EVALUATING COMMERCIAL WEB APPLICATION SECURITY

By Aaron Parke

Outline

- Project background – What and why?
  - Targeted sites
- Testing process
- Burp’s findings
- Technical talk
- My findings and thoughts
- Questions

Project Goals

- Test the security of several commercial web sites using Burp Suite
- Compare the security of these sites to one another
- Present solutions to any issues found on the sites as if hired by the companies to test their security
- Provide information about web application security in general
- Improve personal skills and understanding of web application security

What Is Web Application Security?

- Applications accessed through Web pages
  - Think online ordering, online forms, etc.
- Web applications have the potential for all kinds of security issues
  - Risks to users
  - Risks to companies
- Web application security involves testing sites for issues attackers could take advantage of
  - Typically, testing a site means seeing how you can manipulate it yourself
  - Looking at it from an attacker's point of view

What Is Burp Suite?

- Burp Suite is a set of tools used to expose web application vulnerabilities sold by Portswigger Web Security
- Proxy, Spider, and Scanner
- Burp has a great reputation, consistently given high ratings in the industry
- Big reason I chose Burp

Targeted Sites

- Target
- Walmart
- Jet’s Pizza
- Imo’s Pizza

Target

- Large company and very large site
  - Over 2500 pages scanned
- Previous security issues

TARGET

Walmart

- Similar company to Target
- Another huge company with a very large site
    - Similar number of pages scanned, around 2500

Jet’s Pizza

- Regional pizza chain
- Medium-sized company and site
- Around 900 pages scanned

Imo’s Pizza

- Similar company to Jet’s
- Smaller site
  - 670 pages scanned

IMO'S
PIZZA
The Square Beyond Compare

"Original St. Louis Style Pizza"

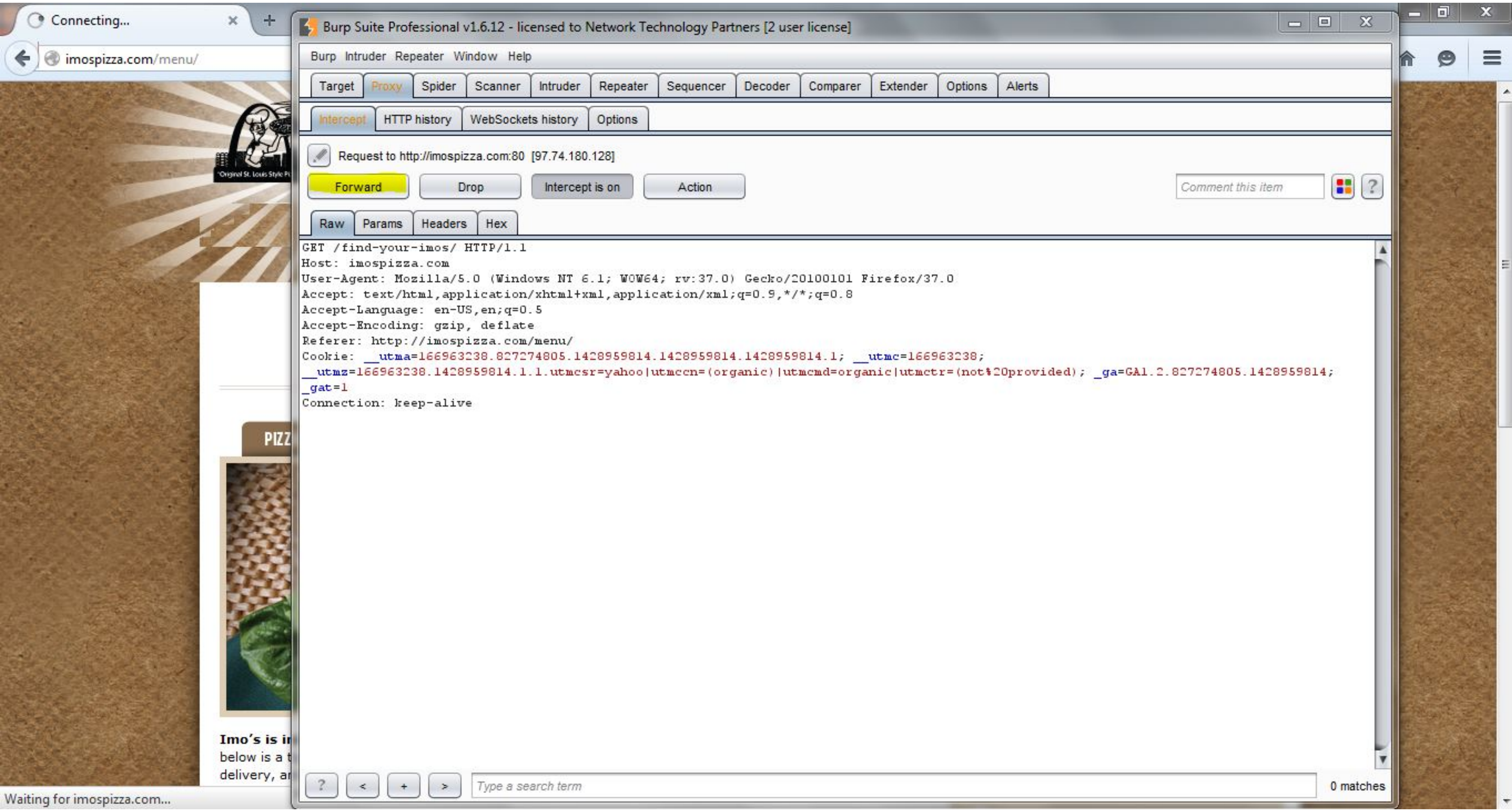
Testing Process

- Burp Proxy
- Burp Spider
- Burp Scanner
- Manual confirmation

Burp Proxy

- Proxy listener
  - Local HTTP server that listens for incoming connections from browser
  - Configured to work with Firefox
- Inspects each HTTP request and response
- User must manually advance through each request

