Security Strategies in Web Applications and Social Networking

Lesson 7

Introducing the Web Application Security Consortium (WASC)

Learning Objective and Key Concepts

Learning Objective

- Analyze common Web site attacks, weaknesses, and security best practices.

Key Concepts

- Sources of Web site attacks and weaknesses
- Attack techniques using available tools and sources
- Web site security best practices

Identify Attacks and Weaknesses

- Web Application Security Consortium (WASC)
  - Lists 34 types of Web attacks and 15 classes of weaknesses
  - Maintains database of Web site hacking incidents

Threats Identified by WASC

using intended func to perform undesirable outcomes
e.g. password recovery, send-mail, etc.

Abuse of Functionality

Brute-Force Attack

aka session hijacking:
atker issues reqs with compromised user's privileges

more data written to blk of mem, or buffer, than allocated to hold; atker can modify portions of target process' addr space
e.g. ctrl process execution, crash process, modify internal vars, etc.

Buffer Overflow

can ctrl proc exec by locating the function pointer and ctrling it

injecting malicious payload that is interpreted as legitimate content of web app

Content Spoofing

Credential/ Session Prediction

xss is atk involving echoing atker-supplied code into a user's browser instance

Cross-Site Scripting (XSS)

types:
1. non-persistent/DOM-based
trigger when submitting form or embedded client such as Adobe Flash or spcfc link with malicious code

2. persistent
trigger when visit site

Cross-Site Request Forgery

Denial of Service (DoS)

A cross-site request forgery is an attack that involves forcing a victim to send an HTTP request to a target destination without their knowledge or intent in order to perform an action as the victim. The underlying cause is application functionality using predictable URL/form actions in a repeatable way. The nature of the attack is that CSRF exploits the trust that a web site has for a user. By contrast, cross-site scripting (XSS) [9] exploits the trust that a user has for a web site. Like XSS, CSRF attacks are not necessarily cross-site, but they can be. Cross-site request forgery is also known as CSRF, XSRF, one-click attack, session riding, confused deputy, and sea surf.

Threats Identified by WASC (Cont.)

If an attacker passes a format string consisting of printf conversion characters (e.g. "%f", "%p", "%n", etc.) as a parameter value to the web application, they may:

- Execute arbitrary code on the server
- Read values off the stack
- Cause segmentation faults / software crashes

format strings, related to integer overflow and buffer overflow. all 3 deal with manipulating memory or its interpretation

first step for attackers usually. footprint target's web presence and gather as much info as possible. use this to develop attack plan

Fingerprinting

e.g. if sending conversion chars such as %f and no args are supplied,

Format String

data from stack will be used in accordance w/order expected by printf function

HTTP Response Smuggling

HTTP response smuggling is a technique to " smuggle" 2 HTTP responses from a server to a client, through an intermediary HTTP device that expects (or allows) a single response from the server.

in order to evade anti-HTTP response splitting measures.

The essence of HTTP Response Splitting is the attacker's ability to send a single HTTP request that forces the web server to form an output stream, which is then interpreted by the target as two HTTP responses instead of one response, in the normal case. atker completely controls 2nd response

HTTP Response Splitting

HTTP Request Smuggling is an attack technique that abuses the discrepancy in parsing of non-RFC-compliant HTTP requests between two HTTP devices (typically a front-end proxy or HTTP-enabled firewall and a back-end web server) to smuggle a request to the second device "through" the first device.

HTTP Request Smuggling

HTTP Request Splitting

HTTP Request Splitting is an attack that enables forcing the browser to send arbitrary HTTP requests, inflicting XSS and poisoning the browser's cache. send 1 instead of 2. 2nd can be attacker's website

e.g. when making purchase, overflow can cause company to give money to attacker (neg. val purc)

arithmetic operation that exceeds max size of integer type used to store it; causes value to be wrapped around

Integer Overflow

can influence val of vars in unintended way of dev

sending set a of requests, but target sees set b of requests
facilitates several possible exploitations,

LDAP Injection

LDAP Injection is an attack technique used to exploit web sites that construct LDAP statements from user-supplied input.

would run with elevated permissions

Mail Command Injection

Mail Command Injection is an attack technique used to exploit mail servers and webmail applications that construct IMAP/SMTP statements from user-supplied input that is not properly sanitized.

