Automation tricks for Burp Suite Pro

Nicolas Grégoire aka @Agarri_FR

Barb'hack 2023

Intro

Who am I?

Nicolas Grégoire 🇫🇷

Twitter → `@Agarri_FR` and `@MasteringBurp`
Bluesky → `agarri.bsky.social`
Email → `nicolas.gregoire@agarri.fr`

Official Burp Suite training partner

Mostly for Europe 🇪🇺
100+ trainees per year (either on-site and online)

More free resources

NahamCon 2023 workshop

NorthSec 2023 talk

Old blog posts

Third-party resources

https://hackademy.agarri.fr/freebies

What is the plan?

Setup

WiFi, Burp Suite Pro and slidedeck

Intro to session handling rules

Terminology and default behavior

Session management for Web apps

Handling CSRF tokens in a generic way

Session management for Web APIs

Using static, dynamic or cached authz headers

Setup

Setup

WiFi

Network `MBSP`
Password `AbracadabrA`

DHCP

DNS at `192.168.42.101`
Gateway at `192.168.42.254`
Test with `ping [WHATEVER].pwn`

Burp Pro installers, license and slides

http://www.attacker.pwn/shared/

Intro to session handling rules

Survey

Who ever used the Scanner?

Who ever used session handling rules?

Terminology

Cookie jar

Place where the cookies are stored (like in a browser)

Macro

Sequence of HTTP requests

Execution is triggered by a session handling rule

Session handling rule

Define when and how sessions are managed

May execute one or several macros (among other things)

Session tracer

Live debugger for macros and rules

Default behavior

Proxy

Cookies are written to the cookie jar

Scanner

Cookies are read from the cookie jar

How to easily improve scan coverage?

Add Extensions to the scope of the default rule

Except when using authorization-testing extensions!

Session management for Web apps

Session management for Web apps

Generic handling of CSRF tokens

Overview

Get a fresh anti-CSRF token via a macro

Parameter `csrf_token` is automatically extracted

Update the token in the original request

All requests read from the cookie jar

Use a single thread

Configure the macro

Select the token-fetching request

Using the macro recorder

Set the cookie to a visually invalid value

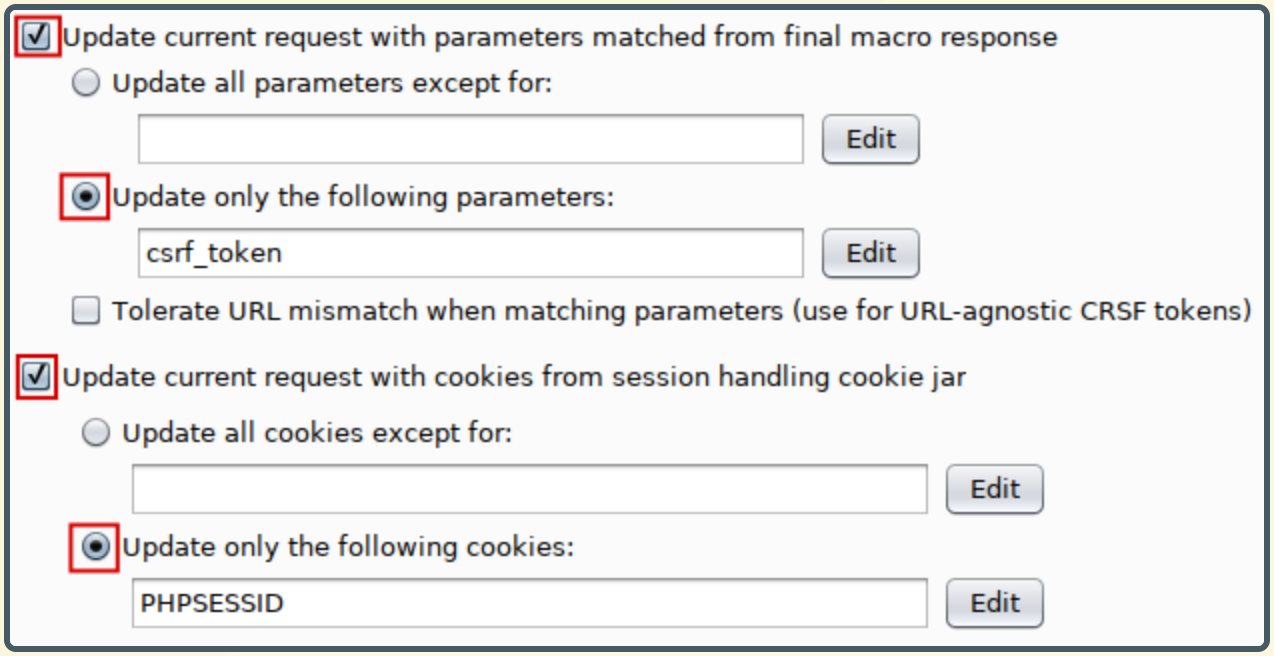
Click on "Configure item"

