Web and Database Attacks

Edited by:
Prof. Ahmed Redha
PSU

Learning Objective

- Perform system hacking, and web and database attacks.

Key Concepts

- Web server vulnerabilities, tools, and exploits
- Web application vulnerabilities, tools, and exploits
- Database attacks and attack tools

Web Server Operations

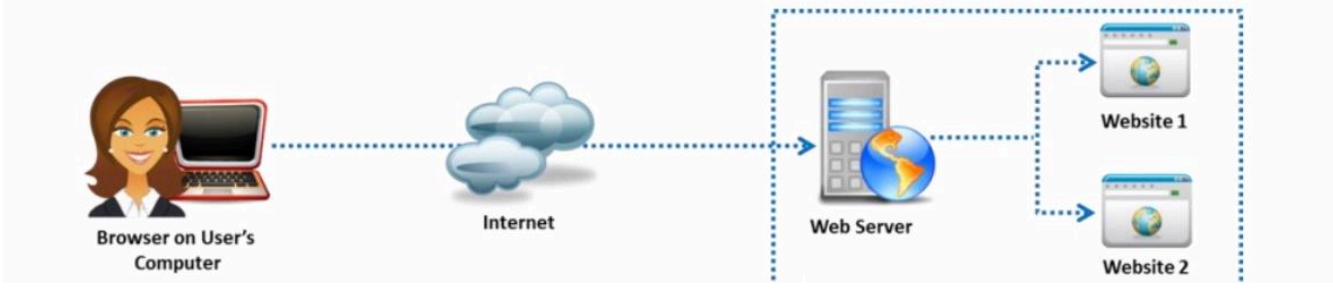
- A web server is a computer system that stores, processes, and delivers web pages to global clients via the Hypertext Transfer Protocol (HTTP).
- In general, a client initiates a communication process through HTTP requests. When a client desires to access any resource such as web pages, photos, and videos, the client's browser generates an HTTP request that is sent to the web server. Depending on the request, the web server collects the requested information/content from the data storage or application servers and responds to the client's request with an appropriate HTTP response.
- If a web server cannot find the requested information, then it generates an error message.

Why web server is an attractive target?

Web Server Security Issues

- A web server is a hardware/software application that hosts websites and makes them accessible over the internet.
- A web server, along with a browser, successfully implements client-server model architecture in which the web server plays the server part in the model and the browser acts as the client.
- To host websites, a web server stores various web pages of the websites and delivers the web page upon request.
- Each web server has a domain name and the IP address associated with that domain name.
- A web server can host more than one website.
- Any computer can act as a web server if it has specific server software (a web server program) installed in it and is connected to the internet.
- Web servers are chosen based on their capability to handle server-side programming, security characteristics, publishing, search engine, and site-building tools.
- An attacker usually targets vulnerability that exists in the software component and configuration errors to compromise web servers.

- Organizations can defend most network-level and OS-level attacks by using network security measures such as firewalls, IDS, IPS, and so on and by following security standards and guidelines. This forces attackers to turn their attention to perform web server and web application-level attacks as web server hosting web applications is accessible from anywhere over the internet. **This makes web servers an attractive target.**
- A poorly configured web server can punch a hole in the most carefully designed firewall system. Attackers can exploit a poorly configured web server with known vulnerabilities to compromise the security of the web application. A leaky server can harm an organization.



Common Goals behind Web Server Hacking

Attackers perform web server attacks with certain goals in mind. These goals may be either technical or non-technical. For example, attackers may breach security of the web server and steal sensitive information for financial gains or only for the sake of curiosity. Following are some goals behind a web server attack:

- Stealing credit cards or other sensitive credentials using phishing techniques
- Integrating the server in a botnet in order to perform Denial of Service (DoS) or Distributed Denial of Service (DDoS) attack
- Compromising a database
- Obtaining closed-source applications
- Hiding and redirecting traffic
- Escalating privileges

Some attacks are performed for personal reasons, rather than financial gains:

- For pure curiosity
- For completing a self-set intellectual challenge
- For damaging the target organization's reputation

Dangerous Security Flaws Affecting Web Server

- Security Web server configuration by poorly trained system administrators may leave security vulnerabilities in the web server.
- Inadequate knowledge, negligence, laziness, and inattentiveness toward security can pose the biggest threats to web server security.

Following are some of the common oversights that make a web server vulnerable to attacks:

- Not updating the web server with the latest patches
- Using the **same sys admin** credentials everywhere
- Allowing unrestricted internal and outbound traffic
- Running unhardened applications and servers

Why Web Servers Are Compromised

Client code
Script

- Web servers are publicly accessible over the Internet
- They host web applications, databases, and sensitive data
- Security controls must allow inbound access

This increases the attack surface

Shift Toward Web Layer Attacks

- Network defenses reduce network-level attacks
- Attackers target web servers and applications
- Web servers act as a bridge to internal systems
- Compromise can bypass perimeter defenses

Webmaster's Perspective

- Web servers may expose internal networks
- Risks include vulnerable scripts and backend code
- Open architecture allows server-side execution
- A single flaw can compromise the server

Network Administrator's Perspective

- Poor configuration creates security gaps
- Weak authentication and access control
- Excessive privileges granted
- Misconfiguration is a major attack vector

End User's Perspective

- Users assume web browsing is safe
- Malicious web content can deliver malware
- Client-side scripts can be exploited
- Compromised servers can attack users

Root Causes of Web Server Compromise

- Unpatched software
- Poor security configuration
- Vulnerable web applications
- Weak authentication
- Human error

Configuration & Human Factors

- Administrator negligence
- Lack of security awareness
- Default credentials
- Unnecessary services enabled

Key Security Principle

- A poorly configured web server can bypass firewalls
- Network security alone is insufficient
- Web server security is critical

Impact of Web Server Compromise

- Data theft and tampering
- Website defacement
- Malware distribution
- Privilege escalation
- Secondary attacks

World Wide Web

File Edit View Help

http://juggyboy.com/index.aspx

You are OWNED!!!!!!!

HACKED!

Hi Master, Your website owned by US, Hacker!

Next target – microsoft.com

Improper or poor web design
Source Code Example 1

```
<form method="post" action="../../cgi-bin/formMail.pl">
<!—Regular FormMail options---->
<input type=hidden name="recipient" value=" someone@someplace.com>
<input type=hidden name="subject" value="Message from website visitor">
<input type=hidden name="required" value="Name,Email,Address1,City,State,
Zip,Phone1">
<input type=hidden name="redirect" value="http://www.someplace.com
/received.htm">
<input type=hidden name="servername" value="https://payments
.someplace.com">
<input type=hidden name="env_report" value="Form Results">
<input type=hidden name="return_link_url" value="http://www.someplace
.com/main.html">
<input type=hidden name="return_link_title" value="Back to Main Page">
<input type=hidden name="missing_fields_redirect" value="http://www
.someplace.com/error.html">
<input type=hidden name="orderconfirmation" value="order@someplace.com">
<input type=hidden name="cc" value="j.halak@someplace.com">
<input type=hidden name="bcc" value="c.price@someplace.com">
  <!—Courtesy Reply Options-->
```

