Nouvelles fonctionnalités des ACL CISCO : les CBACS

■ Gabrielle FELTIN, Gabrielle.Feltin@loria.fr
Loria, Nancy

Evaluation des "Context-Based Access Control" (CBAC), en site bêta pour CISCO, phase II.
La sécurité extérieure de notre réseau repose essentiellement sur des listes d'accès (ACL) implémentées sur notre routeur. Cette solution n'est pas satisfaisante, car un certain nombre d'applications utilise des ports dynamiques et non statiques, et les ACL ne savent pas les traiter.

■ Les filtres dans un routeur : pourquoi ?

Politique de sécurité du réseau

Une politique de sécurité réseau est basée sur le contrôle de l'accès réseau aux divers serveurs et services. Rappelons qu'une politique de sécurité réseau a pour but de :

- contrôler le trafic réseau et son usage,
- identifier les ressources réseau et les menaces,
- définir l'utilisation du réseau et les responsabilités,
- définir un plan d'action lors de violation de la politique de sécurité.

Lors du déploiement d'une politique de sécurité réseau, il est possible de renforcer la sécurité sur plusieurs frontières défensives appelées perimeter networks.

Dans ce cas, le réseau le plus à l'extérieur (outermost perimeter network) est l'aire la moins sécurisée du réseau et donc réservée aux routeurs, firewall et serveurs publics. Les machines situées dans cette zone (donc les plus exposées) ne devront contenir aucune information sensible.

Suite à la mise en place d'une politique de sécurité, les réseaux internes seront considérés comme réseau de confiance (trusted network).

Caractéristiques du firewall (garde-barrière)

Son rôle consistera à contrôler l'ensemble des paquets (entrants ou sortants) envers une politique de sécurité réseau. Le firewall sera la passerelle incontournable pour toutes les communications entre le réseau sécurisé et les autres. Il permettra donc en vertu d'un certain nombre de règles de sécurité définies par l'administrateur d'autoriser ou d'interdire le passage de trames. Il apporte une possibilité de journalisation des événements liés à des problèmes de sécurité ainsi que des fonctions d'inspection, d'audit et d'alerte.

Différentes architectures matérielles et logicielles pour la mise en œuvre des firewall sont apparues. Nous allons les décrire dans l'ordre de leur apparition.

Les filtres de paquets

Ils constituent la première génération de garde barrière. Ces firewalls filtrent (laissent passer ou rejettent) les paquets (entrants ou sortants) sur des critères établis par l'administrateur. Ces règles sont statiques et sont basées sur des informations contenues dans les champs des protocoles jusqu'au niveau transport (adresses IP et numéros de ports sources et destinations). Leur principal avantage réside dans leur mise en œuvre facile et leur rapidité de traitement (matériel car basé sur tests logiques).

Leurs principaux inconvénients sont liés à l'utilisation des ACL que nous étudierons dans le chapitre suivant. De plus ils ne permettent pas une authentification stricte des utilisateurs car celle-ci est basée sur les adresses IP.

Ces firewalls peuvent être incorporés aux routeurs (routeur écran). C'est la solution la plus courante dans notre environnement.

Les proxy de circuit

Ils correspondent à la deuxième génération de firewall. Ils permettent de gérer les connexions TCP entre applications. Ils maintiennent une table des connexions ouvertes. Chaque connexion possède son état, des informations de séquencement, les adresses IP associées ainsi que les interfaces physiques d'entrée et de sortie. Leur fonctionnement est différent de celui des filtres de paquets, les règles sont évaluées uniquement à l'établissement des connexions. Ils permettent la translation des adresses grâce aux informations de connexion (=> possibilité de cacher le réseau interne). Ils sont très rapides pour les mêmes raisons que les filtres de paquets.

Leurs principaux inconvénients sont liés à l'authentification toujours basée sur l'adresse IP, à l'impossibilité de restreindre les parties de protocole, à la faiblesse de leur possibilité d'audit et d'alarme.

Les gardes barrière de niveau applicatif

Ils examinent les données jusqu'au niveau application et par-là même accèdent à l'authentification des utilisateurs et au type de requêtes. Il n'y a dans ce cas plus de communication directe entre le client et le serveur mais passage par un relais entre l'extérieur et des agents internes. Leur fonctionnement s'appuie sur un agent proxy par service offert. Un serveur proxy teste la requête et s'il l'accepte, il l'envoi au client proxy qui contactera le serveur réel. Ils permettent de maintenir secrète la topologie du réseau interne, un filtrage plus fin, des possibilités d'authentification plus fortes ainsi qu'une journalisation assez précise. Malheureusement, ils nécessitent beaucoup de calculs et sont donc plus lents. Ils ne prennent pas en compte UDP, RPC…

Les filtres de paquets dynamiques

Ils permettent de prendre en compte les trafics ICMP et UDP pour des applications de type requête-réponse. Ils associent une connexion virtuelle à un trafic UDP. Lorsqu'une requête « sort » du firewall, il y a mémorisation des informations relatives à l'adresse et au port cible, permettant une confrontation du trafic entrant et une identification de la réponse.

■ Fonctionalités des ACL classiques

Les ACL permettent de mettre en œuvre la politique de filtrage associée à chaque interface d'un routeur. Elles sont appliquées pour le sens spécifié (in et/ou out) lors de leur déclaration sur l'interface considérée. In et Out sont toujours référencés par rapport au routeur (in = ce qui entre, out = ce qui sort). Une ACL est constituée d'un ensemble de filtres exécutés séquentiellement dans l'ordre de leur déclaration. Ces filtres peuvent préciser l'acceptation (permit) ou le rejet (deny) d'un paquet lorsque celui-ci répond au critère spécifié (le parcours des entrées d'une ACL s'arrête lorsqu'une des conditions est remplie). Ces numéros de ports et le protocole employés pour les applications sont définis dans le RFC (1700) ASSIGN NUMBER.

