Contents

SECTION - HTTP Proxies

Overview of Section - HTTP Proxies

- What is an HTTP Proxy?
- Example HTTP Proxies?
- Why use an HTTP Proxy?
- How to direct REST Client through Proxy?
  - Inspect Traffic
  - Filter Traffic (System Proxies)
  - Port Config
  - Replay Request
- Fuzzing

Exercises: in browser - GET, viewing traffic, Ajax requests

What is an HTTP Proxy?

- HTTP Proxy captures HTTP Traffic
- Allows replay of requests
- Allows manipulation of responses

Which proxies?

- Fiddler
  - Windows (Beta: Linux, Mac)
- Charles

  - Commercial but allows 30 mins in 'shareware' mode
- BurpSuite
  - Free edition good enough for API Testing
- Owasp ZAP
  - Open Source

Fiddler & Charles act as System Proxies making them easy to use with Postman.

Why use when Testing API?

- Record requests
- Create evidence of your testing
- Replay requests outside of client tool
- Fuzzing

Using a Proxy with Postman

- Change Proxy Settings with "File Settings" and then "Proxy" tab
  - on Mac use "Postman Preferences" and then "Proxy" tab
- Postman can use a Global Proxy by setting the IP address and Port
  - e.g. BurpSuite, OWasp Zap (or Fiddler and Charles)
- Postman can hook into System Proxy e.g
  - Charles, Fiddler
- Otherwise start postman with `--proxy-server`

Using a Proxy with Postman

Setting Postman proxy from commandline

For full details see blog post

- Mac (type all on one line):

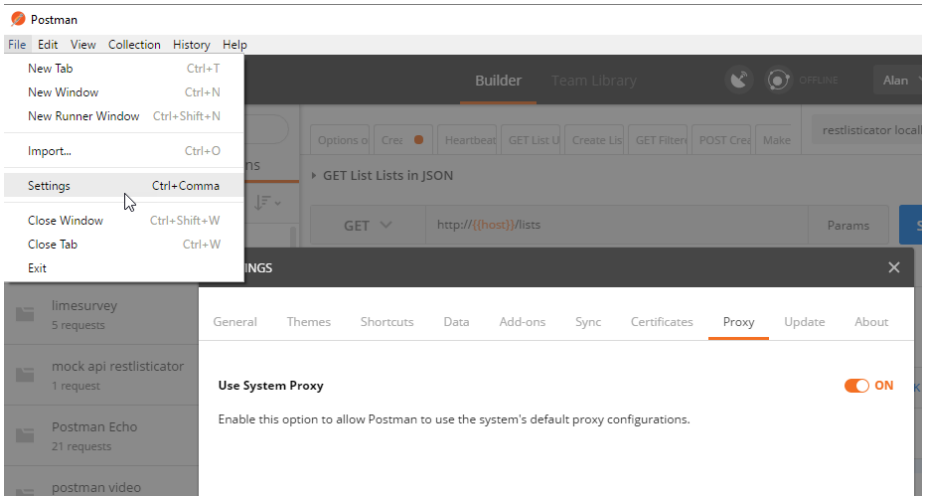

Figure 1: Using a Proxy with Postman

```
open /Applications/Postman.app --args
      --proxy-server=localhost:8888
```

- Windows:

```
cd C:\Users\Alan\AppData\Local\Postman\app-4.9.3\
postman.exe --proxy-server=localhost:8888
```

Using a Proxy with Insomnia

- Application Preferences
- on Mac "Insomnia preferences"

Use of Proxies

- Examples of Fiddler - with screenshots
- Examples of Charles with screenshots
- Examples of Owasp Zap with screenshots
- Examples of BurpSuite with screenshots

