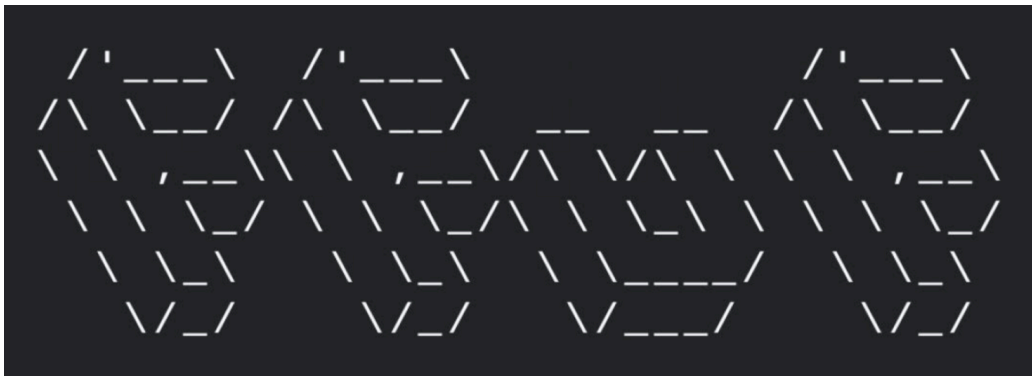
INFORMATION SECURITY IA-2

REPORT



Topic: Fuzz Faster U Fool (FFUF)

Sr. No.	Title
1	Introduction
2	Features / Characteristics
3	Methodology
4	Results
5	Conclusion

Team Members:
Isha Khandalekar: 16010121083
Kedar Kulkarni: 16010121096
Himanshu Patil: 16010121141

GitHub Repository: https://github.com/agntgalahad/ffuf-Implementation/

YouTube Demonstration: https://youtu.be/3cZgFY3jI4A

Introduction

Fuzz Faster U Fool (FFUF) stands as a dynamic and swift web fuzzer crafted in Go, tailored specifically for penetration testing and meticulous web application analysis. This tool operates with remarkable agility, unleashing a barrage of HTTP requests adorned with diverse parameters to meticulously scrutinize web applications. By executing brute force assaults across various facets of a web server, FFUF unveils hidden content, sensitive files, directories, and other clandestine elements, illuminating potential entry points for malicious exploitation.

This report endeavors to provide an insightful exploration of FFUF, delving into its functionalities, strengths, and limitations. Just as with Aircrack-ng, it elucidates the intricacies of FFUF's operation, demonstrating how its multifaceted approach aids in the identification and mitigation of web application vulnerabilities. Through practical examples and real-world use cases, this report illuminates FFUF's prowess in fortifying web application security.

Features/Characteristics

Some of the well known features of FFUF are:

1. **Flexibility and Speed:** It is particularly noted for its speed and flexibility, enabling users to quickly integrate it into their tooling and workflows. This makes it a preferred choice for bug bounty hunters and security professionals.

2. **Web Fuzzing:** FFUF is designed for discovering elements and content within web applications or servers, uncovering hidden content, directories, and other sensitive information.

3. **Support for Multiple HTTP Methods:** FFUF supports various HTTP methods, including GET, POST, and others, enabling comprehensive testing of web applications.

4. **Recursive Directory Scanning:** FFUF enables recursive directory scans, facilitating the discovery of hidden directories and files within web applications.

5. **Advanced Response Filtering:** FFUF provides advanced filtering options to reduce false positives and focus on relevant responses, thereby enhancing the efficiency of the fuzzing process.

6. **Performance Optimization:** Equipped with features for optimizing performance, such as request throttling and delays, FFUF prevents overwhelming the target server..

7. **Integration with Other Tools**: FFUF seamlessly integrates with other security tools like **Burp Suite**, expanding its utility in broader security testing contexts.

8. **Brute Force Login Pages:** Supporting modes like **clusterbomb** and **pitchfork**, FFUF is effective for brute-forcing login pages, facilitating testing of authentication mechanisms.

9. **Support for TLS/SSL Connections:** FFUF handles secure connections, ensuring that fuzzing can be performed on applications using HTTPS.

