

Prise en charge des logiciels malveillants

Mme Alix

## Qu'est-ce qu'un logiciel malveillant ?

- Un logiciel malveillant (logiciel malveillant) est tout programme qui va à l'encontre des intérêts du système. utilisateur ou propriétaire.
- Question : Un programme qui espionne les habitudes de navigation Web des employés d'une entreprise est-il considéré comme un malware ?
- Et si le PDG autorisait l'installation du programme d'espionnage ?

## Utilisations des logiciels malveillants

- Pourquoi les gens développent-ils et déploient-ils un logiciel malveillant ?
  - Gain financier
  - Pulsions psychologiques et désirs enfantins de « Battre le système ».
  - Accéder aux données privées
  - ...

# Objectifs typiques des logiciels malveillants

- Accès par porte dérobée :
  - L'attaquant obtient un accès illimité à la machine. •

## Attaques par déni de service (DoS) :

- Infecter un grand nombre de machines pour tenter simultanément de connectez-vous à un serveur cible dans l'espoir de le submerger et le faisant planter.

- Vandalisme : –

Par exemple, dégradation d'un

site Web. • Vol de ressources :

- Par exemple, voler les ressources informatiques et réseau d'autres utilisateurs, comme utiliser le réseau sans fil de vos voisins. • Vol

d'informations :

- Par exemple, voler les numéros de carte de crédit d'autres utilisateurs.

## Types de logiciels malveillants

- Virus
- Vers
- Chevaux de Troie •  
Portes dérobées
- Code mobile
- Logiciel publicitaire
- Logiciel collant

# Virus

- Les virus sont des programmes auto-répliquants qui ont généralement une intention malveillante.
- Type de logiciel malveillant à l'ancienne qui a devenu moins populaire depuis la généralisation utilisation d'Internet.
- L'aspect unique des virus informatiques est leur capacité à s'auto-répliquer.
- Cependant, quelqu'un (par exemple, un utilisateur) doit exécuter eux pour qu'ils se propagent.

# Virus (suite)

- Certains virus sont nuisibles (par exemple) :
  - suppriment des informations précieuses du système d'exploitation d'un ordinateur.
  - disque,
  - geler l'ordinateur.
- D'autres virus sont inoffensifs (par exemple) :
  - afficher des messages ennuyeux pour attirer l'utilisateur attention,
  - il suffit de se reproduire.

# Virus : fonctionnement

- Les virus s'attachent généralement aux fichiers de programme exécutables – par exemple, les fichiers .exe dans MS Windows.
- Ensuite, le virus se duplique lentement dans de nombreux fichiers exécutables sur le système infecté.
- Les virus nécessitent une intervention humaine pour se répliquer.



# Origine du terme virus informatique

- Le terme virus informatique a été utilisé pour la première fois publication académique de Fred Cohen dans son 1984 papier Expériences avec des virus informatiques. •

Cependant, un roman de science-fiction du milieu des années 1970 de David Gerrold, Quand HARLIE était un, comprend un description d'un programme informatique fictif appelé VIRUS.

- Le roman de John Brunner de 1975, The Shockwave Rider, décrit des programmes appelés ténias qui se propager via un réseau pour supprimer des données. •

Le terme virus informatique apparaît également dans la bande dessinée livre Uncanny X-Men en 1982.

# Les premiers virus informatiques

- Un programme appelé Elk Cloner est considéré comme le premier virus informatique à apparaître « dans la nature ». Écrit en 1982 par Rich Skrenta, il s'est attaché au système d'exploitation Apple DOS 3.3 et s'est diffusé sur disquette.
- Le premier virus PC était un virus du secteur de démarrage appelé (c)Brain, créée en 1986 par deux frères, Basit et Amjad Farooq Alvi, opérant à Lahore, au Pakistan.

# Vers

- Les vers sont des programmes malveillants qui utilisent Internet se propager.
- Semblable à un virus, un ver s'auto-reproduit. •

Contrairement à un virus, un ver n'a pas besoin d'être humain intervention à reproduire. •

Les vers ont la capacité de se propager de manière incontrôlée période de temps très brève.

- Presque tous les systèmes informatiques du monde sont connectés au même réseau.

# Vers : opération

- Un ver peut se propager à cause d'un logiciel

exploit de vulnérabilité : –

Tire parti du système d'exploitation ou d'un programme d'application avec vulnérabilités du programme qui lui permettent de se cacher dans un environnement apparemment paquet de données innocent.

