OWASP Raider: a novel framework for manipulating HTTP processes of persistent sessions

Daniel Neagaru

2023-04-18 Tue, OWASP Hamburg Stammtisch

Outline

$ whoami

Daniel Neagaru

- Working at Akkodis (prev. Modis) for 1+ year
- Penetration Tester for 5+ years
- 5+ years in IT mostly as a sysadmin
- Started building Raider early 2021
- Became an OWASP project leader August 2021

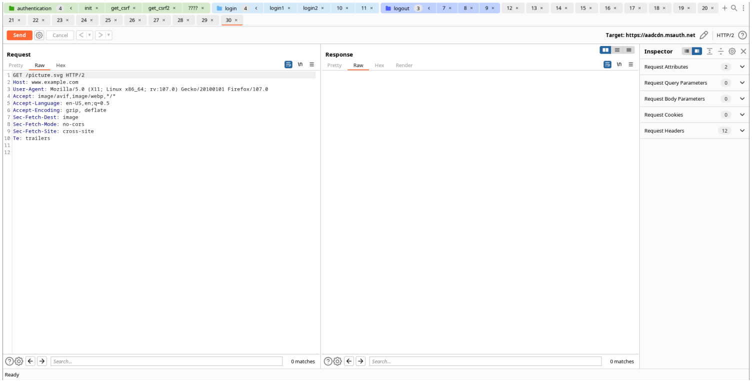
$ whatis raider

Raider was incepted with the goal to help test web authentication systems but has evolved and now can be used for HTTP processes of arbitrary complexity.

- A framework for manipulating HTTP processes
- Defines a DSL to describe the client-server information exchange
- Modular architecture with flexibility in its DNA
- Main code written in Python
- Configuration files written in hylang (LISP)

BurpSuite

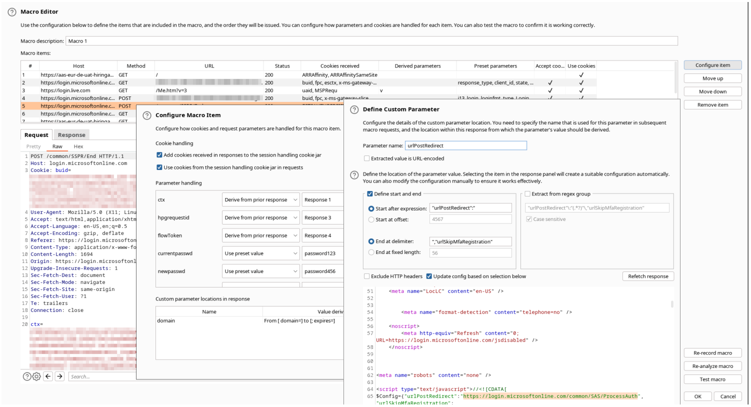
When testing authentication process in BurpSuite I ended up with many poorly organized Repeater tabs.



```
GET /picture.svg HTTP/2
Host: www.example.com
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:107.0) Gecko/20100101 Firefox/107.0
Accept: image/avif,image/webp,*/*
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Sec-Fetch-Dest: image
Sec-Fetch-Mode: no-cors
Sec-Fetch-Site: cross-site
Te: trailers


```

BurpSuite

After figuring out how authentication works, I had to write new BurpSuite Macros.



Macro Editor

Use the configuration below to define the items that are included in the macro, and the order they will be issued. You can configure how parameters and cookies are handled for each item. You can also test the macro to confirm it is working correctly.

Macro description: Macro 1

Macro items:

#	Host	Method	URL	Status	Cookies received	Derived parameters	Preset parameters	Accept coo...	Use cookies
1	https://aas-eur-de-uat-hiringa...	GET	/	200	ARRAffinity, ARRAffinitySameSite				✓
2	https://login.microsoftonline.c...	GET	[redacted]	200	buid, fpc, esctx, x-ms-gateway-...		response_type, client_id, state, ...	✓	✓
3	https://login.live.com	GET	/Me.htm?v=3	200	uaid, MSPRequ	v		✓	✓
4	https://login.microsoftonline.c...	POST	[redacted]	200	buid, fpc, x-ms-gateway-slice		i13, login, loginfmt, type, Login	✓	✓
5	https://login.microsoftonline.c...	POST							
6	https://login.microsoftonline.c...	GET							
7	https://aas-eur-de-uat-hiringa...	GET							

Request | Response

Pretty | Raw | Hex

```
POST /common/SSPR/End HTTP/1.1
Host: login.microsoftonline.com
Cookie: buid=
[redacted]
User-Agent: Mozilla/5.0 (X11; Linu
Accept: text/html,application/xhtm
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: https://login.microsofton
Content-Type: application/x-www-fo
Content-Length: 1694
Origin: https://login.microsoftonl
Upgrade-Insecure-Requests: 1
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Te: trailers
Connection: close

ctx=
[redacted]
```

Configure Macro Item

Configure how cookies and request parameters are handled for this macro item.

Cookie handling

- ✓ Add cookies received in responses to the session handling cookie jar
- ✓ Use cookies from the session handling cookie jar in requests

Parameter handling

ctx	Derive from prior response	Response 1
hpgrequestid	Derive from prior response	Response 3
flowToken	Derive from prior response	Response 4
currentpasswd	Use preset value	password123
newpasswd	Use preset value	password456

Custom parameter locations in response

Name	Value deriv
domain	From [domain=] to [; expires=]

Define Custom Parameter

Configure the details of the custom parameter location. You need to specify the name that is used for this parameter in subsequent macro requests, and the location within this response from which the parameter's value should be derived.

Parameter name: urlPostRedirect

☐ Extracted value is URL-encoded

Define the location of the parameter value. Selecting the item in the response panel will create a suitable configuration automatically. You can also modify the configuration manually to ensure it works effectively.

