Finding Vulnerabilities

SoftUni Team

Technical Trainers

Software University

SoftUni

Software University

https://softuni.bg

Have a Questions?

Software University

sli.do

#Cyber_Security

Table of Contents

Finding Vulnerabilities Theory

Finding Vulnerabilities in a Nutshell

- **Finding Vulnerabilities** is an act of art
- This is where we must combine all detailed information from previous steps, and weaponize it
- The idea here is to seek for something unusual, something that can be abused or something that is off order
- For example:
  - Old version, **missing field sanitization**, **missing HTTPS** and many, many more
- This can be done by fuzzing (poking the application with specific "**malicious**" payloads)

Finding Vulnerabilities in a Nutshell

- In order to find a vulnerability, you must be aware what vulnerability is and what types of vulnerabilities are there
- You must have the idea of what you are trying to find
- This means you must create and implement a methodology
- Every pentester have his unique methodology, or way of doing things
- Example methodology from hacktricks:
  - *https://hacktricks.boitatech.com.br/pentesting/pentesting-web*

Finding Vulnerabilities in a Nutshell

Software University

- Do not look at finding vulnerabilities like a **linear process**, it is not going through big checklist
- Instead be as creative as you can, perform illogical things, as well as **logical ones**
- Force the application to behave unexpected
- Reasearch and **Find the Bug**!

Nutshell

Finding Vulnerabilities Techniques

Active Enumeration

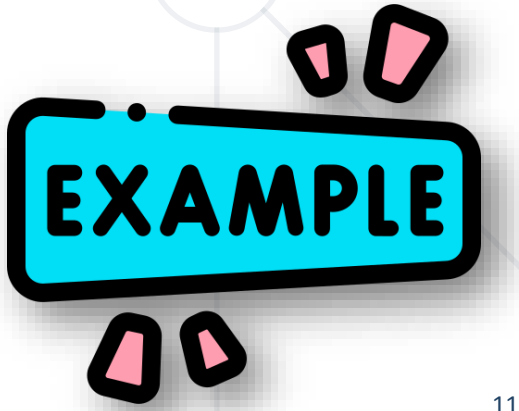
- Sometimes vulnerabilities can arise just from **service enumeration**
- That's why scanning must be complex and detailed, it can find the **low hanging fruits** like:
  - Obsolete versions
  - Hardcoded credentials
  - Default credentials
  - Backup files with credentials
  - Vulnerable plugins and extensions and more

Fuzzing Services

- Vulnerabilities can appear from **every angle**
- It is a must to go through every **single opened service**, trying to determine:
  - What it is and how it behaves?
  - What happens when you supply strange input?
  - Is this **input** / **output** pattern matches a vulnerability?
    - If so, continue testing deeper
    - If not, continue testing for different vulnerabilities

Example

- You find ftp service
- You try to connect as **anonymous user** and it was successful
- You try to **upload files**, and it failed
- The service is up to date and has no active exploits available, nor sensitive information to get
- You move to **port 80**



Example

- You found a blog web application
- You input at the search and the application **returned 500** (Internal Server Error)
- You copy the request with burpsuite and run it with sqlmap, and dumped the full database
- The application returning 500 was **unexpected** and **proved** something is going on
- That's the act of finding vulnerability (by fuzzing the application for **sql injection with**)

Searching for Business Logic Vulnerabilities

- While fuzzing is great at finding vulnerabilities on different components, most usually the most dangerous vulnerabilities are **business logic ones**
- The good practice is to always run something on the background while you search for something no machine can find
- **Nessus** and **Burp** are great at fuzzing and can fuzz better than everyone

Searching for Business Logic Vulnerabilities

- Business logic vulnerabilities are harder to find and exploit since you must completely understand the target scope
- For example a normal vulnerability is **sql injection** and a lot of **automatic tools** can fuzz it
- On the other hand business logic vulnerability is for example user password reset misconfiguration, weak session management, misconfigured active-directory certificate service, or even chaining multiple attack vectors, like **uploading to ftp** and **execution on web**

Automatic Vulnerability Scanning

- Just run **Nessus** or **Burp** active scan and they will do the heavy lifting for you
- Nessus is mainly used for infrastructure scope
- Burp is used for **Web App scope**
- Even though Nessus can scan web applications, **Burp Pro** is considered the best

Active Reconnaissance Tools

Tools cannot save you here!