Burp Proxy

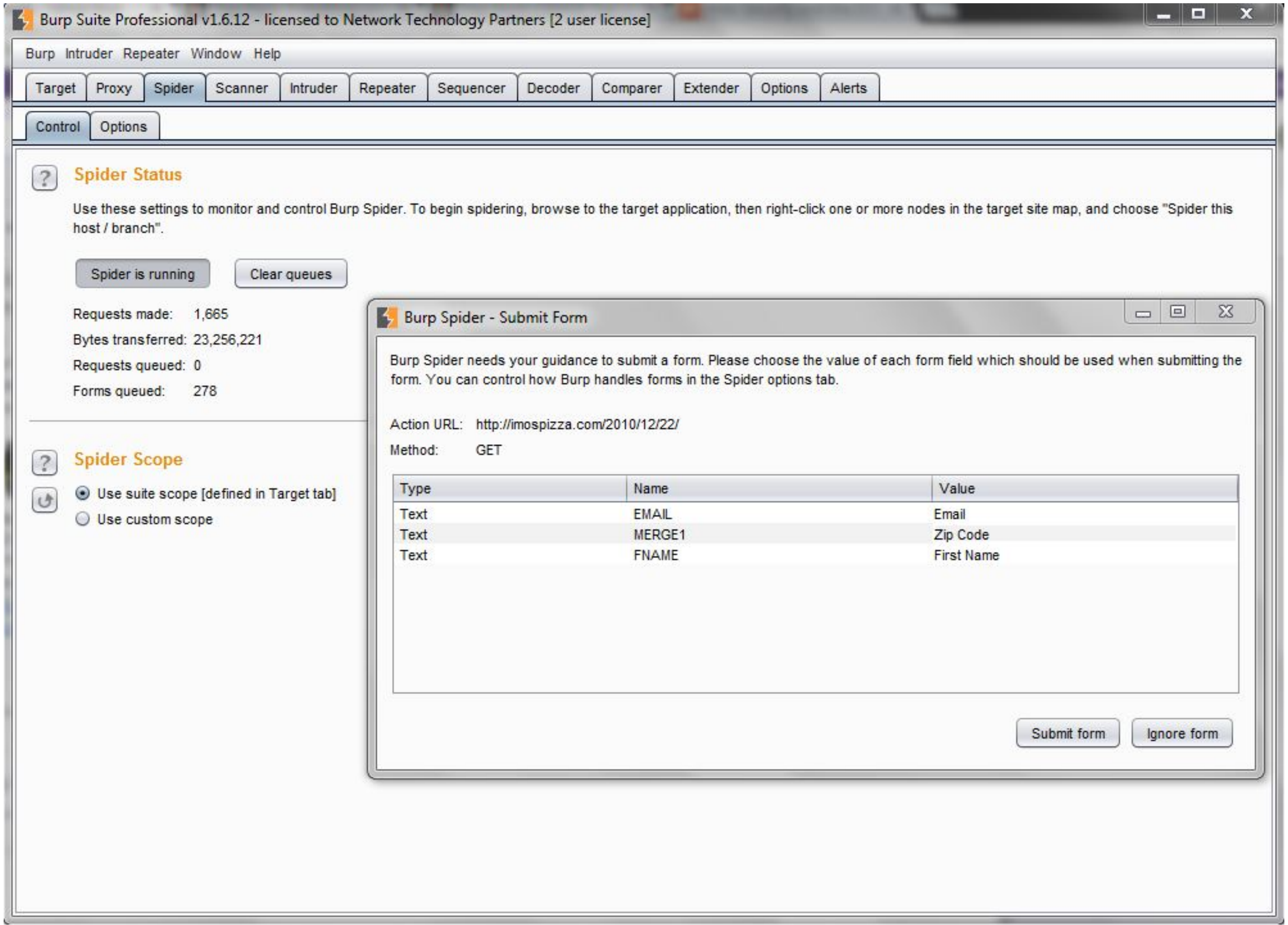

Connecting...
imospizza.com/menu/
Burp Suite Professional v1.6.12 - licensed to Network Technology Partners [2 user license]
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Intercept HTTP history WebSockets history Options
Request to http://imospizza.com:80 [97.74.180.128]
Forward
Drop
Intercept is on
Action
Comment this item
Raw Params Headers Hex
Type a search term
0 matches
Waiting for imospizza.com...

```
GET /find-your-imos/ HTTP/1.1
Host: imospizza.com
User-Agent: Mozilla/5.0 (Windows NT 6.1; WOW64; rv:37.0) Gecko/20100101 Firefox/37.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://imospizza.com/menu/
Cookie: __utma=166963238.827274805.1428959814.1428959814.1428959814.1; __utmc=166963238;
__utmz=166963238.1428959814.1.1.utmcsr=yahoo|utmccn=(organic)|utmcmd=organic|utmctr=(not%20provided); _ga=GA1.2.827274805.1428959814;
_gat=1
Connection: keep-alive
```

Burp Spider

- Basic web crawler
- Creates a map of a site by following all links and submitting forms
- Used to find all pages of a domain
    - Sent to Scanner

Burp Spider


Burp Suite Professional v1.6.12 - licensed to Network Technology Partners [2 user license]
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Control Options

Spider Status

Use these settings to monitor and control Burp Spider. To begin spidering, browse to the target application, then right-click one or more nodes in the target site map, and choose "Spider this host / branch".

Spider is running
Clear queues

Requests made:	1,665
Bytes transferred:	23,256,221
Requests queued:	0
Forms queued:	278

Spider Scope

[✓ Use suite scope [defined in Target tab]]
Use custom scope

Burp Spider - Submit Form

Burp Spider needs your guidance to submit a form. Please choose the value of each form field which should be used when submitting the form. You can control how Burp handles forms in the Spider options tab.

