**Content Discovery**

Để khám phá nội dung trên 1 trang web, có 3 cách chính:

* Thủ công
* Tự động
* Osint

Nội dung đó có thể là trang web dành cho nhân viên, những phiên bản cũ của trang web, tập tin sao lưu, cấu hình, mục quản trị ,…

**Khám phá thủ công:**

**robots.txt**

Tập tin robots.txt là tài liệu cho công cụ tìm kiếm biết những trang đó là và không được phép hiển thị trên kết quả của công cụ tìm kiếm hoặc cấm hoàn toàn các công cụ tìm kiếm thu thập dữ liệu trang web.

Nó có thể hạn chế một số khu vực của trang web hiển thị như mục quản trị, các tệp khách hàng

**Favicon**

Favicon là icon nhỏ hiển thị trong thanh địa chỉ của trình duyệt hoặc tab mở trang web.

Thỉnh thoảng khi những thư viện dùng để dựng 1 trang web, favicon là quá trình cài đặt còn sót lại, và nếu nhà phát triển web không thay thế nó, điều này có thể cho chúng ta manh mối về thư viện đang dùng. OWASP lưu trữ danh sách những icons thư viện để bạn có thể tra cứu.

**Sitemap.xml**

Không giống tập tin robots.txt, hạn chế cái công cụ tìm kiếm có thể thấy, sitemap.xml đưa danh sách tập tin mà trang web muốn liệt kê trên công cụ tìm kiếm.

Chúng chứa những khu vực của trang web khó chuyển hướng hơn hoặc những trang web cũ mà ứng dụng không còn dùng nữa nhưng vẫn hoạt động sau lưng.

**HTTP Headers:**

Chứa nhiều thông tin hữu ích như phần mềm máy chủ, thậm chí ngôn ngữ lập trình sử dụng.

Nhờ đó, chúng ta có thể thấy những lỗ hổng ở phiên bản cũ mà phần mềm đang dùng.

**Framework Stack:**

Khi bạn biết thư viện ứng dụng đang dùng nhờ favicon hay comments,.. Sau đó bạn vào trang web của thư viện. Từ đó, chúng ta có thể học nhiều hơn về phần mềm và thông tin khác dẫn đến khám phá nhiều nội dung hơn.

**OSINT:**

**Google Hacking/Dorking:**

|  |  |  |
| --- | --- | --- |
| **Filter** | **Example** | **Description** |
| site | site:tryhackme.com | returns results only from the specified website address |
| inurl | inurl:admin | returns results that have the specified word in the URL |
| filetype | filetype:pdf | returns results which are a particular file extension |
| intitle | intitle:admin | returns results that contain the specified word in the title |

**Wappalyzer:** là 1 công cụ online và tiện ích của trình duyệt giúp xác định công nghệ mà trang web đó sử dụng như thư viện, hệ quản trị nội dung(CMS), bộ xử lí thanh toán thậm chí là phiên bản của chúng.

**Wayback machine:** là 1 kho lưu trữ lịch sử của các trang web. Dịch vụ này có thể giúp bạn khám phá những trang web cũ vẫn còn hoạt động trên ứng dụng.

**Github:** Gitlà 1 hệ thống theo dõi sự thay đổi của tập tin trong 1 dự án. Github là dịch vụ lưu trữ của Git trên internet. Bạn có thể dùng tính năng tìm kiếm của Github để tìm kiếm tên công ty, trang web thuộc mục tiêu tấn công của bạn.

**S3 Buckets:** là 1 dịch vụ lưu trữ được cung cấp bởi Amazon AWS, cho phép mọi người lưu tập tin, thậm chí nội dung tĩnh của trang web trong đám mây có thể truy cập qua HTTP hay HTTPS

Định dạng của S3 Buckets là http(s)://**{name}.**[**s3.amazonaws.com**](http://s3.amazonaws.com/)

**Khám phá tự động:** gobuster, ffuf,…

**Nmap:**

nmap là 1 công cụ rất mạnh mẽ, mã nguồn mở và miễn phí để khám phá máy chủ và các dịch vụ trên mạng máy tính.

|  |  |
| --- | --- |
| **nmap flag** | **Description** |
| -sV | Attempts to determine the version of the services running |
| -p <x> or  -p- | Port scan for port <x> or scan all ports |
| -Pn | Disable host discovery and just scan for open ports |
| -A | Enables OS and version detection, executes in-build scripts for further enumeration |
| -sC | Scan with the default nmap scripts |
| -v | Verbose mode |
| -sU | UDP port scan |
| -sS | TCP SYN port scan |

**Tìm kiếm thư mục bằng GoBuster:**

GoBuster là một công cụ tấn công dồn dập Urls (thư mục và tập tin), những tên miền và máy chủ.

Chạy GoBuster với wordlist:

**gobuster dir -u http://<ip>:3333 -w <word list location>**

|  |  |
| --- | --- |
| **GoBuster flag** | **Description** |
| -e | Print the full URLs in your console |
| -u | The target URL |
| -w | Path to your wordlist |
| -U and -P | Username and Password for Basic Auth |
| -p **<x>** | Proxy to use for requests |
| -c <http cookies> | Specify a cookie for simulating your auth |

**Subdomain Enumeration:**

Liệt kê tên miền con giúp chúng ta mở rộng phạm vi tấn công, tăng cơ hội khám phá được những lỗ hổng tiềm năng.

Chúng ta sẽ làm việc này bằng 3 phương pháp khác nhau: tấn công dồn dập, OSINT và máy chủ ảo.

**OSINT – Chứng chỉ SSL/TLS**

Khi chứng chỉ được tạo ra cho tên miền bởi CA, nó sẽ được ghi logs. Mục đích của logs này là dừng sử dụng những chứng chỉ độc hại. Chúng ta có thể tận dụng lợi ích của dịch vụ này để khám phá tên miền con. Trang web https://crt.sh cung cấp 1 cơ sở dữ liệu về những chứng chỉ bạn muốn tìm.