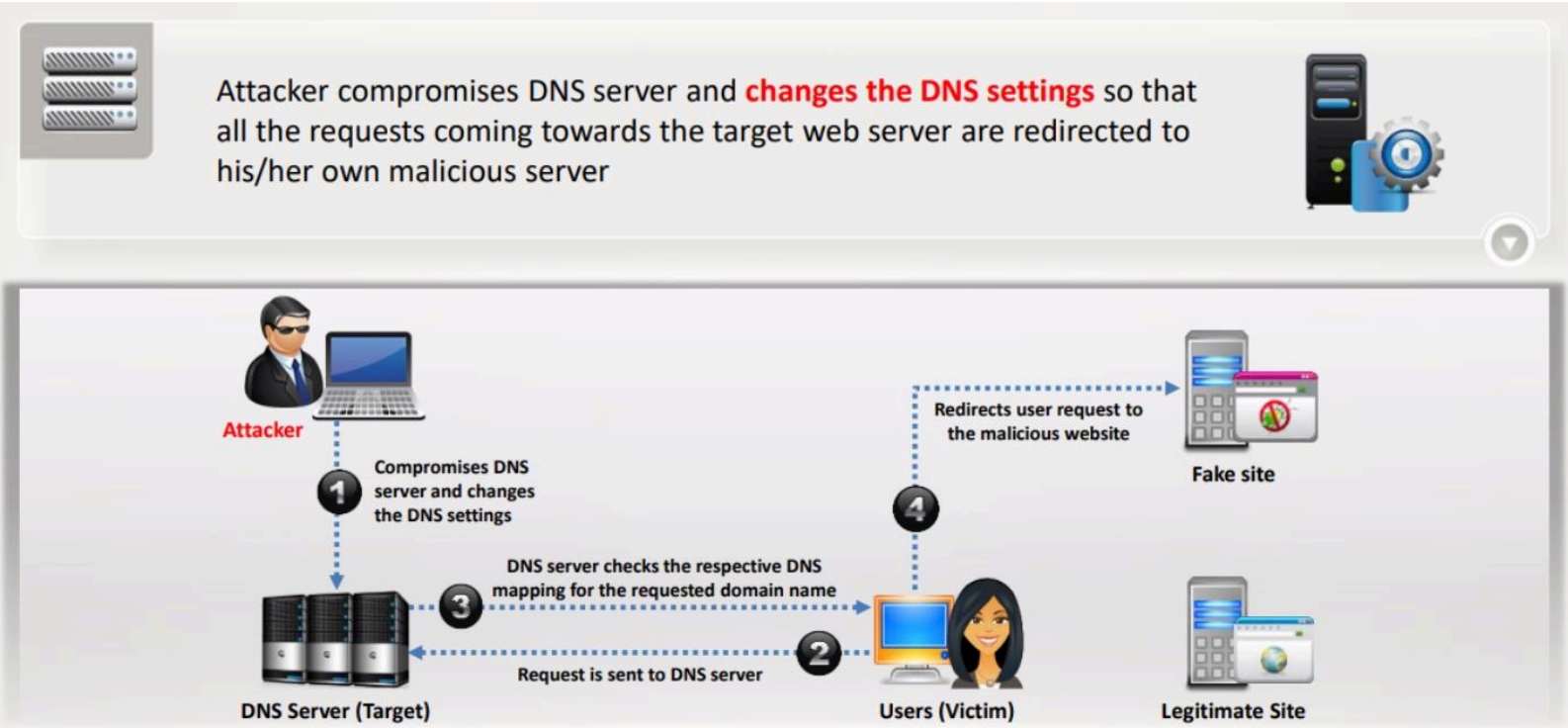
Web Server Vulnerabilities

- Buffer overflow
- Denial of service (DoS) attack
- Distributed denial of service (DDoS) attack
- Banner information
- Permissions
- Error messages
- Unnecessary features
- User accounts
- Structured Query Language (SQL) injections

Web Server Attacks

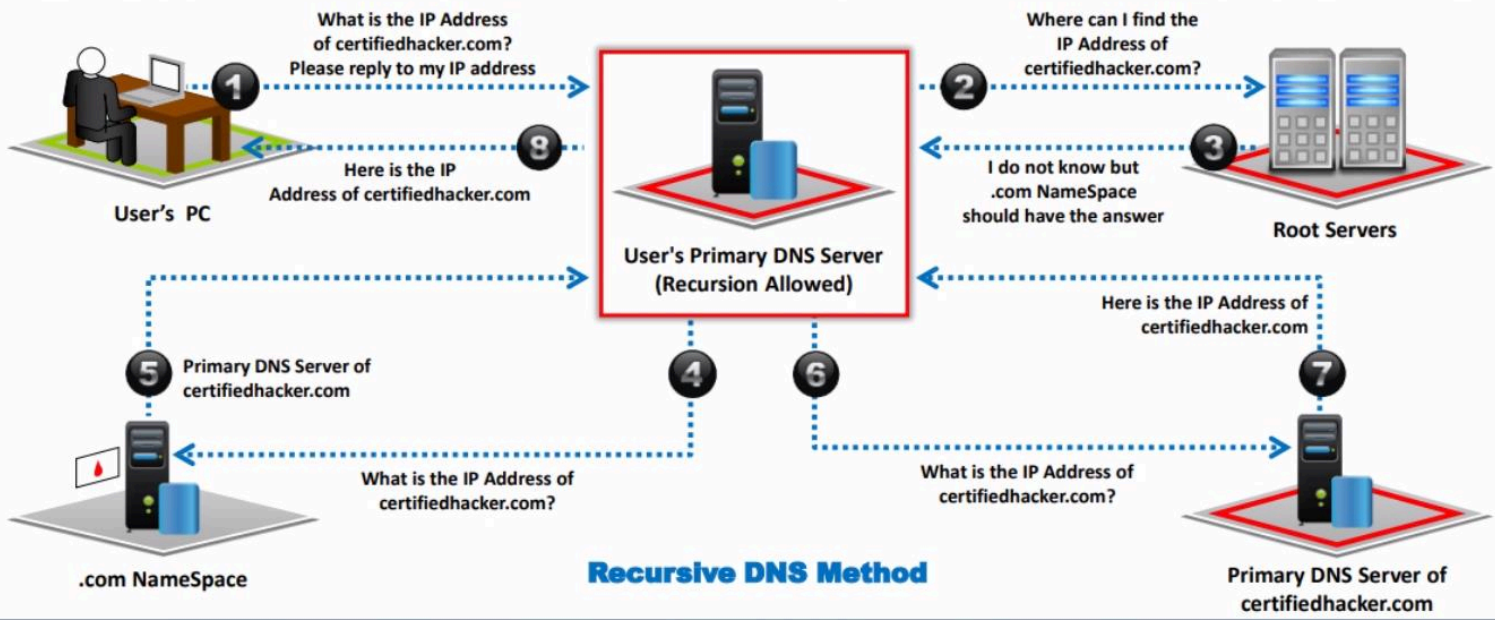
DNS Server Hijacking

- Domain Name System (DNS) resolves a domain name to its corresponding IP address. A user queries the DNS server with a domain name, and it delivers the corresponding IP address. In a DNS server hijacking, an attacker compromises the DNS server and changes the mapping settings of the target DNS server to redirect toward a rogue DNS server so that it would redirect the user's requests to the attacker's rogue server. Thus, when the user types the legitimate URL in a browser, the settings will redirect to the attacker's fake site.