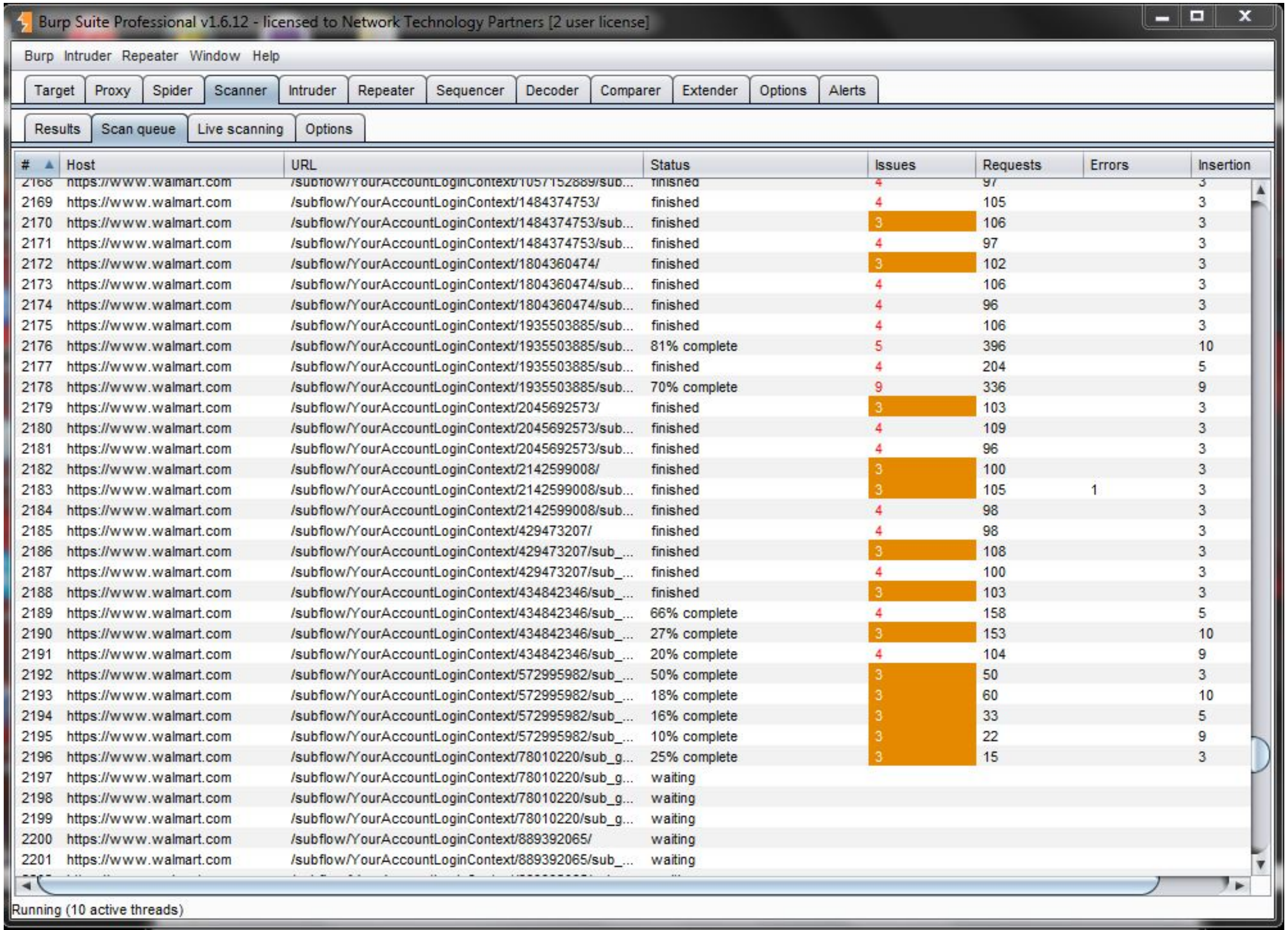
Action URL: http://imospizza.com/2010/12/22/
Method: GET

Type	Name	Value
Text	EMAIL	Email
Text	MERGE1	Zip Code
Text	FNAME	First Name

Burp Scanner

- Checks for common vulnerabilities
  - For example, Burp will check each page for the possibility of XSS by sending a payload of random numbers and seeing how the application responds
- The Scanner gives ratings for both severity and confidence
  - Severity – Informational, Low, Medium, High
  - Confidence – Tentative, Firm, Certain

