



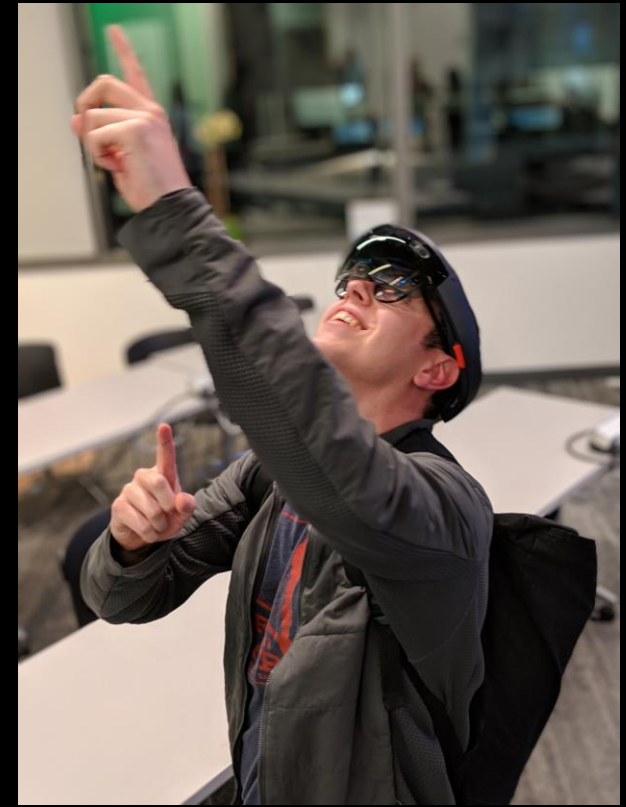
Sponsored by 

# Manual JavaScript Analysis is a Bug

Day of Shecurity

# About Me

- Sr. Security Consultant @ Synopsys
  - Formerly Cigital
- AngularSF Organizer
  - <https://www.meetup.com/Angular-SF/>
- B.Sc. in Computer Security and Ethical Hacking
  - Founder of <http://leedshackingsociety.co.uk/>
- JavaScript Enthusiast!

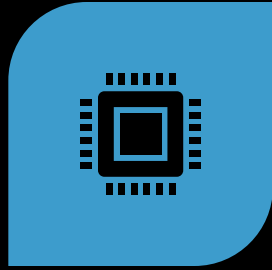


**SYNOPSYS®**

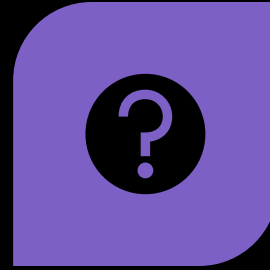
# Agenda



MANUAL WORK IS A  
BUG



WHAT CAN SHOULD  
WE AUTOMATE?



