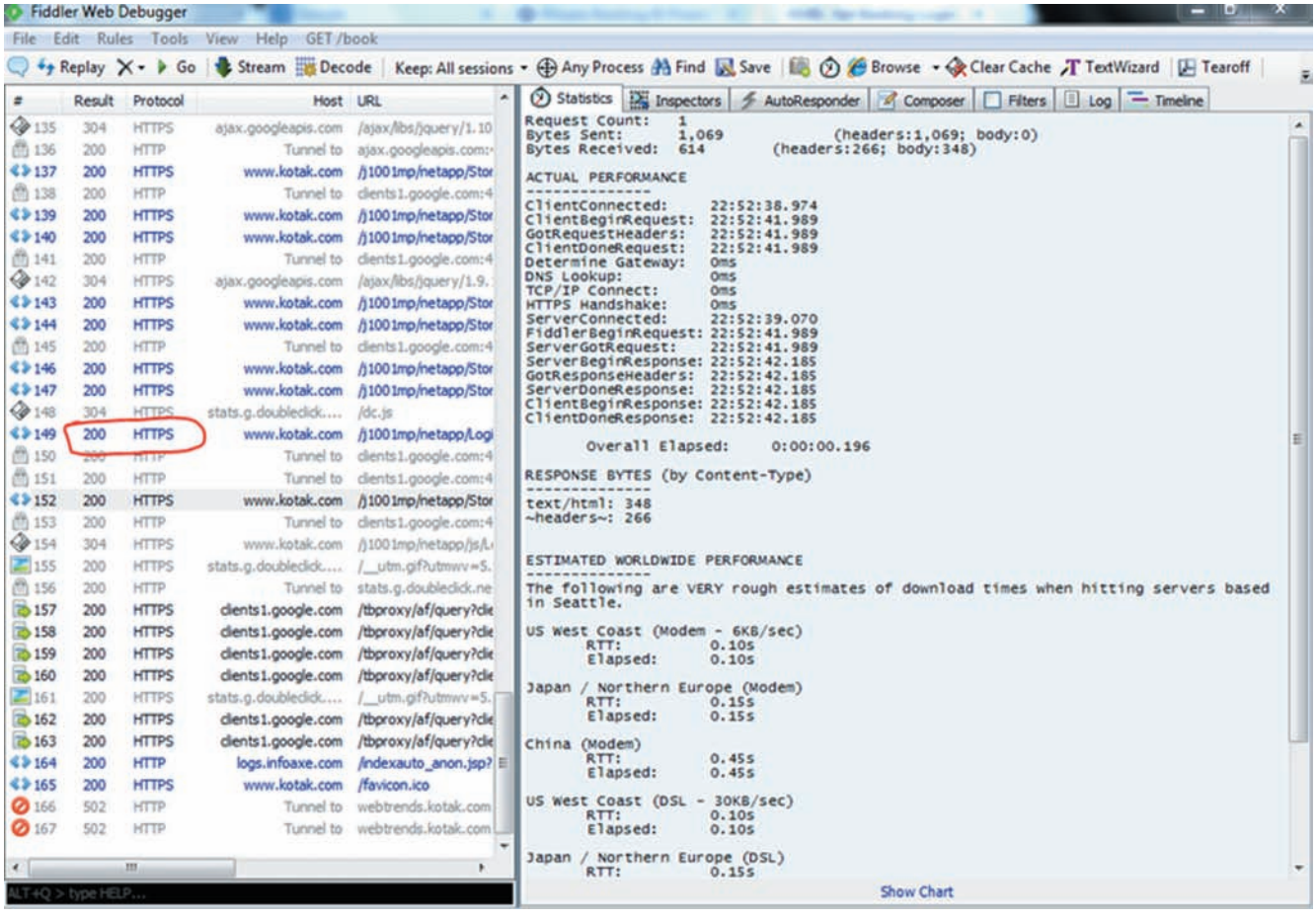


Figure 6.5: Selecting Sessions to be Tested

6. Once the sessions needed are selected, the details about the session can be seen in the right pane and the tabs can be selected accordingly as shown in figure 6.6.