Burp Scanner

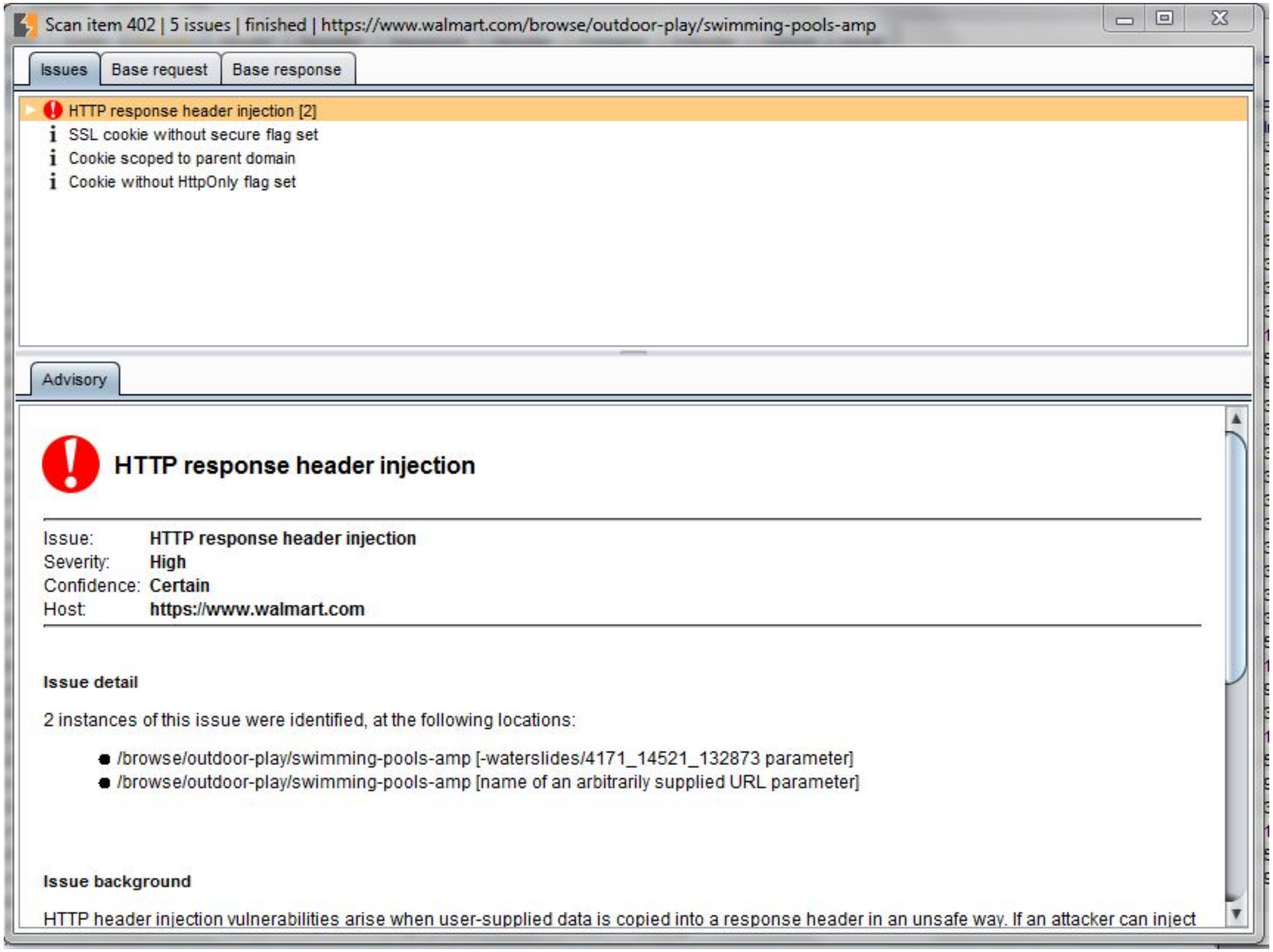


Burp Suite Professional v1.6.12 - licensed to Network Technology Partners [2 user license]
Burp Intruder Repeater Window Help
Target Proxy Spider Scanner Intruder Repeater Sequencer Decoder Comparer Extender Options Alerts
Results Scan queue Live scanning Options

#	Host	URL	Status	Issues	Requests	Errors	Insertion
2168	https://www.walmart.com	/subflow/YourAccountLoginContext/1057152889/sub...	finished	4	97		3
2169	https://www.walmart.com	/subflow/YourAccountLoginContext/1484374753/	finished	4	105		3
2170	https://www.walmart.com	/subflow/YourAccountLoginContext/1484374753/sub...	finished	3	106		3
2171	https://www.walmart.com	/subflow/YourAccountLoginContext/1484374753/sub...	finished	4	97		3
2172	https://www.walmart.com	/subflow/YourAccountLoginContext/1804360474/	finished	3	102		3
2173	https://www.walmart.com	/subflow/YourAccountLoginContext/1804360474/sub...	finished	4	106		3
2174	https://www.walmart.com	/subflow/YourAccountLoginContext/1804360474/sub...	finished	4	96		3
2175	https://www.walmart.com	/subflow/YourAccountLoginContext/1935503885/sub...	finished	4	106		3
2176	https://www.walmart.com	/subflow/YourAccountLoginContext/1935503885/sub...	81% complete	5	396		10
2177	https://www.walmart.com	/subflow/YourAccountLoginContext/1935503885/sub...	finished	4	204		5
2178	https://www.walmart.com	/subflow/YourAccountLoginContext/1935503885/sub...	70% complete	9	336		9
2179	https://www.walmart.com	/subflow/YourAccountLoginContext/2045692573/	finished	3	103		3
2180	https://www.walmart.com	/subflow/YourAccountLoginContext/2045692573/sub...	finished	4	109		3
2181	https://www.walmart.com	/subflow/YourAccountLoginContext/2045692573/sub...	finished	4	96		3
2182	https://www.walmart.com	/subflow/YourAccountLoginContext/2142599008/	finished	3	100		3
2183	https://www.walmart.com	/subflow/YourAccountLoginContext/2142599008/sub...	finished	3	105	1	3
2184	https://www.walmart.com	/subflow/YourAccountLoginContext/2142599008/sub...	finished	4	98		3
2185	https://www.walmart.com	/subflow/YourAccountLoginContext/429473207/	finished	4	98		3
2186	https://www.walmart.com	/subflow/YourAccountLoginContext/429473207/sub_...	finished	3	108		3
2187	https://www.walmart.com	/subflow/YourAccountLoginContext/429473207/sub_...	finished	4	100		3
2188	https://www.walmart.com	/subflow/YourAccountLoginContext/434842346/sub_...	finished	3	103		3
2189	https://www.walmart.com	/subflow/YourAccountLoginContext/434842346/sub_...	66% complete	4	158		5
2190	https://www.walmart.com	/subflow/YourAccountLoginContext/434842346/sub_...	27% complete	3	153		10
2191	https://www.walmart.com	/subflow/YourAccountLoginContext/434842346/sub_...	20% complete	4	104		9
2192	https://www.walmart.com	/subflow/YourAccountLoginContext/572995982/sub_...	50% complete	3	50		3
2193	https://www.walmart.com	/subflow/YourAccountLoginContext/572995982/sub_...	18% complete	3	60		10
2194	https://www.walmart.com	/subflow/YourAccountLoginContext/572995982/sub_...	16% complete	3	33		5
2195	https://www.walmart.com	/subflow/YourAccountLoginContext/572995982/sub_...	10% complete	3	22		9
2196	https://www.walmart.com	/subflow/YourAccountLoginContext/78010220/sub_g...	25% complete	3	15		3
2197	https://www.walmart.com	/subflow/YourAccountLoginContext/78010220/sub_g...	waiting				
2198	https://www.walmart.com	/subflow/YourAccountLoginContext/78010220/sub_g...	waiting				
2199	https://www.walmart.com	/subflow/YourAccountLoginContext/78010220/sub_g...	waiting				
2200	https://www.walmart.com	/subflow/YourAccountLoginContext/889392065/	waiting				
2201	https://www.walmart.com	/subflow/YourAccountLoginContext/889392065/sub_...	waiting				