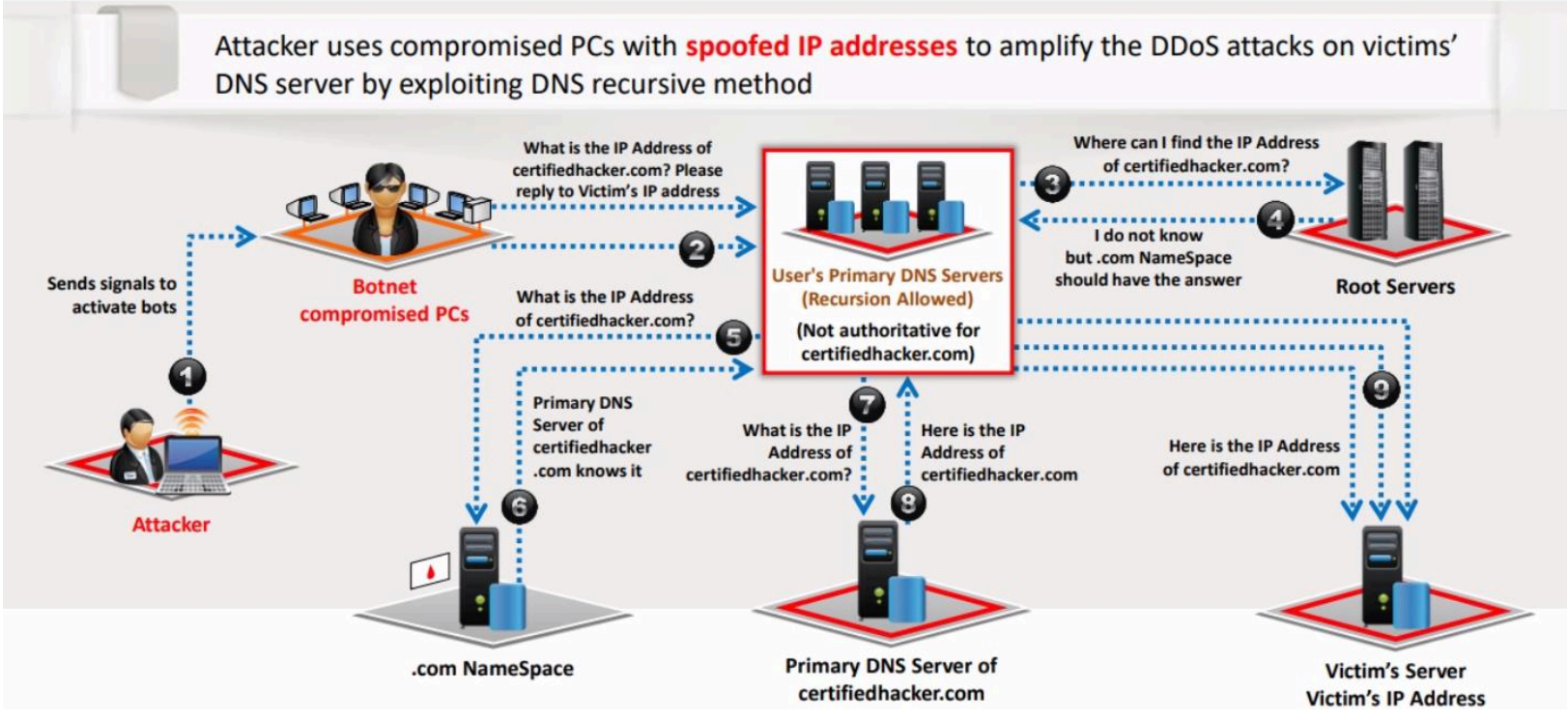


DNS Amplification Attack

- Attacker takes advantage of **DNS recursive method** of DNS redirection to perform DNS amplification attack

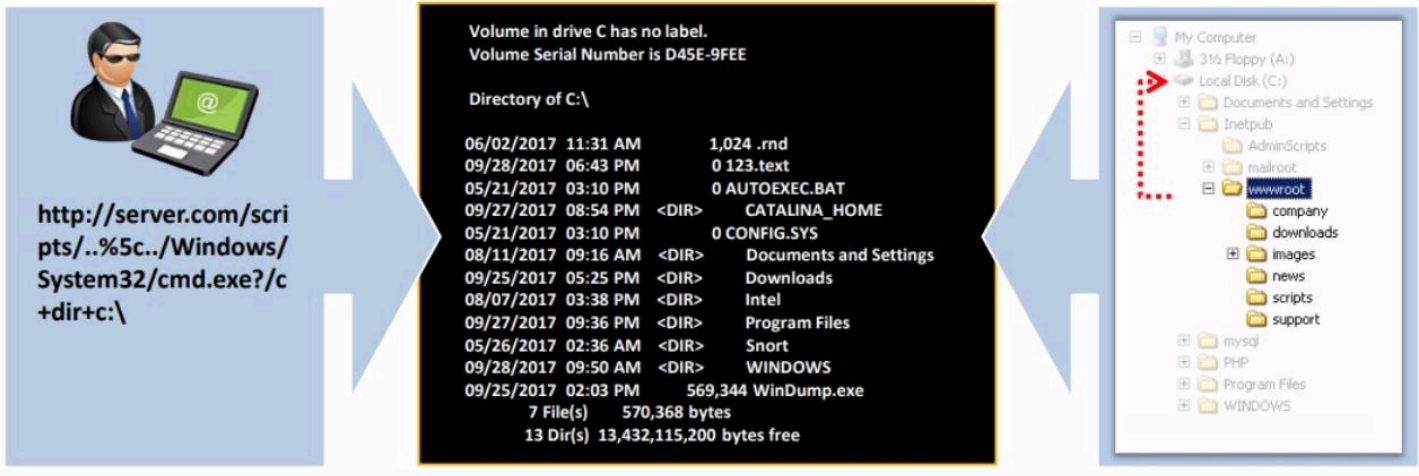


DNS Amplification Attack (cont'd)



Directory Traversal Attack

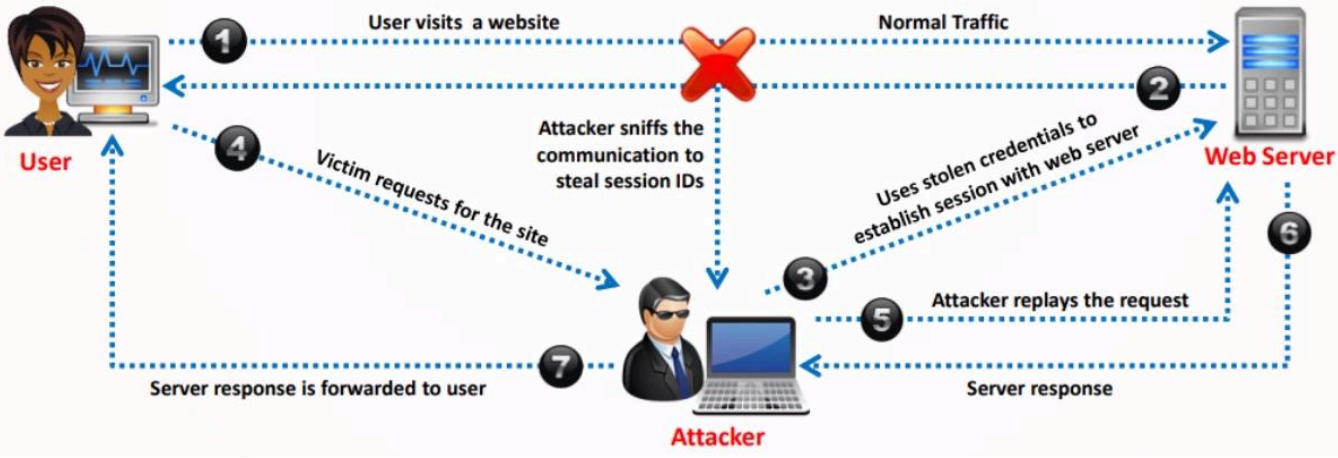
- In directory traversal attacks, attackers use **../ (dot-dot-slash)** sequence to access restricted directories outside of the web server root directory
- Attackers can use **trial and error method** to navigate outside of the root directory and access sensitive information in the system



Man-in-the-Middle/Sniffing Attack

01 Man-in-the-Middle (MITM) attacks allow an attacker to access sensitive information by **intercepting and altering communications** between an end-user and web servers

02 Attacker **acts as a proxy** such that all the communication between the user and web server passes through him



Content inside the cache

Web Cache Poisoning Attack

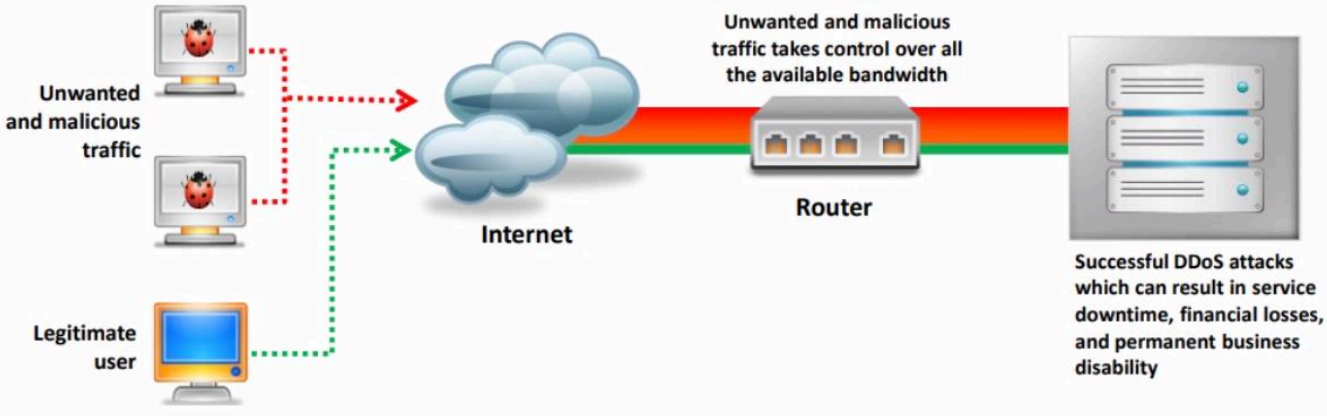
solution: refresh

Web cache poisoning attacks the reliability of an intermediate web cache source.

