AVBF2_6

Avoid brute force für Linux-Kernel 2.6

Allgemeines

Dieses hervorragende Tool wurde speziell für Asusrouter mit der alternativen Firmware von Oleg/Ily geschrieben, wenn der Router im Access-Point Mode betrieben wird. Prinzipiell kann AVBF auch im Gateway/Router Mode betrieben werden, dazu sind jedoch einige Besonderheiten zu beachten und eventuell einige Routinen zu erweitern.
AVBF verwendet nur Iptables und beschränkt sich auf die INPUT Chain.

Wann setzt man AVBF ein

Wenn sie einen kleinen Homeserver in ihrem Netzwerk betreiben wollen und dazu einen Asus Router im Accesspoint-Mode verwenden, haben sie in der Regel keine Firewall durch ihren vorgeschalteten Router. Je nachdem, welcher Service eingesetzt wird, werden Ports von ihrem Einwahlrouter/Modemrouter an ihren Asus weiter geleitet oder gar eine sog. DMZ auf ihren Asus geschaltet. AVBF schützt in solchen Fällen ihr Netzwerk, bzw. Ihren Asus und die darauf laufenden Services. Die geschützten Ports sind FTP (Port 21), SSH (Port 2222) und Webserver (Port 80), wobei der FTP und SSH Port aber beliebig eingestellt werden kann.

Lizenz

GNU GPL V2 http://www.gnu.de/documents/gpl-2.0.de.html
Bitte beachten sie, dass alle Kopien dieser Software einschliesslich eventueller Änderungen bzw. Anpassungen immer nur vollständig weiter gegeben werden dürfen, sowie ebenfalls die GNU GPL V2 zu Grunde gelegt werden muss. Eigene Anpassungen und Veröffentlichung sind im Sinne der GNU GPL jederzeit möglich, sofern der angepasste Script ebenfalls unter der GNU GPL2 veröffentlicht wird und der Ursprung samt Copyright nicht verändert wird, sowie öffentlich zugänglich ist. Weiters sind alle Files welche unter Punkt “Installation von AVBF” vollständig mit dieser Dokumentation bzw. angepasster Dokumentation zu liefern.

Hardware und Firmware

Prinzipiell ist eine Installation auch auf jeder Linux 2.6.x.x Platform möglich. Natürlich sind dann die Pfade zur Konfiguration usw. anzupassen. Für einen normalen Linux-User sollte das kein Problem darstellen. Dazu beachten sie bitte, dass die Module xt_recent, xt_connlimit und xt_multiport vorhanden sind und die Pfade am Anfang des Scripts zu den Modulen angepasst wurden. Die Module müssen vorhanden sein, die wesentlichen Schutzfunktionen sind auf diese Module angewiesen. Lediglich auf xt_connlimit kann verzichtet werden, dann muss jedoch die Variable conlimit auf “false” gesetzt werden.
AVBF kann für die auf http://code.google.com/p/wl500g/ gelistete Hardware und Fimware bedenkenlos (nur Kernel 2.6.22.19) im AP-Mode ohne wesentliche Änderung eingesetzt werden.
ACHTUNG: Die alte Version AVBF läuft nur auf einem 2.4.x.x Kernel. Die Kennzeichnung der verwendeten Version steht im Header des jeweiligen Scripts. Stellen sie sicher, dass sie eine Version für den 2.6.x.x Kernel brauchen.

Installation von AVBF

Die Installation von AVBF ist sehr einfach. Grundsätzlich sind zwei verschiedene Installationsarten möglich. Der beste Installationsort ist das /opt/sbin Verzeichnis für AVBF und die Datei hosts.deny im /opt/etc Verzeichnis am Router (HDD). Dennoch kann man auch AVBF

im /tmp/local/bin und die Datei hosts.deny im /tmp/etc Verzeichnis installieren.