Running (10 active threads)

Burp Scanner



Scan item 402 | 5 issues | finished | https://www.walmart.com/browse/outdoor-play/swimming-pools-amp

- HTTP response header injection [2]
- SSL cookie without secure flag set
- Cookie scoped to parent domain
- Cookie without HttpOnly flag set

HTTP response header injection

Issue:	**HTTP response header injection**
Severity:	**High**
Confidence:	**Certain**
Host:	**https://www.walmart.com**

Issue detail

2 instances of this issue were identified, at the following locations:

- /browse/outdoor-play/swimming-pools-amp [-waterslides/4171_14521_132873 parameter]
- /browse/outdoor-play/swimming-pools-amp [name of an arbitrarily supplied URL parameter]

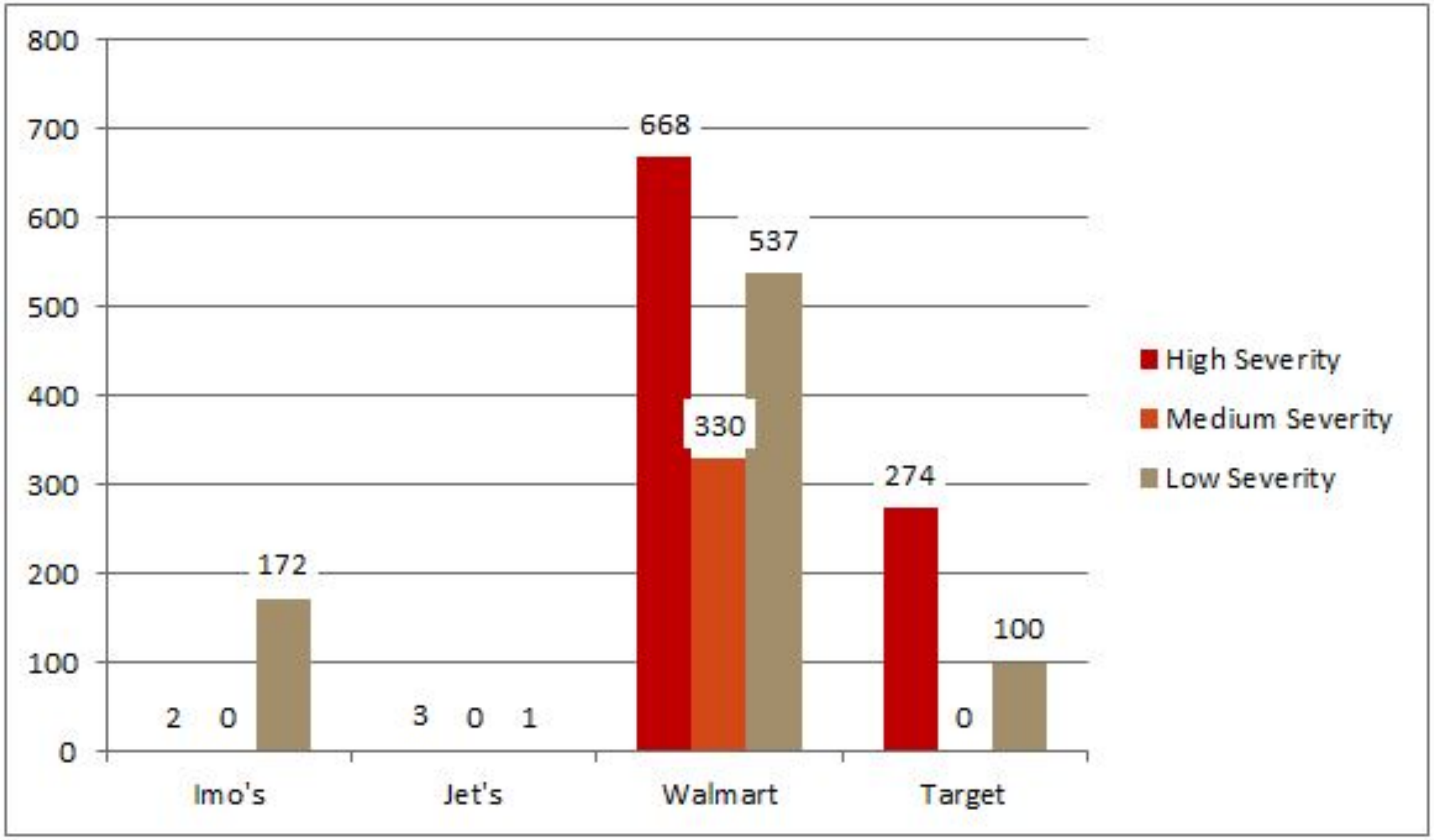
Issue background

HTTP header injection vulnerabilities arise when user-supplied data is copied into a response header in an unsafe way. If an attacker can inject

