27 sept 2008

Analyser les configurations des équipements et des services réseau

Catégorie : Administration réseau Tags : misc

~~Retrouvez cet article dans : **Misc 19**~~

Les configurations des équipements réseau (commutateurs, routeurs, pare-feu, systèmes de détection d’intrusion, etc.) mettent en œuvre la sécurité « logique » d'un réseau. Cette dernière est établie par des règles de configuration précises réalisées sur ces équipements comme la configuration des règles de filtrage d’un pare-feu, d’un routeur, etc. Toutes ces règles sont solidairement responsables de la mise en place de la politique de sécurité réseau. Si une règle est mal configurée, le principe du maillon le plus faible de la chaîne s’applique.

1. La problématique de la consistance des configurations

Les inconsistances des configurations des équipements réseau dues à des erreurs de configuration, qu’elles soient volontaires ou involontaires, peuvent mettre en danger le réseau mais aussi les serveurs attachés à ce réseau. Ces inconsistances peuvent venir notamment des règles de filtrage ou ACL (Access Control List), définies mais jamais appliquées ou des règles de filtrage appliquées mais jamais définies. On peut aussi avoir au sein d’une ACL des règles qui peuvent être redondantes, voire contradictoires. Cet exemple illustre les redondances et incohérences contenues dans l'ACL suivante [1,2] :

```
 access-list 101 permit IP 10.0.0.0 0.255.255.255 10.0.0.0 0.255.255.255
access-list 101 permit IP 14.0.0.0 0.255.255.255 14.0.0.0 0.255.255.255
access-list 101 permit IP 14.7.6.0 0.0.0.255 14.7.6.0 0.0.0.255
access-list 101 deny IP 14.4.0.0 0.0.255.255 14.4.0.0 0.0.255.255
```

Les inconsistances détectées sont les suivantes :

```
[2] access-list 101 permit ip 14.0.0.0 0.255.255.255 14.0.0.0 0.255.255.255
[3] access-list 101 permit ip 14.7.6.0 0.0.0.255 14.7.6.0 0.0.0.255
```

Les lignes 2 et 3 sont redondantes.

```
[2] access-list 101 permit ip 14.0.0.0 0.255.255.255 14.0.0.0 0.255.255.255
[4] access-list 101 deny ip 14.4.0.0 0.0.255.255 14.4.0.0 0.0.255.255
```

Les lignes 2 et 4 sont contradictoires.
De manière générale, la configuration d'un équipement réseau est constituée de lignes référant à des éléments de configuration. Ces lignes de configuration suivent une syntaxe et un ordre de configuration qui sont propres à chaque type d'équipement. Par exemple, la configuration d'un équipement CISCO est foncièrement différente de celle d'un équipement JUNIPER. Même au sein d'un même constructeur comme CISCO, on peut constater que la configuration d'un PIX est très différente de celle d'un routeur. Chaque équipement met également en œuvre une sémantique différente qui peut engendrer des problèmes subtils comme appliquer une ACL sans pour autant être obligé de la définir ! Quels que soient cette syntaxe et cet ordre, si des éléments de configuration sont définis mais jamais appliqués, si des éléments de configuration sont appliqués sans être définis, si des lignes de configuration sont redondantes ou contradictoires, la configuration de l'équipement réseau est alors inconsistante et peut engendrer des comportements anormaux ou inattendus. L'inconsistance d'une configuration peut donc engendrer de sérieux problèmes de sécurité.

2. Exemple d'analyse

de la consistance de configuration des ACL
Les ACL sont des éléments de configuration permettant de filtrer les flux réseau à des fins de sécurité. Il est donc essentiel d'analyser ces ACL en profondeur. Toute inconsistance détectée sur une ACL impliquant la sécurité, comme le filtrage des protocoles réseau, doit être connue et répertoriée. Nous considérerons qu'une configuration est consistante par rapport aux ACL si les deux conditions suivantes (ou invariants) sont remplies :

- Tous les éléments de filtrage de type ACL définis doivent être référencés.
- Tous les éléments de filtrage de type ACL référencés doivent être définis.