Folgende Files müssen sie kopieren (mit WinSCP):

Filename	Speicherort am Router	Rechte (chmod)
avbf	/opt/sbin/avbf oder	755
	/tmp/local/bin/avbf	755
deny.hosts	/opt/etc/hosts.deny und	644
	/tmp/etc/hosts.deny	644
S03avbf2_6	/opt/etc/init.d/S03avbf2_6	755
ipts	/opt/sbin/ipts oder	755
	/tmp/local/bin/ipts	755
ashow	/opt/sbin/ashow oder	755
	/tmp/local/bin/ashow	755
recent	/opt/etc/logrotate.d/recent	644
iptables	/opt/etc/logrotate.d/iptables	644

Nachfolgend gelistete Files können sie verwenden, falls sie AVBF noch vor allen anderen Services starten wollen. Diese Files sind die Start/Stop-Scripte im Directory /opt/etc/init.d für rc.unslung Die Erklärungen und weiter gehende Erläuterungen zu diesen Scripts finden sie am Ende dieser Dokumentation. Nur soviel vorab: Die verwendeten Services müssen im Webinterface disabled werden und die Configfiles müssen sich im Verzeichnis /opt/etc befinden.

S01syslogd
S03avbf2_6
S04dropbear
S05mountnfs
S06vsftpd
S07samba
S10cron

Anpassen der Variablen

Die Variablen im Script AVBF sollten sie zumindest auf ihre Richtigkeit prüfen. Prüfen sie auf jeden Fall die Variablen der Modulpfade und die statische IP-Adresse vom Asus. Der Speicherort der Module kann mit der Firmwareversion abweichen. Für weitere Einstellungen lesen sie unbedingt den nächsten Abschnitt “Die Funktionen” um die Wirkung veränderter Werte zu verstehen.
Die Variablen finden sie im Script avbf-Header. Öffnen sie dazu den File mit WinSCP, vi, nano oder irgend einen Linux-kompatiblen Editor.

Die Funktionen

Der Script AVBF schützt sie wenn sie einen FTP, SSH und Webserver betreiben und der Router im AP-Mode betrieben wird.
Zugriffe aus dem eigenen Netzwerk sind per Default stets erlaubt, ebenso ein Zugriff vom Loopback-Device 127.0.0.1
Natürlich kann man ganz einfach den Schutz auch auf das eigene Netzwerk ausdehnen. Dazu sind alle Zeilen mit 192.168.x.x einfach mit dem Kommentarzeichen (#) zu versehen. Diese Zeilen haben dann keine Funktion.

Zusätzlich sind im Original-Script einige Zeilen für Testzweck vorhanden, jedoch sind diese bereits mit dem Kommentarzeichen versehen.

Der Script avbf beinhaltet folgende Funktionen bei Zugriffen (in dieser Reihenfolge):

-Zugriffe aus dem 192.168.x.x Netzwerk werden erlaubt
-Zugriffe vom Loopback (Localhost) 127.0.0.1 werden erlaubt
-Bogus Pakete und ungültige IP-Adressen nach RFC1918 werden verworfen
-Zugriffe aus Netzwerke welche im File hosts.deny eingetragen sind werden geloggt "TCP/UDP/ICMP Banned" und verworfen
-Fragmentierte Pakete werden geloggt "FRAGMENTED" und verworfen
-Alle IP's mit Scanfunktion auf die in der Variablen forwardport definierten Ports werden für einen Tag komplett geblockt und geloggt. Der Logeintrag für diese Ports ist "Portscan" und danach wird auch ein weiterer Zugriff auf Port 80 und die in der Variablen ftp_ssh_ports definierten Ports mit "SCANNER banned" geloggt und verworfen.
-Bereits bestehende Verbindungen sind erlaubt
-Von einer IP können maximal die in der Variable "maxconn" festgelegten **neuen** Verbindungen erstellt werden, werden mehr neue Verbindungen versucht werden alle neuen, weiteren Verbindungen mit der Netzmaske 8 versehen, geloggt "Too much connections" und die über die Anzahl der erlaubten neuen Verbindungen geblockt.
-Pro Minute können maximal die in der Variable maxconn festgelegten neue Verbindungen mit dem TCP Protokoll geöffnet werden, darüber hinaus gehende neue Verbindungen werden geloggt "Limit exceeded" und verworfen.
-Werden die in der Variablen Webhitcount definierten, neuen Verbindungen auf Port 80 von einer IP überschritten, wird diese IP für die in der Variablen block_time_http festgelegten Zeit (Sekunden) geloggt "IP banned from Webserver" und für diese Zeit neue Verbindungen auf Port 80 verworfen.
-Werden die in der Variablen ssh_hitcount definierten, neuen Verbindungen auf den in der Variablen ftp_ssh_ports festgelegten Ports überschritten, wird diese IP für die in der Variablen block_time_ssh festgelegten Zeit (Sekunden) geloggt "BRUTE FORCE IP banned" und verworfen.
-Alle weiteren TCP Zugriffe werden geloggt "End of input chain" und die TCP Verbindung mit der Meldung an die Gegenstelle: reject-with tcp-reset beendet.
-Broadcast Messages aus dem eigenen Netzwerk werden geloggt "LAN-PC/DEVICE-STARTUP" und akzepiert.
-Alle anderen Zugriffe werden verworfen.