WHAT ALREADY  
EXISTS



PROOF OF CONCEPT  
APP

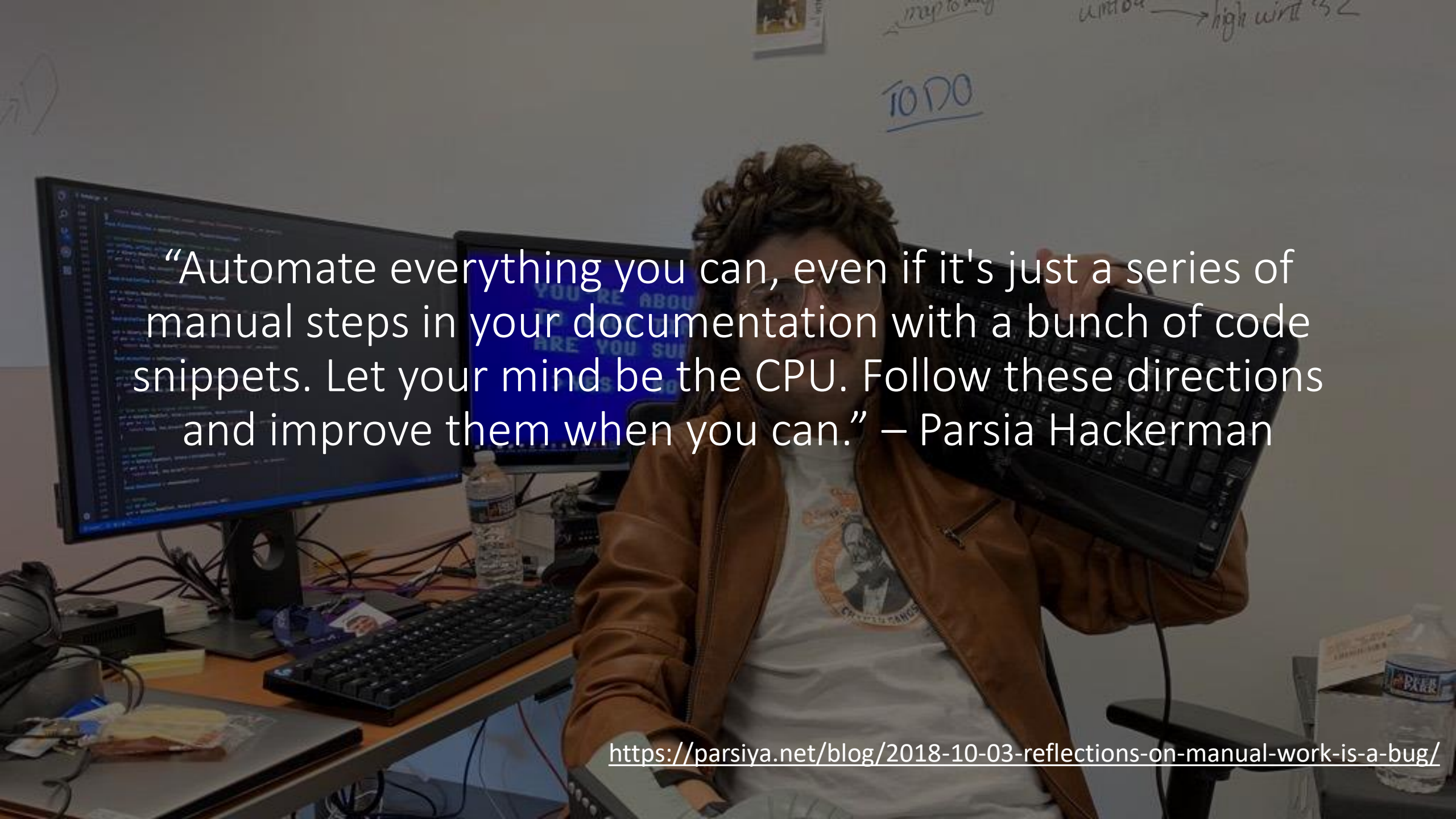
# Manual Work Is a Bug

A.B.A: always be automating – Thomas A. Limoncelli

- Automation can be used to help improve productivity and reduce repetitive work
- Four Phases
  1. Document the steps
  2. Create automation equivalents
  3. Create automation
  4. Self-service and autonomous systems

<https://queue.acm.org/detail.cfm?id=3197520>

<https://parsiya.net/blog/2018-10-03-reflections-on-manual-work-is-a-bug/>



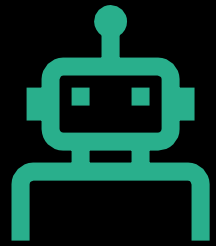
“Automate everything you can, even if it's just a series of manual steps in your documentation with a bunch of code snippets. Let your mind be the CPU. Follow these directions and improve them when you can.” – Parsia Hackerman

<https://parsiya.net/blog/2018-10-03-reflections-on-manual-work-is-a-bug/>

# Automation Process:

1. Target something boring and repetitive
2. Automate it. It does not have to be code. Steps are fine
3. Follow your steps when doing that task and improve your steps in real-time (do not delay this)
4. Use your newly acquired spare time to target a new task
5. Share the documentation (wiki, git repo, confluence, etc.) and collaborate with others
  - A. Good Example <https://github.com/EdOverflow/can-i-take-over-xyz>

Automation will allow you to spend your time focusing on things which require your energy such as business logic flaws, or identifying more complex attack chains



What ~~can~~/should  
we automate?





We should not have to manually identify what URLs, and assets exist in an application!

# Endpoint Discovery



<https://github.com/PortSwigger/js-link-finder>



<https://github.com/ettic-team/EndpointFinder>

# Endpoint Discovery: js-link-finder

- Useful for finding additional domains and paths through JavaScript
  - Example AOL:

## LinkFinder Log:

```
37 - https://learn.one.aol.com/one_by_one_for_haver-112137/mq3/mq3vsn_comfiguring_no_blocker_2_to_work_with_one_by_one
38 - //learn.one.aol.com
39 - //help.one.aol.com
40 - http://nexage-dev.demohoster.com
41 - http://qa-ge-iqservice001.us-ec.adtech.com:8080/h2/index.do
42 - http://mydev.aol.com:9003/
43 - http://mydev.aol.com:9004/
44 - https://onevideo.aol.com
45 - http://m-dev-aop-devint05.advertising.aol.com:8083/aop-aux/
46 - http://m-prd-aopadminui001.advertising.aol.com/admin/
47 - http://uk.admin.adlearnop.advertising.aol.com/
48 - http://jp.admin.adlearnop.advertising.aol.com/
49 - http://sj-qa03.sj.adap.tv:4310/platform/index.html#/login
50 - http://mydev.aol.com:9010/analytics/
51 - https://portal.vidible.tv
52 - http://mydev.aol.com:9002/
53 - https://oneapi-dev.aol.com/one-central/
54 - https://one-dev.aol.com/one-central/
55 - https://id-uat.b2b.oath.com/identity/oauth2/authorize
56 - https://id-uat.b2b.oath.com/identity/XUI/#logout/
57 - https://one-dev.aol.com/one-central/oidc.html
58 - https://one-dev.aol.com/one-central/oidc.html
```

