logiciel Nessus - un scanner de vulnérabilité

source

https://fr.wikipedia.org/wiki/Nessus_(logiciel)
http://igm.univ-mlv.fr/~dr/XPOSE2009/Nessus/nessus_scan.html
https://www.tenable.com/products/nessus
https://sites.google.com/site/dossierperso1/nos-menus-et-formules/les-menus-composes

c'est quoi ?

Nessus est un scanner de sécurité réseau, il détecte et signale les failles potentielles ou avérées comme :

- les service vulnérables à des attaques permettant la prise de contrôle de la machine, l'accès à des informations sensibles, des dénis de service
- les fautes de configuration (relais de messagerie ouvert par exemple)
- les patchs de sécurité non appliqué
- les mot de passe par défaut, quelques mots de passe commun, et l'absence de mot de passe sur certains comptes systèmes. Nessus peut aussi appeler le programme externe hydra pour attaquer les mot de passe à l'aide d'un dico
- les services jugés faibles (on suggère par exemple de remplacer Telnet par SSH)
- les dénis de service contre la pile TCP/IP

Comment fait-il ça ?

- soit en identifiant un numéro de version, mais ce procédé est limité aux failles de service de réseau exploitables seulement localement.
- soit en récupérant la liste des logiciels ou paquets installés sur la machine testée et en la comparant aux patchs publiés par les éditeurs.

fonctionnement ?

Par rapport à d'autres scanner de vulnérabilité, Nessus à la particularité d'être basé sur une architecture client/serveur et d'être compatible avec Windows et Linux. En plus, Nessus stocke et gère toutes ses failles de sécurité grâce à un système de plugins. Ces plugins sont en C compilé ou en NASL (Nessus Attack Scripting Language)

le programme se divise en 2 parties :

1. Nessusd : la partie serveur qui est un daemon qui exécute les requêtes et la communication avec la cible. En l'installant on a accès à des exécutables:
    a. nessus-update-plugins : màj plugins

    b. nessus-adduser : création user. Le client en a besoin pour s'authentifier auprès du serveur
    c. nessus-mkcert : permet de créer des certificats côté serveur. Cela permet de protéger les communications entre le client et le serveur en utilisant SSL.

   Le serveur possède un fichier de configuration qui viendra écraser d'éventuelles configurations clientes, il possède aussi un fichiers permettant de centraliser les règles pour les scans
2. Nessus : la partie client qui sert à se connecter au serveur et à lui envoyer la liste des machines à scanner. Il possède aussi des exécutables pour la génération de rapports sous différents formats et pour leur conversions dans d'autres formats.

Le découpage est classique, le daemon tourne avec des privilèges élevés (root) alors que l'interface graphique, plus complexe et donc vulnérable, tourne sous l'identité d'un utilisateur non privilégié.

Le processus de fonctionnement est le suivant :

1. D'abord il va détecter si la machine visée est vivante ou non. C'est grâce au plugin ping_host.nasl, qui va effectuer des pings graduels allant de requête ARP au ping ICMP. Et il va terminer par un ping UDP applicatif
2. après ça, il va scanner les ports des machines vivantes avec un des 4 scanners de ports internes ou externes comme nmap. En sachant que Nessus est optimisé s'il utilise ses scanners de port. Il est donc déconseillé d'utiliser nmap pour des raisons de performances.
3. Selon la configuration de l'utilisateur, Nessus effectue un scan local ou distant.
4. récupération d'information
    a. type et version des divers services
    b. connexion (SSH, Telnet ou rsh) pour récupérer la liste des packages installés
5. attaques simples, peu agressives. par exemple directory traversal, test de relais de messagerie ouverts etc
6. attaques susceptibles d'être destructrices
7. dénis de services (contre les logiciels visés)
8. dénis de services contre la machine ou les équipements réseaux intermédiaires;

Les derniers tests plus agressifs peuvent avoir des conséquences sur la machines cible, sur sa disponibilité (crash de la machine voir des équipements réseau), ces tests peuvent être désactivé dans le mode safe check.

Le scan distant

Le scanner distant est un scanner de port à la nmap. Il permet de déterminer les ports ouverts et le type de service réseaux de machines distantes. Il reconnaît la version des services en parsant les bannières de bienvenue ou les en-têtes dans les trames de réponses. Il effectue donc une analyse TCP/IP complète et reconnaît les algorithmes de logiciels selon le type de trames de réponse.

Avantages

1. Il peut avoir plusieurs instances du client mais aussi du serveur. Il peut également effectuer plusieurs attaques simultanément sur une même machine.
2. Il peut lancer ses attaques sur plusieurs hôtes cibles simultanément

La problématique est de trouver un bon compromis entre le nombre d'hôtes que l'on souhaite scanner et la bande passante du serveur Nessus.

