Compromettre son réseau en l'auditant ?

R. Deraison - Tenable Network Security

Agenda

- But de cette présentation
- Injection de vulnérabilités
- Empecher l'audit
- Obtention des identifiants
- Reverse overflow
- Présentation du design de NASL

But de cette présentation

- Pas une recette pour compromettre un réseau avec les outils des autres
- Réfléchir quand au choix d'un outil de sécurité donné avant de lui donner les clés du réseau
- Présenter les questions que l'on se pose lors du développement de Nessus et les réponses que nous avons apporté
- Attention: mon point de vue n'est pas neutre

Qu'est ce qu'un scanner ?

- "Couteau suisse" d'implémentations clientes
- SMTP, NNTP, FTP, SMB, NFS, DHCP, HTTP, etc...
- La plupart des protocoles sont codés de manière ad-hoc, souvent rapidement (suite à la détection d'une faille)
- Probabilité d'un bug dans au moins une des implémentation ?

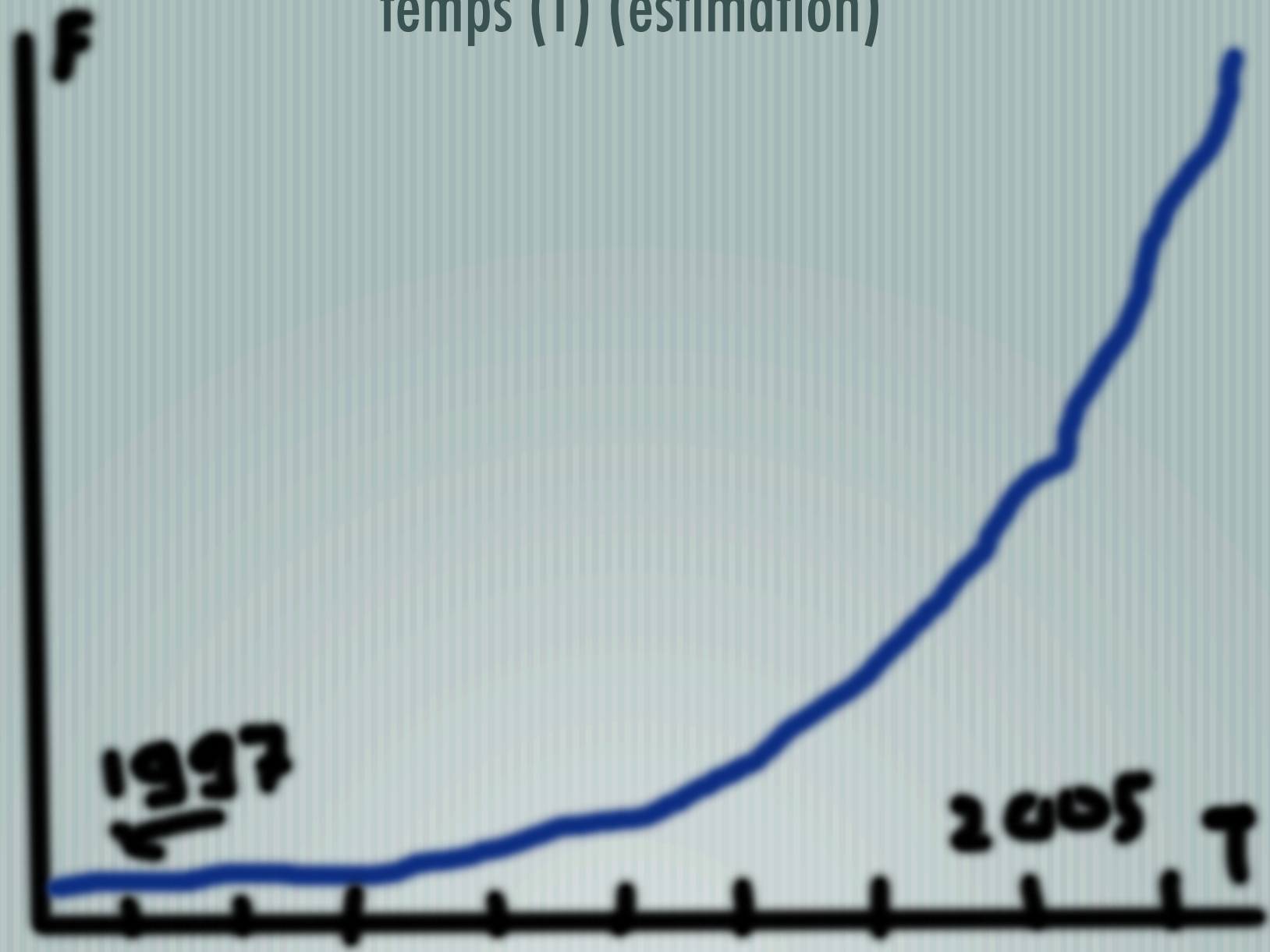
Qu'est ce qu'un scanner ?

