Adding more Security To your Testing and Automating

Saucecon 2021, Alan Richardson

@EvilTester, EvilTester.com

Security Testing:

- a highly technical set of skills,
- a wide domain of knowledge,
- a long time to learn and practice.

Any Hacks for doing more faster?

Yes.

Augment and Extend our current approaches.

That doesn't mean you shouldn't learn Security Testing

It just isn't the fastest way to add security into your process

I did learn Security Testing (Some)

Loved it. Interesting. Challenging. Deep Dive into Technology.

Because I could Test & Automate & Code

I could see overlap and natural extensions.

A Natural Extension to Technical Exploratory Testing

If you are already:

— Using Dev Tools.
— Using Proxies.
— Reading HTML & JS.
— Pushing the Edge Cases.
— Bypassing the validation.
— etc.

A Natural Extension to Automating

If you are already:

— Writing reusable Abstraction Layers.
— Combining Libraries and tools.
— Data Parameterising your Execution paths.
— Auto Generating Test Data.
— Monitoring Logs.
— etc.

On Learning Security Testing

You already know the technology.

Next learn vulnerabilities and exploits.

RoadMaps

- portswigger.net/web-security
- www.hacker101.com
- github.com/sundowndev/hacker-roadmap
- github.com/onlurking/awesome-infosec
- github.com/ericpqmor/security-study-plan

Practice

- Bug Bounties:
  - hackerone.com
  - bugcrowd.com
  - yeswehack.com
- OWASP Vulnerable Web Applications Directory

But... this all takes a lot of time

How can we add more security to our Testing and Automating?

What does more Security mean?

— More secure system.
— More Job Security.
— More trust in our automated execution.
— More chance of finding security issues early.

Does that mean we have to "Shift Left"?

I do not like the term "shift left".

I do not want to shift left.

I just want to... test.

And do it at the time that's best.

In a dev process where testing's enmeshed.

I do not want to shift right.

I want a process that brings all problems to light.

"Secure Software" is not a sound bite.

Quality Software is Secure, and built with foresight.

P.S. my book of Children's Poetry is available now:

thereAreHats.com

What makes Testing and Security Testing hard?

— Where to aim?
— When to stop?
— Unknowns.

Chasing unknowns is expensive

Important, but expensive.

Building in - is easier to answer

— Have you added coverage of SQL Injection? Y/N
— Has the coverage been reviewed by AppSec? Y/N
— Have you fuzzed it with common SQL Injection Payloads? Y/N
— Any issues found from that? Y/N

Explore gaps in the coverage where Unknowns still exist.

Security Do's, Don'ts

Security Do's, Don'ts

- Do not reduce the security of your application to make it easier to test
  - e.g. automatable captchas, url param config (`?nocaptcha=true`)
- Do internalize your test environment
  - unsecured test environments on easy to find subdomains
- Do secure your test environment if public

Risks to Consider

- Security risks of live testing
  - test users with extensive permissions
  - test users with easy to guess usernames and passwords
  - leaving test users lying around in the environment
- Using Security Testing Tools without knowing Security Testing

Add additional tooling to augment existing testing and automating

Tooling

- Adding passive security testing into your process
  - Running automated API/GUI execution through a security proxy
  - Exploratory Testing through a proxy
    - proxy config browsers and API tooling
- OWasp ZAP can passively scan traffic we proxy through
  - But can also scan using WebDriver

Proxy Tool Scanning

— Point all test traffic through a proxy
— It 'passively' scans as you test
— check the results
— run an active scan later
— check the results

Running Proxy With Selenium

— Install Owasp ZAP
— Configure browsers with the Dynamic Certificate
— Configure Selenium To Run with Proxy