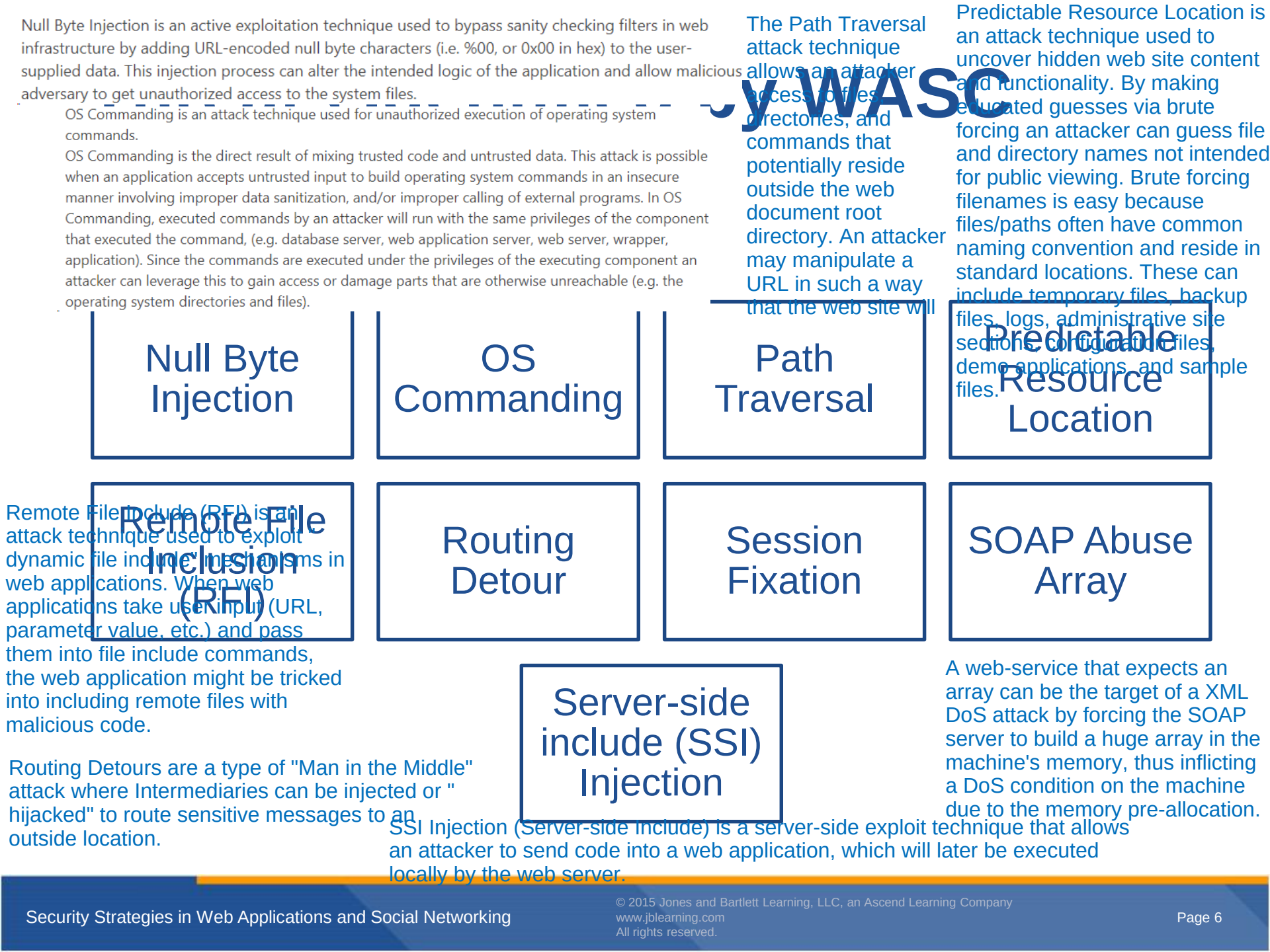
Null Byte Injection is an active exploitation technique used to bypass sanity checking filters in web infrastructure by adding URL-encoded null byte characters (i.e. %00, or 0x00 in hex) to the user-supplied data. This injection process can alter the intended logic of the application and allow malicious adversary to get unauthorized access to the system files.

