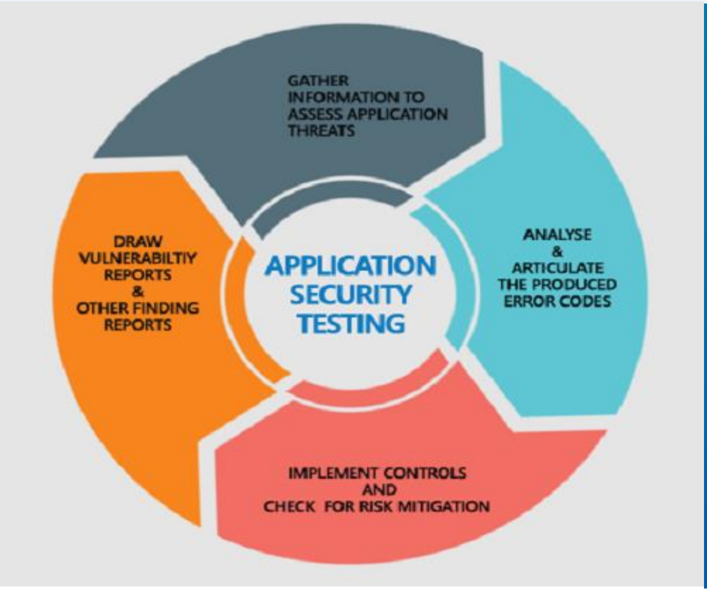
Web Application Security


APPLICATION SECURITY TESTING
GATHER INFORMATION TO ASSESS APPLICATION THREATS
ANALYSE & ARTICULATE THE PRODUCED ERROR CODES
IMPLEMENT CONTROLS AND CHECK FOR RISK MITIGATION
DRAW VULNERABILTIY REPORTS & OTHER FINDING REPORTS

E.R. Ramesh, M.C.A., M.Sc., M.B.A.,

Unit - 1

Introduction to Web Application Security

- Introduction
- Web Application Architecture
- Methodology of Web Hacking
- Profiling
- Hacking Web Servers
- Surveying the Application

Web Application

- A **web application** is a **software** or **program** which is **accessible using any web browser**. Its **frontend** is usually created using languages like **HTML, CSS, Javascript**, which are supported by major browsers.

- While the **backend** could use **any programming stack** like **LAMP, MEAN**, etc. Unlike mobile apps, **there is no specific SDK for developing web applications**.

- Web Applications **came to prominence with the advent of Software as a Service (SaaS) movement**.

Working of Web Application

- Web applications are **usually coded in browser-supported language such as JavaScript and HTML** as these languages **rely on the browser** to render the program executable.

- **Some of the applications are dynamic**, requiring **server-side processing**. Others are completely **static with no processing required at the server.**

- The **web application requires a web server to manage requests** from the client, an **application server to perform the tasks requested**, and, sometimes, **a database to store the information.**

Working of Web Application

Here's what a typical **web application flow** looks like:

1. User triggers a request to the web server over the Internet, either through a web browser or the application's user interface
2. Web server forwards this request to the appropriate web application server
3. Web application server performs the requested task – such as querying the database or processing the data – then generates the results of the requested data
4. Web application server sends results to the web server with the requested information or processed data
5. Web server responds back to the client with the requested information that then appears on the user's display

Example of a Web Application

- Web applications **include online forms**, **shopping carts**, **word processors**, **spreadsheets**, **video and photo editing**, **file conversion**, **file scanning**, and **email programs** such as **Gmail, Yahoo and AOL.**

- Popular applications include **Google Apps** and **Microsoft 365**. Google Apps Work has **Gmail, Google Docs, Google Sheets, Google Slides, online storage** and more.

- Other functionalities include **online sharing of documents and calendars**. This lets **all team members access the same version of a document simultaneously**.

Benefits of a Web Application

- Web applications **run on multiple platforms** regardless of OS or device **as long as the browser is compatible**.
- All **users access the same version, eliminating** any **compatibility issues**.
- They are **not installed on the hard drive**, thus **eliminating space limitations.**
- They **reduce software piracy in subscription-based** web applications (i.e. SaaS).
- They **reduce costs for both the business** and **end user** as there is **less support and maintenance required** by the business and lower requirements for the end user's computer.

