USBGuard

Per protegir un sistema de "bad USBs" (com són https://maltronics.com/collections/malduinos o https://shop.hak5.org/products/usb-rubber-ducky-deluxe, per exemple) o simplement de dispositius USB que podrien venir carregats amb malware (el qual podria robar dades del nostre sistema o directament espatllar-lo) es poden utilitzar o bé dispositius hardware (com https://github.com/robertfisk/USG/wiki) o bé solucions software (com la configuració de paràmetres Sysfs i/o de regles Udev que permeten deshabilitar certs ports USB, o activar-ne alguns mitjançant l'ús de determinats mètodes d'autenticació -com ara el bloqueig del salvapantalles-, etc). No obstant, per a què el maneig d'aquestes solucions software sigui més còmode i fàcil, es pot fer servir el programari **USBGuard** (https://usbguard.github.io), el qual es basa en Udev per construir llistes negres o blanques (i les seves regles associades) basant-se en els atributs dels dispositius USB reconeguts, però permet fer-ho d'una forma molt més senzilla (com veurem al proper exercici).

> NOTA: També és interessant el programa **Ukip** (https://github.com/google/ukip), que és una eina especialitzada específicament en protegir el sistema d'injeccions de pulsacions de tecles (podeu consultar un breu tutorial d'aquest programa aquí: https://opensource.googleblog.com/2020/03/usb-keystroke-injection-protection.html)
> NOTA: Desgraciadament, no existeix protecció software contra el "USB Killer": https://usbkill.com/products/usbkill-v4

1.-a) Arrenca una màquina virtual VirtualBox (Ubuntu o Fedora) i instal·la-hi el paquet "usbguard". Assegura't, però, que el servei associat (amb *systemctl status usbguard*) no estigui encès, perquè primer haurem de definir les regles que admetran (o denegaran) la connexió dels dispositius USB desitjats.

b) La manera de definir aquestes regles és assegurar-se de tenir endollats en aquest moment tots els dispositius USB que volguem afegir a la "llista blanca" i llavors generar l'arxiu de regles a utilitzar pel servei Usbguard a partir d'aquest estat actual, amb la següent comanda: *usbguard generate-policy | sudo tee /etc/usbguard/rules.conf* (executa-la a la màquina virtual de treball)

Observa el contingut de l'arxiu "rules.conf" creat al pas anterior i concretament, l'estructura de les regles autogenerades, que serà similar a la següent (la qual és prou autoexplicativa; els valors morats indiquen que allà hi ha d'aparéixer un valor explícit -el comodí "*" també està permès!- i els valors verds indiquen que no són tal qual):

> *allow id idVendor:idProduct nomAtribut1 "Valor atribut 1" nomAtribut2 "Valor atribut 2" ...*

> NOTA: Existeixen múltiples atributs que es poden indicar per identificar, més enllà de l'"idVendor" i l'"idProduct", un determinat dispositiu (i/o el bus de connexió que aquest utilitzi): *name*, *serial*, *hash* (valor, computat per USBGuard, únic pel dispositiu en qüestió), *with-interface* (indica la interfície USB que el dispositiu proporciona, amb la forma "xx:xx:xx", on cada parella de nombres hexadecimals representa la "classe", "subclasse" i "protocol" del dispositiu en qüestió, els quals són valors estàndards que es poden trobar aquí: https://www.usb.org/defined-class-codes; -per exemple, "08:*:*" representaria qualsevol dispositiu de la classe "**usb-storage**"), *via-port* (identificador del port USB on està connectat el dispositiu, amb la forma "nº-nº'") o *with-connect-type*, entre altres. Per tenir més informació sobre la sintaxi de les regles i més exemples, consulteu *man usbguard-rules.conf* (o https://usbguard.github.io/documentation/rule-language.html, que és el mateix)
> NOTA: A més de la paraula "allow", al principi d'una regla es poden escriure les paraules *reject* o *block* Les dues deneguen l'ús del dispositiu en qüestió però la primera a més "l'esborra" del sistema (com si no existís); la segona no. No obstant, cal saber que si no apareixen cap d'aquestes dues paraules (és a dir, que només apareixen regles de tipus "allow", com passa per defecte), l'acció per defecte per qualsevol dispositiu no indicat a l'arxiu de regles serà la de "block", així que és rar que apareguin escrites explícitament aquest tipus de regles

c) Ara sí, inicia el servei USBGuard (amb *sudo systemctl start usbguard*) i, si vols, habilita'l també (amb *sudo systemctl enable usbguard*). Tot seguit, executa la comanda *usbguard list-devices* per confirmar l'estat (permés o denegat) de cada dispositiu USB connectat al sistema segons la regla aplicada a cadascun.