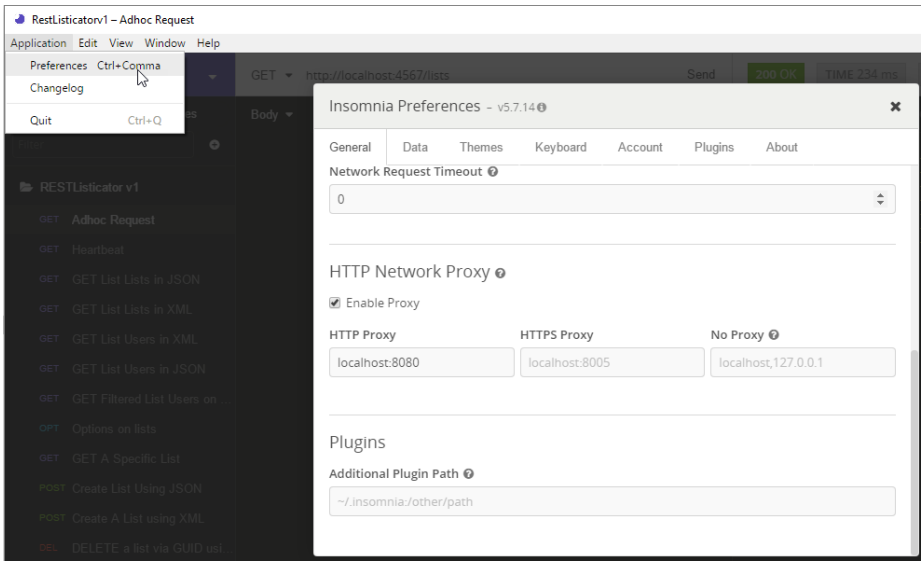

Figure 2: Using a Proxy with Insomnia

Fiddler Filter Requests

- ctrl+X - to clear traffic history
- filter by process

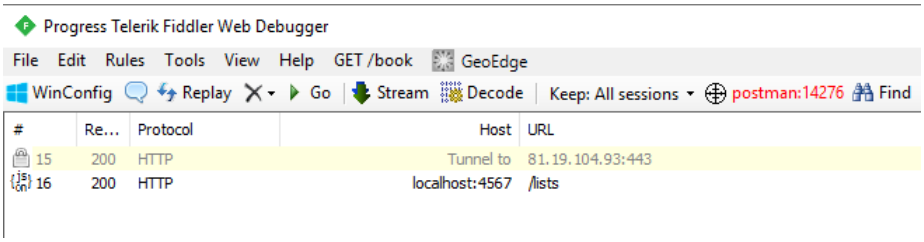

Figure 3: Fiddler Filter Requests

Fiddler Inspect Traffic

- Traffic shown in list - use inspectors to view request and response

---

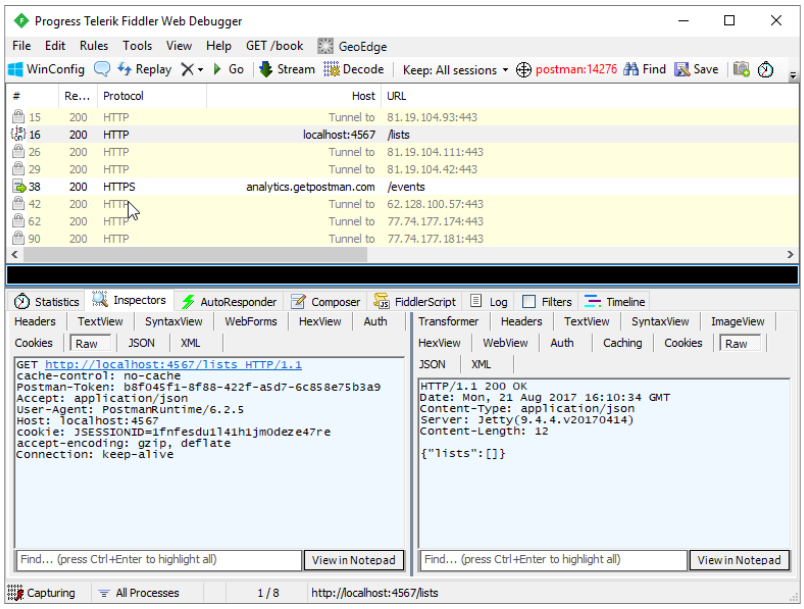

Figure 4: Fiddler Inspect Traffic

