UNIT 1 WEB SECURITY CONCEPTS

Structure **Page Nos.**

1.0 Introduction 5
1.1 Objectives 5
1.2 Web Services and its Advantages 6
1.3 Web Security Concepts 8
1.3.1 Integrity
1.3.2 Confidentiality
1.3.3 Availability
1.3.4 SSL/TSL
1.4 HTTP Authentication 12
1.4.1 HTTP Basic Authentication
1.4.2 HTTP Digest Authentication
1.4.3 Form Based Authentication
1.4.4 HTTPS Client Authentication
1.5 Summary 15
1.6 Solutions/Answers 16
1.7 Further Readings/References 16

1.0 INTRODUCTION

Web Security may be defined as technological and managerial procedures applied to computer systems to ensure the availability, integrity, and confidentiality of information. It means that protection of integrity, availability and confidentiality of computer assets and services from associated threats and vulnerabilities.

The security of the web is divided into two categories (a) computer security, and (b) network security. In generic terms, computer security is the process of securing a single, standalone computer; while network security is the process of securing an entire network of computers.

(a) **Computer Security:** Technology and managerial procedures applied to computer systems to ensure the availability, integrity, and confidentiality of the data managed by the computer.

(b) **Network Security:** Protection of networks and their services from unauthorised modification destruction, or disclosure and provision of assurance that the network performs its critical functions correctly and that are nor harmful side effects.

The major points of weakness in a computer system are hardware, software, and data. However, other components of the computer system may be targeted. In this unit, we will focus on web security related topics.

1.1 OBJECTIVES

After going through this unit, you should be able to :

- achieve integrity, confidentiality and availability of information on the internet integrity and confidentiality can also be enforced on web services through the use of SSL(Secure Socket Layer); and

- ensure secure communication on the Internet for as e-mail, Internet faxing, and other data transfers.

1.2 WEB SERVICES AND ITS ADVANTAGES

According to the W3C a Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface that is described in a machine-processable format such as WSDL. Other systems interact with the Web service in a manner prescribed by its interface using messages, which may be enclosed in a SOAP envelope. These messages are typically conveyed using HTTP, and normally comprise XML in conjunction with other Web-related standards. Software applications written in various programming languages and running on various platforms, can use web services to exchange data over computer networks like, the Internet, in a manner similar to inter-process communication on a single computer. This interoperability (i.e. between Java and Python, or Microsoft Windows and Linux applications) is due to the use of open standards.

Web services and its advantages

- Web services provide interoperability between various software applications running on disparate platforms/operating systems.

- Web services use open standards and protocols. Protocols and data formats are text-based where possible, which makes it easy for developers to undestand.

- By utilising HTTP, web services can work through many common firewall security measures without having to make changes to the firewall filtering rules.

- Web services allow software and services from different companies and locations to be combined easily to provide an integrated service.

- Web services allow the reuse of services and components within an infrastructure.

- Web services are loosely coupled thereby, facilitating a distributed approach to application integration.

So it is necessary to provide secure communication in web communication.

WS-Security

WS-Security (Web Services Security) is a communications protocol providing a means for applying security to Web Services Originally developed by IBM, Microsoft, and VeriSign, the protocol is now officially called WSS and developed via a committee in Oasis-Open.

The protocol contains specifications on how integrity and confidentiality can be enforced on Web Services messaging.

Three basic security concepts important to information on the Internet are confidentiality, integrity, and availability.

A Common Windows Security Problem

Unfortunately, many Microsoft Windows users are unaware of a common security leak in their network settings.

This is a common setup for network computers in Microsoft Windows:

- Client for Microsoft Networks
- File and Printer Sharing for Microsoft Networks
- NetBEUI Protocol
- Internet Protocol TCP/IP

If your setup allows NetBIOS over TCP/IP, you have a security problem:

- Your files can be shared all over the Internet
- Your logon-name, computer-name, and workgroup-name are visible to others.