✓ Define start and end

- ● Start after expression: "urlPostRedirect":"
- ○ Start at offset: 4567
- ● End at delimiter: ","urlSkipMfaRegistration"
- ○ End at fixed length: 56

☐ Extract from regex group

"urlPostRedirect"\:"(.*?)"\,"urlSkipMfaRegistration"

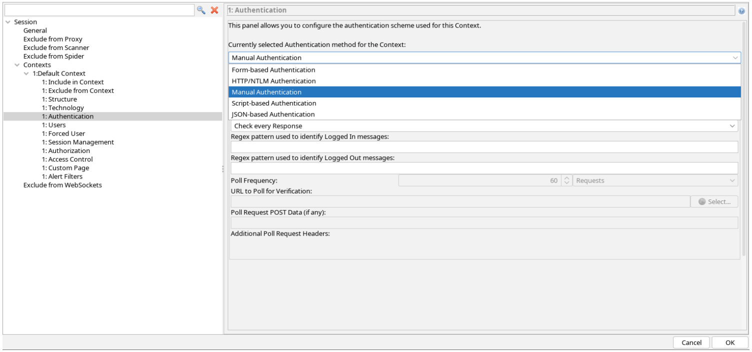
✓ Case sensitive

☐ Exclude HTTP headers ✓ Update config based on selection below

```
    <meta name="LocLC" content="en-US" />


        <meta name="format-detection" content="telephone=no" />

    <noscript>
        <meta http-equiv="Refresh" content="0;
URL=https://login.microsoftonline.com/jsdisabled" />
    </noscript>



<meta name="robots" content="none" />

<script type="text/javascript">//<![CDATA[
$Config={"urlPostRedirect":"https://login.microsoftonline.com/common/SAS/ProcessAuth",
"urlSkipMfaRegistration":
```

ZAProxy

To automate authentication in ZAProxy you have to set up the context properly.



```
Session
  General
  Exclude from Proxy
  Exclude from Scanner
  Exclude from Spider
  Contexts
    1:Default Context
      1: Include in Context
      1: Exclude from Context
      1: Structure
      1: Technology
      1: Authentication
      1: Users
      1: Forced User
      1: Session Management
      1: Authorization
      1: Access Control
      1: Custom Page
      1: Alert Filters
  Exclude from WebSockets
```

1: Authentication

This panel allows you to configure the authentication scheme used for this Context.

Currently selected Authentication method for the Context: Manual Authentication

- Form-based Authentication
- HTTP/NTLM Authentication
- Manual Authentication
- Script-based Authentication
- JSON-based Authentication

Check every Response

Regex pattern used to identify Logged In messages:

Regex pattern used to identify Logged Out messages:

Poll Frequency: 60 Requests

URL to Poll for Verification: Select...

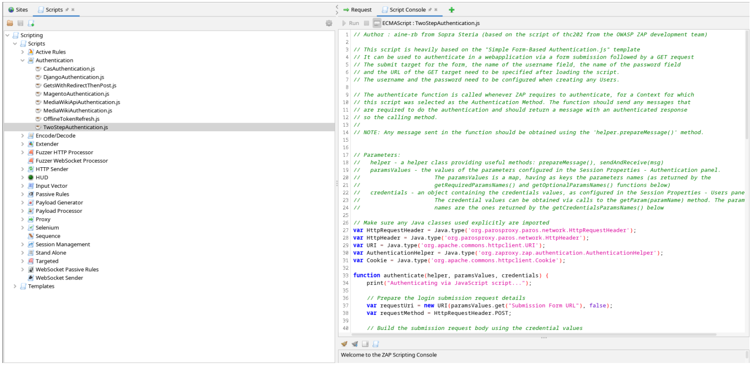
Poll Request POST Data (if any):

Additional Poll Request Headers:

Cancel OK

ZAProxy

ZAProxy provides some authentication scripts.

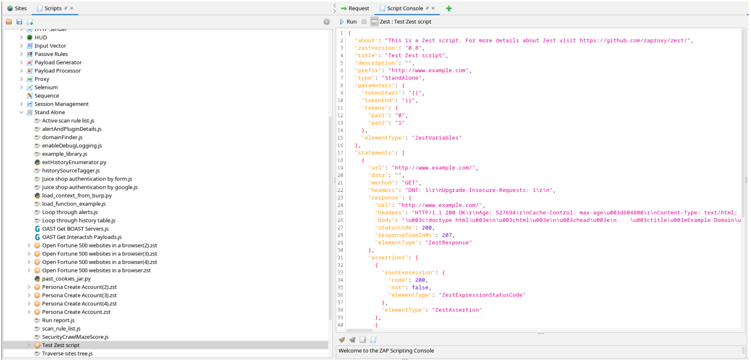


```
// Author : aine-rb from Sopra Steria (based on the script of thc202 from the OWASP ZAP development team)

// This script is heavily based on the "Simple Form-Based Authentication.js" template
// It can be used to authenticate in a webapplication via a form submission followed by a GET request
// The submit target for the form, the name of the username field, the name of the password field
// and the URL of the GET target need to be specified after loading the script.
// The username and the password need to be configured when creating any Users.

// The authenticate function is called whenever ZAP requires to authenticate, for a Context for which
// this script was selected as the Authentication Method. The function should send any messages that
// are required to do the authentication and should return a message with an authenticated response
// so the calling method.
//
// NOTE: Any message sent in the function should be obtained using the 'helper.prepareMessage()' method.


// Parameters:
//   helper - a helper class providing useful methods: prepareMessage(), sendAndReceive(msg)
//   paramsValues - the values of the parameters configured in the Session Properties - Authentication panel.
//                        The paramsValues is a map, having as keys the parameters names (as returned by the
//                        getRequiredParamsNames() and getOptionalParamsNames() functions below)
//   credentials - an object containing the credentials values, as configured in the Session Properties - Users pane
//                        The credential values can be obtained via calls to the getParam(paramName) method. The param
//                        names are the ones returned by the getCredentialsParamsNames() below

// Make sure any Java classes used explicitly are imported
var HttpRequestHeader = Java.type('org.parosproxy.paros.network.HttpRequestHeader');
var HttpHeader = Java.type('org.parosproxy.paros.network.HttpHeader');
var URI = Java.type('org.apache.commons.httpclient.URI');
var AuthenticationHelper = Java.type('org.zaproxy.zap.authentication.AuthenticationHelper');
var Cookie = Java.type('org.apache.commons.httpclient.Cookie');

function authenticate(helper, paramsValues, credentials) {
    print("Authenticating via JavaScript script...");

    // Prepare the login submission request details
    var requestUri = new URI(paramsValues.get("Submission Form URL"), false);
    var requestMethod = HttpRequestHeader.POST;

    // Build the submission request body using the credential values
```

```
Welcome to the ZAP Scripting Console
```

ZAProxy

HTTP processes can also be automated using Zest scripts.