Technology used to build a Website

- **Java web application architecture** makes it **possible to combine different Java tools** (continuous integration, version control, etc.) and **frameworks** (development and testing) **to build apps** regardless of their complexity.

- **Cloud-based architecture** involves **storing all data** and **functions on the cloud** or **local servers**, which forms an environment where **systems can interact with each other while not directly connected.**

Technology used to build a Website

- **RabbitMQ** is a widely-used **message broker** that **acts as a middleman between applications**, storing **queued messages until the receiving app can access them**. It can be used in **transactional systems**, where you **want things done in a certain order.**

- **.NET** web server architecture **can handle cross-platform software**, **Docker, microservices**, and **execution multiple app versions on the same computer**. It **doesn’t demand source code for storing information** and **enables optimizing and improving the feature set** and development thanks to ASP.NET Core and .NET Core.

Technology used to build a Website

- **PHP** : The tools and features offered by **PHP frameworks allow for less code and guarantee strong protection, prompt development, simple operation**, and **enhanced support from the developer community.**
- **Angular** is a framework that brings a variety of advantages, **including UX with lazy loading effect added** to **minimize code size.**
- **Python** features **compressed, legible, and supported code**. Python **contributes to the increased web application speed and is exempt from app maintenance**. Additionally, it is extremely limber and well-integrated with other languages.

Technology used to build a Website

- **Node.js** framework creates competition for many other computer languages. Developers **use it for coding services and UIs due to its consistency**, **codesharing and reusing**, and hassle-free knowledge exchange. It offers many free tools and **enables integrating multiple systems and services via a single UI.** Thus, you can expect more efficiency and a quicker design process.

- **Ruby on Rails**. Relying on the “Convention over Configuration” principle (where convention means hypotheses), this platform contributes to effective and fast web design. Conventions are used for decision making.

Web Application Architecture

- A **web application is a program that uses an Internet browser to perform certain functions**. It **comes with middleware and UIs that connect both the client** (what a user sees and utilizes in the browser), server (the backend of operations), and database(s).

- While **backend scripting keeps data**, the **frontend transfers that data to the consumers supporting data exchange**.

- In simple terms, **web app architecture is how the system works during your everyday browsing — entering a web address, viewing the site, and interacting with it** — whereas the **browser conveys data to the server**.

Web Application Architecture

A well-formed architecture:

- Maintains visual component.
- Guarantees prompt UI.
- Ensures security.
- Enables stability and self-regulation.
- Scales and records bugs in the easiest way.

Web Application Components

UI/UX Web Application Components

- This includes activity logs, dashboards, notifications, settings, statistics, etc. These components have nothing to do with the operation of web application architecture. Instead, they are part of the interface layout plan of a web app.

Structural Components

- The two major structural components of a web app are client and server sides.

Web Application Components

Client Component

- The client component is developed in CSS, HTML, and JS. As it exists within the user's web browser, there is no need for operating system or device related adjustments.

- The client component is a representation of a web application's functionality that the end-user interacts with.

Web Application Components

Server Component

- The server component can be build using one or a combination of several programming languages and frameworks, including Java, .Net, NodeJS, PHP, Python, and Ruby on Rails.
- The server component has at least two parts; app logic and database. The former is the main control centre of the web application while the latter is where all the persistent data is stored.

Models of Web Application Components

- Depending on the total number of servers and databases used for a web application, the model of a web app is decided. It can be any of the following three:

One Web Server, One Database

- It is the most simple as well as the least reliable web app component model. Such a model uses a single server as well as a single database. A web app builds on such a model will go down as soon as the server goes down. Hence, it isn't much reliable.

- One web server, one database web application component model is not typically used for real web applications.

Models of Web Application Components

- It is mostly used for running test projects as well as with the intent of learning and understanding the fundamentals of the web application.

Models of Web Application Components

Multiple Web Servers, One Database (At a Machine Rather than the Web server)

- The idea with this type of web application component model is that the web server doesn’t store any data. When the web server gets information from a client, it processes the same and then writes it to the database, which is managed outside of the server. This is sometimes also referred to as a stateless architecture.

- At least 2 web servers are required for this web application component model. This is all for avoiding failure. Even when one of the web servers goes down, the other one will take charge.

Models of Web Application Components

- All requests made will be redirected automatically to the new server and the web app will continue execution. Hence, reliability is better as compared to the single server with inherent database model. However, if the database crashes the web app will follow to do the same.

