

INFORMATION DISCLOSURE

(Owasp ref.: Security Misconfiguration)

Disclaimer:

This is just for an educational purpose and as a method of cyber security awareness. I highly recommend not to harm any institution/Organization without proper permissions and ROE.

Author:

Hey guys, I am vijay reddy. An enthusiastic guy who loves to explore and learn more about cyber security and other fields. I have worked on both INFRA and application security.

Inspiration:

We have many things to learn and knowledge to share.

Art of Thanks:

Thanks to my friends and family.

Bibliography:

Port swigger (<https://portswigger.net/>)

Hacksplanning (<https://www.hacksplaining.com/prevention/information-leakage>)

Tool: Burp suite

Information disclosure vulnerabilities

When a website unintentionally reveals sensitive information to its users. Depending on the context, websites may leak all kinds of information to a potential attacker, including:

- Data about other users, such as usernames or financial information.
- Sensitive commercial or business data.
- Technical details about the website and its infrastructure.

Examples of information disclosure?

- Revealing the names of hidden directories, their structure, and their contents via a robots.txt file or directory listing.
- Providing access to source code files via temporary backups
- Explicitly mentioning database table or column names in error messages
- Unnecessarily exposing highly sensitive information, such as credit card details
- Hard-coding API keys, IP addresses, database credentials, and so on in the source code
- Hinting at the existence or absence of resources, usernames, and so on via subtle differences in application behaviour

What leads to information disclosure?

- Failure to remove internal content from public content. – Dev Comments
- Insecure configuration of the website and related technologies. – Debug / verbose
- Flawed design and behaviour of the application. - Error

Impact:

- Impact increases when it is available to person who can exploit this.
- When a zero-day vulnerability is discovered, hackers will immediately try to find a way to exploit it. If your site leaks information about the technology you use, you could well become subject to automated attacks.

Severity: Depend on type of data been disclosed.

Prevention:

- Developer must be aware of this
- Audit code as a part of QA
- Use generic error message instead of more descriptive or informational.
- Double-check if debug and diagnostic is disabled.
- Study of used 3rd party apps.
- Sanitize Data Passed to the Client

Identification:

High-level techniques and tools that you can use to help identify information disclosure vulnerabilities during testing.

- Fuzzing
- Using Burp Scanner
- Using Burp's engagement tools
- Engineering informative responses

Fuzzing

Try submitting unexpected data types and specially crafted fuzz strings

Automate much of this process using tools such as Burp Intruder.

- Add payload positions to parameters and use pre-built wordlists of fuzz strings to test a high volume of different inputs in quick succession.
- Easily identify differences in responses by comparing HTTP status codes, response times, lengths, and so on.
- Use grep matching rules to quickly identify occurrences of keywords, such as error, invalid, SELECT, SQL, and so on.
- Apply grep extraction rules to extract and compare the content of interesting items within responses.

You can also use the Logger++ extension, available from the BApp store. In addition to logging requests and responses from all of Burp's tools, it allows you to define advanced filters for highlighting interesting entries. This is just one of the many Burp extensions that can help you find any sensitive data that is leaked by the website.

- Burp Scanner (Alert for pvt key, email, CC no in response)
- Burp engagement tool (Simply right click the request/proxy)
- Search (expression for fine-tune with adv search option)
- Find comments
- Discover content (Add content & functionality which is not linked from website visible content)
- Engineering informative responses (Study error msg)

Common sources of information disclosure

- Files for web crawlers (/robots.txt & /sitemap.xml)
- Directory listings (list directories that do not have an index page present)
- Developer comments (hidden dir. or app logic)
- Error messages (Ver disclosure)
- Debugging data (Key, Hostname, File, encryption method)
- User account pages (Accessing other user data through IDOR e.g.?user=carlos)
- Backup files (Source code, logic, Hard coded key. Check ~ or .bak)
- Insecure configuration (TRACE)
- Version control history (/.git)

LAB 1: Information disclosure in error messages

There was a functionality in website to view full data of a product by simply clicking on it.

When reviewing the URL found that the product is mapped/retrieved with product id(integer).

