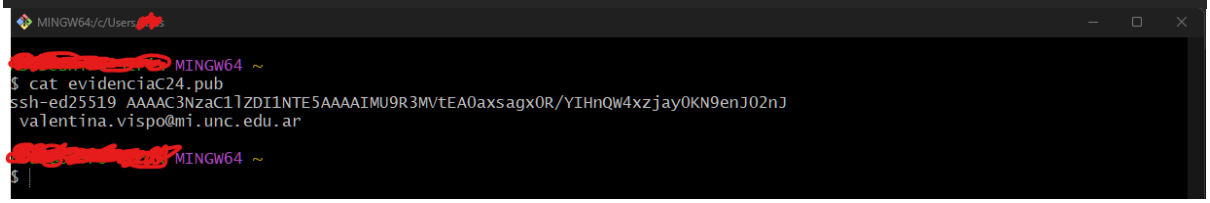
Control de acceso usando ssh y técnicas de protección

1. Generación de una nueva clave para SSH y adición al agente SSH

```
ssh-keygen -t ed25519 -C "your_email@example.com"
```

Para chequear que se ha generado con éxito:

```
cat /home/YOU/.ssh/ALGORITHM
```



¿Cuáles son los algoritmos que puede utilizar ssh ? ¿RSA sigue vigente?

- **RSA**: Sí, RSA sigue siendo ampliamente utilizado en SSH para la autenticación y el intercambio de claves. Sin embargo, en términos de cifrado de datos, RSA no es tan eficiente como otros algoritmos más modernos y seguros.
- **DSA (Digital Signature Algorithm)**: Aunque DSA fue una vez ampliamente utilizado en SSH, su uso ha disminuido debido a problemas de seguridad y se desaconseja su uso en nuevas implementaciones.
- **ECDSA (Elliptic Curve Digital Signature Algorithm)**: ECDSA es una alternativa más eficiente y segura a DSA. Utiliza curvas elípticas en lugar de operaciones con números enteros grandes, lo que lo hace más rápido y más seguro.
- **Ed25519**: Este es un algoritmo de firma digital basado en curvas elípticas, que ofrece una seguridad sólida y un buen rendimiento. Es cada vez más popular en las implementaciones modernas de SSH.
- **AES (Advanced Encryption Standard)**: Además de los algoritmos de firma digital, SSH también utiliza algoritmos de cifrado simétrico como AES para proteger los datos transmitidos.

Alumna: Valentina Solange Vispo - Ciberseguridad

Subir la clave pública ssh al servidor de entrada y al segundo servidor

1. Ingresamos a la máquina “debian server” o “debian public”
2. Verificamos la ip
    a. **ip address**

```
root@publicserver:~# ip address
1: lo: <LOOPBACK,UP,LOWER_UP> mtu 65536 qdisc noqueue state UNKNOWN group default qlen 1000
    link/loopback 00:00:00:00:00:00 brd 00:00:00:00:00:00
    inet 127.0.0.1/8 scope host lo
       valid_lft forever preferred_lft forever
    inet6 ::1/128 scope host
       valid_lft forever preferred_lft forever
2: enp0s3: <BROADCAST,MULTICAST,UP,LOWER_UP> mtu 1500 qdisc pfifo_fast state UP group default qlen 1
000
    link/ether 08:00:27:34:94:9c brd ff:ff:ff:ff:ff:ff
    inet 192.168.68.111/24 brd 192.168.68.255 scope global dynamic enp0s3
       valid_lft 6499sec preferred_lft 6499sec
    inet6 fe80::a00:27ff:fe34:949c/64 scope link
       valid_lft forever preferred_lft forever
root@publicserver:~# _
```

    b.
    c. **sudo ifconfig**

```
osboxes@publicserver:~$ sudo ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST>  mtu 1500
        inet 192.168.68.111  netmask 255.255.255.0  broadcast 192.168.68.255
        inet6 fe80::a00:27ff:fe34:949c  prefixlen 64  scopeid 0x20<link>
        ether 08:00:27:34:94:9c  txqueuelen 1000  (Ethernet)
        RX packets 30  bytes 3262 (3.1 KiB)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 14  bytes 1380 (1.3 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

osboxes@publicserver:~$ _
```

