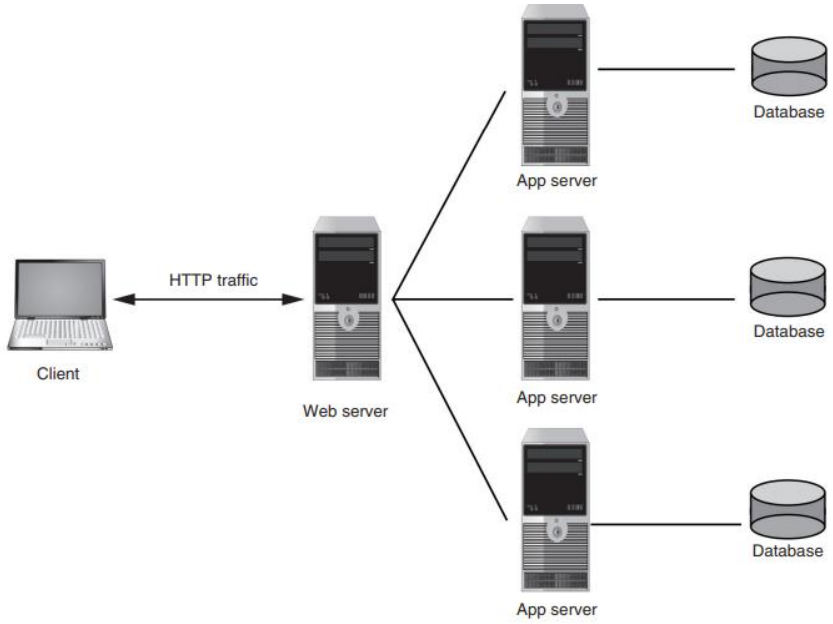
Application Attacks

- Server-Side Web Application Attacks
- Client-side Application Attacks

Server-Side Web Application Attacks

- On the Internet, a web server provides services that are implemented as web applications.
- An important characteristic of server-side web applications is that they create dynamic content based on inputs from the user.
- Many server-side web application attacks target the input that the applications accept from users