- Dissection de contenu ("content parser"), plus limitée mais néanmoins existante
- HTML/JavaScript, fichiers .exe (GetFileVersion), etc...
- Parser HTML ad-hoc ou basé sur un moteur existant (IE?)
- Probabilité d'une faille dans un des dissecteurs ?

Evolution des failles

- Augmentation des failles clients (clients mail, tftp, etc...)
- Augmentation des failles de type content-parsing (anti-virus, suite bureautique, IE, etc...)

Evolution du nombre de failles client/content-parser (F) au cours du temps (T) (estimation)



Qu'est-ce qu'un administrateur facétieux (ou un pirate) pourrait faire pour géner l'équipe de sécurité ?

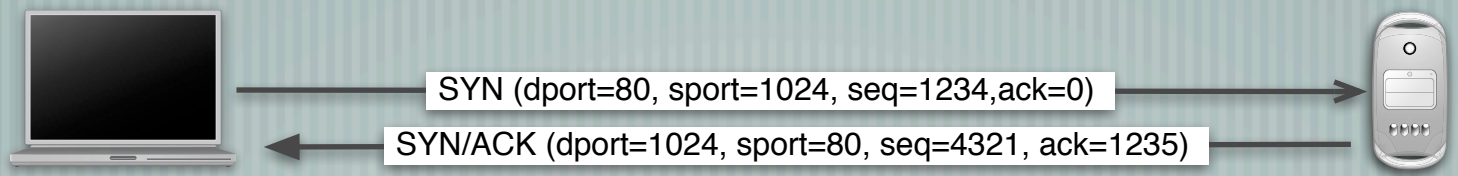
1. Injecter de fausses informations

Injection de fausses données dans le rapport

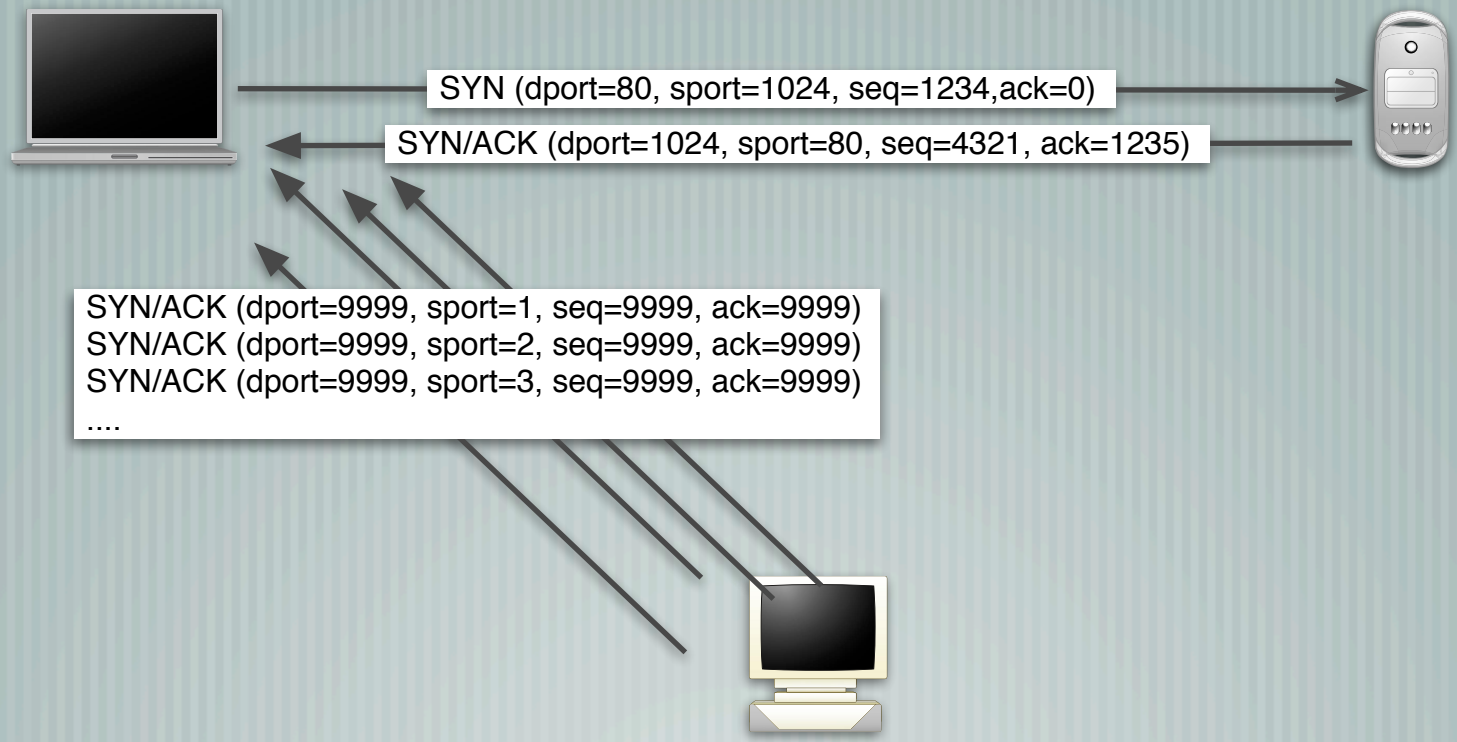
- ippersonality, fausses bannières : mentir sur l'état de sa propre machine
- Solution: ne pas faire entièrement confiance à l'OS detecté, tenter de repérer des incohérences dans les bannieres (dns_fingerprint.nasl, www_fingerprint_hmap.nasl, smtpscan.nasl, etc...)
- Mais rien à faire contre un administrateur particulièrement facétieux

Mentir sur l'état de machines tierces