Bien qu'il soit difficile d'associer un niveau d'impact si l'une de ces deux règles n'est pas respectée, on peut dire qu'une ACL définie mais non référencée peut être un sérieux trou de sécurité si celle-ci doit jouer un rôle de filtrage important. De même, une ACL référencée mais non définie est généralement traitée comme une

ACL permissive, c'est-à-dire que tout est permis. Cela n'est évidemment pas souhaitable si l'ACL joue un rôle de filtrage de sécurité.
La vérification de ces invariants peut s'exprimer par le pseudo-code suivant :

```
# lecture d'une configuration
Lire le fichier de configuration

# stockage des filtrages dans des tableaux associatifs
POUR chaque ligne de la configuration FAIRE
                Stocker dans acl_defined si la ligne définit une ACL
                Stocker dans acl_referenced si la ligne référence une ACL
FIN POUR

# vérifier que les filtrages définis sont référencés
POUR chaque élément dans acl_defined FAIRE
                             Si élément n'appartient pas à acl_referenced
                             Alors une ACL est définie et pas référencée.
FIN POUR

# vérifier que les filtrages référencés sont définis
 POUR chaque élément dans acl_referenced FAIRE
                            Si élément n'appartient pas à acl_defined
                            Alors une ACL est référencée mais pas définie.
FIN POUR
```

Le script ~~acl_check.sh~~ (http://www.miscmag.com/articles/19-MISC/conf-reseau/) vérifie que les filtrages de type ACL définis sont référencés et que les filtrages de type ACL référencés sont définis (vérification des invariants de consistance). Ce script est un exemple non exhaustif et devra donc être complété. Par ailleurs, il est écrit en langage AWK et s'exécute sur une configuration CISCO.

De nombreuses autres sections d'une configuration doivent être vérifiées, notamment tout ce qui concerne le routage réseau [2]. Quant aux ACL, la détection des règles inconsistantes, redondantes et inutiles fait l'objet de travaux de recherche [3,4].

3. Exemple d'analyse de la consistance de configuration d'IPSEC

De même que pour les ACL et afin d'assurer les services réseau attendus, les configurations des services réseau doivent être analysées pour s'assurer de l'application de la politique de sécurité.

3.1. Consistance de configuration des CryptoMap

Dans les MISC 15 & 16 – « IPSEC et router CISCO – Fiche Technique » –, des configurations CISCO implémentant IPSEC étaient expliquées. Nous proposons ici de vérifier la consistance d'une configuration IPSEC en considérant la configuration CISCO ~~conf_test~~ suivante :

```
hostname conf_test
!
crypto isakmp key 4cewao6wcbw83 address 192.168.1.154
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
crypto ipsec transform-set chiff_auth esp-3des esp-md5-hmac
!
crypto map VPN_1_1 10 ipsec-isakmp
 set peer 192.168.1.154
 set transform-set chiff_auth
 match address 110
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
 crypto map VPN_1_1
!
access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
!
end
```

Le script ~~ipsec.sh~~ (http://www.miscmag.com/articles/19-MISC/conf-reseau/) vérifie que les éléments IPSEC définis sont référencés et que les éléments IPSEC référencés sont définis (vérification des invariants de consistance). Ce script est un exemple non exhaustif et devra donc être complété. Par ailleurs, il est écrit en langage AWK et s'exécute sur une configuration CISCO. Si on exécute ce script sur la configuration IPSEC ~~conf_test~~, on obtient alors le résultat suivant (aucune inconsistance n'a été détectée) :

```
bash$ awk -f ./ipsec.sh ./conf_test
bash$
```

Modifions la configuration IPSEC afin d'introduire des inconsistances comme l'illustre la commande UNIX ~~diff~~ entre les deux fichiers ~~conf_test~~ et ~~conf_test1~~ :

```
bash$ diff ./conf_test ./conf_test1
1c1
< hostname conf_test
---
> hostname conf_test1
11c11
< crypto ipsec transform-set chiff_auth esp-3des esp-md5-hmac
---
> crypto ipsec transform-set chiff_auth1 esp-3des esp-md5-hmac
16c16
<  match address 110
---
>  match address 120
20c20
<  crypto map VPN_1_1
---
>  crypto map VPN_1_11
```