Then I tried some special character, which then encountered an internal server error revealing the information about the technology and version used.



```
Internal Server Error: java.lang.NumberFormatException: For input string: ""
    at java.base/java.lang.NumberFormatException.forInputString(NumberFormatException.java:67)
    at java.base/java.lang.Integer.parseInt(Integer.java:654)
    at java.base/java.lang.Integer.parseInt(Integer.java:786)
    at lab.i.h.v.x.b(Unknown Source)
    at lab.z.z.e.m.i(Unknown Source)
    at lab.z.z.l.s.q.P(Unknown Source)
    at lab.z.z.l.e.lambda$handleSubRequest$0(Unknown Source)
    at m.k.i.n.lambda$null$3(Unknown Source)
    at m.k.i.n.N(Unknown Source)
    at m.k.i.n.lambda$uncheckedFunction$4(Unknown Source)
    at java.base/java.util.Optional.map(Optional.java:260)
    at lab.z.z.l.e.W(Unknown Source)
    at lab.s.h.w.i.k(Unknown Source)
    at lab.z.z.t.k(Unknown Source)
    at lab.s.h.w.a.X(Unknown Source)
    at lab.s.h.w.a.Z(Unknown Source)
    at m.k.i.n.lambda$null$3(Unknown Source)
    at m.k.i.n.N(Unknown Source)
    at m.k.i.n.lambda$uncheckedFunction$4(Unknown Source)
    at lab.s.e.g.W(Unknown Source)
    at lab.s.h.w.a.V(Unknown Source)
    at lab.s.h.c.d.q(Unknown Source)
    at lab.s.h.o.C(Unknown Source)
    at lab.s.l.p(Unknown Source)
    at lab.s.l.X(Unknown Source)
    at lab.s.l.I(Unknown Source)
    at m.k.e.u.p.m(Unknown Source)
    at m.k.e.u.p.M(Unknown Source)
    at m.k.e.u.p.run(Unknown Source)
    at java.base/java.util.concurrent.ThreadPoolExecutor.runWorker(ThreadPoolExecutor.java:1136)
    at java.base/java.util.concurrent.ThreadPoolExecutor$Worker.run(ThreadPoolExecutor.java:635)
    at java.base/java.lang.Thread.run(Thread.java:833)

Apache Struts 2 2.3.31
```

As seen above information about the technology and version are disclosed.

This helps an attacker to find and try known exploit.

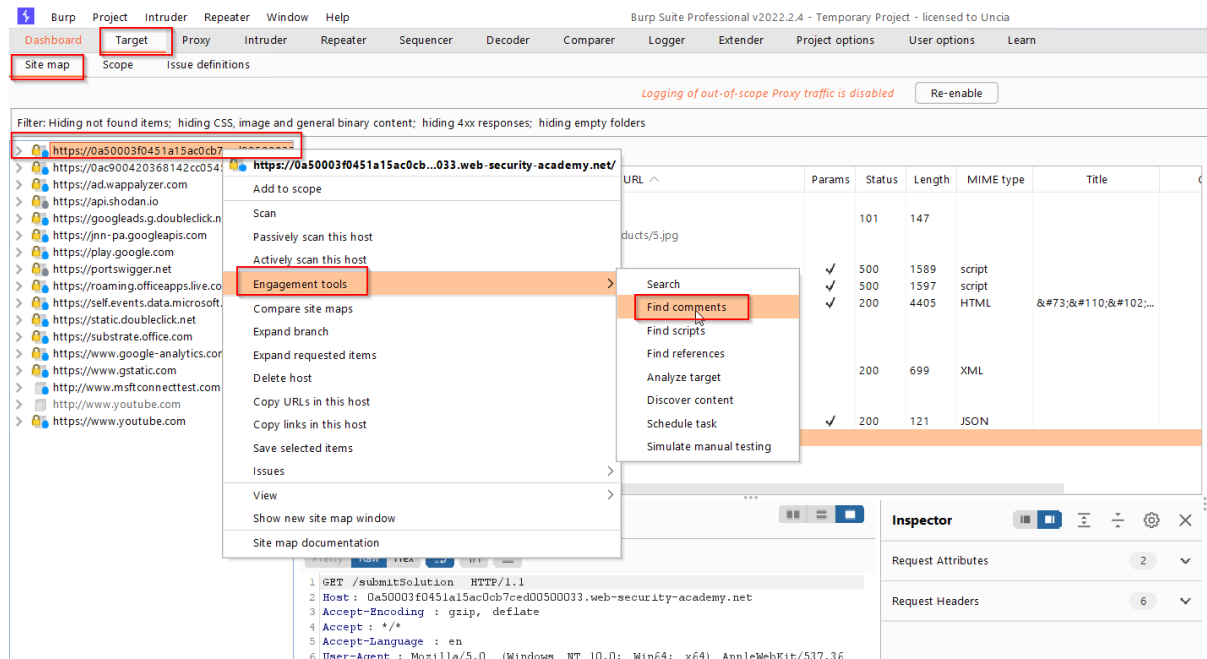
They are more dangerous if older or vulnerable versions are running on.

Cause: Improper error handling.