Fiddler listens on port 8888 by default

- find out/change port in `tools \ options`

Replay Request in Fiddler

- drag request from history to Composer to edit and replay

Charles Filter Requests

- Quick filter to `localhost`

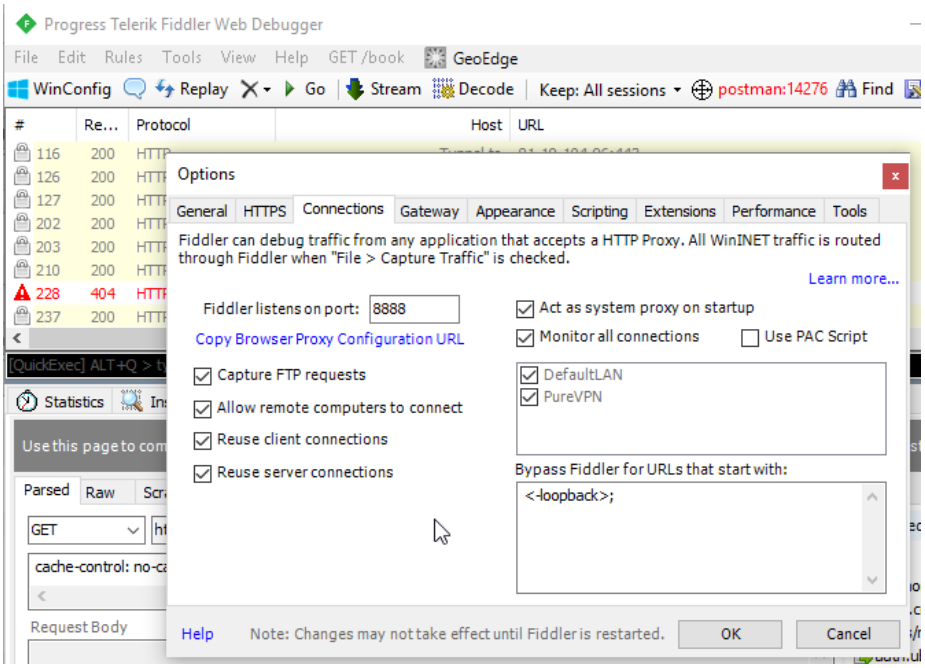

Figure 5: Fiddler listens on port 8888 by default

Charles Inspect Traffic

- inspect traffic

Charles Replay Traffic

- right click 'compose'