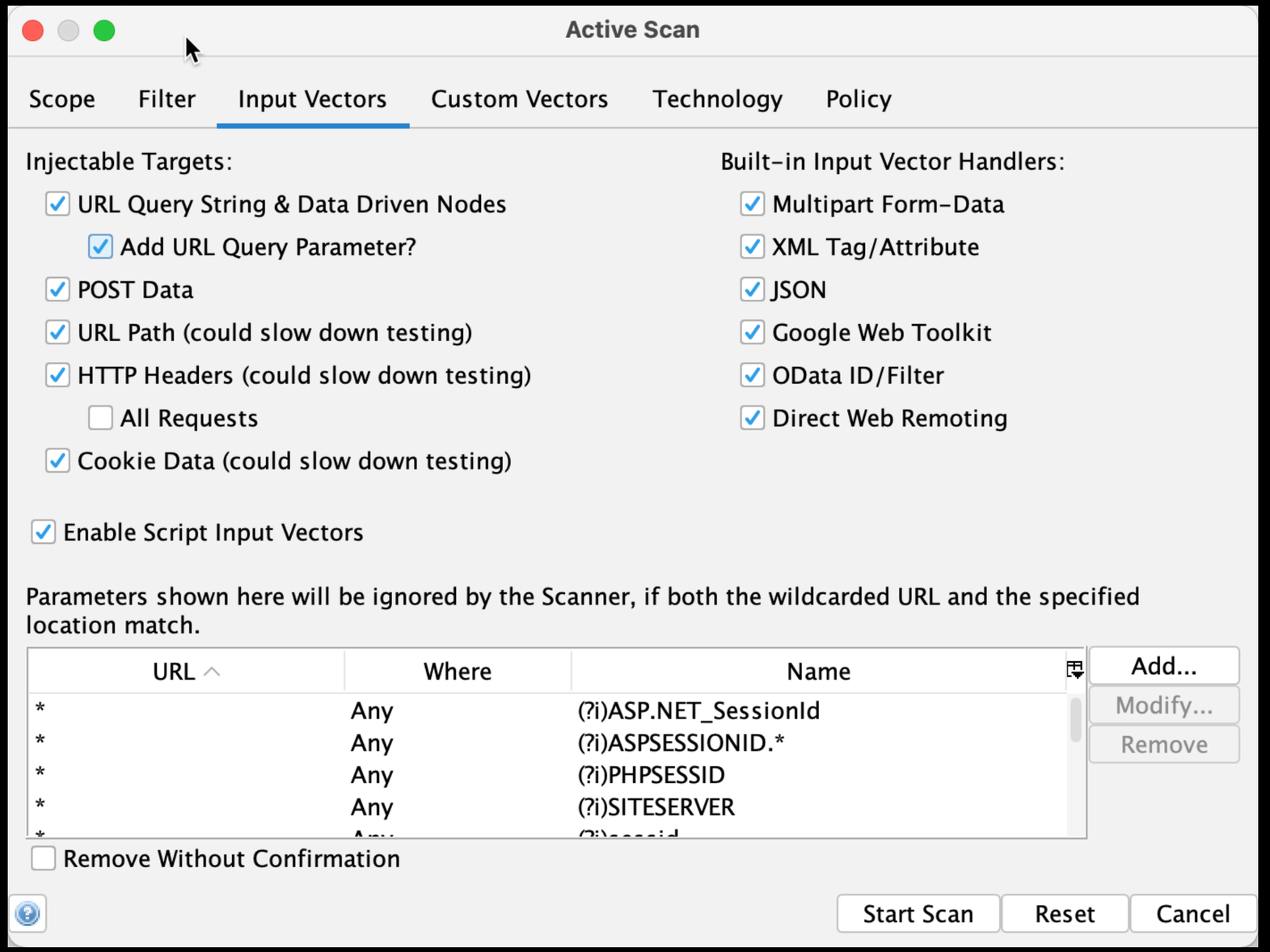
Running Proxy With Selenium

```
Proxy proxy = new Proxy();
proxy.setHttpProxy("127.0.0.1:8080");
proxy.setSslProxy("127.0.0.1:8080");
ChromeOptions options = new ChromeOptions();
options.setCapability("proxy", proxy);
driver = new ChromeDriver(options);
```

