Ethical Hacking

Workshop - Day 1

- Aazar | 20th January 2024

“I went to prison for my hacking. Now people hire me to do the same things I went to prison for, but in a legal and beneficial way.”

- Kevin D. Mitnick, Ghost in the Wires: My Adventures as the World's Most Wanted Hacker

What’s in it for me?

What to expect from this workshop.

- Comprehensive Understanding of Cybersecurity
- Epic Show-and-Tell Skills
- Hands-On Ethical Hacking Experience
- A Certificate, of course
- 10 points in Defence Against the Dark Arts
- $$$$$$$$$

The Why!?


Estimated Cost of Cyber Crime Worldwide
Trillion USD
30
24
18
12
6
2018
2019
2020
2021
2022
2023
2024
2025
2026
2027
As of November 2022. Sources: Statista Technology

The OWASP Top 10


OWASP TOP 10
Broken Access Control
Cryptographic Failures
Injection
Insecure Design
Security Misconfiguration
Vulnerable and Outdated Components
Identification and Authentication Failures
Software and Data Integrity Failures
Security Logging and Monitoring Failures
Server-Side Request Forgery

Broken Access Control

Description:

Access control enforces policy such that users cannot act outside of their intended permissions. Failures typically lead to unauthorised information disclosure, modification, or destruction of all data or performing a business function outside the user's limits.

Example Attack Scenario:

An attacker simply forces browses to target URLs. Admin rights are required for access to the admin page.

https://example.com/app/getappInfo

https://example.com/app/admin_getappInfo

If an unauthenticated user can access either page, it's a flaw. If a non-admin can access the admin page, this is a flaw.

Cryptographic Failures

Description:

The first thing is to determine the protection needs of data in transit and at rest. For example, passwords, credit card numbers, health records, personal information, and business secrets require extra protection, mainly if that data falls under privacy laws.

Example Attack Scenario:

A site doesn't use or enforce TLS for all pages or supports weak encryption. An attacker monitors network traffic (e.g., at an insecure wireless network), downgrades connections from HTTPS to HTTP, intercepts requests, and steals the user's session cookie. The attacker then replays this cookie and hijacks the user's (authenticated) session, accessing or modifying the user's private data. Instead of the above they could alter all transported data, e.g., the recipient of a money transfer.

Injection

Description:

An application is vulnerable to attack when:

- User-supplied data is not validated, filtered, or sanitised by the application.

- Dynamic queries or non-parameterised calls without context-aware escaping are used directly in the interpreter.

- Hostile data is used within object-relational mapping (ORM) search parameters to extract additional, sensitive records.

- Hostile data is directly used or concatenated. The SQL or command contains the structure and malicious data in dynamic queries, commands, or stored procedures.

Example Attack Scenario:

An application uses untrusted data in the construction of the following vulnerable SQL call:

```
String query = "SELECT \* FROM accounts WHERE custID='" + request.getParameter(“id”) + “‘“;
```

Insecure Design

Description:

Insecure design in software development refers to weaknesses in the overall architectural design, characterised by ineffective or missing security controls. It differs from implementation defects and implies that necessary security measures were not originally incorporated.

Example Attack Scenario:

A credential recovery workflow might include “questions and answers,” which is prohibited by NIST 800-63b, the OWASP ASVS, and the OWASP Top 10. Questions and answers cannot be trusted as evidence of identity as more than one person can know the answers, which is why they are prohibited. Such code should be removed and replaced with a more secure design.

Security Misconfiguration

Description:

Security misconfiguration refers to vulnerabilities in a system or application arising from improperly configured security settings, permissions, or features. This can include missing security hardening, enabling unnecessary features, unchanged default credentials, insecure error handling, and outdated software, posing a risk to the overall security of the system.

Example Attack Scenario:

The application server comes with sample applications not removed from the production server. These sample applications have known security flaws attackers use to compromise the server. Suppose one of these applications is the admin console, and default accounts weren't changed. In that case, the attacker logs in with default passwords and takes over.

Vulnerable and Outdated Components

Description:

The security risk associated with using software components, both client-side and server-side, that are either outdated, unsupported, or have unknown versions. This vulnerability is heightened when there's a lack of regular vulnerability scanning, failure to subscribe to security bulletins, delays in fixing or upgrading components, and inadequate testing for compatibility with updated libraries.