Weitere Erklärungen

Die in der Variablen forwardport definierten Ports werden auch Honeyports (Honigports) genannt. Diese Ports sollten möglichst für den Angreifer lukrativ sein, also mindestens Port 22 (SSH) und Port 23 (Telnet) umfassen. Selbstverständlich können beliebig viele Ports angegeben werden, aber alle sollten zumindest bei einem "Quick Scan" z.Bspl. mit Nmap angesprochen werden. Beachten Sie bitte aber auch, dass diese Ports auch auf ihren Asus geleitet werden sollen/müssen, ansonsten sind die Honeyports wirkungslos und können keinen Scan erkennen Honeyports haben keine Funktion und es läuft auch kein Service auf diesen Ports, aber man kann dadurch einen Scan ganz einfach austricksen, indem man diese IP einfach sperrt.

Netzwerkeinbindung

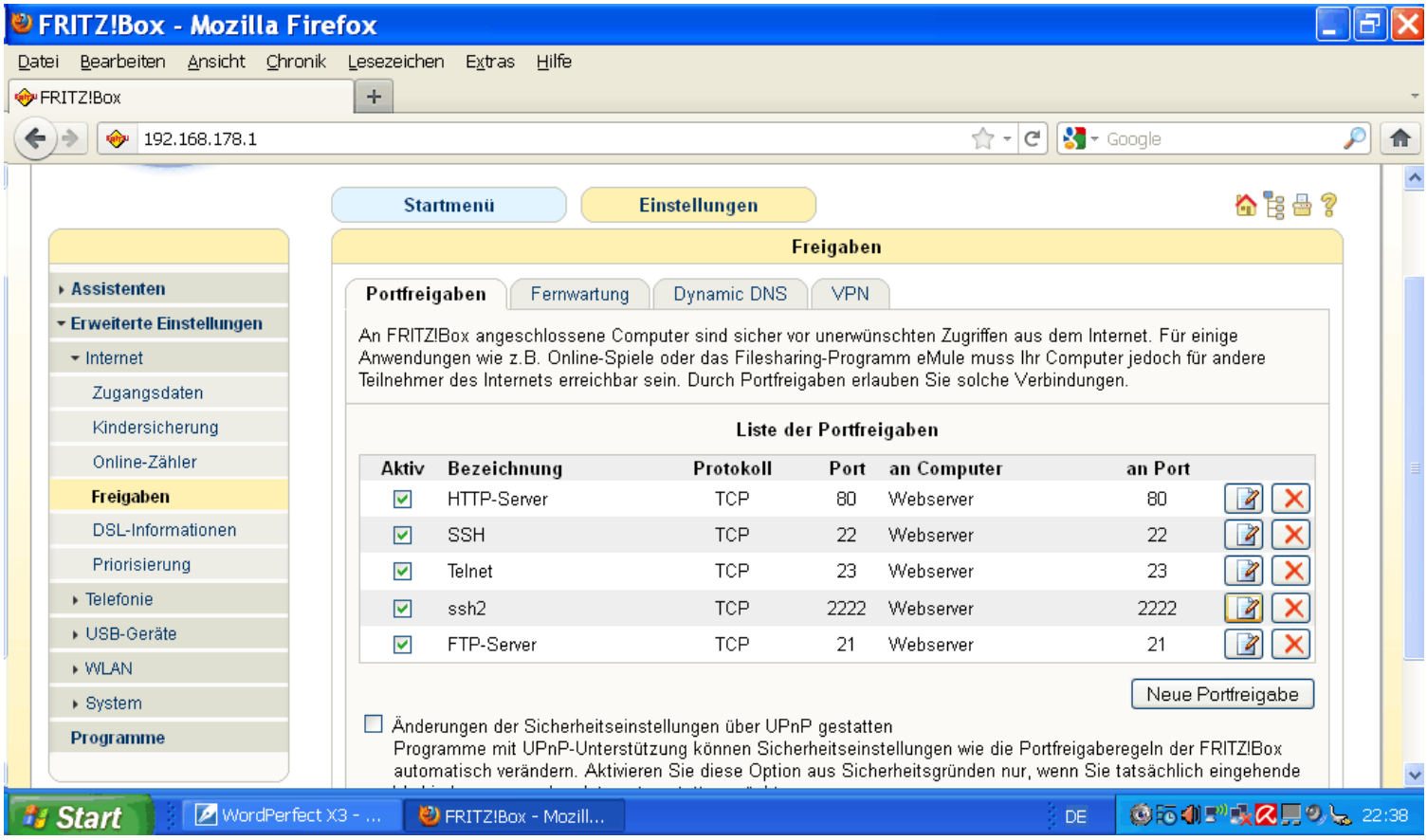
Wird der Asus-Router als Wireless-Access-Point in ihr bestehendes Netzwerk eingebunden, sind einige Vorbedingungen notwendig. Als Beispiel wird hier eine Fritzbox als Einwahlrouter und ein RT-N16 als Accesspoint verwendet. Folgende Ports werden in dieser Konfiguration benötigt und sollen/können an den RT-N16 von der Fritzbox weiter geleitet werden (Defaulteinstellung):
Port 21 FTP (vsftpd)
Port 22 SSH (kein Service, da Dropbear auf Port 2222 läuft)
Port 23 Telnet (kein Service)
Port 80 HTTP (ev. lighttpd oder ein anderer Webserver)
Port 2222 SSH (Dropbear für RT-N16)