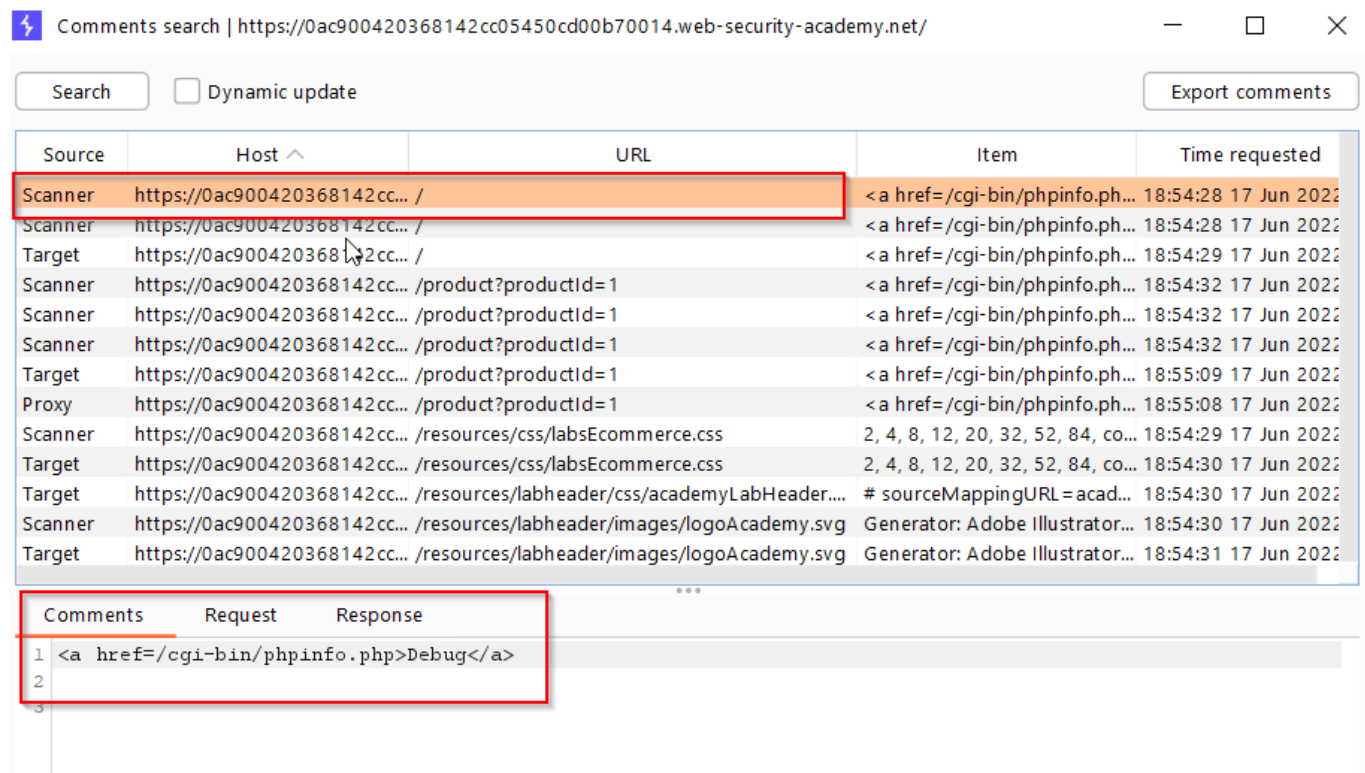
LAB 2: Information disclosure on debug page

At the time of development and testing many of the developers add comments to understand and mark the functionality also when troubleshooting or debugging a code.

So, to find such comments. I have used "find comments" option available under section "Engagement tools" of burpsuite.



Post search found a debug path with phpinfo.php file location.



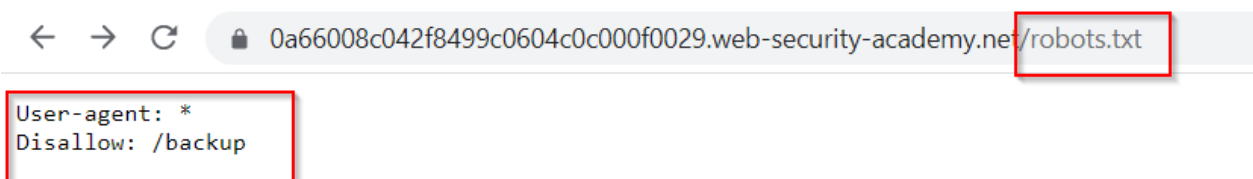
Output of /phpinfo.php

0ac900420368142cc05450cd00b70014.web-security-academy.net/cgi-bin/phpinfo.php		
Directive	Local Value	Master Value
zlib.output_compression	Off	Off
zlib.output_compression_level	-1	-1
zlib.output_handler	no value	no value
Additional Modules		
Module Name		
Environment		
Variable	Value	
GATEWAY_INTERFACE	CGI/1.1	
SUDO_GID	10000	
REMOTE_HOST	103.199.136.20	
USER	carlos	
HTTP_SEC_CH-UA	"Not A.Brand";v="99", "Chromium";v="102", "Google Chrome";v="102"	
SECRET_KEY	f1szkl7nv0gl6p8dr2a6kj8gz3wdne	
HTTP_SEC_FETCH_USER	?1	
QUERY_STRING	no value	
HOME	/home/carlos	

It contains sensitive information of servers on which this website is hosted also reveals the crucial data such as secret_key and others.

LAB 3: Source code disclosure via backup files

As a test case tried /robots.txt to find the allowed path.
Found /backup as shown below.



