6DISS Workshop - Rabat

28 - 30 March '07

Security

Bernard.Tuy@renater.fr
Simon.Muyal@renater.fr
Philippe.Bereski@alcatel-lucent.fr

Securing the servers

1°) Boot on linux, check that the IPv6 connectivity is fine.

2°) From application hands-on, a web server should be running on your host. Add filters to make sure that your server filters incoming IPv6 packets except TCP 80 packets that access the server (help: Appendix A)

3°) Ask other students to check that rules are properly working.

4°) Now configure the filter to allow TCP port 80 to be accessed for IPv6 except from certain groups of the room

5°) Again ask other students to check that rules are properly working.

Securing the network

Laboratory information

Refer to the routing laboratories for getting access parameters.

Access list configuration

1°) Take an IPv4 access-list that you have configured in your network. Build a similar access-list for IPv6 and configure it on the lab router you manage. Don't apply it on an interface to avoid causing access damage to the router (Help: Appendix B).

Appendix A

Mise en œuvre du filtrage

Dans le cadre du filtrage on n'utilisera que la table « Filter » qui concerne trois chaînes par défaut :

- INPUT : contrôle les paquets à destinations des applications.
- FORWARD : filtre les paquets qui passent d'une interface réseau à une autre.
- OUTPUT : contrôle les paquets sortants des applications.

Remarque : il est possible de créer des chaînes supplémentaires si nécessaire.

La commande qui permet de manipuler les tables est ip6tables (voir à la fin du document : principales options de la commande ip6tables). Cette commande peut être utilisée en ligne de commande, mais les tables n'étant pas permanentes il est vivement conseillé de créer un script qui génère les tables et de le lancer automatiquement au démarrage du système (à positionner sous ../init.d). Toutes les commandes ip6tables s'appliquent à la table spécifiée : filter par défaut.

Au lancement de notre script, les actions suivantes sont réalisées :

- Nettoyage pour partir sur des bases claires : Vider les règles pour toutes les tables par la commande **ip6tables -F** et supprimer les chaînes qui ne sont pas par défaut dans la table filter par la commande **ip6table -X**
- Ajout des nouvelles chaînes par la commande **ip6tables –N nom-de-la-chaine suivi des actions**
- Définition des politiques de sécurité par défaut pour chaque chaîne : généralement DROP ou ACCEPT, par la commande **ip6tables –P nom-de-la-chaine action**
- Définition des règles d'usage local sous la forme **Ip6tables –A nom-de-la-chaîne règle**

Où règle défini par exemple le protocole, le port source et/ou destination, l'adresse source et/ou destination ou le réseau source et/ou le réseau destination et l'action à entreprendre. Pour plus de détail sur la syntaxe se référer au manuel d'ip6tables. L'ordre des règles de filtrage est important car lorsqu'une règle correspond les autres ne sont pas évaluées.

Pour lister les règles (en ligne de commande) utiliser la commande **ip6tables –L** qui affiche le contenu des règles en abrégé, pour une forme plus explicite utiliser les options : ip6tables -L -n -v –line-numbers.