Example Attack Scenario:

Components typically run with the same privileges as the application itself, so flaws in any component can result in serious impact. Such flaws can be accidental (e.g., coding error) or intentional (e.g., a backdoor in a component). Some example exploitable component vulnerabilities discovered are:

CVE-2017-5638, a Struts 2 remote code execution vulnerability that enables the execution of arbitrary code on the server, has been blamed for significant breaches.

Identification and Authentication Failures

Description:

"Identification and Authentication Failures" involve vulnerabilities in confirming user identity and managing sessions. This can include weaknesses like automated attacks, use of weak passwords, insecure recovery processes, inadequate multi-factor authentication, and session-related issues like exposing or reusing identifiers.

Example Attack Scenario:

Credential stuffing, the use of lists of known passwords, is a common attack. Suppose an application does not implement automated threat or credential stuffing protection. In that case, the application can be used as a password oracle to determine if the credentials are valid.

Software and Data Integrity Failures

Description:

"Software and Data Integrity Failures" involve vulnerabilities in code and infrastructure that do not adequately safeguard against integrity violations. This can include relying on untrusted sources for plugins or libraries, insecure CI/CD pipelines, and vulnerabilities related to auto-update mechanisms or insecure deserialisation.

Example Attack Scenario:

A React application calls a set of Spring Boot microservices. Being functional programmers, they tried to ensure that their code is immutable. The solution they came up with is serialising the user state and passing it back and forth with each request. An attacker notices the "rO0" Java object signature (in base64) and uses the Java Serial Killer tool to gain remote code execution on the application server.

Security Logging and Monitoring Failures

Description:

"Security Logging and Monitoring Failures" involve vulnerabilities that impede the detection, escalation, and response to active breaches. This includes issues such as inadequate logging of auditable events, unclear log messages for warnings and errors, insufficient monitoring for suspicious activity, local storage of logs, and absence of effective alerting and response processes.

Example Attack Scenario:

A children's health plan provider's website operator couldn't detect a breach due to a lack of monitoring and logging. An external party informed the health plan provider that an attacker had accessed and modified thousands of sensitive health records of more than 3.5 million children. A post-incident review found that the website developers had not addressed significant vulnerabilities. As there was no logging or monitoring of the system, the data breach could have been in progress since 2013, a period of more than seven years.

Server-Side Request Forgery (SSRF)

Description:

SSRF flaws occur whenever a web application is fetching a remote resource without validating the user-supplied URL. It allows an attacker to coerce the application to send a crafted request to an unexpected destination, even when protected by a firewall, VPN, or another type of network access control list (ACL).

Example Attack Scenario:

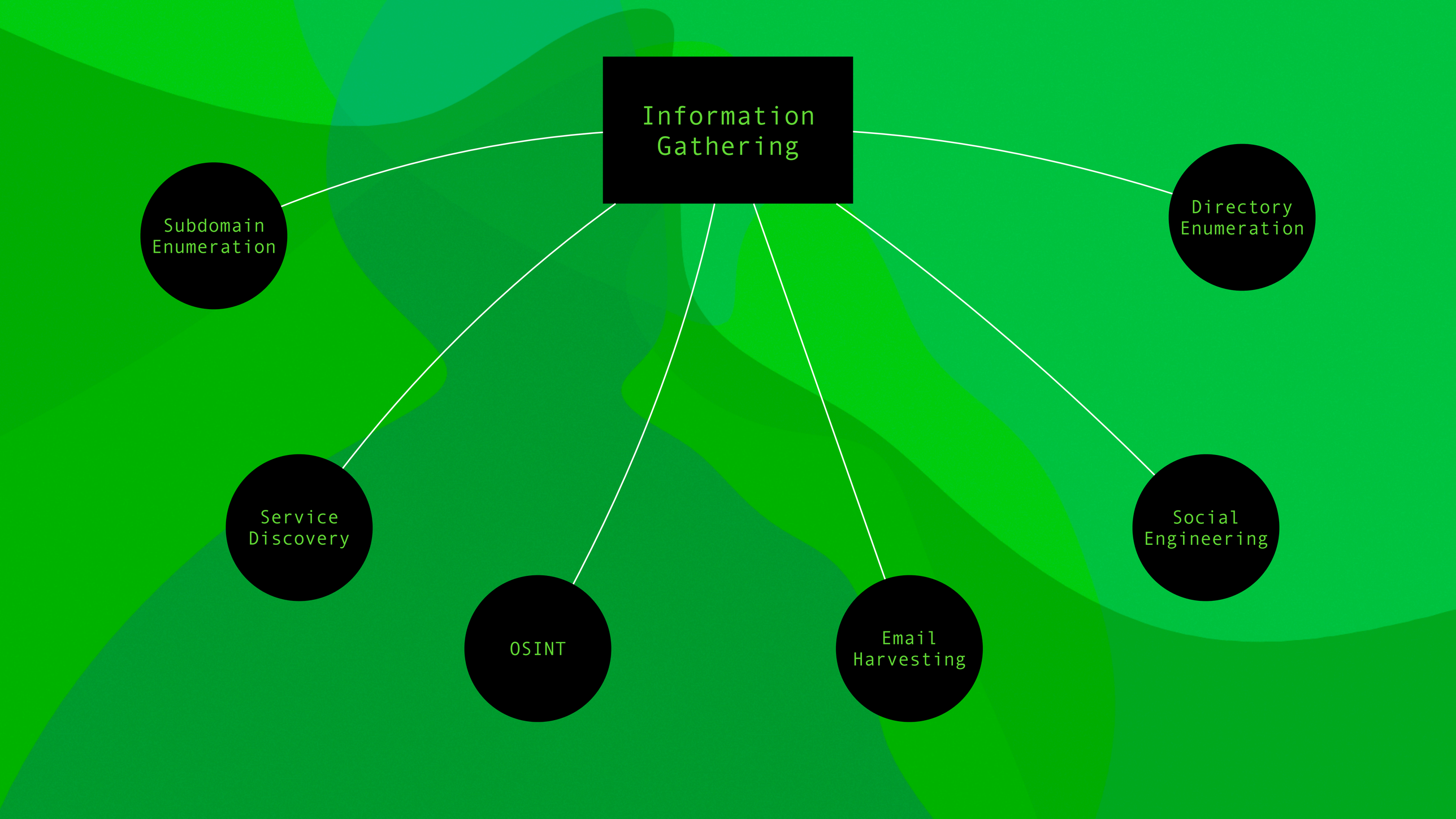
Port scan internal servers – If the network architecture is unsegmented, attackers can map out internal networks and determine if ports are open or closed on internal servers from connection results or elapsed time to connect or reject SSRF payload connections.