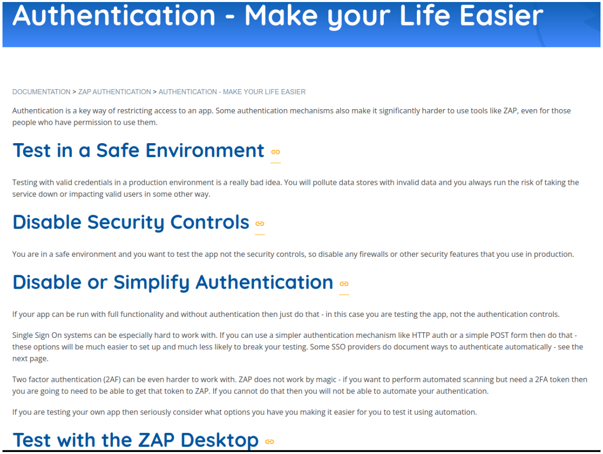
- HTTP Sender
- HUD
- Input Vector
- Passive Rules
- Payload Generator
- Payload Processor
- Proxy
- Selenium
- Sequence
- Session Management
- Stand Alone
  - Active scan rule list.js
  - alertAndPluginDetails.js
  - domainFinder.js
  - enableDebugLogging.js
  - example_library.js
  - extHistoryEnumerator.py
  - historySourceTagger.js
  - Juice shop authentication by form.js
  - Juice shop authentication by google.js
  - load_context_from_burp.py
  - load_function_example.js
  - Loop through alerts.js
  - Loop through history table.js
  - OAST Get BOAST Servers.js
  - OAST Get Interactsh Payloads.js
  - Open Fortune 500 websites in a browser(2).zst
  - Open Fortune 500 websites in a browser(3).zst
  - Open Fortune 500 websites in a browser(4).zst
  - Open Fortune 500 websites in a browser.zst
  - past_cookies_jar.py
  - Persona Create Account(2).zst
  - Persona Create Account(3).zst
  - Persona Create Account(4).zst
  - Persona Create Account.zst
  - Run report.js
  - scan_rule_list.js
  - SecurityCrawlMazeScore.js
  - Test Zest script
  - Traverse sites tree.js

```
{
  "about": "This is a Zest script. For more details about Zest visit https://github.com/zaproxy/zest/",
  "zestVersion": "0.8",
  "title": "Test Zest script",
  "description": "",
  "prefix": "http://www.example.com",
  "type": "StandAlone",
  "parameters": {
    "tokenStart": "{{",
    "tokenEnd": "}}",
    "tokens": {
      "par1": "0",
      "par2": "1"
    },
    "elementType": "ZestVariables"
  },
  "statements": [
    {
      "url": "http://www.example.com/",
      "data": "",
      "method": "GET",
      "headers": "DNT: 1\r\nUpgrade-Insecure-Requests: 1\r\n",
      "response": {
        "url": "http://www.example.com/",
        "headers": "HTTP/1.1 200 OK\r\nAge: 527694\r\nCache-Control: max-age\u003d604800\r\nContent-Type: text/html;
        "body": "\u003c!doctype html\u003e\n\u003chtml\u003e\n\u003chead\u003e\n    \u003ctitle\u003eExample Domain\u
        "statusCode": 200,
        "responseTimeInMs": 207,
        "elementType": "ZestResponse"
      },
      "assertions": [
        {
          "rootExpression": {
            "code": 200,
            "not": false,
            "elementType": "ZestExpressionStatusCode"
          },
          "elementType": "ZestAssertion"
        },
        {
```

Welcome to the ZAP Scripting Console

ZAProxy

Authentication makes your life hard.



Authentication - Make your Life Easier

DOCUMENTATION > ZAP AUTHENTICATION > AUTHENTICATION - MAKE YOUR LIFE EASIER

Authentication is a key way of restricting access to an app. Some authentication mechanisms also make it significantly harder to use tools like ZAP, even for those people who have permission to use them.

Test in a Safe Environment

Testing with valid credentials in a production environment is a really bad idea. You will pollute data stores with invalid data and you always run the risk of taking the service down or impacting valid users in some other way.

Disable Security Controls

You are in a safe environment and you want to test the app not the security controls, so disable any firewalls or other security features that you use in production.

Disable or Simplify Authentication

If your app can be run with full functionality and without authentication then just do that - in this case you are testing the app, not the authentication controls.

Single Sign On systems can be especially hard to work with. If you can use a simpler authentication mechanism like HTTP auth or a simple POST form then do that - these options will be much easier to set up and much less likely to break your testing. Some SSO providers do document ways to authenticate automatically - see the next page.

Two factor authentication (2AF) can be even harder to work with. ZAP does not work by magic - if you want to perform automated scanning but need a 2FA token then you are going to need to be able to get that token to ZAP. If you cannot do that then you will not be able to automate your authentication.

If you are testing your own app then seriously consider what options you have you making it easier for you to test it using automation.