Manual Confirmation

- This is where the bulk of the work comes in – Burp informs users some vulnerabilities may be false positives, and the only way to find them is to test for the issues manually
- Checked for:
  - XSS
    - Reflected
    - DOM-based
  - Xpath Injection
  - LDAP Injection
  - HTTP Response Header Injection
  - Other less severe vulnerabilities

Initial Results



SQL Injection

- Open Web Application Security Project (OWASP) definition:
  - A SQL injection attack consists of insertion or "injection" of a SQL query via the input data from the client to the application.
- On a site where a user could enter their username to see their information, the SQL statement sent to the server might look like:
  - SELECT * FROM table WHERE user = ‘input’;
- This statement could be injected with the following input to retrieve information for all users:
  - ‘ OR ‘2’=‘2

SQL Injection

- Which would send this statement to the server:
  - SELECT * FROM table WHERE user = ‘’ OR ‘2’=‘2’;
- Escape characters
- Risks of SQL injection testing
- In several cases, Burp suspected SQL vulnerabilities in spots where it could be tested nonintrusively

Target SQL Injection

- Burp suspected SQL vulnerabilities through HTTP headers or parameters on 21 pages
  - Tested without the risk of dropping or modifying tables
  - Entered ‘ as value for parameter in first request and received error, ‘’ as value the second time with no error
- Using www.hurl.it to modify and send HTTP requests, I was able to duplicate these results on 15 of the pages
- These elements are likely vulnerable to SQL injection

Walmart SQL Injection

- Burp tried a similar nonintrusive technique on some of Walmart's parameters and cookies and suspected vulnerabilities on 658 pages
  - The advisory message read that the "two requests resulted in different responses"
- None of the reported instances were accurate
  - Entering "or true" values generated an Access Denied error, whereas Target gave a general error in completing the request
  - Entering "or false" the site behaved normally
- Could be viewed as concerning that it's possible to exit the correct context, but it would not be possible to read any data

Cross-site Scripting (XSS)

- OWASP definition:
  - Cross-Site Scripting (XSS) attacks are a type of injection, in which malicious scripts are injected into otherwise benign and trusted web sites. XSS attacks occur when an attacker uses a web application to send malicious code, generally in the form of a browser side script, to a different end user.
- Basically, an attacker finds an issue that allows them to modify a page’s code, then delivers that modified code to a victim
- 2 types detected by Burp in scanning
  - Reflected (malicious script reaches server before returning to user)
  - DOM-based (client-side only, performance of the page changes rather than the page itself)

Walmart XSS

- Burp reported 2 instances of reflected XSS on Walmart pages
  - Inaccurate
  - “<“ sanitized to “%lt”
- 2 obscure pages contained 3 possibly accurate instances of possible DOM-based XSS vulnerabilities
  - These pages read the location property from the user

Target XSS

- 245 reported instances of DOM-based XSS
  - The pages do use the location property, but it does not write the data to the page in a way that this cannot be exploited
- Also 1 reported instance of reflected XSS on the store locator page
  - Inaccurate

Jet’s XSS

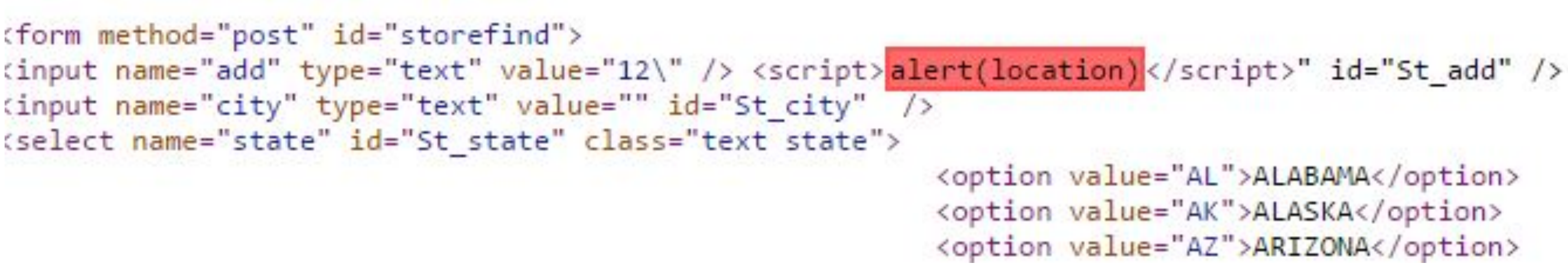
- 3 instances of DOM-based XSS were reported from one of the Javascript assets
  - Just like Target, the code does uses the location property but in a safe way

Imo’s XSS

- Imo’s did have an actual instance of reflected XSS vulnerability
- When searching for a nearby store, it is possible to exit entering data into the address field and inject script into the page
    - Page sanitizes ‘ or “
    - Does not sanitize < or >
    - Wasn’t able to create custom alert, but could use variable
- I used this string in the address field to demonstrate
    - “ /> <script>alert(location)</script>