Fausser le scan de ports



Mentir sur l'état de machines tierces



ex: nmap

```
ip = forge_ip_packet(ip_v : 4,
                     ip_hl : 5,
                     ip_tos : 0,
                     ip_id  : IP_ID,
                     ip_len : LEN,
                     ip_off : 0,
                     ip_p   : IPPROTO_TCP,
                     ip_src : "192.168.10.5",
                     ip_ttl : 0x40);


for ( i = 0 ; i < 1024 ; i ++ )
{
 tcp = forge_tcp_packet(ip:ip, th_sport:i, th_dport:20, th_flags:TH_SYNITH_ACK,
th_seq:rand(), th_ack:0, th_off:5, th_win:512);
 send_packet(tcp, pcap_active:FALSE);
}
```

ex: nmap

```
crashtest:/usr/local/lib/nessus/plugins root# nmap -g 20 -sS 192.168.10.5 -p 1-10

Starting nmap 3.81 ( http://www.insecure.org/nmap/ ) at 2005-06-03 04:28 EDT
Interesting ports on 192.168.10.5:
PORT   STATE SERVICE
1/tcp  open  tcpmux
2/tcp  open  compressnet
3/tcp  open  compressnet
4/tcp  open  unknown
5/tcp  open  rje
6/tcp  open  unknown
7/tcp  open  echo
8/tcp  open  unknown
9/tcp  open  discard
10/tcp open  unknown
MAC Address: 00:0A:95:F2:6D:61 (Apple Computer)

Nmap finished: 1 IP address (1 host up) scanned in 0.982 seconds
```