Inconvénients

1. peut générer des faux positifs.
   Nessus peut détecter une faille de sécurité sur un service obsolète, alors qu'un patch de sécurité est appliqué sur la machine local pour corriger les bugs de cette version. Pour corriger ça, il y a besoin d'un scan local.
2. peut générer une surcharge du réseau.
   Dans le cas où l'on scannerait un réseau entier et que le serveur Nessus ne se trouve pas dans ce réseau, les requêtes envoyées par Nessus peuvent saturer la table de translation de firewall intermédiaire.

De plus, toutes les adresses IP d'un réseau ne sont pas forcément utilisées. Nessus peut donc envoyer des requêtes à travers le réseau qui n'aboutiront jamais. La détection des machines vivantes est donc nécessaire mais ne rend pas le scan optimal.

Le scan local

Pour un tel scan, Nessus doit se connecter en local à la machine cible et envoie ses attaques. Il doit utiliser un compte utilisateur local avec les droits les plus importants de la machine pour qu'un maximum de tests puissent être valides.

Avantages

Ce type de scan est plus complet que le scan distant et permet de faire des tests sans avoir à se soucier des faux positifs. Comme tests on retrouve :

- des mots de passe par défaut ou de faible complexités
- des fautes de configurations de logiciels
- des versions de dll ou de package obsolète
- des services désactivés et vulnérables

Un utilisateur Nessus qui souhaite uniquement déterminer si ses machines sont à jour, exécutera un scan local car c'est extrêmement rapide.

Inconvénient

Pour effectuer ses tests, Nessus lance de réelles attaques contre les machines cibles. En d'autres termes, avec ses scans locaux, Nessus peut facilement effectuer un déni de service. Il est donc prudent de prévenir cette attaque soit, en effectuant le scan sur une

plage horaire où la machine cible ne craint pas de perte de données soit, en spécifiant dans les fichiers de configuration que l'on souhaite effectuer, un scan safecheck.

De plus, certains IDS (comme snort) peuvent détecter les scans de Nessus comme étant une attaque de hacker.
Il faut donc également penser à configurer correctement son IDS.(intrusion detection system)

La gestion de rapports

A la fin d'un scan, Nessus génère un rapport qui permet d'établir la liste des machines scannées, leurs OS, leurs services et leurs vulnérabilités. Pour chaque vulnérabilité est associé une criticité, une description et une solution. Grâce à ça il est possible de connaître le risque d'une faille de sécurité, les éventuels effets sur le système si quelqu'un venait à l'exploiter et les patchs de sécurité à appliquer pour supprimer le risque.

Ces rapports peuvent donc être généré sous différents formats :

- HTML simple ou avec graph
- XML
- LaTEX
- NSR
- NBE
- TXT

Par défaut ils sont en NBE(Nessus Back ENd report). Dans ce type de rapports, les informations sont séparés par des pipes.
la première information représente un flag:

- timestamp : donne des indications temporelles sur le scan.(montrer exemple)
- result : décrit le résultat du test effectué. Permet de connaître :
    - sous réseau de la machine
    - le nom de la machine/ adresse IP
    - protocol de communication utilisé
    - le type de vulnérabilité et sa description

```
timestamps|||scan_start|Wed Oct 21 18:05:26 2009|
timestamps||192.168.0.2|host_start|Wed Oct 21 18:05:31 2009|
results|192.168.0|192.168.0.2|general/tcp|10180|Security Note|The remote host is up\n
results|192.168.0|192.168.0.2|general/icmp|10114|Security Warning|\nThe remote host ans
results|192.168.0|192.168.0.2|general/udp|10287|Security Note|For your information, her
results|192.168.0|192.168.0.2|general/tcp|19506|Security Note|Information about this sc
results|192.168.0|192.168.0.2|general/tcp|9999|Security Hole|\nYou are running a versio
timestamps||192.168.0.2|host_end|Wed Oct 21 18:07:53 2009|
timestamps|||scan_end|Wed Oct 21 18:07:53 2009|
```

La sécurité dans Nessus

Sécurité du serveur

Il paraît évident de placer la machine physique dans un endroit sécurisé.

Au niveau logique, la sécurité du serveur Nessus est plus complexe. Pour une sécurité maximale, il faudrait placer le serveur dans un réseau local inaccessible de l'extérieur. D'une manière globale, le serveur Nessus doit définir une politique de sécurité pour contrôler son accès.