Si on exécute ce script sur la nouvelle configuration IPSEC ~~conf_test1~~, on obtient alors le résultat suivant pointant sur les inconsistances de configuration :

```
bash$ awk -f ./ipsec.sh ./conf_test1
./conf_test; déf/non réf;110;access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255(line 22)
./conf_test; déf/non réf;chiff_auth1;crypto ipsec transform-set chiff_auth1 esp-3des esp-md5-hmac(line 11)
./conf_test; déf/non réf;VPN_1_1;crypto map VPN_1_1 10 ipsec-isakmp(line 13)
./conf_test; réf/not déf;chiff_auth ; set transform-set chiff_auth(line 15)
./conf_test; réf/not déf;120 ; match address 120(line 16)
./conf_test; réf/not déf;VPN_1_11 ;FastEthernet0; crypto map VPN_1_11(line 20)
bash$
```

Il doit être noté que CISCO ne permet pas de supprimer une cryptomap si celle-ci est appliquée sur des interfaces, à moins de la supprimer sur toutes les interfaces préalablement.

En revanche, l'inconsistance de configuration d'une ACL, non définie mais appliquée dans une cryptomap, peut effectivement bloquer le routeur et le service réseau associé.

3.2. Consistance de configuration des politiques ISAKMP

Si on désire vérifier la consistance de configuration de la politique IPSEC ~~isakmp~~ de deux routeurs, une technique un peu simpliste consiste à parcourir les configurations, à extraire les éléments clés de la politique ~~isakmp~~ et à contrôler les éléments qui ne sont pas des doublons. Voici le script écrit en langage AWK qui s'exécute sur deux configurations. Ce script est un exemple non exhaustif et devra donc être complété.

```
# !/bin/sh

awk '
/crypto isakmp policy/, /!/ {
if ($0 ~ /crypto isakmp policy/) {
            policy=$0;
} else {
            # imprime une ligne de la politique de sécurité
            if ($0 !~ /!/) print policy, $0;
}

# On trie et on imprime les doublons
}' $1 $2 | sort | uniq -u
```

Si on exécute ce script sur deux fichiers (~~conf_test~~ et ~~conf_test1~~) contenant la même configuration ~~isakmp~~ IPSEC, on obtient alors le résultat suivant (les politiques ~~isakmp~~ sont identiques) :

```
bash$ ./ipsec_isakmp.sh ./ conf_test ./conf_test1
bash$
```

En revanche, si on modifie dans ~~conf_test~~ la politique ~~isakmp~~ (~~3des~~ en ~~des~~ et ~~group~~

2 en group 1), on obtient alors le résultat suivant pointant sur les déviations de la politique isakmp :

```
bash$ ./ipsec_isakmp.sh ./ conf_test ./conf_test1
crypto isakmp policy 10  encr 3des
crypto isakmp policy 10  encr des
crypto isakmp policy 10  group 1
crypto isakmp policy 10  group 2
```

3.3. Consistance de configuration des périmètres IPSEC

Si on désire vérifier les périmètres de configuration des réseaux privés virtuels IPSEC, l'approche consiste à analyser le graphe IPSEC engendré par les configurations des VPN IPSEC. Pour cela, nous considérerons tout d'abord que le nom d'une cryptomap suit la règle de configuration suivante :

VPN_X_Y

- X : identifiant unique d'un VPN IPSEC
- Y : instance d'une nouvelle politique pour un VPN IPSEC

Par exemple, la cryptomap VPN_1_1 correspond au VPN 1 et à la politique de sécurité 1. De même, VPN_1_2 correspond au VPN 1 et à la politique de sécurité 2.

Ensuite, si pour chaque configuration on arrive à renseigner les champs de la table IPSEC suivante (il peut avoir plusieurs enregistrements par configuration de routeur), il est alors possible de construire le graphe IPSEC que nous détaillerons par la suite :

Table IPSEC	
champ *NomRouteur*	nom du routeur
champ *CryptomapId*	identifiant unique d'un VPN IPSEC
champ *IpAdresse*	adresse IP de l'interface où est appliquée une cryptomap
champ *IpAdresseDestination*	adresse IP destinatrice du tunnel IPSEC

