Penetration Test Report

Diary for travellers

April, 2018

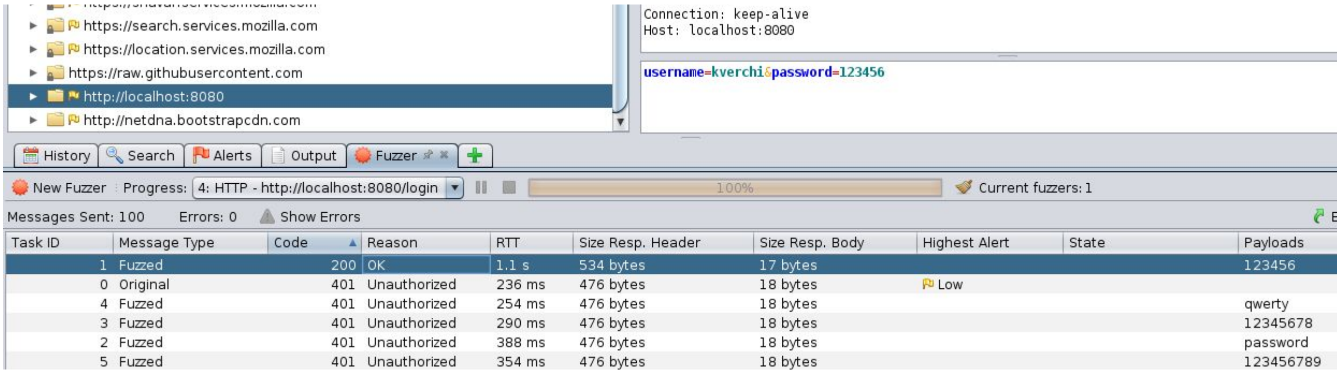
Table of contents

Attack narrative

Attacking Authentication

Bad passwords

The authentication mechanism of web application Diary for travellers requires at least 6 symbols for password. But it allows users to use weak passwords, such as common dictionary words or names and the same as the username, as illustrated in Figure 1.



- https://search.services.mozilla.com
- https://location.services.mozilla.com
- https://raw.githubusercontent.com
- http://localhost:8080
- http://netdna.bootstrapcdn.com

```
Connection: keep-alive
Host: localhost:8080
```

```
username=kverchi&password=123456
```

Progress: 4: HTTP - http://localhost:8080/login 100% Current fuzzers: 1

Messages Sent: 100 Errors: 0

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
1	Fuzzed	200	OK	1.1 s	534 bytes	17 bytes			123456
0	Original	401	Unauthorized	236 ms	476 bytes	18 bytes	Low		
4	Fuzzed	401	Unauthorized	254 ms	476 bytes	18 bytes			qwerty
3	Fuzzed	401	Unauthorized	290 ms	476 bytes	18 bytes			12345678
2	Fuzzed	401	Unauthorized	388 ms	476 bytes	18 bytes			password
5	Fuzzed	401	Unauthorized	354 ms	476 bytes	18 bytes			123456789

Figure 1: An account with weak password

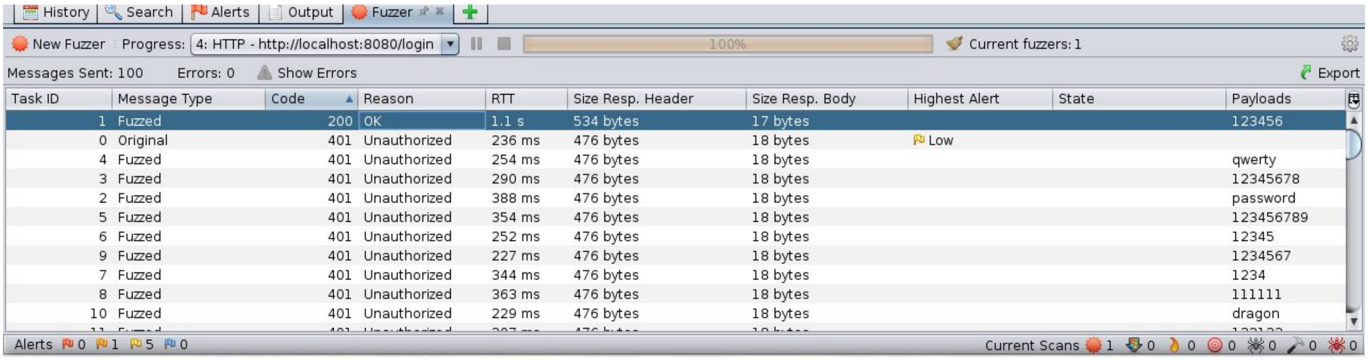
Solving bad passwords issue

The application need to use strong credentials. Password must contain:

- alphabetic, numeric, and typographic characters
- the appearance of both uppercase and lowercase characters
- the avoidance of dictionary words, names, and other common passwords
- preventing a password from being set to the username
- preventing a similarity or match with previously set passwords [1]

Brute-Forcible Login

As it’s shown in Figure 2, application allows an attacker to make repeated login attempts with different passwords until he guesses the correct one.



Progress: 4: HTTP - http://localhost:8080/login 100% Current fuzzers: 1

Messages Sent: 100 Errors: 0

Task ID	Message Type	Code	Reason	RTT	Size Resp. Header	Size Resp. Body	Highest Alert	State	Payloads
1	Fuzzed	200	OK	1.1 s	534 bytes	17 bytes			123456
0	Original	401	Unauthorized	236 ms	476 bytes	18 bytes	Low		
4	Fuzzed	401	Unauthorized	254 ms	476 bytes	18 bytes			qwerty
3	Fuzzed	401	Unauthorized	290 ms	476 bytes	18 bytes			12345678
2	Fuzzed	401	Unauthorized	388 ms	476 bytes	18 bytes			password
5	Fuzzed	401	Unauthorized	354 ms	476 bytes	18 bytes			123456789
6	Fuzzed	401	Unauthorized	252 ms	476 bytes	18 bytes			12345
9	Fuzzed	401	Unauthorized	227 ms	476 bytes	18 bytes			1234567
7	Fuzzed	401	Unauthorized	344 ms	476 bytes	18 bytes			1234
8	Fuzzed	401	Unauthorized	363 ms	476 bytes	18 bytes			111111
10	Fuzzed	401	Unauthorized	229 ms	476 bytes	18 bytes			dragon
11	Fuzzed	401	Unauthorized	207 ms	476 bytes	18 bytes			123123

Alerts 0 1 5 0 Current Scans 1 0 0 0 0 0 0

Figure 2: Unlimited attempts to guess a password

Solving Brute-Forcible Login issue

Web application must implement policy of temporary account suspension to prevent brute-force attacks. In addition, an application can be protected through the use of CAPTCHA (Completely Automated Public Turing test to tell Computers and Humans Apart) challenges on every page that may be a target for brute-force attacks [1].

Conclusion

Risk rating

Recommendations