    d.
    e. ping

```
C:\Users\Asus>ping 192.168.68.111

Haciendo ping a 192.168.68.111 con 32 bytes de datos:
Respuesta desde 192.168.68.111: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.68.111: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.68.111: bytes=32 tiempo<1m TTL=64
Respuesta desde 192.168.68.111: bytes=32 tiempo<1m TTL=64

Estadísticas de ping para 192.168.68.111:
    Paquetes: enviados = 4, recibidos = 4, perdidos = 0
    (0% perdidos),
Tiempos aproximados de ida y vuelta en milisegundos:
    Mínimo = 0ms, Máximo = 0ms, Media = 0ms
```

    f.
3. Intentamos ingresar al servidor con ssh:
    a. ssh root@192.168.68.111

b.
```
C:\Users\Asus>ssh root@192.168.68.111
ssh: connect to host 192.168.68.111 port 22: Connection refused
```
c. ssh osboxes@192.168.68.111
d.
```
C:\Users\Asus>ssh osboxes@192.168.68.111
ssh: connect to host 192.168.68.111 port 22: Connection refused
```
e. Tampoco pude utilizar la ip brindada en el tutorial.
f.
```
C:\Users\Asus>ssh osboxes@192.168.4.71
ssh: connect to host 192.168.4.71 port 22: Connection timed out
```
g. ¿Puede ingresar con root ? ¿por qué?
   i. No podemos ingresar como root por que en la configuración no lo permite. En general, por default ssh no permite el ingreso como root.

4. Copiar la clave pública

```
ssh-copy-id osboxes@192.168.4.71
ssh osboxes@192.168.4.71 -v
cat .ssh/authorized_keys
```

5. Impedir el ingreso con contraseña
   a. sudo nano /etc/ssh/sshd_config
6. Reiniciar el servicio ssh
   a. sudo systemctl reload sshd, o
   b. sudo service ssh restart
7. Cree un nuevo usuario
   a. sudo adduser nonpriv
      i. y
8. Intente cambiarse a este nuevo usuario
   a. su nonpriv
   b. ssh nonpriv@...
      i. Desde su computadora real o máquina virtual
   c. ¿Puede ingresar? ¿Por qué? ¿Cómo podemos hacer para poder ingresar?
      i. Supongo que no se puede ingresar.

NOTA: No me permite ingresar como osboxes. A partir de acá voy a realizar un descriptivo de los pasos siguientes.

Fail2ban

```
sudo apt update –y & sudo apt install fail2ban iptables
```

```
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 48  bytes 3626 (3.5 KiB)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

lo: flags=73<UP,LOOPBACK,RUNNING>  mtu 65536
        inet 127.0.0.1  netmask 255.0.0.0
        inet6 ::1  prefixlen 128  scopeid 0x10<host>
        loop  txqueuelen 1000  (Local Loopback)
        RX packets 0  bytes 0 (0.0 B)
        RX errors 0  dropped 0  overruns 0  frame 0
        TX packets 0  bytes 0 (0.0 B)
        TX errors 0  dropped 0 overruns 0  carrier 0  collisions 0

osboxes@publicserver:~$ sudo apt update -y & sudo apt install fail2ban iptables
[1] 642
[sudo] password for osboxes:
Reading package lists... Done
Hit:1 http://deb.debian.org/debian bullseye-updates InRelease
Hit:2 http://security.debian.org/debian-security bullseye-security InRelease
Hit:3 http://deb.debian.org/debian bullseye InRelease
Building dependency tree... Done
Reading state information... Done
The following additional packages will be installed:
  libip6tc2 libnetfilter-conntrack3 libnfnetlink0 python3-pyinotify python3-systemd whois
Suggested packages:
  mailx monit sqlite3 firewalld python-pyinotify-doc
The following NEW packages will be installed:
  fail2ban iptables libip6tc2 libnetfilter-conntrack3 libnfnetlink0 python3-pyinotify
  python3-systemd whois
0 upgraded, 8 newly installed, 0 to remove and 57 not upgraded.
Need to get 1,068 kB of archives.
After this operation, 5,669 kB of additional disk space will be used.
Reading package lists... Done]
Building dependency tree... Done
Reading state information... Done
57 packages can be upgraded. Run 'apt list --upgradable' to see them.
```