Le script ipsec_extract.sh (http://www.miscmag.com/articles/19-MISC/conf-reseau/), écrit en langage AWK, s'exécute sur une configuration CISCO et permet d'extraire ces informations. Ce script est un exemple non exhaustif et devra donc être complété.
Si on exécute ce script sur la configuration CISCO suivante :

```
hostname conf_test
!
crypto isakmp key 4cewao6wcbw83 address 192.168.1.154
```

```
crypto isakmp key pvntl2o9xsra5 address 192.168.1.155
crypto isakmp key 6rtzlmkw6awvp address 192.168.1.156
crypto isakmp key p0vzuxb74uvjx address 192.165.1.154
crypto isakmp key pfjgkw1ml3hl8 address 192.165.1.155
crypto isakmp key qgp5h3fblo92p address 192.165.1.156
!
crypto isakmp policy 10
 encr 3des
 hash md5
 authentication pre-share
 group 2
!
crypto ipsec transform-set chiff_auth1 esp-3des esp-md5-hmac
!
crypto map VPN_1_1 10 ipsec-isakmp
 set peer 192.168.1.154
 set peer 192.168.1.155
 set peer 192.168.1.156
 set transform-set chiff_auth
 match address 110
!
crypto map VPN_2_1 10 ipsec-isakmp
 set peer 192.165.1.154
 set peer 192.165.1.155
 set peer 192.165.1.156
 set transform-set chiff_auth
 match address 120
!
interface FastEthernet0
 ip address 192.168.1.1 255.255.255.0
 crypto map VPN_1_1
!
interface FastEthernet1
 ip address 192.165.1.1 255.255.255.0
 crypto map VPN_2_1
!
access-list 110 permit ip 10.0.1.0 0.0.0.255 10.0.2.0 0.0.0.255
access-list 120 permit ip 10.0.3.0 0.0.0.255 10.0.4.0 0.0.0.255
!
end
```

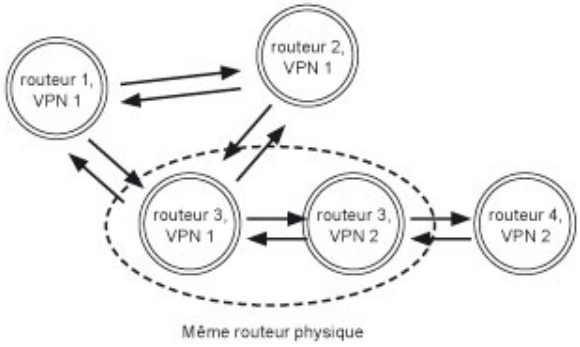
On obtient alors le résultat suivant :

```
bash$ awk -f ./ipsec_extract.sh ./conf_test
./conf_test 1 192.168.1.1 192.168.1.154
./conf_test 1 192.168.1.1 192.168.1.155
./conf_test 1 192.168.1.1 192.168.1.156
./conf_test 2 192.165.1.1 192.165.1.154
./conf_test 2 192.165.1.1 192.165.1.155
./conf_test 2 192.165.1.1 192.165.1.156
```

Une fois la table IPSEC construite à partir de l'extraction des informations contenues dans les configurations, le produit cartésien de la table IPSEC par elle-même, conditionné par le fait que l'adresse IP de l'interface (où est appliquée une cryptomap) soit égale à l'adresse IP destinatrice du tunnel IPSEC et que les cryptomapId soient identiques, donne alors tous les arcs de notre graphe IPSEC comme l'illustre la requête SQL suivante :

```
SELECT
        Ipsec.NomRouteur, Ipsec.CryptoMapId, Ipsec.IpAdresse, Ipsec.IpAdresseDestination,
        Ipsec_1.NomRouteur, Ipsec_1.CryptoMapId, Ipsec_1.IpAdresse, Ipsec_1.IpAdresseDestination
FROM
        Ipsec, Ipsec AS Ipsec_1
WHERE
        Ipsec.IpAdresseDestination=Ipsec_1.IpAdresse and
           Ipsec.CryptoMapId = Ipsec_1.CryptoMapId
```

Un sommet du graphe IPSEC est donc représenté par le couple (NomRouteur, cryptomapId) et un arc par un enregistrement trouvé par le produit cartésien précédemment décrit. Par ailleurs, l'asymétrie de configuration d'un tunnel IPSEC indique que le graphe IPSEC construit est dirigé comme l'illustre le schéma 1.