> NOTA: Existeix també la comanda *usbguard list-rules*, que simplement llista les regles carregades

d) Clica amb el botó dret sobre la icona dels dispositius USB que apareix a la zona inferior dreta de la finestra de la màquina virtual i apareixerà la llista de dispositius USB connectats a la màquina real que es poden "vincular" a la màquina virtual. Tria qualsevol d'ells (que no interfereixi amb el teu treball habitual a la màquina real) i tot seguit observa (amb la mateixa comanda *usbguard list-devices* com aquest dispositiu haurà estat denegat per l'USBGuard de la màquina virtual (apareixerà la paraula "block" en lloc d'"allow").

e) Suposant que el dispositiu bloquejat a l'apartat anterior sigui el número 8, executa la comanda *usbguard allow-device 8* i tot seguit torna a executar *usbguard list-devices* ¿Quin canvi observes?

NOTA: La comanda *usbguard allow-device* només té efecte mentre la màquina romangui encesa (ja que en reiniciar, el dimoni USBGuard tornarà a aplicar les regles guardades a l'arxiu "rules.conf") Si volguéssim tenir habilitat per sempre el dispositiu "nou" en qüestió, caldria, un cop habilitat amb *usbguard allow-device* , tornar a executar la comanda *usbguard generate-policy* per fer permanent aquest nou estat

NOTA: També existeixen les comandes *usbguard reject-device nº* i *usbguard block-device nº* (les dues deneguen l'ús del dispositiu en qüestió però la primera a més "l'esborra" del sistema, com si no existís -la segona no-)

NOTA: Una alternativa a la comanda usbguard allow-device és la comanda *usbguard append-rule "allow ..."*, la qual permet afegir "en calent" la regla indicada en lloc d'escriure-la manualment dins del fitxer "rules.conf". En tot cas, el que caldria indicar després de la paraula "allow" seria tot el que apareix a la sortida de la comanda *usbguard list-devices* associada al dispositiu en qüestió després del nº i la paraula "block". Igualment, també es pot eliminar l'aplicació en temps real d'una regla qualsevol en qualsevol moment amb la comanda *usbguard remove-rule nº*

NOTA: Per saber més opcions sobre la comanda usbguard, consulteu *man usbguard*

2.-Després de consultar la ja mencionada pàgina del manual *man usbguard-rules.conf*, dedueix quin/s dispositiu/s USB estarien permesos (i, per tant, quins no!) si apareixen les següents regles escrites dins d'un hipotètic fitxer "rules.conf"

a) *allow with-interface equals { 08:*:* }*

NOTA: Noteu que la paraula "equals" força a què l'atribut indicat (en aquest "with-interface") valgui exactament el valor indicat. Aquest comportament provoca que la regla anterior no signifiqui el mateix que la regla següent: *allow with-interface 08:*:** , la qual indica que el dispositiu està autoritzat si s'autoidentifica amb la classe "08:*:*" independentment de si s'autoidentifica o no amb més classes (els dispositius USB es poden autoidentificar amb múltiples classes a la vegades). Aquesta diferència pot ser important en el cas, per exemple, que un llapis d'emmagatzematge USB implementi, a més, un teclat o una interfície de xarxa (quelcom molt sospitós). En aquest sentit, el següent conjunt de regles formaria una política similar (que no igual) a la única regla indicada en aquest apartat, ja que permet els discs USB però rebutjant explícitament els dispositius amb una interfície addicional i sospitosa (com són les indicades amb les classes HID/Teclat, Comunicacions i Wireless):

allow with-interface 08::**
reject with-interface all-of { 08::* 03:00:* }*
reject with-interface all-of { 08::* 03:01:* }*
reject with-interface all-of { 08::* e0:*:* }*
reject with-interface all-of { 08::* 02:*:* }*