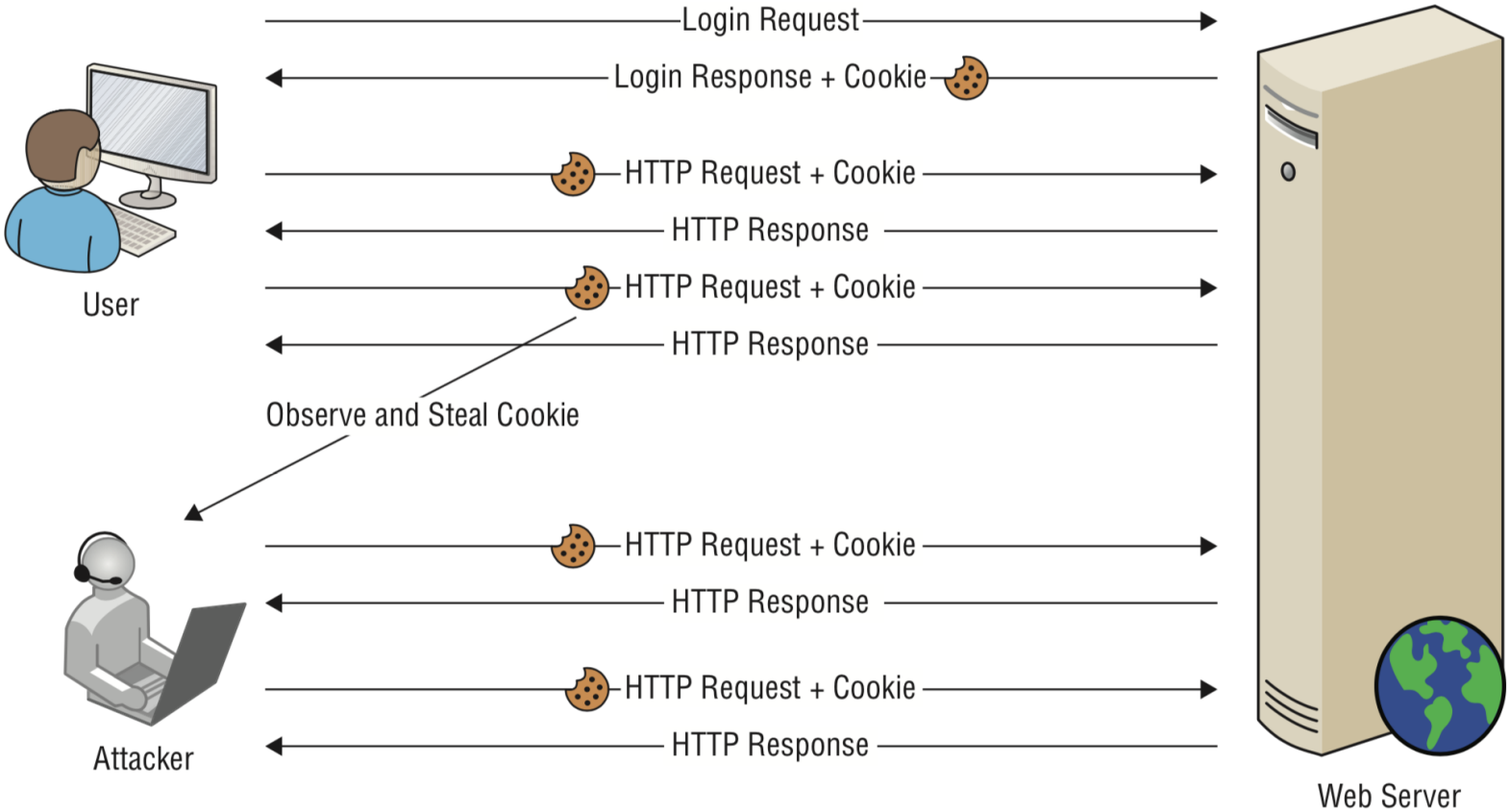
In this attack, the attackers swap cached content for a random URL with infected content. Users of the web cache source can unknowingly use the poisoned content instead of true and secured content when requesting the required URL through the web cache. An attacker forces the web server's cache to flush its actual cache content and sends a specially crafted request to store in cache. In this case, all the users of that web server cache will get malicious content until the servers flush the web cache. Web cache poisoning attacks are possible if the web server and application has HTTP Response-Splitting flaws.

DoS/DDoS Attack

- Attackers may send numerous **fake requests** to the web server which results in the **web server crashing** or becoming unavailable to the legitimate users
- Attackers may target **high profile web servers** such as banks, credit card payment gateways, government owned services, etc. to **steal user credentials**



Session Hijacking with cookies



Web Server Attack Methodology

Banner Information

- A banner reveals information about a web server
- Connect to port 80 (HTTP) on a web server to receive web server's banner;

```
[root@prober]# nc www.targethost.com 80
HEAD / HTTP/1.1

HTTP/1.1 200 OK
Date: Mon, 11 May 2009 22:10:40 EST
Server: Apache/2.0.46 (Unix)  (Red Hat/Linux)
Last-Modified: Thu, 16 Apr 2009 11:20:14 PST
ETag: "1986-69b-123a4bc6"
Accept-Ranges: bytes
Content-Length: 1110
Connection: close
Content-Type: text/html
```