Solution

- Utiliser un TCP connect() plutot qu'un SYN scan
- Ou utiliser un scanner de ports TCP qui garde une vraie table d'états

Mentir sur la présence de machines tierces

- Avec honeyd, labrea, etc...
- Pas de solution 100% efficace pour se prémunir contre ce problème
- Possibilité de fingerprinter labrea, honeyd, mais un pirate peut toujours modifier les services pour échapper au fingerprint

Empecher l'audit

Ralentir le scanner

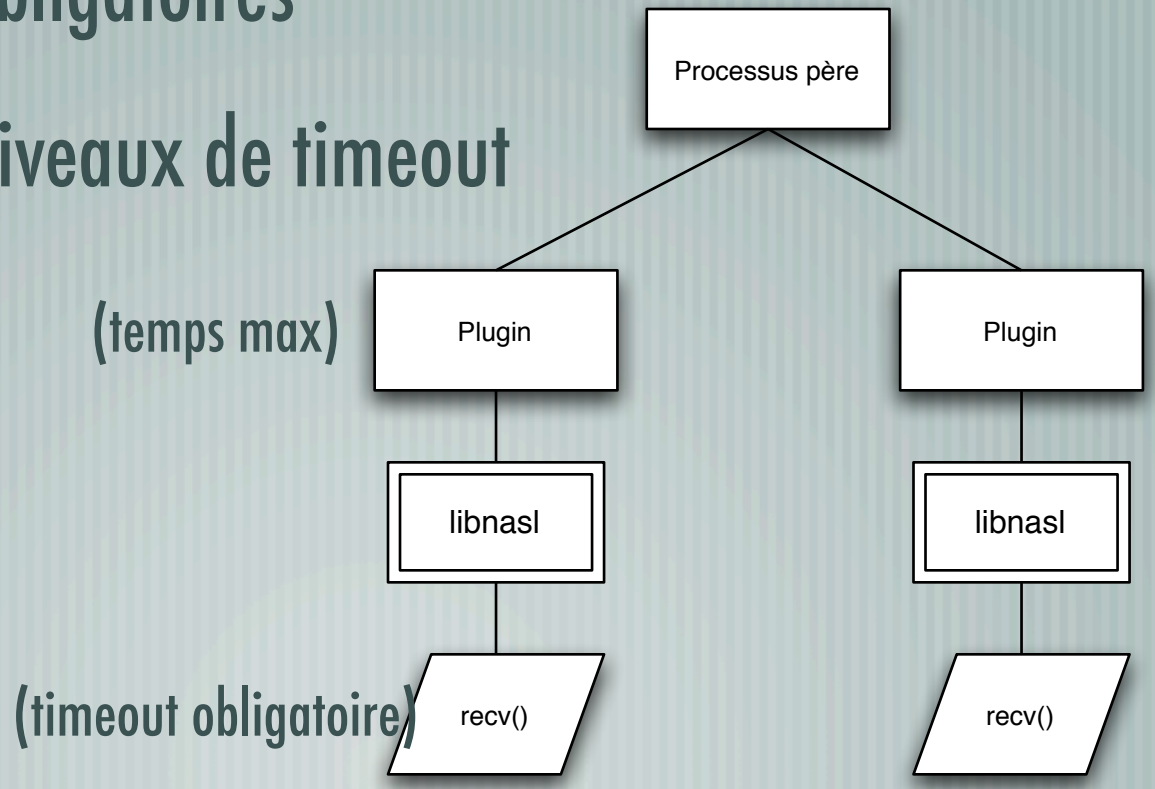
- Conséquence de l'injection de fausses machines (LaBrea/ Honeyd)
- Augmenter artificiellement le nombre de services sur sa machine (ie: mettre le service discard sur les ports 1 à 65535)