**OSINT – Công cụ tìm kiếm:**

Công cụ tìm kiếm chứa hàng triệu đường dẫn đến hàng tỷ trang web, có thể là 1 mã nguồn tuyệt vời để tìm tên miền con. Sử dụng phương thức tìm kiếm nâng cao như site: filter có thể thu hẹp phạm vi tìm kiếm. Ví dụ: “site:\*.tryhackme.com”

**Tấn công dồn dập DNS:**

Tấn công dồn dập DNS để liệt kê là 1 phương pháp thử hàng triệu tên miền con phổ biến. Vì phương pháp này gửi rất nhiều yêu cầu nên cần 1 công cụ tự động là dnsrecon.

dnsrecon -t brt -d acmeitsupport.thm

**OSINT – Sublist3r:**

Để tăng quá trình khám phá tên miền con, chúng ta có thể tự động phương pháp trên nhờ sự giúp sức của công cụ như sublist3r

./sublist3r.py -d acmeitsupport.thm

**Máy chủ ảo:**

Một vài tên miền con không được lưu trữ công cộng trong kết quả DNS như phiên bản phát triển của ứng dụng web hay trang quản trị. Thay vào đó, bản ghi DNS có thể được giữ ở máy chủ DNS riêng hoặc ghi lại trên máy của nhà phát triển ở tập tin /etc/hosts (tập tin c:\windows\system32\drivers\etc\hosts đối với người dùng Window) sẽ gắn tên miền đến địa chỉ ip.

Vì máy chủ web có thể lưu trữ nhiều trang web từ 1 máy chủ khi 1 trang web được yêu cầu từ máy khách, máy chủ biết trang web nào máy khách muốn từ header Host. Chúng ta có thể thay đổi giá trị host này và theo dõi phản hồi xem có khám phá được trang web mới không.

Như tấn công dồn dập DNS, chúng ta có thể tự động quá trình này bằng việc dùng từ điển những tên miền con phổ biến

Thử lệnh sau:

ffuf -w /usr/share/wordlists/SecLists/Discovery/DNS/namelist.txt -H "Host: FUZZ.acmeitsupport.thm" -u http://10.10.120.40

Câu lệnh trên dùng -w để chỉ định từ điển chúng ta sẽ dùng. -H giúp chỉnh sửa header (trong trường hợp này là Host header), dùng tự khóa FUZZ ở vị trí tên miền con.

**FFUF**

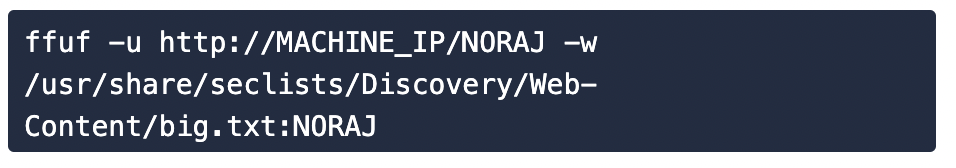
**Basic:**

Tối thiểu chúng ta cần hai tham số: -u để chỉ định url và -w để chỉ định wordlist

Từ khóa FUZZ để thông báo cho ffuf nơi để tiêm wordlist.

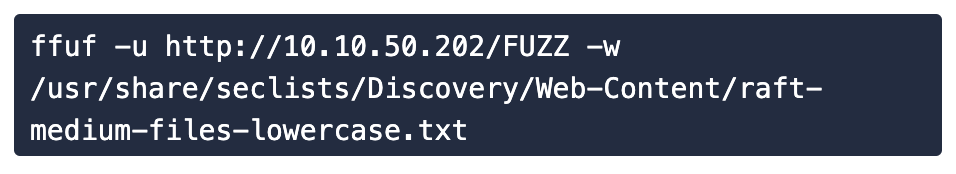


Bạn cũng có thể dùng từ khóa khác để thay thế cho FUZZ bằng cách khai báo như sau: wordlist.txt:KEYWORD



**Finding pages and directories:**

Đầu tiên, bạn nên bắt đầu liệt kê các tệp phổ biến ở raft-medium-files-lowercase.txt



Tuy nhiên, sử dụng 1 wordlist lớn như vậy sẽ chứa các phần mở rộng tệp không liên quan dẫn đến hiệu suất kém.

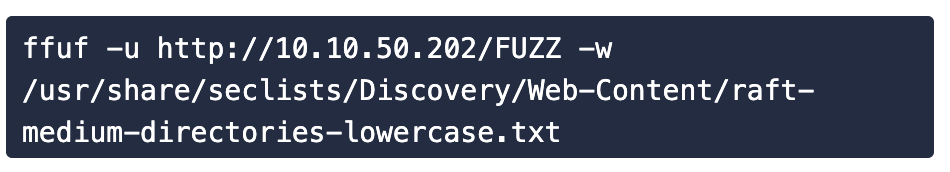
Thay vào đó chúng ta có thể giả định index.<extension> là trang mặc định của hầu hết trang web vì vậy chúng ta có thể thử các extensions phổ biến chỉ với trang index. Với phương pháp này, chúng ta có thể xác định ngôn ngữ lập trình mà trang web sử dụng.



Bây giờ chúng ta biết các extension hỗ trợ rồi, chúng ta có thể thử 1 nhóm khác mà không có extension (ngoài index) và áp dụng các extension mới kiếm được.