Le principal inconvénient du filtrage de paquets à l'aide des ACL provient du fait que de nombreuses applications utilisent des numéros de port dynamiques (ex ftp ports serveur 20 et 21, ports clients > 1023). Dans ce cas, les gestionnaires de réseau ont deux possibilités. Soit, ils « bloquent » les applications utilisant des numéros de ports imprédictibles (plutôt que de risquer d'exposer le réseau interne). Soit, ils protègent les ports connus et laissaient certaines plages de numéros de ports accessibles depuis l'extérieur, entraînant par la même des failles dans la mise en œuvre de la sécurité.

Conclusion sur les ACL

L'application de la méthode de filtrage basée sur l'autorisation de ce qui est demandé et l'interdiction du reste paraît plus sûre. Néanmoins nous constatons qu'un certain nombre de protocoles utilisent des ports définis de manière dynamique. Ce problème est particulièrement sensible et nous oblige à laisser un grand nombre de ports (tcp et udp) accessibles depuis l'extérieur. Cela ne va donc pas dans le sens de la sécurité attendue.

Pour pallier ce problème, Cisco a imaginé un mécanisme de contrôle basé sur l'analyse du trafic sortant, c'est l'objet du chapitre consacré aux CBAC.

■ Fonctionnalités du package Firewall et des CBAC

Définition des CBAC

Le nouveau IOS de CISCO propose un ensemble de fonction Firewall : CISCO IOS Firewall Feature set. Une des ces fonctions est les "context-based access control" ou CBAC. Les CBAC de Cisco fournissent un nouveau mécanisme de filtrage basé sur l'état des connexions. Elles examinent non seulement les informations des couches réseau et transport mais examinent aussi les informations de la **couche application** (comme ftp) pour apprendre et inspecter l'état des sessions TCP et UDP. Les CBAC maintiennent des informations d'état des connections, pour chaque connexion, dans leurs propres structures de données. Ces informations d'état sont

utilisées pour prendre les décisions sur quels paquets doivent être autorisés ou refusés. Ce mécanisme crée et détruit des entrées temporaires dans les ACL pour changer dynamiquement les critères de filtrage (= décisions intelligentes permit/deny). A la fermeture d'une session, l'entrée des ACL associée est effacée. Les CBAC autorisent un routeur à supporter des protocoles appliquant des créations de canaux multiples résultants d'une négociation dans un canal de contrôle, pour les protocoles qu'il sait gérer.

Fonctionnement des CBAC

Inspection des paquets

Les paquets arrivant sont comparés à l'access-list de l'interface associée. Si le paquet est autorisé à transiter (ACL), il est inspecté. S'il initialise une nouvelle connexion ou ouvre un nouveau port de donnée, l'access-list correspondante est modifiée pour permettre aux paquets relatifs à la nouvelle connexion de passer. Les paquets sont inspectés quand ils entrent ou sortent d'un réseau protégé, pour chaque interface configurée pour l'inspection par les CBAC.

Une entrée de la table d'état est crée si le paquet est celui commençant une nouvelle session TCP ou s'il est le premier paquet UDP contenant une adresse et un port non récent. Le trafic de retour n'est permis que si la table d'état contient des informations indiquant que le paquet appartient à une session valide. Quand la session se termine (TCP) ou finit (Time-out sur UDP) l'entrée de la table d'état d'une session est effacée.

Remarque sur UDP : UDP ne fonctionne pas en mode connecté donc la notion de session n'existe pas. Or CBAC travaille sur des paradigmes de session, il approxime des sessions en examinant les informations des paquets UDP et détermine si le paquet est similaire à d'autres paquets UDP récemment vus. CBAC inspecte les adresses et ports source et destination ainsi que le temps jusqu'au paquet le plus proche.

Sécurité et CBAC : CBAC n'autorise pas beaucoup de trous de sécurité sur les interfaces de sortie, cette restriction contrecarre le scan de ports. Les seules ouvertures qu'un scanner de port peut découvrir sont les services que les clients requièrent pour autoriser des sessions externes sur le réseau protégé. Des scans de port peuvent encore trouver des ports ouverts mais des mécanismes préviennent les attaques les plus connues.

Seuls les protocoles de transport prédéfinis dans le routeur pourront bénéficier de ce type de traitement.

Configuration des CBACS

Pour configurer les CBAC, il est nécessaire de réaliser les différentes tâches suivantes.

Choisir une interface (interne ou externe)

Interne fait référence au réseau sécurisé, externe à l'extérieur. Ces dénominations nous paraissent purement abstraites car le routeur ne connaît pas la topologie du réseau, des ACL pouvant très bien être appliquées sur les interfaces situées coté sécurisé et coté extérieur. Ce qui nous paraît intéressant est plutôt le sens dans lequel un CBAC sera appliqué à une interface. Dans ce cas la règle suivante (in = entrée, out =sortie du routeur) sera appliquée.

Il est possible de configurer les CBAC dans les deux directions mais uniquement dans le cas ou un firewall est entre deux réseaux qui nécessitent une protection mutuelle.

La Figure illustre l'utilisation des CBAC sur l'interface externe. Cela ne permettra pas au trafic de traverser le firewall et de pénétrer le réseau interne s'il ne fait pas partie d'une session initiée depuis le réseau interne. Dans ce cas l'application des CBAC sera réalisée en inspectant le trafic entrant sur l'interface interne ou sortant sur l'interface externe, l'ACL étant positionnée en entrée de l'interface externe.