OS Commanding is an attack technique used for unauthorized execution of operating system commands.

OS Commanding is the direct result of mixing trusted code and untrusted data. This attack is possible when an application accepts untrusted input to build operating system commands in an insecure manner involving improper data sanitization, and/or improper calling of external programs. In OS Commanding, executed commands by an attacker will run with the same privileges of the component that executed the command, (e.g. database server, web application server, web server, wrapper, application). Since the commands are executed under the privileges of the executing component an attacker can leverage this to gain access or damage parts that are otherwise unreachable (e.g. the operating system directories and files).

The Path Traversal attack technique allows an attacker access to files, directories, and commands that potentially reside outside the web document root directory. An attacker may manipulate a URL in such a way that the web site will

Predictable Resource Location is an attack technique used to uncover hidden web site content and functionality. By making educated guesses via brute forcing an attacker can guess file and directory names not intended for public viewing. Brute forcing filenames is easy because files/paths often have common naming convention and reside in standard locations. These can include temporary files, backup files, logs, administrative site sections, configuration files, demo applications, and sample files.



Remote File Include (RFI) is an attack technique used to exploit dynamic file include mechanisms in web applications. When web applications take user input (URL, parameter value, etc.) and pass them into file include commands, the web application might be tricked into including remote files with malicious code.

Routing Detours are a type of "Man in the Middle" attack where Intermediaries can be injected or " hijacked" to route sensitive messages to an outside location.

A web-service that expects an array can be the target of a XML DoS attack by forcing the SOAP server to build a huge array in the machine's memory, thus inflicting a DoS condition on the machine due to the memory pre-allocation.

SSI Injection (Server-side Include) is a server-side exploit technique that allows an attacker to send code into a web application, which will later be executed locally by the web server.

Threats Identified by WASC (con't)

SQL Injection is an attack technique used to exploit applications that construct SQL statements from user-supplied input. When successful, the attacker is able to change the logic of SQL statements executed against the database.

xpath is similar to sql, except it is for XML Path Language queries formed from user-supplied input

XML Attribute Blowup is a denial of service attack against XML parsers. The attacker provides a malicious XML document, which vulnerable XML parsers process in a very inefficient manner, leading to excessive CPU load.

SQL Injection	URL Redirector Abuse	XPath Injection	XML Attribute Blowup
XML External Entities	XML Entity Expansion	XML Injection	XQuery Injection

XXE takes advantage of a feature of XML to build documents dynamically at the time of processing. An XML message can either provide data explicitly or by pointing to an URI where the data exists. In the attack technique, external entities may replace the entity value with malicious data, alternate referrals or may compromise the security of the data the server/XML application has access to.

overwhelms xml parsers; recursive entity expansion

essentially a DoS
eats up avail srv rsrc

The injection of unintended XML content and/or structures into an XML message can alter the intend logic of the application.