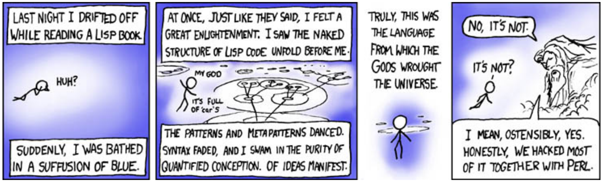
Test with the ZAP Desktop

Why not JSON?

- JSON gets complicated fast
- File gets long and editing it manually is painful
- Referencing previously defined items is ugly
- Complex syntax needed to process items (encode/decode/split/etc. . .)
- Can't reuse previously defined parts of json
- No user access to real code
- Not easily extendable
- And many more issues. . .

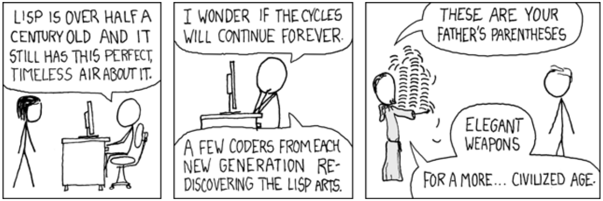
(Why LISP?)

- Need to create a language to describe the information exchange
- LISP languages are ideal for creating a custom DSL
- Ability to define own syntax
- Homoiconicity, i.e. code is data, data is code concept
- Metaprogramming, LISP macros can be defined to generate pieces of code


LAST NIGHT I DRIFTED OFF WHILE READING A LISP BOOK.
HUH?
SUDDENLY, I WAS BATHED IN A SUFFUSION OF BLUE.
AT ONCE, JUST LIKE THEY SAID, I FELT A GREAT ENLIGHTENMENT. I SAW THE NAKED STRUCTURE OF LISP CODE UNFOLD BEFORE ME.
MY GOD
IT'S FULL OF car'S
THE PATTERNS AND METAPATTERNS DANCED. SYNTAX FADED, AND I SWAM IN THE PURITY OF QUANTIFIED CONCEPTION. OF IDEAS MANIFEST.
TRULY, THIS WAS THE LANGUAGE FROM WHICH THE GODS WROUGHT THE UNIVERSE.
NO, IT'S NOT.
IT'S NOT?
I MEAN, OSTENSIBLY, YES. HONESTLY, WE HACKED MOST OF IT TOGETHER WITH PERL.

(Why hylang?)

- LISP-stick on a Python
- Compiles into Python code
- Access all the Python libraries
- Combines LISP flexibility with Python power and simplicity
- Easy to learn if you know Python

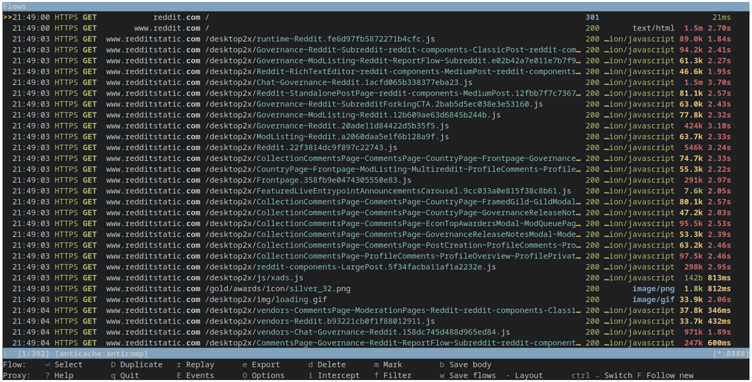

LISP IS OVER HALF A CENTURY OLD AND IT STILL HAS THIS PERFECT, TIMELESS AIR ABOUT IT.
I WONDER IF THE CYCLES WILL CONTINUE FOREVER.
A FEW CODERS FROM EACH NEW GENERATION RE-DISCOVERING THE LISP ARTS.
THESE ARE YOUR FATHER'S PARENTHESES
ELEGANT WEAPONS
FOR A MORE... CIVILIZED AGE.

(Understanding authentication)

- Often seen as a black box by pentesters
- Many bugs are overlooked
- Raider aims to make it easier to test and understand complex HTTP processess, like authentication

(Understanding authentication)