- Un ver peut également se propager par courrier électronique.

- Les vers de publipostage analysent la liste de contacts et le courrier de l'utilisateur eux-mêmes à chaque contact figurant sur une telle liste.
- Dans la plupart des cas, l'utilisateur doit ouvrir une pièce jointe pour déclencher la propagation du ver (qui ressemble plus à un virus).

# chevaux de Troie

- Un cheval de Troie est une application apparemment innocente qui contient du code malveillant caché quelque part à l'intérieur.
- Les chevaux de Troie sont souvent des programmes utiles qui ont des effets secondaires imperceptibles, mais néanmoins nuisibles.

# Chevaux de Troie : Opération (1)

- Intégrez un élément malveillant dans un programme par ailleurs inoffensif. • La victime:
  - 1. reçoit le programme infecté, 2. le lance, 3. ignore le fait que le système a été infecté.
- L'application continue de fonctionner normalement pour éliminer tout soupçon.

# Chevaux de Troie : Opération (2)

- Faire croire aux utilisateurs qu'un fichier contenant un programme malveillant est en réalité un fichier innocent tel qu'un clip vidéo ou une image.
- C'est facile à faire sous MS Windows car les types de fichiers sont déterminés par leur extension par opposition à en examinant les en-têtes des fichiers. •

Par

exemple, – « Une superbe image.jpg .exe »

- Le .exe peut ne pas être visible dans le navigateur.
- L'auteur du cheval de Troie peut créer une icône d'image qui est le icône par défaut de MS Windows pour les fichiers .jpg.

# Portes dérobées

- Une porte dérobée est un malware qui crée un canal d'accès secret que l'attaquant peut utiliser pour :
  - se connecter,
  - contrôler, –
  - espionner,
  - ou interagir de toute autre manière avec la victime système.



# Portes dérobées : fonctionnement

- Les portes dérobées peuvent être intégrées dans le réel programmes qui, une fois exécutés, permettent à l'attaquant de se connecter et d'utiliser le système à distance.
- Des portes dérobées peuvent être implantées dans le code source par des développeurs de logiciels malveillants avant la sortie du produit.
  - Il est plus difficile de s'en sortir si le programme est open source.

# Code mobile

- Le code mobile est une classe de programmes inoffensifs qui sont : –  
destinés à être mobiles, –  
destinés à être exécutés sur un grand nombre de systèmes, – non  
destinés à être installés explicitement par les utilisateurs  
finaux. • La plupart des codes mobiles sont conçus pour créer un environnement plus actif  
expérience de navigation sur le Web.
  - Par exemple, applets Java, contrôles ActiveX.

# Code mobile (suite)

- Les scripts Java sont distribués dans les sources forme de code, ce qui les rend faciles à analyser.
- Les composants ActiveX sont conventionnels exécutables contenant de l'IA-32 natif langage machine.
- Les applets Java sont sous forme de bytecode, ce qui les rend faciles à décompiler.

# Code mobile : Fonctionnement

- Les sites Web téléchargent et lancent rapidement un programme sur le système de l'utilisateur final.
- L'utilisateur peut voir un message avertissant d'un programme qui est sur le point d'être installé et lancé.
  - La plupart des utilisateurs cliquent sur OK pour autoriser l'exécution du programme.
  - Ils ne peuvent pas envisager la possibilité qu'un code malveillant soit sur le point d'être téléchargé et exécuté sur leur système.

## Logiciel publicitaire

- Un logiciel publicitaire est un programme qui force les utilisateurs non sollicités publicité auprès des utilisateurs finaux. • Les

logiciels publicitaires sont une nouvelle catégorie de logiciels malveillants.

programmes devenus très populaires. • Les logiciels

publicitaires sont généralement fournis avec des logiciels gratuits qui est financé par les publicités affiché par le programme Adware.

# Adware : Opération (1)

- Le programme rassemble des statistiques sur la fin  
les habitudes de navigation et d'achat de l'utilisateur.
  - Les données peuvent être transférées vers un serveur distant.
- Ensuite, l'Adware utilise les informations pour  
afficher des publicités ciblées jusqu'au bout  
utilisateur.

# Adware : Opération (2)

- Les logiciels publicitaires peuvent être bogués et limiter la diffusion performances de la machine infectée.
  - Par exemple, MS IE peut geler pendant une longue période car une DLL Adware est mal mis en œuvre et n'utilise pas multithread correctement.
- Ironiquement, les buggy Adware battent le but de l'Adware lui-même.