Charles port settings

- `tools \ proxy settings`

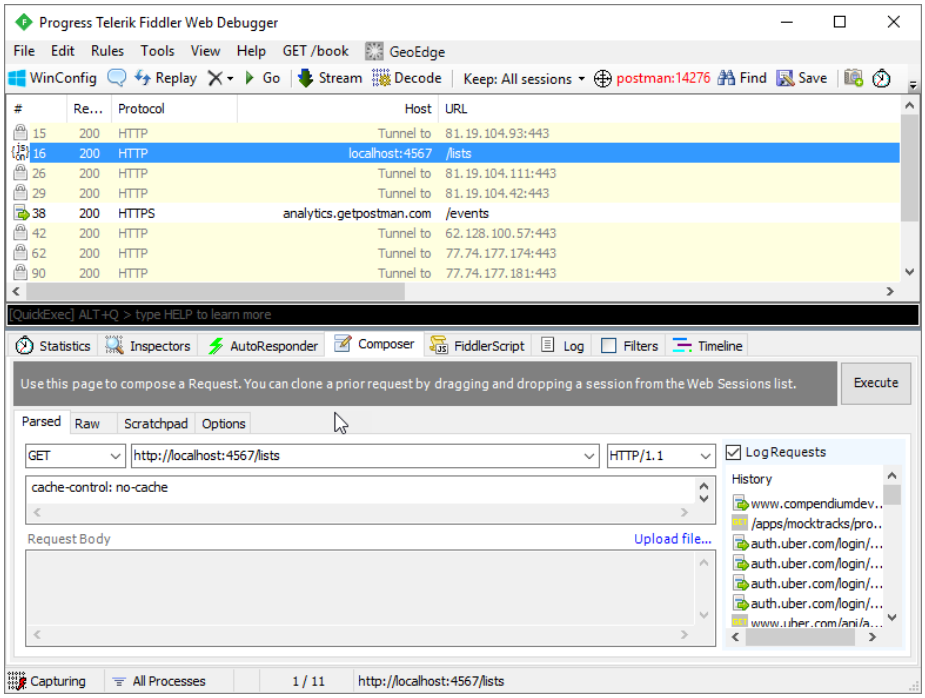

Figure 6: Replay Request in Fiddler

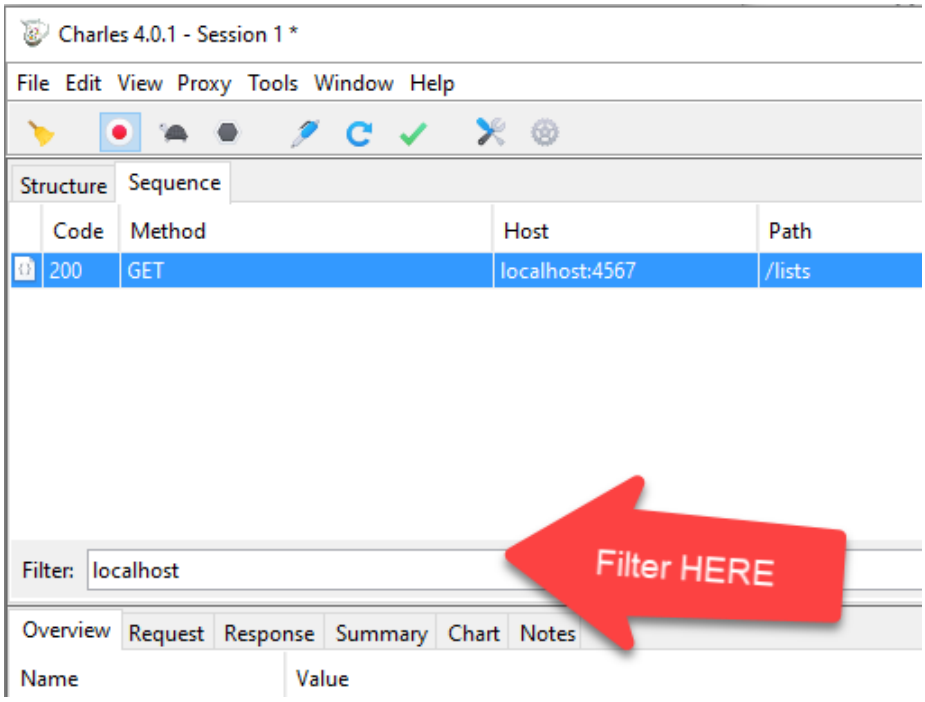

Figure 7: Charles Filter Requests

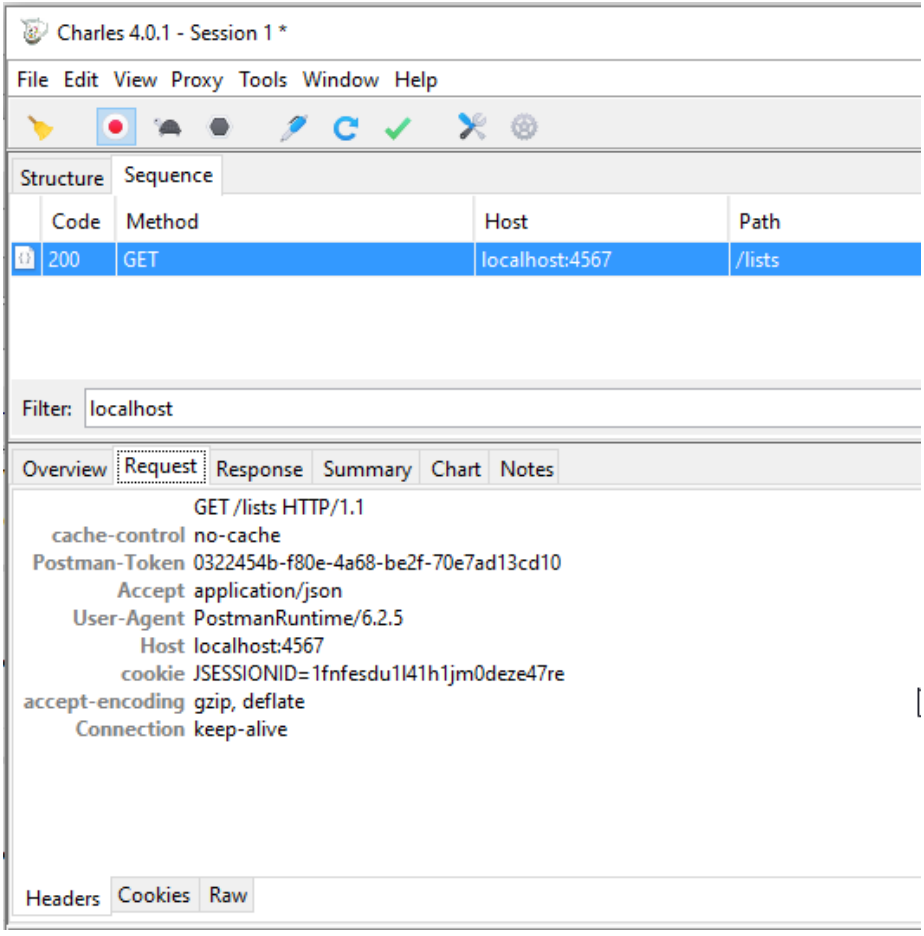

Figure 8: Charles Inspect Traffic

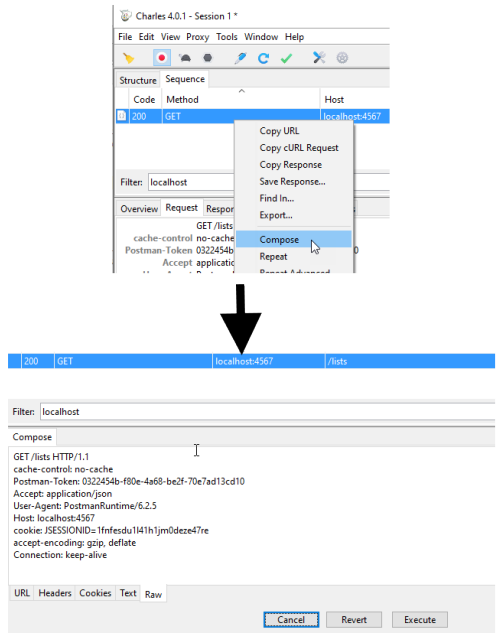

Figure 9: Charles Replay Traffic