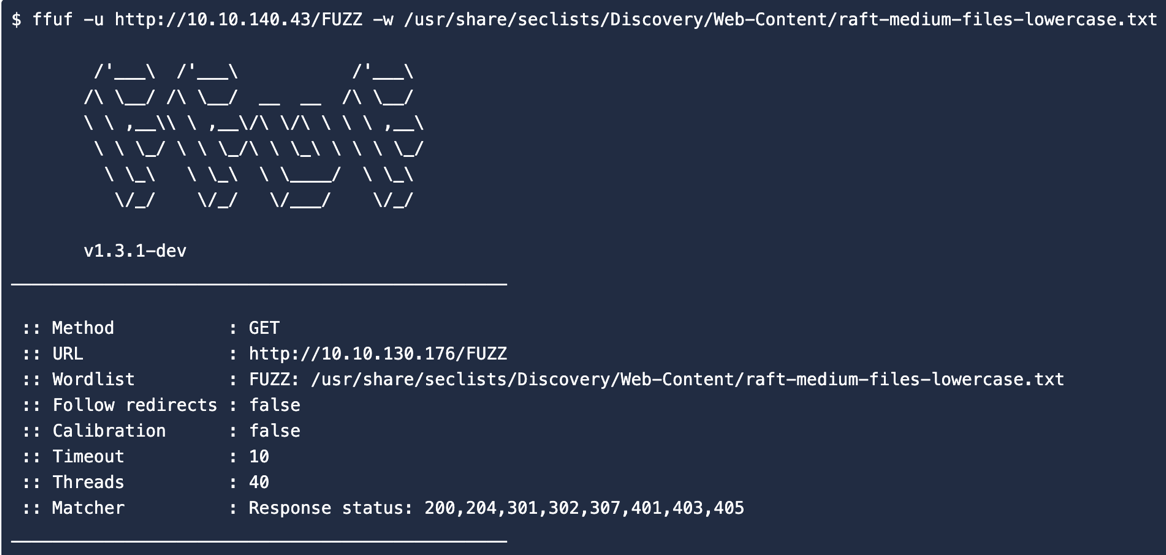


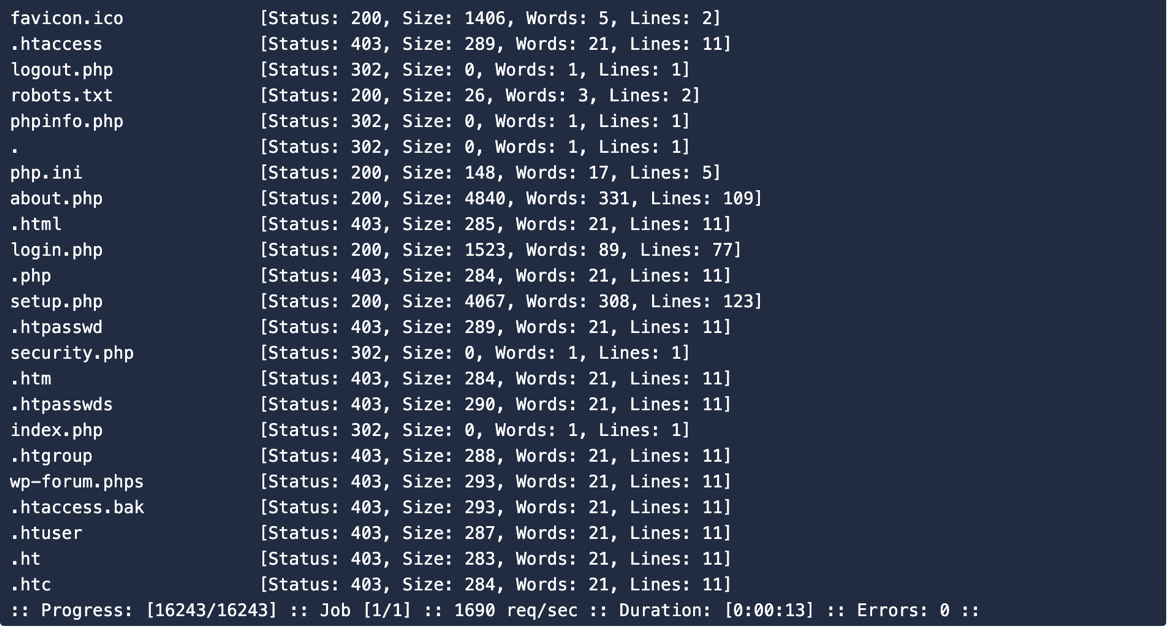
Các tên thư mục thường không phụ thuộc vào loại môi trường mà bạn đã liệt kê và thường là điểm bắt đầu tốt trước khi cố gắng fuzz files. Nếu chúng ta muốn fuzz thư mục chúng ta chỉ cần cung cấp 1 wordlist.



**Using filters**

Nhớ lệnh chúng ta chạy ở bài trước chứ:

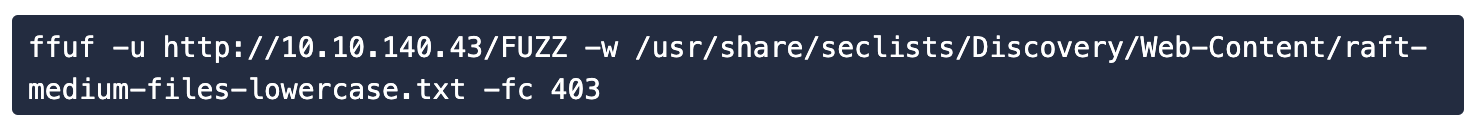




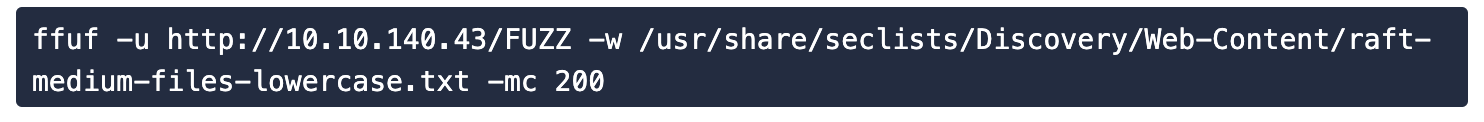
Chúng ta có nhiều đầu ra nhưng không có nhiều cái hữu dụng.