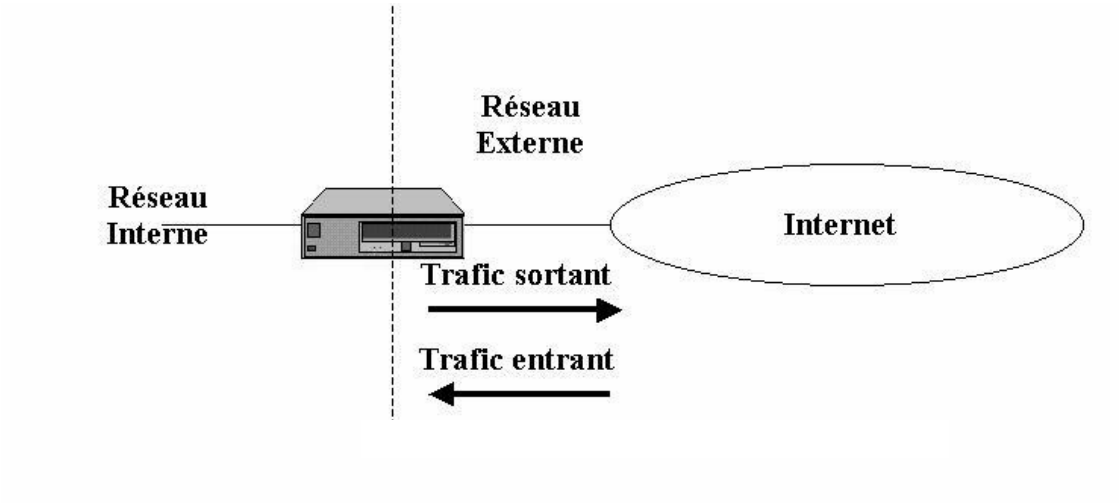
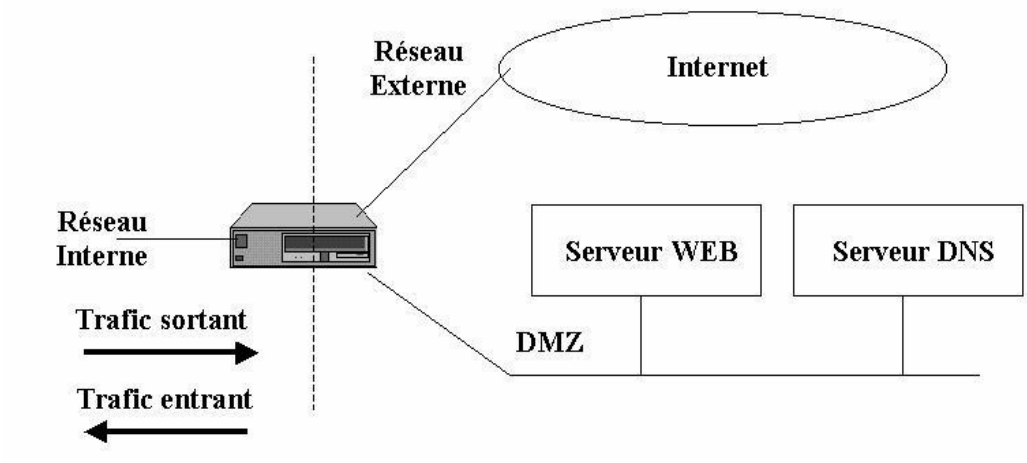


Figure 1. Utilisation des CBAC sur l'interface externe.

La seconde topologie illustre l'utilisation des CBAC sur l'interface interne. Cela permet au trafic d'accéder à la zone démilitarisée (DMZ) tout en interdisant au trafic l'accès au réseau interne s'il ne fait pas partie d'une session initiée depuis le réseau interne. Dans ce cas le CBAC sera appliqué sur le trafic entrant sur l'interface interne.

Figure 2. Utilisation des CBAC sur l'interface interne.



Configurer l'access list de l'interface

Les deux règles suivantes doivent être respectées :

- Permettre au trafic utilisant les CBAC de traverser le firewall et ceci sur toutes les interfaces (si le trafic n'est pas permis, il ne sera pas inspecté par CBAC).
- Utiliser les access list étendues pour interdire le trafic de retour lié à l'utilisation des CBAC.

Configurer les time-outs et les seuils

CBAC utilise des time-outs et des seuils pour déterminer le temps pendant lequel il gère les informations relatives aux sessions et pour déterminer quand une session disparaît.

Ces time-outs et seuils s'appliquent globalement à toutes les sessions. Nous pouvons utiliser les valeurs par défaut ou les modifier.

Remarque : CBAC dispose d'un certain nombre de seuils relatifs aux sessions en cours. Ils lui permettent de contrôler le nombre total de sessions ouvertes ainsi que celles nouvellement établies sur une certaine durée (une minute). Cela permet le contrôle et la riposte à des attaques de type SYN flooding (refus de service).

CBAC gère des compteurs de demi-sessions. Pour TCP, cela signifie qu'une session n'a pas atteint l'état établi. L'établissement de la connexion en 3 phases (three-way handshake) n'ayant pas été réalisé.

Cette stratégie s'applique également à UDP pour lequel une demi-session signifie que le routeur n'a pas détecté de trafic de retour.

Définir les règles d'inspection

Elles spécifient quel trafic IP sera inspecté (niveau application) par les CBAC. Il est nécessaire de configurer l'inspection au niveau de la couche application et éventuellement au niveau de TCP ou d'UDP (cf. H.323).

```
ip inspect name nom_cbac protocol
```

Appliquer les règles d'inspection à l'interface

```
ip inspect nom_cbac in/out
```

Un CBAC est applicable sur une ou plusieurs interfaces et, sur une interface, il peut inspecter le trafic entrant (in) et/ou le trafic sortant(out).

Protocoles supportés par les CBACS

TCP/UDP Applications	Telnet, HTTP, TFTP, SNMP
File Transfer Protocol (FTP)	Both active and passive modes
Multimedia Applications	Inspects control channel stream to ensure video or audio channels are open during each session. Supports H.323 applications, CU-SeeMe, RealAudio, Streamworks, and VDOLive
Electronic Mail Services (SMTP protocol inspection)	Detects invalid SMTP commands. Eliminates requirement for external mail relay in "demilitarized zones"
Remote Procedure Call (RPC) Services	Inspects port mapper requests to open channels as needed to support RPC traffic
Berkeley Standard Distribution (BSD)-Rcmds	Inspects server reply to open any secondary channels
Oracle Database Application Support	Inspects redirect messages from Oracle listener processes. Opens port for clients to connect to servers
H.323-Based Video-conference Applications	Inspects Q931 and H.245 control messages to open additional UDP channels for video and audio data

■ Plate-forme de tests

L'évaluation a été réalisée avec les machines suivantes :