Server-Side Web Application Attacks

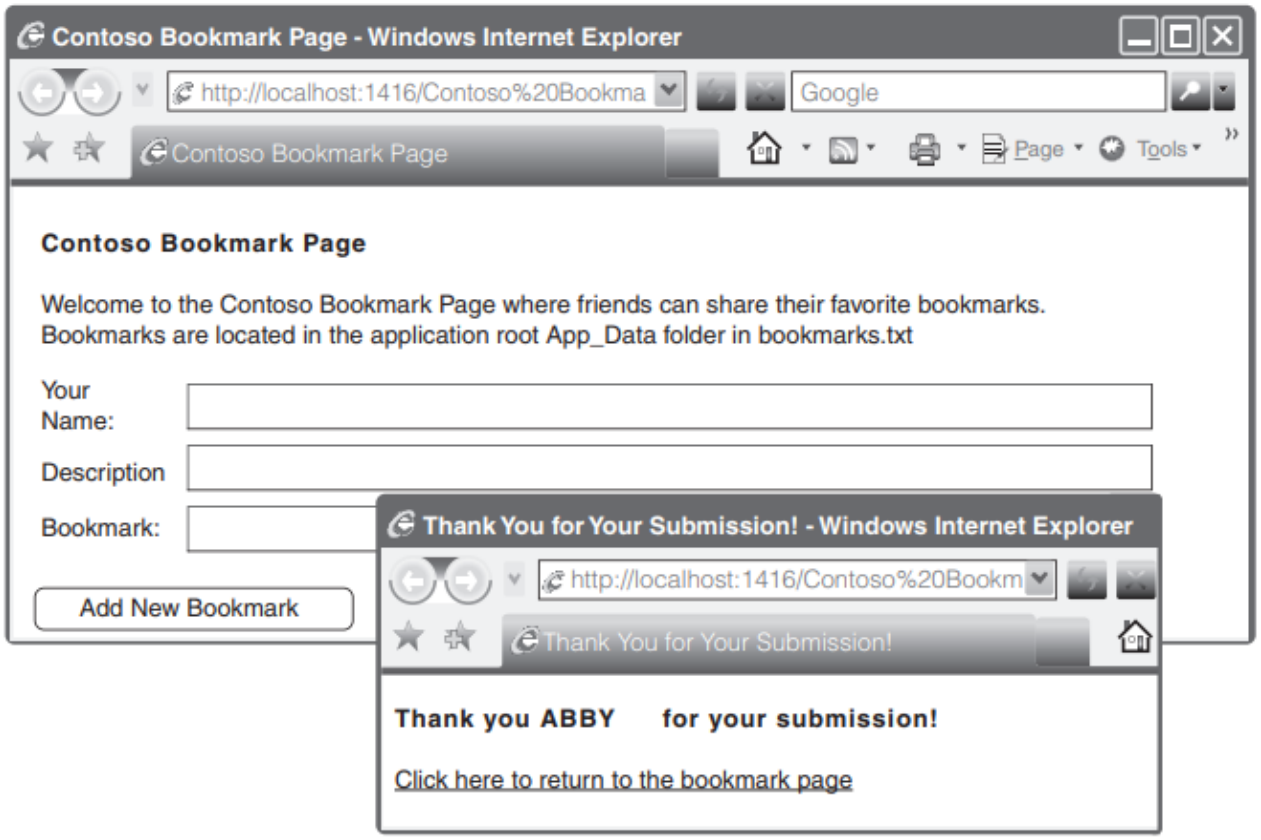
- Cross-Site Scripting (XSS)
- SQL Injection
- XML Injection
- Directory Traversal/Command Injection

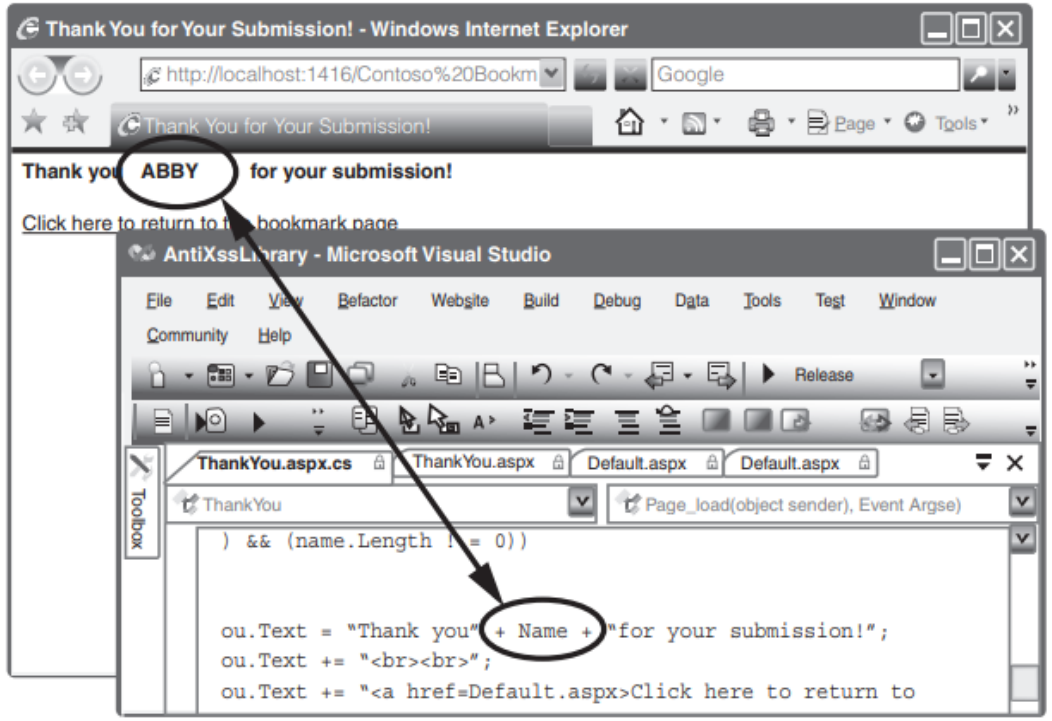
Cross-Site Scripting (XSS)

