Cyber Warriors: A Comprehensive Introduction to Cybersecurity Tools and Techniques

June 24-28, 2024

Murtuza Jadliwala
murtuza.jadliwala@utsa.edu
Introduction to Web & Internet Security
Web Applications

• Web applications are complex Internet services which employ a multi-tiered architecture involving multiple servers:
  • Application and web servers (public facing).
  • Middleware services, backend and data servers (on the internal network).

• Given the complexity of web services, it is important for a hacker or pen-tester to adapt to each site's specific architecture and service parameters.
  • Testing process must also be consistent to ensure that nothing is missed.
Categories of Web Application Hacks

- Server Side
  - Injection Flaws
  - Remote Code Execution
  - Remote File Inclusion
  - Directory Path Traversal
  - Session Management
  - Logical Implementation Flaws
  - Information Disclosure Vulnerabilities
  - Web Services Exposures

- Client Side
  - Flash Vulnerabilities
  - Browser Exploits
    - JavaScript Vulnerabilities
    - Other HTML5 vulnerabilities
  - Network DNS Exploits
    - DNS Pinning
    - Non-DNS pinning
  - Cross-site scripting
    - Reflected
    - Stored
    - Dom-Based
Reconnaissance and Vulnerability Scanning

Web Application Hacking Methodology

- Set Target
- Spider and Enumerate
- Vulnerability Scanning
- Exploitation
- Cover Tracks
- Maintain Access
Reconnaissance and Vulnerability Scanning

• Specific activities related to web application reconnaissance include:
  • Identifying where and how the target web app is hosted.
  • Enumerating target site directory structure and content management system (CMS), if used, by spidering and offline analysis.
  • Identifying the authentication and authorization mechanisms and determining how the session state is maintained during a transaction with that web service. This usually involves analysis of cookies and how they are used, utilizing a proxy tool.
  • Enumerating and evaluating all forms. As these are primary means for clients to input data and interact with the web service, they are the location of several exploitable vulnerabilities, such as, SQL/XML/JSON injection attacks and cross-site scripting.
  • Identifying other areas that accept input, such as pages that allow for file upload, as well as, any restrictions on accepted upload types.
  • Identifying how errors are handled, and the actual error messages that are received by a user; frequently, the error will provide valuable internal information such as the software version used, or internal filenames and processes.
Detection of Web Application Firewall and Load Balancers

- Identification/Determination of the presence of network-based protective devices such as Web Application Firewalls (WAFs) is important!
- Important because if WAFs are being used, attacks should be designed to bypass them, especially those that rely on specially crafted input.
- WAFs can be identified by manually inspecting cookies (some WAFs tag or modify the cookies that are communicated between the web server and the client), or by changes to the header information in the HTTP response packets identifiable by manually connecting to port 80 using a command-line tool.
- The process of WAF detection can be automated using:

  ```bash
  root@kali:~# nmap -p 80 --script http-waf-detect.nse www
  Starting Nmap 7.70 (https://nmap.org) at 2018-12-23 11:10 EST
  Stats: 0:00:41 elapsed: 0 hosts completed (1 up), 1 undergoing Script Scan
  NSE Timing: About 0.00% done
  Nmap scan report for www.
  Host is up (0.28s latency).
  Other addresses for www.
  PORT STATE SERVICE
  80/tcp open http
  | http-waf-detect: IDS/IPS/WAF detected:
  |   www. [REDACTED]:80/?p4y104d3=<script>alert(document.cookie)</script>
  Nmap done: 1 IP address (1 host up) scanned in 45.61 seconds
  ```
- The above `nmap` script identifies that a WAF is present or not; however, it may not always be accurate and the returned results may be too general to guide an effective strategy for firewall bypass.
Reconnaissance and Vulnerability Scanning

**Detection of Web Application Firewall and Load Balancers**

- Load balancing detector (or lbd) is a Bash shell script that determines whether a given domain uses DNS and/or HTTP load balancing.

- Important information, as it can explain seemingly anomalous results during pen-testing when the load balancer switches requests between multiple servers.

- lbd uses a variety of checks to identify the presence of load balancing.

```
root@kali:~# lbd www._example.com
lbd - load balancing detector 0.4 - Checks if a given domain uses load-balancing.
Written by Stefan Behle (http://ge.mine.nu)
Proof-of-concept! Might give false positives.

Checking for DNS-Loadbalancing: FOUND
www.example.com has address 10.0.0.25
www.example.com has address 10.1.0.25

Checking for HTTP-Loadbalancing [Server]:
cloudflare
NOT FOUND


Checking for HTTP-Loadbalancing [Diff]: FOUND
< CF-RAY: 48dd6093b3f8f6a91-LHR
> CF-RAY: 48dd609463e360e-LHR

www.example.com does Load-balancing. Found via Methods: DNS HTTP[Diff]
```
Reconnaissance and Vulnerability Scanning

Fingerprinting a Web Application and CMS Detection

• Web Application Fingerprinting: Typically first task in reconnaissance & vulnerability scanning done to find out the version and type of the web server running the application, and the implemented web technologies.

• Allows attackers to determine known vulnerabilities and appropriate exploits.

• One way to accomplish this: Use the netcat command to connect to the victim host on port 80, and issue the HTTP HEAD command to identify what is being run on the target web server.

• This returns an HTTP server response, which includes the type of web server that the application is being run on, and the server section providing detailed information about the technology used to build the application.