Die Ports 22 und 23 werden als Honeyport verwendet. Diese Ports werden z.Bspl. von NMAP gerne als erste Ports gescannt. Will eine externe IP nun auf diese Ports zugreifen, kann es sich nur um einen Portscan handeln. Dabei wird diese IP sofort für einen Tag (86400 Sekunden) gesperrt und alle Pakete verworfen. Nach aussen sieht es nun aus, als ob keine Services laufen, da keine Antwort mehr erfolgt. Script Kiddies und auch Hacker machen nie einen vollen Scan, dieser würde bei vielen Rechnern viel zu lange dauern. Darüber hinaus schützt aber AVBF selbst vor vollen Scans, da sofort jede Anfrage einfach verworfen wird. Das erkennt man im Logeintrag an einem ersten Eintrag “Portscan” und bei weiteren versuchten Zugriffen am darauf folgenden Eintrag “SCANNER banned”

Sollte es mit AVBF in Verbindung mit einem Webserver zu einem oder mehreren unerklärlichen Abbruch kommen, müssen sie die Defaulteinstellungen von AVBF ändern. Das kann meist nur in Verbindung mit Fotogalerien passieren, wenn viele Thumbnails immer eine neue Verbindung erfordern. Dann sollten sie die Variable maxconn erhöhen. In besonders schwierigen Fällen muss man die Variable webhitcount auch erhöhen und eventuell block_time_http etwas vermindern. Versuchen sie aber unbedingt die Grenzen zu probieren. Unbedingt zu beachten ist, dass alle Variablen für “hitcount” die Zahl 20 nicht überschreiten dürfen. Sollte dies unbedingt notwendig werden, gibt es dazu eine Möglichkeit, erfordert aber etwas umfangreicheres googeln. Eine bessere Möglichkeit ist die Variable block_time_http zu vermindern.