- XSS injects scripts into a web application server to direct attacks at unsuspecting clients.
- XSS attacks occur when an attacker takes advantage of web applications that accept user input without validating it and then present it back to the user.

User input	Variable that contains input	Web application response	Coding example
Search term	*search_term*	Search term provided in output	"Search results for *search_term*"
Incorrect input	*user_input*	Error message that contains incorrect input	"*user_input* is not valid"
User's name	*name*	Personalized response	"Welcome back *name*"

Cross-Site Scripting (XSS)



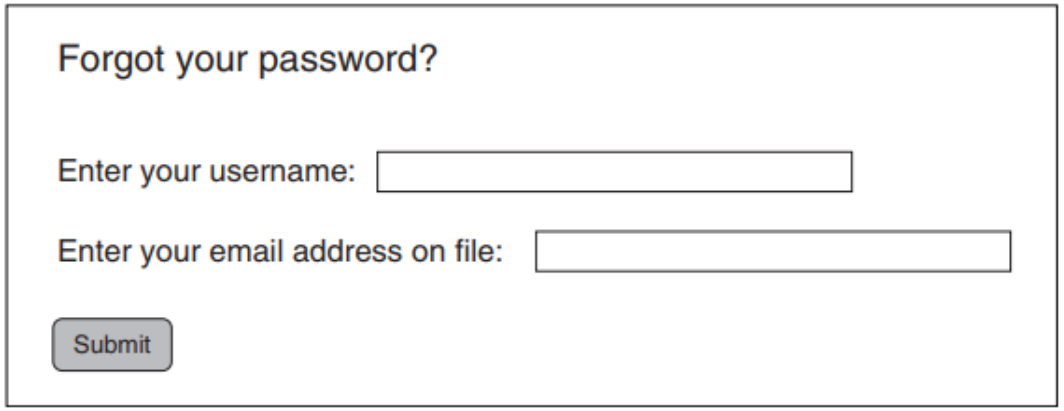


Cross-Site Scripting (XSS)

- An XSS attack requires a website that meets two criteria: it accepts user input without validating it, and it uses that input in a response.
- A typical XSS attack may take advantage of a blogger’s website that asks for user comments
- When an unsuspecting victim visits the blogger’s site and clicks on the attacker’s comment, the malicious script is downloaded to the victim’s web browser where it is executed

SQL Injection

- SQL stands for Structured Query Language, a language used to view and manipulate data that is stored in a relational database.
- SQL injection targets SQL servers by introducing malicious commands into them



SELECT fieldlist FROM table WHERE field = '$EMAIL'

SQL Injection Step 1

- An attacker using an SQL attack would begin by first entering a fictitious email address on this webpage that included a single quotation mark as part of the data, such as braden.thomas@fakemail.com’.
- If the message *E-mail Address Unknown* is displayed, it indicates that user input is being properly filtered.
- If the error message *Server Failure* is displayed, it means that the user input is not being filtered and all user input is sent directly to the database. This is because the Server Failure message is due to a syntax error created by the additional single quotation mark: the fictitious email address entered would be processed as *braden.thomas@fakemail.com’ ’* .

SQL Injection Step 2

- Attacker would enter this command, which would let him view all the email addresses in the database: ***'whatever' or 'a'='a***
- This command is stored in the variable *$EMAIL*
- The expanded SQL statement would read:

SELECT fieldlist FROM table WHERE field = 'whatever' or 'a'='a'

  - ***'whatever'***. This can be anything meaningless.
  - ***or***. The SQL or means that as long as either of the conditions are true, the entire statement is true and will be executed.
  - ***'a'='a'***. This is a statement that will always be true.