Show simply I changed the URL to /backup. Which leads to other vulnerability directory listing and found one backup file (.bak)



Index of /backup

Name	Size
ProductTemplate.java.bak	1643B

When clicked on it redirects to another URL. Which was revealing the whole source code which can be used to analyse the code and found the loop wholes.

```

← → ↺ 0a66008c042f8499c0604c0c000f0029.web-security-academy.net/backup/ProductTemplate.java.bak

private transient Product product;

public ProductTemplate(String id)
{
    this.id = id;
}

private void readObject(ObjectInputStream inputStream) throws IOException, ClassNotFoundException
{
    inputStream.defaultReadObject();

    ConnectionBuilder connectionBuilder = ConnectionBuilder.from(
        "org.postgresql.Driver",
        "postgresql",
        "localhost",
        5432,
        "postgres",
        "postgres",
        "t89bwymklotiaiztofmmtqdm90w9scw"
    ).withAutoCommit();
    try
    {
        Connection connect = connectionBuilder.connect(30);
        String sql = String.format("SELECT * FROM products WHERE id = '%s' LIMIT 1", id);
        Statement statement = connect.createStatement();
        ResultSet resultSet = statement.executeQuery(sql);
        if (!resultSet.next())
        {
            return;
        }
        product = Product.from(resultSet);
    }
    catch (SQLException e)
    {
        throw new IOException(e);
    }
}

public String getId()
{
    return id;
}

public Product getProduct()
{
    return product;
}
}

```

Found hard coded secret key and other parameter used for building a SQL connection also able to find inline SQL query used to fetch product details using product id from table named "products".

LAB 4: Authentication bypass via information disclosure

When logged in with given credential /my-account URL was accessed.

Request

```

1 GET /my-account HTTP/1.1
2 Host: 0a66008c042f8499c0604c0c000f0029.web-security-academy.net
3 Cookie: session=f1ao0Jncf32Arm4Wih0Q1p1gBSFbFax
4 Cache-Control: max-age=0
5 Upgrade-Insecure-Requests: 1
6 User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/102.0.0.0 Safari/537.36
7 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
8 Sec-Fetch-Site: same-origin
9 Sec-Fetch-Mode: navigate
10 Sec-Fetch-User: ?1
11 Sec-Fetch-Dest: document
12 Sec-Ch-UA: "Not A;Brand";v="99", "Chromium";v="102", "Google Chrome";v="102"
13 Sec-Ch-UA-Mobile: ?0
14 Sec-Ch-UA-Platform: "Windows"
15 Referer: https://0a490001038e2d96c0e37cbf00a200c9.web-security-academy.net/login
16 Accept-Encoding: gzip, deflate
17 Accept-Language: en-US,en;q=0.9
18 Connection: close
19
20