Dans notre première configuration IPSEC ~~conf_test~~, la table IPSEC serait alors renseignée par les données suivantes si nous avions dans ~~conf_test1~~ la configuration IPSEC associée :

Nom routeur	CryptomapId	Adresse IP de l'interface	Adresse IP du tunnel IPSEC
Conf_test	1	192.168.1.1	192.168.1.154
Conf_test1	1	192.168.1.154	192.168.1.1

Maintenant, si on réalise le produit cartésien précédemment décrit, on obtient alors les informations suivantes (en fait les adresses IP nous permettent d'établir les relations de connectivité entre les sommets) : voir tableau ci-dessous.
Les arcs du graphe IPSEC sont les suivants :
• ~~(Conf_test,1)~~ est ipsec-connecté à ~~(Conf_test1,1)~~
• ~~(Conf_test1,1)~~ est ipsec-connecté à ~~(Conf_test,1)~~
On a donc bien un tunnel IPSEC entre ~~(Conf_test,1)~~ et ~~(Conf_test1,1)~~ par l'asymétrie des connexions entre ces deux sommets. De manière générale, si on considère le graphe IPSEC ayant pour sommets ~~(Conf_test,1)~~ et ~~(Conf_test1,1)~~, le périmètre du VPN IPSEC correspond alors à une composante fortement connexe du graphe IPSEC (si pour toute paire de sommets (x, y) de la composante, il existe un chemin de x à y et de y à x).
Dans notre cas, il y a une seule composante fortement connexe qui est

{(Conf_test,1), (Conf_test1,1)} et qui représente le périmètre de sécurité du VPN 1 [5].

Nom routeur	CryptoMapId	Adresse IP de l'interface	Adresse IP du tunnel IPSEC	Nom routeur	CryptoMapId	Adresse IP de l'interface	Adresse IP du tunnel IPSEC
Conf_test	1	192.168.1.1	192.168.1.154	Conf_test1	1	192.168.1.154	192.168.1.1
Conf_test1	1	192.168.1.154	192.168.1.1	Conf_test	1	192.168.1.1	192.168.1.154

De manière théorique, les composantes fortement connexes du graphe IPSEC donnent les périmètres de sécurité des VPN IPSEC qui ont pu être définis dans les configurations. Ces périmètres de sécurité montrent alors soit l'isolation d'un VPN IPSEC donné, soit des interconnexions avec d'autres VPN IPSEC. Par ailleurs, l'extraction de toutes les composantes fortement connexes d'un graphe est un problème facile [5].
Prenons un réseau plus complexe composé des routeurs et des configurations IPSEC comme illustré dans le schéma 2.