# Source Maps (\*.js.map)

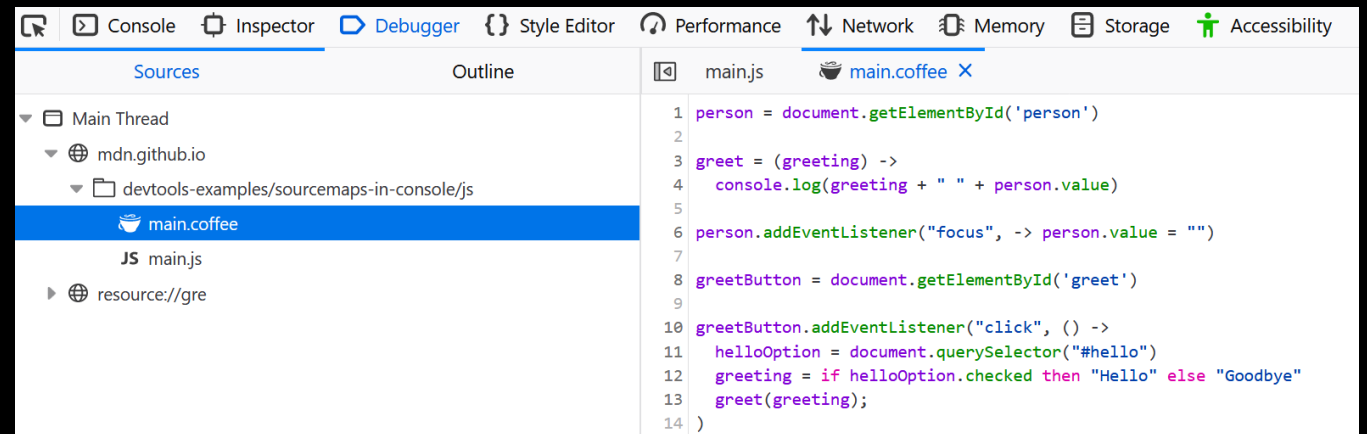
- Production minified JavaScript is intimidating to review:
  - Computer Transpiled: TypeScript, Coffee Script
  - Minified: Webpack, rollup, Pracel

```

window.webpackJsonp(window.webpackJsonp||[]).push([["vendors/Chat-Governance-Reddit"],["./node_modules/base64/base64.js"],function(e,t,n){function i(var
est,n="ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz0123456789+=",function o(e){this.message=e,o.prototype.name=Error,o.prototype.name="InvalidCharact
e.btoa=function(e){for(var t,r,s=String(e),i=0,a=n,s=s.charAt(0|1){[a+=s.charAt(0|1)](a=="i",i+=1)}+u=a.charCodeAt(63&!!t&8-i)<1)}{if((r=s.charCodeAtAt(i+=.75))>255)throw new o("
string to be encoded contains characters outside of the Latin range.");return t}<8|t|u=a.charCodeAt(i),e.atob=function(e){var
t=String(e).replace(/%20/g,"");if(t.length%4==1)throw new o("atob failed: The string to be decoded is not correctly encoded.");for(var r,i,s,o,a=0,u=0;st.
r=(t.charAt(4*i)+s),i+=4)}+String.fromCharCode(255&!!r>-2)<86|o=s.charCodeAt(i),return u)})),["./node_modules/base64-js/index.js"],function(e,t,n){use
strict";t.byteLength=function(e){var t=l(e),o=t[0],n=t[1],r=3*(n-o+4-o)/4,t.toByteArray=function(e){for(var t,n,l=e,o=n[0],o=n[1],a=new s(function(e,t,n
t,n){4-n,o,i}),u=0,c=i>0?74-o:0,d=0,d<4?c:d+=4,t[r].charCodeAt(d)<18|e[r].charCodeAt(d+1)<12|e[r].charCodeAt(d+2)<18|e[r].charCodeAt(d+3)<16|e[r].charCodeAt(d+4)<16|e[r].charCodeAt(d+5)<16|e[r].charCodeAt(d+6)<16|e[r].charCodeAt(d+7)<16|e[r].charCodeAt(d+8)<16|e[r].charCodeAt(d+9)<16|e[r].charCodeAt(d+10)<16|e[r].charCodeAt(d+11)<16|e[r].charCodeAt(d+12)<16|e[r].charCodeAt(d+13)<16|e[r].charCodeAt(d+14)<16|e[r].charCodeAt(d+15)<16|e[r].charCodeAt(d+16)<16|e[r].charCodeAt(d+17)<16|e[r].charCodeAt(d+18)<16|e[r].charCodeAt(d+19)<16|e[r].charCodeAt(d+20)<16|e[r].charCodeAt(d+21)<16|e[r].charCodeAt(d+22)<16|e[r].charCodeAt(d+23)<16|e[r].charCodeAt(d+24)<16|e[r].charCodeAt(d+25)<16|e[r].charCodeAt(d+26)<16|e[r].charCodeAt(d+27)<16|e[r].charCodeAt(d+28)<16|e[r].charCodeAt(d+29)<16|e[r].charCodeAt(d+30)<16|e[r].charCodeAt(d+31)<16|e[r].charCodeAt(d+32)<16|e[r].charCodeAt(d+33)<16|e[r].charCodeAt(d+34)<16|e[r].charCodeAt(d+35)<16|e[r].charCodeAt(d+36)<16|e[r].charCodeAt(d+37)<16|e[r].charCodeAt(d+38)<16|e[r].charCodeAt(d+39)<16|e[r].charCodeAt(d+40)<16|e[r].charCodeAt(d+41)<16|e[r].charCodeAt(d+42)<16|e[r].charCodeAt(d+43)<16|e[r].charCodeAt(d+44)<16|e[r].charCodeAt(d+45)<16|e[r].charCodeAt(d+46)<16|e[r].charCodeAt(d+47)<16|e[r].charCodeAt(d+48)<16|e[r].charCodeAt(d+49)<16|e[r].charCodeAt(d+50)<16|e[r].charCodeAt(d+51)<16|e[r].charCodeAt(d+52)<16|e[r].charCodeAt(d+53)<16|e[r].charCodeAt(d+54)<16|e[r].charCodeAt(d+55)<16|e[r].charCodeAt(d+56)<16|e[r].charCodeAt(d+57)<16|e[r].charCodeAt(d+58)<16|e[r].charCodeAt(d+59)<16|e[r].charCodeAt(d+60)<16|e[r].charCodeAt(d+61)<16|e[r].charCodeAt(d+62)<16|e[r].charCodeAt(d+63)<16|e[r].charCodeAt(d+64)<16|e[r].charCodeAt(d+65)<16|e[r].charCodeAt(d+66)<16|e[r].charCodeAt(d+67)<16|e[r].charCodeAt(d+68)<16|e[r].charCodeAt(d+69)<16|e[r].charCodeAt(d+70)<16|e[r].charCodeAt(d+71)<16|e[r].charCodeAt(d+72)<16|e[r].charCodeAt(d+73)<16|e[r].charCodeAt(d+74)<16|e[r].charCodeAt(d+75)<16|e[r].charCodeAt(d+76)<16|e[r].charCodeAt(d+77)<16|e[r].charCodeAt(d+78)<16|e[r].charCodeAt(d+79)<16|e[r].charCodeAt(d+80)<16|e[r].charCodeAt(d+81)<16|e[r].charCodeAt(d+82)<16|e[r].charCodeAt(d+83)<16|e[r].charCodeAt(d+84)<16|e[r].charCodeAt(d+85)<16|e[r].charCodeAt(d+86)<16|e[r].charCodeAt(d+87)<16|e[r].charCodeAt(d+88)<16|e[r].charCodeAt(d+89)<16|e[r].charCodeAt(d+90)<16|e[r].charCodeAt(d+91)<16|e[r].charCodeAt(d+92)<16|e[r].charCodeAt(d+93)<16|e[r].charCodeAt(d+94)<16|e[r].charCodeAt(d+95)<16|e[r].charCodeAt(d+96)<16|e[r].charCodeAt(d+97)<16|e[r].charCodeAt(d+98)<16|e[r].charCodeAt(d+99)<16|e[r].charCodeAt(d+100)<16|e[r].charCodeAt(d+101)<16|e[r].charCodeAt(d+102)<16|e[r].charCodeAt(d+103)<16|e[r].charCodeAt(d+104)<16|e[r].charCodeAt(d+105)<16|e[r].charCodeAt(d+106)<16|e[r].charCodeAt(d+107)<16|e[r].charCodeAt(d+108)<16|e[r].charCodeAt(d+109)<16|e[r].charCodeAt(d+110)<16|e[r].charCodeAt(d+111)<16|e[r].charCodeAt(d+112)<16|e[r].charCodeAt(d+113)<16|e[r].charCodeAt(d+114)<16|e[r].charCodeAt(d+115)<16|e[r].charCodeAt(d+116)<16|e[r].charCodeAt(d+117)<16|e[r].charCodeAt(d+118)<16|e[r].charCodeAt(d+119)<16|e[r].charCodeAt(d+120)<16|e[r].charCodeAt(d+121)<16|e[r].charCodeAt(d+122)<16|e[r].charCodeAt(d+123)<16|e[r].charCodeAt(d+124)<16|e[r].charCodeAt(d+125)<16|e[r].charCodeAt(d+126)<16|e[r].charCodeAt(d+127)<16|e[r].charCodeAt(d+128)<16|e[r].charCodeAt(d+129)<16|e[r].charCodeAt(d+130)<16|e[r].charCodeAt(d+131)<16|e[r].charCodeAt(d+132)<16|e[r].charCodeAt(d+133)<16|e[r].charCodeAt(d+134)<16|e[r].charCodeAt(d+135)<16|e[r].charCodeAt(d+136)<16|e[r].charCodeAt(d+137)<16|e[r].charCodeAt(d+138)<16|e[r].charCodeAt(d+139)<16|e[r].charCodeAt(d+140)<16|e[r].charCodeAt(d+141)<16|e[r].charCodeAt(d+142)<16|e[r].charCodeAt(d+143)<16|e[r].charCodeAt(d+144)<16|e[r].charCodeAt(d+145)<16|e[r].charCodeAt(d+146)<16|e[r].charCodeAt(d+147)<16|e[r].charCodeAt(d+148)<16|e[r].charCodeAt(d+149)<16|e[r].charCodeAt(d+150)<16|e[r].charCodeAt(d+151)<16|e[r].charCodeAt(d+152)<16|e[r].charCodeAt(d+153)<16|e[r].charCodeAt(d+154)<16|e[r].charCodeAt(d+155)<16|e[r].charCodeAt(d+156)<16|e[r].charCodeAt(d+157)<16|e[r].charCodeAt(d+158)<16|e[r].charCodeAt(d+159)<16|e[r].charCodeAt(d+160)<16|e[r].charCodeAt(d+161)<16|e[r].charCodeAt(d+162)<16|e[r].charCodeAt(d+163)<16|e[r].charCodeAt(d+164)<16|e[r].charCodeAt(d+165)<16|e[r].charCodeAt(d+166)<16|e[r].charCodeAt(d+167)<16|e[r].charCodeAt(d+168)<16|e[r].charCodeAt(d+169)<16|e[r].charCodeAt(d+170)<16|e[r].charCodeAt(d+171)<16|e[r].charCodeAt(d+172)<16|e[r].charCodeAt(d+173)<16|e[r].charCodeAt(d+174)<16|e[r].charCodeAt(d+175)<16|e[r].charCodeAt(d+176)<16|e[r].charCodeAt(d+177)<16|e[r].charCodeAt(d+178)<16|e[r].charCodeAt(d+179)<16|e[r].charCodeAt(d+180)<16|e[r].charCodeAt(d+181)<16|e[r].charCodeAt(d+182)<16|e[r].charCodeAt(d+183)<16|e[r].charCodeAt(d+184)<16|e[r].charCodeAt(d+185)<16|e[r].charCodeAt(d+186)<16|e[r].charCodeAt(d+187)<16|e[r].charCodeAt(d+188)<16|e[r].charCodeAt(d+189)<16|e[r].charCodeAt(d+190)<16|e[r].charCodeAt(d+191)<16|e[r].charCodeAt(d+192)<16|e[r].charCodeAt(d+193)<16|e[r].charCodeAt(d+194)<16|e[r].charCodeAt(d+195)<16|e[r].charCodeAt(d+196)<16|e[r].charCodeAt(d+197)<16|e[r].charCodeAt(d+198)<16|e[r].charCodeAt(d+199)<16|e[r].charCodeAt(d+200)<16|e[r].charCodeAt(d+201)<16|e[r].charCodeAt(d+202)<16|e[r].charCodeAt(d+203)<16|e[r].charCodeAt(d+204)<16|e[r].charCodeAt(d+205)<16|e[r].charCodeAt(d+206)<16|e[r].charCodeAt(d+207)<16|e[r].charCodeAt(d+208)<16|e[r].charCodeAt(d+209)<16|e[r].charCodeAt(d+210)<16|e[r].charCodeAt(d+211)<16|e[r].charCodeAt(d+212)<16|e[r].charCodeAt(d+213)<16|e[r].charCodeAt(d+214)<16|e[r].charCodeAt(d+215)<16|e[r].charCodeAt(d+216)<16|e[r].charCodeAt(d+217)<16|e[r].charCodeAt(d+218)<16|e[r].charCodeAt(d+219)<16|e[r].charCodeAt(d+220)<16|e[r].charCodeAt(d+221)<16|e[r].charCodeAt(d+222)<16|e[r].charCodeAt(d+223)<16|e[r].charCodeAt(d+224)<16|e[r].charCodeAt(d+225)<16|e[r].charCodeAt(d+226)<16|e[r].charCodeAt(d+227)<16|e[r].charCodeAt(d+228)<16|e[r].charCodeAt(d+229)<16|e[r].charCodeAt(d+230)<16|e[r].charCodeAt(d+231)<16|e[r].charCodeAt(d+232)<16|e[r].charCodeAt(d+233)<16|e[r].charCodeAt(d+234)<16|e[r].charCodeAt(d+235)<16|e[r].charCodeAt(d+
```