NOTA: A més de la paraula "all-of" de l'exemple anterior, també es poden fer servir les paraules "one-of"o "none-of", per exemple així: *allow with-interface one-of { 03:00:* 03:01:* }*

b) *allow id 1050:0011 serial "0001234567" via-port "1-2"*
reject via-port "1-2"

Al final d'una regla (després dels eventuals atributs del dispositiu indicat) es pot indicar opcionalment una condició. Aquesta condició ha de retornar *true* per a què la regla ni tan sols sigui llegida per USBGuard. La sintaxi i tipus de condicions es poden consultar a la mateixa pàgina del manual *man usbguard-rules.conf* però podem destacar-ne per exemple *if localtime(hh:mm[:ss]-hh:mm[:ss])* -per definir un rang de temps; també es pot invertir la condició amb el símbol "!", així: *if !localtime(hh:mm[:ss]-hh:mm[:ss])* -, o *if random(0.8)* -per definir una probabilitat del 80% de retornar *true*-, etc

c) *allow if localtime(09:00-12:00)*
allow id 8086:1234 if random(0.3)

3.-L'arxiu de configuració del dimoni USBGuard és "**/etc/usbguard/usbguard-daemon.conf**". Per saber sobre les possibles directives que hi podem trobar en aquest fitxer, es pot consultar la pàgina del manual *man usbguard-daemon.conf* (o el que és el mateix, https://usbguard.github.io/documentation/configuration.html). Més en concret, esbrina per a què serveixen les següents directives:

> NOTA: Per saber sobre els possibles paràmetres que pot admetre el binari "usbguard-daemon" en sí (normalment invocat via *systemctl*), consulteu en canvi *man usbguard-daemon*
> NOTA: La majoria de directives de configuració poden ser consultades amb la comanda *usbguard get-parameter nomDirectiva* i, el que és més interessant, si l'usuari té els permisos IPC adients, també poden ser modificades en temps real (sense haver de reiniciar el servidor, però tampoc sense persistència quan aquest reinicïi) amb la comanda *usbguard set-parameter nomDirectiva valor*

a) *RuleFile=* i *RuleFolder=*

b) *ImplicitPolicyTarget=*

c) *PresentDevicePolicy=*

d) *IPCAllowedUsers= i IPCAllowedGroups=* (veure nota següent)

> NOTA: "IPC" és el mecanisme intern que utilitza el dimoni USBGuard per poder rebre ordres des de la comanda de terminal *usbguard* (per tal d'afegir o treure regles en temps real, per exemple).
> NOTE: Es poden afegir també usuaris i grups del sistema autoritzats a comunicar-se amb el dimoni USBGuard a través de la pròpia comanda *usbguard*, concretament així: *usbguard add-user nomUsuari [opcions]* i també es poden treure amb *usbguard remove-user nomUsuari [opcions]* (en tot cas, caldrà reiniciar el dimoni per a què aquests canvis tinguin efecte).

e) *AuditBackend=* i *AuditFilePath=* ¿Quina informació pots veure que apareix guardada al fitxer indicat en aquesta darrera directiva?

> NOTA: A Fedora existeix el paquet "usbguard-dbus" (a Ubuntu el seu contingut ja ve integrat dins del paquet general "usbguard") que proporciona la capacitat d'actuar al servidor USBGuard com un servei **D-Bus**. Això permet interactuar dinàmicament amb altres components del sistema. Per exemple, aquest servei és utilitzat per Gnome (concretament, "gnome-settings-daemon") en l'apartat del seu panell de control anomenat "Protecció USB",el qual, si s'activa (mitjançant el botó pertinent o també amb la comanda *gsettings set org.gnome.desktop.privacy usb-protection true -i també gsettings set org.gnome.desktop.privacy usb-protection-level always*) proporciona una interfície gràfica per interaccionar amb el dimoni USBGuard d'una forma molt còmoda i amigable.
> NOTA: A Fedora existeix el paquet "usbguard-tools" (a Ubuntu el seu contingut ja ve integrat dins del paquet general "usbguard") que proporciona la comanda *usbguard-rule-parser*, la qual serveix per comprovar la sintaxis de les regles escrites dins del fitxer que se l'indiqui.