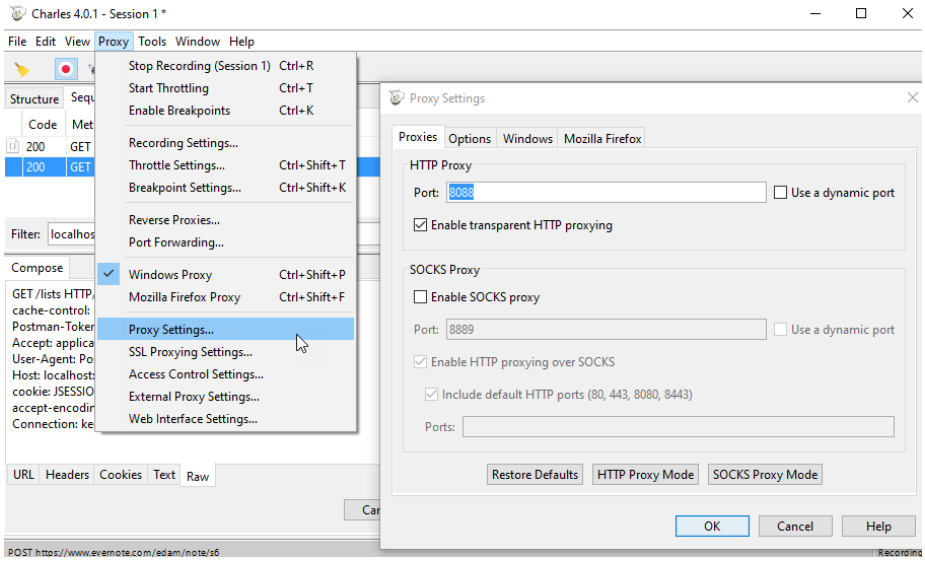

Figure 10: Charles port settings

BurpSuite Port Config

- tabs `proxy \ options`

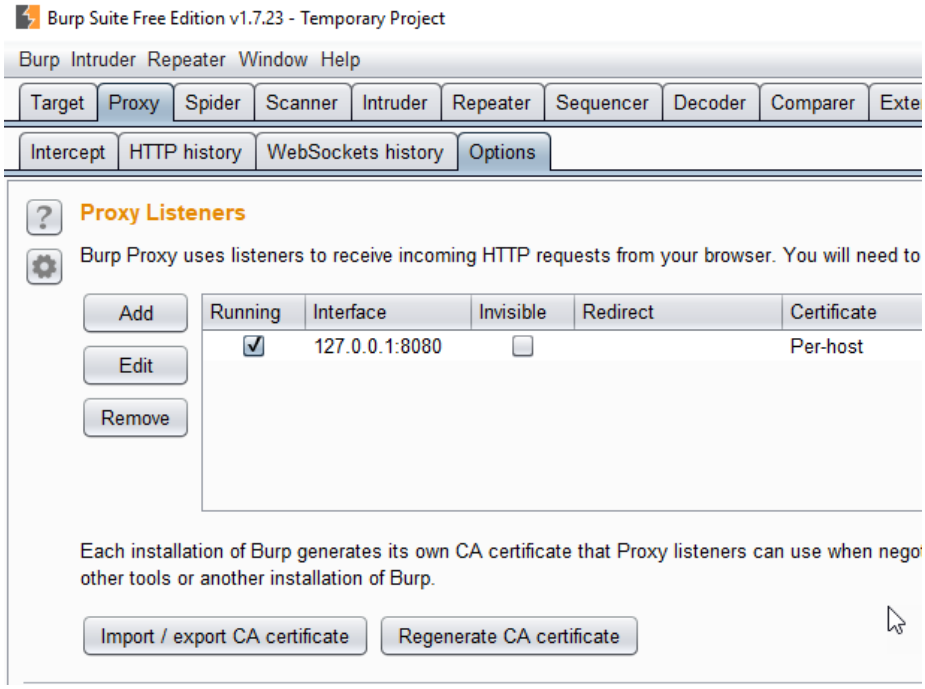

Figure 11: BurpSuite Port Config

BurpSuite Inspect Traffic

- Ensure intecept is off, view HTTP History

BurpSuite Replay Request

- right click and use repeater

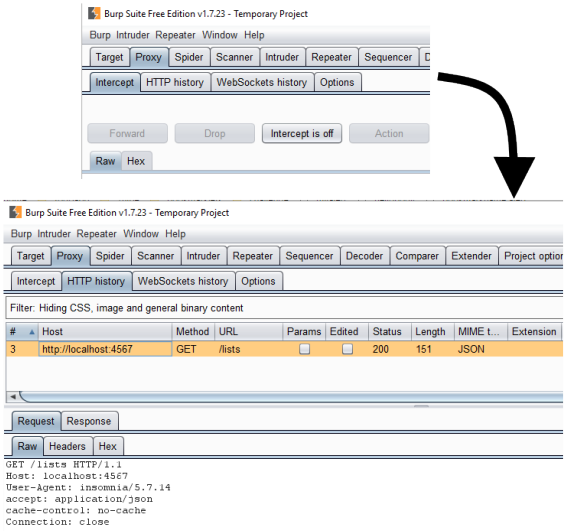

