Forensic analysis

Authors:

Tomás Costa - 89016
João Marques - 89234

Topics:

O que estava implementado a nível de confinamento de aplicações?

Ao analisar os logs de sistema da máquina (neste caso, analisamos na máquina referência) no ficheiro `syslog` não encontramos nenhuma implementação de um sistema de confinamento de aplicações.

Por outro lado, realizamos uma procura diretamente por aplicações de confinamento que esperássemos encontrar. Nesse sentido, encontrámos:

- AppArmor: permite confinar programas a um conjunto limitado de recursos, com perfis carregados diretamente no *kernel*. Tem um modo de **enforcement**, no qual confina verdadeiramente a aplicação, e um modo **complain**, no qual apenas regista se uma aplicação violar o seu perfil.

```
root@vm /m/reference_root# grep -r "AppArmor"
Binary file usr/bin/setpriv matches
Binary file usr/bin/systemd-analyze matches
Binary file usr/bin/dbus-daemon matches
Binary file usr/lib/systemd/systemd matches
usr/share/doc/systemd/NEWS:        * A new unit file option AppArmo
usr/share/doc/systemd/NEWS:          set the AppArmor profile for t
usr/share/doc/systemd/NEWS:        * Support for detecting the IMA 
usr/share/doc/systemd/NEWS:          this condition already support
usr/share/doc/dbus/NEWS:• AppArmor integration has been merged, wit
usr/share/doc/dbus/NEWS:  Ubuntu's GetConnectionAppArmorSecurityCon
usr/share/doc/dbus/NEWS:• AppArmor integration requires libapparmor
usr/share/doc/dbus/NEWS:• Don't duplicate audit subsystem integrati
usr/share/doc/dbus/NEWS:• Log audit events for AppArmor/SELinux pol
usr/share/doc/dbus/NEWS:• On Linux, add support for AppArmor mediat
usr/share/doc/dbus/NEWS:  support), and eavesdropping (a new check,
```

```
Binary file usr/share/locale/cs/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/da/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/de/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/es/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/fi/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/fr/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/ja/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/nl/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/pl/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/pt_BR/LC_MESSAGES/util-linux.mo matche
Binary file usr/share/locale/uk/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/vi/LC_MESSAGES/util-linux.mo matches
Binary file usr/share/locale/zh_CN/LC_MESSAGES/util-linux.mo matche
```

- Docker: permite criar um ambiente virtual no qual a aplicação está confinada a um sistema que apenas contém os serviços que precisa e apenas expõe os necessários. Pela análise do `syslog`, o *driver* foi inicializado, mas não sabemos ainda se foi usado no sistema.

```
root@vm /m/reference_root# grep -r "Docker"
etc/services:docker           2375/tcp                # Docker REST A
etc/services:docker-s         2376/tcp                # Docker REST A
Binary file etc/udev/hwdb.bin matches
Binary file var/lib/rpm/Packages matches
Binary file var/cache/dnf/packages.db matches
Binary file var/cache/dnf/updates.solv matches
Binary file var/cache/dnf/updates-filenames.solvx matches
Binary file var/cache/dnf/updates-presto.solvx matches
Binary file var/cache/dnf/updates-updateinfo.solvx matches
Binary file var/cache/dnf/fedora.solv matches
Binary file var/cache/dnf/fedora-filenames.solvx matches
Binary file usr/lib/systemd/systemd-pull matches
Binary file usr/lib/python3.4/site-packages/sos/plugins/__pycache__
Binary file usr/lib/python3.4/site-packages/sos/plugins/__pycache__
usr/lib/python3.4/site-packages/sos/plugins/docker.py:class Docker(
usr/lib/python3.4/site-packages/sos/plugins/docker.py:    """Docker
usr/lib/python3.4/site-packages/sos/plugins/docker.py:class RedHatD
usr/lib/udev/rules.d/85-nm-unmanaged.rules:# in another net namespa
usr/lib/udev/hwdb.d/20-pci-vendor-model.hwdb: ID_MODEL_FROM_DATABAS
usr/share/doc/curl/CHANGES:  for Docker[4] which uses a special URL
usr/share/doc/systemd/NEWS:        * Docker containers are now dete
usr/share/hwdata/pci.ids:       1179 0002  PCI FastEther LAN on Doc
usr/share/vim/vim74/filetype.vim:" Dockerfile
usr/share/vim/vim74/filetype.vim:au BufNewFile,BufRead Dockerfile
usr/share/vim/vim74/ftplugin/dockerfile.vim:" Language: Dockerfile
usr/share/vim/vim74/syntax/dockerfile.vim:" dockerfile.vim - Syntax
```