If your setup allows File and Printer Sharing over TCP/IP, you also have a problem:

- Your files can be shared all over the Internet

Solving the Problem

For Windows 95/98/2000 users:

You can solve your security problem by disabling NetBIOS over TCP/IP:

You must also disable the TCP/IP Bindings to Client for Microsoft Networks and File and Printer Sharing.

If, you still want to share your Files and Printer over the network, you must use the NetBEUI protocol instead of the TCP/IP protocol. Make sure you have enabled it for your local network.

🗇 Check Your Progress 1

1) Compare and Contrast Computer and Network Security.

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

2) Explain IP protocol Suit.

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

3) What is web security?Explain with suitable examples.

……………………………………………………………..…………………………...

……………………………………………………………………………………

……………………………………………………………………………………

1.3 WEB SECURITY CONCEPTS

In this section, we will describe briefly four concepts related to web security.

Three basic security concepts important to information on the Internet are confidentiality, integrity, and availability**.** Concepts relating to the people who use that information are authentication, authorisation, and nonrepudiation. Integrity and confidentiality can also be enforced on Web Services through the use of Transport Layer Security (TLS). Both SSL and TSL (Transport Layer Security) are the same. The dependencies among these concepts (also called objects) is shown in *Figure 1*.

1.3.1 Integrity

Integrity has two facets:

Data Integrity: This property, that data has not been altered in an unauthorised manner while in storage, during processing or while in transit. Another aspect of data integrity is the assurance that data can only be accessed and altered by those authorised to do so. Often such integrity is ensured by use of a number referred to as a Message Integrity Code or Message Authentication Code. These are abbreviated as MIC and MAC respectively.

System Integrity: This quality that a system has when performing the intended function in an unimpaired manner, free from unauthorised manipulation.
Integrity is commonly an organisations most important security objective, after availability. Integrity is particularly important for critical safety and financial data used for activities such as electronic funds transfers, air traffic control, and financial accounting.

1.3.2 Confidentiality

Confidentiality is the requirement that private or confidential information should not to be disclosed to unauthorised individuals. Confidentiality protection applies to data in storage, during processing, and while in transit.

For many organisations, confidentiality is frequently behind availability and integrity in terms of importance. For some types of information, confidentiality is a very important attribute. Examples include research data, medical and insurance records, new product specifications, and corporate investment strategies. In some locations, there may be a legal obligation to protect the privacy of individuals.

This is particularly true for banks and loan companies; debt collectors; businesses that extend credit to their customers or issue credit cards; hospitals, doctors' offices, and medical testing laboratories; individuals or agencies that offer services such as psychological counseling or drug treatment; and agencies that collect taxes.

1.3.3 Availability

Availability is a requirement intended to assure that systems work promptly and service is not denied to authorised users. This objective protects against:

- Intentional or accidental attempts to either:
  - perform unauthorised deletion of data or
  - otherwise cause a denial of service or data.

- Attempts to use system or data for unauthorised purposes.

Availability is frequently an organisations foremost security objective. To make information available to those who need it and who can be trusted with it, organisations use authentication and authorisation.

Authentication

Authentication is proving that a user is whom s/he claims to be. That proof may involve something the user knows (such as a password), something the user has (such as a "smartcard"), or something about the user that proves the person's identity (such as a fingerprint).

Authorisation

Authorisation is the act of determining whether a particular user (or computer system) has the right to carry out a certain activity, such as reading a file or running a program. Authentication and authorisation go hand in hand. Users must be authenticated before carrying out the activity they are authorised to perform.

Accountability (to be individual level)

Accountability is the requirement that actions of an entity may be traced uniquely to that entity. Accountability is often an organisational policy requirement and directly supports repudiation, deterrence, fault isolation, intrusion detection and prevention, and after action recovery and legal action.

Assurance (that the other four objectives have been adequately met)

Assurance is the basis for confidence that the security measures, both technical and operational, work as intended to protect the system and the information it processes. Assurance is essential, without it other objectives cannot be met.

The above mentioned security objects, dependencies are shown in *Figure 1*.