Ví dụ, Http status code 403 chỉ ra chúng ta bị cấm truy cập vào mã nguồn đó. Cùng làm ẩn những phản hồi có trạng thái code là 403 nào.

Bằng việc thêm -fc 403 (filter code) chúng ta sẽ ẩn các đầu ra có trạng thái http là 403.

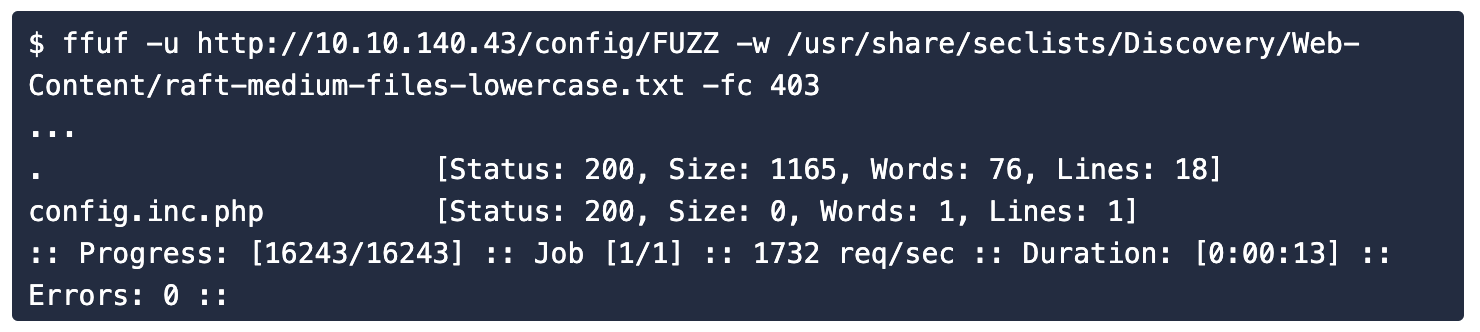


Thỉnh thoảng bạn muốn lọc ra nhiều trạng thái code như 500,302,301,401, vv. Trong trường hợp đó, nếu bạn muốn nhìn thấy các phản hồi có trạng thái 200, bạn có thể dùng -mc 200 (match code) thay vì có 1 danh sách dài các mã cần lọc.



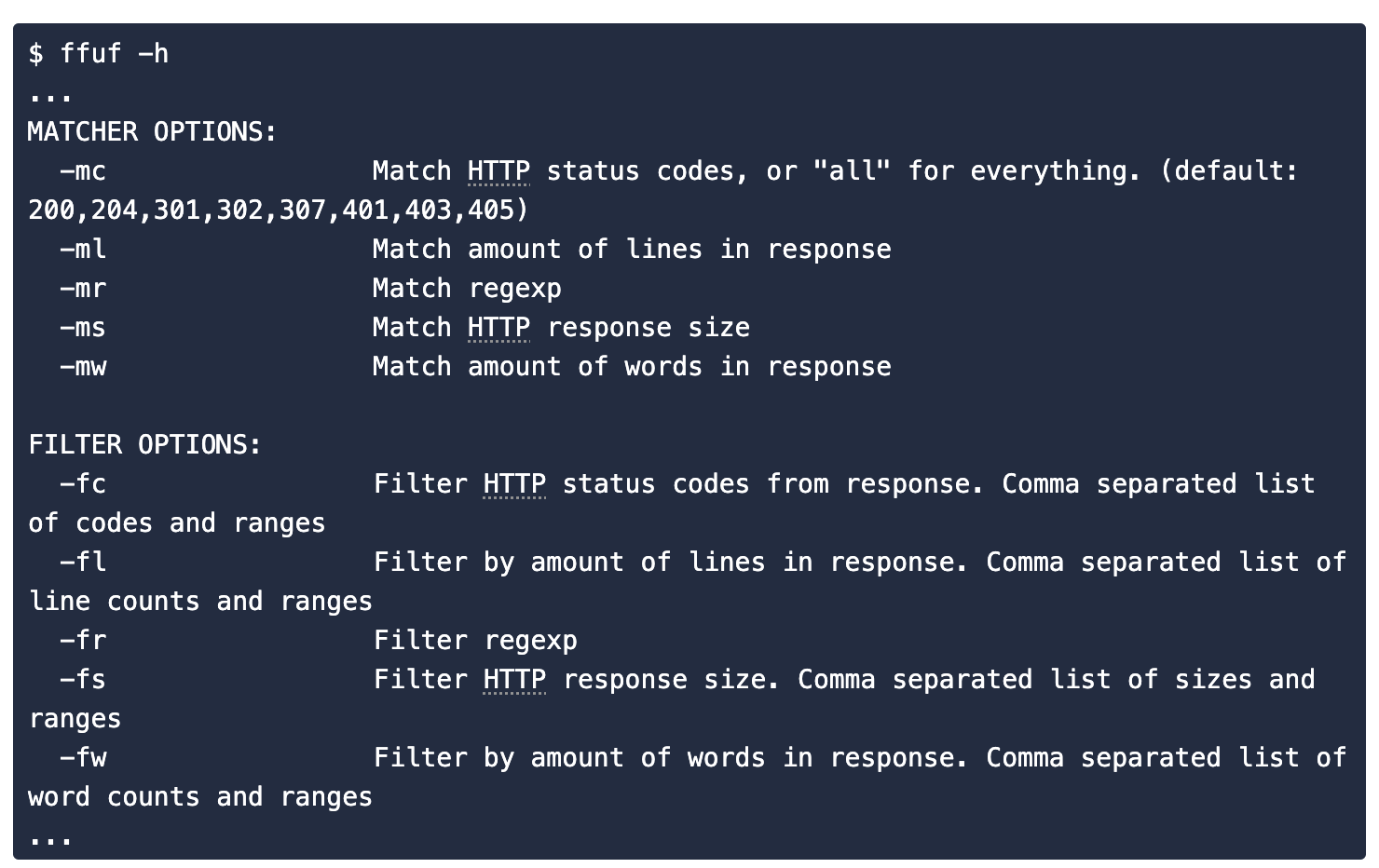
Thỉnh thoảng việc xem các yêu cầu mà máy chủ không xử lí được (có trạng thái code là 500) rất có lợi -mc 500. Tìm kiếm những hành vi bất thường có thể giúp hiểu hơn cách ứng dụng web hoạt động.