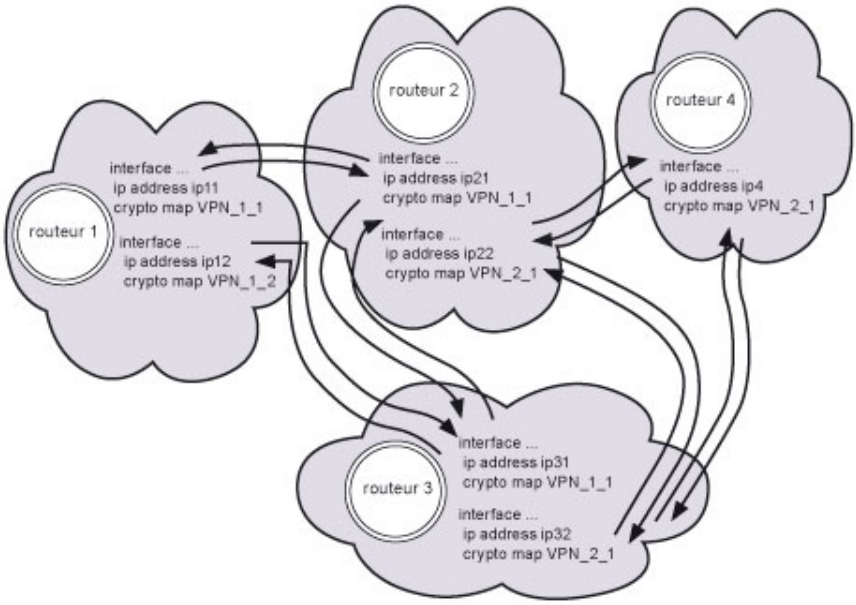


Fig. 2

Si on extrait des configurations les éléments permettant de construire la table IPSEC, on obtient alors après le produit cartésien les informations suivantes :

```
(routeur1,1) est ipsec-connecté à (routeur2,1) && (routeur2,1) est ipsec-connecté à (routeur1,1)
(routeur1,1) est ipsec-connecté à (routeur3,1) && (routeur3,1) est ipsec-connecté à (routeur1,1)
(routeur2,1) est ipsec-connecté à (routeur3,1) && (routeur3,1) est ipsec-connecté à (routeur2,1)
(routeur2,2) est ipsec-connecté à (routeur4,2) && (routeur4,2) est ipsec-connecté à (routeur2,2)
(routeur3,2) est ipsec-connecté à (routeur4,2) && (routeur4,2) est ipsec-connecté à (routeur3,2)
(routeur3,2) est ipsec-connecté à (routeur2,2) && (routeur2,2) est ipsec-connecté à (routeur3,2)
```

Les composantes fortement connexes de notre graphe IPSEC sont donc {(routeur1,1), (routeur2,1), (routeur3,1)} et {(routeur2,2), (routeur3,2),

~~(routeur3,2)~~} et permettent alors de vérifier les périmètres configurés. Dans notre exemple, les périmètres de sécurité sont bien restreints aux VPN IPSEC définis dans les configurations (VPN 1 et VPN 2).
Il est donc possible à partir d'un grand nombre de configurations réseau de retrouver les périmètres des réseaux privés virtuels IPSEC par une analyse des composantes fortement connexes du graphe IPSEC [5].
Enfin, si les composantes connexes (s'il existe un chemin entre toute paire de sommets (x, y) de la composante) du graphe IPSEC ne sont pas égales aux composantes fortement connexes (si pour toute paire de sommets (x, y) de la composante, il existe un chemin de x à y et de y à x) du graphe IPSEC, il y a alors des inconsistances de configuration des VPN IPSEC. De même, toute configuration non bidirectionnelle entre deux sommets montre aussi des inconsistances de configuration des VPN IPSEC.

4. Vers des politiques de consistance des configurations réseau

Les configurations des équipements réseau détiennent toute l'information permettant de construire le réseau et ses services. La politique de sécurité réseau « logique » peut se décliner en l'ensemble de règles de sécurité génériques suivantes garantissant la disponibilité et l'intégrité du réseau et de ses services : voir tableau ci-dessous.

Règles de sécurité génériques	Description
Consistance du plan d'adressage	Il s'agit des règles qui garantissent la consistance du plan d'adressage des équipements réseau. Il ne doit pas exister de doublons dans le plan d'adressage global du réseau, et il ne doit pas exister de doublons dans le plan d'adressage d'un VPN donné.
Consistance des configurations	Il s'agit des règles qui garantissent la consistance des configurations des équipements réseau. Tout élément de configuration défini doit être appliqué, et tout élément de configuration appliqué doit être défini. Ces règles peuvent être basiques (vérification de la présence de lignes spécifiques de configuration) jusqu'à des règles plus complexes (vérification de la grammaire associée au langage de configuration).
Consistance des filtrages	Il s'agit des règles qui garantissent la consistance des filtrages utilisés pour contrôler par exemple les flux de données ou de routage. Les éléments constituant un filtrage ne doivent être ni redondants, ni contradictoires entre eux. Ces règles peuvent être basiques (vérification de la présence de lignes spécifiques de configuration) jusqu'à des règles plus complexes (vérification des règles inutiles).
Routage	Il s'agit des règles de configuration relatives à la protection du routage réseau. Ces règles s'appliquent à la fois au routage interne du réseau ainsi qu'aux interconnexions de routage du réseau avec l'extérieur. Ces règles peuvent être basiques (vérification de la présence de lignes spécifiques de configuration) jusqu'à des règles plus complexes (vérification de la topologie du routage interne et externe du réseau, consistance de la politique de routage, etc.).
Service	Il s'agit des règles de configuration relatives à la protection des services du réseau. Ces règles peuvent être basiques (vérification de la présence de lignes spécifiques de configuration) jusqu'à des règles plus complexes (vérification des périmètres de sécurité d'un VPN).
Partenaires	Il s'agit des règles de configuration relatives à la protection des interconnexions avec les services réseau d'un partenaire. Ces règles sont généralement très basiques (vérification de la présence de lignes spécifiques de configuration).
Administration	Il s'agit des règles de configuration relatives à la protection des équipements réseau. Ces règles sont généralement très basiques (vérification de la présence de lignes spécifiques de configuration).

