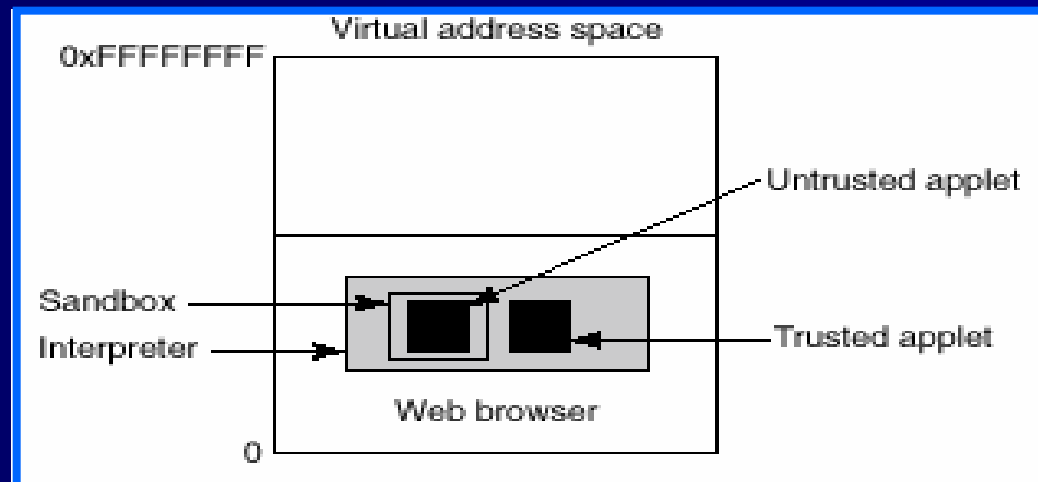


5.1. Java Applet

- **Java Applets** are small Java programs compiled to JVM (Java Virtual Machine).
- Once downloaded, applets are inserted into a JVM interpreter inside the browser
- **Interpretation is more favourable than compilation**
 - Checking addresses in instruction
 - System calls are treated in accordance with a security policy.
- If applet is not trusted then it's confined within a **Sandbox**



ActiveX control

- **ActiveX** is a Microsoft technology similar to Java that allows viewing of Windows files via an Internet Browser
- **Interacts with the operating system and executed without a sandbox.**
 - it can cause security problems within the system
- **for protection: need to decide whether you want to run ActiveX or not.**
 - The method that Microsoft chose for making this decision is based on the idea of **code signing**
 - Each ActiveX control is accompanied by a digital signature
 - Browser verifies the signature prior to execution.

JavaScript

- JavaScript does not have formal security model
- Each vendor handles security in a different way
 - Sandbox to run JavaScript code: code can only perform web-related actions, not general-purpose programming tasks like creating files.
 - JavaScript scripts are constrained by the same origin policy: scripts from one web site do not have access to information such as usernames, passwords, or cookies sent to another site.

6. Web application Security

- 1. SQL injection
- 1. Common Gateway Interface

SQL injection

- SQL (Structured Query Language) is a language that Communicates with DBs, Example:
 - *Select * from Users where username ='admin' and password = 'somepasswd'*
 - Looks for user whose username = admin and password = somepasswd
- SQL injection is a technique to inject crafted SQL into user input fields that are a part of web forms, can be used to:
 - bypass custom login to a web site,
 - Log in to a web site, or
 - take over a site

SQL injection: Simple login bypassing

- Consider the following web site's login form:

```
...  
<form action = "login.asp" method = "post">  
<p> Username:<input type=text name= "username" /> </p>  
<p> Password:<input type=password name= "password" />  
  </p>  
<p> <input type=submit name= "submit" value="login" />  
  </p>  
</form>  
...
```

- It's a web page that requests 2 pieces of information from the user username and password and it submits the information in the fields to login.asp (written in asp)

SQL injection: Simple login bypassing

- The file login.asp:

```
Dim adoConnection
Set
    adoConnection=server.CreateObject("ADODB.Connection")
    on")

...
Dim strLoginSQL
strLoginSQL="select * from users where username = '
    & Request.Form("username") & " 'and password ='
    " & Request.Form("password") & " ' "
Dim adoResult
Set adoResult=adoConnection.Execute(strLoginSQL)
If not adoResult.EOF Then
    'We are here all went ok
Else
    'Wrong login
End If
```