Add received cookies to the cookie jar

Use cookies from the cookie jar

No need to configure any data extraction

Configure the rule



Mini App - Chall 02

Using Intruder

Attack type: "Sniper"

Make sure the rule's scope includes Intruder

Session management for Web APIs

Overview of Juice Shop

Log into the API

URL → `/rest/user/login`

Extract the JWT from the response

Non-greedy regex → `token":"(.*?)"`

Validate the JWT

URL → `/api/users`

Header → `Authorization: Bearer [value]`

Known account

jim@juice-sh.op / ncc-1701 (feel free to create your own)

Session management for Web APIs

Static authz header

Static authz header

Pick a token from your Proxy History

Create a session handling rule

Action "Set a specific header value"

Make sure to include the `Bearer[space]` prefix

Access the API

Session management for Web APIs

Dynamic authz header

Dynamic authz header

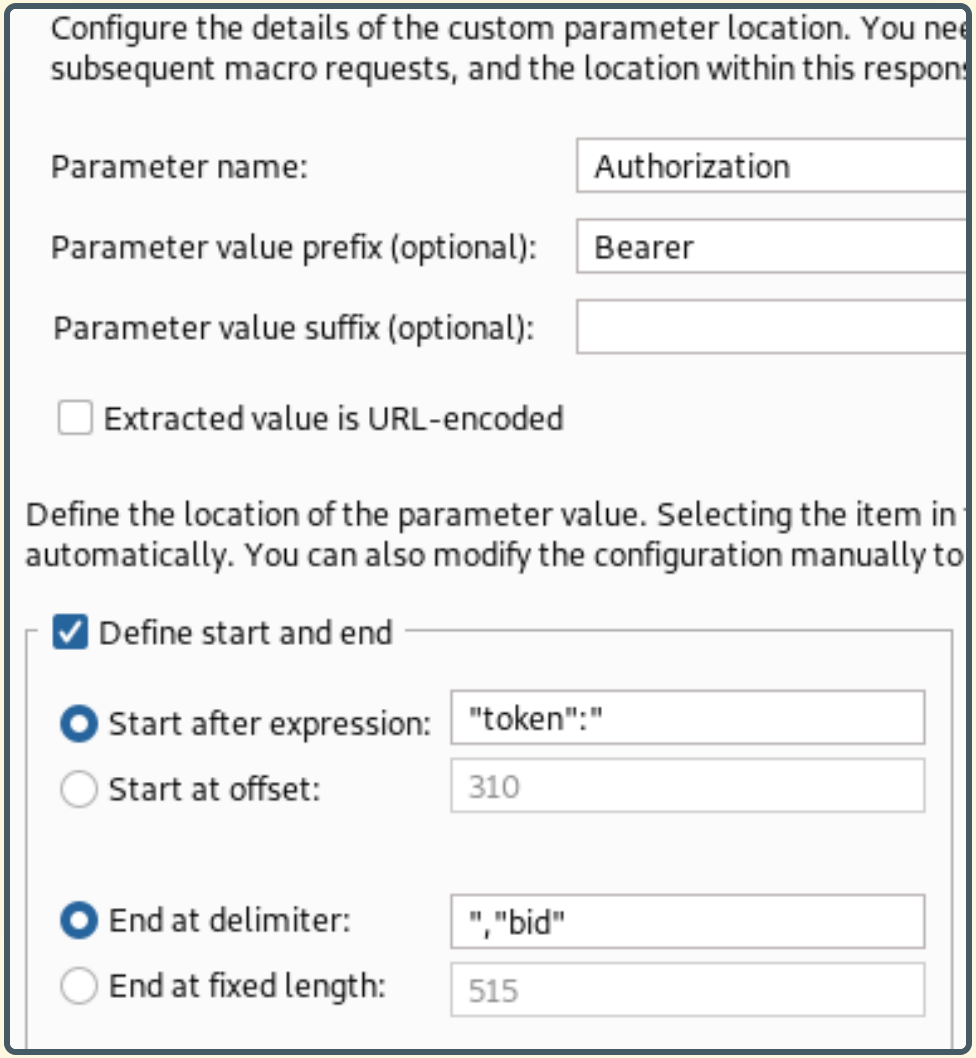
Create a login macro

Configure data extraction

In "Configure item > Custom parameter locations"