Revisamos el estado del servicio fail2ban

```
systemctl status fail2ban.service
```

```
Setting up libip6tc2:amd64 (1.8.7-1) ...
Setting up fail2ban (0.11.2-2) ...
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/f
ail2ban.service.
Setting up python3-pyinotify (0.9.6-1.3) ...
Setting up libnfnetlink0:amd64 (1.0.1-3+b1) ...
Setting up python3-systemd (234-3+b4) ...
Setting up libnetfilter-conntrack3:amd64 (1.0.8-3) ...
Setting up iptables (1.8.7-1) ...
update-alternatives: using /usr/sbin/iptables-legacy to provide /usr/sbin/iptables (iptables) in aut
o mode
update-alternatives: using /usr/sbin/ip6tables-legacy to provide /usr/sbin/ip6tables (ip6tables) in
auto mode
update-alternatives: using /usr/sbin/iptables-nft to provide /usr/sbin/iptables (iptables) in auto m
ode
update-alternatives: using /usr/sbin/ip6tables-nft to provide /usr/sbin/ip6tables (ip6tables) in aut
o mode
update-alternatives: using /usr/sbin/arptables-nft to provide /usr/sbin/arptables (arptables) in aut
o mode
update-alternatives: using /usr/sbin/ebtables-nft to provide /usr/sbin/ebtables (ebtables) in auto m
ode
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+deb11u6) ...
[1]+  Done                    sudo apt update -y
osboxes@publicserver:~$ systemctl status fail2ban.service
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-05-31 19:17:07 EDT; 16s ago
       Docs: man:fail2ban(1)
    Process: 1094 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, status=0/SUCCESS)
   Main PID: 1095 (fail2ban-server)
      Tasks: 5 (limit: 2323)
     Memory: 16.7M
        CPU: 131ms
     CGroup: /system.slice/fail2ban.service
             └─1095 /usr/bin/python3 /usr/bin/fail2ban-server -xf start
osboxes@publicserver:~$
```