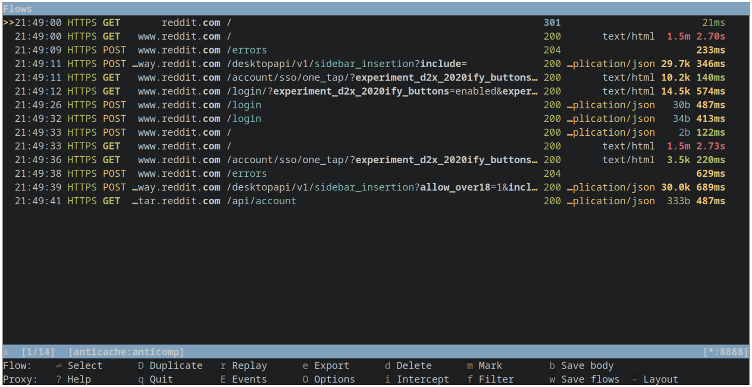
392 requests captured during Reddit authentication. Most are irrelevant.



```
Flows
>>21:49:00 HTTPS GET            reddit.com /                                                                                301                        21ms
  21:49:00 HTTPS GET        www.reddit.com /                                                                                200       text/html  1.5m 2.70s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/runtime~Reddit.fe6d97fb5872271b4cfc.js                                200 …ion/javascript 89.0k 1.84s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Governance~Reddit~Subreddit~reddit-components-ClassicPost~reddit-com… 200 …ion/javascript 94.2k 2.41s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Governance~ModListing~Reddit~ReportFlow~Subreddit.e02b42a7e011e7b7f9… 200 …ion/javascript 61.3k 2.27s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Reddit~RichTextEditor~reddit-components-MediumPost~reddit-components… 200 …ion/javascript 46.6k 1.95s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Chat~Governance~Reddit.1acfd065b338377eba23.js                        200 …ion/javascript  1.5m 3.70s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Reddit~StandalonePostPage~reddit-components-MediumPost.12fbb7f7c7367… 200 …ion/javascript 81.1k 2.57s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Governance~Reddit~SubredditForkingCTA.2bab5d5ec038e3e53160.js         200 …ion/javascript 63.0k 2.43s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Governance~ModListing~Reddit.12b609ae63d6845b244b.js                  200 …ion/javascript 77.8k 2.32s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Governance~Reddit.20ade11d84422d5b35f5.js                             200 …ion/javascript  424k 3.10s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/ModListing~Reddit.a2060daa5e1f6b128a9f.js                             200 …ion/javascript 63.7k 2.33s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Reddit.22f3814dc9f897c22743.js                                        200 …ion/javascript  546k 3.24s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/CollectionCommentsPage~CommentsPage~CountryPage~Frontpage~Governance… 200 …ion/javascript 74.7k 2.33s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/CountryPage~Frontpage~ModListing~Multireddit~ProfileComments~Profile… 200 …ion/javascript 55.3k 2.22s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/Frontpage.358fb9e0474305550e83.js                                     200 …ion/javascript  291k 2.97s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/FeaturedLiveEntrypointAnnouncementsCarousel.9cc033a0e815f38c8b61.js   200 …ion/javascript  7.6k 2.05s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/CollectionCommentsPage~CommentsPage~CountryPage~FramedGild~GildModal… 200 …ion/javascript 80.1k 2.57s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/CollectionCommentsPage~CommentsPage~CountryPage~GovernanceReleaseNot… 200 …ion/javascript 47.2k 2.03s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/CollectionCommentsPage~CommentsPage~EconTopAwardersModal~ModQueuePag… 200 …ion/javascript 95.5k 2.53s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/CollectionCommentsPage~CommentsPage~GovernanceReleaseNotesModal~Mode… 200 …ion/javascript 53.3k 2.39s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/CollectionCommentsPage~CommentsPage~PostCreation~ProfileComments~Pro… 200 …ion/javascript 63.2k 2.46s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/CollectionCommentsPage~ProfileComments~ProfileOverview~ProfilePrivat… 200 …ion/javascript 97.5k 2.46s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/reddit-components-LargePost.5f34facba11af1a2232e.js                   200 …ion/javascript  298k 2.95s
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/js/xads.js                                                            200 …ion/javascript  142b 813ms
  21:49:03 HTTPS GET  www.redditstatic.com /gold/awards/icon/silver_32.png                                                  200       image/png  1.8k 812ms
  21:49:03 HTTPS GET  www.redditstatic.com /desktop2x/img/loading.gif                                                       200       image/gif 33.9k 2.06s
  21:49:04 HTTPS GET  www.redditstatic.com /desktop2x/vendors~CommentsPage~ModerationPages~Reddit~reddit-components-Classi… 200 …ion/javascript 37.8k 346ms
  21:49:04 HTTPS GET  www.redditstatic.com /desktop2x/vendors~Reddit.b93221cb0f1f88012911.js                                200 …ion/javascript 33.7k 432ms
  21:49:04 HTTPS GET  www.redditstatic.com /desktop2x/vendors~Chat~Governance~Reddit.158dc745d488d965ed84.js                200 …ion/javascript  971k 1.89s
  21:49:04 HTTPS GET  www.redditstatic.com /desktop2x/CommentsPage~Governance~Reddit~ReportFlow~Subreddit~reddit-component… 200 …ion/javascript  247k 600ms
⇩  [1/392] [anticache:anticomp]                                                                                                                   [*:8888]
Flow:    ⏎ Select      D Duplicate   r Replay     e Export     d Delete     m Mark      b Save body
Proxy:   ? Help        q Quit        E Events     O Options    i Intercept  f Filter    w Save flows  - Layout     ctrl → Switch F Follow new
```

(Understanding authentication)

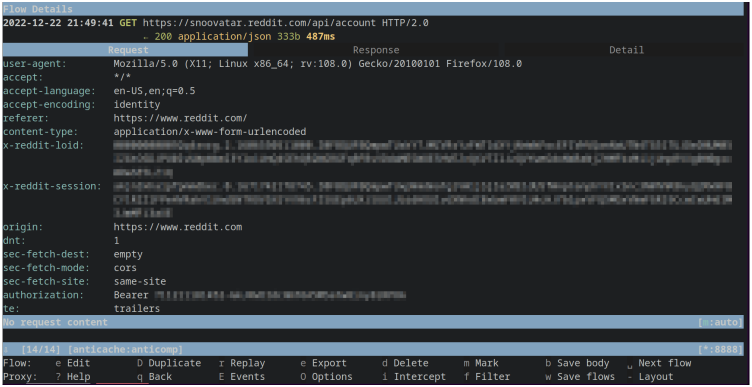
Removing obviously irrelevant requests makes it easier to analyse.



```
Flows
>>21:49:00 HTTPS GET       reddit.com /                                                   301                         21ms
  21:49:00 HTTPS GET   www.reddit.com /                                                   200       text/html  1.5m 2.70s
  21:49:09 HTTPS POST  www.reddit.com /errors                                             204                        233ms
  21:49:11 HTTPS POST …way.reddit.com /desktopapi/v1/sidebar_insertion?include=           200 …plication/json 29.7k 346ms
  21:49:11 HTTPS GET   www.reddit.com /account/sso/one_tap/?experiment_d2x_2020ify_buttons… 200    text/html 10.2k 140ms
  21:49:12 HTTPS GET   www.reddit.com /login/?experiment_d2x_2020ify_buttons=enabled&exper… 200    text/html 14.5k 574ms
  21:49:26 HTTPS POST  www.reddit.com /login                                              200 …plication/json   30b 487ms
  21:49:32 HTTPS POST  www.reddit.com /login                                              200 …plication/json   34b 413ms
  21:49:33 HTTPS POST  www.reddit.com /                                                   200 …plication/json    2b 122ms
  21:49:33 HTTPS GET   www.reddit.com /                                                   200       text/html  1.5m 2.73s
  21:49:36 HTTPS GET   www.reddit.com /account/sso/one_tap/?experiment_d2x_2020ify_buttons… 200    text/html  3.5k 220ms
  21:49:38 HTTPS POST  www.reddit.com /errors                                             204                        629ms
  21:49:39 HTTPS POST …way.reddit.com /desktopapi/v1/sidebar_insertion?allow_over18=1&incl… 200 …plication/json 30.0k 689ms
  21:49:41 HTTPS GET   …tar.reddit.com /api/account                                       200 …plication/json  333b 487ms

⇩  [1/14]  [anticache:anticomp]                                                                              [*:8888]
Flow:   ⏎ Select     D Duplicate   r Replay    e Export    d Delete     m Mark      b Save body
Proxy:  ? Help       q Quit        E Events    O Options   i Intercept  f Filter    w Save flows  - Layout
```