OWASP ZAP

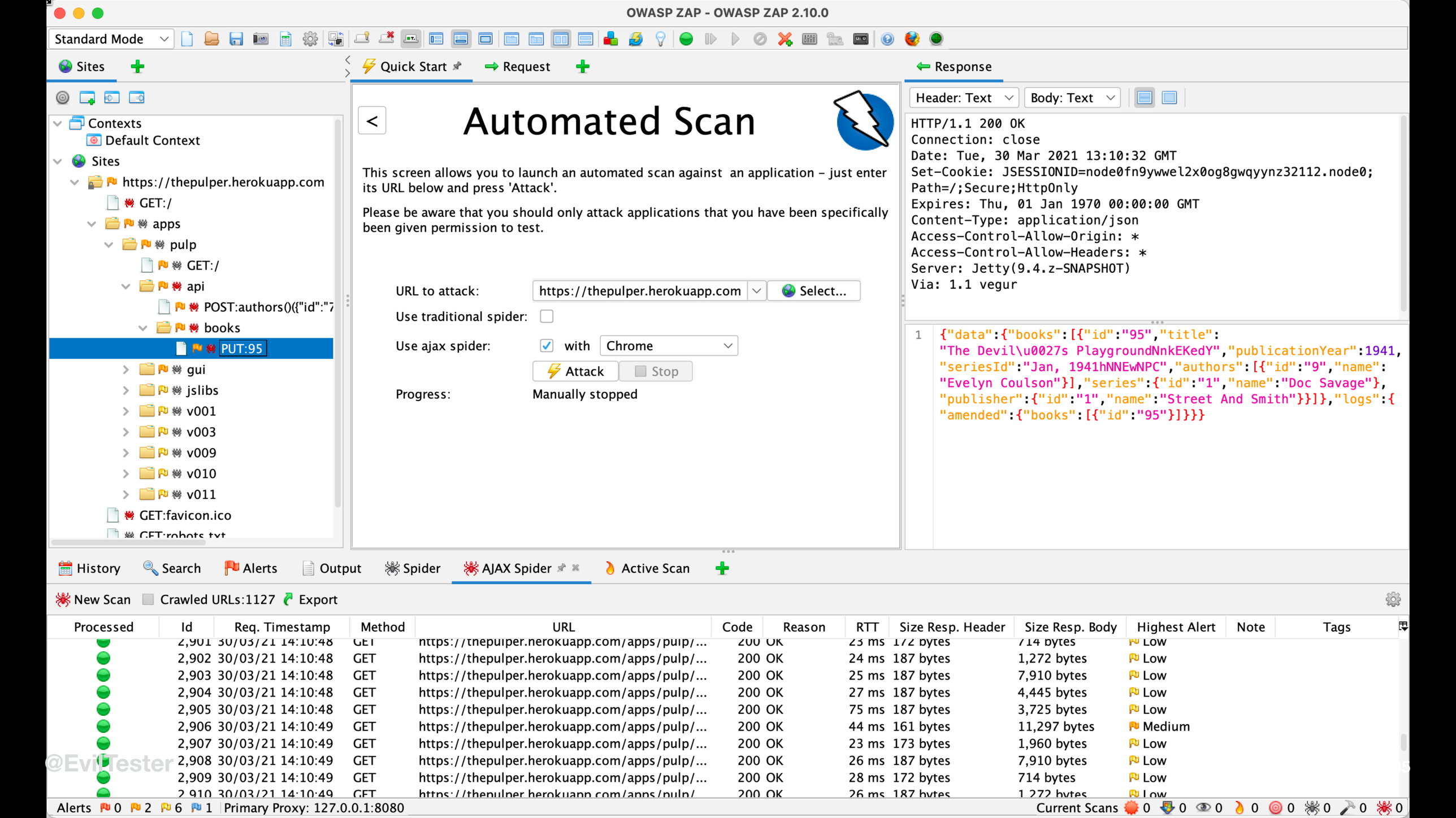
— Passive Scan for Vulnerabilities
— Build Sitemap during scan (visited, found)
— Active Scan - crawls and adds params
— Save session file as 'proof' of coverage
— Inspect sitemap for missing coverage

Active Scan Config


Active Scan
Scope
Filter
Input Vectors
Custom Vectors
Technology
Policy
Injectable Targets:
URL Query String & Data Driven Nodes
Add URL Query Parameter?
POST Data
URL Path (could slow down testing)
HTTP Headers (could slow down testing)
All Requests
Cookie Data (could slow down testing)
Built-in Input Vector Handlers:
Multipart Form-Data
XML Tag/Attribute
JSON
Google Web Toolkit
OData ID/Filter
Direct Web Remoting
Enable Script Input Vectors
Parameters shown here will be ignored by the Scanner, if both the wildcarded URL and the specified location match.
URL
Where
Name
Add...
Modify...
Remove
* Any (?i)ASP.NET_SessionId
* Any (?i)ASPSESSIONID.*
* Any (?i)PHPSESSID
* Any (?i)SITESERVER
Remove Without Confirmation
Start Scan
Reset
Cancel

Using Proxy Tools for Exploratory Testing

— Feed your browsers/API Tooling through proxy
— Test
— Revisit requests made, study, fuzz
— Save session file as 'proof' of coverage
— AJAX Spider - uses WebDriver to 'crawl' site in browser
— Fuzz Individual requests