Um den RT-N16 auch als Server einsetzen zu können, sollten sie alle Services erst nach AVBF starten. Das geht leichter als man denkt, verwenden sie einfach die mitgelieferten Start/Stop-Files welche sie benötigen und speichern diese nach /opt/etc/init.d. Vergessen sie nicht die benötigten/gewünschten Files mit den Ausführungsrechten zu versehen (chmod 755 /opt/etc/init.d/SxxFilename). Wichtig ist jedoch, dass VSFTPD und Dropbear im Webinterface vom RT-N16 disabled werden. (Ev. Auch Samba disablen)
Ein weiterer wichtiger Aspekt sind die in den Startfiles von VSFTPD und Samba verwendeten Konfigurations-Files. Die Konfigurationsfile werden in /opt/etc erwartet - dadurch sind Änderungen an VSFTPD und Samba viel leichter möglich - ein einfaches “/opt/etc/init.d/S07samba restart” reicht, dass z.Bspl. Änderungen in der smb.conf sofort wirksam werden.
Auch der Cron Start/Stop-Script S10cron sollte getauscht werden, da der originale Script Mängel aufweist.
Falls sie ein NFS an ihren Asus mounten wollen, verwenden sie den Start/Stop-Script S05mountnfs. Ändern sie die im Script angegebenen Mountpoints auf ihre Bedürfnisse und beachten sie, dass der NFS Server im Webinterface enabled ist.

Fritzbox Einstellungen

Natürlich kann man auch andere Modem/Einwahlrouter verwenden, nachstehendes Bild zeigt lediglich die an den RT-N16 (Webserver) weiter geleiteten Ports.