```

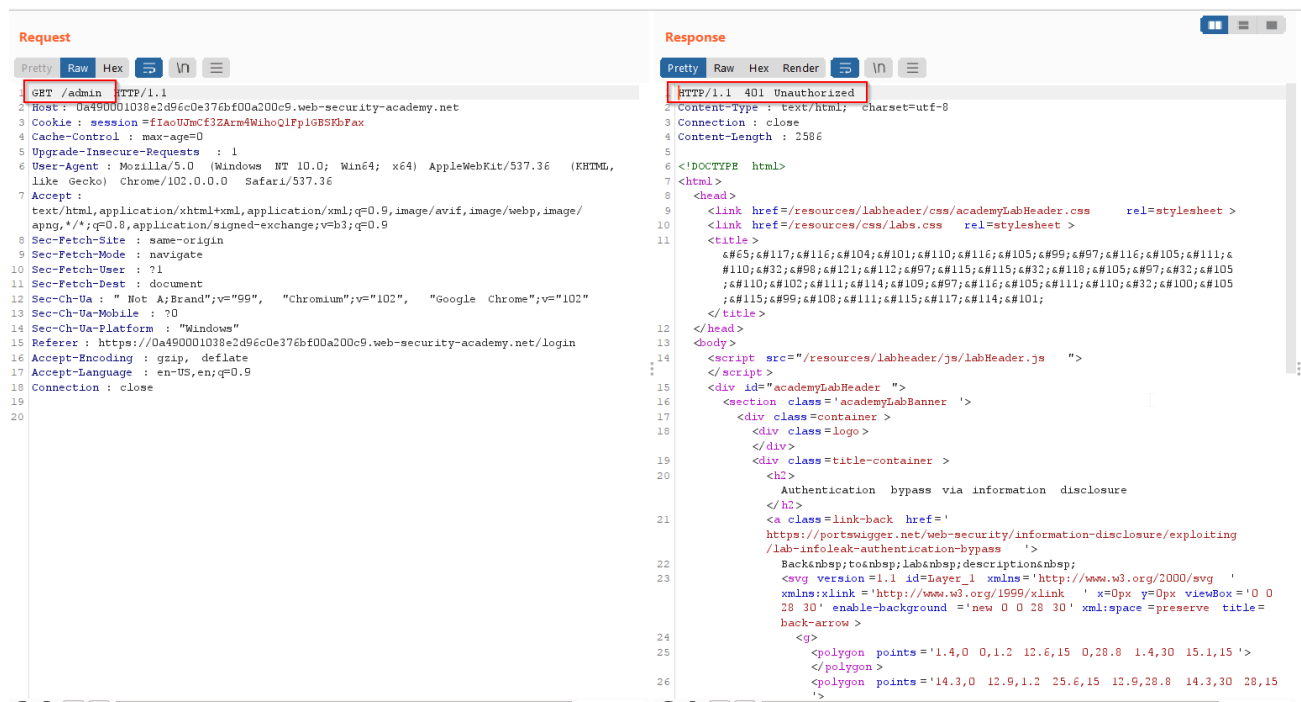
Response

```

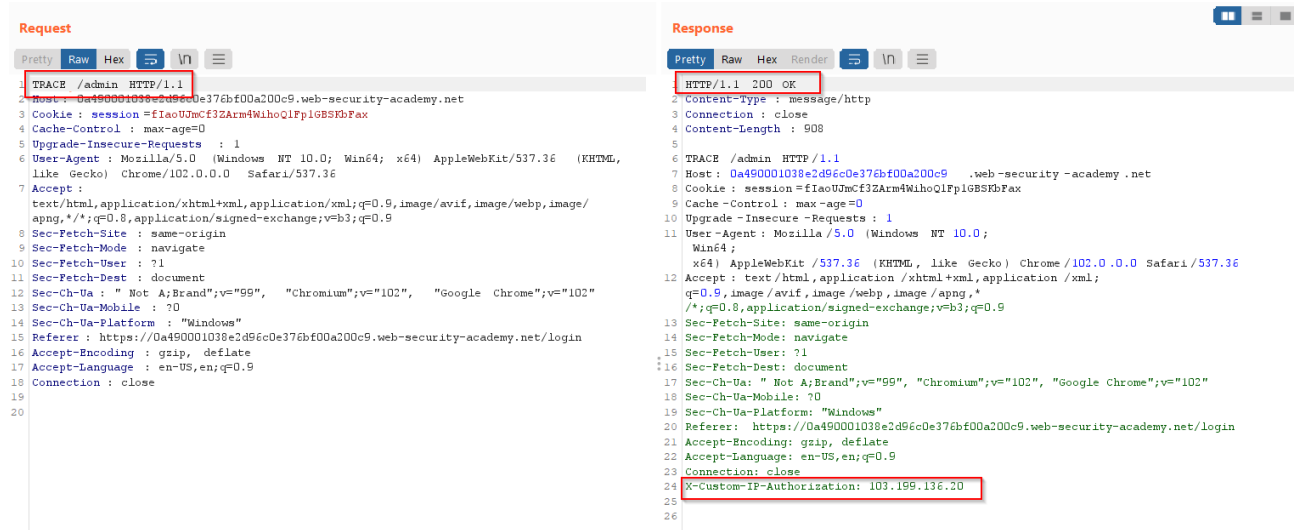
1 HTTP/1.1 200 OK
2 Content-Type: text/html; charset=utf-8
3 Cache-Control: no-cache
4 Connection: close
5 Content-Length: 3239
6
7 <!DOCTYPE html>
8 <html>
9 <head>
10 <link href=/resources/labheader/css/academyLabHeader.css rel=stylesheet >
11 <link href=/resources/css/labs.css rel=stylesheet >
12 <title>
13 <#65;#117;#116;#104;#101;#110;#116;#105;#99;#97;#116;#105;#111;#110;#32;#98;#121;#112;#97;#115;#115;#32;#118;#105;#97;#32;#105;#10;#102;#111;#114;#109;#97;#116;#105;#105;#111;#110;#32;#100;#105;#115;#99;#108;#111;#115;#117;#114;#101;
14 </title>
15 </head>
16 <body>
17 <script src=/resources/labheader/js/labHeader.js >
18 </script>
19 <div id=academyLabHeader >
20 <section class=academyLabBanner >
21 <div class=container >
22 <div class=logo >
23 </div>
24 <div class=title-container >
25 <h2>
26 Authentication bypass via information disclosure
27 </h2>
28 </div>
29 </div>
30 </div>
31 </body>
32 </html>

```


Let's try to access /admin URL. When tried encountered an unauthorized error "401 unauthorized" instead of an error "page not found".



Which confirms the presence of page and there might be some factor which is checking for authorization lets deep dive to find the same using TRACE method.



I found a parameter **X-Custom-IP-Authorization: 103.199.136.20** which shows an internal IP is authorized for same.

So, In Burp proxy I have replaced blank with given internal IP and when tried to access /my-account over the browser found admin panel option was enabled for me.

Let's move ahead and delete user carlos id to complete the LAB.

Burp proxy option to replace any phrase.

The screenshot shows the Burp Suite Proxy Options dialog. The 'Match and Replace' tab is selected. A table lists various match and replace rules. A red box highlights the 'Add' button. Another red box highlights the 'Replace' field in the 'Add match/replace rule' dialog, which contains the text 'X-Custom-IP-Authorization: 127.0.0.1'.