Figure 12: BurpSuite Inspect Traffic

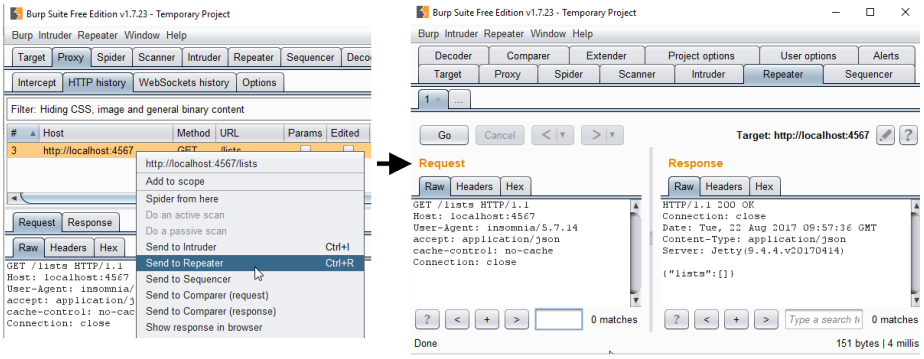

Figure 13: BurpSuite Replay Request

Owasp Zap Port Config

- `tools \ options \ local proxy`

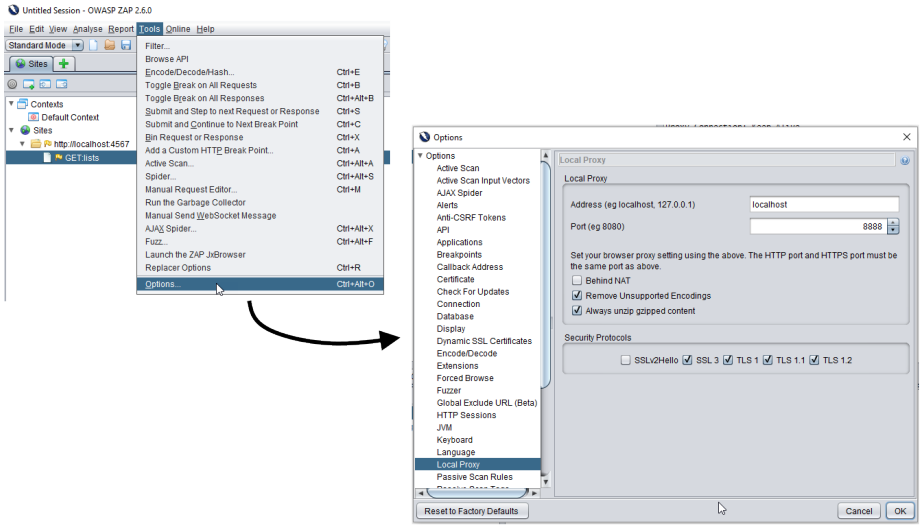

Figure 14: Owasp Zap Port Config

Owasp Zap Inspect Traffic

- history and 'sites', view in top right

Owasp Zap Replay Request

- right click 'Resend', edit and send

