# ???????

**Fuff**

FFUF легко адаптируется к внешнему инструментарию. Подставляет наборы из словарей в соответствующие точки адреса и анализирует ответ.

```
ffuf -u http://test.ru/FUZZ/ -w dict.txt
```

Здесь вместо FUZZ будет подставлено каждое значение из dict.txt. По умолчанию GET, затем анализ статуса ответа

**Мультисловарь**

```
ffuf -u https://ignitetechnologies.in/W2/W1/ -w dict.txt:W1 -w dns_dict.txt:W2
```

Общие параметры

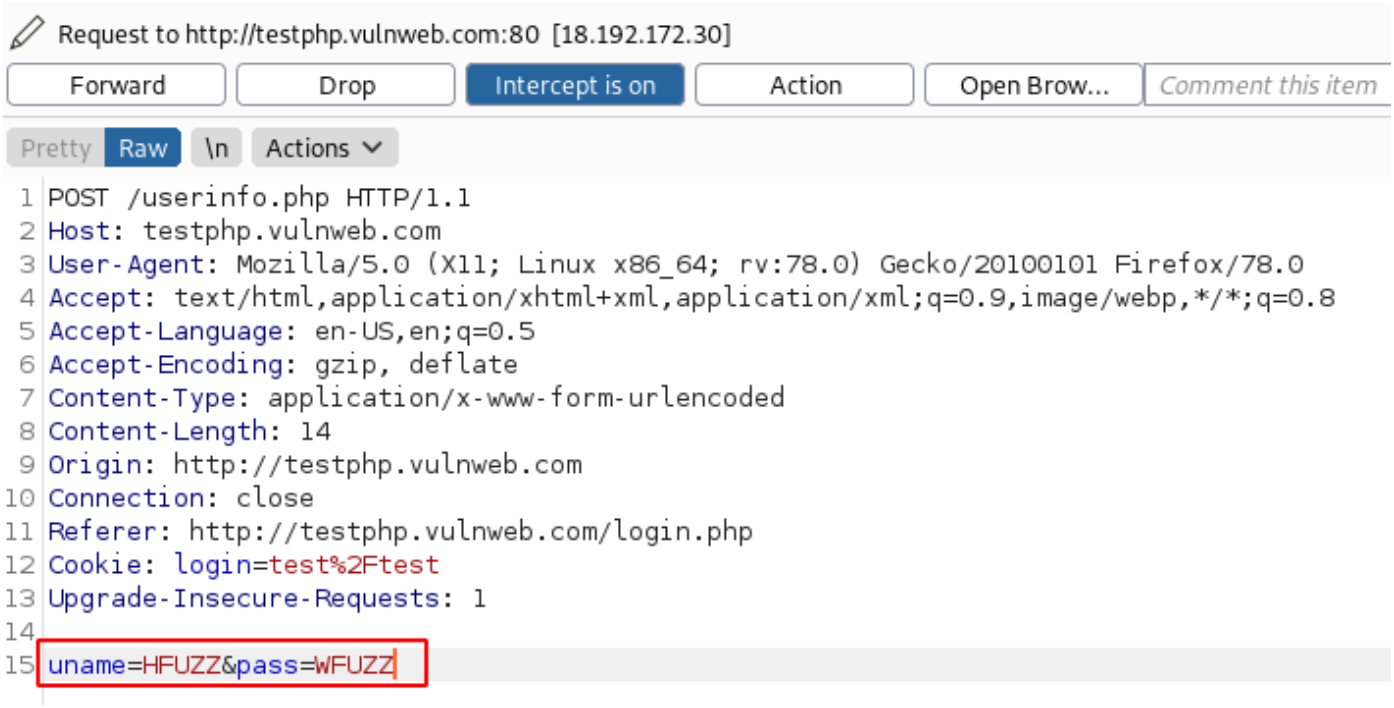
| | |
|---|---|
| -ic -s | скрывает из вывода баннер fuff и дополнительную информацию, оставляя только найденные пути |
| -e .php | определяет расширение файла, которое мы ищем. |
| -b "" | Установка cookie |
| -p 1 | Задержка 1 секунда перед отправкой следующего запроса |
| -rate 500 | Максимальное количество запросов в секунду |
| -t 1000 | Количество потоков. По умолчанию 40. |
| -o fname -of format | Выходной файл и формат результата<br>-of html<br>-of csv |

Параметры соответствия / фильтрации

| | |
|---|---|
| -mc / -fc | Какие статусы отображать. например -mc 200 / Какие статусы фильтровать |
| -ml / -fl | Соответствие / исключение количеству строк в ответе |
| -mw / -fw | Соответствие / исключение количеству слов в ответе |
| -ms / -fs | Соответствие / исключение размера ответа |
| -mr / -fr | Соответствие / исключение в ответе регулярному выражению |