Có những filter và matcher khác. Ví dụ, bạn có thể gặp những điểm vào có trạng thái 200 nhưng kích thước phản hồi là 0 byte như functions.php hay inc/myfile.php.

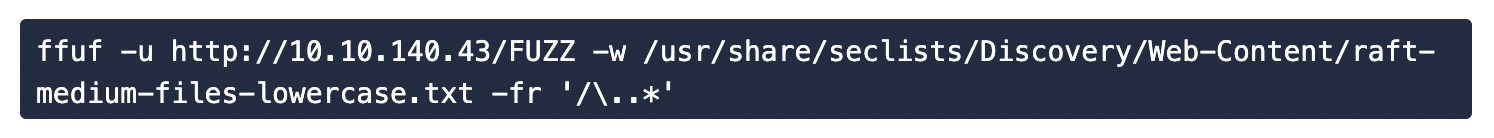


Nếu chúng ta không muốn lfi những loại file không thú vị, chúng ta có thể dùng -fs 0 (filter size).

Đây là tất cả filters và matchers:



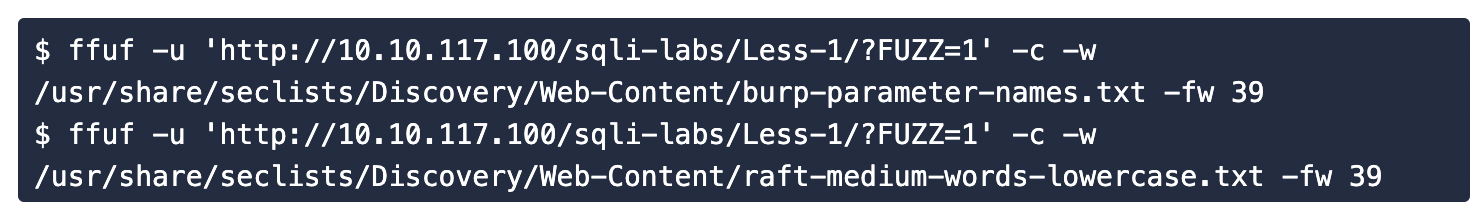
Chúng ta thường thấy dương tính giả với các tệp bắt đầu bằng dấu . (.htgroups, .php, …). Chúng đưa ra lỗi 403, tuy nhiên các tệp này không tồn tại. Rất đáng để dùng -fc 403 nhưng điều này có thể ẩn những tệp quan trọng mà chúng ta không thể truy cập. Thay vào đó chúng ta có thể dùng regexp để phù hợp với tất cả tệp bắt đầu bằng .dot.



**Fuzzing parameters:**

Bạn sẽ làm gì khi bạn tìm thấy 1 trang hoặc điểm vào api nhưng không biết tham số nào được chấp nhận? Bạn fuzz nó.

Khám phá 1 tham số có lỗ hổng có thể dẫn đến file inclusion, path disclosure, xss, sql injection, hoặc command injection. Vì ffuf cho phép bạn đặt keyword ở bất cứ nơi đâu nên chúng ta có thể dùng nó để fuzz tham số.

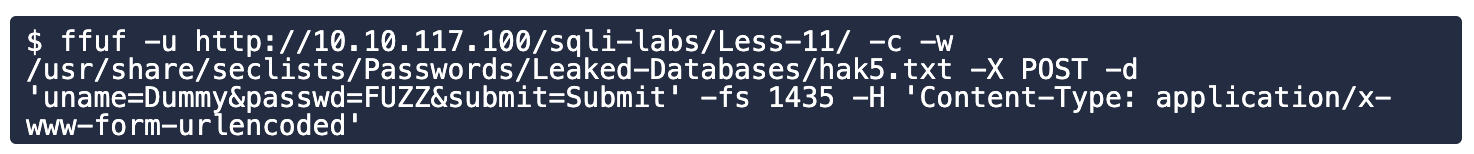


Bây giờ chúng ta đã tìm thấy 1 tham số đã chấp nhận các giá trị số nguyên chúng ta sẽ bắt đầu fuzzing giá trị.

Ở điểm này, chúng ta có thể tạo 1 wordlist và lưu 1 tệp chứa các số nguyên. Để cắt bỏ 1 bước chúng ta có thể dùng -w - yêu cầu ffuf đọc wordlist từ stdout. Điều này sẽ cho phép chúng ta tạo 1 danh sách các số nguyên bằng 1 lệnh do chúng ta chọn sau đó chuyển đầu ra cho ffuf. Dưới là 1 danh sách 5 cách khác nhau để tạo các số từ 0-255.