5. Vers des outils d'analyse des configurations (Router Audit Tool)

Écrit par George M. Jones (<gmj@cisecurity.org>), RAT est distribué par le CIS

(Center for Internet Security). Disponible gratuitement sur Internet pour un usage personnel, RAT est composé des programmes suivants :

- ~~rat~~ : le programme principal.
- ~~snarf~~ : qui permet de télécharger les configurations des routeurs.
- ~~ncat_config~~ : qui permet de générer une configuration d'audit.
- ~~ncat_report~~ : qui permet de générer des rapports d'audit.

Avant de lancer tout audit, un fichier de configuration d'audit doit être défini afin de préciser les commandes d'audit ou de vérification de la configuration qui seront lancées. Ce fichier s'appuie sur une bibliothèque de règles définissant des standards de sécurité. Cette bibliothèque est aujourd'hui divisée en deux parties, ~~Level-1 Benchmark~~, qui contient un ensemble de règles élémentaires et ~~Level-2 Benchmark~~, qui contient des règles plus étendues.
Chaque règle contient les champs ou attributs suivants :

- **Impact de sécurité** : décrit l'impact de sécurité associé si la règle n'est pas appliquée.
- **Importance** : associe une valeur entre 1 et 10 reflétant l'importance de l'impact de sécurité si la règle n'est pas appliquée.
- **Actions associées à la règle** : décrit les actions pour corriger et donc appliquer la règle.
- **Expression régulière définissant la règle** : décrit une expression régulière à partir de laquelle l'outil vérifie si une règle est appliquée.

L'outil RAT est aussi accompagné d'un outil d'audit permettant de noter les éléments suivants :

- nombre de tests réalisés ;
- nombre de règles de sécurité appliquées ;
- nombre de règles de sécurité non appliquées ;
- pourcentage de règles de sécurité appliquées ;
- score pondéré par l'importance des règles de sécurité appliquées ;
- score pondéré par l'importance si toutes les règles de sécurité sont appliquées.

RAT est le premier outil permettant d'analyser les configurations des routeurs. Il évolue sans cesse et intègre désormais des vérifications de plus en plus évoluées. Cependant, les scores fournis doivent être interprétés non pas comme un niveau de sécurité, mais plutôt comme un indicateur d'application des règles de sécurité.

6. Conclusion

Les configurations des équipements réseau détiennent toute l'information permettant de construire le réseau et ses services. La vérification de ces configurations est et deviendra un enjeu majeur dans les années à venir.
Par ailleurs, il s'agira aussi de définir un cadre complet permettant de décrire une politique de sécurité réseau, de vérifier l'application de la politique de sécurité dans

les configurations des équipements réseau, de définir des indicateurs de sécurité afin d'établir un tableau de bord de la sécurité réseau, de quantifier le risque associé à la non application de la politique de sécurité et enfin de définir des priorités pour corriger les faiblesses de sécurité les plus critiques [6,7].

Références

- [1] Valois (D.), Llorens (C.), « Network device configuration validation », Proceedings of 14th annual FIRST conference, Hawaii, 2002.
- [2] Llorens (C.), Valois (D.), Le Teigner (Y.), Gibouin (A.), « Computational complexity of the network routing logical security », Proceedings of the IEEE international Information Assurance Workshop, Darmstadt – Germany, p. 37-49, 2003.
- [3] Warkhede (P.), Suri (S.), Varghese (G.), « Fast packet classification for two-dimensional conflict-free filters », Proceedings of the Twentieth Annual Joint Conference of the IEEE Computer and Communications Societies, vol. 3, p. 1434-1443, 2001.
- [4] Eppstein (D.), Muthukrishnan (S.), « Internet packet filter management and rectangle geometry », Proceedings of the twelfth annual ACM-SIAM symposium on Discrete algorithms, p. 827-835, 2001.
- [5] Brassard (G.), Bratley (P.), Fundamentals of algorithmics, Prentice Hall, ASIN : 0133350681, 1995.
- [6] Llorens (C.), Levier (L.), Tableaux de bord de la sécurité réseau, Eyrolles, ISBN : 2-212-11273-4, 2003.
- [7] Llorens (C.), Conférence SAR 2004, http://www.hds.utc.fr/sar04/files /llorens-pres.pdf

~~Retrouvez cet article dans : **Misc 19**~~

Posté par (La rédaction) | Signature : Cédric Llorens | Article paru dans 

Laissez une réponse

Vous devez avoir ouvert une session pour écrire un commentaire.