Bloquer le scanner

- Par défaut, la fonction recv() est bloquante: elle attend indéfiniment des données sur une socket
- En ne respectant pas les protocoles ou en les faisant mentir, alors il est possible de bloquer un scan entièrement
- Ex: “la revanche des imprimantes HP”

Bloquer le scanner

- Solution:
  - Timeouts obligatoires
  - Plusieurs niveaux de timeout



Faire planter le scanner

- En faisant délibérement mentir certains protocoles
- Exemple: HTTP

Faire planter le scanner

```
[renaud@myth Home]$ cat http_fake_reply
HTTP/1.1 200 OK
Date: Tue, Apr 19 2005 16:10:29 GMT
Server: Apache/1.3.33
Connection: Close
Content-Length: -1



[renaud@myth Home]$ nc -l -p 8080 < http_fake_reply
```

Faire planter le scanner (subtile variante)

```
[renaud@myth Home]$ cat http_fake_reply
HTTP/1.1 200 OK
Date: Tue, Apr 19 2005 16:10:29 GMT
Server: Apache/1.3.33
Connection: Close
Content-Length: 4147483643
```

Le scanner va-t-il tenter d'allouer 4Go de RAM ?

Forcer l'OS à consommer de la bande passante

- Les scanners sous Windows utilisent les fonctions SMB de Microsoft
- En écrivant un faux serveur SMB sur les ports 139 / 445 envoyant en continu \xff\xff\xff\xff, Windows va lire une quantité infinie de données sur la socket - même après le scan!

Solutions (coté Nessus)

- Usage de setrlimit() pour forcer l'OS à tuer les plugins tentant de consommer trop de mémoire
- Compteurs et limites en dur dans les plugins lisant une taille donnée
- Eviter l'usage du C/C++ et de bibliothèques externes lors de l'implémentation de protocoles, mais préférer un langage de script (NASL) à la place (demo)

Obtenir logins et mots de passes

Logins et mots de passe

- Il est possible de donner les clés du réseau à un scanner (login/mot de passe de l'administrateur du domaine, mot de passe SSH root, POP, IMAP, FTP, etc...)
- Un scanner se connectant à chaque machine du réseau, il revient à broadcaster les mots de passes, soit en clair (FTP), soit de manière chiffrée (SSH), soit hachée (SMB)

SSH

- Vulnérable à du man-in-the-middle
- Garder la clé publique de chaque machine scannée n'est pas réalisable à grande échelle (DHCP, nombre de machines, etc...)
- Un pirate n'a qu'a modifier son serveur SSH afin d'enregistrer toutes les tentatives de logins accompagnées des mots de passe
- Solution : utiliser une clé publique à la place. Problème de facilité d'utilisation vs. niveau de sécurité

SMB

- En écrivant un faux serveur SMB, il est possible de forcer le scanner à utiliser le niveau de hash le plus faible (ex: LanManv1)
- Si le scanner utilise Samba, la configuration par défaut pose encore plus de problème...

SMB: Samba

```
 if (!(cli->sec_mode & NEGOTIATE_SECURITY_USER_LEVEL))
        passlen = 0;

 if (passlen > 0 && (cli->sec_mode & NEGOTIATE_SECURITY_CHALLENGE_RESPONSE) && pa
        /* Encrypted mode needed, and non encrypted password supplied. */
        lm_response = data_blob(NULL, 24);
        if (!SMBencrypt(pass, cli->secblob.data,(uchar *)lm_response.data)) {
                DEBUG(1, ("Password is > 14 chars in length, and is therefore in
tication\n"));
                return False;
        }
 } else if ((cli->sec_mode & NEGOTIATE_SECURITY_CHALLENGE_RESPONSE) && passlen ==
        /* Encrypted mode needed, and encrypted password supplied. */
        lm_response = data_blob(pass, passlen);
 } else if (passlen > 0) {
        /* Plaintext mode needed, assume plaintext supplied. */
        passlen = clistr_push(cli, pword, pass, sizeof(pword), STR_TERMINATE);
        lm_response = data_blob(pass, passlen);
 }
```

SMB: solutions

- Samba: éditer smb.conf:
  - client plaintext auth = no
  - client lanman auth = no
- Windows: Empecher l'usage de Lanman (politique de sécurité)
- Nessus: option "Only use NTLMv2" dans les prefs

Telnet/IMAP/POP/FTP...