SQL injection statements

SQL injection statement	Result
whatever' AND email IS NULL; --	Determine the names of different fields in the database
whatever' AND 1=(SELECT COUNT(*) FROM tabname); --	Discover the name of the table
whatever' OR full_name LIKE '%Mia%'	Find specific users
whatever'; DROP TABLE members; --	Erase the database table
whatever'; UPDATE members SET email = 'attacker-email@evil.net' WHERE email = 'Mia@good.com';	Mail password to attacker's email account

XML Injection

- A markup language is a method for adding annotations to the text so that the additions can be distinguished from the text itself.
- XML is a markup language.
- XML is designed to carry data instead of indicating how to display it.
- XML does not have a predefined set of tags; instead, users define their own tags.

XML injection

```
<?xml version="1.0" encoding="utf-8"?>
<Employees>
<Employee ID="1000">
<FirstName>James</FirstName>
<LastName>Crockett</LastName>
<UserName>James_Crockett</UserName>
<Password>19mv85sb</Password>
<Type>Administrator</Type>
</Employee>
```

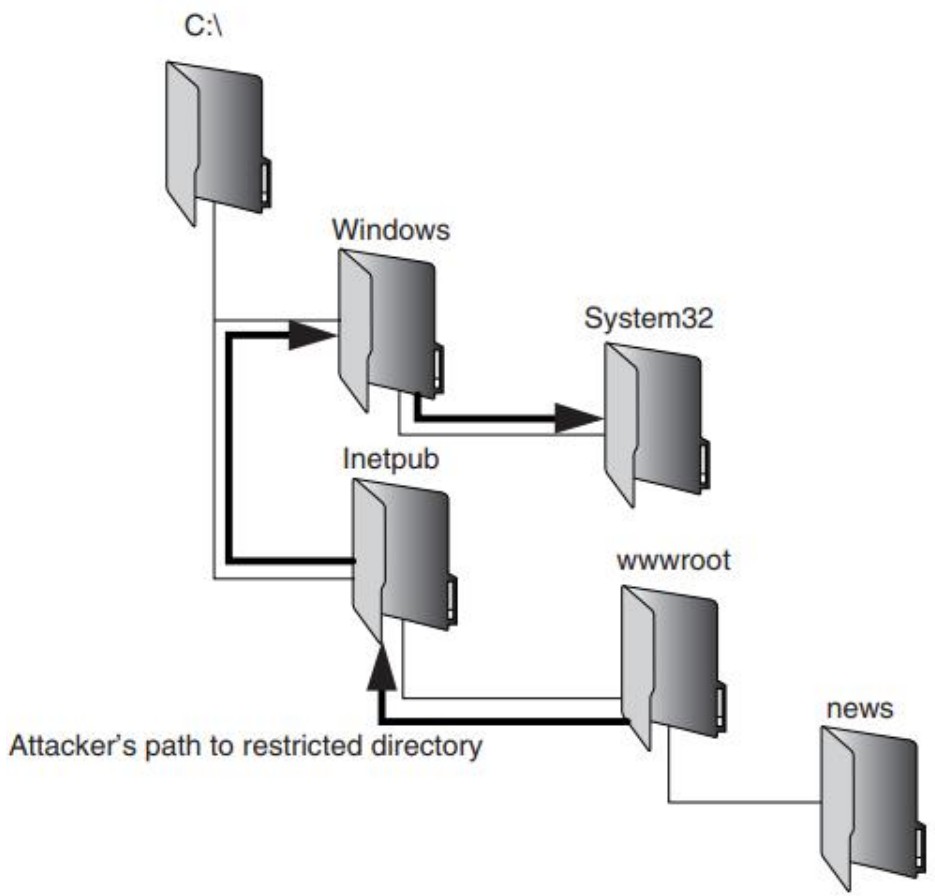
XML injection

- An XML injection attack is similar to an SQL injection attack.
- An attacker who discovers a website that does not filter input user data can inject XML tags and data into the database.
- A specific type of XML injection attack is an XPath injection, which attempts to exploit the XML Path Language (XPath) queries that are built from user input.