- Chroot: isola o processo num diretório que aparenta ser o *root* e impede que o mesmo aceda a ficheiros para os quais não tem permissão.

```
root@vm /m/reference_root# grep -r "Chroot"
etc/ssh/sshd_config:#ChrootDirectory none
Binary file var/cache/dnf/updates-filenames.solvx matches
Binary file usr/sbin/sshd matches
Binary file usr/lib64/librpm.so.7.0.0 matches
Binary file usr/lib64/httpd/modules/mod_unixd.so matches
usr/share/httpd/manual/mod/directives.html:<li><a href="mod_unixd.h
usr/share/httpd/manual/mod/mod_unixd.html:<li><img alt="" src="../i
usr/share/httpd/manual/mod/mod_unixd.html:<div class="directive-sec
usr/share/httpd/manual/mod/mod_unixd.html:<tr><th><a href="directiv
usr/share/httpd/manual/mod/quickreference.html:<tr><td><a href="mod
usr/share/vim/vim74/syntax/aptconf.vim: \ Build-Options Chroot-Dire
usr/share/vim/vim74/syntax/sshdconfig.vim:syn keyword sshdconfigKey
Binary file usr/libexec/mysqld matches
Binary file srv/chroot-mariadb/usr/libexec/mysqld matches
```

Estas informações permitem-nos apenas concluir que estas aplicações estão presentes no sistema, não que estão a ser aplicadas. A ausência de *logs* relativos a estas aplicações no ficheiro `syslog` permite-nos assumir que não foram aplicadas no sistema em produção, apesar de instaladas.

No caso da aplicação **Chroot**, como verificamos a existência de ficheiros como `srv/chroot-mariadb/usr/libexec/mysqld`, assumimos que tenha sido aplicado confinamento à aplicação **MariaDB**/**MySQL**.

Verificámos, também, por análise de vários ficheiros, como explicado posteriormente, que os vários serviços estavam divididos entre diferentes utilizadores do sistema. Apesar de isto ser normal e o comportamento por defeito destas aplicações, produz certo nível de confinamento das aplicações.

Qual a sequência de ações que o atacante tomou?