(Understanding authentication)

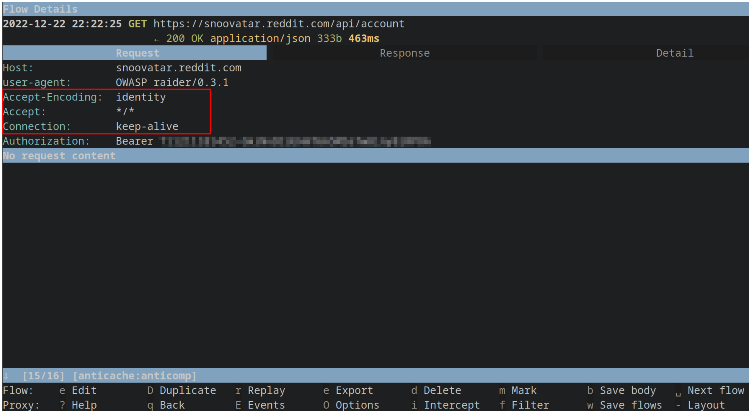
Find one request which only works when authenticated.



```
Flow Details
2022-12-22 21:49:41 GET https://snoovatar.reddit.com/api/account HTTP/2.0
                         ← 200 application/json 333b 487ms
                 Request                          Response                          Detail
user-agent:          Mozilla/5.0 (X11; Linux x86_64; rv:108.0) Gecko/20100101 Firefox/108.0
accept:              */*
accept-language:     en-US,en;q=0.5
accept-encoding:     identity
referer:             https://www.reddit.com/
content-type:        application/x-www-form-urlencoded
x-reddit-loid:       [redacted]
x-reddit-session:    [redacted]
origin:              https://www.reddit.com
dnt:                 1
sec-fetch-dest:      empty
sec-fetch-mode:      cors
sec-fetch-site:      same-site
authorization:       Bearer [redacted]
te:                  trailers
No request content                                                                  [m:auto]

⇩  [14/14] [anticache:anticomp]                                                     [*:8888]
Flow:    e Edit       D Duplicate   r Replay    e Export    d Delete     m Mark     b Save body   ␣ Next flow
Proxy:   ? Help       q Back        E Events    O Options   i Intercept  f Filter   w Save flows  - Layout
```

(Understanding authentication)

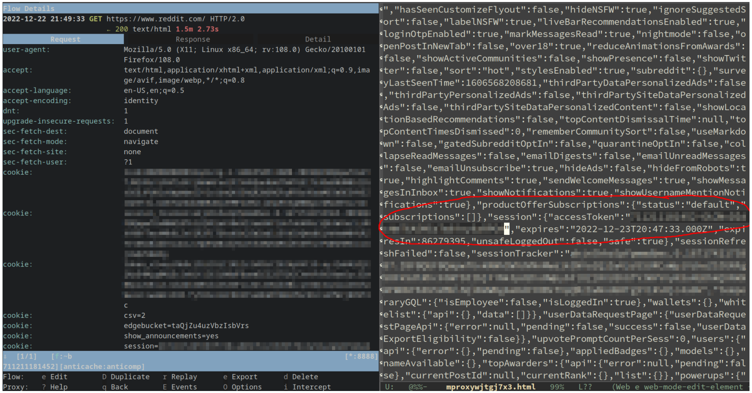
Only Authorization header is needed here. Host/User-agent are required by HTTP. Highlighted headers are generated automatically.



```
Flow Details
2022-12-22 22:22:25 GET https://snoovatar.reddit.com/api/account
                        ← 200 OK application/json 333b 463ms
                  Request                         Response                          Detail
Host:                 snoovatar.reddit.com
user-agent:           OWASP raider/0.3.1
Accept-Encoding:      identity
Accept:               */*
Connection:           keep-alive
Authorization:        Bearer [redacted]
No request content




⇩  [15/16]  [anticache:anticomp]
Flow:    e Edit        D Duplicate   r Replay    e Export    d Delete     m Mark     b Save body   ␣ Next flow
Proxy:   ? Help        q Back        E Events    O Options   i Intercept  f Filter   w Save flows  - Layout
```

(Understanding authentication)