Chúng ta cũng có thể dùng ffuf để tấn công dồn dập dựa trên wordlist, ví dụ thử mật khẩu trên trang xác thực:

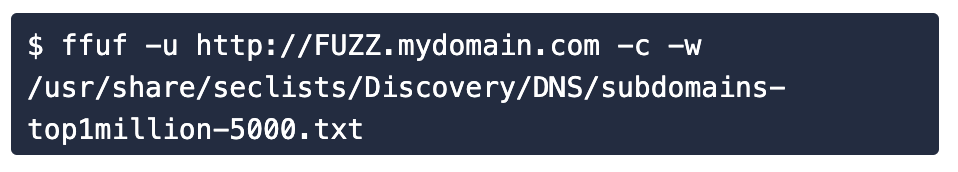


Ở đây chúng ta dùng phương thức POST (được chỉ định bằng -X) và gửi dữ liệu (với -d) nơi mà chúng ta bao gồm từ khóa FUZZ ở nơi đặt mật khẩu.

Chúng ta cũng có thể chỉ định 1 header tùy chỉnh -H 'Content-Type: application/x-www-form-urlencoded' vì ffuf không đặt header content-type tự động như curl làm được.

**Finding vhosts and subdomains**

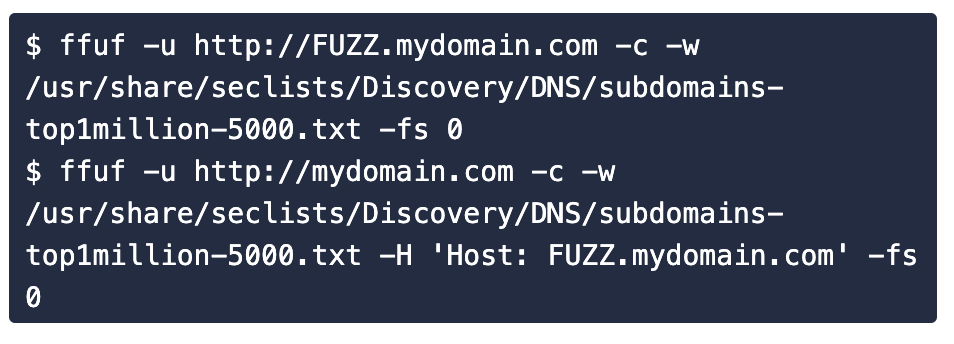
ffuf có thể không hiệu quả như những công cụ chuyên dụng khi nó liệt kê tên miền con nhưng nó có thể làm được.



Một vài tên miền con có thể không được resolve bởi máy chủ DNS mà bạn đang dùng và chỉ được resolve trong mạng cục bộ bởi các máy chủ DNS riêng. Vì vậy một vài máy chủ ảo có thể tồn tại với private subdomain nên lệnh trước sẽ không tìm thấy chúng. Để tìm kiếm private subdomains chúng ta sẽ phải dùng Host http header để các yêu cầu có thể được chấp nhận bởi máy chủ web.

Lưu ý: các máy chủ ảo (vhosts) là tên được dùng trong Apache httpd nhưng với Nginx nó là Server Blocks.

Bạn có thể so sánh các kết quả nhận được từ quá trình liệt kê tên miền con và vhosts:

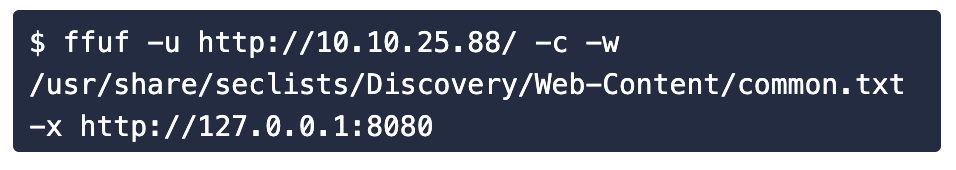


Ví dụ, bạn không thể tìm 1 tên miền con khi liệt kê subdomain trực tiếp nhưng có thể thấy nó khi liệt kê vhost.

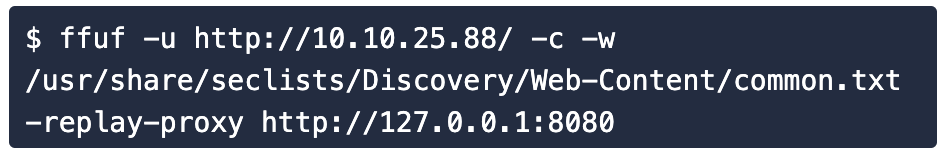
Kĩ thuật liệt kê vhost không nên được xem nhẹ vì nó có thể dẫn đến việc khám phá ra nội dung không được phép truy cập từ bên ngoài.

**Proxifying ffuf traffic**

Cho dù nó là network pivoting hay sử dụng BurpSuite plugins bạn có thể gửi tất cả lưu lượng của ffuf qua 1 web proxy (http hay SOCKS5).



Cũng có thể chỉ gửi những cái phù hợp đến proxy của bạn để lặp lại:



Điều này có thể rất hữu ích nếu bạn không cần tất cả lưu lượng đi qua 1 upstream proxy và muốn giảm thiểu việc sử dụng tài nguyên hoặc tránh ngộ độc lịch sử proxy của bạn.

**Username Enumeration:**

Thực hiện câu lệnh sau:

ffuf -w /usr/share/wordlists/SecLists/Usernames/Names/names.txt -X POST -d "username=FUZZ&email=x&password=x&cpassword=x" -H "Content-Type: application/x-www-form-urlencoded" -u http://10.10.72.192/customers/signup -mr "username already exists"

-w: chọn vị trí file trên máy tính chứa danh sách tên người dùng

-X: phương thức của yêu cầu (mặc định là Get)

-d: dữ liệu chúng ta gửi, đặt giá trị của username là FUZZ.