SQL injection: Simple login bypassing

- If the user enters `admin` as a username and `adminpasswd`, the following sql command is constructed:

```
Select * from users where username ='admin' and  
password = 'adminpasswd'
```

- The username and password are placed inside the SQL string, but without any checks:
 - What happens if an attacker enter `'a'` or `"1"="1"` as a username and any password?
 - The resulting SQL string is:

```
Select * from users where username = 'a' or  
"1"="1" -- ' and password = 'anypassword'
```
 - This code will return data because `"1"="1"`
 - the attacker bypass the login.

SQL injection

■ Worse!

- The attacker can use built-in procedures to read or write files, or to invoke programs in the database computer
- For example the `xp_cmdshell` stored procedure invokes shell commands on the server's computer like `dir`, `copy`, `rename`, etc.
- From the last example, a hacker can enter some username as a username and a `'exec master..xp_cmdshell 'del c:\winnt\system32*.dll'` as a password.
 - This will cause the database to delete all DLLs in the specified directory.

SQL injection: Solutions

- Filter all input fields for apostrophes to prevent unauthorized logins
- Filter all input fields for SQL commands like `insert`, `select`, `delete`, and `exec` to prevent server manipulation
- Limit input field length (which will limit hackers' options), and validate the input length with server-side scripts.
- Place the database on a different computer than the web server.
 - If the database is hacked, it'll be harder to reach the web server.
- Limit the user privileges of the server-side scripts.
- Delete all unneeded extended stored procedures to limit hackers' possibilities.

Common Gateway Interface

- Common Gateway Interface (CGI)
 - meta-language for translating URLs or HTML forms into executable programs.
- An attacker may exploit bugs in CGI scripts to gain unauthorized access to files on the web server, or even to take control of the host.
- CGI scripts can present security holes in two ways:
 - they may intentionally or unintentionally **leak information** about the host system that will help hackers break in.
 - Scripts that process user input may be vulnerable to attacks in which the remote user tricks them into executing commands (always remember: “user input is evil”).

7. Communication Security

■ Vulnerabilities

- Tapping or eavesdropping: occurs when a device is placed near or into the cabling.
- Sniffing: using Sniffers (special programs) in order to eavesdrop on the network traffic.
- IP spoofing:
 - An attacker can place any IP address as the source address of an IP datagram, so can be dangerous to base access control decisions on raw IP addresses alone.

7. Communication Security

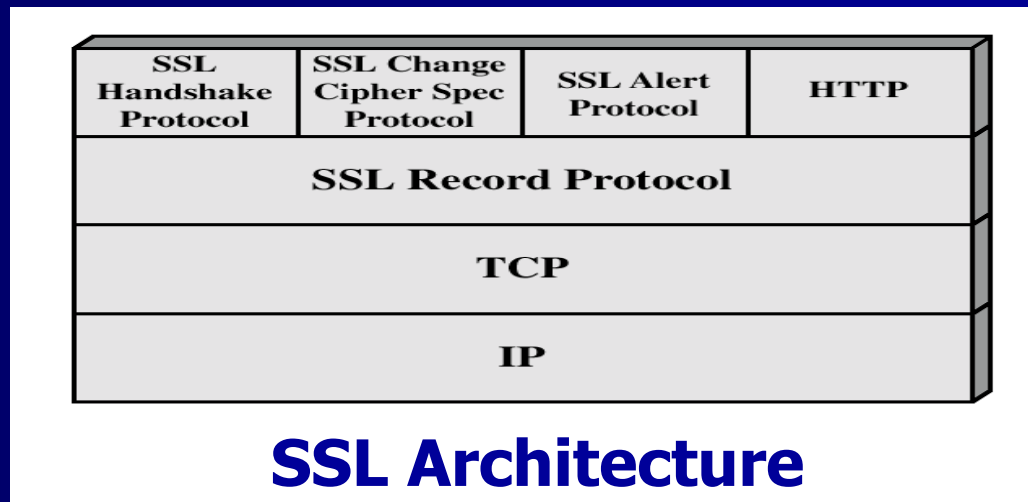
- An attacker may be able to replay, delay, reorder, modify or inject IP datagrams.
- DNS spoofing: DNS server is lured to translate names (eg, www.scs-net.org) into attackers' IP addresses.
- Solution:
 - Communication Protection: SSL, IPsec, ...