- le routeur (cbac-gw) prêté par CISCO un CISCO 7204 (NPE200) avec 57344K/8192K bytes of memory, avec une carte mère avec une interface à 100Mb/s, une carte interface 100Mb/s

```
IOS (tm) 7200 Software (C7200-IO2-Mz), Experimental Version 12.0
ROM: System Bootstrap, Version 11.1(13)CA, EARLY DEPLOYMENT RELEASE SOFTWARE
(fc1)
BOOTFLASH: 7200 Software (C7200-BOOT-M), Version 12.0(1), RELEASE SOFTWARE
(fc1)
```

- deux stations Sun ULTRA 5 à 128MB de mémoire, avec une carte 100Mb/s
  - une machine sur le réseau interne avec une route sur le réseau à protéger,
  - une machine externe.
- Plate-forme de test pour les CBACS
  Pour tester les possibilités de ce routeur en terme de routage, nous avons du le relier à un commutateur Catalyst disposant de deux interfaces à 100Mbits/s. Cela nous permet donc de créer un Vlan sur lequel divers autres appareils sont connectés. Le schéma suivant illustre la topologie de notre plate-forme de tests.

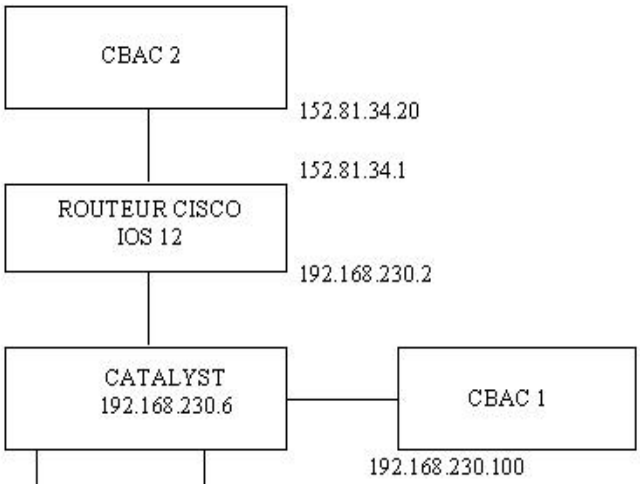


Figure 3. Topologie de la plate-forme de tests.

Test FTP

Nous avons commencé d'étudier le fonctionnement des CBAC en abordant le protocole FTP. Celui-ci répondant parfaitement au besoin de notre étude car il utilise des ports dynamiques ouverts par le serveur (FTP actif). Dans le protocole FTP (serveur actif) nous constatons que le serveur initie une connexion FTP vers le client. Ce numéro de port est défini par le client de manière dynamique (>1023) puis transmis au serveur qui initie la connexion.

TEST FTP sens interne-externe

Le client FTP se trouve à l'intérieur, le serveur dans le réseau externe. Avec des ACL, on doit ouvrit les ports supérieurs à 1023 en tcp pour le port client. Avec les CBACS, on n'a plus l'ouverture de cette plage. La Figure nous indique les contrôles associés aux interfaces. Un inspect sur FTP est associé à l'entrée interne du routeur.

Voici les modifications apportées à la configuration initiale.

```
! CBAC
ip inspect name monsite ftp
!
interface FastEthernet0/0
ip address 192.168.230.2 255.255.255.0
ip inspect monsite in
no ip directed-broadcast
!
interface FastEthernet2/0
ip address 152.81.34.1 255.255.255.0
ip access-group 102 in
!
no access-list 102
access-list 102 deny ip any any
```

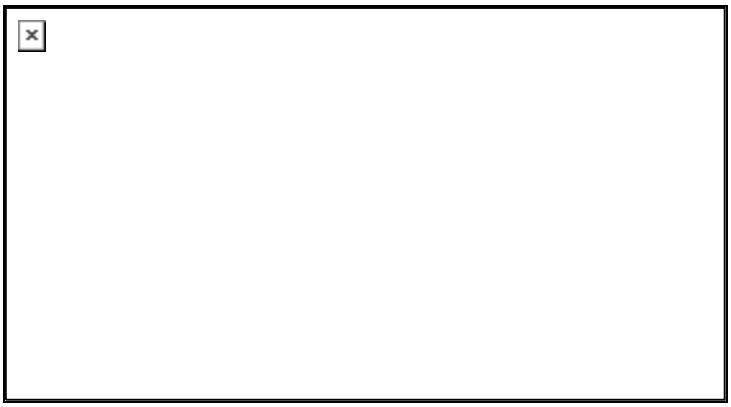


Figure 4. Contrôles associés aux interfaces.

- **Conclusion**

Le comportement est bien celui attendu, ftp sur CBAC2 est autorisé depuis CBAC1, ftp sur CBAC1 n'est pas autorisé depuis CBAC2.

- **Les traces**

Il est possible de visualiser l'ensemble des objets créés et détruits. Cela s'effectue en modifiant la configuration du routeur à l'aide des commandes de debug ou en capturant les logs sur un serveur.

```
terminal monitor
configure terminal
logging monitor debugging
```

Les traces nous permettent de suivre une session FTP accompagnée des traces en mode debug :