Models of Web Application Components

Multiple Web Server, Multiple Databases

- It is the most efficient web application component model because neither the web servers nor the databases have a single point of failure.

- There are two options for this type of model. Either to store identical data in all the employed databases or distribute it evenly among them.

- Not more than 2 databases are required typically for the former case, while for the latter case some data might become unavailable in the scenario of a database crash.

Models of Web Application Components

- DBMS normalization is used, however, in both scenarios. When the scale is large i.e. more than 5 web servers or databases or both, it is advised to install **load balancers**.

Types of Web Application Architecture

- Web application architecture is a pattern of interaction between various web application components. The type of web application architecture depends on how the application logic is distributed among the client and server sides.

- There are three primary types of web application architecture. Each one of them is explained as follows:

- **Single-Page Applications** (SPAs) – Instead of loading completely new pages from the server each time for a user action, single page web applications allows for a dynamic interaction by means of providing updated content to the current page.

Types of Web Application Architecture

- AJAX, a concise form of Asynchronous JavaScript and XML, is the foundation for enabling page communications and hence, making SPAs a reality.

- Because single-page applications prevent interruptions in user experience, they, in a way, resemble traditional desktop applications. SPAs are designed in a way so that they request for most necessary content and information elements.

- This leads to the procurement of an intuitive as well as interactive user experience.

Types of Web Application Architecture

- **Microservices** – These are small and lightweight services that execute a single functionality. The Microservices Architecture framework has a number of advantages that allows developers to not only enhance productivity but also speed up the entire deployment process.

- The components making up an application build using the Microservices Architecture aren't directly dependent on each other. As such, they don't necessitate to be built using the same programming language.

- Hence, developers working with the Microservices Architecture are free to pick up a technology stack of choice. It makes developing the application simpler and quicker.

Types of Web Application Architecture

- **Serverless Architectures** – In this type of web application architecture, an application developer consults a third-party cloud infrastructure services provider for outsourcing server as well as infrastructure management.

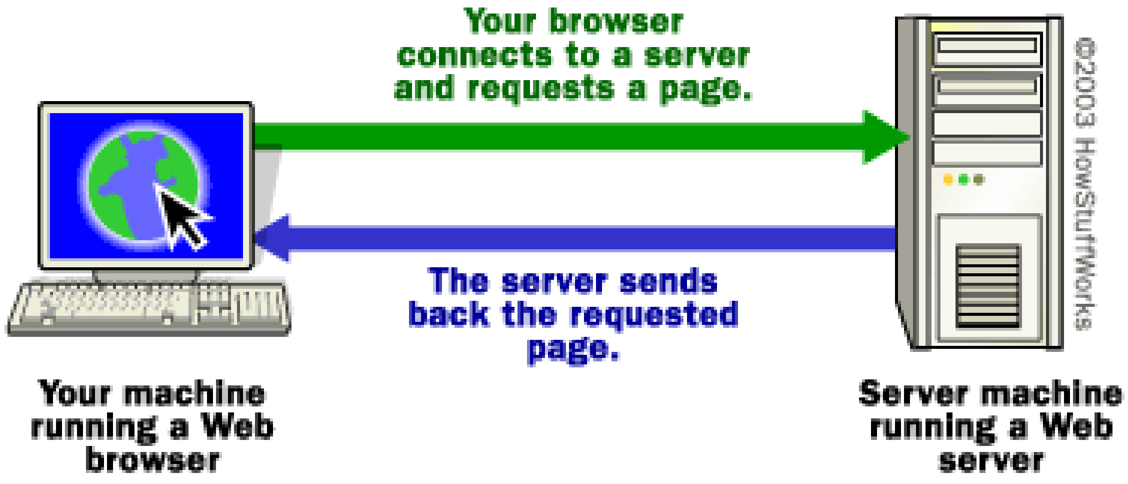
- The benefit of this approach is that it allows applications to execute the code logic without bothering with the infrastructure related tasks.

- The Serverless Architecture is best when the development company doesn’t want to manage or support the servers as well as the hardware they have developed the web application for.

Web Servers

1. The browser breaks the URL into three parts:
   1. The protocol ("http")
   2. The server name ("www.website.com")
   3. The file name ("webpage.html")
2. The browser communicates with a name server, which translates the server name, www.website.com, into an IP address
3. The browser then forms a connection to the Web server at that IP address on port 80.
4. Following the HTTP protocol, the browser sends a GET request to the server, asking for the file http://webpage.html.
5. The server sends the HTML text for the Web page to the browser.
6. The browser reads the HTML tags and formats the page onto the screen.