SSL

- **Secure Sockets Layer (SSL)** was developed (in 1994) by Netscape Corporation to provide security between web client and server.
- SSL designed to be under HTTP:
 - HTTP | SSL | TCP
- **SSL permits:**
 - Authentication of peer entities
 - **Exchange of secret keys**
 - Use of exchanged keys to authenticate and encrypt transmitted data between communicating peer entities.

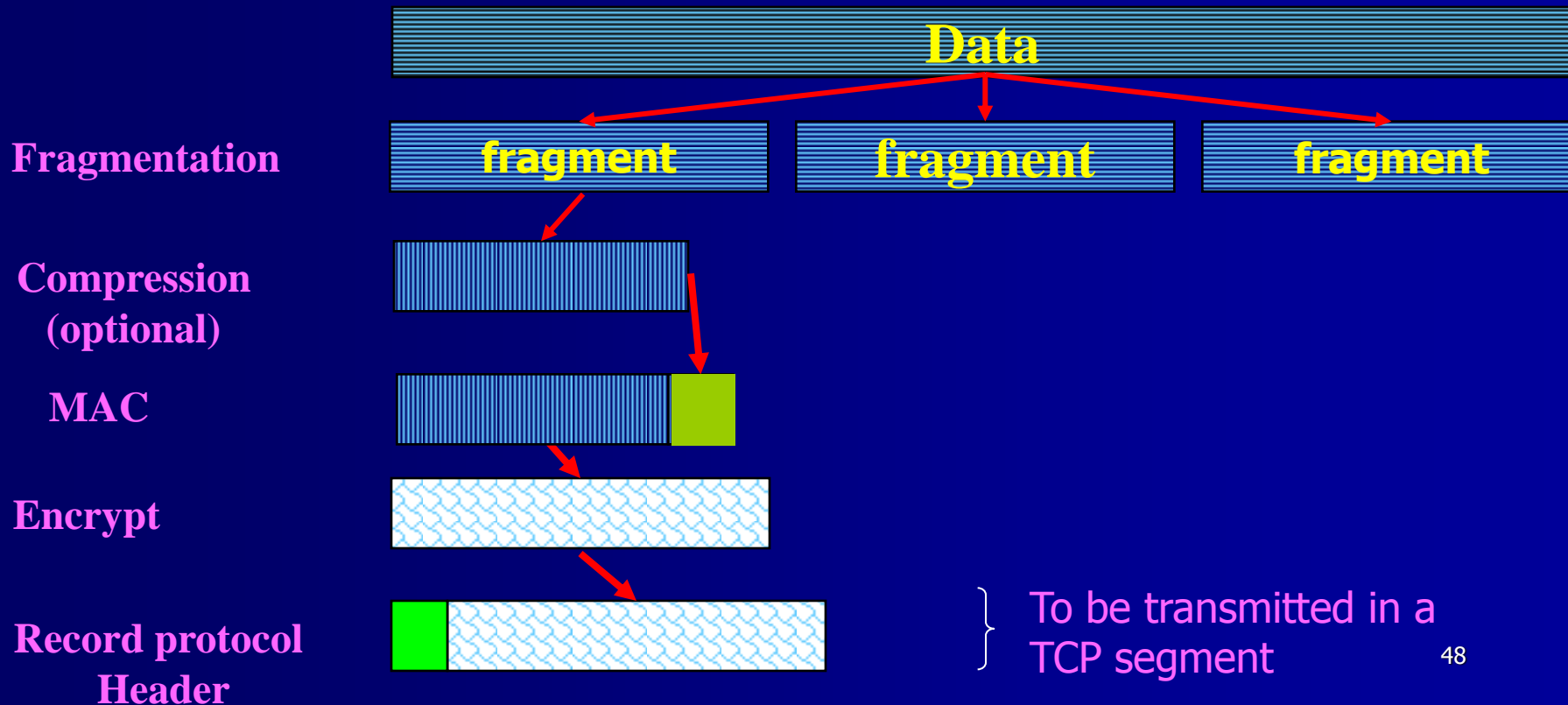
SSL Architecture

- **SSL consists of two sublayers:**
 - **SSL Record Protocol:** provide security services to higher-layer protocols (in particular, HTTP) including SSL management protocols.
 - **SSL Management protocols:** Handshake, Cipher Change, and Alert Protocols



SSL Record Protocol

- The SSL Record Protocol uses the keys derived from the Handshake Protocol's master key to securely deliver data.
- Provides two security functions:
 - Confidentiality and Message Integrity



SSL Record Protocol

- Protected data : SSL Record protocol allows application protocols above SSL to be secured.
- Fragmentation: messages are broken into blocks
- Compression: optional
 - Compression algorithm is not specified
- MAC: computed over compressed data.
 - SSL MAC is similar to HMAC
 - MAC key is derived from the master key.
- Encryption may be stream or block mode.
 - Symmetric encryption is used
 - There are only a limited selection of ciphers and MAC algorithms that are allowed (eg, DES, 3DES, IDEA, RC4, etc)

SSL Handshake Protocol

- Used to allow the server and client to
 - authenticate each other using certificates,
 - negotiate encryption and MAC algorithms, and
 - establish keys to be used to protect data sent in SSL Record.
- Used before any application data is transmitted.

JSSE

- **Java Secure Socket Extension (JSSE)** provides a framework and an implementation for a Java version of the SSL
- Includes functionality for:
 - data encryption,
 - server authentication,
 - message integrity,
 - and optional client authentication.
- JSSE, developers can provide for the secure passages of HTTP, FTP, Telnet, etc.

JSSE

- JSSE is included in `java.net.ssl` as a sub-package of `java.net`.