********** Traces de la session ftp **********

```
ftp cbac2
Connected to cbac2.
220 cbac2 FTP server (SunOS 5.7) ready.
Name (cbac2:root): kolb
331 Password required for kolb.
Password:
230 User kolb logged in.
ftp
********** Traces sur le routeur **********
2d03h: CBAC Pak 611338F8 sis 60EBEC54 initiator_addr
(192.168.230.100:38007) reponder_addr
(152.81.34.20:21)
initiator_alt_addr (192.168.230.100:38007)
responder_alt_addr (152.81.34.20:21)
2d03h: CBAC OBJ_CREATE: create sis 60EBEC54
2d03h: CBAC OBJ_CREATE: create acl wrapper 611993E8 -- acl item 60EBE4E4
2d03h: CBAC Src 152.81.34.20 Port [21:21]
2d03h: CBAC Dst 192.168.230.100 Port [38007:38007]
2d03h: CBAC OBJ_CREATE: create host entry 60EBEDC8 addr 152.81.34.20 bucket 255
2d03h: CBAC OBJ_DELETE: delete host entry 60EBEDC8 addr 152.81.34.20
********** Traces de la session ftp **********
ftp get toto
```

```
200 PORT command successful.
150 ASCII data connection for toto (192.168.230.100,38008) (0 bytes).
226 ASCII Transfer complete.
************* Traces sur le routeur **************
2d03h: CBAC OBJ_CREATE: create pre-gen sis 611B162C
2d03h: CBAC OBJ_CREATE: create acl wrapper 60EBEDC8 -- acl item 611CB75C
2d03h: CBAC Src 152.81.34.20 Port [1:65535]
2d03h: CBAC Dst 192.168.230.100 Port [38008:38008]
2d03h: CBAC Pre-gen sis 611B162C created: 152.81.34.20[1:65535]
192.168.230.100[38008:38008]
2d03h: CBAC sis 611D5B6C initiator_addr (152.81.34.20:20) responder_addr
192.168.230.100:38008)
initiator_alt_addr (152.81.34.20:20) responder_alt_addr (192.168.230.100:38008)
2d03h: CBAC OBJ_DELETE: delete pre-gen sis 611B162C
2d03h: CBAC OBJ_CREATE: create sis 611D5B6C
2d03h: CBAC OBJ_CREATE: create host entry 611D5CE0 addr 192.168.230.100 bucket
198
2d03h: CBAC OBJ_DELETE: delete sis 611D5B6C
2d03h: CBAC OBJ_DELETE: delete create acl wrapper 60EBEDC8 -- acl item 611CB75C
2d03h: CBAC OBJ_DELETE: delete host entry 611D5CE0 addr 192.168.230.100
************* Traces de la session ftp **************
ftp put toto
200 PORT command successful.
150 ASCII data connection for toto (192.168.230.100,38009).
226 Transfer complete.
************* Traces sur le routeur **************
2d03h: CBAC OBJ_CREATE: create pre-gen sis 611B162C
2d03h: CBAC OBJ_CREATE: create acl wrapper 60EBEDC8 -- acl item 611CB75C
2d03h: CBAC Src 152.81.34.20 Port [1:65535]
2d03h: CBAC Dst 192.168.230.100 Port [38009:38009]
2d03h: CBAC Pre-gen sis 611B162C created: 152.81.34.20[1:65535]
192.168.230.100[38009:38009]
2d03h: CBAC sis 611D5B6C initiator_addr (152.81.34.20:20) responder_addr
(192.168.230.100:38009)
initiator_alt_addr (152.81.34.20:20) responder_alt_addr (192.168.230.100:38009)
2d03h: CBAC OBJ_DELETE: delete pre-gen sis 611B162C
2d03h: CBAC OBJ_CREATE: create sis 611D5B6C
2d03h: CBAC OBJ_CREATE: create host entry 611D5CE0 addr 192.168.230.100 bucket
198
2d03h: CBAC OBJ_DELETE: delete sis 611D5B6C
2d03h: CBAC OBJ_DELETE: delete create acl wrapper 60EBEDC8 -- acl item 611CB75C
2d03h: CBAC OBJ_DELETE: delete host entry 611D5CE0 addr 192.168.230.100
************* Traces de la session ftp **************
ftp quit
221 Goodbye.
************* Traces sur le routeur **************
2d03h: CBAC OBJ_DELETE: delete sis 60EBEC54
2d03h: %FW-6-SESS_AUDIT_TRAIL: ftp session initiator (192.168.230.100:38007) sent
115 bytes -- responder (152.81.34.20:21) sent
```

```
360 bytes
2d03h: CBAC OBJ_DELETE: delete create acl wrapper 611993E8 -- acl item 60EBE4E4
```

La création de l'access-list dynamique, mise en-tête de l'access-list peut aussi être visualisée :

```
sh ip access-list 102
Extended IP access list 102
permit tcp host 152.81.34.20 eq ftp host 192.168.230.100 eq 38009 (6 matches)
```

- **Conclusion**

La configuration ne pose aucun problème, et la solution paraît valide(des essais d'intrusion ont été réalisés sans succès). Les CBAC permettent d'enlever la plage ouverte pour les ports dynamiques avec des ACL.

Test R-Command

Avec des ACL classiques, les ports clients 1023, 1022,1021..., devaient être ouverts pour les applications rexec, rlogin, rsh, rdist. Voici les tests faits avec les commandes rlogin et rsh, pour accéder à l'extérieur du réseau.

La commande rlogin

Voici la configuration pour ne laisser sortir de l'intérieur le rlogin et interdit tout autre trafic.

```
ip inspect name monsite rcmd
!
! Interdiction tout trafic entrant
access-list 102 deny ip any any
!
! Interdiction tout trafic sortant sauf rlogin en complement inspect
no access-list 103
access-list 103 permit tcp any any eq login
access-list 103 deny ip any any
```

La commande rsh

La configuration suivante permet de laisser sortir le rsh et interdit tout autre trafic :

```
ip inspect name monsite rcmd
!
! Interdiction tout trafic entrant
no access-list 102
access-list 102 deny ip any any
!
! Interdiction tout trafic sortant sauf rsh en complement inspect
no access-list 103
access-list 103 permit tcp any any eq cmd
access-list 103 deny ip any any
```

Conclusion

Les R-commandes ne posent aucun problème pour le traitement par les CBACS et permettent également de supprimer des plages dans les ACL.

Test TCP/X

Ce test permet de voir le comportement des CBAC sur le protocole TCP, sur de l'allocation de ports dynamiques, sans que le protocole xwin soit spécifié. Avec des ACL, on doit ouvrir une plage de ports sur 6000-6064.