Listing AVBF2_6

```
#!/bin/sh
#avbf2_6 for Linux iptables with Kernel 2.6.x.x
#avoid brute force in AP Mode on an Asus WL500gPV1 or newer (gPV2,500W,RT-N16)
#This script works with iptables and xt_recent module, xt_connlimit and
#xt_multiport
#Script works with oleg/ily's firmware for Asus-Routers with Kernel => 2.6.22.19
#written by newbiefan at wl500g.info (for asus users)
#Please read the entire howto of this script in order to understand which ports
#are protected
#contact: hirau@gmx.at
#Licence of this script: GNU GPL2
#
#
#
#be aware: this is not a firewall!!


#Message for logfile
logger -t AVBF "IPTABLES Script started"
```

```
################################## A D J U S T ####################################
#where are the modules located........
ipt=/usr/sbin/iptables  #path to iptables
mod_ipt1=/lib/modules/2.6.22.19/xt_recent.ko
mod_ipt2=/lib/modules/2.6.22.19/xt_connlimit.ko
mod_ipt3=/lib/modules/2.6.22.19/xt_multiport.ko

forwardport="22,23" #use forwarded ports (all honeyports, comma-separated)
ftp_ssh_ports="21,2222" #ports for ftp and ssh (comma-separated)

conlimit=true #activate connectionlimit to maxconn per ip for port 80, set to true
#or false

maxconn=16 #set the max connections per ip for tcp, consider, MS-IE can have 8
#connections at the same time
#for instance when somebody open a gallery with many thumbnails, increase this
#value
#each picture can open a new connection

ssh_hitcount=3 #how many new connections within block_time_ssh seconds to port
#21&2222 are allowed

webhitcount=12 #how many new connections within block_time_http seconds to your
#webserver are allowed
#when you run into problems with your webserver, set it to max 20, if more
#hitcounts are needed
#you have to set it to more as 20, just google how to set iptables

block_time_ssh=360 #how long to block port 21&2222 when hitcount reached
block_time_http=10 #how long to block port 80 when hitcount reached
################## E N D   O F   A D J U S T M E N T ############################

#use the newest file located at /opt/etc/hosts.deny
if [ -f /opt/etc/hosts.deny ] ; then
   denyhosts=/opt/etc/hosts.deny
 else
   denyhosts=/tmp/etc/hosts.deny
fi
logger -t AVBF "Use file $denyhosts to ban ip's"

#check for module xt_recent
if [ -z "`lsmod | grep xt_recent`" ] ; then
  echo AVBF: "Module xt_recent not loaded, will do it now...."
  logger -t AVBF "Module xt_recent loaded"
  /sbin/insmod $mod_ipt1 ; sleep 1
 else
  logger -t AVBF "Module xt_recent already loaded"
fi
#check for module xt_connlimit when conlimit is set to true
if $conlimit ; then
if [ -z "`lsmod | grep xt_connlimit`" ] ; then
  echo AVBF: "Module xt_connlimit not loaded, will do it now...."
  logger -t AVBF "Module xt_connlimit loaded"
  /sbin/insmod $mod_ipt2 ; sleep 1
 else
  logger -t AVBF "Module xt_connlimit already loaded"
fi
logger -t AVBF "xt_connlimit allows max $maxconn connections from one IP/netmask \
8"
fi
#check for module xt_multiport
if [ -z "`lsmod | grep xt_multiport`" ] ; then
  echo AVBF: "Module xt_multiport not loaded, will do it now...."
  logger -t AVBF "Module xt_multiport loaded"
  /sbin/insmod $mod_ipt3 ; sleep 1
 else
  logger -t AVBF "Module xt_multiport already loaded"
fi
```

```
#emty all chains
$ipt -F
#and delete own chains
$ipt -X
sleep 1

###############we create own chains
#$ipt -N BLOCKIT
$ipt -N Banned
$ipt -N limitlog
$ipt -N webprotect
$ipt -N BRUTE
$ipt -N SCANNER
$ipt -N logdrop
$ipt -N logaccept
$ipt -N UDPCHECK
$ipt -N logonly
$ipt -N logintern

######################### FILTER INPUT CONNECTIONS ##############################

#just make sure, that ips from lan and localhost get never banned and/or
#firewalled (this is very important in case of AP mode)
#RFC 1918 and RFC 1122 (Private Class C Network and Loopback)
#$ipt -A INPUT -i br0 -s 192.168.0.0/16 -m conntrack --ctstate NEW -j logintern