-H: thêm header cho yêu cầu

-mr: văn bản trên trang để xác thực tên người dùng hợp lệ

**Brute force:**

ffuf -w valid\_usernames.txt:W1,/usr/share/wordlists/SecLists/Passwords/Common-Credentials/10-million-password-list-top-100.txt:W2 -X POST -d "username=W1&password=W2" -H "Content-Type: application/x-www-form-urlencoded" -u http://10.10.72.192/customers/login -fc 200

Phân tích câu lệnh trên:

Ở ví dụ trước ta dùng từ khóa FUZZ để chọn vị trí chèn wordlists, nhưng vì chúng ta dùng nhiều wordlists, chúng ta phải tự chỉ định từ khóa FUZZ của mình : W1, W2.

-fc: kiểm tra trạng thái code của http lớn hơn 200.

**Lưu ý:** ffuf không có match code 404 nên khi scan, không có http response 404 🡪 Bạn nên quét thêm các phản hồi có trạng thái 404 (-mc 404).

**Hydra**

**Khái niệm:**

Hydra là 1 chương trình bẻ khóa mật khẩu trực tuyến mạnh mẽ, 1 công cụ hack mật khẩu đăng nhập của hệ thống 1 cách nhanh chóng.

Hydra có thể chạy qua 1 danh sách và tấn công dồn dập một vài dịch vụ xác thực. Tưởng tượng đoán mật khẩu của ai đó một cách thủ công trên 1 dịch vụ xác định (SSH, Web Application Form, FTP hay SNMP) - chúng ta có thể dùng Hydra để chạy qua 1 danh sách các mật khẩu và đẩy nhanh tốc độ cho chúng ta, xác định đúng mật khẩu.

Theo kho chứa chính thức của Hydra, nó có thể tấn công dồn dập các giao thức sau:

“Asterisk, AFP, Cisco AAA, Cisco auth, Cisco enable, CVS, Firebird, FTP, HTTP-FORM-GET, HTTP-FORM-POST, HTTP-GET, HTTP-HEAD, HTTP-POST, HTTP-PROXY, HTTPS-FORM-GET, HTTPS-FORM-POST, HTTPS-GET, HTTPS-HEAD, HTTPS-POST, HTTP-Proxy, ICQ, IMAP, IRC, LDAP, MEMCACHED, MONGODB, MS-SQL, MYSQL, NCP, NNTP, Oracle Listener, Oracle SID, Oracle, PC-Anywhere, PCNFS, POP3, POSTGRES, Radmin, RDP, Rexec, Rlogin, Rsh, RTSP, SAP/R3, SIP, SMB, SMTP, SMTP Enum, SNMP v1+v2+v3, SOCKS5, SSH (v1 and v2), SSHKEY, Subversion, TeamSpeak (TS2), Telnet, VMware-Auth, VNC and XMPP.”

Điều đó cho thấy sự quan trọng của việc dùng 1 mật khẩu mạnh; nếu mật khẩu của bạn phổ biến, không chứa các kí tự đặc biệt và có 8 kí tự trở lên, nó sẽ rất dễ đoán. Một danh sách 100 triệu mật khẩu chứa những mật khẩu phổ biến, vì vậy khi 1 ứng dụng bên ngoài sử dụng 1 mật khẩu dễ dàng để đăng nhập, thay đổi nó từ mặc định! CCTV camera và các thư viện web thường dùng admin:password như thông tin đăng nhập mặc định, rõ ràng là không đủ mạnh.

**Using Hydra**

**Hydra Commands:**

Các lựa chọn chúng ta truyền vào Hydra phụ thuộc vào dịch vụ mà chúng ta muốn tấn công. Ví dụ, nếu chúng ta muốn tấn công dồn dập ftp với username là user và 1 danh sách mật khẩu passlist.txt, chúng ta sẽ dùng lệnh sau:

hydra -l user -P passlist.txt ftp://target-ip

**SSH:**

hydra -l <username> -P <full path to pass> 10.10.46.1 -t 4 ssh

Option Description

-l specifies the (SSH) username for login

-P indicates a list of passwords

-t sets the number of threads to spawn

Ví dụ,

hydra -l root -P passwords.txt 10.10.46.1 -t 4 ssh (Lưu ý: mac không có option -t)

**Post Web Form:**

Chúng ta có thể dùng Hydra để tấn công dồn dập web forms luôn. Bạn bắt buộc biết loại yêu cầu đang tạo: phương thức Get hay Post. Bạn có thể dùng tab Network trong trình duyệt (trong developer tools) để xem các loại yêu cầu hay mã nguồn.

sudo hydra -l <username> -P <wordlist> 10.10.46.1 http-post-form "<path>:<login\_credentials>:<invalid\_response>"

| Option | Description |
| --- | --- |
| -l | the username for (web form) login |
| -P | the password list to use |
| http-post-form | the type of the form is POST |
| <path> | the login page URL, for example, login.php |
| <login\_credentials> | the username and password used to log in, for example, username=^USER^&password=^PASS^ |
| <invalid\_response> | part of the response when the login fails |
| -V | verbose output for every attempt |

Ví dụ: hydra -l <username> -P <wordlist> 10.10.46.1 http-post-form "/:username=^USER^&password=^PASS^:F=incorrect" -V (Lưu ý: mac không có option -V)

* Trang đăng nhập ở /
* username là mục mà username được nhập
* username được xác định sẽ thay thế ^USER^
* password là mục mà password được nhập
* Các mật khẩu được cung cấp sẽ thay thế ^PASS^
* Cuối cùng, F=incorrect là 1 chuỗi xuất hiện trong phản hồi của máy chủ khi đăng nhập thất bại.