Pour une session X, avec des CBACS : (de cbac2 sur cbac1, donc de l'extérieur vers l'intérieur)

```
ip inspect audit-trail
ip inspect name monsite tcp
```


```
      ip inspect name monsite rcmd
      ip audit notify log
      ip audit po max-events 100
      !
   interface FastEthernet0/0
      !description Interface interne pour (CBAC1)
      ip address 192.168.230.2 255.255.255.0
      ip inspect monsite in
   interface FastEthernet2/0
      !description Interface externe pour (CBAC2)
      ip address 152.81.34.1 255.255.255.0
      ip access-group 102 in
      !
      access-list 102 permit icmp any any
      access-list 102 permit udp any any eq echo
      access-list 102 permit tcp any any eq echo
      access-list 102 permit tcp any 192.168.230.0 0.0.0.255 eq cmd
      access-list 102 permit tcp any 192.168.230.0 0.0.0.255 range 1020 1024
      access-list 102 deny ip any any log
   RESS_AUDIT_TRAIL: tcp session initiator (192.168.230.100:37550) sent 2944 bytes -
   responder (152.81.34.20:6000) sent 13612 bytes
   cbac-gw#sh ip access-lists 102
   Extended IP access list 102
      permit tcp host 152.81.34.20 eq 6000 host 192.168.230.100 eq 37555 (47 mat-
ches)
      permit icmp any any
      permit udp any any eq echo
      permit tcp any any eq echo
      permit tcp any 192.168.230.0 0.0.0.255 eq cmd (18 matches)
      permit tcp any 192.168.230.0 0.0.0.255 range 1020 1024 (2 matches)
      deny ip any any log
```

Conclusion

Le problème est que ce mécanisme permet donc d'ouvrir une session X sur les machines du réseau interne à partir du réseau externe, mais ne permet pas l'ouverture du session X sur une machine de l'extérieur à partir du réseau interne (ceci est du au fonctionnement de X, c'est le client qui initialise). C'est pourtant ce dernier fonctionnement qui est intéressant. Il faut passer alors par des proxy.

Test Netmeeting

Tests du routeur sur le protocole T.120 et H.323

Pour réaliser l'étude de Netmeeting, nous avons ajouté deux PC à la topologie précédente. L'un d'eux était relié au réseau externe via un concentrateur 100/10 Mbits/s, l'autre était relié au commutateur catalyst via un concentrateur 10 Mbits/s. Les deux PC disposaient de l'application Netmeeting sous Window 95.

Les protocoles utilisés par Netmeeting sont variés et complexes. Ils utilisent TCP,UDP, T120, H.323 (RTP, RTCP) sur des ports dynamiques. Nous signalons que leur utilisation nécessite l'accès aux ports TCP 1503 (pour T.120) et 1720 (H.323 call setup) par l'initiateur de la communication. Les ports TCP 389 (Internet Locator Server) et 522 (User Location Server) peuvent aussi être utilisés. Le port TCP 1731 est aussi utilisé (Audio Call control).

Tests de Netmeeting dans le sens interne-externe

Ci-dessous les modifications apportées à la configuration initiale. On peut noter la présence de l'inspect sur l'interface interne portant sur le protocole h323 et sur tcp.

```
ip inspect name m2 h323
ip inspect name m2 tcp
!
interface FastEthernet0/0
ip address 192.168.230.2 255.255.255.0
ip inspect m2 in
!
interface FastEthernet2/0
ip address 152.81.34.1 255.255.255.0
```

- **Conclusion**

Les différents tests au niveau de Netmeeting sont concluants, l'appel de CBAC2 par CBAC1 permet d'initier un dialogue alors que le contraire pose des problèmes de communication. Le routeur réagit selon nos attentes. Il est à souligner qu'il est nécessaire d'inspecter le protocole H.323 ainsi que TCP comme décrit dans la documentation de Cisco car Netmeeting utilise un canal TCP non défini dans les spécifications de H.323.

Tests de Netmeeting dans le sens externe-interne

L'inspect est cette fois sur l'interface externe avec une access-list qui limite l'entrée du trafic au protocole h323.

```
ip inspect name monsite h323
!
interface FastEthernet0/0
ip address 192.168.230.2 255.255.255.0
!
interface FastEthernet2/0
ip address 152.81.34.1 255.255.255.0
ip access-group 102 in
ip inspect monsite in
!
no access-list 102
access-list 102 permit tcp any any eq 1503
access-list 102 permit tcp any any eq 1720
access-list 102 deny ip any any
```

- **Conclusion**

Les différents tests au niveau de Netmeeting sont concluants, l'appel de CBAC1 par CBAC2 permet d'initier un dialogue alors que le contraire pose des problèmes de communication. La configuration employée n'est pas conventionnelle. En effet pour pouvoir faire fonctionner Netmeeting dans le sens extérieur vers intérieur, nous devons autoriser l'accès aux ports TCP 1503 et 1720 (normal pour l'instant), cependant il est nécessaire d'inspecter le trafic entrant sur l'interface de sortie. Ceci ne peut-être acceptable que si l'interface externe n'est pas l'interne mais par exemple un Intranet.

Autres fonctionnalités de l'IOS Firewall

Nous n'avons pas trop regardé les seuils pour les différents protocoles, et nous avons travaillé avec ceux positionnés en standard. Pour les détections d'attaques, il sera peut-être nécessaire de les adapter.

D'autres fonctionnalités sont à rajouter à notre étude, qui concernent la détection des deny de service et la prévention :

- Authentification et habilitation dynamiques des utilisateurs.
- détection d'attaques de deny de service (syn flooding, port scanning, packet injection).
- détection des intrusions (59 signatures) avec réaction possible.

- dropper des paquets envoyés par l'attaqueur.
- inspection des numéros de séquence des connections TCP.
- détection d'attaque smtp (détecte les commandes smtp invalides).
- blocage des applets JAVA.

Il y aussi la compatibilité avec l'encryption CISCO (VPN) et IPSEC.

■ Exemple de configuration

Des tests ont été effectués avec des ACL de type standard pour un site recherche. Nous avons modifié ces ACL type pour enlever les plages d'ouverture (ftp, r-commandes, udp...,) sur les ports clients, et nous avons introduit les CBACS à la place.

Configuration type avec des ACL

Nous sommes partis de la configuration recommandée avec des ACL PAR L'UREC.