But these one surely helps a lot

Searchsploit / Exploitdb

- What if you find a service is running at **older version**?
- Next step is to see if there is an active exploit for that version

```
┌──(kali㉿kali)-[~]
└─$ searchsploit wordpress
---------------------------------------------------------------------------------------------------------------------
 Exploit Title                                                                                                      |  Path
---------------------------------------------------------------------------------------------------------------------
Joomla! Plugin JD-WordPress 2.0 RC2 - Remote File Inclusion                                                         | php/webapps/9890.py
Joomla! Plugin JD-WordPress 2.0-1.0 RC2 - 'wp-comments-post.php' Remote File Inclusion                              | php/webapps/28295.txt
Joomla! Plugin JD-WordPress 2.0-1.0 RC2 - 'wp-feed.php' Remote File Inclusion                                       | php/webapps/28296.txt
Joomla! Plugin JD-WordPress 2.0-1.0 RC2 - 'wp-trackback.php' Remote File Inclusion                                  | php/webapps/28297.txt
Mulitple WordPress Themes - 'admin-ajax.php?img' Arbitrary File Download                                            | php/webapps/34511.txt
Multiple WordPress Orange Themes - Cross-Site Request Forgery (Arbitrary File Upload)                               | php/webapps/29946.txt
Multiple WordPress Plugins (TimThumb 2.8.13 / WordThumb 1.07) - 'WebShot' Remote Code Execution                     | php/webapps/33851.txt
Multiple WordPress Plugins - 'timthumb.php' File Upload                                                             | php/webapps/17872.txt
Multiple WordPress Plugins - Arbitrary File Upload                                                                  | php/webapps/41540.py
Multiple WordPress Themes - 'upload.php' Arbitrary File Upload                                                      | php/webapps/37417.php
Multiple WordPress UpThemes Themes - Arbitrary File Upload                                                          | php/webapps/36611.txt
Multiple WordPress WooThemes Themes - 'test.php' Cross-Site Scripting                                               | php/webapps/35830.txt
Multiple WordPress WPScientist Themes - Arbitrary File Upload                                                       | php/webapps/38167.php
phpWordPress 3.0 - Multiple SQL Injections                                                                          | php/webapps/26608.txt
Wordpress 4.9.6 - Arbitrary File Deletion (Authenticated) (2)                                                       | php/webapps/50456.js
WordPress 5.0.0 - Image Remote Code Execution                                                                       | php/webapps/49512.py
WordPress 5.7 - 'Media Library' XML External Entity Injection (XXE) (Authenticated)                                 | php/webapps/50304.sh
WordPress Core - 'load-scripts.php' Denial of Service                                                               | php/dos/43968.py
WordPress Core / MU / Plugins - '/admin.php' Privileges Unchecked / Multiple Information Disclosures                | php/webapps/9110.txt
WordPress Core 0.6/0.7 - 'Blog.header.php' SQL Injection                                                            | php/webapps/23213.txt
WordPress Core 1.0.7 - 'Pool index.php' Cross-Site Scripting                                                        | php/webapps/30520.txt
WordPress Core 1.2 - 'admin-header.php?redirect_url' Cross-Site Scripting                                           | php/webapps/24642.txt
WordPress Core 1.2 - 'bookmarklet.php' Multiple Cross-Site Scripting Vulnerabilities                                | php/webapps/24643.txt
WordPress Core 1.2 - 'categories.php?cat_ID' Cross-Site Scripting                                                   | php/webapps/24644.txt
WordPress Core 1.2 - 'edit-comments.php' Multiple Cross-Site Scripting Vulnerabilities                              | php/webapps/24646.txt
WordPress Core 1.2 - 'edit.php?s' Cross-Site Scripting                                                              | php/webapps/24645.txt
WordPress Core 1.2 - 'wp-login.php' HTTP Response Splitting                                                         | php/webapps/24667.txt
WordPress Core 1.2 - 'wp-login.php' Multiple Cross-Site Scripting Vulnerabilities                                   | php/webapps/24641.txt
```

SQLmap

- SQLmap is used for automatic **SQL injection discovery** and **exploitation**
- It supports multiple **DB engines** and it is easy to use
- One of the coolest part about sqlmap is that it supports **Burp Requests**
- This makes scanning even easier

```
┌──(kali㉿kali)-[~]
└─$ sqlmap
        ___
       __H__
 ___ ___[']_____ ___ ___  {1.6.10#stable}
|_ -| . [(]     | .'| . |
|___|_  [']_|_|_|__,|  _|
      |_|V...       |_|   https://sqlmap.org

Usage: python3 sqlmap [options]

sqlmap: error: missing a mandatory option (-d, -u, -l, -m, -r, -g, -c, --wizard, --shell, --update, --purge, --list-tampers or --dependencies). Use -h for basic and -hh for advanc
ed help
```

Web Browser

- This is one of the main tools for discovering web vulnerabilities
- It is helpful for web enumeration and exploitation
- Preferred browser is **firefox**

BurpSuite

- This is the best tool for discovering and exploiting web vulnerabilities
- It works as a **http proxy**, meaning it can capture **requests** and **responses**, allowing the pentester to alter them before execution
- The paid version can also perform automatic vulnerability scanning