“The more you know about the enemy, the better you can fight them.”

- Sun Tzu, The Art of War

Know thy enemy

Information Gathering


Information Gathering
Subdomain Enumeration
Service Discovery
OSINT
Email Harvesting
Social Engineering
Directory Enumeration

Subdomain Enumeration

Purpose:

To discover subdomains of a target domain.

Tools & Techniques:

Tools like Subfinder, Amass; DNS recon tools.

Significance:

Uncovering subdomains helps in creating a more complete security profile of the target, identifying areas that may be less monitored or secured, and providing a comprehensive view of an organisation's online presence, which is crucial for both defensive and offensive cybersecurity strategies.

Example Command:

```
subfinder -d target.com
```

Service Discovery

Purpose:

Identifying open ports and the services running on them.

Tools & Techniques:

RustScan, followed by detailed scanning with Nmap.

Significance:

Service discovery is integral for understanding the network architecture of a target, including identifying potentially vulnerable services, outdated software, or misconfigurations. It lays the groundwork for deeper penetration testing and vulnerability assessment.

Example Command:

```
rustscan -a target.com —ulimit 10000 — -A
```

OSINT (Open Source Intelligence)

Purpose:

Gathering data from publicly available sources.

Tools & Techniques:

Search engines, social media platforms, WHOIS tools.

Significance:

OSINT allows for a comprehensive collection of publicly accessible information that can reveal critical insights into a target’s infrastructure, employee details, technology stack, and other operational details, which can be crucial for both defensive cybersecurity posture assessments and offensive planning.

Example Command:

whois “targetip"

Email Harvesting

Purpose:

Collecting email addresses associated with the target domain.

Tools & Techniques:

Email scraping tools, social media, WHOIS records.

Significance:

Collecting emails is not just about crafting phishing campaigns; it also helps in understanding the organisation's internal structure, employee roles, and potential data breaches through exposed email credentials, contributing significantly to the overall risk assessment.

Example Command:

```
theHarvester -d target.com -l 500 -b duckduckgo
```

Social Engineering

Purpose:

Manipulating individuals to divulge confidential information.

Tools & Techniques:

Pretexting, phishing emails, baiting.

Significance:

Targets the human element of security, often considered the weakest link in cybersecurity. Social engineering can lead to unauthorised access to sensitive information, manipulation of personnel for malicious purposes, and can reveal the effectiveness of an organisation's security training and awareness programs.

Example Technique:

Phishing email campaign mimicking internal IT communication.