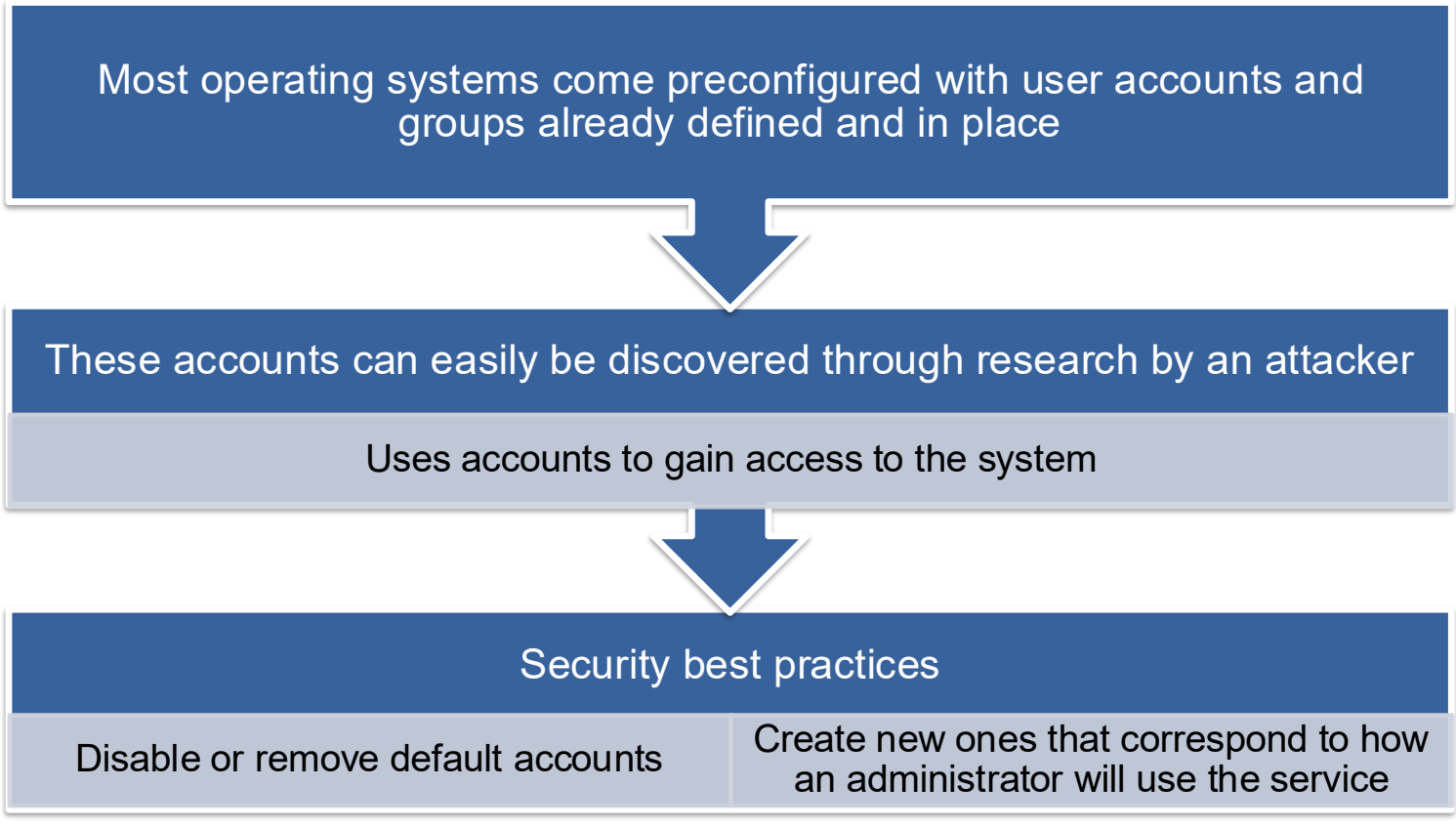
Permissions

Permissions:	Incorrectly assigned permissions:
• Control access to a server and its content • Can easily be incorrectly configured	• Have potential to allow access to locations on web server that should not be accessible

Unnecessary Features

- Servers should be purpose-built to the role they will fill in the organization.
- Anything not essential to this role should be eliminated
- Hardening removes features, services, and applications that are not necessary for a system to do its appointed job.
- If a feature or service is not needed, it should be disabled or, better yet, uninstalled.

User Accounts



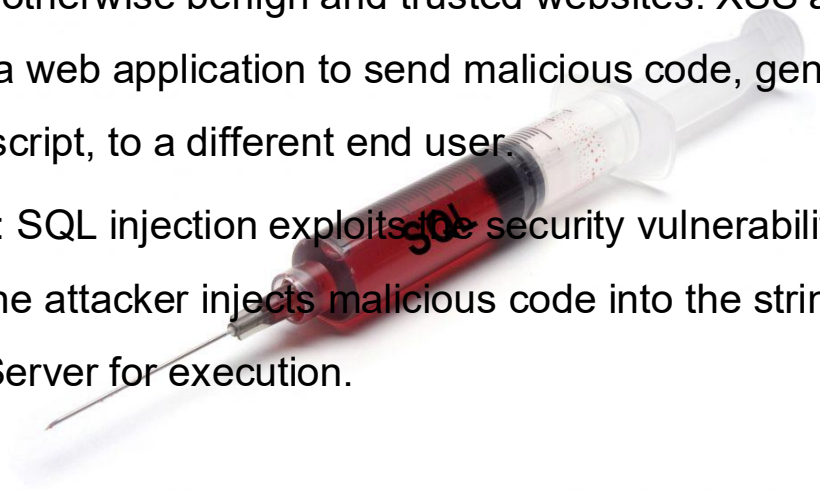
Web Application Attacks

Web Application Attacks

- Even if web servers are configured securely or are secured using network security measures such as firewalls, a poorly coded web application deployed on the web server may give a path to an attacker to compromise the web server's security.
- If the web developers do not adopt secure coding practices while developing web applications, it may give attackers the chance to exploit vulnerabilities and compromise web applications and web server security. An attacker can perform different types of attacks on vulnerable web applications to breach web server security.

Web Application Attacks (cont'd)

- **Cookie Tampering:** Cookie tampering attacks occur when sending a cookie from the client side to the server. Different types of tools help in modifying persistent and non-persistent cookies.
- **Unvalidated Input and File Injection Attacks:** Unvalidated input and file injection attacks are performed by supplying an unvalidated input or by injecting files into a web application.
- **Cross-Site Scripting (XSS)** attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted websites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.
- **SQL Injection Attacks**: SQL injection exploits the security vulnerability of a database for attacks. The attacker injects malicious code into the strings, later passed on to the SQL Server for execution.



Session Management Issues

Session

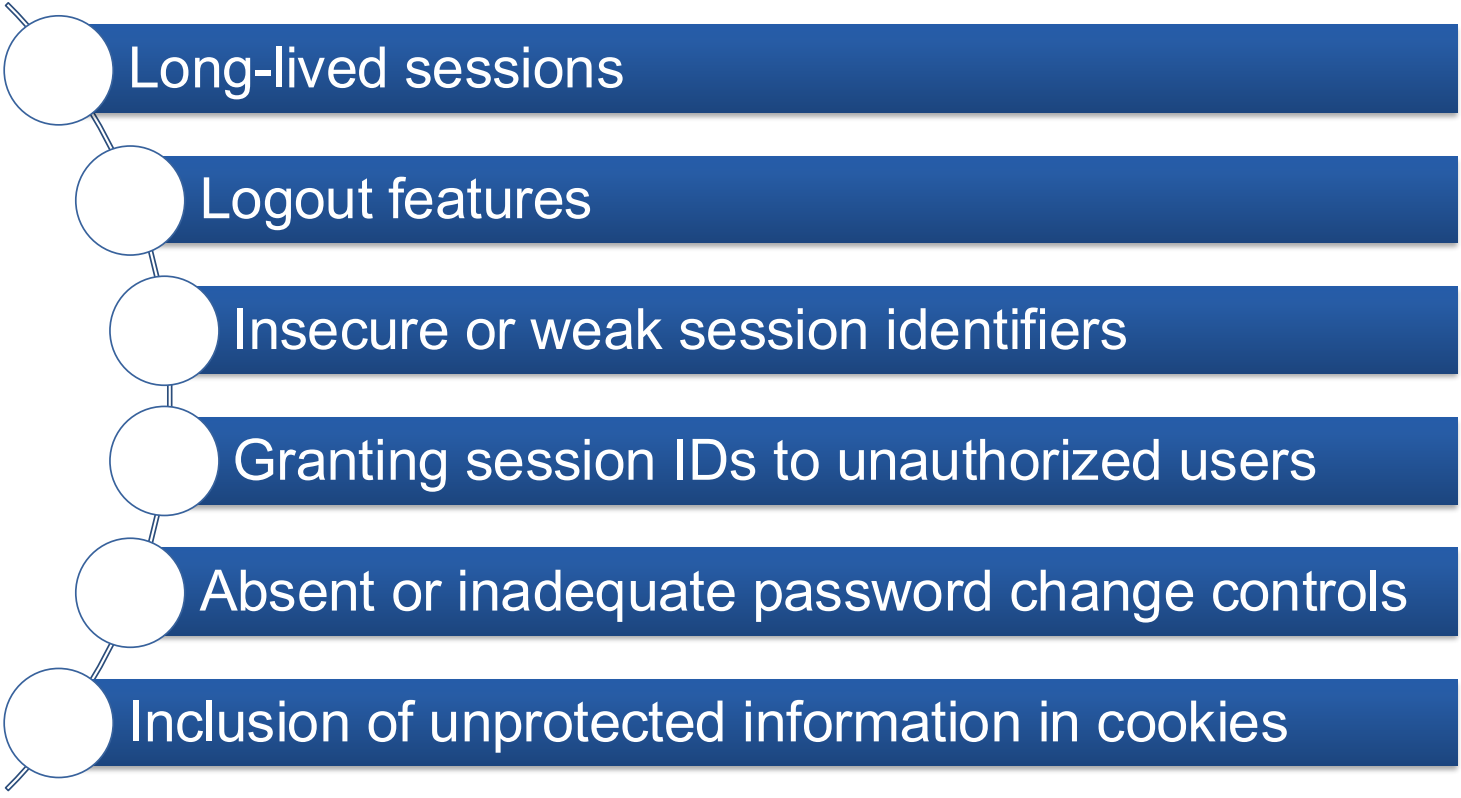
Represents connection that a client has with the server application