Nos movemos a la carpeta de fail2ban para realizar la configuración. Para ello, copiamos la configuración actual. Revisaremos la configuración con nano:

```
cd /etc/fail2ban

sudo cp jail.conf jail.local
sudo nano jail.local
```

```
Created symlink /etc/systemd/system/multi-user.target.wants/fail2ban.service → /lib/systemd/system/f
ail2ban.service.
Setting up python3-pyinotify (0.9.6-1.3) ...
Setting up libnfnetlink0:amd64 (1.0.1-3+b1) ...
Setting up python3-systemd (234-3+b4) ...
Setting up libnetfilter-conntrack3:amd64 (1.0.8-3) ...
Setting up iptables (1.8.7-1) ...
update-alternatives: using /usr/sbin/iptables-legacy to provide /usr/sbin/iptables (iptables) in aut
o mode
update-alternatives: using /usr/sbin/ip6tables-legacy to provide /usr/sbin/ip6tables (ip6tables) in
auto mode
update-alternatives: using /usr/sbin/iptables-nft to provide /usr/sbin/iptables (iptables) in auto m
ode
update-alternatives: using /usr/sbin/ip6tables-nft to provide /usr/sbin/ip6tables (ip6tables) in aut
o mode
update-alternatives: using /usr/sbin/arptables-nft to provide /usr/sbin/arptables (arptables) in aut
o mode
update-alternatives: using /usr/sbin/ebtables-nft to provide /usr/sbin/ebtables (ebtables) in auto m
ode
Processing triggers for man-db (2.9.4-2) ...
Processing triggers for libc-bin (2.31-13+deb11u6) ...
[1]+  Done                    sudo apt update -y
osboxes@publicserver:~$ systemctl status fail2ban.service
● fail2ban.service - Fail2Ban Service
     Loaded: loaded (/lib/systemd/system/fail2ban.service; enabled; vendor preset: enabled)
     Active: active (running) since Fri 2024-05-31 19:17:07 EDT; 16s ago
       Docs: man:fail2ban(1)
    Process: 1094 ExecStartPre=/bin/mkdir -p /run/fail2ban (code=exited, status=0/SUCCESS)
   Main PID: 1095 (fail2ban-server)
      Tasks: 5 (limit: 2323)
     Memory: 16.7M
        CPU: 131ms
     CGroup: /system.slice/fail2ban.service
             └─1095 /usr/bin/python3 /usr/bin/fail2ban-server -xf start
osboxes@publicserver:~$ cd /etc/fail2ban
osboxes@publicserver:/etc/fail2ban$ sudo cp jail.conf jail.local
osboxes@publicserver:/etc/fail2ban$ sudo nano jail.local_
```

```
  GNU nano 5.4                                  jail.local
#_JAILS
#

#
# SSH servers
#

[sshd]

# To use more aggressive sshd modes set filter parameter "mode" in jail.local:
# normal (default), ddos, extra or aggressive (combines all).
# See "tests/files/logs/sshd" or "filter.d/sshd.conf" for usage example and details.
#mode   = normal
port    = ssh
logpath = %(sshd_log)s
backend = %(sshd_backend)s


[dropbear]

port     = ssh
logpath  = %(dropbear_log)s
backend  = %(dropbear_backend)s


[selinux-ssh]

port     = ssh
logpath  = %(auditd_log)s


#
# HTTP servers

^G Help       ^O Write Out  ^W Where Is   ^K Cut        ^T Execute    ^C Location   M-U Undo
^X Exit       ^R Read File  ^\ Replace    ^U Paste      ^J Justify    ^_ Go To Line M-E Redo
```

Reiniciamos el servicio

```
sudo systemctl restart fail2ban
```

Intente ingresar con la contraseña equivocada múltiples veces ¿qué sucede? ¿cuál es la salida del comando “sudo iptables -L”?

NOTA: Dado que no puedo ingresar con el servicio de SSH no puedo realizar este apartado. Lo que supongo que se debe visualizar es que se ha agregado nuestra IP a las bloqueadas, y si intentamos en 10min podemos ingresar nuevamente.

Comando sudo iptables –L

```
osboxes@publicserver:/etc/fail2ban$ sudo iptables -L
Chain INPUT (policy ACCEPT)
target     prot opt source               destination

Chain FORWARD (policy ACCEPT)
target     prot opt source               destination

Chain OUTPUT (policy ACCEPT)
target     prot opt source               destination
osboxes@publicserver:/etc/fail2ban$
```

Y este comando debe ser para desbanear una dirección IP:

```
sudo fail2ban-client set sshd unbanip xxx.xxx.xxx.xxx
```

Reflexión

¿En qué casos conviene listar los usuarios a denegar el acceso? ¿sería más interesante listar solamente a quienes deseo permitir el acceso? ¿En qué casos puede tener sentido?

No veo casos en los que sea más importante “denegar a algunos, pero permitir todo” que “denegar a todos y permitir algunos”. La idea de especificar a quiénes no dar acceso, permite que los intrusos, si no son “conocidos” (estar dentro de esta lista), puedan realizar todas las acciones que deseen. Solo se me ocurre que, si se tiene un sistema muy aislado, y que sea puramente hardware, hay beneficios de esta política, pero como los agentes son personas, no lo sé.

¿Qué otros servicios pueden vigilar fail2ban? ¿Se puede configurar para que envíe alertas cuando se active?

Otros servicios que puede vigilar fail2ban son:

- Denegar el acceso a la IP del atacante,
- Notificar por correo electrónico la IP que originó un problema de intentos fallidos,