• The above information can be used in conjunction with a vulnerability database such as CVE to determine exploitable vulnerabilities: (see https://www.cvedetails.com/vulnerability-list/vendor_id-74/product_id-128/PHP-PHP.html).
Reconnaissance and Vulnerability Scanning

Web Crawling Applications: Automated scanning tools used to determine the existing directory and file structure of a particular website, including downloading and analyzing it for vulnerabilities.

Typical findings of web crawlers include administration portals, configuration files (current and previous versions) that may contain hardcoded access credentials and information on the internal structure, backup copies of the website, administrator notes, confidential personal information, and source code.

Kali supports several web crawlers, including Free Burp Suite, DirBuster, OWASP-ZAP, Vega, WebScarab, and WebSlayer.

For example, DirBuster is a GUI-driven crawling tool that performs a brute-force analysis of a website's file tree structure.
Reconnaissance and Vulnerability Scanning

**Mirroring a Web Application**

- **Web Mirroring (Cloning) Applications**: Automated crawling tools that make an offline copy of the website.
  - Cloned/downloaded contents of the target site can be used as input to a program such as crunch, which will produce a personalized word list to support password cracking.

- **Kali** provides an inbuilt application, **httrack**, which provides the option to download all the contents of a website to the local system.

- **httrack** is both a command-line and GUI utility. For example, the following command can be used on the terminal:

  ```bash
  httrack http://targetwebapp/ -O outputfolder
  ```
Reconnaissance and Vulnerability Scanning

Other Web Application Vulnerability Scanners on Kali

<table>
<thead>
<tr>
<th>Application</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Arachnid</td>
<td>An open source Ruby framework that analyzes HTTP responses received during scanning to validate responses and eliminate false positives.</td>
</tr>
<tr>
<td>GoLismero</td>
<td>A scanner that maps web applications and detects common vulnerabilities. The results are saved in TXT, CVS, HTML, and RAW formats.</td>
</tr>
<tr>
<td>Nikto</td>
<td>A Perl-based open source scanner that allows IDS evasion and user changes to scanned modules. This original web scanner is beginning to show its age, and is not as accurate as some of the more modern scanners.</td>
</tr>
<tr>
<td>Skipfish</td>
<td>A scanner that completes a recursive crawl and dictionary-based crawl to generate an interactive site map of the targeted website, annotated with the output from additional vulnerability scans.</td>
</tr>
<tr>
<td>Vega</td>
<td>A GUI-based open source vulnerability scanner. As it is written in Java, it is cross-platform (Linux, macOS, and Windows) and can be customized by the user.</td>
</tr>
<tr>
<td>w3af</td>
<td>A scanner that provides both a graphical and command-line interface to a comprehensive Python testing platform. It maps a target website and scans for vulnerabilities. This project has been acquired by Rapid7, so there will be closer integration with the Metasploit framework in the future.</td>
</tr>
<tr>
<td>Wapiti</td>
<td>A Python-based open source vulnerability scanner.</td>
</tr>
<tr>
<td>Webscarab</td>
<td>OWASP’s Java-based framework for analyzing HTTP and HTTPS protocols. It can act as an intercepting proxy, a fuzzer, and a simple vulnerability scanner.</td>
</tr>
<tr>
<td>Webshag</td>
<td>A Python-based website crawler and scanner that can utilize complex IDS evasion.</td>
</tr>
<tr>
<td>WebSploit</td>
<td>An advanced man-in-the-middle (MiTM) framework, useful in wireless and Bluetooth attacks.</td>
</tr>
</tbody>
</table>
Application-specific Attacks

Brute-forcing Access Credentials of a Web Application

- **What is a access authentication brute-force attack against a website or its services?**: Guessing username and password to access the website or service.

- This attack has a high success rate because users tend to select easy-to-remember credentials or reuse credentials, and also because system administrators frequently don’t control multiple access attempts.

- Kali comes with hydra, a command-line tool, and hydra-gtk, which has a GUI.

```bash
Hydra v8.6 (c) 2017 by van Hauser/THC - Please do not use in military or secret service organizations, or for illegal purposes.

Hydra (http://www.thc.org/thc-hydra) starting at 2018-12-23 15:11:02
[DATA] max 6 tasks per 1 server, overall 6 tasks, 6 login tries (1:1/p:6), -1 try per task
1 of 1 target successfully completed, 1 valid password found
Hydra (http://www.thc.org/thc-hydra) finished at 2018-12-23 15:11:18
```

```bash
```
Application-specific Attacks

Injection Attacks: Command Injection in Web Applications

- Command Injection: Pass malicious “values” or “commands” to vulnerable web applications through HTTP Post method or as URL parameters. These malicious “commands” are then executed by the target OS (through the vulnerable web application) resulting in an privilege escalation or unauthorized access/execution!
  - Primarily caused due to insufficient input validation.

- Command injection exploiter (commix) is an automated tool written in Python (precompiled in Kali) used for testing command injection vulnerabilities in web applications.

- The command injection attacks are independent of the operating system in use. They can target Linux, Unix, and Windows as well. They are also independent of the programming languages used as they can inject themselves into many programming languages including C, C++, PHP, Python, and Java.
Application-specific Attacks

**Injection Attacks: Command Injection in Web Applications**

- Commix (https://github.com/commixproject/commix) also comes as an additional plugin in various penetration testing frameworks such as TrustedSec's Penetration Testers Framework (PTF) and OWASP's Offensive Web Testing Framework (OWTF).

- Attackers may use all the functionalities provided by commix by entering commix -h in the Terminal.

- Commix website has links to various test pwnable (compromisable) VMs and testbeds, usage examples and several cool demos!

- A useful demo of a command injection using commix: https://www.youtube.com/watch?v=A57pbJA706U
Questions