Session information can give an attacker access to confidential information

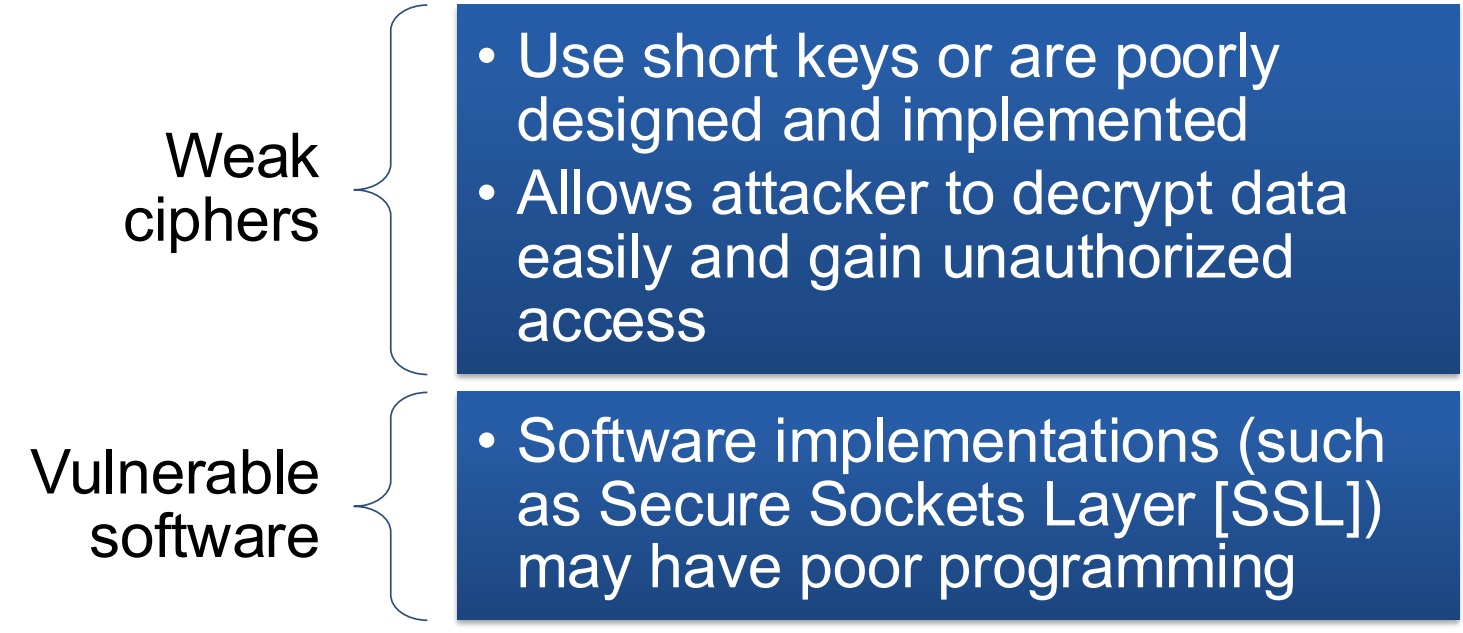
Session will have:

- Unique identifier
- Encryption
- Other parameters

Session Vulnerabilities



Encryption Weaknesses



Database Attacks

Database Attacks and Vulnerabilities Overview

- Databases store critical and sensitive information
- Often connected to web applications
- A single weakness can expose entire systems
- Databases are high-value attack targets

Database Vulnerabilities

- Weak or default passwords
- Excessive user privileges
- Unpatched database software
- Insecure configuration
- Lack of encryption

Database Vulnerabilities (cont.)

- Shared database accounts
- Weak authentication mechanisms
- No role-based access control
- Open database ports
- Poor backup protection

Database Vulnerabilities (cont.)

- Default settings left unchanged
- Unnecessary database services enabled
- Weak auditing and logging
- Inadequate monitoring
- Human configuration errors

Database Attacks

- SQL Injection
- Credential brute-force attacks
- Privilege escalation
- Unauthorized data access
- Data manipulation attacks

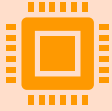
Database Attacks (cont.)

- Abuse of stored procedures
- Exploitation of misconfigured permissions
- Password cracking attacks
- Bypassing authentication controls
- Insider attacks

Impact of Database Attacks

- Data theft
- Data tampering or deletion
- Loss of confidentiality and integrity
- Service disruption
- Legal and reputational damage