Configuration des droits du serveur

On peut configurer Nessus de façon à ce qu'il ne scan pas certaines adresses IP ou certains plugins.
Pour éviter que les scans locaux ne viennent DDoS les machines cible, il est possible d'activer un safecheck. Cette option va forcer le serveur Nessus à reconnaître les services uniquement sur les bannières de bienvenue signalées par les hôtes cibles au lieu d'essayer de reconnaître l'empreinte du service. De plus, cette option va désactiver tous les plugins qui peuvent effectuer des buffers overflow sur les hôtes cibles.

Configuration du serveur pour effectuer des scans locaux

Pour les scans locaux, il faut que le serveur se connecte localement à la machine. Pour se faire, Nessus propose différents types de connection qui se trouve dans le fichier source du serveur :

```
SNMP settings[entry]:SNMPv3 user name : =
SNMP settings[password]:SNMPv3 authentication password : =
SNMP settings[password]:SNMPv3 privacy password : =
# Pour telnet, rsh, rexec :
Cleartext protocols settings[entry]:User name : =
Cleartext protocols settings[password]:Password (unsafe!) : =
SSH settings[entry]:SSH user name : = root
SSH settings[radio]:Elevate privileges with : = Nothing;sudo;su
SSH settings[entry]:Preferred SSH port : = 22
SSH settings[password]:SSH password (unsafe!) : =
SSH settings[file]:SSH public key to use : =
SSH settings[file]:SSH private key to use : =
SSH settings[password]:Passphrase for SSH key : =
SSH settings[password]:su/sudo password : =
SSH settings[file]:SSH known_hosts file : =
SSH settings[entry]:Client version : = OpenSSH_5.0
Kerberos configuration[entry]:Kerberos Key Distribution Center (KDC) : =
Kerberos configuration[entry]:Kerberos Realm (SSH only) : =
Services[file]:SSL certificate : =
Services[file]:SSL private key : =
Services[password]:PEM password : =
Services[file]:CA file : =
Database settings[entry]:Login : =
Database settings[password]:Password : =
Database settings[entry]:Database SID : =
Database settings[entry]:Database port to use : =
Login configurations[entry]:HTTP account : =
Login configurations[password]:HTTP password (sent in clear) : =
Login configurations[entry]:NNTP account : =
Login configurations[password]:NNTP password (sent in clear) : =
Login configurations[entry]:POP2 account : =
Login configurations[password]:POP2 password (sent in clear) : =
Login configurations[entry]:POP3 account : =
Login configurations[password]:POP3 password (sent in clear) : =
Login configurations[entry]:IMAP account : =
Login configurations[password]:IMAP password (sent in clear) : =
Login configurations[entry]:SMB account : = admin
Login configurations[password]:SMB password : = password
Login configurations[entry]:SMB domain (optional) : =
Login configurations[entry]:SNMP community (sent in clear) : =
SLAD Init[file]:slad SSH public key: =
SLAD Init[file]:slad SSH private key: =
SLAD Init[password]:slad SSH key passphrase: =
Global variable settings[checkbox]:Do not log in with user accounts not specified in the policy = no
Global variable settings[file]:SSL certificate to use : =
Global variable settings[file]:SSL CA to trust : =
Global variable settings[file]:SSL key to use : =
Global variable settings[password]:SSL password for SSL key : =
```

Certaines connexions comme l'accès aux bases de données ou encore aux serveurs web sont configurées pour tenter des attaques par injections de code.
D'autres connexions sont établies (avec une authentification en clair : login + mot de passe) dans le but de rechercher les fautes de configuration (comme samba, snmp, nntp, pop2, pop3, imap). Enfin, pour se connecter aux systèmes d'exploitation, plusieurs méthodes d'authentification sont proposées par Nessus :

- telnet: à proscrire car l'authentification transite en clair sur le réseau
- SSH avec login + pw : à proscrire car sensible aux attaques par dictionnaire et l'inscription du login et du pw sont en clair dans le fichier de configuration
- SSH clé privé + publique : à conseiller
- NTML et NTMLv2 pour l'authentification sous windows
- Kerberos : méthode permettant l'authentification de users sur des systèmes Win ou Linux dans un réseau (nécessite une architecture spécifique

Pour sécuriser la connexion entre client/serveur, le serveur doit générer un certificat à l'aide d'un de ces plugins. Grâce au protocol SSL, la communication client/serveur est sécurisé

Conclusion ?

C’est un logiciel qui effectue de réelles attaques et présente le résultat de ces attaques sous forme de rapport complet.
Son utilisation peut donc être à double tranchant.
D’un côté, une équipe sécurité peut l’utiliser pour scanner son réseau dans le but de prévenir les intrusions et le dénis de service.
D’un autre côté, un hacker peut l’utiliser à des fins opposées au premier groupe et en profiter pour exploiter des vulnérabilités déclarées.