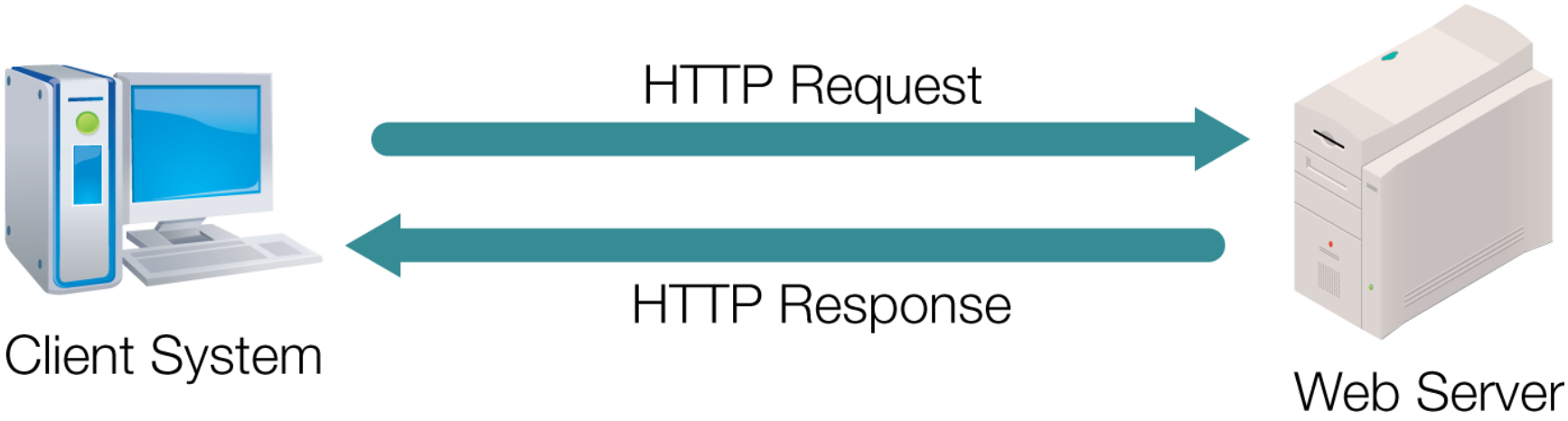
XQuery Injection is a variant of the classic SQL injection attack against the XML XQuery Language.

HTTP Communication Process

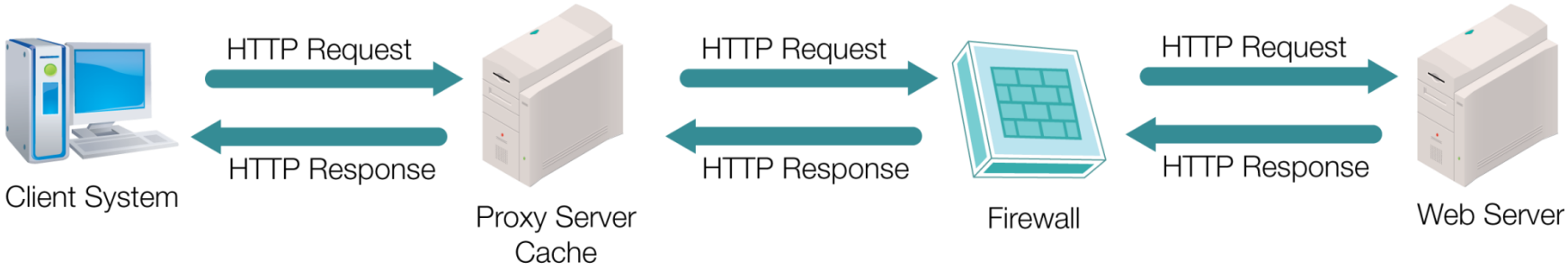
HTTP Response Splitting

In the HTTP Response Splitting attack, there are always 3 parties (at least) involved:

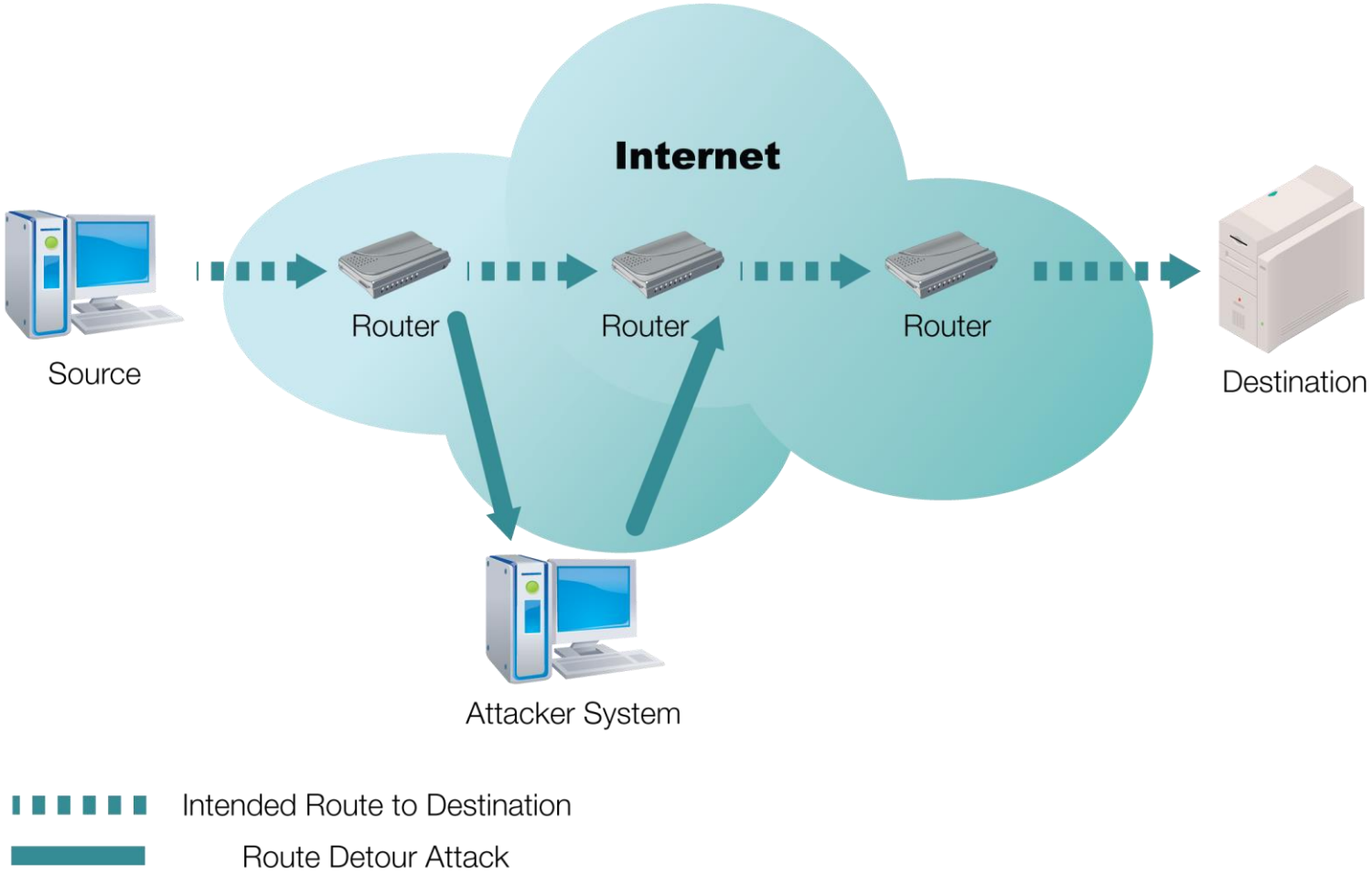
- Web server, which has a security hole enabling HTTP Response Splitting
- Target - an entity that interacts with the web server perhaps on behalf of the attacker. Typically this is a cache server forward/reverse proxy), or a browser (possibly with a browser cache).
- Attacker - initiates the attack