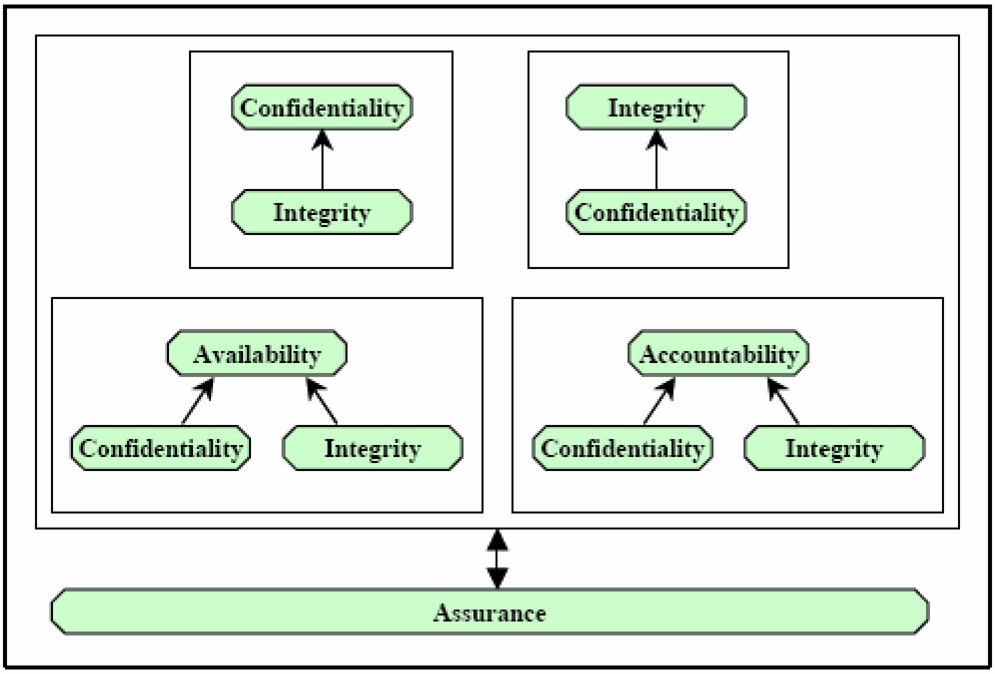

Figure 1: Security Objects Dependencies

🗇 Check Your Progress 2

1) List the basic security concepts.

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

2) What do you understand by information assurance?

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

3) Compare and contrast data integrity and system integrity.

……………………………………………………………………………………

……………………………………………………………………………………

……………………………………………………………………………………

1.3.4 Secure Socket Layer (SSL)/Transport Layer Security(TLS)

Secure Socket Layer (SSL) and Transport Layer Security (TLS), its successor, are cryptographic protocols which provide secure communication on the Internet for as e-mail, internet faxing, and other data transfers.

Description

SSL provides endpoint authentication and communication privacy over the Internet using cryptography. In typical use, only the server is authenticated (i.e. its identity is ensured) while the client remains unauthenticated; mutual authentication requires public key infrastructure (PKI) deployment to clients. The protocols allow client/server applications to communicate in a way designed to prevent eavesdropping, tampering, and message forgery.

Tampering may relate to:

- **Tampering (Sports):** The practice, often illegal, of professional sports teams negotiating with athletes of other teams.
- **Tamper-evident:** A device or process that makes unauthorised access to a protected object easily detected.
- **Tamper proofing:** A methodology used to hinder, deter or detect unauthorized access to a device or circumvention of a security system.

Message Forgery

In cryptography, message forgery is the sending of a message to deceive the recipient of whom the real sender is. A common example is sending a spam e-mail from an address belonging to someone else

SSL involves three basic phases:

1) Peer negotiation for algorithm support,

2) Public key encryption-based key exchange and certificate-based authentication, and

3) Symmetric cipher-based traffic encryption.

During the first phase, the client and server negotiation uses cryptographic algorithms. Current implementations support the following choices:

- For public-key cryptography: RSA, Diffie-Hellman, DSA or Fortezza;
- For symmetric ciphers: RC2, RC4, IDEA, DES, Triple DES or AES;
- For one-way hash functions: MD5 or SHA.