Para determinar a seguência de ações do atacante, baseamo-nos na diferença entre os diretórios da máquina de referência e da máquina atacada:

```
Files reference_root/etc/httpd/logs/access_log and hacked_root/etc/httpd
Files reference_root/etc/httpd/logs/error_log and hacked_root/etc/httpd/
Files reference_root/etc/httpd/logs/ssl_error_log and hacked_root/etc/ht
Files reference_root/etc/issue and hacked_root/etc/issue differ
Files reference_root/lib/issue and hacked_root/lib/issue differ
Files reference_root/lib/os.release.d/issue-fedora and hacked_root/lib/o
Files reference_root/srv/chroot-mariadb/var/log/mariadb/mariadb.log and
Only in hacked_root/srv/chroot-mariadb/var/tmp: x.txt
Files reference_root/usr/lib/issue and hacked_root/usr/lib/issue differ
Files reference_root/usr/lib/os.release.d/issue-fedora and hacked_root/u
Files reference_root/var/cache/dnf/expired_repos.json and hacked_root/va
Only in reference_root/var/cache/dnf/fedora-fe3d2f0c91e9b65c: metalink.x
Only in reference_root/var/cache/dnf/fedora-fe3d2f0c91e9b65c/repodata: 0
Only in reference_root/var/cache/dnf/fedora-fe3d2f0c91e9b65c/repodata: 8
```

```
Only in reference_root/var/cache/dnf/fedora-fe3d2f0c91e9b65c/repodata: 8
Only in reference_root/var/cache/dnf/fedora-fe3d2f0c91e9b65c/repodata: r
Only in reference_root/var/cache/dnf: fedora-filenames.solvx
Only in reference_root/var/cache/dnf: fedora.solv
Only in hacked_root/var/cache/dnf: metadata_lock.pid
Only in reference_root/var/cache/dnf/updates-e042e478e0621ea6: metalink.
Only in reference_root/var/cache/dnf/updates-e042e478e0621ea6/repodata:
Only in reference_root/var/cache/dnf/updates-e042e478e0621ea6/repodata:
Only in reference_root/var/cache/dnf/updates-e042e478e0621ea6/repodata:
Only in reference_root/var/cache/dnf/updates-e042e478e0621ea6/repodata:
Only in reference_root/var/cache/dnf/updates-e042e478e0621ea6/repodata:
Only in reference_root/var/cache/dnf/updates-e042e478e0621ea6/repodata:
Only in reference_root/var/cache/dnf: updates-filenames.solvx
Only in reference_root/var/cache/dnf: updates-presto.solvx
Only in reference_root/var/cache/dnf: updates.solv
Only in reference_root/var/cache/dnf: updates-updateinfo.solvx
File reference_root/var/lib/gssproxy/default.sock is a socket while file
Files reference_root/var/lib/mlocate/mlocate.db and hacked_root/var/lib/
Only in reference_root/var/lib/NetworkManager: dhclient-4c619efa-fd8b-44
Only in hacked_root/var/lib/NetworkManager: dhclient-654f0ae0-663a-4ed2-
Only in hacked_root/var/lib/NetworkManager: dhclient-7625647c-766a-4ce2-
Only in reference_root/var/lib/NetworkManager: dhclient-a98b76e8-8bef-48
Only in hacked_root/var/lib/NetworkManager: dhclient-ec763d29-5b76-4030-
Files reference_root/var/lib/NetworkManager/timestamps and hacked_root/v
Files reference_root/var/lib/rpm/__db.001 and hacked_root/var/lib/rpm/__
Files reference_root/var/lib/rpm/__db.002 and hacked_root/var/lib/rpm/__
Files reference_root/var/lib/rpm/__db.003 and hacked_root/var/lib/rpm/__
Files reference_root/var/lib/rsyslog/imjournal.state and hacked_root/var
Files reference_root/var/lib/systemd/random-seed and hacked_root/var/lib
Files reference_root/var/log/audit/audit.log and hacked_root/var/log/aud
Files reference_root/var/log/btmp and hacked_root/var/log/btmp differ
Files reference_root/var/log/cron and hacked_root/var/log/cron differ
Files reference_root/var/log/dnf.librepo.log and hacked_root/var/log/dnf
Files reference_root/var/log/dnf.log and hacked_root/var/log/dnf.log dif
Files reference_root/var/log/dnf.rpm.log and hacked_root/var/log/dnf.rpm
Files reference_root/var/log/hawkey.log and hacked_root/var/log/hawkey.l
Files reference_root/var/log/httpd/access_log and hacked_root/var/log/ht
Files reference_root/var/log/httpd/error_log and hacked_root/var/log/htt
Files reference_root/var/log/httpd/ssl_error_log and hacked_root/var/log
Files reference_root/var/log/journal/b74ff8c513354faa8633ee944bc76c73/sy
Files reference_root/var/log/maillog and hacked_root/var/log/maillog dif
Files reference_root/var/log/mariadb/mariadb.log and hacked_root/var/log
Files reference_root/var/log/messages and hacked_root/var/log/messages d
Files reference_root/var/log/secure and hacked_root/var/log/secure diffe
Files reference_root/var/log/syslog and hacked_root/var/log/syslog diffe
Files reference_root/var/log/wtmp and hacked_root/var/log/wtmp differ
Files reference_root/var/www/html/images/road.jpg and hacked_root/var/ww
Only in hacked_root/var/www/html: r.php
```

Análise dos ficheiros diferentes

Por análise do ficheiro `etc/httpd/logs/access_log`, verificamos que o atacante começou por realizar vários ataques por SQL Injection, como por exemplo:

```
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=75
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2.
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2%
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2%
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2%
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2%
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2%
192.168.56.1 - - [14/Dec/2019:01:38:18 +0000] "GET /products.php?type=2%
```

Um dos principais e mais impactantes ataques foi induzido pela seguinte linha:

```
192.168.56.1 - - [14/Dec/2019:01:39:10 +0000] "GET /index.php?a=<?php sy
```