Enabled	Item	Match	Replace	Type	Comment
<input type="checkbox"/>	Request header	^If-None-Match.*\$		Regex	Require non-cached response
<input type="checkbox"/>	Request header	^Referer.*\$		Regex	Hide Referer header
<input type="checkbox"/>	Request header	^Accept-Encoding.*\$		Regex	Require non-compressed respons...
<input type="checkbox"/>	Response header	^Set-Cookie.*\$		Regex	Ignore cookies
<input type="checkbox"/>	Request header	^Host: foo.example.org\$	Host: bar.example.org	Regex	Rewrite Host header
<input type="checkbox"/>	Request header	Origin: foo.example.org		Literal	Add spoofed CORS origin
<input type="checkbox"/>	Response header	^Strict-Transport-Sec...		Regex	Remove HSTS headers
<input type="checkbox"/>	Response header	X-XSS-Protection: 0		Literal	Disable browser XSS protection

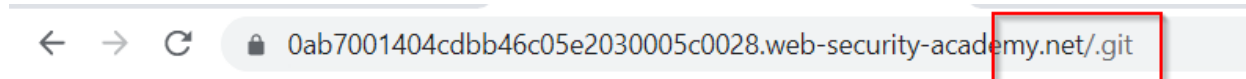
Found admin panel.

The screenshot shows a web application interface. The URL bar displays '0a490001038e2d96c0e376bf00a200c9.web-security-academy.net/my-account'. The page title is 'WebSecurity Academy'. The main heading is 'Authentication bypass via information disclosure'. A green button labeled 'LAB' and a status 'Not solved' are visible. The navigation bar includes links for 'Home', 'Admin panel', 'My account', and 'Log out'. The 'My Account' section shows the message 'Your username is: wiener' and a form with an 'Email' input field and an 'Update email' button.

LAB 5: Information disclosure in version control history

To access git file, we used .git and we found some file listed as below.

Ref: <https://www.tutorialspoint.com/what-is-git-folder-and-why-is-it-hidden>

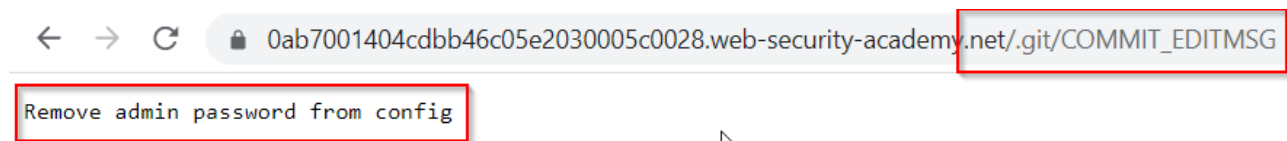


Index of /.git

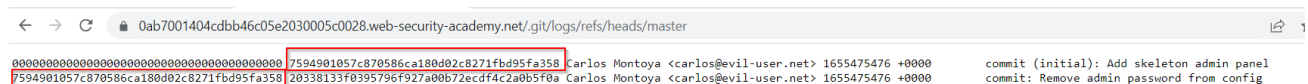
Name	Size
<branches>	
description	73B
<hooks>	
<info>	
<refs>	
HEAD	23B
config	152B
<objects>	
index	225B
COMMIT_EDITMSG	34B
<logs>	

By randomly accessing all file. I found a COMMIT_EDITMSG file.

In which it was written as remove admin password from config file.



So, I tried searching that file eventually found it but it was encoded.



Let's move to kali and try to decode or undo the task.

Command:

wget -r <https://website.com/.git>

```

kali@kali: ~
File Actions Edit View Help
(kali@kali)~$ wget -- https://0ab7001404cddb46c05e2030005c0028.web-security-academy.net/.git
--2022-06-17 10:41:50-- https://0ab7001404cddb46c05e2030005c0028.web-security-academy.net/.git
Resolving 0ab7001404cddb46c05e2030005c0028.web-security-academy.net (0ab7001404cddb46c05e2030005c0028.web-security-academy.net)... 34.246.129.62, 79.125.84.16
Connecting to 0ab7001404cddb46c05e2030005c0028.web-security-academy.net (0ab7001404cddb46c05e2030005c0028.web-security-academy.net)[34.246.129.62]:443... connected.
HTTP request sent, awaiting response... 200 OK
Length: 1201 (1.2K) [text/html]
Saving to: '0ab7001404cddb46c05e2030005c0028.web-security-academy.net/.git'

0ab7001404cddb46c05e2030005c0028.web-security-academy.net/ 100%[=====] 1.17K --.-KB/s
2022-06-17 10:41:51 (32.5 MB/s) - '0ab7001404cddb46c05e2030005c0028.web-security-academy.net/.git' saved [1201/1201]

```

Command:ls

```

kali@kali: ~
File Actions Edit View Help
(kali@kali)~$ ls
0ab7001404cddb46c05e2030005c0028.web-security-academy.net Desktop Documents Downloads Music Pictures Public Templates Videos

(kali@kali)~$ cd 0ab7001404cddb46c05e2030005c0028.web-security-academy.net

(kali@kali)~/0ab7001404cddb46c05e2030005c0028.web-security-academy.net$ ls

(kali@kali)~/0ab7001404cddb46c05e2030005c0028.web-security-academy.net$ ls

(kali@kali)~/0ab7001404cddb46c05e2030005c0028.web-security-academy.net$

```

Tried above steps but I was unable to view files from console so need to download git-cola.