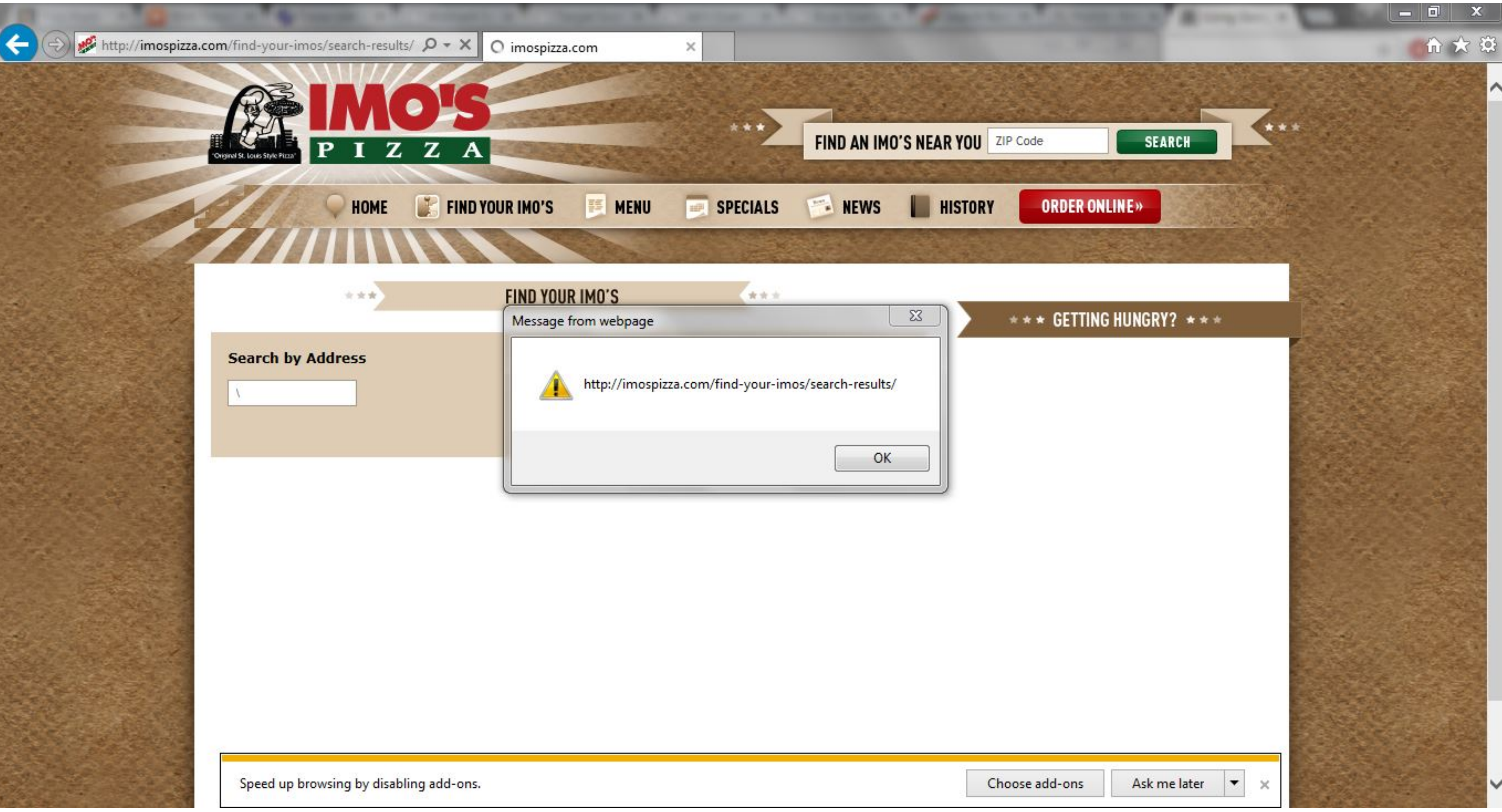
Chrome's Response

- Chrome recognizes this as XSS. Viewing the source code shows:



```
form method="post" id="storefind">
input name="add" type="text" value="12\" /> <script>alert(location)</script>" id="St_add" />
input name="city" type="text" value="" id="St_city"  />
select name="state" id="St_state" class="text state">
                                                        <option value="AL">ALABAMA</option>
                                                        <option value="AK">ALASKA</option>
                                                        <option value="AZ">ARIZONA</option>
```

Internet Explorer's Response



http://imospizza.com/find-your-imos/search-results/

Message from webpage

http://imospizza.com/find-your-imos/search-results/

Speed up browsing by disabling add-ons.

Cross-Origin Resource Sharing (CORS)

- Burp definition:
  - The HTML5 cross-origin resource sharing policy controls whether and how content running on other domains can perform two-way interaction with the domain which publishes the policy. If another domain is allowed by the policy, then that domain can potentially attack users of the application.
- Target's gift registry pages allow for CORS
- These pages allow access to requests from arbitrary domains
  - Probably any page would have access