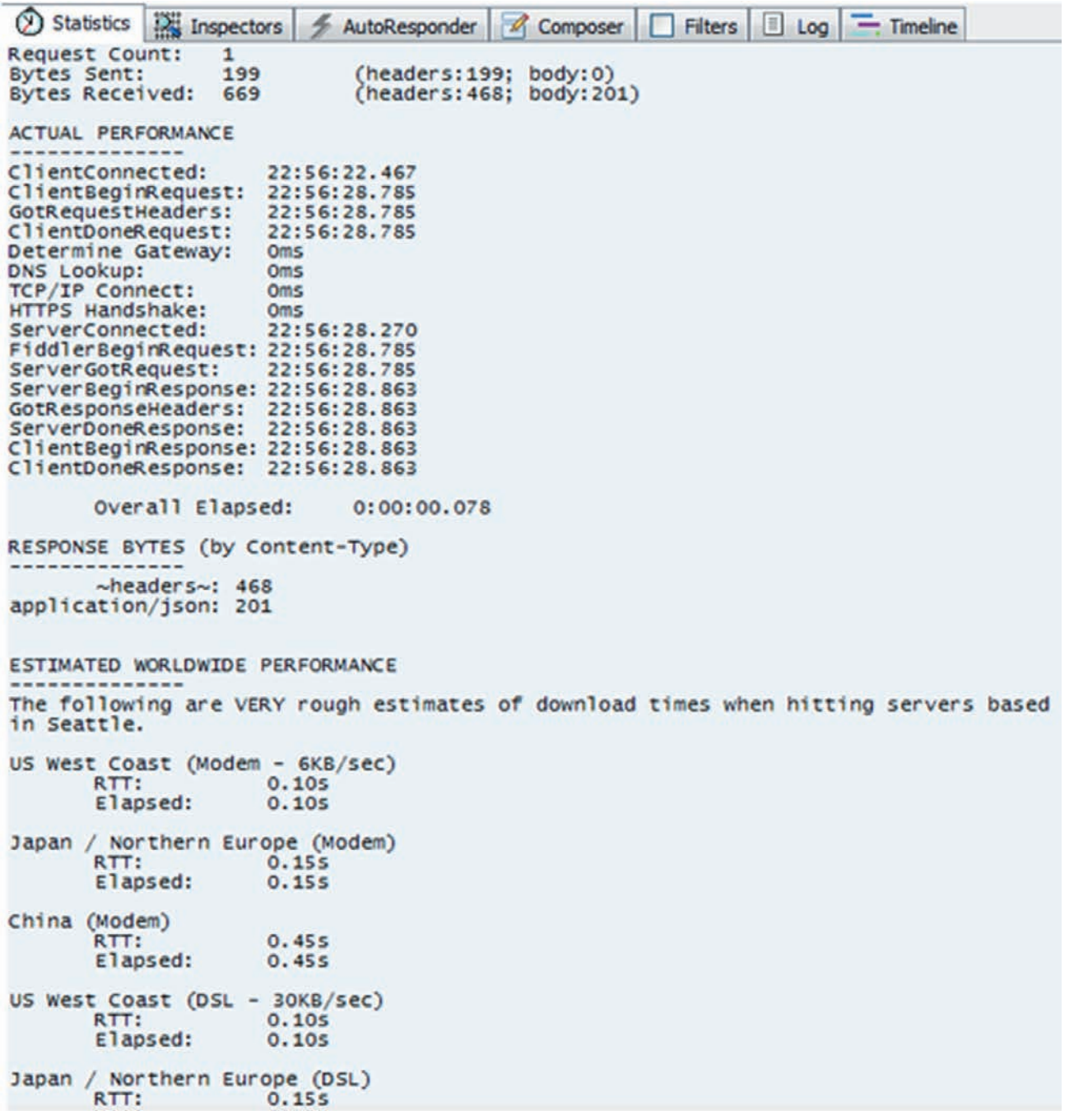


Figure 6.6: Showing Site Information

Concepts

6.3.5 Other Tools for Monitoring Web Application Security

Following are some additional tools for monitoring Web application security:

- **mon.itor.us**: It is a free monitoring tool with lots of useful features which help to maintain high uptime. It provides tons of information about the Website and Web server which can help to spot potential issues which may result in vulnerability. Mon.itor.us has the ability for sending downtime alerts through text message, RSS, and email. It also has an intuitive dashboard GUI, real-time visitor monitoring, and also capability to monitor from multiple geographical locations. It is an easy-to-use tool boasting a low setup time of only about five minutes.