Esta injeção de código *PHP* permite criar uma **backdoor**, pela qual pode ser possível aceder a uma *shell* do sistema. Depois de criado o *backdoor*, o atacante descarregou um *script* e executou-o.

```
192.168.56.1 - - [14/Dec/2019:01:39:10 +0000] "GET /display.php?type=1&l
192.168.56.1 - - [14/Dec/2019:01:39:10 +0000] "GET /display.php?type=1&l
192.168.56.1 - - [14/Dec/2019:01:39:10 +0000] "GET /display.php?type=1&l
192.168.56.1 - - [14/Dec/2019:01:39:11 +0000] "GET /display.php?type=1&l
192.168.56.1 - - [14/Dec/2019:01:39:11 +0000] "GET /display.php?type=1&l
192.168.56.1 - - [14/Dec/2019:01:39:12 +0000] "GET /display.php?type=1&l
```

O atacante acedeu também a várias tabelas da base de dados, incluido tabelas de sistema, e descarregou vários ficheiros de configuração, obtendo informação que lhe permitiu injetar o código referido.

Ao analisar o ficheiro `etc/httpd/logs/error_log` notamos que ocorreram vários erros causados por tentativas de acesso a vários ficheiros por parte do atacante que, sendo feitos pelo utilizador **Apache** a dados do utilizador **MySQL**, resultavam em erros de permissão de acesso.

No ficheiro `srv/chroot-mariadb/usr/libexec/mysqld` não encontramos nada anormal, pelo que assumimos que nada foi atacado nesta aplicação em particular.

Comprovámos também que, no diretório `srv/chroot-mariadb/var/tmp`, estava presente o ficheiro `x.txt`, que, conforme analisado em `etc/httpd/logs/access_log`, foi inserido:

```
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /details.php?prod=1%2
```

Ao analisar o ficheiro `var/log/secure` vimos que o atacante efectuou várias tentativas de login por **ssh** à *backdoor* criada anteriormente:

```
Dec 14 01:39:06 localhost sshd[1905]: Invalid user <?php system($_GET["c
Dec 14 01:39:06 localhost sshd[1905]: input_userauth_request: invalid us
Dec 14 01:39:08 localhost sshd[1905]: pam_unix(sshd:auth): check pass; u
Dec 14 01:39:08 localhost sshd[1905]: pam_unix(sshd:auth): authenticatio
Dec 14 01:39:10 localhost sshd[1905]: Failed password for invalid user <
Dec 14 01:39:10 localhost sshd[1905]: error: maximum authentication atte
Dec 14 01:39:10 localhost sshd[1905]: Disconnecting: Too many authentica
```

Verificamos também que foi criado o ficheiro `/var/www/html/r.php`:

```php
<?php
echo "Road Runner was here";
?>
```

Este ficheiro indica que o user conseguiu acesso a uma terminal, pois devido ao **Chroot** no servidor de base de dados, não era possivel o user escrever ficheiros na pasta var/html/www, no entanto como vimos no exemplo em cima, o user conseguiu escrever nessa pasta.

Não encontramos, no entanto, nenhuma referência à criação deste ficheiro nos *logs*. Assumimos, assim, que o atacante ganhou acesso à máquina através do *backdoor* previamente mencionado e pôde apagar os logs que registaram os seus movimentos.

Que vulnerabilidades foram exploradas e como?

O atacante explorou a vulnerabilidade a **SQL Injection** no *PHP* que tinha sido reportada na auditoria realizada. Esta vulnerabilidade foi explorada de forma a **visualizar** e **descarregar** informação do sistema, assim como **inserir** outros dados e *scripts*.

Por outro lado, aproveitou a falta de **confinamento** das aplicações e explorou esta vulnerabilidade de forma a aceder a todo o sistema a partir da vulnerabilidade persente no servidor *Apache*.

Além disso, aproveitou o facto da máquina do servidor web ter um servidor **ssh** aberto com autenticação por password (vulnerável a ataques por *brute force*) e de o servidor da base de dados estar na mesma máquina.

Quais o atacante tentou explorar mas foram barradas?

Verificamos, principalmente por análise do ficheiro `etc/httpd/logs/error_log`, que o atacante tentou aproveitar as vulnerabilidades no *PHP* para aceder à base de dados e a outros dados de sistema, mas não conseguiu devido ao **confinamento** de cada serviço a um utilizador diferente e à limitação de **permissões** para cada ficheiro.