Locating Databases on the Network



Network Database Scanner

Effective at locating “rogue” or unknown database installations

SQLRecon

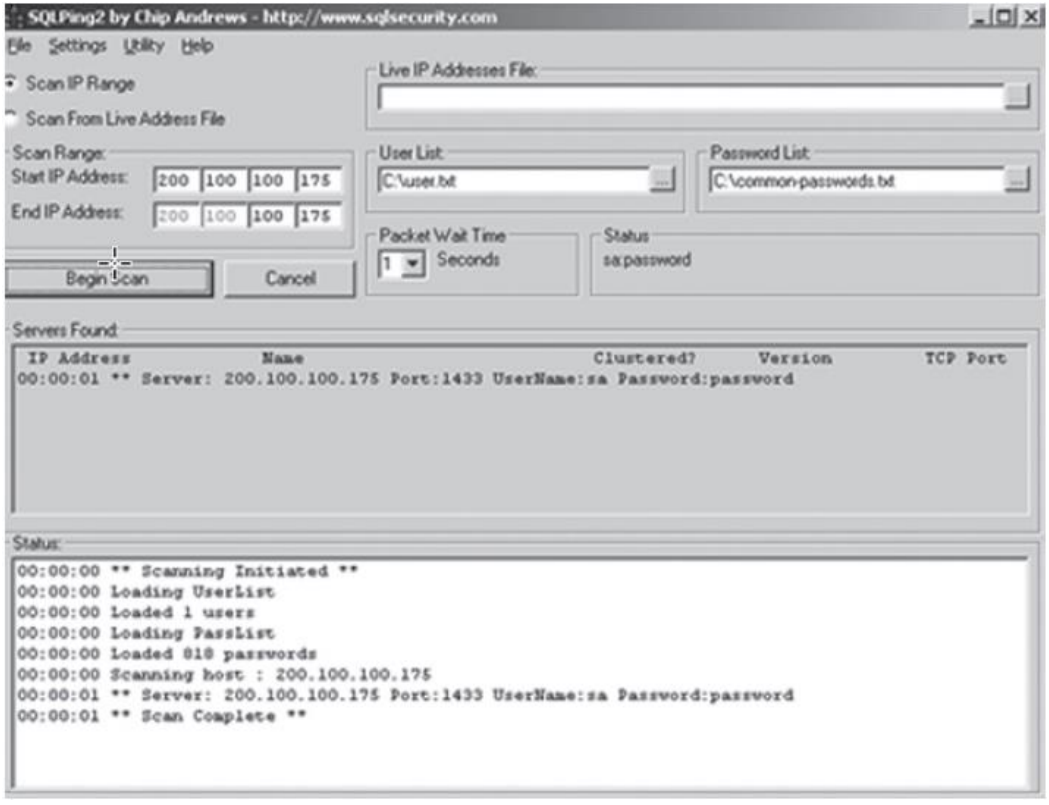
Similar to Network Database Scanner

For Microsoft SQL installations

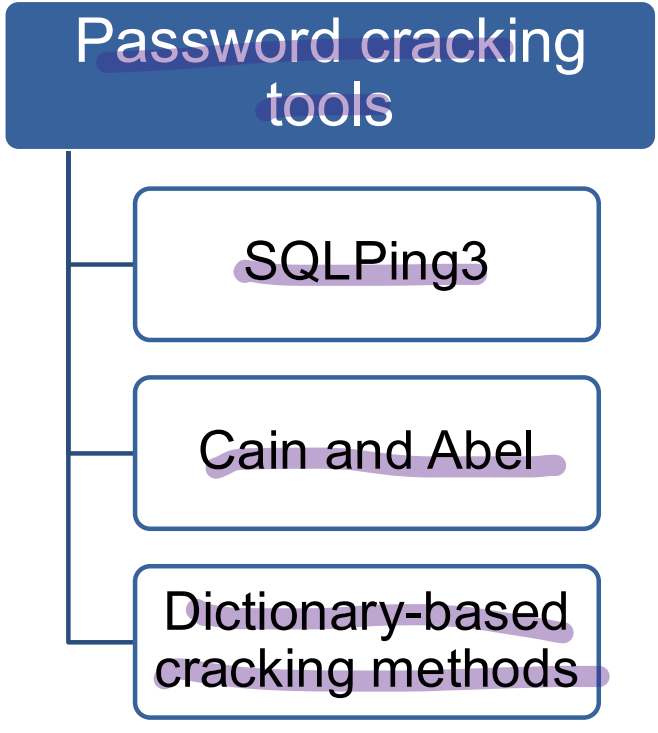
Oscanner

For Oracle installations

Network Database Scanner



Database Server Password Cracking



Locating Vulnerabilities in Databases

- Common vulnerabilities include:
  - Used stored procedures
  - Services account privilege issues
  - Wear or poor authentication methods enabled
  - No or limited audit log settings

Out of Sight, Out of Mind

Learn the security features in the database system

Evaluate the use of nonstandard ports

Keep up to date

Secure the operating system

Use a firewall

Web Server Penetration Testing

من هنا للاخير
قاعد بسوي Summary
على كل التشابتر

- Web server pen testing is used to **identify**, **analyze**, and **report vulnerabilities** such as authentication weaknesses, configuration errors, protocol related vulnerabilities, etc. in a web server
- The best way to perform penetration testing is to **conduct a series of methodical and repeatable tests**, and to work through all of the different application vulnerabilities

Why Web Server Pen Testing?

Verification of Vulnerabilities

To exploit the vulnerability in order to test and fix the issue

Remediation of Vulnerabilities

To retest the solution against vulnerability to ensure that it is completely secure

Identification of Web Infrastructure

To identify make, version, and update levels of web servers; this helps in selecting exploits to test for associated published vulnerabilities

