Bloqueando intentos de cifrado en shares Samba

Holaquetal.

Andamos algo revolucionados en el trabajo, así que no puedo estar mucho ante las teclas.

Una de las revoluciones con la que nos hemos encontrado es la proliferación de ransomwares, que en el curro llamamos *Cristolockers*, juego de palabras a cuenta de los cipostios que lían y que ahora no viene al caso.

En fin, por Twitter me llegó hoy un enlace con un método para evitar en la medida de lo posible el cifrado de archivos en servidores de ficheros Windows. Como complemento, aquí dejo una posible manera de lograr lo mismo en servidores linux con el rol de servidor de ficheros vía Samba.

El tinglado consiste en unas directivas de auditoría Samba para los shares que el servidor va a presentar, y un filtro a medida para fail2ban que examine el log del sistema y banee al listoferias que haya abierto un mail de correos sobre un paquete certificado que no le han podido entregar.

Al lío.

No es muy complicado, empezaremos añadiendo las directivas de auditoría en el smb.conf:

```
# Recurso de acceso R/W  publico
[publico]
        comment = Directorio compartido
        path = /ruta/al/share/publico
        vfs objects = full_audit
        full_audit: failure = none
        full_audit: success = pwrite write rename
        full_audit: prefix = IP = %I | USER = %u  | MACHINE = %m | VOLUME = %S
        full_audit: facility = local7
        full_audit: priority = NOTICE
        valid users = @winusers
        read only = No
        create mask = 0664
        directory mask = 0775
        force group = winusers
        force directory mode = 0775
        wide links = Yes
```

En negrita, las distintas directivas de auditoría. En concreto y por orden, qué operaciones se van a registrar en caso de que fallen (failure), las que se registrarán en caso de realizarse correctamente (success, pwrite -subida-, write -escritura-, rename -renombrado-), La cadena que se escribirá en el log (prefix: detallamos la ip, el usuario, la máquina y el volumen), el "facility" (local7, /var/log/syslog) y su prioridad. El resto del snip, vemos que son configuraciones bastante normales de un share samba.

Bien. Con estas directivas de auditoría, y una vez reiniciado samba, cada vez que un usuario que tenga mapeada la unidad de red \\servidor\publico escriba, suba ficheros o renombre los mismos, dejará una traza en el log similar a ésta:

```
Apr 21 21:02:50 srv1 smbd[31353]: IP = 192.168.1.35 | USER = fulano | MACHINE =
pote | VOLUME = publico|pwrite|ok|ruta/al/archivo/archivo.extension
```

Lo cual nos da un par de patterns chulos para revisar el log con fail2ban.

Para ello, una vez instalado creamos un fichero de filtro en /etc/fail2ban/filter.d/. Llamémosle samba.conf:

```
[Definition]
failregex = smbd.*\:\ IP\ =\ \ \|.*\.0x0$
            smbd.*\:\ IP\ =\ \ \|.*\.1999$
            smbd.*\:\ IP\ =\ \ \|.*\.*obleep$
            smbd.*\:\ IP\ =\ \ \|.*\.LOL!$
            smbd.*\:\ IP\ =\ \ \|.*\.aaa$
            smbd.*\:\ IP\ =\ \ \|.*\.abc$
            smbd.*\:\ IP\ =\ \ \|.*\.bleep$
            smbd.*\:\ IP\ =\ \ \|.*\.ccc$
[...]
ignoreregex =
```

Esto es un fragmento recortado, pero se deja entender bastante. Básicamente es una compilación de extensiones conocidas de archivos cifrados por los ransom más *populares*. Yo me la he pillado de aquí y adaptado al resto de línea de log para que cuadre con la regex. Ahora mismo, me rondan las 50 extensiones. También te lo puedes currar y añadir los nombres de archivo conocidos con las instrucciones para pagar el rescate, pero ya empezaría a ser un poco demasiado para andar comprobando.

Bueno, hora de añadir nuestro filter al final de /etc/fail2ban/jail.conf:

```
[samba]
filter = samba
enabled = true
```

```
action = iptables-multiport[name=samba, port="135,139,445,137,138",
protocol=tcp]
         mail[name=samba, dest=vfmbofh@mi.correo.no.te.dire]
logpath = /var/log/syslog
maxretry = 1
findtime = 600
bantime = 86400
```

Baneamos 24 horas al primer intento, y enviamos un correo avisando del tema. El findtime, lo dejamos cuadrado con el que haya por default en la instalación.

Iniciamos /etc/init.d/fail2ban start y listos:

```
root@srv1:~# iptables -nL -v
Chain INPUT (policy ACCEPT 3697 packets, 303K bytes)
 pkts bytes target     prot opt in     out     source               destination
    0     0 fail2ban-samba  tcp  --  *      *       0.0.0.0/0
0.0.0.0/0           multiport dports 135,139,445,137,138
    0     0 fail2ban-ssh  tcp  --  *      *       0.0.0.0/0            0.0.0.0/0
multiport dports 22

Chain FORWARD (policy ACCEPT 1047 packets, 157K bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain OUTPUT (policy ACCEPT 3175 packets, 995K bytes)
 pkts bytes target     prot opt in     out     source               destination

Chain fail2ban-samba (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0

Chain fail2ban-ssh (1 references)
 pkts bytes target     prot opt in     out     source               destination
    0     0 RETURN     all  --  *      *       0.0.0.0/0            0.0.0.0/0
```

Y con ésto y un bizcocho…