Exemple de script

```
#/bin/bash

# ports bas
dw_ports="0:1023"

# ports hauts
```

```
up_ports="1024:65535"

start_fw()
{

# Vidage des règles pour toutes les tables**
ip6tables -F

# permet l'effacement de toutes les chaînes qui ne sont pas par défaut dans
la
# table filter notamment LOG_ACCEPT
ip6tables -X

# Ajout d'une nouvelle chaîne qui a pour fonction de logguer
# dans syslog ce qui est accepté
ip6tables -N LOG_ACCEPT
ip6tables -A LOG_ACCEPT -j LOG -m limit --limit 500/hour --log-level 6 -
log-prefix "[ip6tables-accept]"
ip6tables -A LOG_ACCEPT -j ACCEPT

# Mise en oeuvre des politiques de sécurité positionnées à DROP : par
# défaut tout est refusé
ip6tables -P INPUT DROP
ip6tables -P OUTPUT DROP
ip6tables -P FORWARD DROP

# La machine est saine et les processus locaux peuvent communiquer
# entre eux via l'interface locale :
ip6tables -A INPUT -i lo -j ACCEPT
ip6tables -A OUTPUT -o lo -j ACCEPT

# Nos règles d'usage local
# Accès SSH depuis la zone de confiance et log
ip6tables -A OUTPUT -p tcp --sport ssh -d X:X:X:X::/64 --dport $up_ports !
–syn -j LOG_ACCEPT
ip6tables -A INPUT -p tcp --dport ssh --sport $up_ports  -s X:X:X:X::/64 -j
LOG_ACCEPT

#Autorisation accès DNS udp 53 vers 53
ip6tables -A OUTPUT -p udp --dport domain --sport domain -j ACCEPT
ip6tables -A INPUT -p udp --sport domain --dport domain -j ACCEPT

#Autorisation acces DNS udp 53 et tcp 53 (requetes longues)
ip6tables -A OUTPUT -p udp --dport domain --sport $up_ports -j ACCEPT
ip6tables -A INPUT -p udp --sport domain --dport $up_ports -j ACCEPT
ip6tables -A OUTPUT -p tcp --dport domain --sport $up_ports -j ACCEPT
ip6tables -A INPUT -p tcp --sport domain --dport $up_ports -j ACCEPT

# Autorisation serveur DNS udp 53 et tcp 53 (requêtes longues)
ip6tables -A INPUT -p udp --dport domain --sport $up_ports -j ACCEPT
ip6tables -A OUTPUT -p udp --sport domain --dport $up_ports -j ACCEPT
ip6tables -A INPUT -p tcp --dport domain --sport $up_ports -j ACCEPT
ip6tables -A OUTPUT -p tcp --sport domain --dport $up_ports -j ACCEPT

#Autorisation ftp pour la zone de confiance
ip6tables -A OUTPUT -p tcp --dport 21 --sport $up_ports -d X:X:X:X::/64 -m
state --state NEW,ESTABLISHED -j ACCEPT
```

```
ip6tables -A INPUT -p tcp --sport 21 --dport $up_ports -s X:X:X:X::/64 -m
state --state ESTABLISHED -j ACCEPT

# Accès http et https
ip6tables -A INPUT -p tcp --sport http --dport $up_ports -m state --state
ESTABLISHED,RELATED -j ACCEPT
ip6tables -A OUTPUT -p tcp --dport http --sport $up_ports -j ACCEPT
ip6tables -A INPUT -p tcp --sport https --dport $up_ports -m state --state
ESTABLISHED,RELATED -j ACCEPT
ip6tables -A OUTPUT -p tcp --dport https --sport $up_ports -j ACCEPT

# Autorisations icmpv6
ip6tables -A INPUT -p icmpv6 –icmpv6-type echo request -j ACCEPT
ip6tables -A OUTPUT -p icmpv6 -j ACCEPT

# Log des paquets rejetés dans syslog
ip6tables -A INPUT -j LOG -m limit --limit 500/hour --log-level 6 --log-
prefix "[ip6tables-in-reject]"
ip6tables -A OUTPUT -j LOG  -m limit --limit 500/hour --log-level 6 --log-
prefix "[ip6tables-out-reject]"

}

load_modules()
{

echo -en " Loading modules : "
echo -en "ip_tables, " ; /sbin/modprobe ip_tables
echo -en "ip_conntrack, " ; /sbin/modprobe ip_conntrack
echo -en "ip_conntrack_ftp, " ;/sbin/modprobe ip_conntrack_ftp
echo "ipt_limit" ; /sbin/modprobe ipt_limit

}

stop_fw()
{

# Vidage des règles pour toutes les tables :
ip6tables -F

# permet l’effacement de toutes les chaînes qui ne sont pas par défaut dans
la
# table filter notamment LOG_ACCEPT
ip6tables -X

# On remet la politique par défaut à ACCEPT dans les trois tables par
défaut
ip6tables -P INPUT ACCEPT
ip6tables -P OUTPUT ACCEPT
ip6tables -P FORWARD ACCEPT

}

case "$1" in

start)

load_modules
```

```
start_fw
echo "firewall started"

stop)

stop_fw
echo "firewall stopped"

restart)

stop_fw
echo "firewall stopped"
load_modules
start_fw
echo "firewall restarted"

*)

echo "usage: $0 [start|stop|restart]" >&2

;;

esac
```

Appendix B

Configure Access-lists

```
Router1# configure terminal
Router1(config)# ipv6 access-list v6test
Router1(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:3::1 any
Router1(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:13::3 any
Router1(config-ipv6-acl)# permit ipv6 host 2001:DB8:CAFE:34::3 any
Router1(config-ipv6-acl)# exit
```

Applying Access Lists to your interfaces

In IPv4 the command to apply an access-list to an interface was *ip access-group <access-list_number_or_name> <in|out>*. In IPvt the command is *ipv6 traffic filter*.

```
Router1(config)# interface FastEthernet1
Router1(config-if)# ipv6 traffic-filter v6test in
```

If you notice, you don't peer with any neighbors now. The reason is that the updates are sent via link local addresses. You have to permit these packets to your router.

```
Router1(config)# ipv6 access-list v6test
Router1(config-ipv6-acl)#  permit ipv6 FE80::/16 any
```

Checking your Access-lists

To explicitly deny any IPv6 ICMP configure:

```
Router1(config)# ipv6 access-list v6test
Router1#(config-ipv6-acl)# deny icmp any any
```

To check the access-list counters do:

```
Router1# show ipv6 access-list v6test
IPv6 access list v6test
permit ipv6 host 2001:DB8:CAFE:3::1 any sequence 10
permit ipv6 host 2001:DB8:CAFE:13::3 any (7 matches) sequence 20
permit ipv6 host 2001:DB8:CAFE:34::3 any sequence 30
permit ipv6 FE00::/8 any (10 matches) sequence 40
deny icmp any any (5 matches) sequence 50
```