Directory Enumeration

Purpose:

To uncover directories and files on a web server.

Tools & Techniques:

Tools like DirBuster, Gobuster.

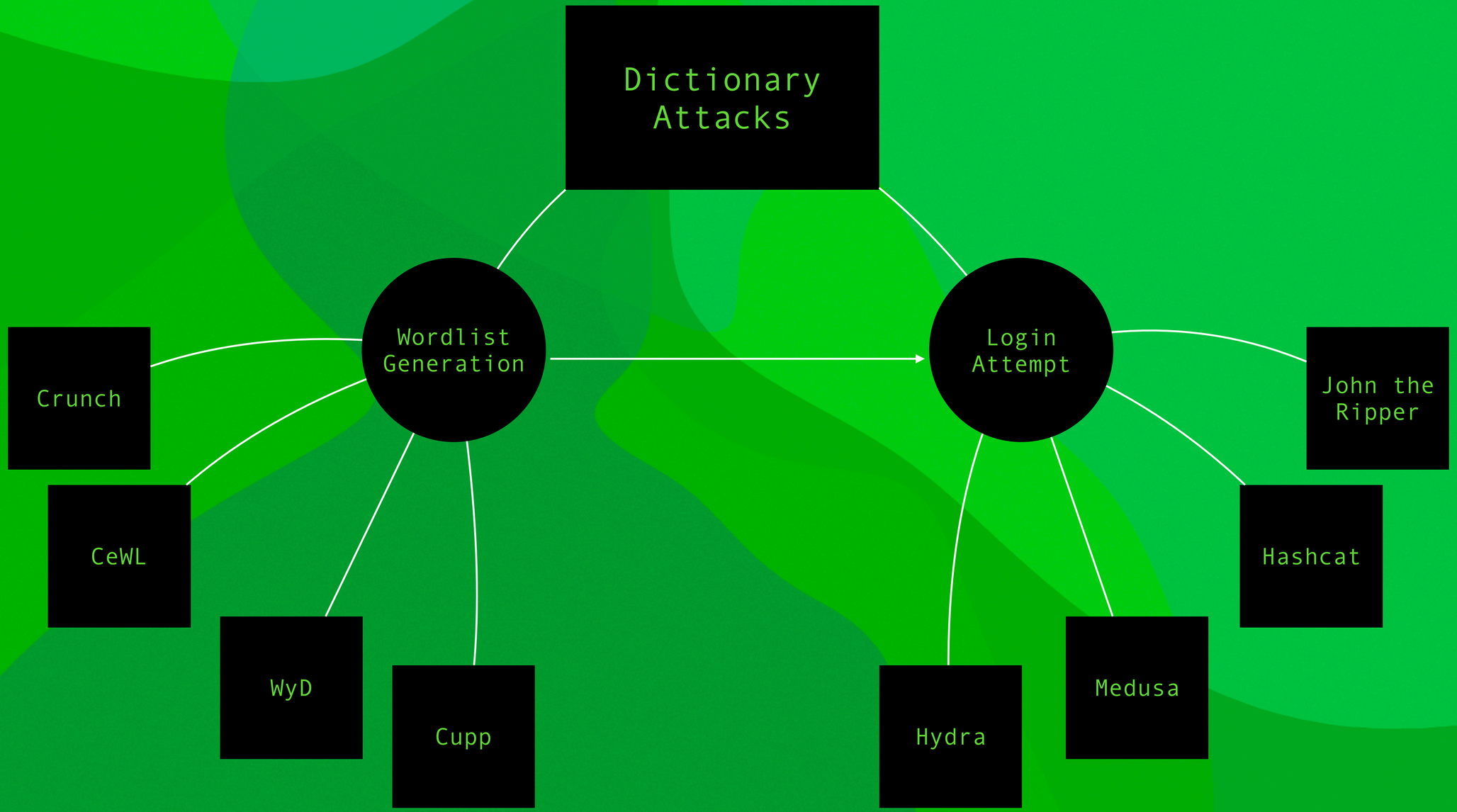
Significance:

Discovering directories and files can lead to the identification of misconfigured permissions, sensitive information leakage, and hidden areas of a website that are not intended for public access, thus broadening the understanding of potential web application vulnerabilities.

Example Command:

```
gobuster dir -u target.com -w dirlist.txt
```

Dictionary attacks


Dictionary Attacks
Wordlist Generation
Login Attempt
Crunch
CeWL
WyD
Cupp
Hydra
Medusa
Hashcat
John the Ripper

Ya’ll after tomorrow