# Source Maps (\*.js.map)

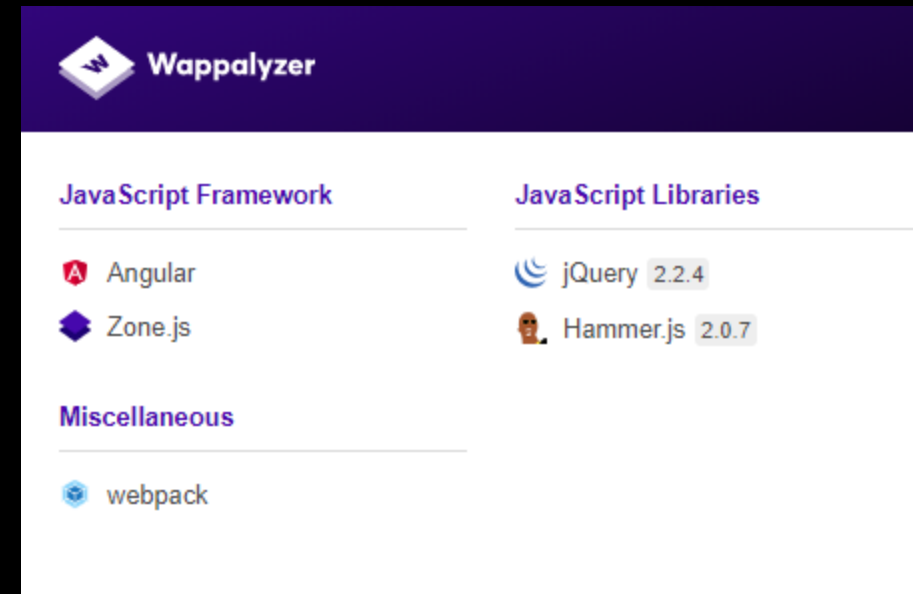
- Tools to reverse source maps
  - Directly in the browser
  - <https://github.com/denandz/sourcemappper>
  - <https://github.com/pavloko/source-map-unpack>
  - <https://github.com/mozilla/source-map>



- \* Opportunity to build a Burp extension to reverse source maps

# Understanding Technologies (Fingerprinting)

- Burp Extensions
  - Asset Discover
    - [https://github.com/redhuntlabs/BurpSuite-Asset\\_Discover](https://github.com/redhuntlabs/BurpSuite-Asset_Discover)
  - Software Version
    - <https://github.com/augustd/burp-suite-software-version-checks>
- Browser Plugins
  - <https://www.wappalyzer.com/>
- Source Code
  - <https://docs.npmjs.com/files/package.json>





We should not have to manually detect 'basic' sources and sinks

# Detecting Issues



ESLint



Burp Suite



#	Task	Time	Action	Issue type	Host
21	2	09:02:28 29 Aug 2019	Issue found	! JavaScript injection (DOM-based)	http://aspnet.testsparker.c...
20	2	09:02:26 29 Aug 2019	Issue found	i Frameable response (potential Clickjacking)	http://aspnet.testsparker.c...
19	2	09:02:26 29 Aug 2019	Issue found	i Cross-domain script include	http://aspnet.testsparker.c...

Advisory Request Response Static analysis

Data is read from **location.href** and passed to **setTimeout()** via the following statements:

- `var taintedVariable = location.href.split("#")[1];`
- `setTimeout("var x=" + taintedVariable, 500);`

#	Task	Time	Action	Issue type	Host
10	4	08:54:49 29 Aug 2019	Issue found	! Cross-site scripting (DOM-based)	http://demo.testfire.net
9	4	08:54:49 29 Aug 2019	Issue found	i HTML does not specify charset	http://demo.testfire.net

Advisory Request Response Static analysis

Data is read from **document.location.hash** and passed to the **'innerHTML'** property of a DOM element via the following statements:

- `var h = document.location.hash.substring(1);`
- `document.getElementById("email").innerHTML += " (" + h + ")";`