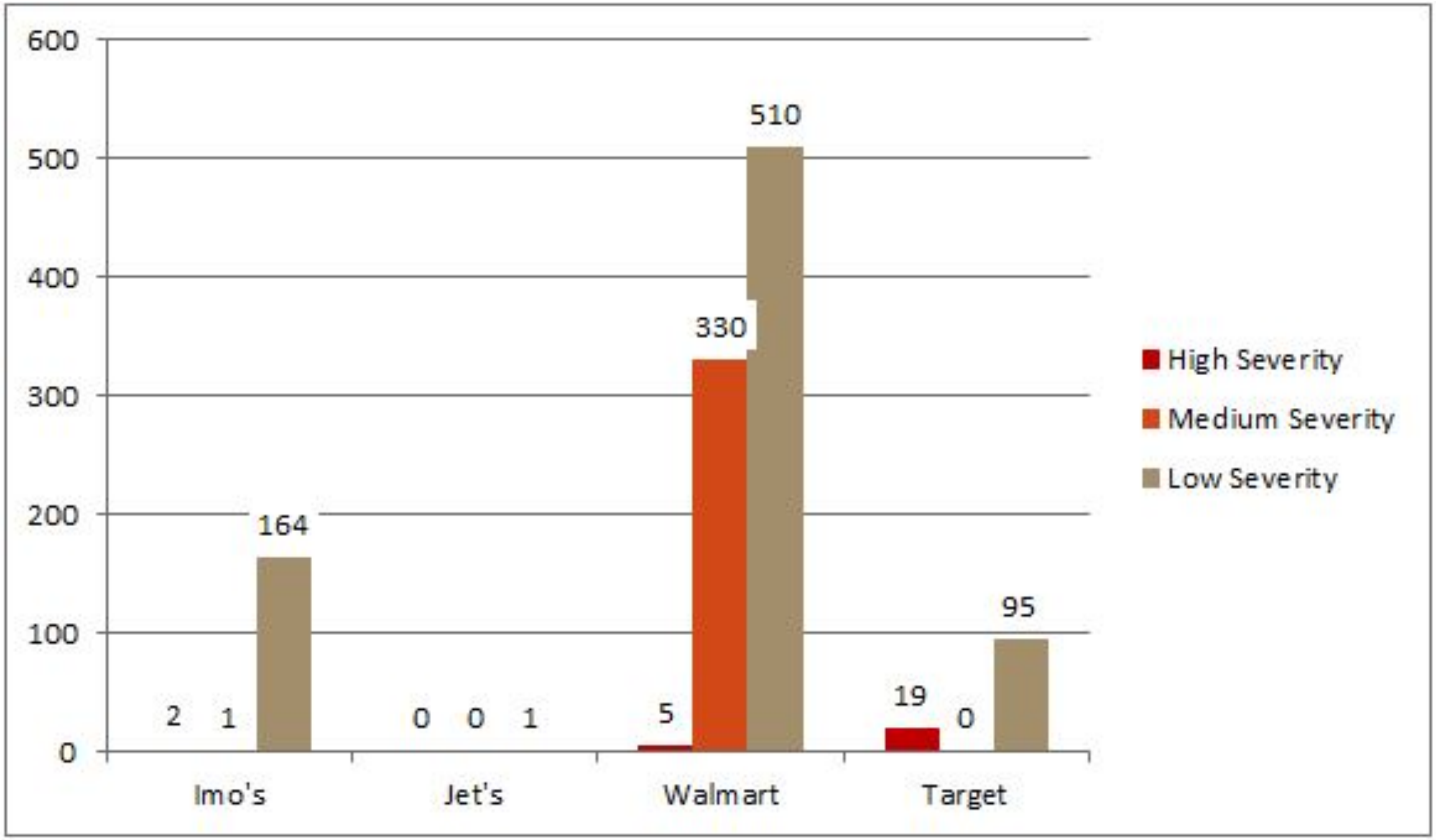
HTTP Response Header Injection

- Response header injection occurs when pages add user input into HTTP headers in an unsafe way
- Similar to XSS in that an attacker modifies how the page responds by adding their own content
- This was detected on one of Walmart’s pages
  - The page allowed for a randomly created URL parameter to be included in the Location header
  - Sanitized
- In the remediation report, I moved this from a High-severity vulnerability to Low

Open Redirection

- OWASP definition:
  - An open redirect is an application that takes a parameter and redirects a user to the parameter value without any validation. This vulnerability is used in phishing attacks to get users to visit malicious sites without realizing it.
- Imo's was vulnerable on one of its shop pages to open redirection
  - On this page, you can enter anything you want as the value for the ReturnTo parameter, and the user will be sent there
  - Could be used in phishing attacks
- In remediation report, bumped this from a Low vulnerability to Medium

Final Results



Final Results – Imo's

- Find-a-store feature vulnerable to Reflected XSS
- One shop page vulnerable to Open Redirection
- The page to login to the company's Wordpress page submits the password in cleartext
    - An attacker monitoring network traffic could easily steal a password

Final Results – Imo’s

- While investigating the cleartext password issue, I found that besides the ordering pages, Imo’s does not allow for HTTPS connection
- Imo’s has some serious security issues, but it being the smallest company and least used website this is to be expected

Final Results – Jet’s

- Jet’s was very impressive – grand total of 1 Low vulnerability
  - It issued one cookie that didn’t have the HTTP-only flag set
  - Setting this flag can prevent scripting attacks from retrieving the cookie’s value
- Being the second smallest company, I did not expect Jet’s to have such a secure site

Final Results - Walmart

- Decent security in my opinion
  - ASP.NET tracing enabled on one page, without knowing the site's infrastructure it's hard to say how much this exposes
  - Possible DOM-based XSS on 3 obscure pages
- Shockingly high numbers of Low (510) and Medium (330) issues
  - Mostly due to flags not being set on HTTP and SSL cookies
- Being one of the largest companies in the world with a huge amount of online business, Walmart can be expected to have tight web security
- I would advise Walmart to look into the issues mentioned above, which are likely very easy to fix

Final Results - Target

- Target had the most High-severity vulnerabilities of any site
  - 15 pages likely vulnerable to SQL injection
  - Cross-origin resource sharing on registry pages
- Also 95 Low vulnerabilities due to cookie issues
- Except for the possibility of being exposed to SQL injection, Target’s site is fairly secure
  - Hard to say without trying to extract data if SQL really is an issue

Challenges and Final Thoughts

- Researching each vulnerability enough to understand what it was, how it was manipulated by Burp, and how to manipulate it myself was very time-consuming
  - Researching syntax for SQL, XPATH, LDAP, HTML entities, URL encoding
  - Many hours spent examining HTTP requests and responses, especially headers and parameters
- Also difficult to find ways to test some of the vulnerabilities ethically
  - SQL frustration
- Worked to understand what were at times some pretty huge and complex HTML files

Challenges and Final Thoughts

- Thoughts on Burp:
  - Hyper-sensitive
  - Needs manual confirmation

QUESTIONS?