|  |  |  |
| --- | --- | --- |
| port knocking  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  port knocking (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  port knocking (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  port knocking (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png  port knocking (suite)  http://upload.wikimedia.org/wikipedia/commons/thumb/3/35/Tux.svg/512px-Tux.svg.png |  | port knocking Sécuriser l'accès à un port  Le **port-knocking** est une méthode permettant de modifier le comportement d'un pare-feu en temps réel pour provoquer l'ouverture d'un port à la suite du lancement préalable d'une série de connexions sur des ports distincts dans le bon ordre, tout comme un code frappé à une porte.  Cette technique est notamment utilisée pour protéger l'accès au port 22 dédié à **Secure shell** (SSH); elle ne nécessite pas beaucoup de ressources et reste facile à mettre en œuvre.  En résumé, le port knocking est donc une méthode simple pour autoriser un accès distant sur un port non constamment ouvert. Cela permet d'éviter les balayages de ports (port *scans*) et certaines attaques.  La méthode de port-knocking est considérée comme sécurisée étant donné qu'elle est située à un niveau bas des couches TCP/IP et qu'elle ne requière pas de port ouvert (le service *knockd* est lui aussi invisible).  C'est une méthode, sans toutefois être infaillible, permettant de masquer la présence d'un serveur derrière un port.  Paramétrage du serveur  Le serveur doit avoir un pare-feu et exécuter le service *knockd*.  Le service est en écoute d'une séquence spécifique (suite de ports associés à un protocole TCP et/ou UDP).  Si la séquence est correcte, alors une commande est exécutée.  Généralement, l'adresse IP source est autorisée à accéder au port d'une application à travers le pare-feu.  La configuration du serveur est très simple puisqu'il suffit d'installer le paquet **knockd**.  **>> sudo apt update**  **>> sudo apt -y upgrade**  **>> sudo apt install -y knockd**  Après avoir installé le paquet, il faut éditer le fichier de configuration **/etc/knockd.conf**.  Deux approches sont, entre autres, possibles …   * la première est destinée aux connexions qui ne nécessitent pas d'être maintenues (comme HTTP) ; * la seconde méthode à celles qui doivent l'être (comme SSH).   **Connexions non maintenues (*no keep-alive*)**  **...**  **[options]**  **logfile = /var/log/knockd.log**  **[openHTTP]**  **sequence = 7000,8000,9000**  **seq\_timeout = 5**  **command = /sbin/iptables -insert INPUT --source %IP% --proto tcp --dport 80 --jump ACCEPT**  **tcpflags = syn**  **[closeHTTP]**  **sequence = 9000,8000,7000**  **seq\_timeout = 5**  **command = /sbin/iptables --delete INPUT --source %IP% --proto tcp --dport 80 --jump ACCEPT**  **tcpflags = syn**  **...**  Quelques explications …   * **sequence** Combinaison qui doit concorder afin d’effectuer la ligne **command**.  Si aucun protocole n'est mentionné, par défaut le protocole TCP est utilisé ; * **seq\_timeout** Temps (en secondes) pendant lequel la combinaison doit être effectuée, après quoi elle est annulée. Il faut alors recommencer la séquence depuis le début ; * **command**Règle **iptables** qui sera ajoutée au pare-feu afin de permettre l'ouverture/fermeture d'un port particulier ; * **tcpflags**Ce paramètre met en évidence les drapeaux (*flags*) devant être positionnés dans les paquets reçus pour que le port soit pris en compte. Si une erreur survient, il faut recommencer le tout.   Deux (2) séquences ont été définies :   * **openHTTP** ouvre le port HTTP 80 si la séquence de ports 7000, 8000 et 9000 a été exécutée; * **closeHTTP** ferme le port HTTP 80 si la séquence de ports 9000, 8000 et 7000 a été exécutée.   **Remarque** … Il est possible de préciser le protocole « frappé » en adaptant la ligne comme suit par exemple … **sequence = 7000:tcp,8000:udp,9000:tcp**  **Connexions maintenues (*keep-alive*)**  Un deuxième exemple est un peu différent du précédent …  **...**  **[options]**  **logfile = /var/log/knockd.log**  **[SSH]**  **sequence = 7000,8000,9000**  **seq\_timeout = 5**  **command = /sbin/iptables --insert INPUT --source %IP% --proto tcp --dport 22 --jump ACCEPT**  **tcpflags = syn**  **cmd\_timeout = 10**  **stop\_command = /sbin/iptables --delete INPUT --source %IP% --proto tcp --dport 22 --jump ACCEPT**  **…**  Il est recommandé de n'ouvrir le port qu'un court instant (10 secondes dans cet exemple).  Pour que cela soit fonctionnel, le pare-feu doit être de type **stateful**  (il doit exister une règle pour accepter les connexions avec **--match state –state RELATED,ESTABLISHED**).  Si un utilisateurs « frappe » dans l'ordre les ports 7000, 8000 puis 9000, la commande sera exécutée  (ouverture du port 22).  10 secondes plus tard, la commande stop\_command sera exécutée, fermant ainsi le port pour les nouvelles connexions (mais non celle en cours)  **Attention** …  Le site officiel du projet knockd (et le fichier de configuration installé par défaut) fait référence à la règle  **iptables --append**. Comme on peut le voir, on utilise à la place *-I* pour insérer la règle au début du pare-feu  (sinon la règle ajoutée avec **-append** ne produit aucun effet puisqu'elle est ajoutée après la règle **DROP**)  Toutefois, tout dépend de la règle par défaut, si la décision est drop le --append fonctionnera, mais --insert fonctionne dans tous les cas.  Il faut impérativement modifier la séquence d'exemple du fichier de configuration, et bien entendu, pensez à communiquer la nouvelle en gardant à l'esprit qu'il est recommandé …   * d'utiliser au moins 3 ports  (il est possible d’ajouter plus de ports) ; * de mélanger pour chaque port les protocoles TCP et UDP.   Il faut également modifier la configuration par défaut dans le fichier **/etc/default/knockd** …   * pour que le service démarre automatiquement,  la valeur de la ligne **START\_KNOCKD=** doit être à **1** ; * pour que le service écoute sur des interfaces,  adapter et activer la ligne **KNOCKD\_OPTS="--in-interface ens33"** (en enlevant le '#') * afin que les modifications soient prises en compte,  redémarrer le service *knockd* avec la commande :  **>> sudo /etc/init.d/knockd restart**   Paramétrage du client  Pour initier une séquence sur les différents ports, il est possible d’utiliser le client de votre choix, comme **telnet** ou **nc**.  **Exemple 01**  Le logiciel client spécifique au port-knocking comme knock, après avoir installé le paquet knockd (sur le client cette fois-ci) sera utilisé.  Pour l'utiliser, lancer simplement la commande (l'utilisation du protocole TCP est faite par défaut) :  **>> knock <Adresse du serveur> <Port-1> <Port-2> <Port-3>**  ou si il est nécessaire de préciser dans la combinaison plusieurs protocoles :  **>> knock 192.168.1.100 7000:tcp 8000:udp 9000:tcp**  ou si la combinaisons n'utilise que des ports udp :  **>> sudo knock -u 192.168.1.100 7000 8000 9000**  **Exemple 02**  Le logiciel netcat sera utilisé afin de frapper aux portes.  netcat utilise seulement une séquence de port du même protocole de la couche Transport soit seulement TCP ou seulement UDP.  La combinaison des deux protocoles pour une même connexion n’est pas possible … **>> nc -z <Adresse IP du serveur> <Port-1> <Port-2> <Port-3> >> nc -z 192.168.1.100 7000 8000 9000**  **Note**…  netcat est aussi disponible sous Windows.  **Adapté du document original**  <http://doc.ubuntu-fr.org/port-knocking> |
|  |  |  |