$ipt -A INPUT -s 192.168.0.0/16 -j ACCEPT

#When you need telnet and/or ssh only (for tests as backdoor)
#$ipt -A INPUT -p tcp -m tcp --dport 23 -s 192.168.0.0/16 -d $asusip -j ACCEPT
#$ipt -A INPUT -p tcp -m tcp --dport 22 -s 192.168.0.0/16 -d $asusip -j logaccept

#accept loopback interface
#$ipt -A INPUT -i br0 -s 127.0.0.1/32 -m conntrack --ctstate NEW -j logintern
$ipt -A INPUT -s 127.0.0.1/32 -j ACCEPT

#check for bogus packets and invalid ip addresses (RFC1918)
$ipt -A INPUT -m conntrack --ctstate INVALID -j DROP #drop invalid packets
$ipt -A INPUT -p tcp -m tcp --tcp-flags SYN,FIN SYN,FIN -j DROP
$ipt -A INPUT -p tcp -m tcp --tcp-flags SYN,RST SYN,RST -j DROP
$ipt -A INPUT -s 169.254.0.0/16 -j DROP
$ipt -A INPUT -s 172.16.0.0/12 -j DROP
$ipt -A INPUT -s 192.0.2.0/24 -j DROP
$ipt -A INPUT -s 10.0.0.0/8 -j DROP

#we block everything of hosts.deny list with netmask, syntax of list: "ip/netmask"
#this allows you to block completely bad networks/provider
if [ -e $denyhosts ] ; then
  for ipnm in `cat $denyhosts` ; do
      $ipt -A INPUT -s $ipnm -j Banned
  done
   logger -t AVBF "List with banned ip's loaded"
 else
   logger -t AVBF "No List with banned ip's available!"
fi

#The first rule is BLOCKIT, so everything has to go through this chain
#Use this chain, when you want to block at runtime IP's aso.
#Just add an IP and/or with netmask to blockit at runtime, particulary when adding
an ip by script
#$ipt -A INPUT -j BLOCKIT

################################LIMITS AND GENERAL PROTECTION#####################
#With paranoia you can log everything, except your network and banned network
#$ipt -A INPUT -j LOG --log-prefix "LOG ALL "

#And fragmented packets are logged and dropped
$ipt -A INPUT -f -m limit --limit $(( $maxconn/2 ))/min -j LOG --log-prefix \
"[FRAGMENTED ] " --log-level=info
$ipt -A INPUT -f -j DROP
```

```
#Block IP's from a portscan for a complete day
#Forwarded ports to your Asus - honeyports (forwardport)
$ipt -A INPUT -p tcp -m multiport --destination-port $forwardport -j SCANNER
$ipt -A INPUT -m recent --rcheck --seconds 86400 --name portscan --rsource -j LOG\
--log-prefix "[SCANNER banned] "
$ipt -A INPUT -m recent --update --seconds 86400 --name portscan --rsource -j DROP

#Allow an established connection, independent of port
$ipt -A INPUT -m conntrack --ctstate RELATED,ESTABLISHED -j ACCEPT

#Limit the number of new parallel connections per class C network (netmask 8) to
protect your server
#Keep the allowed max. connections as low as possible
#run some tests.....(set var connlimit to true or false
if $conlimit ; then
$ipt -A INPUT -p tcp --syn -m connlimit --connlimit-above $maxconn --connlimit-
mask 8 -j LOG --log-prefix "[Too much connections] "
$ipt -A INPUT -p tcp --syn -m connlimit --connlimit-above $maxconn --connlimit-
mask 8 -j DROP
fi

# Set proofed limits for http 1.1 clients (1.0 does not use pipelining), this part
#is taken from debianroot.de
$ipt -A INPUT -p tcp --syn -j limitlog
##########################END OF LIMITS AND GENERAL PROTECTION######################

#Protect your webserver
$ipt -A INPUT -p tcp -m tcp --dport 80 -m conntrack --ctstate NEW -j webprotect

#FTP&SSH check for new connections and count at chain BRUTE
$ipt -A INPUT -p tcp -m multiport --destination-port $ftp_ssh_ports -m tcp --tcp-
flags FIN,SYN,RST,ACK SYN -j BRUTE
$ipt -A INPUT -p tcp -m multiport --destination-port $ftp_ssh_ports -m conntrack -
-ctstate NEW -j BRUTE

#When you need external access to your routers webinterface on port 88
#$ipt -A INPUT -p tcp --dport 88 -j logaccept   #This is from wan to your
#webinterface
#$ipt -A INPUT -p tcp --dport 88 -s 192.168.0.0/16 -j logaccept  #Just from your
#network in case you need it

#All other tcp packets are logged and rejected, have a closer look at your
#modem/router, something is wrong!!
$ipt -A INPUT -p tcp -j logonly