How Web Servers work?


Your browser connects to a server and requests a page.
The server sends back the requested page.
Your machine running a Web browser
Server machine running a Web server
©2003 HowStuffWorks

- You type URL into your browser and press return.
- And magically, no matter where in the world that URL lives, the page pops up on your screen.
- Your browser formed a connection to a Web server, requested a page and received it.

How Web Servers work?

- The browser broke the URL into three parts:
  - The protocol ("http")
  - The server name ("www.howstuffworks.com")
  - The file name ("web-server.htm")
- The browser communicated with a name server to translate the server name "www.howstuffworks.com" into an **IP Address**, which it uses to connect to the server machine.
- The browser then formed a connection to the server at that IP address on port 80.

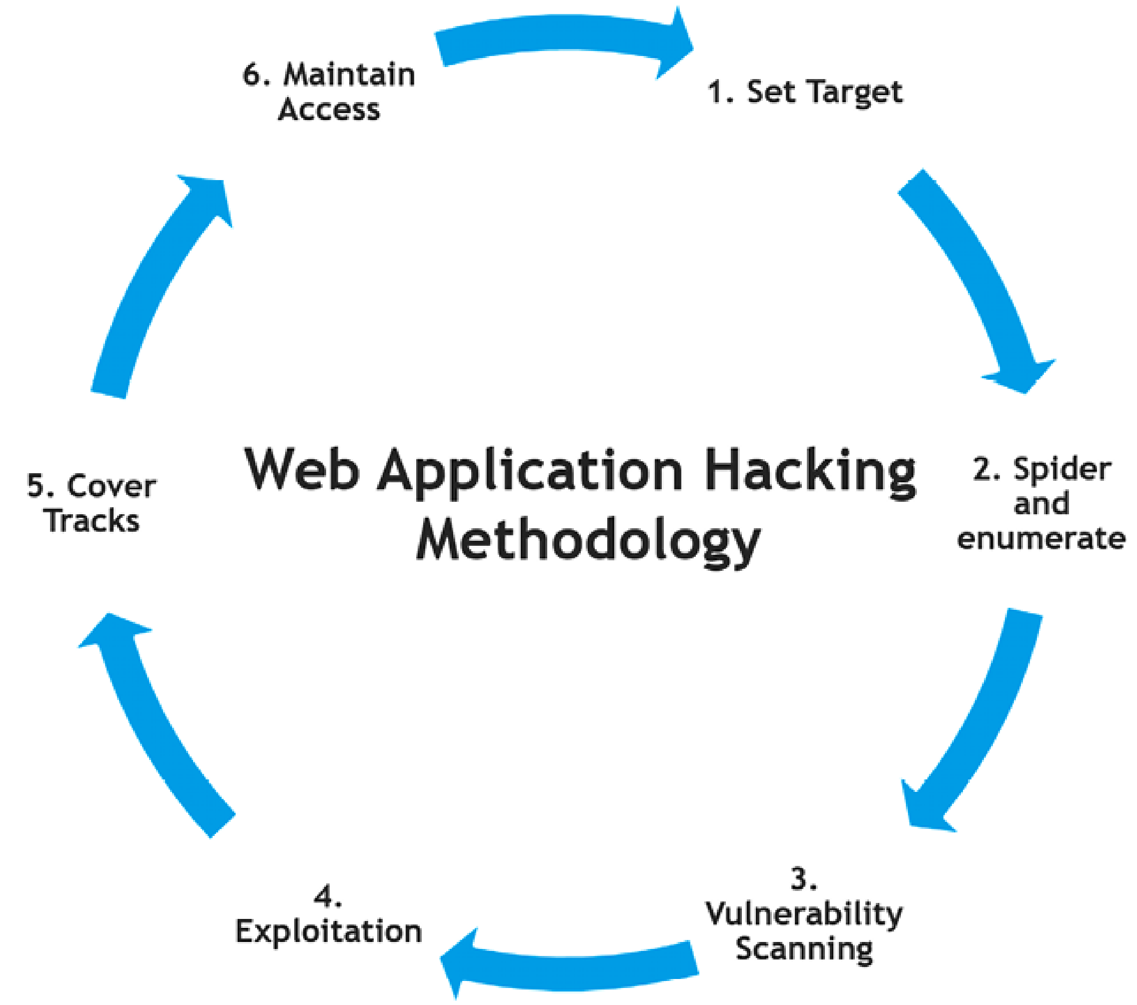
How Web Servers work?