Command: sudo apt-get install git-cola

Has this was not working I have installed a tar file online and then unzip and executed the same from terminal.

```

(root@kali)~/Downloads$ ls
git-cola-4.0.1  git-cola-4.0.1.tar.gz  VijayReddy.ovpn

(root@kali)~/Downloads$ cd git-cola-4.0.1

(root@kali)~/Downloads/git-cola-4.0.1$ ls
bin  CHANGES.rst  cola  contrib  CONTRIBUTING.md  COPYING  COPYRIGHT  docs  extras  Makefile  pynsist.cfg  pyproject.toml  pytest.ini  qtpy  README.md  requirements  setup.cfg  setup.py  share  test  tox.ini

(root@kali)~/Downloads/git-cola-4.0.1$ ./setup.py
./setup.py: 1: import: not found
./setup.py: 5: Syntax error: Bad function name

(root@kali)~/Downloads/git-cola-4.0.1$ cd bin

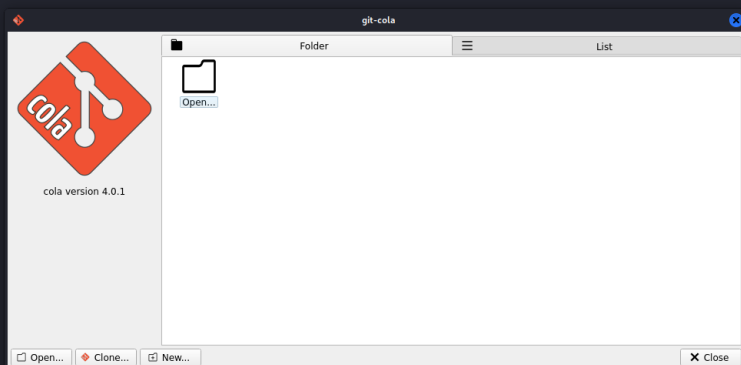
(root@kali)~/Downloads/git-cola-4.0.1/bin$ ls
git-cola  git-cola-sequence-editor  git-dag  README.md

(root@kali)~/Downloads/git-cola-4.0.1/bin$ cd git-cola
cd: not a directory: git-cola

(root@kali)~/Downloads/git-cola-4.0.1/bin$ ls -lah
total 24K
drwxrwxr-x  2 kali kali 4.0K Jun 10 20:25 .
drwxrwxr-x 12 kali kali 4.0K Jun 10 20:25 ..
-rwxrwxr-x  1 kali kali 328 Jun 10 20:25 git-cola
-rwxrwxr-x  1 kali kali 348 Jun 10 20:25 git-cola-sequence-editor
-rwxrwxr-x  1 kali kali 326 Jun 10 20:25 git-dag
-rw-rw-r--  1 kali kali 580 Jun 10 20:25 README.md

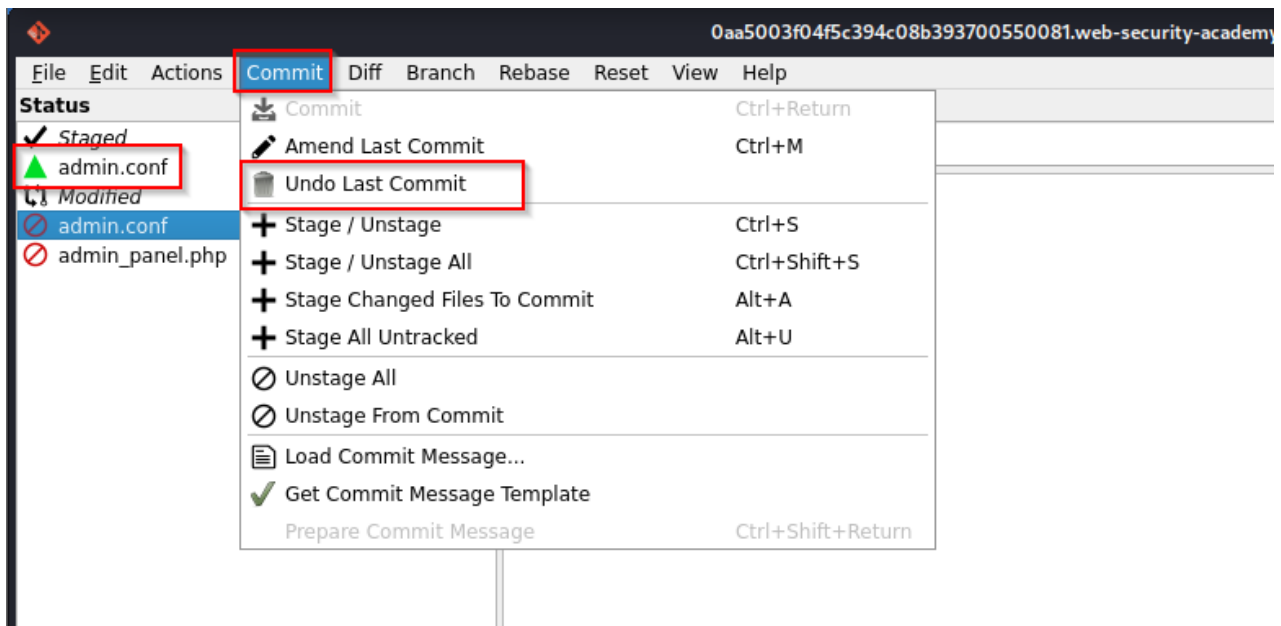
(root@kali)~/Downloads/git-cola-4.0.1/bin$ ./git-cola
StandardPaths: XDG_RUNTIME_DIR not set, defaulting to '/tmp/runtime-root'