Methodology

1. `ffuf -w ~/wordlists/common.txt -u http://ffuf.me/cd/basic/FUZZ`

This command utilizes the FFuF tool to perform web content discovery (fuzzing) by replacing the "FUZZ" keyword in the given URL with each word from the common.txt wordlist file, attempting to find valid endpoints or directories on the target website.

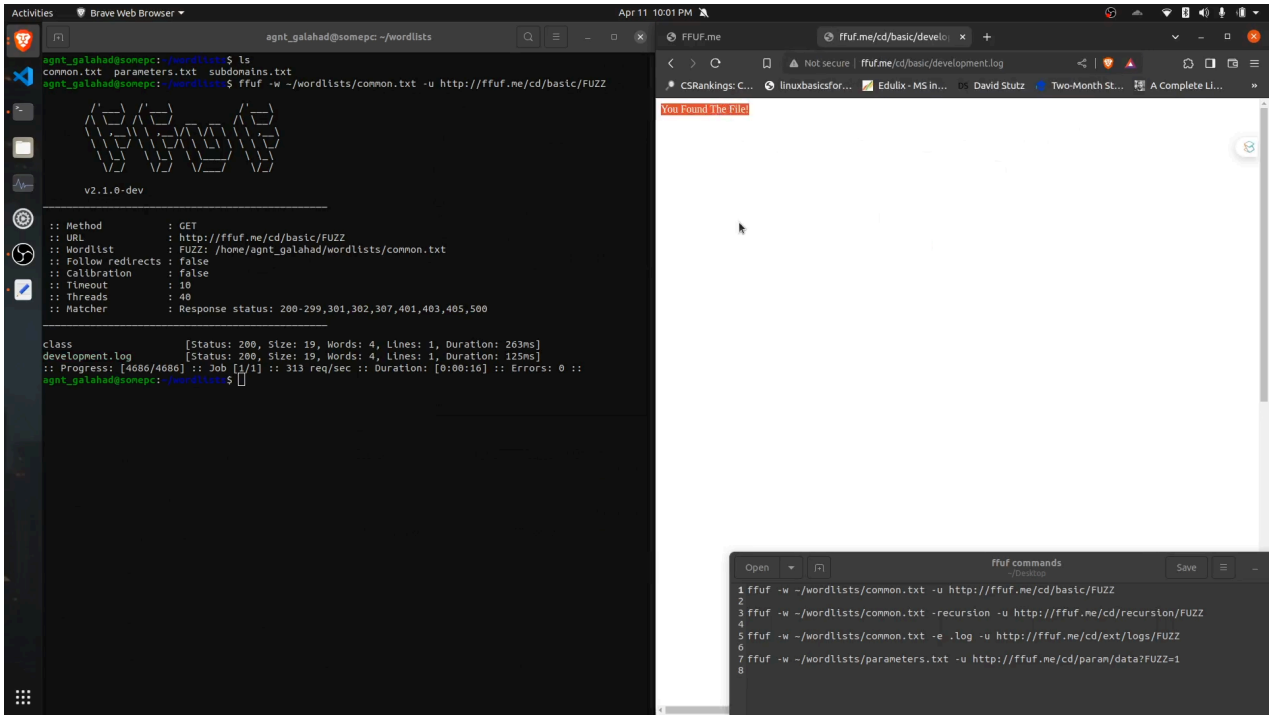
2. ffuf -w ~/wordlists/common.txt -recursion -u http://ffuf.me/cd/recursion/FUZZ