Можно настроить proxy

Атака clusterbomb - подстановка логинов и паролей из словаря в соответствующие поля запроса для подбора. Идея в том, что в параметре request указывается текст запроса. При помощи burp копируем текст запроса, в нашем случае в brute.txt заменяя значения на переменные




Затем задаем тип атаки -mode и словари для переменных.

```
ffuf -request brute.txt -request-proto http -mode clusterbomb -w users.txt:HFUZZ -w
pass.txt:WFUZZ -mc 200
```

**Типы атак:**

| | |
|---|---|
| clusterbomb | Полный перебор. Берет все элементы из первого набора и комбинирует со всеми элементами из второго, третьего и т.д. Идеален для перебора логина и пароля. |
| pitchfork | Попарное сопоставление. Берет первый элемент из первого набора, первый из второго и т.д., затем второй из первого, второй из второго. Используется для согласованных данных. |
| sniper | Снайпер (режим по умолчанию). Использует один набор полезных нагрузок для атаки на одну или несколько позиций по очереди. |
| batteringram | Таран. Использует один набор полезных нагрузок и подставляет одно и то же значение во все позиции одновременно в одном запросе. |

| | |
|---|---|
| null | "Холостой" режим. Отправляет запрос только один раз, без замены в позициях. Полезные нагрузки не используются. Получить "чистый" ответ от сервера без каких-либо атакующих payloads, чтобы использовать его как базовый для фильтрации. |
| multinull | Многократный "холостой" режим. Отправляет один и тот же базовый запрос несколько раз. Стабильность ответа: Проверить, возвращает ли сервер всегда один и тот же ответ на идентичный запрос. Помогает отсеять случайные отклонения. |
| single | Одиночная замена. Использует один набор полезных нагрузок, но подставляет каждую нагрузку только в первую позицию FUZZ. Последующие позиции FUZZ в запросе игнорируются. Упрощенный вариант Sniper, когда у вас в запросе случайно оказалось несколько слов FUZZ, но вы хотите фаззить только первую позицию. |
| iterator | Нужен файл смарт-загрузки (JSON) Позволяет создать цепочку из других режимов атаки. Например, сначала выполнить clusterbomb для двух наборов, а затем результат передать в sniper для третьего. Очень гибкий, но и сложный в настройке. |
| trampoline | Нужен файл смарт-загрузки (JSON) Позволяет использовать результат одного запроса (например, извлечь токен из ответа) в качестве полезной нагрузки для следующего запроса. Идеален для обхода защиты, требующей выполнения предыдущего шага (например, получения CSRF-токена перед отправкой формы). |

T e fuff повторяет типы Burb, добавляя свои варианты. Iterator и trampoline являются аналогами скриптов в Burp. Возможно настройка менее удобная.

**Список словарей:**

- https://github.com/empty-jack/YAWR раздел web -> files and directories -> fuzz.txt
- SecLists (GitHub) — самый большой и универсальный набор: директории, файлы, пароли, URL, поддомены и т.п. git clone https://github.com/danielmiessler/SecLists.git.
- Assetnote — Wordlists wordlists.assetnote.io — наборы, регулярно обновляемые для content-/subdomain-discovery. Есть web-interface и репозиторий.
- commonspeak2-wordlists (Assetnote / GitHub) — wordlists, сгенерированные из больших публичных датасетов (подходит для content discovery и поддоменов). git clone https://github.com/assetnote/commonspeak2-wordlists.
- FuzzDB (GitHub / проект) — словари атак, шаблоны и предсказуемые пути — полезно для более «потенциально опасных» и специальных путей.

- PayloadsAllTheThings (GitHub / сайт) — коллекция полезных payload’ов и bypass-паттернов (не просто имена файлов, а полезные полези для инъекций и тестов).
- DirBuster wordlists (архив/репозитории) — классические wordlists, используемые DirBuster / Dirsearch (хороши для brute-директорий). Есть архивы и отдельные репозитории с наборами.
- Kali / пакеты wordlists & seclists — Kali поставляет готовые пакеты (seclists, wordlists с rockyou и т.д.), удобно ставить прямо через apt на тестовой машине.
- rockyou.txt (популярный парольный словарь) — классика для перебора паролей; часто нужен при тестах аутентификации. Доступен в архивах wordlists (Kali) и на mirror-сайтах. weakpass.com
- Assetnote commonspeak / blog & наборы рекомендаций по поддоменам — гайды и готовые наборы поддоменов (полезно сочетать с автоматикой обновления).
- Доп. коллекции и агрегаторы (Payloads-and-wordlists, другие GitHub-репы) — небольшие коллекции и специализированные листы (например, payload-наборы для Burp / Intruder). Полезны для конкретных задач.

---

Revision #9
Created 20 September 2025 15:21:26 by Admin
Updated 6 October 2025 05:16:21 by Admin