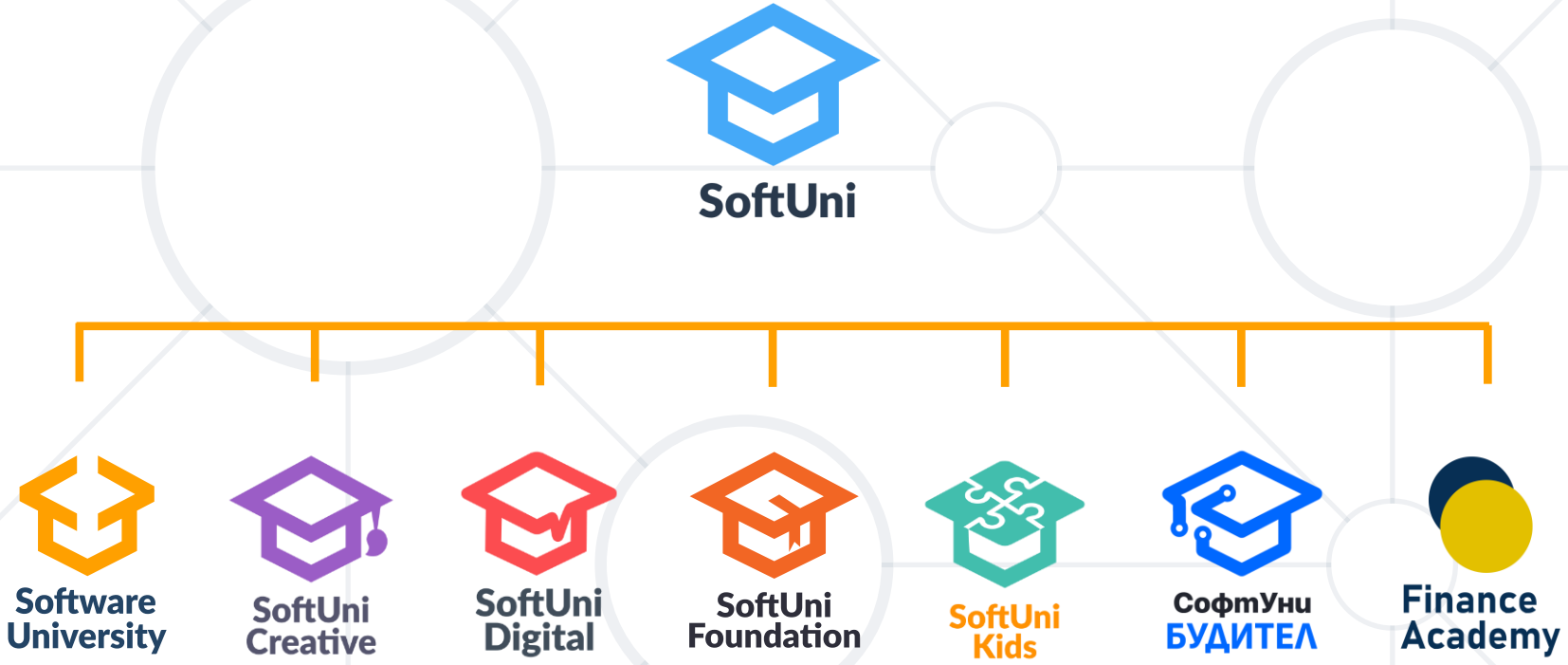
Nessus

- Nessus is considered the best vulnerability scanner
- It supports high numbers of plugins, allowing it to scan for series of vulnerabilities
- It is mostly used against infrastructure (other services from **HTTP/S**)
- It has **free version**, it supports not that many plugins but it is nice for beginners

Summary

- Finding Vulnerabilities in a Nutshell
- Finding Vulnerabilities Techniques
  - Active Enumeration
  - Fuzzing Services
- Active Reconnaissance Tools
  - Searchsploit / Exploitdb
  - SQLmap
  - Web Browser
  - BurpSuite
  - Nessus

Questions?



SoftUni Diamond Partners

Software University

SmartIT

Coca-Cola HBC Bulgaria

SUPER HOSTING .BG

createX

Postbank

Решения за твоето утре

DRAFT KINGS

THE CROWN IS YOURS

VIVACOM

AMBITIONED

INDEAVR

Serving the high achievers

PHAR VISION

Trainings @ Software University (SoftUni)

- Software University – High-Quality Education, Profession and Job for Software Developers
  - softuni.bg, about.softuni.bg
- Software University Foundation
  - softuni.foundation
- Software University @ Facebook
  - facebook.com/SoftwareUniversity

License

- This course (slides, examples, demos, exercises, homework, documents, videos and other assets) is **copyrighted content**
- Unauthorized copy, reproduction or use is illegal
- © SoftUni – https://about.softuni.bg/
- © Software University – https://softuni.bg