- JSSE main classes:
 - `Java.net.ssl.SSLSocket`
 - `Java.net.ssl.SSLSocketFactory`
 - `Java.net.ssl.SSLServerSocket`
 - `Java.net.ssl.SSLServerSocketFactory`
- `SocketFactory` and `ServerSocketFactory` are factory classes for generating classes of the corresponding socket classes.

8. ID and FW

- Viruses and intrusion are the most publicized threats to system security
- **Intrusion: illegally gaining access to systems**
- Intrusion techniques: acquiring protected information (often user passwords)
 - Passwords are associated with users in files
 - Password files must be protected
- Countermeasures: prevention and detection
 - If intrusion prevention fails,
 - Intrusion detection is the real defense line.

ID and FW

- Intrusion detection is based on the assumption that the behavior of the intruder differs from that of a legitimate user in ways that can be quantified.
- **Intrusion detection approaches:**
 - Statistical anomaly detection
 - Rule-based detection
- Audit Records: is a fundamental tool for intrusion detection
 - A detection record may contain subject (user, process), action (login, read, write), object (files, programs), resource usage, timestamp
- Examples of IDS:
 - Cisco's Secure IDS
 - ISS RealSecure
 - Snort

Firewall

- A firewall is any device used as a network-level access control mechanism for a particular network or a set of networks
 - Firewall is used to prevent outsiders from accessing an internal network.
- Firewalls may be stand-alone computers, routers, or firewall appliances (sometimes with their own OS)
- They serve as control points to and from networks
- They check whether or not network traffic should be allowed according to sets of rules or policies.
- Pitfalls: slowing data transmission, impairing networking

Types of firewalls

- Packet filtering routers
- Stateful-inspection firewalls
- Application-level gateway (also called proxy server)
- Examples:
 - CheckPoint's Firewall-1: Stateful-inspection-based
 - Cisco's PIX: stateful packet filter-based
 - Border's FireWall Server: Proxy-based
 - Tiny Software's Tiny Personal Firewall: Packet filter-based

Questions?