Directory Traversal/Command Injection

- The root directory is a specific directory on a web server's file system.
- For example, the default root directory of Microsoft's Internet Information Services (IIS) web server is *C:\Inetpub\wwwroot.*
- Users have access to this directory and subdirectories beneath this root (C:\Inetpub\wwwroot\news) if given permission, but do not have access to other directories in the file system, such as C:\Windows\System32.

Directory Traversal/Command Injection



Directory Traversal/Command Injection

- A directory traversal uses malformed input or takes advantage of a vulnerability to move from the root directory to restricted directories.
- Once the attacker has accessed a restricted directory, she can enter (inject) commands to execute on a server (called command injection) or view confidential files.

Directory Traversal/Command Injection

- For example, a browser requesting a compiled dynamic webpage (dynamic.asp) from a web server (www.server.net) to retrieve a file (display.html) in order to display it would generate the request using the URL:
  - *http://www.server.net/dynamic.asp?view=display.html*
- The attacker could create the input
  - *http://www.server.net/dynamic.asp?view=../../../../../TopSecret.docx*
- This could display the contents of a document

Client-side Application Attacks

- Client-side attacks target vulnerabilities in client applications that interact with a compromised server or process malicious data.
- One example of a client-side attack results in a user's computer becoming compromised just by viewing a webpage and not even clicking on any content.
- One commonly attack is *drive-by-download*

Client-side Application Attacks

- Header Manipulation
- Cookies
- Attachments
- Session Hijacking
- Malicious Add-ons
- Impartial Overflow Attacks

Header Manipulation

- The HTTP header consists of fields that contain information about the characteristics of the data being transmitted.
- An attacker can modify the HTTP headers to create an attack using HTTP header manipulation.
- HTTP header manipulation is not an actual attack, but rather the vehicle through which other attacks, such as XSS, can be launched

HTTP field name	Source	Explanation	Example
Server	Web server	Type of web server	*Server: Apache*
Referer or Referrer	Web browser	The address of the previous webpage from which a link to the currently requested page was followed	*Referer: http://www.askapache .com/show-error-502/*
Accept-Language	Web browser	Lists of acceptable languages for content	*Accept-Language:en-us,en;q=0.5*
Set-Cookie	Web server	Parameters for setting a cookie on the local computer	*Set-Cookie: UserID=ThomasTrain; Max-Age=3600; Version=1*