- Mots de passes envoyés en texte clair
- Ne pas les fournir au scanner lors d'un scan de réseau (les donner lors d'un scan de machine unique, voire pas du tout)

Execution de code

Execution de code

- Principe: utiliser des overflows dans l'implémentation du client pour executer du code arbitraire sur la machine
- Un pirate peut aisément gagner soit le privilège SYSTEM soit administrateur du domaine sur la machine qui fait le scan

Ex: MS05-011

- http://www.microsoft.com/technet/security/bulletin/ms05-011.mspx
- Faille dans la partie cliente SMB: la lecture d'un dossier contenant des informations mal formées permet de provoquer un overflow
- Un pirate n'a qu'a modifier Samba pour mentir sur le contenu de C$ et attendre qu'un scanner utilisant l'API Windows ne vienne se connecter

SunRPC

- Test à faire à la maison:
  - Ecrire un programme envoyant \xff\xff\xff\xff (...) sur le port 2049/udp
  - Le scanner avec certains outils du marché
  - Certains problèmes peuvent être dus à une implémentation ad-hoc et rapide du protocole SunRPC, en C (ou C++)

Solutions

- Utiliser un langage interprété plutot que du code C directement
- Minimisation (et factorisation) du nombre de failles possibles
- Ecrire les plugins dans l'optique d'un langage non sécurisé (tests de longueur, etc...)

NASL

NASL

- Langage de script de Nessus, volontairement limité dans ses fonctionalités
- Scripts communiquant entre eux à travers une Knowledge Base (KB) contenant les faits à propos de la machine distante
- “Sandbox” : les scripts génériques ne peuvent pas executer de commandes locale
- Scripts signés avec plus de privilèges (execution de commandes, possibilité de lire la partie “secrète” de la KB)

NASL

- Les scripts NASL ont un temps d'execution maximum (5mn), sauf exception (nmap.nasl)
- Espace d'execution maximum (20Mo)
- Controle des informations renvoyées au serveur (et donc client)

NASL

- Sandbox: un script ne peut qu'envoyer des paquets contre "la" cible du moment - impossible de remonter des infos à un serveur tiers
- "Invulnérable" aux dépassements de buffers, aux mauvais accès en mémoire, aux chaines de formatage
- Permet de factoriser toutes les fonctions utilisées par les 8,000 plugins dans un seul endroit

NASL

- Les script signés par Nessus.org ont plus de privilèges:
  - Peuvent lire la clé SSH utilisée pour se logguer dans la machine distante (logins et mot de passes windows disponible à tous les scripts pour le moment)
  - Peuvent executer des commandes locales
  - Peuvent configurer Nessus pour se connecter sur une machine tierce (kerberos)

NASL

Système d'include permettant d'implémenter un grand nombre de protocoles :

HTTP
SMB
FTP
SMTP
POP3
IMAP
Kerberos
SSH
TFTP
Telnet

NASL

- Système d'include pour des "meta"-API
- Ex: smb_hotfixes.inc

```
include("smb_hotfixes.inc");
if ( hotfix_check_sp(win2k:5) <= 0 ) exit(0);
if ( hotfix_missing(name:"894320") > 0 )
  security_hole(0);
```

Conclusion

Conclusion

- Lors de la détection d'une machine suspecte, la scanner immédiatement n'est pas nécéssairement une bonne idée
- Donner des privilèges à un scanner n'est pas un acte anodin
- Evaluer le design de sécurité du scanner et autres outils d'analyser (Ethereal...) avant de les utiliser sur un réseau
- Nessus pas le seul scanner à utiliser un langage de script (SAINT, FoundStone...)
- Questions ?