Também não foi possível inserir ficheiros e dados em determinados diretórios do sistema pela mesma razão.

Que alterações foram realizadas e qual o propósito aparente?

Uma das alterações principais foi realizada na pasta `var/www/html/images` , à imagem road.jpg. Esta nova versão da imagem contêm uma mensagem escondida na imagem, algo denominado esteganografia. Ao dar decode da imagem usando uma ferramenta, conseguimos extrair a mensagem no inicio da imagem que diz:

```
Parabéns!
https://elearning.ua.pt/mod/assign/view.php?id=647250
```

Esta alteração foi possivel porque o utilizador descarregou um ficheiro pelo link. Este script, que não conseguimos transferir, foi a arma do ataque e realizou as mudanças na imagem atraves de estaganografia. O link já não se encontra disponivel e como o script foi colocado no diretório /tmp/, já não está presente no sistema.

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

Sim foram realizadas transferências, o atacante deve ter notado que ao clicarmos no catálogo de carros, o método para descarregar ficheiros era através de um php denominado **downloads.php**, pelo que o user tentou aceder outros ficheiros através desse link e conseguiu com sucesso, visto que nao havia confinamento a nivel do diretório de downloads e ele pode voltar atras nos diretórios.

```
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=Br
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php HTTP/1.
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=Br
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
192.168.56.1 - - [14/Dec/2019:01:39:04 +0000] "GET /download.php?item=..
```

Apesar de as respostas terem sido todas de sucesso (200), nem todos os ficheiros foram transferidos, então analisamos no wireshark os ficheiros que o utilizador recebeu e chegamos a conclusão que ele transferiu todos exceto o ficheiro x.txt, visto que este não existia no sistema. E como conseguimos ver, o atacante conseguiu transferir alguns ficheiros criticos de sistema como **config.php** e **display.php** que lhe permitiram visualizar informação **crítica** do sistema.

Por outro lado, o atacante pode **visualizar** o nome e conteúdo de muitas tabelas da base de dados graças à vulnerabilidade a SQL Injection referida anteriormente.

Porque é que a Firewall externa detetou transferências mas nao detetou as restantes ações?

Ao pesquisarmos nos ficheiros do sistema, conseguimos encontrar dois ficheiros que são de extrema importancia, pois regem as configurações das firewalls.

```
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 ACCEP
iptables -A INPUT -j DROP
iptables -A FORWARD -j DROP
user@vm:/mnt/hacked_root$ sudo cat root/shieldsdown.sh
[sudo] password for user:
iptables -A INPUT -j ACCEPT
iptables -A FORWARD -j ACCEPT
iptables --flush
```

Através da análise destas configurações do iptables, conseguimos perceber que a configuração inicial do `shieldsup.sh` aceita:

- Pacotes do tipo ICMP
- Pacotes vindo da interface lo
- Pacotes novos TCP ao porto 80 E recusa:
- Tudo o que não tiver sido declarado anteriormente, logo adotando uma politica de whitelisting.

Já o `shieldsdown.sh` parece aceitar todo o tipo de pacotes, dai o nome dos escudos estarem em baixo, pois a proteção oferecida foi retirada.

O facto de termos encontrado estes ficheiros na maquina do servidor web leva-nos a crer que, no sistema em produção, a máquina que actua como firewall externa é a mesma que o próprio servidor, o que potenciou o acesso descrito anteriormente sem qualquer controlo.

Ao analisar os pacotes capturados pela firewall, verificamos que, por exemplo, o ficheiro `steg_drop.py`, obtido através do comando *wget* a uma página externa, não foi detetado. Isto pode dever-se ao facto de a firewall não estar a implementar, efetivamente, uma defesa em perímetro, e assim passar por alto muito tráfego da máquina para outros sites. Portanto, outro tráfego criado pelo atacante para outras ações pode facilmente não ter sido detetado.

Suspeitamos que o atacante tenha inicialmente realizado transferências e dai as podermos visualizar no tráfego da firewall externa.

No entanto, tendo criada a **backdoor**, o user ganhou acesso ao sistema por completo, executando comandos numa sessão *ssh* encriptada e, portanto, impossível de analisar pela firewall. Por outro lado, pôde remover as limitações da firewall uma vez dentro da máquina.