# Logiciel collant

- Le logiciel Sticky implémente des méthodes qui empêchent ou dissuader les utilisateurs de le désinstaller manuellement.
- Une solution simple consiste à ne pas proposer de désinstallation programme.
- Une autre solution sous Windows implique :
  - installer des clés de registre qui demandent à Windows de toujours lancez le malware dès le démarrage du système.
  - Le malware surveille les modifications apportées au registre et remplace leurs clés sont supprimées par l'utilisateur.
  - Le malware utilise deux processus de surveillance mutuelle pour s'assurer que l'utilisateur ne met pas fin au malware avant supprimer les clés.



# Futurs logiciels malveillants

- Les logiciels malveillants actuels ne sont que la pointe de l'iceberg. •

La prochaine génération de logiciels malveillants pourrait prendre contrôle des niveaux bas de l'ordinateur système (par exemple, BIOS, Firmware).

- Le logiciel antidote sera sous le contrôle du

les logiciels malveillants...

- Le vol d'informations précieuses peut également aboutir à le retenir contre rançon.

# Vers voleurs d'informations

- Les logiciels malveillants actuels ne prennent pas avantage considérable de la cryptographie. •

Le chiffrement asymétrique crée de nouvelles possibilités de création de vers voleurs d'informations. • Un ver

crypte des données précieuses sur le système infecté utilisant un système asymétrique chiffrer et conserver les données en rançon.

# Vol d'informations vers : opération

1. Le ver Kleptographic embarque un public clé de cryptage dans son corps.
2. Il commence à chiffrer chaque bit de données précieuses sur le hôte en utilisant la clé publique.
3. Le décryptage des données est impossible sans le Clé privée.
4. L'attaquant fait chanter la victime en exigeant une rançon.
5. L'attaquant échange la clé privée contre la rançon tout en gardant l'anonymat.
  - Théoriquement possible en utilisant des preuves sans connaissance – L'attaquant prouve qu'il possède la clé privée sans l'exposer.

# Logiciel malveillant du BIOS/micrologiciel

- Les programmes antivirus supposent qu'il existe toujours des couche de confiance du système.
  - Des programmes antivirus naïfs analysent le disque dur à la recherche de fichiers infectés à l'aide du service de système de fichiers de haut niveau.
  - Un virus intelligent peut intercepter les appels du système de fichiers et présenter le virus avec de fausses versions (original/non infecté) des fichiers sur le disque.
- Les programmes antivirus sophistiqués se situent à un niveau bas niveau suffisant (dans le noyau du système d'exploitation) pour que les logiciels malveillants ne puissent pas déformer leur vision du système.

# Logiciel malveillant du BIOS/micrologiciel :

## Opérations (1)

- Quel est le malware qui a altéré une couche de niveau extrêmement bas du système ? • La

plupart des processeurs/périphériques matériels exécutent du code de très bas niveau qui implémente chaque instruction du langage assembleur à l'aide d'instructions de bas niveau (micro-ops). • Le

code micro-ops qui s'exécute à l'intérieur du processeur est appelé firmware. • Le

micrologiciel peut être mis à jour à l'aide d'un programme de mise à jour du micrologiciel programme.

# Logiciel malveillant du BIOS/micrologiciel :

## Opérations (2)

- Des micrologiciels malveillants peuvent (en théorie) être inclus dans des logiciels malveillants qui déjouent les programmes antivirus.
- Le matériel sera compromis par le micrologiciel malveillant.
- Pas évident à réaliser en pratique car les fichiers de mise à jour du firmware sont cryptés (clé privée à l'intérieur du processeur).

# Programmes antivirus

- Les programmes antivirus identifient les logiciels malveillants en recherchant signatures uniques dans le code de chaque programme (c'est-à-dire, virus potentiel) sur un ordinateur.
  - Une signature est une séquence unique de code trouvée dans une partie de le programme malveillant. •

Le programme antivirus maintient une version fréquemment mise à jour base de données de signatures de virus.

- L'objectif est que la base de données contienne une signature pour chaque programme malveillant connu. • Les logiciels antivirus bien connus incluent :
  - Symantec (<http://www.symantec.com>)
  - McAfee (<http://www.mcafee.com>)

# Virus polymorphes

- Le polymorphisme est une technique qui contrecarre programmes d'identification basés sur les signatures.
- Les virus polymorphes codent ou chiffrent le code du programme en préservant la sémantique.
- L'idée est de chiffrer le code de manière aléatoire clé et déchiffrez-la au moment de l'exécution.
  - Chaque copie du code est différente en raison du utilisation d'une clé aléatoire.