**Web Enumeration**

**Gobuster**

| **Flag** | **Long Flag** | **Description** |
| --- | --- | --- |
| -t | --threads | Number of concurrent threads (default 10) |
| -v | --verbose | Verbose output |
| -z | --no-progress | Don't display progress |
| -q | --quiet | Don't print the banner and other noise |
| -o | --output | Output file to write results to |

**“dir” Modes:**

Cho phép người dùng liệt kê thư mục trang web.

gobuster dir -u http://10.10.10.10 -w /usr/share/wordlists/dirbuster/directory-list-2.3-medium.txt

| **Flag** | **Long Flag** | **Description** |
| --- | --- | --- |
| -c | --cookies | Cookies to use for requests |
| -x | --extensions | File extension(s) to search for |
| -H | --headers | Specify HTTP headers, -H 'Header1: val1' -H 'Header2: val2' |
| -k | --no-tls-validation | Skip TLS certificate verification |
| -n | --no-status | Don't print status codes |
| -P | --password | Password for Basic Auth |
| -s | --status-codes | Positive status codes |
| -b | --status-codes-blacklist | Negative status codes |
| -U | --username | Username for Basic Auth |

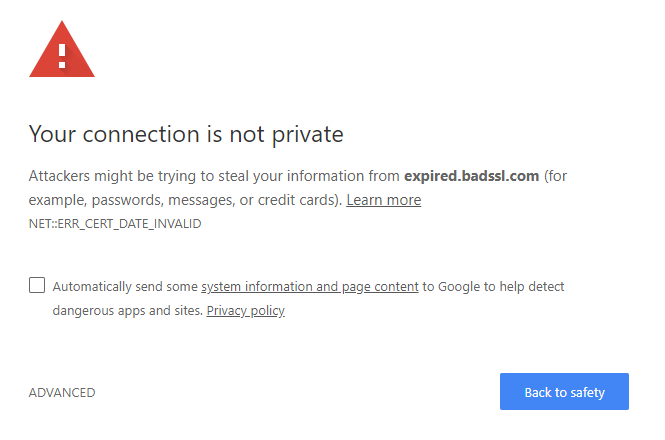
Gobuster có thể dùng -x để tìm kiếm nội dung của những thư mục bạn đã liệt kê bằng việc cung cấp 1 danh sách phần mở rộng tập tin.

gobuster dir -u http://10.10.252.123/myfolder -w /usr/share/wordlists/

dirbuster/directory-list-2.3-medium.txt -x .html,.css,.js

**Flag -k:**

Cờ -k rất đặc biệt vì nó có công dụng quan trọng trong quá trình kiểm thử. Nếu https được phép, bạn sẽ gặp phải lỗi chứng chỉ không hợp lệ như dưới:



Nếu bạn chạy Gobuster không có flag -k, nó sẽ không trả về phản hồi gì. Đừng lo lắng, chỉ cần thêm flag -k, bạn sẽ bỏ qua được chứng chỉ không hợp lệ và tiếp tục quét.

**“DNS” Mode:**

gobuster dns -d mydomain.thm -w /usr/share/wordlists/SecLists/Discovery/DNS/

subdomains-top1million-5000.txt

| **Flag** | **Long Flag** | **Description** |
| --- | --- | --- |
| -c | --show-cname | Show CNAME Records (cannot be used with '-i' option) |
| -i | --show-ips | Show IP Addresses |
| -r | --resolver | Use custom DNS server (format server.com or server.com:port) |

**“vhost” Mode**

Chế độ cuối cùng chúng ta sẽ tập trung là vhost. Nó cho phép Gobuster tấn công dồn dập máy chủ ảo. Máy chủ ảo là những trang web khác nhau trên cùng 1 máy ảo. Trong 1 số trường hợp, nó trông giống như tên miền con. Rất đáng để Gobuster quét ở chế độ này. Bạn không bao giờ biết, nó có thể tìm được 1 vài thứ.

gobuster vhost -u http://example.com -w /usr/share/wordlists/SecLists/

Discovery/DNS/subdomains-top1million-5000.txt

**Wpscan**

**Nikto**

**Putting it all together**

**Đọc source HTML**

* Xem các thư mục lưu trữ tập tin (một số thư mục bị lỗi index do cấu hình sai)
* Xem tập tin quan trọng (css,js,php,…)
* Xem đường dẫn Url
* Kiểm tra Comment

**Tìm thư mục bị ẩn**

Level 1: Recon thư mục

ffuf -w /usr/share/wordlists/dirbuster/directory-list-1.0.txt -u http://target-ip/FUZZ/ -c

Bạn sẽ dùng 2 wordlists:

* directory-list-1.0.txt
* directory-list-2.3-medium.txt

Khi bạn gặp thư mục có trạng thái 403, tiếp tục recon level 2

Level 2: Recon file trong thư mục

ffuf -w /usr/share/wordlists/dirb/common.txt -u http://target-ip/folder/FUZZ -c -e .aspx

**Tìm subdomain và vhost**

ffuf -u http://FUZZ.mydomain.com -c -w /usr/share/Seclists/Discovery/DNS/subdomains-top1million-5000.txt

ffuf -u http://mydomain.com -c -w /usr/share/Seclists/Discovery/DNS/subdomains-top1million-5000.txt -H 'Host: FUZZ.mydomain.com'

DNS Zone Transfer

dig axfr mydomain.com @target-ip

**Tìm username và password**

Thu thập tất cả tên người dùng và mật khẩu

cewl -d 3 -m 6 --lowercase -w wordlists.txt http://target-ip

Nếu gặp hash password thì dùng john để crack

**Tìm công nghệ và phiên bản sử dụng**

Chú ý tên, phiên bản của webapp, máy chủ web, ngôn ngữ server-side

* Public Exploit
* Tài khoản mặc định của ứng dụng
* Tệp cấu hình
* Cấu trúc thư mục của webapp