Un réseau interne 192.56.62.0/24 (classe C). On place des filtres sur le routeur d'entrée du site Rs

Tout est interdit sauf :

- 192.56.62.70 est serveur DNS, SMTP, WWW, NTP, FTP, telnet,
- 192.56.62.80 ne doit pas communiquer avec l'extérieur,
- 192.56.62.90 est serveur telnet et ftp uniquement,
- les autres stations peuvent être clientes uniquement.

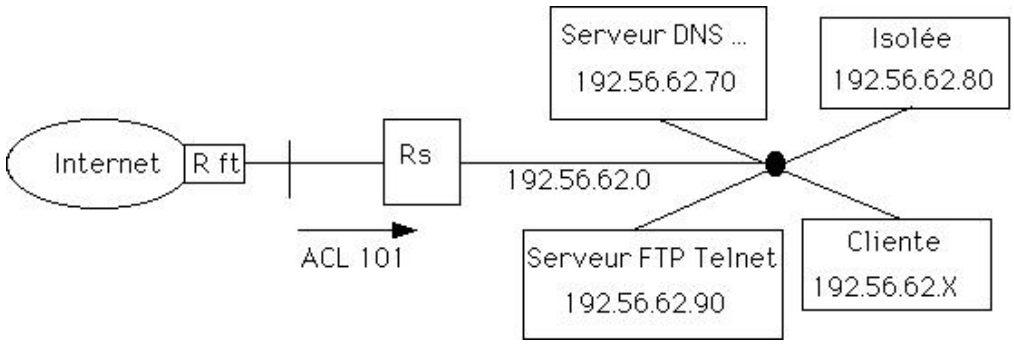


Figure 5.

```
! Description de l'interface du routeur d'entrée cote Internet
interface Ethernet0
ip address 193.5.5.1 255.255.255.0
ip access-group 101 in
! ATTENTION : Nous avons couper toute l'acces-list sur les ports statiques
! Autorise toutes les machines a accéder à l'Internet en mode client
! TCP 1023 pour telnet..., et légèrement < 1023 pour les r-commandes : 960
! Il faut UDP 1023 mais interdit 2000-2003 (OpenWin), 2049 (NFS), 6000-6003 (X11)
access-list 101 permit tcp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 gt 960
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2000
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2001
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2002
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2003
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2049
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 6000
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 6001
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 6002
```

```
access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 6003
access-list 101 permit udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 gt 1023
! TOUT LE RESTE EST INTERDIT:
```

Configuration proposée avec des CBAC

Tout est interdit sauf :

- 192.56.62.70 est serveur DNS, SMTP, WWW, NTP, FTP
- 192.56.62.90 est serveur telnet et ftp uniquement
- Les autres stations peuvent être clientes uniquement

Autorise toutes les machines à accéder à l'Internet en mode client, si l'initialisation de la connection se fait dans le réseau interne et que les ports ne sont plus négociés par la suite pour les protocoles non connus par les CBAC. On peut donc utiliser par exemple les r-commandes, ftp, netmeeting...

- Les inspections à définir (de nom monsite) :

```
ip inspect name monsite tcp
ip inspect name monsite rcmd
ip inspect name monsite ftp
ip inspect name monsite h323
ip inspect name monsite netshow
ip inspect name monsite smtp
ip inspect name monsite tftp
ip inspect name monsite udp
ip inspect name monsite vdolive
```

- L'inspection à faire sur le traffic en sortie du réseau interne, donc en in sur l'interface correspondant au réseau interne
- L'access-list à positionner sur l'entrée du réseau externe (Internet)

ATTENTION CECI PEUT CONTENIR DES ERREURS
! NE PAS APPLIQUER SANS COMPRENDRE CHAQUE LIGNE
! Description de l'interface du routeur d'entrée cote Internet
interface Ethernet0
ip address 193.5.5.1 255.255.255.0
ip access-group 101 in
! ATTENTION : **nous avons couper toute l'acces-list sur les ports statiques**
! **On peut enlever ces autorisations faites maintenant par les CBAC**
!Autorise toutes les machines a accéder a l'Internet en mode client

```
! TCP 1023 pour telnet..., et légèrement < 1023 pour les r-commandes : 960
! Il faut UDP 1023 mais interdit 2000-2003 (OpenWin), 2049 (NFS), 6000-6003 (X11)
!access-list 101 permit tcp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 gt 960
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2000
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2001
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2002
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2003
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 2049
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 6000
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 6001
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 6002
!access-list 101 deny udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 eq 6003
!access-list 101 permit udp 0.0.0.0 255.255.255.255 192.56.62.0 0.0.0.255 gt 1023
! TOUT LE RESTE EST INTERDIT:
```

access-list 101 deny ip any any log

Performances

Outils utilisés

Les tests de performance sont basés sur l'utilisation de l'outil Netperf. Cet outil est libre de droits et est constitué d'un client et d'un serveur.

Tests avec des ACL

Ces tests ont pour but de mesurer l'influence du nombre de règles sur les performances du routeur, bien qu'un nombre de règles égal à 1000 soit tout à fait irréaliste.

Les résultats sont identiques pour UDP et TCP. Dans la plage de valeur généralement utilisée, on constate que le routeur est insensible au nombre de règles. Nous sommes agréablement surpris par les résultats obtenus avec des ACL.