Finding the Authorization header token in the response body of a previous request.



```
Flow Details
2022-12-22 21:49:33 GET https://www.reddit.com/ HTTP/2.0
                       ← 200 text/html 1.5m 2.73s
         Request                          Response                     Detail
user-agent:                   Mozilla/5.0 (X11; Linux x86_64; rv:108.0) Gecko/20100101
                              Firefox/108.0
accept:                       text/html,application/xhtml+xml,application/xml;q=0.9,ima
                              ge/avif,image/webp,*/*;q=0.8
accept-language:              en-US,en;q=0.5
accept-encoding:              identity
dnt:                          1
upgrade-insecure-requests:    1
sec-fetch-dest:               document
sec-fetch-mode:               navigate
sec-fetch-site:               none
sec-fetch-user:               ?1
cookie:                       [redacted]
cookie:                       [redacted]
cookie:                       [redacted]
                              c
cookie:                       csv=2
cookie:                       edgebucket=taQjZu4uzVbzIsbVrs
cookie:                       show_announcements=yes
cookie:                       session=[redacted]
⇕  [1/1]   [f:~b                                                         [*:8888]
711211181452][anticache:anticomp]
Flow:    e Edit      D Duplicate   r Replay    e Export    d Delete
Proxy:   ? Help      q Back        E Events    O Options   i Intercept
```

```
","hasSeenCustomizeFlyout":false,"hideNSFW":true,"ignoreSuggestedS
ort":false,"labelNSFW":true,"liveBarRecommendationsEnabled":true,"
loginOtpEnabled":true,"markMessagesRead":true,"nightmode":false,"o
penPostInNewTab":false,"over18":true,"reduceAnimationsFromAwards":
false,"showActiveCommunities":false,"showPresence":false,"showTwit
ter":false,"sort":"hot","stylesEnabled":true,"subreddit":{},"surve
yLastSeenTime":1606568208681,"thirdPartyDataPersonalizedAds":false
,"thirdPartyPersonalizedAds":false,"thirdPartySiteDataPersonalized
Ads":false,"thirdPartySiteDataPersonalizedContent":false,"showLoca
tionBasedRecommendations":false,"topContentDismissalTime":null,"to
pContentTimesDismissed":0,"rememberCommunitySort":false,"useMarkdo
wn":false,"gatedSubredditOptIn":false,"quarantineOptIn":false,"col
lapseReadMessages":false,"emailDigests":false,"emailUnreadMessages
":false,"emailUnsubscribe":true,"hideAds":false,"hideFromRobots":t
rue,"highlightComments":true,"sendWelcomeMessages":true,"showMessa
gesInInbox":true,"showNotifications":true,"showUsernameMentionNoti
fications":true},"productOfferSubscriptions":{"status":"default","
subscriptions":[]},"session":{"accessToken":"[redacted]
[redacted]","expires":"2022-12-23T20:47:33.000Z","expi
resIn":86279395,"unsafeLoggedOut":false,"safe":true},"sessionRefre
shFailed":false,"sessionTracker":"[redacted]
rariyGQL":{"isEmployee":false,"isLoggedIn":true},"wallets":{},"whit
elist":{"api":{},"data":[]}},"userDataRequestPage":{"userDataReque
stPageApi":{"error":null,"pending":false,"success":false,"userData
ExportEligibility":false}},"upvotePromptCountPerSess":0,"users":{"
api":{"error":{},"pending":false},"appliedBadges":{},"models":{},"
nameAvailable":{},"topAwarders":{"api":{"error":null,"pending":fal
se},"currentPostId":null,"currentRank":{},"list":{}},"powerups":{"
U:   @%%-    mproxywjtgj7x3.html    99%    L??    (Web e web-mode-edit-element
```

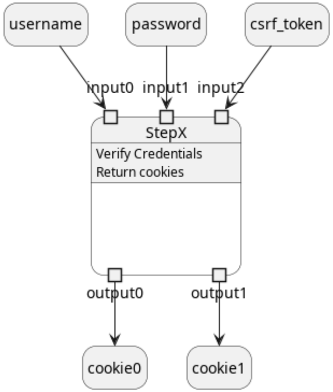
(Abstracting Authentication using Finite State Machines)

Finite State Machine (FSM) is a mathematical model of computation. It allows for a detailed analysis of how a computer system functions.

- A system is stateful if it remembers preceding events
- A state is the remembered information about the system
- Can be in exactly one of the finite number of states at any given time
- *Mealy* FSMs can be used to model authentication systems
- Output values are determined both by current state and inputs

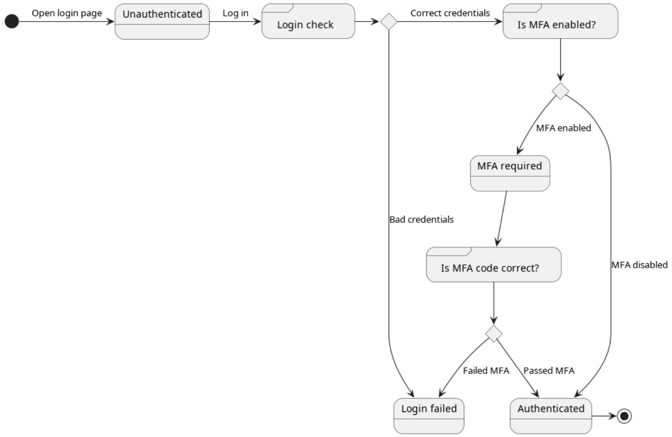
(Abstracting Authentication using Finite State Machines)

One step (state) of the authentication with its inputs and outputs:



(Abstracting Authentication using Finite State Machines)

Authentication represented as FSM with its inputs/outputs hidden:



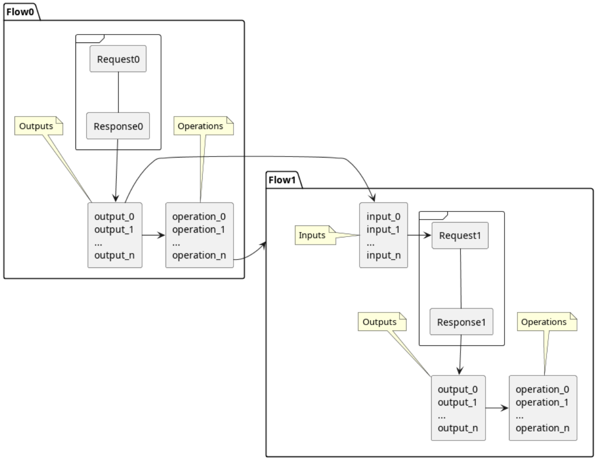
(Flows)

Flows are used to describe the information exchange between the client and the server with one pair of HTTP request and the response.