$ipt -A INPUT -p tcp -j REJECT --reject-with tcp-reset

#This is for broadcast udp packets from your network. Such a packet is logged
$ipt -A INPUT -p udp -j UDPCHECK

#Consider, when you need udp you have to add a chain/rule here to allow udp
$ipt -A INPUT -j DROP
###################### END OF INPUT CHAIN ##############################

#chain banned, log depending to protocol and drop
$ipt -A Banned -p tcp -m limit --limit 1/min --limit-burst 3 -j LOG --log-prefix
"[TCP Banned] " --log-level=info
$ipt -A Banned -p udp -m limit --limit 1/min --limit-burst 3 -j LOG --log-prefix
"[UDP Banned] " --log-level=info
$ipt -A Banned -p icmp -m limit --limit 1/min --limit-burst 3 -j LOG --log-prefix
"[ICMP Banned] " --log-level=info
$ipt -A Banned -j DROP

#allow half of maxconn up to maxconn connections per minute
$ipt -A limitlog -m limit  --limit $(( $maxconn/2 ))/min --limit-burst $maxconn -j
RETURN
$ipt -A limitlog -m limit --limit 1/min --limit-burst 2 -j LOG --log-prefix
"[Limit exceeded] "
$ipt -A limitlog -j DROP
```

```
#Block nmap and other port scanner
$ipt -A SCANNER -j LOG --log-prefix "[Portscan] "
$ipt -A SCANNER -m recent --set --name portscan --rsource -j DROP

#Do not allow too much new connections to your webserver
$ipt -A webprotect -m recent --name webprotect --rcheck --seconds $block_time_http
--hitcount $webhitcount --rsource -j LOG --log-prefix "[IP banned from Webserver]
"
$ipt -A webprotect -m recent --name webprotect --update --seconds $block_time_http
--hitcount $webhitcount --rsource -j DROP
$ipt -A webprotect -m recent --set --name webprotect --rsource -j logaccept

#... handling of white/blacklists in BRUTE chain
$ipt -A BRUTE -m recent --rcheck --seconds $block_time_ssh --hitcount \
$ssh_hitcount --name BRUTE --rsource -j LOG --log-prefix "[BRUTE FORCE IP banned]
"
$ipt -A BRUTE -m recent --update --seconds $block_time_ssh --hitcount
$ssh_hitcount --name BRUTE --rsource -j DROP
$ipt -A BRUTE -m recent --set --name BRUTE --rsource -j logaccept

#Write to log and accept
$ipt -A logaccept -m conntrack --ctstate NEW -j LOG --log-prefix "[ACCEPT] " --
log-tcp-sequence --log-tcp-options --log-ip-options --log-macdecode
$ipt -A logaccept -j ACCEPT

#Write to log and drop
$ipt -A logdrop -m conntrack --ctstate NEW -j LOG --log-prefix "[DROP] " --log-
tcp-sequence --log-tcp-options --log-ip-options --log-macdecode
$ipt -A logdrop -m conntrack --ctstate INVALID -m limit --limit 5/min -j LOG --
log-prefix "[INVALID] " --log-level=info
$ipt -A logdrop -j DROP

#When you find the message "End of input chain" you forward other ports as 21,22
and 80 (and maybe 88), so something can be wrong
$ipt -A logonly -p tcp -j LOG --log-prefix "[End of Input chain] " --log-tcp-
sequence --log-tcp-options --log-ip-options --log-macdecode
$ipt -A logonly -j RETURN

#Chain udpcheck write to log any startup of a device inside your network
#broadcast messages from your network, RFC 919, 922
$ipt -A UDPCHECK -s 0.0.0.0 -d 255.255.255.255 -m limit --limit 1/h --limit-burst
1 -j LOG --log-prefix "[LAN-PC/DEVICE-STARTUP] "
$ipt -A UDPCHECK -s 0.0.0.0 -d 255.255.255.255 -j ACCEPT
$ipt -A UDPCHECK -j RETURN

#chain logintern, log depending to protocol and ACCEPT (Consider, only new packets
are logged)
$ipt -A logintern -p tcp -m limit --limit 1/min --limit-burst 2 -j LOG --log-
prefix "[TCP intern] " --log-level=info
$ipt -A logintern -p udp -m limit --limit 1/min --limit-burst 2 -j LOG --log-
prefix "[UDP intern] " --log-level=info
$ipt -A logintern -p icmp -m limit --limit 2/min --limit-burst 3 -j LOG --log-
prefix "[ICMP intern] " --log-level=info
$ipt -A logintern -j ACCEPT

#Chain BLOCKIT is just a placeholder for possibility to block bad ip's before any
other rule, but it is not needed!
#see script checklog and add a checklog call to crontab every x minute.
#$ipt -A BLOCKIT -j RETURN
```