- Following the HTTP protocol, the browser sent a GET request to the server, asking for the file https://www.howstuffworks.com/web-server.htm.
- Note that **cookies** may be sent from browser to server with the GET request.
- The server then sent the HTML text for the Web page to the browser. (Cookies may also be sent from server to browser in the header for the page.) The browser read the HTML tags and formatted the page onto your screen.

Popular Web Servers and Common Security Threats

- Apache Web Server
- IIS Web Server
- Sun ONE Web Server
- Nature of Security Threats in a Web Server Environment.
  - Bugs or Web Server Misconfiguration.
  - Browser-Side or Client Side Risks.
  - Sniffing
  - Denial of Service Attack.

Methodology of Web Hacking



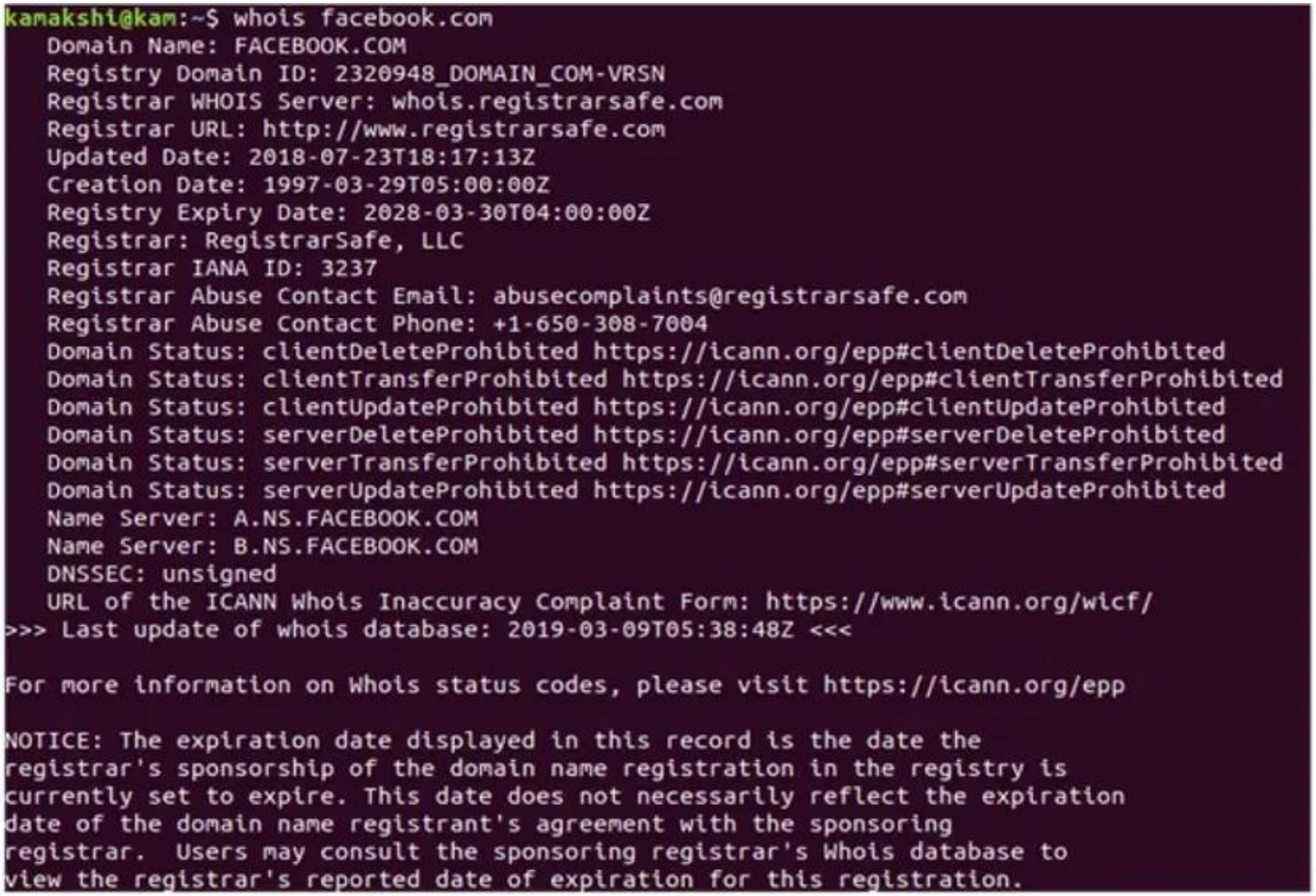
Methodology of Web Hacking

Hacking Methodology

Information Gathering

- Information Gathering is a process of gathering different information about the victim/target by using various platforms such as Social engineering, internet surfing, etc.