- Requires a Request with URL
- Optionally `outputs`
  - defines what to extract from the response
- Optionally `operations`
  - arbitrary actions to run after receiving response
  - links to other Flows (could be conditional, and nested)

(Flows)

Raider Flows represent one state in the FSM.



(Request)

- The only required parameter is the URL, rest are optional
  - Requests HTTP method are specified via the class methods. `Request.get`, `Request.post`, `Request.put`, `Request.custom`
  - `:cookies`, `:headers`
  - `:params`, `:data`, `:json`, `:multipart`
- All Request parameters can contain Plugins. Use those to share data between Flows.

(Plugins)

- Small pieces of code
- Used as inputs and/or outputs
- Extract data
- Parse data/Plugins
- Encode/Decode data/Plugins
- Some can be nested
- User can write their own without touching the core

(Operations)

- Run *after* Response is received
- Execute arbitrary code
- Control the information flow
  - `Next, Success, Failure`
- Run other Operations conditionally (can be nested)
- Can run even real hylang code by using the LISP *quote*
- User can write their own without touching the core

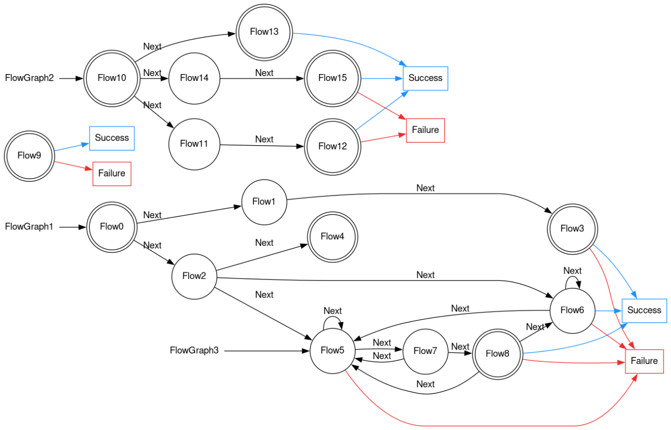
(FlowGraphs)

FlowGraphs are used to chain multiple Flows together and follow the links until the end, or until `(Success)/(Failure)` operations.

- Pointer to a starting Flow
- Optionally a test Flow. Checks if the FlowGraph ran successfully, i.e. if user is authenticated

(FlowGraphs)

Complex systems can be simulated and tested with this architecture:

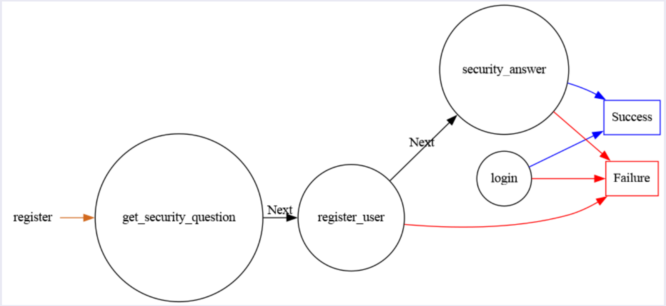


(Demo: Automating juiceshop attacks)

Automate registration and login

Register a new user and log in. Run `register` FlowGraph then the `login` Flow.

```
$ raider run juiceshop register,login   # First run
$ raider run juiceshop login            # Subsequent runs
```

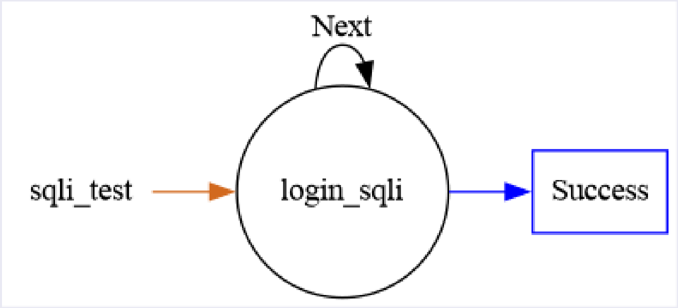


(Demo: Automating juiceshop attacks)

Test SQL injection in email field

Email input is vulnerable to SQL injection. We run the Flow in a loop and exit on (Success).

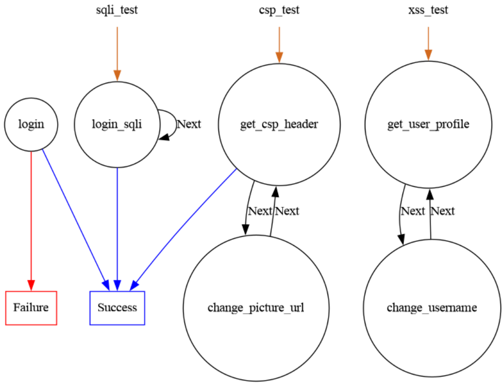
```
$ raider run juiceshop sqli_test
```



(Demo: Automating juiceshop attacks)

Username field is vulnerable to persistent XSS. Input is filtered with a regex, and can be bypassed. Content-Security-Policy header is used as well, and due to another bug can be bypassed too.

```
$ raider run login,csp_test,xss_test
$ raider run sqli_test,csp_test,xss_test
```



(What's next?)

- Documentation
- UI/UX
  - Debugging
  - Improve CLI
  - Raider REPL (read-eval-print loop)
  - Generate hyfiles using LLMs (help needed)
- Features
  - Fuzzing
  - Sessions
  - Macros
- Integrating with other tools

(Limitations)

- Steep learning curve
- LISP parentheses
- Limited community support
- Limited documentation
- Limited OS support

(Questions/Answers)

Raider is not just a toy anymore, it evolved enough to work on complex real life systems. There's still a lot of work to do and room for improvement

Contact me

- Mastodon: @raiderauth@infosec.exchange
- E-mail: hello@raiderauth.com

Links

- Website: raiderauth.com
- Source: github.com/OWASP/raider
- Documentation: docs.raiderauth.com