# Detecting Issues: Burp Suite

# When To Perform Manual Analysis

- Understanding dataflow is important to confirm issues
- Learning JavaScript inbuilt-functions
- Get comfortable with:
  - Developer console
  - Debugging

# When To Perform Manual Analysis

- Browsers interpret **window.location.hash** differently



```
var h = document.location.hash.substring(1);

if (h && h != "") {

    var re = new RegExp( ".+@.+" );
    if (h.match(re)) {
        document.getElementById("email").innerHTML += " (" + h + ")";
    }
}
```

[http://demo.testfire.net/high\\_yield\\_investments.htm#<img src=x onerror=prompt\(1\)>test@test.com](http://demo.testfire.net/high_yield_investments.htm#<img src=x onerror=prompt(1)>test@test.com)

# Detecting Issues: ESLint

- Linting can identify issues in source-code

```
LON-SP-S0G8WP:js-analysis jwilkin$ eslint sub-script.js -c eslintrc-light.js
```

```

                                     ./js-analysis/sub-script.js
   9:1   error    Unsafe call to document.write for argument 0  no-unsanitized/method
  76:5   warning   Assignment to href can be unsafe              scanjs-rules/assign_to_href
 170:9   error    Unsafe assignment to innerHTML                no-unsanitized/property
 174:3   error    Unsafe assignment to innerHTML                no-unsanitized/property
 185:4   warning   Assignment to location can be unsafe          scanjs-rules/assign_to_location
 187:4   warning   Assignment to location can be unsafe          scanjs-rules/assign_to_location
 203:20  warning   The function setTimeout can be unsafe         scanjs-rules/call_setTimeout
```

✖ 7 problems (3 errors, 4 warnings)

<https://medium.com/greenwolf-security/linting-for-bugs-vulnerabilities-49bc75a61c6>

<https://eslint.org>

# Detecting Out-Of-Bound Responses

- Burp Collaborator is **king** for Out-Of-Bound Detection
  - XXE
  - SSRF
  - bXSS
  - SQLi
- Setting up your own server is a good exercise

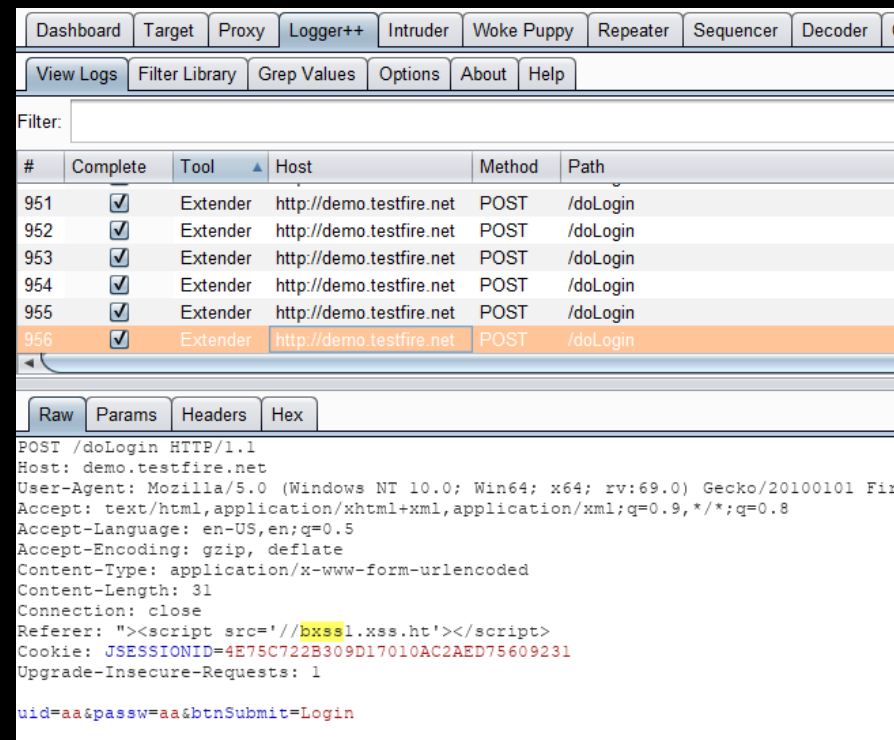
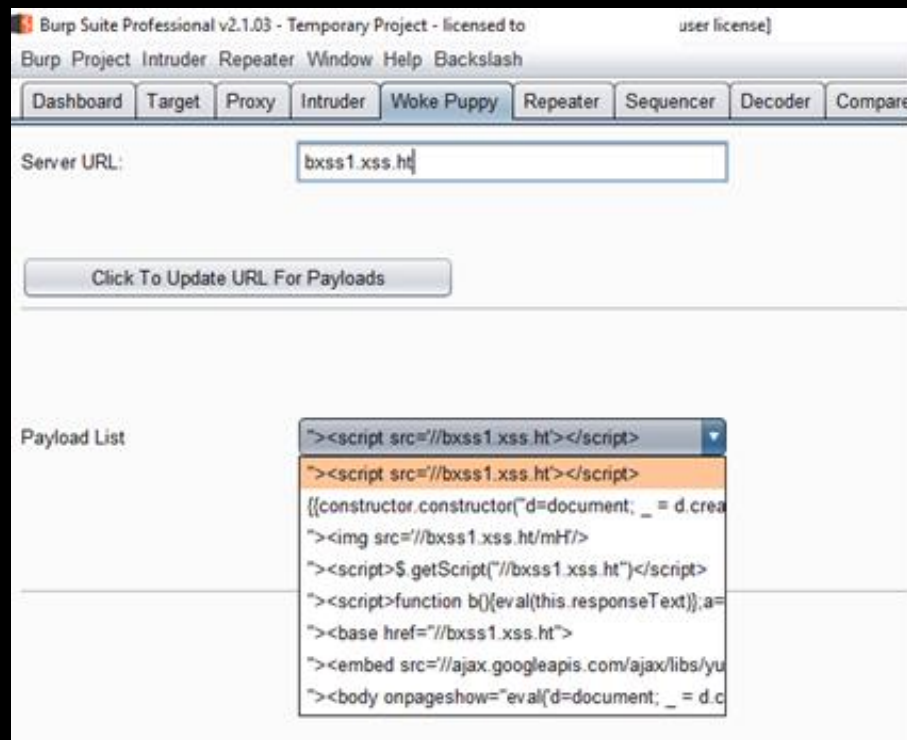
# Detecting Out-Of-Bound Responses: bXSS