SSL working

The SSL protocol exchanges records. Each record can be optionally compressed, encrypted and packed with a Message Authentication Code (MAC). Each record has a "content type" field that specifies which upper level protocol is being used.
When the connection begins, the record level encapsulates another protocol, the handshake protocol. The client then sends and receives several handshake structures:

- It sends a *ClientHello* message specifying the list of cipher suites, compression methods and the highest protocol version it supports. It also sends random bytes which will be used later.

- Then it receives a *ServerHello*, in which the server chooses the connection parameters from the choices offered by the client earlier.

- When the connection parameters are known, client and server exchange certificates (depending on the selected public key cipher). These certificates are currently X.509, but there is also a draft specifying the use of OpenPGP based certificates.

- The server can request a certificate from the client, so that the connection can be mutually authenticated.

- Client and server negotiate a common secret called "aster secret", possibly using the result of a Diffie-Hellman exchange, or simply encrypting a secret with a public key that is decrypted with the peer's private key. All other key data is derived from this "master secret" (and the client- and server-generated random values), which is passed through a carefully designed "Pseudo Random Function".

TLS/SSL have a variety of security measures:

- Numbering all the records and using the sequence number in the MACs.

- Using a message digest enhanced with a key.

- Protection against several known attacks (including man in the middle attacks), like those involving a downgrade of the protocol to previous (less secure) versions, or weaker cipher suites.

- The message that ends the handshake ("Finished") sends a hash of all the exchanged data seen by both parties.

- The pseudo random function splits the input data in 2 halves and processes them with different hashing algorithms (MD5 and SHA), then XORs them together. This way it protects itself in the event that one of these algorithms is found to be vulnerable.

Public key cryptography is a form of cryptography which generally allows users to communicate securely without having prior access to a shared secret key. This is done by using a pair of cryptographic keys, designated as public key and private key, which are related mathematically.

The term asymmetric key cryptography is a synonym for public key cryptography though a somewhat misleading one. There are asymmetric key encryption algorithms that do not have the public key-private key property noted above. For these algorithms, both keys must be kept secret, that is both are private keys.

In public key cryptography, the private key is kept secret, while the public key may be widely distributed. In a sense, one key "locks" a lock; while the other is required to unlock it. It should not be possible to deduce the private key of a pair, given the public key, and in high quality algorithms no such technique is known.

There are many forms of public key cryptography, including:

- *Public key encryption* keeping a message secret from anyone that does not possess a specific private key.
- *Public key digital signature* allowing anyone to verify that a message was created with a specific private key.
- *Key agreement* generally, allowing two parties that may not initially share a secret key to agree on one.

1.4 HTTP AUTHENTICATION

A web client can authenticate a user to a web server using one of the following mechanisms:

- HTTP Basic Authentication
- HTTP Digest Authentication
- Form Based Authentication
- HTTPS Client Authentication

1.4.1 HTTP Basic Authentication

HTTP Basic Authentication, which is based on a username and password, is the authentication mechanism defined in the HTTP/1.0 specification. A web server requests a web client to authenticate the user. As a part of the request, the web server passes the realm (a string) in which the user is to be authenticated. The realm string of Basic

Authentication does not have to reflect any particular security policy domain (confusingly also referred to as a realm). The web client obtains the username and the password from the user and transmits them to the web server. The web server then authenticates the user in the specified realm.

Basic Authentication is not a secure authentication as user passwords are sent in simple base64 ENCODING (not ENCRYPTED !), and there is no provision for target server authentication. Additional protection mechanism can be applied to mitigate these concerns: a secure transport mechanism (HTTPS), or security at the network level (such as the IPSEC protocol or VPN strategies) can be deployed. This is shown in the following role-based authentication.

```
<web-app>
        <security-constraint>
                <web-resource-collection>
                        <web-resource-name>User Auth</web-resource-
name>
                        <url-pattern>/auth/*</url-pattern>
                </web-resource-collection>
                <auth-constraint>
                        <role-name>admin</role-name>
                        <role-name>manager</role-name>
                </auth-constraint>
        </security-constraint>

        <login-config>
                <auth-method>BASIC</auth-method>
                <realm-name>User Auth</realm-name>
        </login-config>

        <security-role>
                <role-name>admin</role-name>
        </security-role>
        <security-role>
                <role-name>manager</role-name>
        </security-role>
</web-app>
```