HTTP Communication Through Intermediary Points



Routing Detour Attack

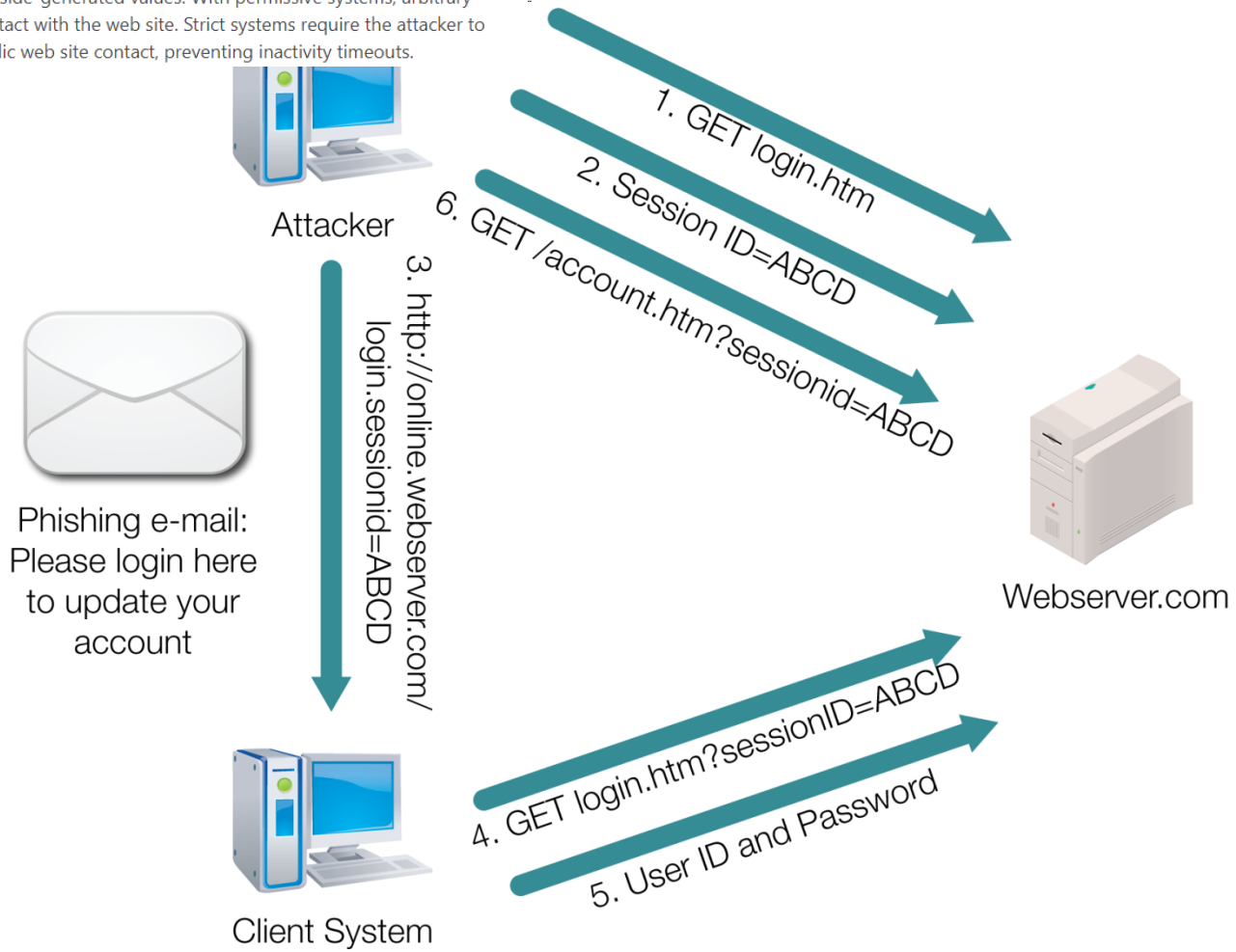


Session Fixation is an attack technique that forces a user's session ID to an explicit value. Depending on the functionality of the target web site, a number of techniques can be utilized to "fix" the session ID value. These techniques range from Cross-site Scripting exploits to peppering the web site with previously made HTTP requests. After a user's session ID has been fixed, the attacker will wait for that user to login. Once the user does so, the attacker uses the predefined session ID value to assume the same online identity.