- Burp Collaborator does detect bXSS, but..
- Sleepy puppy was the initial 'bXSS' tool (now deprecated)
- Current contenders:
  - XSS Hunter (Top Dog)
  - bXSS (Hot Dog)

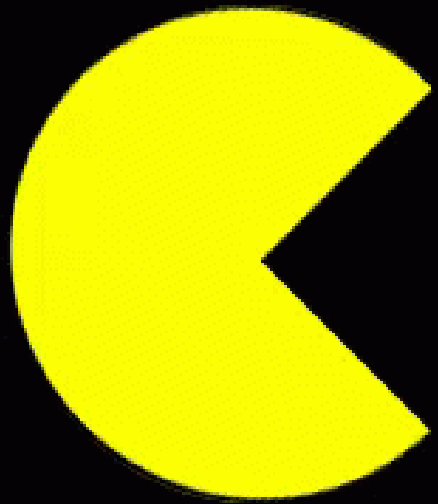
<https://github.com/Netflix-Skunkworks/sleepy-puppy>  
<https://github.com/LewisArdern/bXSS>  
<https://xsshunter.com>

# Detecting Out-Of-Bound Responses: bXSS

- Modifying SP Burp Extension to detect bXSS



<https://github.com/LewisArdern/sleepy-puppy>  
<https://ardern.io/2019/06/20/payload-bxss/>



We should not have to  
manually find 'hungry'  
regular expressions



# Detecting ReDoS

- Certain vulnerable regular expressions loop exponentially when validating specific strings
  - The expression  $\wedge(a+)+\$$  takes 65536 steps to check the string *aaaaaaaaaaaaaaaaaX*
  - The number of steps doubles for each additional “a”
  - An attacker can exploit vulnerable expressions to consume server resources and create a DoS condition



```
const regularExpression = /\w(a+)+$/  
  
const string = 'aaaaaaaaaaaaaaaaaX'  
  
regularExpression.test(string)
```

# Detecting ReDoS

- Useful reading:
  - <https://www.usenix.org/system/files/conference/usenixsecurity18/sec18-staicu.pdf>
- Useful tools for detecting ReDoS
  - <https://github.com/davisjam/vuln-regex-detector>
  - <https://www.npmjs.com/package/redos>
  - <https://github.com/substack/safe-regex>

# Detecting ReDoS With vuln-regex-detector

```
For File /mnt/c/Users/lewis/Documents/Projects/test123/javascript-only/foo/abc.js:
{ extractReport:
  { language: 'javascript',
    file: '/mnt/c/Users/lewis/Documents/Projects/test123/javascript-only/foo/abc.js',
    regexps: [ [Object] ] },
  vulnRegexes: [ '([a-z])++$' ],
  couldExtractRegexes: 1,
  file: '/mnt/c/Users/lewis/Documents/Projects/test123/javascript-only/foo/abc.js',
  checkRegexReports:
    [ { pattern: '([a-z])++$',
      validateReport: [Object],
      detectReport: [Object],
      isVulnerable: 1 } ],
  anyVulnRegexes: 1 }
```



```
var r = /([a-z])+$/
var s = 'aaaaaaaaaaaaaaaaaaaaaaaaaaaaaa!'

r.test(s)
```



We should not have to manually find API keys or secrets in source code

```
C:\Users\lewis\Documents\Projects\test123>rg -ia "GITHUB_TOKEN=" -j 12 --no-filename --no-line-number --pretty
GITHUB_TOKEN="HI Amsterdam!"
```

## Detecting Secrets

- TruffleHog  
<https://github.com/dxa4481/truffleHog>
- Ripgrep  
<https://github.com/BurntSushi/ripgrep>
- git-secrets  
<https://github.com/awslabs/git-secrets>

<https://edoverflow.com/2019/ci-knew-there-would-be-bugs-here/>



We should not have to manually identify known security issues in third-party components

## Detecting Vulnerable Third-Party Components

Example	Command
npm	npm audit
yarn	yarn audit
bower	auditjs --bower bower.json
Client-Side JavaScript	retire --js /path/
Node.js Open-Source	snyk test


<https://docs.npmjs.com/cli/audit>  
<https://retirejs.github.io/retire.js/>

# NPM Audit & RetireJS

Advisory

Request

Response

**Vulnerable version of the library 'jquery' found**

---

Issue:

Vulnerable version of the library 'jquery' found

Severity:

Low

Confidence:

Certain

Host:

https://fiddle.jshell.net

Path:

/015jxu8s/show/

---

**Note:** This issue was generated by the Burp extension: Retire.js.

**Issue detail**

The library **jquery** version 3.3.1 has known security issues.  
For more information, visit those websites:

- <https://blog.jquery.com/2019/04/10/jquery-3-4-0-released/>
- <https://nvd.nist.gov/vuln/detail/CVE-2019-11358>
- <https://github.com/jquery/jquery/commit/753d591aea698e57d6db58c9f722cd0808619b1b>

**Affected versions**

The vulnerability is affecting all versions prior 3.4.0 (between \* and 3.4.0)

```
C:\Users\lewis\Documents\Projects\metasecjs>npm audit
```

```
=== npm audit security report ===
```

```
found 0 vulnerabilities  
in 1391 scanned packages
```



# Let's Talk About Dependencies:



<https://twitter.com/kkotowicz/status/1175355336838062081?s=20>



What if we had a tool such as npm audit that could help drive secure-development and issue discovery?