Fuzzing in BurpSuite

- right click, intruder, hightlight and add position, edit payload, start attack

---

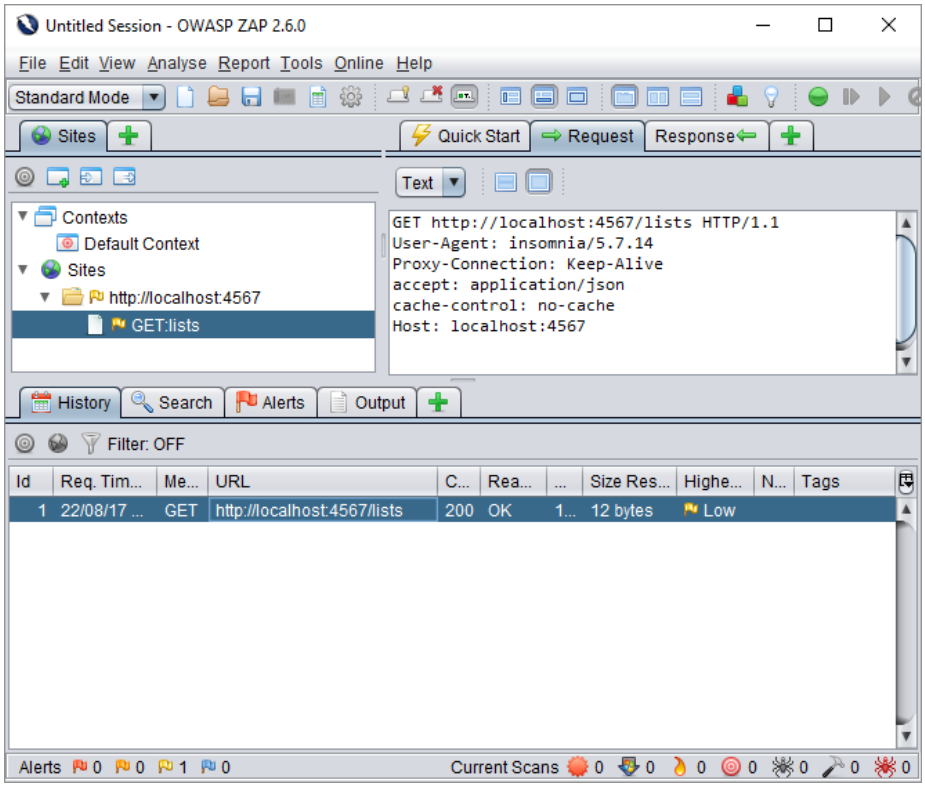

Figure 15: Owasp Zap Inspect Traffic

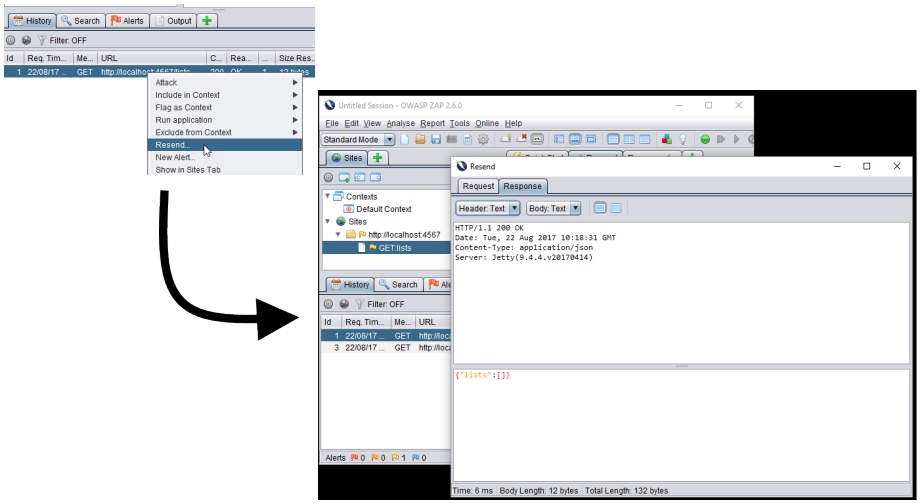

Figure 16: Owasp Zap Replay Request

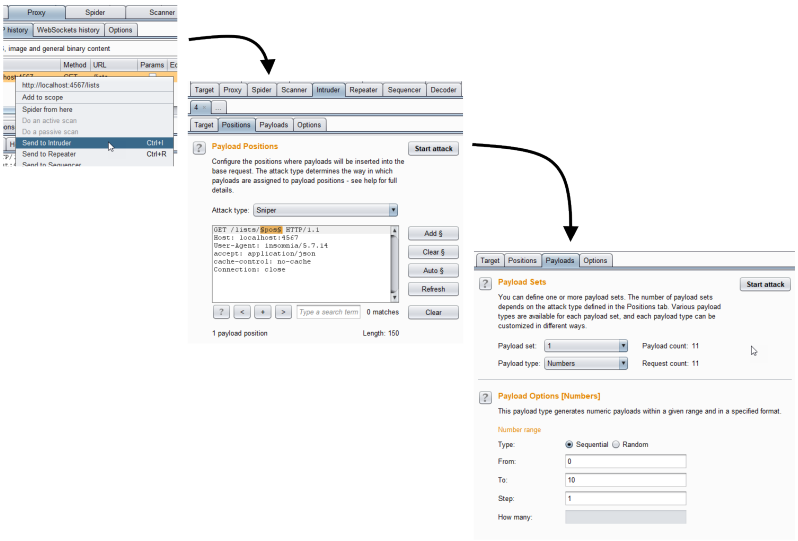

Figure 17: Fuzzing in BurpSuite

Fuzzing in Owasp ZAP