- Select the authentication token
- Set the parameter name to `authorization`
- Set the parameter prefix to `Bearer[space]`

Dynamic authz header



Dynamic authz header

Create a session handling rule

Configure its actions

Add action "Set a specific header value"

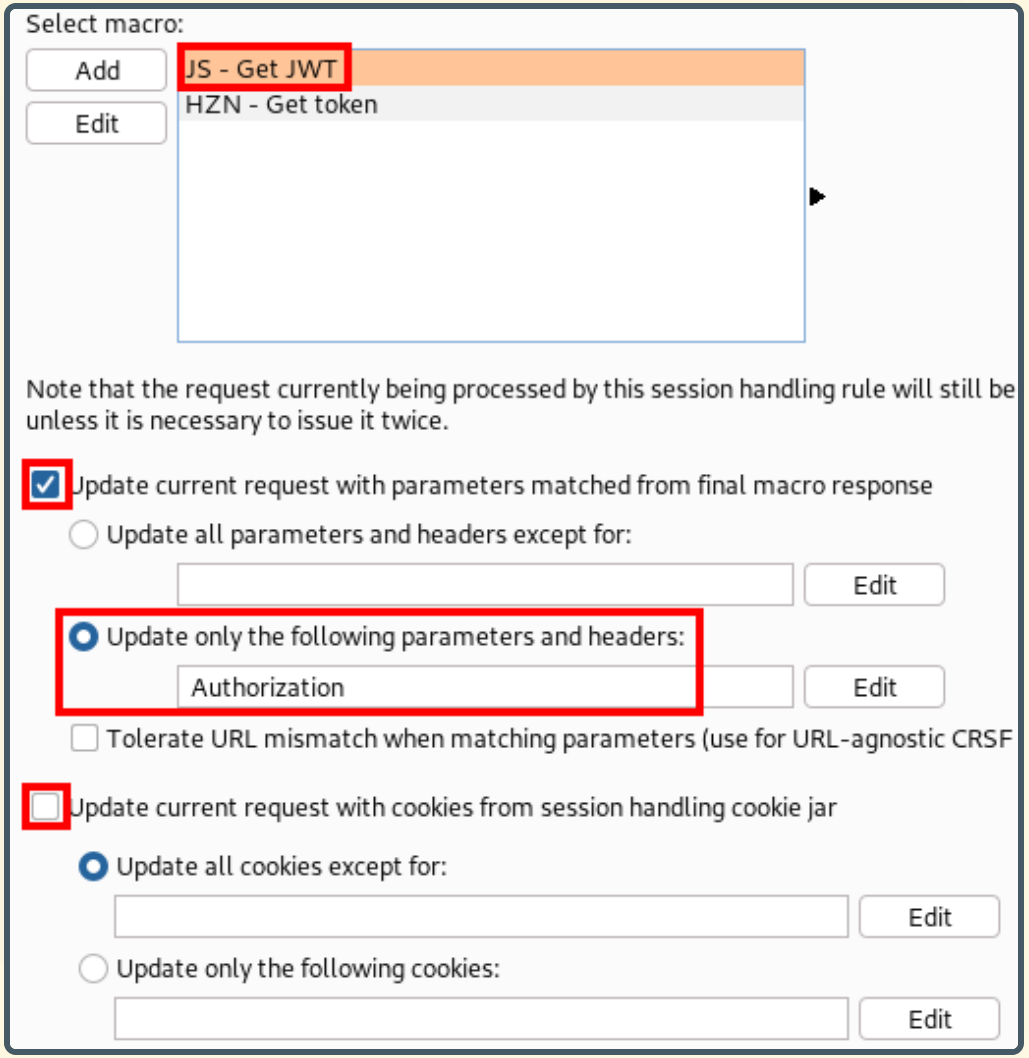
- Make sure to check "Add if not present"

Add action "Run macro"

- Select the login macro
- Update the `Authorization` header

Access the API

Dynamic authz header



Session management for Web APIs

Cached authz header

Cached authz header

Download extension "JWT ReAuth" (v1.0.1)

http://www.attacker.pwn/shared/

https://github.com/nccgroup/jwt-reauth/releases

Load it in Burp Suite

Cached authz header

From "Proxy History"