OWASP ZAP - OWASP ZAP 2.10.0
Standard Mode
Sites
Quick Start
Request
Response
Contexts
Default Context
Sites
https://thepulper.herokuapp.com
GET:/
apps
pulp
GET:/
api
POST:authors()({"id":"7
books
PUT:95
gui
jslibs
v001
v003
v009
v010
v011
GET:favicon.ico
Automated Scan
This screen allows you to launch an automated scan against an application - just enter its URL below and press 'Attack'.
Please be aware that you should only attack applications that you have been specifically been given permission to test.
URL to attack:
https://thepulper.herokuapp.com
Select...
Use traditional spider:
Use ajax spider:
with
Chrome
Attack
Stop
Progress:
Manually stopped
Header: Text
Body: Text
HTTP/1.1 200 OK
Connection: close
Date: Tue, 30 Mar 2021 13:10:32 GMT
Set-Cookie: JSESSIONID=node0fn9ywwel2x0og8gwqyynz32112.node0;
Path=/;Secure;HttpOnly
Expires: Thu, 01 Jan 1970 00:00:00 GMT
Content-Type: application/json
Access-Control-Allow-Origin: *
Access-Control-Allow-Headers: *
Server: Jetty(9.4.z-SNAPSHOT)
Via: 1.1 vegur
1
{"data":{"books":[{"id":"95","title":
"The Devil\u0027s PlaygroundNnkEKedY","publicationYear":1941,
"seriesId":"Jan, 1941hNNEwNPC","authors":[{"id":"9","name":
"Evelyn Coulson"}],"series":{"id":"1","name":"Doc Savage"},
"publisher":{"id":"1","name":"Street And Smith"}}]},"logs":{
"amended":{"books":[{"id":"95"}]}}}
History
Search
Alerts
Output
Spider
AJAX Spider
Active Scan
New Scan
Crawled URLs:1127
Export
Processed
Id
Req. Timestamp
Method
URL
Code
Reason
RTT
Size Resp. Header
Size Resp. Body
Highest Alert
Note
Tags
2,902 30/03/21 14:10:48 GET https://thepulper.herokuapp.com/apps/pulp/... 200 OK 24 ms 187 bytes 1,272 bytes Low
2,903 30/03/21 14:10:48 GET https://thepulper.herokuapp.com/apps/pulp/... 200 OK 25 ms 187 bytes 7,910 bytes Low
2,904 30/03/21 14:10:48 GET https://thepulper.herokuapp.com/apps/pulp/... 200 OK 27 ms 187 bytes 4,445 bytes Low
2,905 30/03/21 14:10:48 GET https://thepulper.herokuapp.com/apps/pulp/... 200 OK 75 ms 187 bytes 3,725 bytes Low
2,906 30/03/21 14:10:49 GET https://thepulper.herokuapp.com/apps/pulp/... 200 OK 44 ms 161 bytes 11,297 bytes Medium
2,907 30/03/21 14:10:49 GET https://thepulper.herokuapp.com/apps/pulp/... 200 OK 23 ms 173 bytes 1,960 bytes Low
2,908 30/03/21 14:10:49 GET https://thepulper.herokuapp.com/apps/pulp/... 200 OK 26 ms 187 bytes 7,910 bytes Low
2,909 30/03/21 14:10:49 GET https://thepulper.herokuapp.com/apps/pulp/... 200 OK 28 ms 172 bytes 714 bytes Low
@EvilTester
Alerts 0 2 6 1 Primary Proxy: 127.0.0.1:8080
Current Scans 0 0 0 0 0 0 0 0

Proactive steps to improve security that are easier to adopt than learning to Hack

Learn basic secure coding

Gotchas related to the languages and libraries in use:

— Static Analysis tooling can help with this
— 1 vulnerability in your code can be exposed by 100 'hacking approaches'
— 1 vulnerability might be fixable with 1 change

Learn to spot what causes the issues

Security Testing Lessons learned to improve our automating

We use Direct Object Reference to make our automating faster.

— IDOR - Insecure Direct Object References

We parameterize our automated execution.

— Fuzzing with insecure payloads

Evaluate your Automating and Testing for Security

— Does your automation setup users, permissions, data easily?
— What are the security controls around that?
— Can anyone do it? == Insecure
— What permissions do you need? == Principle of least Privilege
— Do you mix HTTP and GUI? How? Sharing cookies? API headers? Is API access same as GUI

Adding More Security

— Security Testing is great to learn, but takes time.
— Overlap between Security Testing, Exploratory Testing, Automating
— Same tooling used
— Add to existing process.
— Finding issues in code, is easier than finding them in running app.

Security Testing is a means to an end.

We can start at the source, then augment and extend.

Roll The End Credits...

Learn to "Be Evil"

- www.eviltester.com
- @eviltester
- www.youtube.com/user/EviltesterVideos

About Alan Richardson

- www.compendiumdev.co.uk
- uk.linkedin.com/in/eviltester

Follow

— Linkedin - @eviltester
— Twitter - @eviltester
— Instagram - @eviltester
— Facebook - @eviltester
— Youtube - EvilTesterVideos
— Pinterest - @eviltester
— Github - @eviltester
— Slideshare - @eviltester

BIO

Alan is a test consultant who enjoys testing at a technical level using techniques from psychotherapy and computer science. In his spare time Alan is currently programming a multi-user text adventure game and some buggy JavaScript games in the style of the Cascade Cassette 50. Alan is the author of the books "Dear Evil Tester", "Java For Testers" and "Automating and Testing a REST API". Alan's main

Related Reading & Videos

— Integrating E2E and Application Security Testing by Abhay Bhargav
— Confessions of an Accidental Security Tester by Alan Richardson