Methodology of Web Hacking



```
kamakshi@kam:~$ whois facebook.com
   Domain Name: FACEBOOK.COM
   Registry Domain ID: 2320948_DOMAIN_COM-VRSN
   Registrar WHOIS Server: whois.registrarsafe.com
   Registrar URL: http://www.registrarsafe.com
   Updated Date: 2018-07-23T18:17:13Z
   Creation Date: 1997-03-29T05:00:00Z
   Registry Expiry Date: 2028-03-30T04:00:00Z
   Registrar: RegistrarSafe, LLC
   Registrar IANA ID: 3237
   Registrar Abuse Contact Email: abusecomplaints@registrarsafe.com
   Registrar Abuse Contact Phone: +1-650-308-7004
   Domain Status: clientDeleteProhibited https://icann.org/epp#clientDeleteProhibited
   Domain Status: clientTransferProhibited https://icann.org/epp#clientTransferProhibited
   Domain Status: clientUpdateProhibited https://icann.org/epp#clientUpdateProhibited
   Domain Status: serverDeleteProhibited https://icann.org/epp#serverDeleteProhibited
   Domain Status: serverTransferProhibited https://icann.org/epp#serverTransferProhibited
   Domain Status: serverUpdateProhibited https://icann.org/epp#serverUpdateProhibited
   Name Server: A.NS.FACEBOOK.COM
   Name Server: B.NS.FACEBOOK.COM
   DNSSEC: unsigned
   URL of the ICANN Whois Inaccuracy Complaint Form: https://www.icann.org/wicf/
>>> Last update of whois database: 2019-03-09T05:38:48Z <<<

For more information on Whois status codes, please visit https://icann.org/epp

NOTICE: The expiration date displayed in this record is the date the
registrar's sponsorship of the domain name registration in the registry is
currently set to expire. This date does not necessarily reflect the expiration
date of the domain name registrant's agreement with the sponsoring
registrar.  Users may consult the sponsoring registrar's Whois database to
view the registrar's reported date of expiration for this registration.
```

Methodology of Web Hacking

Footprinting

- Footprinting is a crucial phase where an attacker may use different tools to gather information about the target. In this phase, an attacker uses passive methods to find information about the victim before performing an attack. The attacker keeps minimum interactions with the victim to avoid detection and alerting the target of the attack.

- Footprinting can quickly reveal the vulnerabilities of the target system and can exploit them. There are various methods to gather information such as Whois, Google Searching, Operating system detection, network enumeration, etc.

Methodology of Web Hacking

Web Server Footprinting

- In webserver footprinting, information is gathered using some specific tools that are focused on web servers such as Maltego, httprecon, Nessus, etc. resulting in details like operating system, running services, type, applications, etc.

Methodology of Web Hacking

1. Vulnerability Scanning

- Vulnerability scanning is the next process taken after performing footprinting to precisely target the attack.
- A vulnerability scanner is a computer program made to discover system weaknesses in computers and networks. Some methods used in vulnerability scanning are port scanning, OS detection, network services, etc.
- Common tools used for scanning are Nmap, Nikto, Nessus, and many more.

Methodology of Web Hacking

Different Types of Vulnerability Scanning

- Vulnerability Scanning is classified into two types:
  - unauthenticated and authenticated scans.

Authenticated Scan: In this, the tester logs in as a network user and finds the vulnerabilities that a regular user can encounter. He also checks all the possible attacks by which a hacker can take benefit.

Unauthenticated Scan: In this, the tester performs all the scans that a hacker would likely do, avoiding direct access to the network. These points can reveal how to get access to a network without signing in.

Methodology of Web Hacking

2. Session Hijacking

- Session Hijacking/ cookie hijacking is an exploitation of the web session. In this attack, the attacker takes over the users' sessions to gain unauthorized access to get information about its services. Session hijacking mostly applies to web applications and browser sessions.