« Précédent Aller au contenu »
Identifiez-vous
Inscription
S'abonner à UNIX Garden

Articles de 1ère page

- Introduction à la programmation de LKM sous NetBSD
- EDIGÉO : échanger de l'information géographique
- Voyage à la frontière du noyau

- Linux Pratique HS N°16 - Janvier/Février 2009 - Chez votre marchand de journaux
- Qt4 – 8 : une scène et des voyeurs
- Le langage Ada – 16 Les caractères des chaînes
- Les Éditions Diamond adhèrent à l'APRIL !
- QDVD-AUTHOR, Le Créateur de DVD-Vidéo !
- QDevelop, un environnement de développement pour Qt
- Dissection de glib : les arbres binaires balancés

Catégories

- Administration réseau
- Administration système

- Agenda-Interview
- Audio-vidéo
- Bureautique
- Comprendre
- Distribution
- Embarqué
- Environnement de bureau
- Graphisme
- Jeux
- Matériel
- News
- Programmation
- Réfléchir
- Sécurité
- Utilitaires
- Web

Articles secondaires

- 30/10/2008
  Google Gears : les services de Google offline

  Lancé à l'occasion du Google Developer Day 2007 (le 31 mai dernier), Google Gears est une extension open source pour Firefox et Internet Explorer permettant de continuer à accéder à des services et applications Google, même si l'on est déconnecté....

  Voir l'article...
  7/8/2008
  Trois questions à...

  Alexis Nikichine, développeur chez IDM, la société qui a conçu l'interface et le moteur de recherche de l'EHM....

  Voir l'article...
  11/7/2008
  Protéger une page avec un mot de passe

  En général, le problème n'est pas de protéger une page, mais de protéger le répertoire qui la contient. Avec Apache, vous pouvez mettre un fichier ~~.htaccess~~ dans le répertoire à protéger....

  Voir l'article...
  6/7/2008
  hypermail : Conversion mbox vers HTML

  Comment conserver tous vos échanges de mails, ou du moins, tous vos mails reçus depuis des années ? mbox, maildir, texte... les formats ne manquent pas.

...

Voir l'article...
6/7/2008
iozone3 : Benchmark de disque

En fonction de l'utilisation de votre système, et dans bien des cas, les performances des disques et des systèmes de fichiers sont très importantes....

Voir l'article...
1/7/2008
Augmentez le trafic sur votre blog !

Google Blog Search (http://blogsearch.google.fr/) est un moteur de recherche consacré aux blogs, l'un des nombreux services proposés par la célèbre firme californienne....

Voir l'article...

- **GNU/Linux Magazine**

  - Les Éditions Diamond adhèrent à l'APRIL !
  - Nouvelle campagne d'adhésion de l'APRIL !
  - GNU/Linux Magazine N°111 - Décembre 2008 - Chez votre marchand de journaux
  - Édito : GNU/Linux Magazine 111
  - GNU/Linux Magazine N°110 - novembre 2008 - Chez votre marchand de journaux

- **GNU/Linux Pratique**

  - Linux Pratique HS N°16 - Janvier/Février 2009 - Chez votre marchand de journaux
  - Édito : Linux Pratique HS N°16
  - Linux Pratique HS 16 - Communiqué de presse
  - Les Éditions Diamond adhèrent à l'APRIL !
  - Nouvelle campagne d'adhésion de l'APRIL !

- **MISC Magazine**

  - Les Éditions Diamond adhèrent à l'APRIL !
  - Misc HS 2 : Cartes à puce, Découvrez leurs fonctionnalités et leurs limites - Novembre/Décembre 2008 - Chez votre marchand de journaux
  - Édito : Misc HS 2
  - MISC HS 2 - Communiqué de presse
  - Références de l'article « Un SDK JavaCard générique ou comment

développer une application carte complète pour toutes les cartes Java » de Vincent Guyot paru dans MISC HS 2