# What would this look like?

## Expertise:

1. Review popular dependencies for:
  - a) Possible misconfigurations
  - b) Possible security issues
2. Add them to a repository to retrieve later

## Tool Creation:

1. Review a package.json file for its dependencies
  - a) Retrieve the name, and version
2. Look-up repository and match version and name
3. Return list of recommendations to investigate in a readable format



```
{
  "dependencies": {
    "dompurify": "^1.0.11",
    "electron": "^6.0.9",
    "express-session": "^1.16.2",
    "mongoose": "^5.7.0"
  }
}
```

# Tool Output

	A	B	C	D	E	F	G
1	Package	Version	Guidance	Outcome	Assessor Comments	References	
2	express-session	^1.62.2	Ensure that the secret is not hard-coded				
3			Ensure that the httpOnly flag is set for applications that do not need access to the cookie, e.g Angular				
4			Ensure the 'secure' flag is set to prevent sessions sent over HTTP				
5			Ensure the sameSet flag is set with 'strict' or 'relaxed'				
6			Ensure the path is set to '/'				
7			Ensure the cookie name beings with a cookie prefix __Host or __Secure				
8			Ensure that the store is not the default 'Memory' store as this can be problematic for memory exhaustion and other security issues				
9	another-package	v9001	It's over 9000!				

# Example: express-session




```
session = require('express-session')
app.use(session({
  secret: config.SESSION_SECRET,
  name: '__Host-auth-cookie',
  cookie: { httpOnly: true, secure: true, sameSite: 'strict', path: '/' },
  store: new MySQLStore({}, sqlConnection)
}));
```

# Never Store The Secret In Source Control



```
session = require('express-session')
app.use(session({
  secret: config.SESSION_SECRET,
  name: '__Host-auth-cookie',
  cookie: { httpOnly: true, secure: true, sameSite: 'strict', path: '/' },
  store: new MySQLStore({}, sqlConnection)
}));
```

# Enforce The Browser To Only Transmit Over HTTPS



```
session = require('express-session')
app.use(session({
  secret: config.SESSION_SECRET,
  name: '__Host-auth-cookie',
  cookie: { httpOnly: true, secure: true, sameSite: 'strict', path: '/' },
  store: new MySQLStore({}, sqlConnection)
}));
```

# Prevent Inclusion In Cross-Origin Requests



```
session = require('express-session')
app.use(session({
  secret: config.SESSION_SECRET,
  name: '__Host-auth-cookie',
  cookie: { httpOnly: true, secure: true, sameSite: 'strict', path: '/' },
  store: new MySQLStore({}, sqlConnection)
}));
```



# Prevent Memory Exhaustion With A Store



```
session = require('express-session')
app.use(session({
  secret: config.SESSION_SECRET,
  name: '__Host-auth-cookie',
  cookie: { httpOnly: true, secure: true, sameSite: 'strict', path: '/' },
  store: new MySQLStore({}, sqlConnection)
}));
```

# Triaged Tool Output

	A	B	C	D	E	F	G
1	Package	Version	Guidance	Outcome	Assessor Comments	References	
2	express-session	^1.62.2	Ensure that the secret is not hard-coded	Good	Uses config		
3			Ensure that the httpOnly flag is set for applications that do not need access to the cookie, e.g Angular	Good	Uses httpOnly		
4			Ensure the 'secure' flag is set to prevent sessions sent over HTTP	Good	Uses secure		
5			Ensure the sameSet flag is set with 'strict' or 'relaxed'	Good	Uses sameSite strict		
6			Ensure the path is set to '/'	Good	Sets path to '/'		
7			Ensure the cookie name beings with a cookie prefix __Host or __Secure	Good	Sets __Host on cookie name		
8			Ensure that the store is not the default 'Memory' store as this can be problematic for memory exhaustion and other security issues	Good	Uses MySQL Database!		
9	another-package	v9001	It's over 9000!	Yes	It is over 900!		



What if we had a tool which combines all the open source projects which help identify security issues and drive secure development?

- Metasec.js is a **meta analysis** tool to review **JavaScript** Applications using **open-source** tools:
- Current Functionality:
  - Help drive secure development based on package.json dependencies
  - Identifies issues with third-party dependencies
  - Looks for secrets
  - Looks for ReDoS
  - Performs security linting against an application
  - Audits the application for electron issues with doynsec/electronegativity if electron is in the package.json file

Metasec.js



Demo

# Future Work

- Currently a **PoC**, and needs a lot of work...
- Future plans:
  - Extend the 'repository' with more libraries
  - Smarter automation
    - E.g. detect when a framework/library is used and run appropriate linting/tools rather than hailmary
  - Extend or improve current capabilities
  - Link finder, and other asset finding capabilities in source-code
- Request for help!

# Conclusion

- Automation helps streamline work processes
- To get the best results you should automate from both a static and dynamic perspective
- Automation can be used to help identify security issues and drive secure development in modern JavaScript applications
- Automate and be AWESOME!



Sponsored by 

# Thank you!

Email: [lewis@ardern.io](mailto:lewis@ardern.io)

Website: <https://ardern.io>

Twitter: <https://twitter.com/LewisArdern>

GitHub: <https://github.com/LewisArdern>