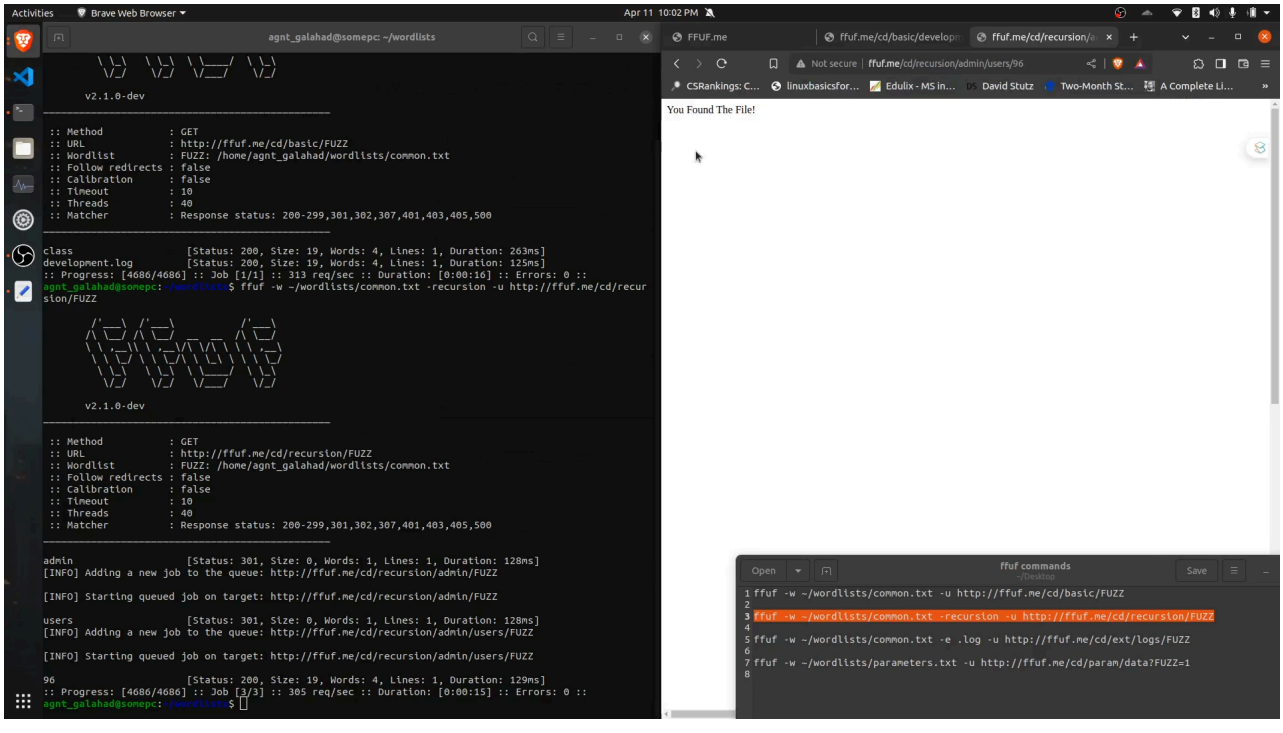
   This command employs FFuF with recursion enabled to perform recursive directory/file discovery on the target website by replacing the "FUZZ" keyword in the URL with each word from the common.txt wordlist, exploring potential subdirectories or files within the target directory.