Send the login request to the extension

- Using action "Set auth request"

Set the extension's scope

- Using action "Add to scope"

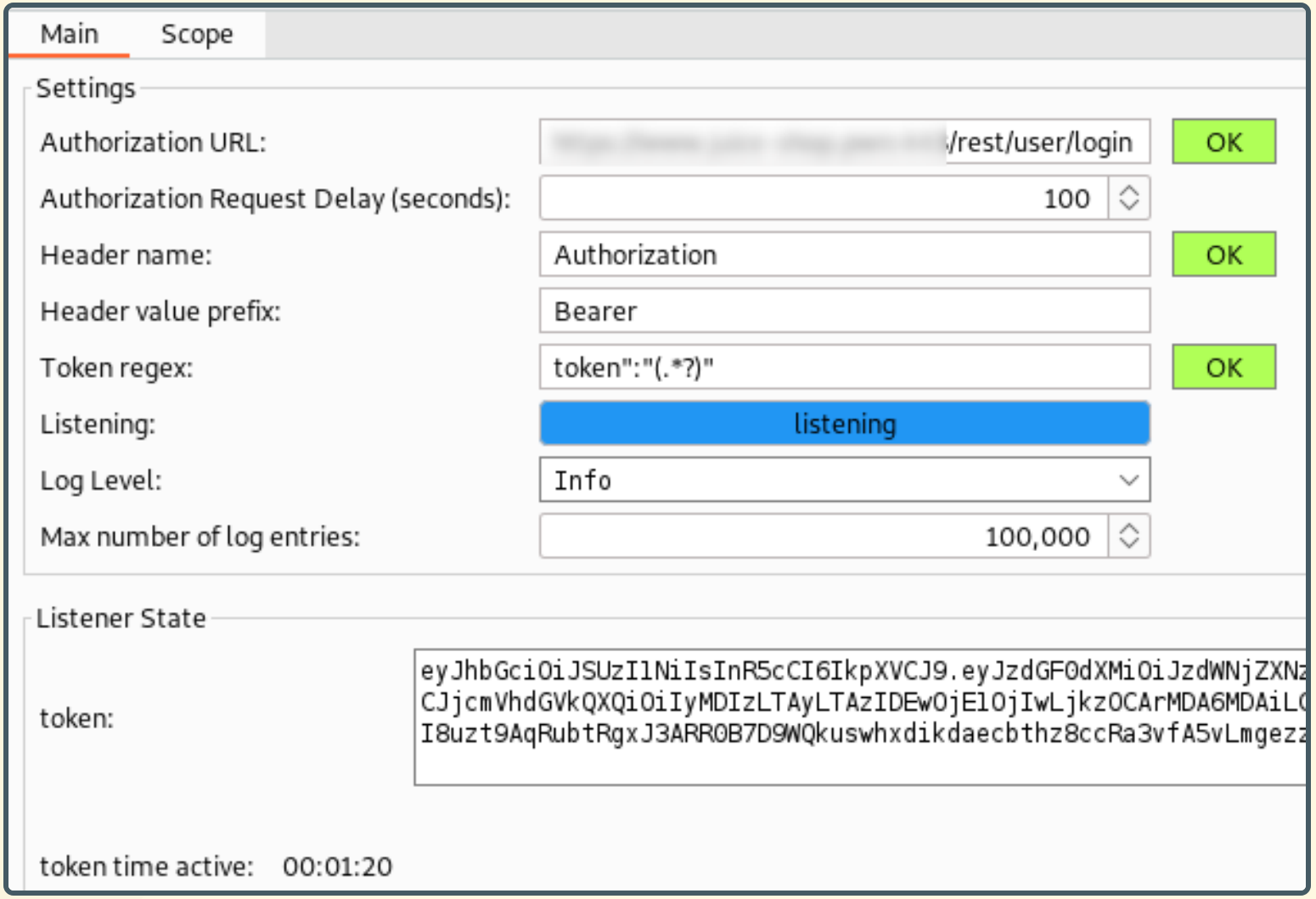
Cached authz header

From the extension's tab

Edit the scope (should end with `/api/`)
Change the "Token regex" field to `token":"(.*?)"`
Change the "Authorization Request Delay" field to `100`
Enable processing (click on `Not listening`)

Access the API

Cached authz header



Thanks for attending!

Now, enjoy the CTF... 😉