- **Montastic**: Montastic is a quick, free, and simple-to-use tool to keep constant knowledge of the Website's availability. It has been developed by Metadot. It also sends warnings whenever a site crashes via email, Mac, and Windows widget or RSS. It allows monitoring up to 100 sites per account, has an elegant and simple end-user interface, and also supports monitoring for HTTPS and HTTP connections.

- **ServerMojo**: ServerMojo is an easy-to-use service to supervise Web server's uptime. The unavailability of the site is alerted via email, twitter, and IM. It permits monitoring of one site at an interval of one hour.

- **HostTracker**: HostTracker is a free Web tool to monitor site availability. Up to two Websites can be monitored at a time to receive weekly, monthly, quarterly, and yearly reports of the Web server's performance for free. It could help to track useful data on the site's availability for diagnostics, distributed monitoring, and also send alerts of the problems through IM, email, or SMS. HostTracker has a nice utility widget to check a Website's availability instantly just by entering the URL and it will ping the server.

- **InternetSeer**: InternetSeer offers a free standard service of 60 minutes interval to monitor the Website performance and uptime. It sends site availability and reports of page response time, real-time error notifications, and also weekly report on server's performance for diagnostics.

There are many other tools such as FreeSiteStatus, SiteUptime, Basic State, and so on that can be used to monitor site performance and detect application vulnerabilities.

6.4 Check Your Progress

1. At _____, the session related information should be destroyed by the server.

(A)	login	(C)	Authentication
(B)	logout	(D)	None of these

2. Match the following Fiddler features with the corresponding description.

	Fiddler Features		Description
a.	Web Session Manipulation	1.	Fiddler can be used for tracking the page weight and also HTTP compression and caching can be glanced.
b.	Security Testing	2.	Any HTTP request can run via Fiddler and the contents of the request such as header, type encoding, and so on can be seen.
c.	Web Debugging	3.	Fiddler can be used to modify or display the requests and decrypt HTTPs traffic using a man-in-middle technique.
d.	Performance Testing	4.	It can also be used to ensure that proper header, cookie, and cache directives are transferred between server and client.

(A)	a-2, b-4, c-1, d-3	(C)	a-3, b-4, c-1, d-2
(B)	a-4, b-1, c-2, d-3	(D)	a-2, b-3, c-4, d-1

3. Which of the following statements about Web application scanner are true?

a.	It is a program which communicates with a Web application via the Web front -end to detect the potential security threats and vulnerabilities in the Web application architecture.
b.	A black box test is performed by it.

(A)	Statement b	(C)	Statements a and b
(B)	Statement a	(D)	None of these

4. A proxy server application for HTTP debugging developed by Microsoft team is called _____.

(A)	Fiddler	(C)	Vega
(B)	InternetSeer	(D)	Appscan

Concepts

5. Which of the following are design principles and measures used during Web application development?

(A)	Reduce contention	(C)	Process independent tasks concurrently
(B)	Avoid unnecessary work	(D)	All of these

6. _____ can be measured as the number of different components taking advantage of data and shared processing.

(A)	Cohesion	(C)	Pool sharing
(B)	Coupling	(D)	None of these

6.4.1 Answers

1.	B
2.	D
3.	C
4.	A
5.	D
6.	A

Summary

- The client-server interactions are minimized by cohesive design and thus, units of work is developed by using coarse-grained services.
- Pools share the resources which are expensive or scarce like creating network connections or using database.
- For increasing scalability of application the two principles used are increasing cohesion and reducing coupling.
- A program which communicates with a Web application via the Web front -end to detect the potential security threats and vulnerabilities in Web application architecture used implement it is termed as Web application security scanner.
- The automated tool which scans a Web application and looks for known security vulnerabilities such as SQL injection, cross-site scripting, and so on is called as vulnerability scanner.
- A proxy server application for HTTP debugging developed by Microsoft team is called Fiddler.
- The other monitoring tools for Web application security are mon.itor.us, Montastic, ServerMojo, and so on.

Onlinevarsity
your e-way to learning

To enhance your knowledge,

visit the **REFERENCES** *page*

@

www.onlinevarsity.com