- The attacker needs to know the Session-Id (session key) to perform session hijacking successfully. It can be obtained by stealing the session or just by clicking on some malicious links provided by the attacker. Once the attacker gets the key, he can take over the session using just the same session key, and the server will now treat the attacker's connection as the initial session.

Methodology of Web Hacking

3. Password Attacks

Password cracking is a method of extracting passwords to gain authorized access to the legitimate user's target system. Password cracking can be performed using social engineering attack, dictionary attack, or password guessing or stealing the stored information that can help obtain passwords that give access to the system.

Password Attacks are classified as:

- Non-Electronic Attack
- Active Online Attack
- Passive Online Attack
- Default Passwords
- Offline Attack

Hacking Methodology

Web Application Hacking Methodology provides attackers with steps to follow to execute a successful attack.

These steps are:

Web Infrastructure Footprinting

Footprinting web infrastructure helps attacker gather information about the target web infrastructure and identify vulnerabilities that can be exploited.

Profiling

What is footprinting and how does it work?

- Footprinting or profiling is an ethical hacking technique used to gather as much data as possible about a specific targeted computer system, an infrastructure and networks to identify opportunities to penetrate them. It is one of the best methods of finding vulnerabilities.

- The process of cybersecurity footprinting involves **profiling organizations** and collecting data about the network, host, employees and third-party partners. This information includes the OS used by the organization, firewalls, network maps, IP addresses, domain name system information, security configurations of the target machine, URLs, virtual private networks, staff IDs, email addresses and phone numbers.

Profiling

- There are two types of footprinting in ethical hacking:

1.active footprinting
2.passive footprinting

What is active footprinting?

Active footprinting describes the process of using tools and techniques, like using the traceroute commands or a ping sweep -- Internet Control Message Protocol sweep -- to collect data about a specific target.

This often triggers the target's intrusion detection system (IDS). It takes a certain level of stealth and creativity to evade detection successfully.

Profiling

How do you start footprinting?

- Reconnaissance is similar to footprinting and is a crucial part of the initial hacking exercise.
- It is a passive footprinting exercise where one collects data about the target's potential vulnerabilities and flaws to exploit while penetration testing.

Profiling

What is passive footprinting?

- As the name implies, passive footprinting involves collecting data about a specific target using innocuous methods, like performing a Google search, looking through Archive.org, using NeoTrace, browsing through employees' social media profiles, looking at job sites and using Whois, a website that provides the domain names and associated networks fora specific organization.

- It is a stealthier approach to footprinting because it does not trigger the target's IDS.

Profiling

- Footprinting also known as profiling, processes start with determining the location and objective of an intrusion. Once ethical hackers identify a specific target, they gather information about the organization using nonintrusive methods, such as accessing the organization's own webpage, personnel directory or employee bios.

- Ethical hackers collect this information and initiate social engineering campaigns to identify security vulnerabilities and achieve ethical hacking goals.

Profiling

Security audits

- Highly structured
- Policy vs. reality
- Business process reviews
- Determines whether controls exist
- References laws/security standards
- High-level tools are often used

Vulnerability assessments

- In-depth view
- Looks at technical flaws
- Scope is often external *and* internal systems
- Relies heavily on lots of good tools
- Typically doesn't include exploitation of weaknesses found
- Often confused with vulnerability "scans"

Penetration testing

- Less structured
- Tightly-defined scope, typically external systems
- Sometimes operational flaws (i.e. social engineering) are included
- Time sensitive
- Relies heavily on limited set of good tools

Hacking Methodology

In this process, the attacker performs:

- Server discovery to learn about the servers that host the application
- Service discovery to determine which service can be attacked
- Server identification to learn information about the server such as version and make
- Hidden content discovery to discover hidden contents

Hacking Methodology

Web server attack

- The information gathered in the footprinting step allows hackers to analyze it, find vulnerabilities to exploit, and use various techniques to launch attacks on the server.

Hacking Methodology

Web application analysis

Attackers analyze target web application to identify its vulnerabilities and exploit them.

To hack the application, attackers need to:

- Identify entry points for user input
- Identify server-side technologies used for generating dynamic web pages
- Identify server-side functionality
- Identify attack areas and associated vulnerabilities

Hacking Methodology

Client-side Controls Evasion

Attackers attempt to bypass client-side control of user inputs and interaction.