```

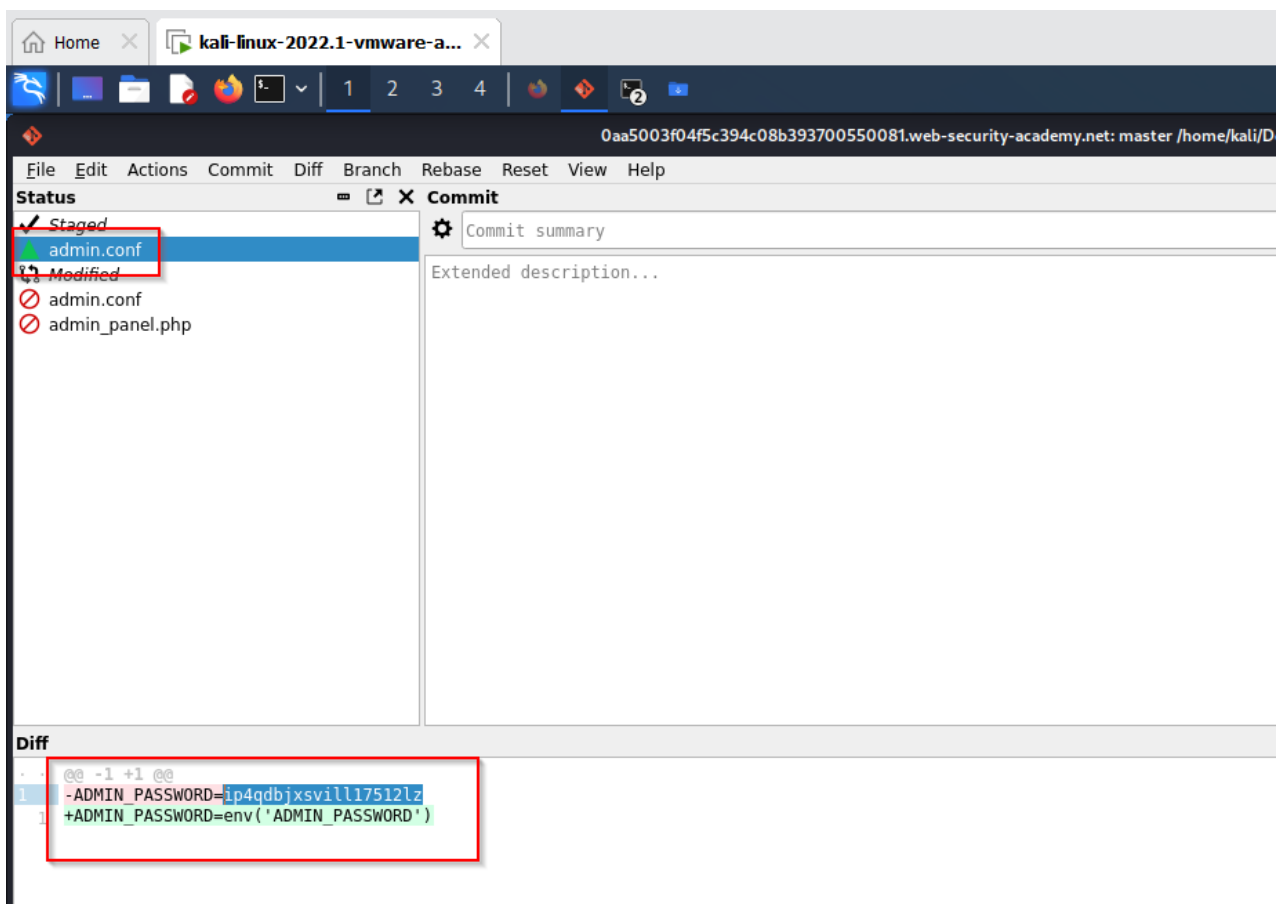


Using commit options let's undo last commit.

Observed green file admin in below attached artifact.



Upon undo found a file "admin.conf" for which we were looking for.



Found the admin credential.

THE END