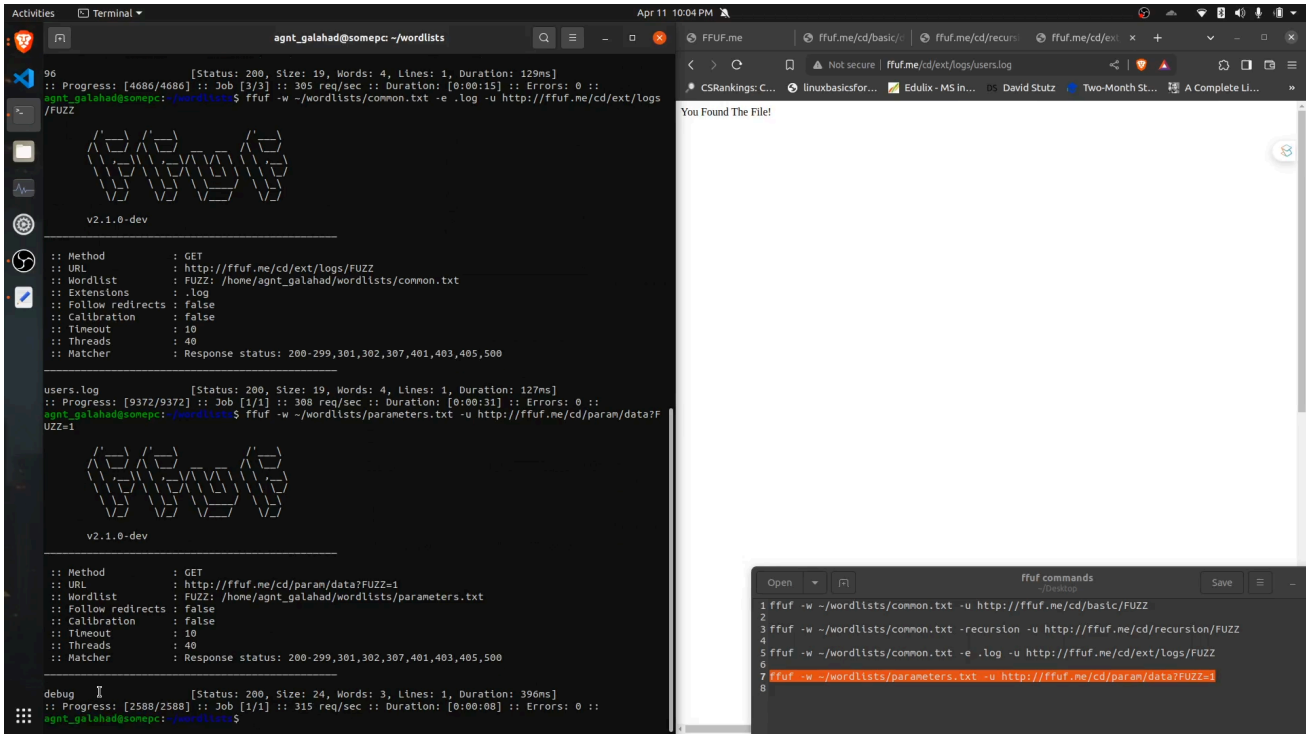
3. ffuf -w ~/wordlists/common.txt -e .log -u http://ffuf.me/cd/ext/logs/FUZZ
   This command utilizes FFuF to perform content discovery on a web server by appending ".log" extension to each word in the common.txt wordlist and replacing the "FUZZ" keyword in the URL, attempting to find accessible log files on the target website.

4. ffuf -w ~/wordlists/parameters.txt -u http://ffuf.me/cd/param/data?FUZZ=1
   This command utilizes FFuF to perform parameter discovery on a web server by replacing the "FUZZ" parameter value in the URL with each word from the parameters.txt wordlist, attempting to identify valid parameters or values that may reveal additional functionality or vulnerabilities on the target website.

Results


Activities
Brave Web Browser
Apr 11 10:01 PM
agnt_galahad@somepc: ~/wordlists
agnt_galahad@somepc:~/wordlists$ ls
common.txt parameters.txt subdomains.txt
agnt_galahad@somepc:~/wordlists$ ffuf -w ~/wordlists/common.txt -u http://ffuf.me/cd/basic/FUZZ
v2.1.0-dev
:: Method : GET
:: URL : http://ffuf.me/cd/basic/FUZZ
:: Wordlist : FUZZ: /home/agnt_galahad/wordlists/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200-299,301,302,307,401,403,405,500
class [Status: 200, Size: 19, Words: 4, Lines: 1, Duration: 263ms]
development.log [Status: 200, Size: 19, Words: 4, Lines: 1, Duration: 125ms]
:: Progress: [4686/4686] :: Job [1/1] :: 313 req/sec :: Duration: [0:00:16] :: Errors: 0 ::
FFUF.me
ffuf.me/cd/basic/development.log
You Found The File!
ffuf commands
1 ffuf -w ~/wordlists/common.txt -u http://ffuf.me/cd/basic/FUZZ
3 ffuf -w ~/wordlists/common.txt -recursion -u http://ffuf.me/cd/recursion/FUZZ
5 ffuf -w ~/wordlists/common.txt -e .log -u http://ffuf.me/cd/ext/logs/FUZZ
7 ffuf -w ~/wordlists/parameters.txt -u http://ffuf.me/cd/param/data?FUZZ=1