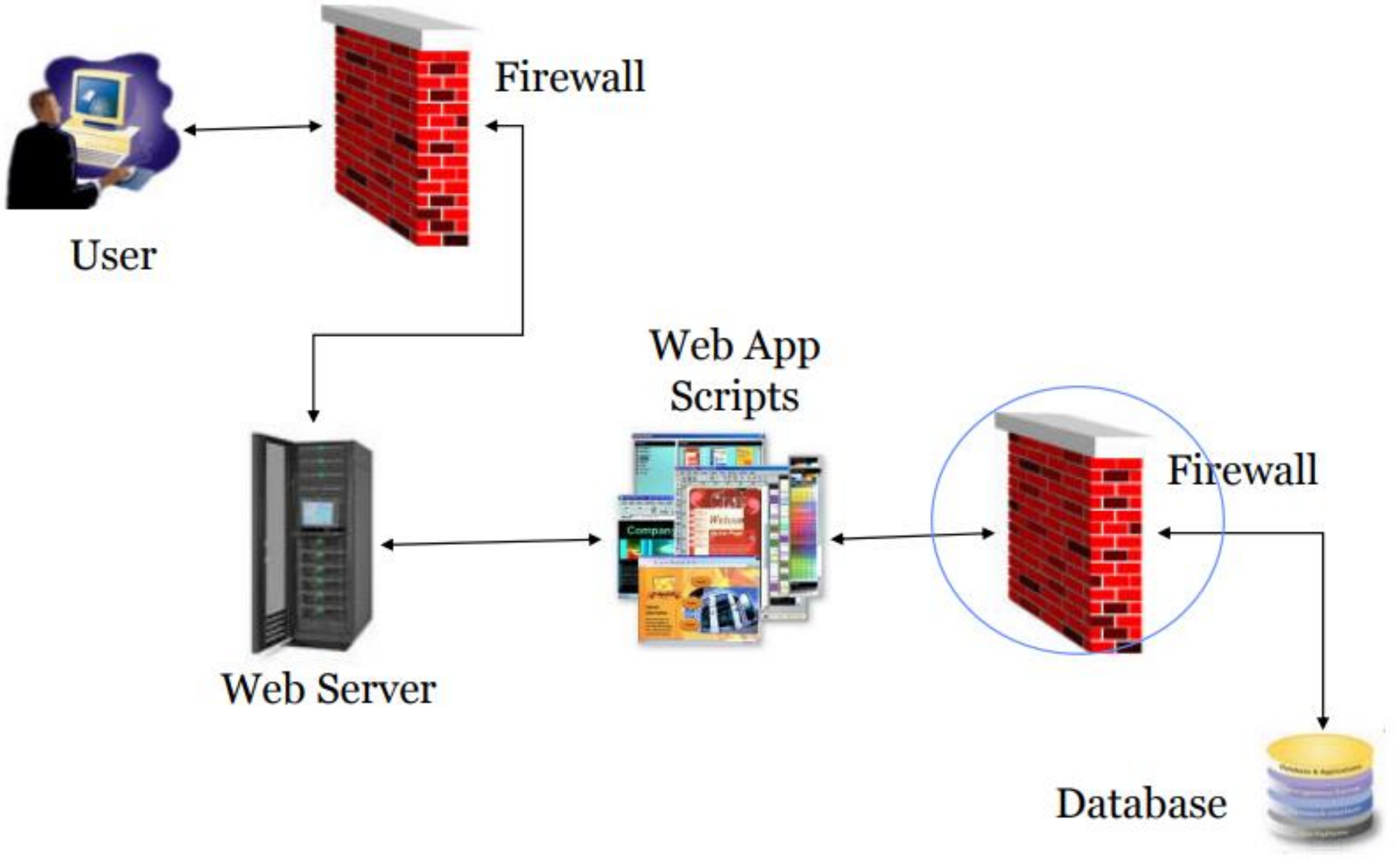
To bypass the client-side controls, attackers attempt to:

- Attack hidden form fields
- Attack browser extensions
- Review the source code

Q & A

E.R. Ramesh, M.C.A., M.Sc., M.B.A.,
98410 59353, 98403 50547
rameshvani@gmail.com

Web Application Security


Firewall
User
Web App Scripts
Firewall
Web Server
Database