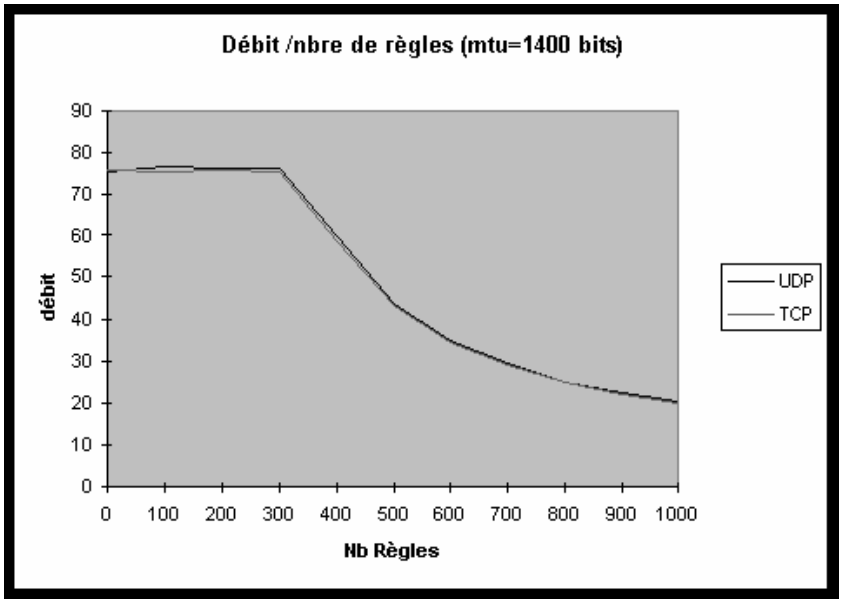


Figure 6. Performances / Nombre de règles.

Tests de performances basés sur le nombre de protocoles inspectés (CBAC)

- L'inspect défini :

```
ip inspect audit-trail
ip inspect name monsite tcp
ip inspect name monsite rcmd
ip inspect name monsite ftp
ip inspect name monsite h323
ip inspect name monsite netshow
ip inspect name monsite smtp
ip inspect name monsite tftp
ip inspect name monsite udp
ip inspect name monsite vdolive
ip inspect name sortie rcmd
ip inspect name sortie tcp
ip audit notify log
ip audit po max-events 100
```

- application à l'interface :

```
interface FastEthernet0/0
description Interface interne pour (CBAC1)
ip address 192.168.230.2 255.255.255.0
no ip directed-broadcast
ip inspect monsite in
media-type MII
full-duplex
```

- performance en TCP :

STREAM TEST to cbac2

Recv Send Send

Socket Socket Message Elapsed

Size Size Size Time Throughput

bytes bytes bytes secs. 10^6bits/sec

65928 65535 65535 60.00 75.73

Nous constatons, sur l'ensemble des mesures, que le nombre de protocoles inspectés **n'influe pas sur les performances**, elles avoisinent les 75 Mbits/s.

Conclusion sur les tests de performance

Les test de performance ont étés réalisés dans le sens d'un trafic depuis le réseau externe vers le réseau interne (trafic le plus pénalisé). Nous constatons que les performances du routeur sont intéressantes et ne se situent qu'à 20 pour-cent en dessous des performances obtenues sans routeur.

En comparaison à Firewall 1 de Sun, les performances du routeur Cisco sont beaucoup plus intéressantes. Rappelons qu'en TCP et UDP Firewall 1 se situait respectivement à 30 et 15 Mbits/s pour une centaine de règles appliquées. Comparativement à PIX, le routeur est moins performant. PIX présentait des performances de l'ordre de 90 Mbits/s en Udp 80 Mbits/s en TCP.

Nous n'avons pas fait de tests sur l'occupation de la mémoire, qui elle est très sollicitée, et peut être pénalisante.

L'introduction des CBAC sur les routeurs nous paraît intéressante dans le sens où un seul et unique équipement permet de réaliser les fonctions associées au routage ainsi que celles d'un firewall. Ce routeur nous paraît apte à réaliser ces deux fonctions dans certaines configurations.

■ Conclusion sur les CBACS

L'IOS Firewall est porté sur Routeur Cisco 800, UBR900, 1600, 1720, 2500, 2600, 3600, 7100 et 7200 et bientôt sur RSM et sur 7500 (octobre 99).

Il permet de renforcer considérablement la sécurité d'un site. Les connexions tcp ou udp initialisées depuis l'intérieur du site peuvent être inspectées, et les ouvertures seront donc dynamiques. La restriction est que l'on doit avoir des ports fixes négociés, hormis les protocoles connus et cités au départ. X11 pose un problème, vu que l'initialisation se fait à l'extérieur du réseau protégé. C'est une solution, sur du matériel existant bien souvent, « tout en un » combinant sécurité, détection des intrusions, authentification et habilitation des utilisateurs, fonction VPN et routage multiprotocole.

■ Petit comparatif avec les autres firewalls

Caractéristiques	IOS FIREWALL	PIX – Firewall hardware	Firewall1 – Firewall applicatif
Performances	bonne	très bonne	moyenne
Filtrage ports dynamiques	certains et internes	certains	tous – développement
Intégration	tout en un	outil dédié à la sécurité	un firewall + antivirus…
Encryption	oui	oui	oui
Alertes	oui	oui	oui
Détection d'intrusion	oui avec signature	oui	non
Prévention des attaques	oui	oui	oui
Authentification	oui	oui (rapide)	oui

■ Documentation

- Guide général sur la sécurité :

Cisco IOS Firewall Overview

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur_c/scprt3/scfirewl.htm

LES pages sécurité UREC

http://www.urec.cnrs.fr/securite/

Les pages sécurité du CRU

http://www.cru.fr/securite/

- Guide sur les CBACs

Contex-based Access Control : Introduction and Configuration

http://www.cisco.com/warp/customer/110/32.html

The CISCO IOS Firewall Feature Set and Context-Based Access Control

http://www.cisco.com/univercd/cc/td/doc/product/software/ios112/ios112p/firewall.htm

Configuring Context-Based Control

http://www.cisco.com/univercd/cc/td/doc/product/software/ios120/12cgcr/secur_c/scprt3/sccbac.htm

Building a Perimeter Security Solution with the Cisco IOS Firewall Feature Set

http://www.cisco.com/warp/customer/732/net_foundation/firew_wp.htm

Cisco IOS Firewall Feature Set

http://www.cisco.com/warp/customer/732/net_foundation/fire_ds.htm

Cisco PIX and CBAC Fragmentation Attack

http://www.cisco.com/warp/customer/770/nifrag.shtml

La page de tests sur les CBACs

http://www.loria.fr/services/moyens-info/securite/CBAC.html