Activities
Brave Web Browser
Apr 11 10:02 PM
agnt_galahad@somepc: ~/wordlists
v2.1.0-dev
:: Method : GET
:: URL : http://ffuf.me/cd/basic/FUZZ
:: Wordlist : FUZZ: /home/agnt_galahad/wordlists/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200-299,301,302,307,401,403,405,500
class [Status: 200, Size: 19, Words: 4, Lines: 1, Duration: 263ms]
development.log [Status: 200, Size: 19, Words: 4, Lines: 1, Duration: 125ms]
:: Progress: [4686/4686] :: Job [1/1] :: 313 req/sec :: Duration: [0:00:16] :: Errors: 0 ::
agnt_galahad@somepc:~/wordlists$ ffuf -w ~/wordlists/common.txt -recursion -u http://ffuf.me/cd/recursion/FUZZ
:: Method : GET
:: URL : http://ffuf.me/cd/recursion/FUZZ
:: Wordlist : FUZZ: /home/agnt_galahad/wordlists/common.txt
:: Follow redirects : false
:: Calibration : false
:: Timeout : 10
:: Threads : 40
:: Matcher : Response status: 200-299,301,302,307,401,403,405,500
admin [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 128ms]
[INFO] Adding a new job to the queue: http://ffuf.me/cd/recursion/admin/FUZZ
[INFO] Starting queued job on target: http://ffuf.me/cd/recursion/admin/FUZZ
users [Status: 301, Size: 0, Words: 1, Lines: 1, Duration: 128ms]
[INFO] Adding a new job to the queue: http://ffuf.me/cd/recursion/admin/users/FUZZ
[INFO] Starting queued job on target: http://ffuf.me/cd/recursion/admin/users/FUZZ
96 [Status: 200, Size: 19, Words: 4, Lines: 1, Duration: 129ms]
:: Progress: [4686/4686] :: Job [3/3] :: 305 req/sec :: Duration: [0:00:15] :: Errors: 0 ::
FFUF.me
ffuf.me/cd/recursion/admin/users/96
You Found The File!
ffuf commands
1 ffuf -w ~/wordlists/common.txt -u http://ffuf.me/cd/basic/FUZZ
3 ffuf -w ~/wordlists/common.txt -recursion -u http://ffuf.me/cd/recursion/FUZZ
5 ffuf -w ~/wordlists/common.txt -e .log -u http://ffuf.me/cd/ext/logs/FUZZ
7 ffuf -w ~/wordlists/parameters.txt -u http://ffuf.me/cd/param/data?FUZZ=1



Conclusion

In conclusion, FFuF emerges as a pivotal tool within the domain of web security, offering indispensable capabilities for fortifying online defenses. With its versatile functionality and intuitive approach, FFuF empowers security practitioners and penetration testers to efficiently uncover potential weaknesses and fortify web infrastructures. As the digital landscape continues to evolve, the ongoing advancement and optimization of tools like FFuF will remain paramount in staying abreast of emerging threats and safeguarding the integrity of online assets.

The practical demonstration outlined through the provided commands underscores FFuF's significance in real-world security contexts. From conducting comprehensive directory and parameter discovery to uncovering hidden endpoints and potentially sensitive files, each operation contributes to a comprehensive assessment of web application security posture. Through continued utilization and refinement, FFuF stands as an essential ally in the continuous battle against web-based vulnerabilities and threats.

References

https://github.com/ffuf/ffuf
http://codingo.io/tools/ffuf/bounty/2020/09/17/everything-you-need-to-know-about-ffuf.html
https://medium.com/quiknapp/fuzz-faster-with-ffuf-c18c031fc480
https://www.freecodecamp.org/news/how-to-fuzz-hidden-directories-files-with-ffuf/
https://github.com/ffuf/ffuf/wiki
https://www.kali.org/tools/ffuf/
https://www.freecodecamp.org/news/web-security-fuzz-web-applications-using-ffuf/
https://medium.com/@techmindxperts/a-comprehensive-guide-to-ffuf-for-web-security-testing-207633f98217
https://www.acceis.fr/ffuf-advanced-tricks/
https://security.packt.com/fuzzing-faster-with-ffuf/