Generally speaking there are two types of session management systems when it comes to ID values. The first type is "permissive" systems that allow web browsers to specify any ID. The second type is "strict" systems that only accept server-side-generated values. With permissive systems, arbitrary session IDs are maintained without contact with the web site. Strict systems require the attacker to maintain the "trap-session", with periodic web site contact, preventing inactivity timeouts.

Without active protection against Session Fixation, the attack can be mounted against any web site that uses sessions to identify authenticated users. Web sites using sessions IDs are normally cookie-based, but URLs and hidden form fields are used as well. Unfortunately, cookie-based sessions are the easiest to attack. Most of the currently identified attack methods are aimed toward the fixation of cookies.

In contrast to stealing a users' session IDs after they have logged into a web site, Session Fixation provides a much wider window of opportunity. The active part of the attack takes place before a user logs in.



Fifteen Web Site Attacks

- Application Misconfiguration
  - Many applications come with unnecessary and unsafe features, such as debug and QA features, enabled by default. gives elevated permissions
- Directory Indexing
  - Automatic directory listing/indexing is a web server function that lists all of the files within a requested directory if the normal base file (index.html/home.html/default.htm/default.asp/default.aspx/index.php) is not present. same as issuing `ls` command
- Improper File System Permissions
  - When improper permissions are set, an attacker may be able to access restricted files or directories and modify or delete their contents.
- Improper Input Handling
  - input handing is used to describe functions like validation, sanitization, filtering, encoding and/or decoding of input data
  - all input should be considered untrusted and potentially malicious.
  - Buffer Overflows, SQL Injection, OS Commanding, Denial of Service
- Improper Output Handling
  - If an application has improper output handling, the output data may be consumed leading to vulnerabilities and actions never intended by the application dev
- Information Leakage
  - Information Leakage is an application weakness where an application reveals sensitive data, such as technical details of the web application, environment, or user-specific data.
- Insecure Indexing
  - search engines index websites, attackers can query search engine to find out about files which are not supposed to be publicly accessible
- Insufficient Anti-Automation
  - e.g. use capcha to prevent automating login forms
- Insufficient Authentication
  - access sensitive w/o authenticating

Fifteen Web Site Attacks (Cont.)

- Insufficient Authorization — auth? okay do w.e u want, idc about what authority u have lol
- Insufficient Password Recovery — need limitation, otherwise can brute force
- Insufficient Process Validation — atker doesnt follow flow intended; circumvents process go from step 1 to 7 for ex
- Insufficient Session Expiration — reuse old session IDs for auth
- Insufficient Transport Layer Protection
- Server Misconfiguration

Insufficient TLP
Websites typically use Secure Sockets Layer / Transport Layer Security (SSL/TLS) to provide encryption at the transport layer [1]. However, unless the website is configured to use SSL/TLS and configured to use SSL/TLS properly, the website may be vulnerable to traffic interception and modification. untrusted third parties can take adv of this

An Example of CAPTCHA

Best Practices

Mitigating Attack Risks	Implement a best practices approach.
	Be security conscious as early as possible.
	Know your infrastructure.
	Be proactive in gaining necessary support at all levels.
Mitigating Weaknesses	Practice due diligence for mitigating weaknesses.
	Be aware of vulnerabilities.
	Be aware of WASC's threats to Web Application security.
	Validate user input.

Summary

- Sources of the Web site attacks and weaknesses
- Attack techniques using available tools and sources
- Web site security best practices

Virtual Lab

- Applying OWASP to a Web Security Assessment