1.4.2 HTTP Digest Authentication

Similar to HTTP Basic Authentication, HTTP Digest Authentication authenticates a user based on a username and a password. However, the authentication is performed by transmitting the password in an ENCRYPTED form, which is much MORE SECURE than the simple base64 encoding used by Basic Authentication, e.g., HTTPS Client Authentication. As Digest Authentication is not currently in widespread use, servlet containers are encouraged but NOT REQUIRED to support it.

The advantage of this method is that the cleartext password is protected in transmission, it cannot be determined from the digest that is submitted by the client to the server. Digested password authentication supports the concept of digesting user passwords. This causes the stored version of the passwords to be encoded in a form that is not easily reversible, but that the web server can still utilise for authentication.

From a user perspective, digest authentication acts almost identically to basic authentication in that it triggers a login dialogue.

The difference between basic and digest authentication is that on the network connection between the browser and the server, the password are encrypted, even on a non-SSL connection. In the server, the password can be stored in clear text or encrypted text, which is true for all login methods, and is independent of the choice that the application deployer makes.

Digested password is authentication based on the concept of hash or digest. In this stored version, the passwords is encoded in a form that is not easily reversible and this is used for authentication. Digest authentication acts almost identically to basic authentication in that it triggers a login dialogue. The difference between basic and digest authentication is that on the network connection between the browser and the server, the password is encrypted, even on a non-SSL connection. In the server, the password can be stored in clear text or encrypted text, which is true for all login methods and is independent of the application deployment.

1.4.3 Form Based Authentication

The look and feel of the 'login screen' cannot be varied using the web browser's built-in authentication mechanisms. This form based authentication mechanism allows a developer to CONTROL the look and feel of the login screens.

The web application deployment descriptor, contains entries for a login form and error page. The login form must contain fields for entering a username and a password. These fields must be named j_username and j_password, respectively.

When a user attempts to access a protected web resource, the container checks the user's authentication. If the user is authenticated and possesses authority to access the resource, the requested web resource is activated and a reference to it is returned.
If the user is not authenticated, all of the following steps occur:

1) The login form associated with the security constraint is sent to the client and the URL path triggering the authentication stored by the container.

2) The user is asked to fill out the form, including the username and password fields.

3) The client posts the form back to the server.

4) The container attempts to authenticate the user using the information from the form.

5) If authentication fails, the error page is returned using either a forward or a redirect, and the status code of the response is set to 200.

6) If authentication succeeds, the authenticated user's principal is checked to see if it is in an authorised role for accessing the resource.

7) If the user is authorised, the client is redirected to the resource using the stored URL path.

The error page sent to a user that is not authenticated contains information about the failure.

Form Based Authentication has the same lack of security as Basic Authentication since the user password is transmitted as a plain text and the target server is not authenticated. Again additional protection can alleviate some of these concerns: a secure transport mechanism (HTTPS), or security at the network level (such as the IPSEC protocol or VPN strategies) are applied in some deployment scenarios.

Form based login and URL based session tracking can be problematic to implement. Form based login should be used only when, sessions are being maintained by cookies or by SSL session information.

1.4.4 HTTPS Client Authentication

End user authentication using HTTPS (HTTP over SSL) is a strong authentication mechanism. This mechanism requires the user to possess a Public Key Certificate (PKC). Currently, PKCs are useful in e-commerce applications and also for a single-sign-on from within the browser. Servlet containers that are not J2EE technology compliant are not required to support the HTTPS protocol.