- right click, attack, fuzz, highlight and add location, add payload, start fuzzer

---

Demo

- Insomnia send request through proxy
- Replay
- Use Fuzzer

Exercises:

- send requests through proxy
- replay amended requests
- use fuzzer to experiment
- see expanded Exercises in Exercises section

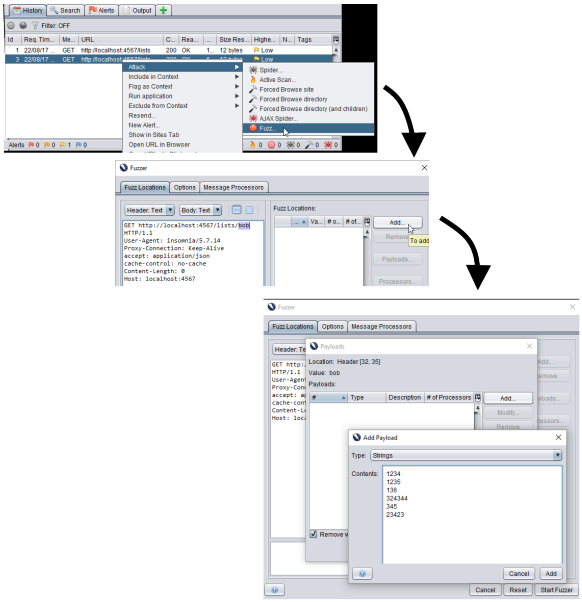

Figure 18: Fuzzing in Owasp ZAP