Cookies

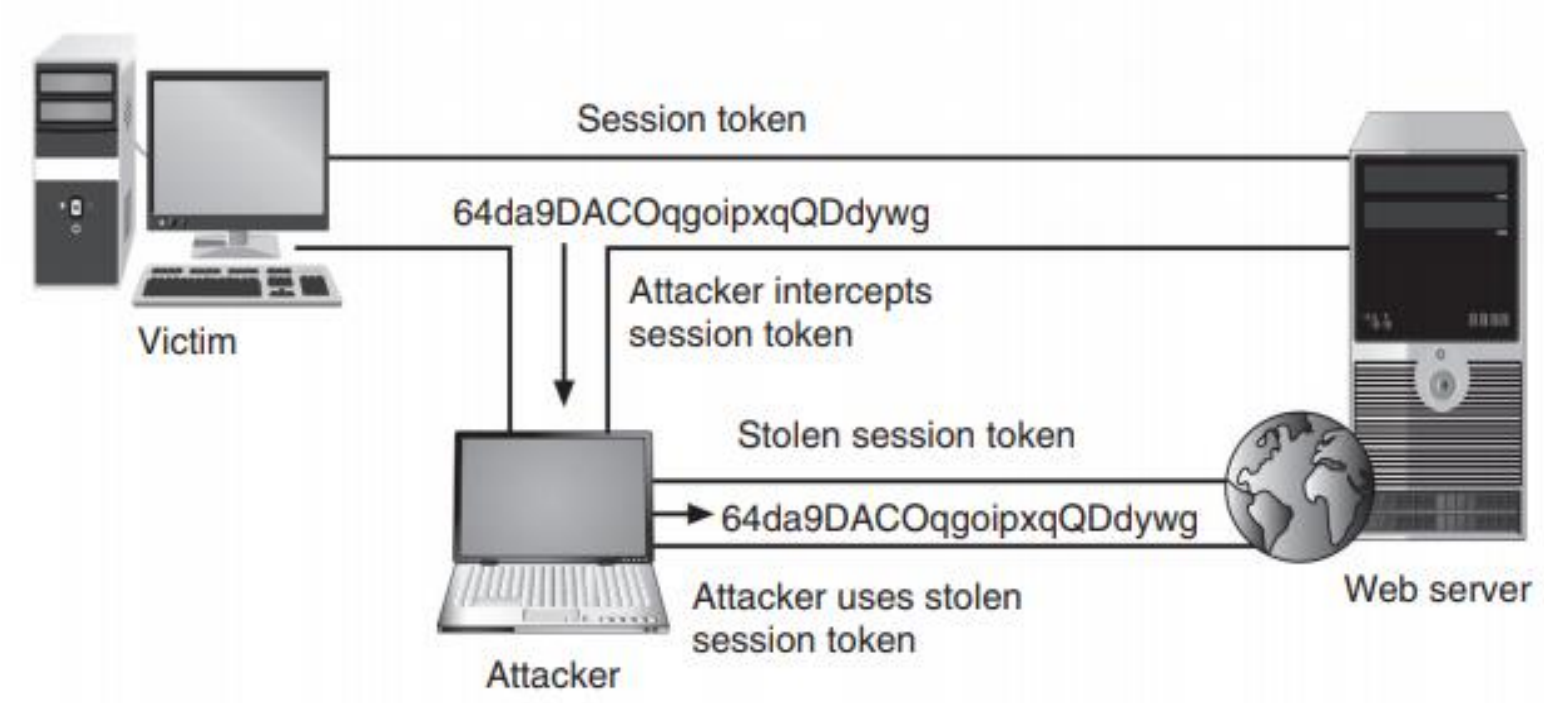
- A cookie can contain a variety of information based on the user’s preferences when visiting a website.
- Several different types of cookies exist: First-party cookie, Third-party cookie, Session cookie.
- First-party cookies can be stolen and used to impersonate the user.
- Third-party cookies can be used to track the browsing or buying habits of a user.

Attachments

- Attachments are files that are coupled to email messages.
- Malicious attachments are commonly used to spread viruses, Trojans, and other malware when they are opened

Session Hijacking

- Session hijacking is an attack in which an attacker attempts to impersonate the user by using her session token.



Malicious Add-ons

- Attackers can create malicious add-ons to launch attacks against the user's computer.
- One way in which these malicious add-ons can be written is by using Microsoft's ActiveX.
- Attackers can take advantage of vulnerabilities in ActiveX to perform malicious attacks on a computer.

Impartial Overflow Attacks

- Buffer Overflow Attack: A buffer overflow attack occurs when a process attempts to store data in RAM beyond the boundaries of a fixed-length storage buffer.
- Integer Overflow Attack: the condition that occurs when the result of an arithmetic operation—like addition or multiplication—exceeds the maximum size of the integer type used to store it.
- Arbitrary/Remote Code Execution: allows an attacker to run programs and execute commands on a different computer.