Diff worm ransom-ware or SPY

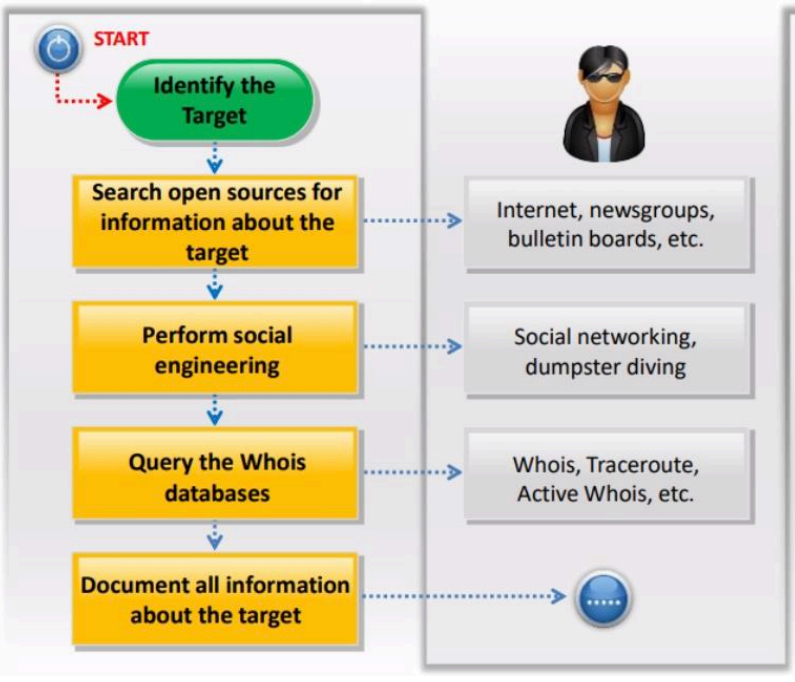
DNS MAN in the middle

Tools

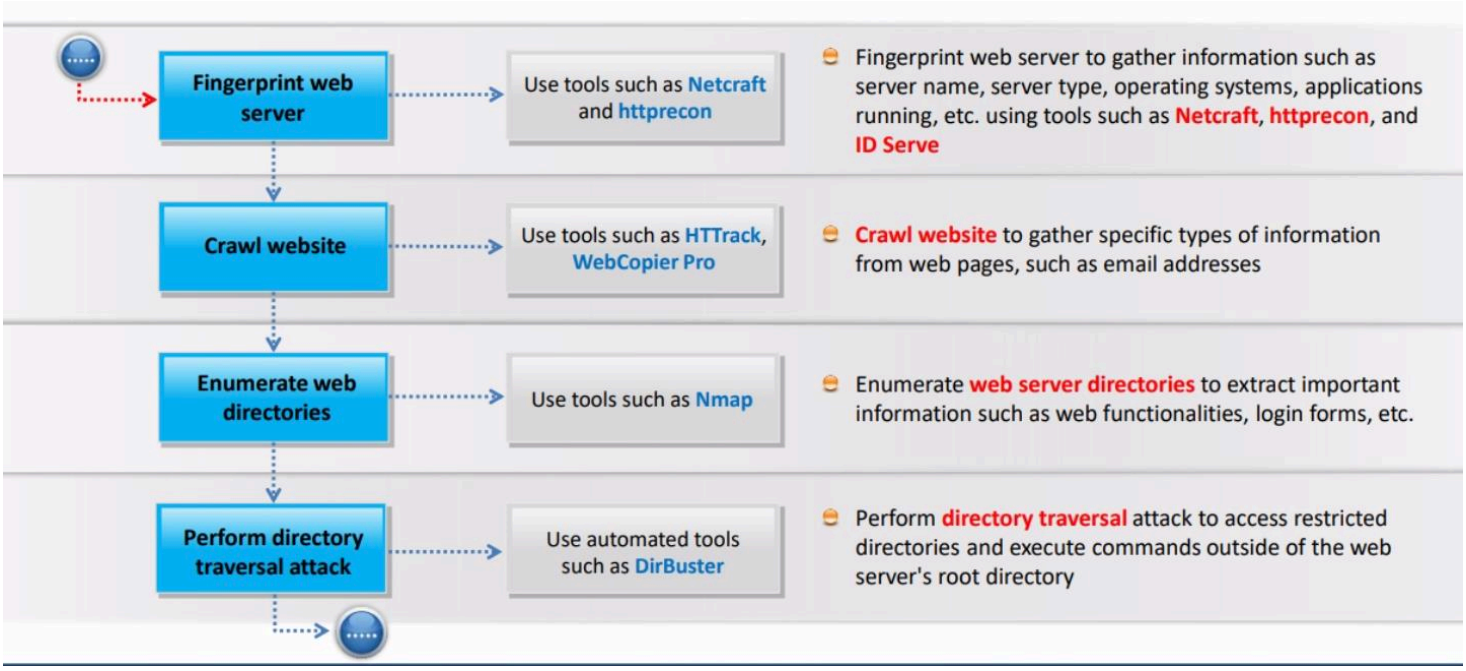
DB slides

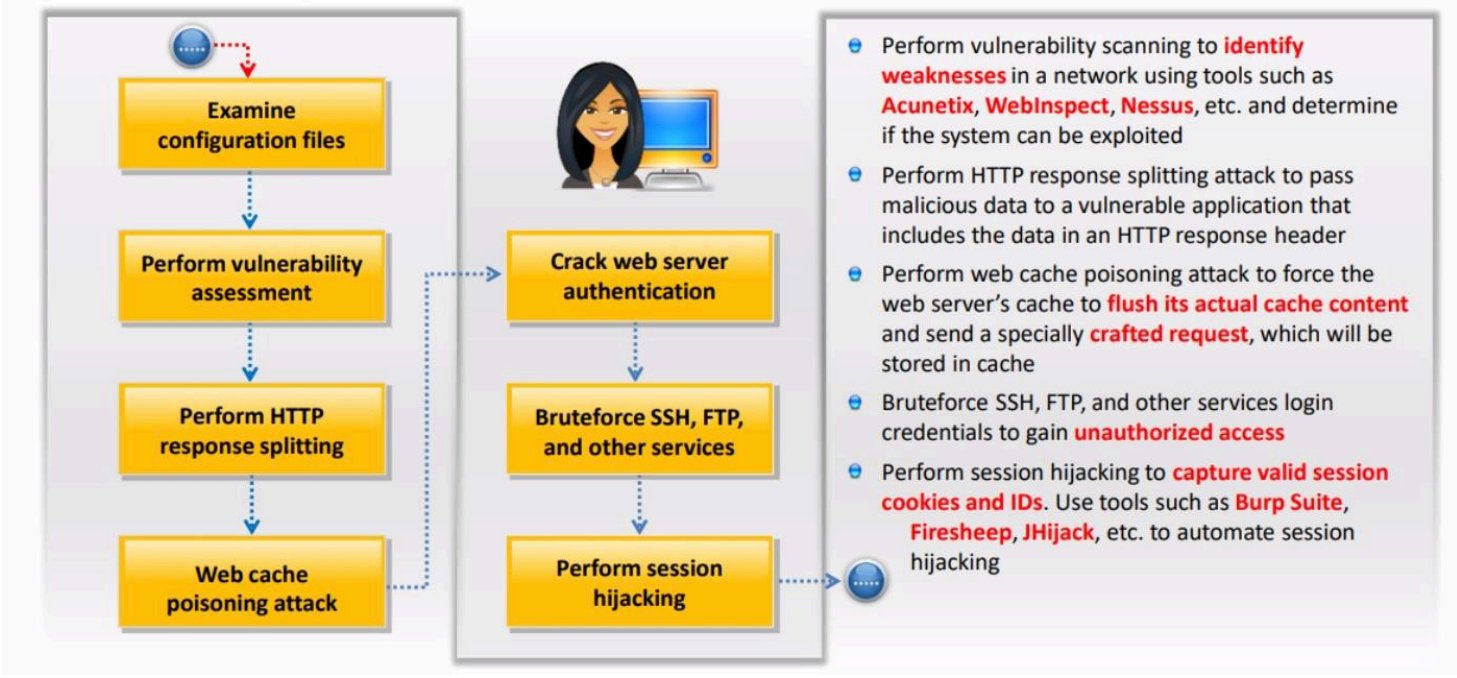
Slides 37, 38

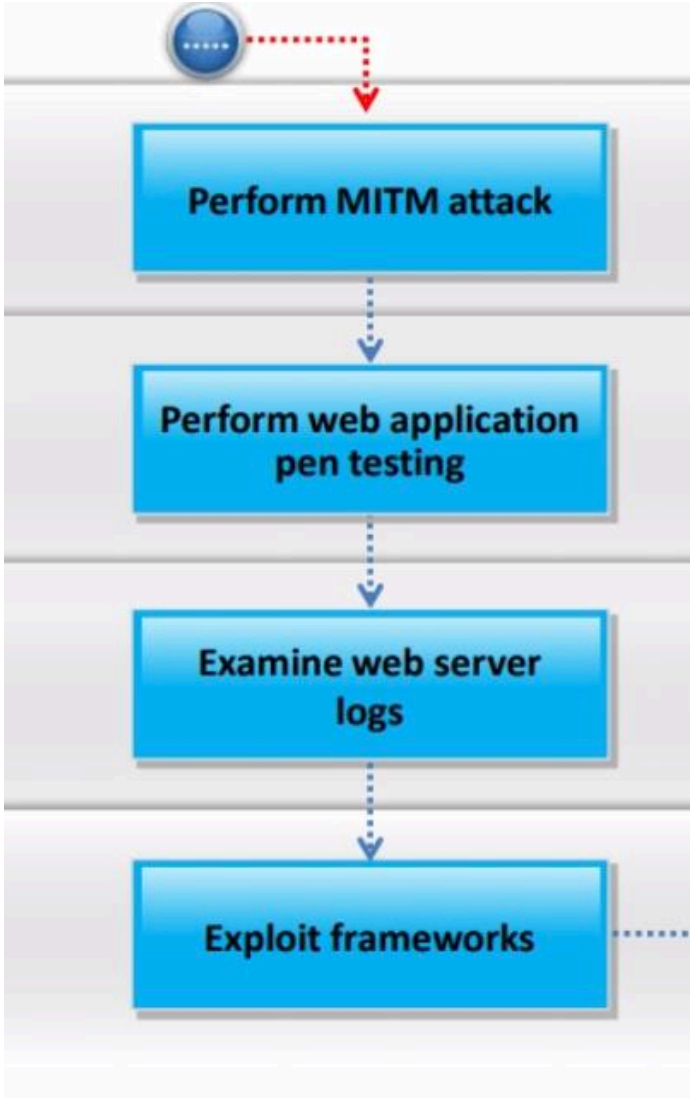
Redo DB ch.



- Web server penetration testing starts with **collecting as much information** as possible about an organization ranging from its physical location to operating environment
- Use **social engineering techniques** to collect information such as human resources, contact details, etc. that may help in **web server authentication testing**
- Use **Whois database query tools** to get the details about the target such as domain name, IP address, administrative contacts, Autonomous System Number, DNS, etc.
- **Note**: Refer Module 02: Footprinting and Reconnaissance for more information gathering techniques


Fingerprint web server
Use tools such as Netcraft and httprecon
Fingerprint web server to gather information such as server name, server type, operating systems, applications running, etc. using tools such as Netcraft, httprecon, and ID Serve
Crawl website
Use tools such as HTTrack, WebCopier Pro
Crawl website to gather specific types of information from web pages, such as email addresses
Enumerate web directories
Use tools such as Nmap
Enumerate web server directories to extract important information such as web functionalities, login forms, etc.
Perform directory traversal attack
Use automated tools such as DirBuster
Perform directory traversal attack to access restricted directories and execute commands outside of the web server's root directory


Examine configuration files
Perform vulnerability assessment
Perform HTTP response splitting
Web cache poisoning attack
Crack web server authentication
Bruteforce SSH, FTP, and other services
Perform session hijacking
Perform vulnerability scanning to identify weaknesses in a network using tools such as Acunetix, WebInspect, Nessus, etc. and determine if the system can be exploited
Perform HTTP response splitting attack to pass malicious data to a vulnerable application that includes the data in an HTTP response header
Perform web cache poisoning attack to force the web server's cache to flush its actual cache content and send a specially crafted request, which will be stored in cache
Bruteforce SSH, FTP, and other services login credentials to gain unauthorized access
Perform session hijacking to capture valid session cookies and IDs. Use tools such as Burp Suite, Firesheep, JHijack, etc. to automate session hijacking


Perform MITM attack
Perform web application pen testing
Examine web server logs
Exploit frameworks

Summary

- Web server vulnerabilities, tools, and exploits
- Web application vulnerabilities, tools, and exploits
- Database attacks and attack tools