Client-certificate authentication is a more secure method of authentication than either BASIC or FORM authentication. It uses HTTP over SSL, in which the server and, optionally, the client authenticate one another with Public Key Certificates. Secure Sockets Layer (SSL) provides data encryption, server authentication, message integrity, and optional client authentication for a TCP/IP connection. You can think of a public key certificate as the digital equivalent of a passport. It is issued by a trusted organisation, which is known as a certificate authority (CA), and provides identification for the bearer. If, you specify client-certificate authentication, the Web server will authenticate the client using the client's X.509 certificate, a public key certificate that conforms to a standard that is defined by X.509 Public Key Infrastructure (PKI). Prior to running an application that uses SSL, you must configure SSL support on the server and set up the public key certificate.

Check Your Progress 3

1) Compare and contrast the authentication types (BASIC, DIGEST, FORM, and CLIENT-CERT); describe how the type works; and given a scenario, select an appropriate type.

…………..………………………………………………………………..

……………………..……………………………………………………..

……………………………..………………………………………………..

1.5 SUMMARY

To Achieve integrity, confidentiality and availability of Information on the internet is the goal of web security integrity. Confidentiality can also be enforced on web services through the use of SSL. Integrity is The property that data has not been altered in an unauthorised manner while in storage, during processing or while in transit. Confidentiality is the requirement that private or confidential information is not to be disclosed to unauthorised individuals. Confidentiality protection applies to data in storage, during processing, and while in transit. Availability is a requirement intended to assure that systems work promptly and that service is not denied to authorised users.

1.6 SOLUTIONS/ANSWERS

Check Your Progress 1

1) Computer Security: Technology and managerial procedures applied to computer systems to ensure the availability, integrity, and confidentiality of the data managed by the computer. Whereas, Network Security is protection of networks and their services from unauthorized modification destruction, or disclosure and provision of assurance that the network performs its critical functions correctly and there are not harmful side effects.

2) IP protocol suit: The different types of protocols used in different layers are Physical, data link, network, transport, session, presentation and applications layer.

3) Web Security can be defined as technological and managerial procedures applied to computer systems to ensure the availability, integrity, and confidentiality of the information. It means that protection of Integrity, Availability & confidentiality of computer assets and services from associated threats and vulnerabilities. Explain with suitable example HTTPS, SSL, IPSec etc.

Check Your Progress 2

1) The basic security concepts are:
Integrity, authenticity, confidentiality, authorisation, availability, and assurance.

2) Information assurance is the basis for confidence that the security measures, both technical and operational, work as intended to protect the system and the information it processes.

3) **Data Integrity:** The property that data has not been altered in an unauthorised manner while in storage, during processing or while in transit.

   System Integrity: The quality that a system has, when, performing the intended function in an unimpaired manner, free from unauthorised manipulation.

Check Your Progress 3

1) Hint: Compare and Authentication type with suitable example in different scenario depending upon the application type finance, stock exchange, simple message exchange between two persons, remote logging, client server authentication etc.

1.7 FURTHER READINGS/REFERENCES

- Stalling William, *Cryptography and Network Security, Principles and Practice,* 2000, SE, PE.
- Daview D. and Price W., *Security for Computer Networks,* New York:Wiley, 1989.
- Chalie Kaufman, Radia Perlman, Mike Speciner, *Network Security,* Pearson Education.
- B. Schnier, *Applied Cryptography*, John Wiley and Sons

- Steve Burnett & Stephen Paine, *RSA Security's Official Guide to Practice*, SE, PE.
- Dieter Gollmann, *Computer Security*, John Wiley & Sons.

Reference websites:

- *World Wide Web Security FAQ:* http://www.w3.org/Security/Faq/www-security-faq.html
- *Web Security*: http://www.w3schools.com/site/site_security.asp
- *Authentication Authorisation and Access Control:* http://httpd.apache.org/docs/1.3/howto/auth.html
- *Basic Authentication Scheme:* http://en.wikipedia.org/wiki/Basic_authentication_scheme
- *OpenSSL Project:* http:/www.openssl.org
- *Request for Comments 2617 :* http://www.ietf.org/rfc/rfc2617.txt
- *Sun Microsystems Enterprise JavaBeans Specification:* http://java.sun.com/products/ejb/docs.html.
- *Javabeans Program Listings*: http:/e-docs.bea.com