# Virus polymorphes : Technique de décryptage

- Une technique de décryptage polymorphe les virus emploient implique un « XORing » chacun octet avec une clé aléatoire qui était sauvegardé par le virus parent. •

L'utilisation d'opérations XOR a le avantage supplémentaire que le cryptage et la routine de décryptage sont les mêmes : –  $a \text{ xor } b = c$  –  $c \text{ xor } b = a$

# Virus polymorphes : Faiblesses

- De nombreux programmes antivirus recherchent les signatures de virus dans mémoire.
  - C'est-à-dire après que le virus polymorphe ait été déchiffré. •

Si le code du virus qui effectue le décryptage est statique,  
le code de décryptage peut alors être utilisé comme signature.

- Cette limitation peut être résolue (quelque peu) si le  
le code de décryptage est brouillé (superficiellement) : –  
randomiser l'utilisation des registres,  
– ajouter des no-ops dans le code, ...

# Virus métamorphiques

- Au lieu de chiffrer le corps du programme et d'apporter de légères modifications au moteur de décryptage, modifiez l'intégralité du programme à chaque fois qu'il est répliqué. • Il

est donc extrêmement difficile pour les auteurs d'antivirus d'utiliser des techniques de correspondance de signatures pour identifier les logiciels

malveillants. • Le métamorphisme nécessite un code puissant moteur d'analyse qui doit être intégré au malware.

# Virus métamorphiques : Opération

- Le moteur métamorphique analyse le code et génère une version différente à chaque fois que le programme est dupliqué. •

Le moteur métamorphique effectue une grande variété de transformations sur le malware et sur le moteur lui-même.

- Instruction et randomisation des registres.
- Commande d'instructions
- Conditions d'inversion (négation) –
- Insertion d'instructions « garbage » –
- Réorganisation de l'emplacement de stockage des fonctions

# Chronologie des malwares célèbres (1982-1988) [wikipédia]

- 1982

- Elk Cloner, écrit pour les systèmes Apple II, est crédité de étant le premier virus

informatique. • 1987

- (c)Brain, le premier virus écrit pour les PC.
- SCA, un virus de secteur de démarrage pour Amiga apparaît immédiatement créant une tempête pandémique d'auteurs de virus. Peu de temps après, SCA libère un autre virus, considérablement plus destructeur, le Byte Bandit. •

1988

- Le ver Morris infecte les machines DEC VAX connectées au Internet, et devient le premier ver à se propager à grande échelle.

# Chronologie des logiciels malveillants célèbres (1998-2000) [wikipedia]

- 1998
  - Virus CIH version 1.
- 1999
  - Le ver Melissa est publié, ciblant Microsoft Word et Systèmes basés sur Outlook et créant un trafic réseau considérable.
- 2000
  - Le ver VBS/Loveletter , également connu sous le nom de « Je t'aime » le virus est apparu. En 2004, il s'agissait du virus le plus coûteux pour les entreprises, causant plus de 10 milliards de dollars de dégâts.

# Chronologie des malwares célèbres

(2001) [wikipédia]

- Ver Klez .
- Ver Nimda . • Ver

Code Red II (se propage en Chine, attaque Services d'informations Internet de Microsoft.

- Ver Sircam (se propage via les e-mails et partages réseau non protégés).
- Ver Sadmind (se propage en exploitant les trous dans les deux Solaris et MS IIS de Sun Microsystem). • Ver

Raman (semblable au ver Morris infecté uniquement les machines Red Hat Linux exécutant la version 6.2 et 7.0, en utilisant trois vulnérabilités dans wu-ftpd, rpc-statd et lpd.

# Chronologie des malwares célèbres

## (2003) [wikipédia]

- Le ver sobre est vu pour la première fois et maintient sa présence jusqu'en 2005 avec de nombreuses nouvelles variantes.
- Propagation du ver Sobig (techniquement le ver Sobig.F) rapidement via courrier et partages réseau.
- Le ver Blaster également connu sous le nom de ver Lovesan se propager rapidement en exploitant les ordinateurs MS.
- Ver slammer SQL également connu sous le nom de Sapphire. ver, a attaqué des vulnérabilités dans Microsoft SQL Server et MSDE, provoque des problèmes généralisés sur l'Internet.



# Chronologie des malwares célèbres

## (2004) [wikipédia]

- Le ver Sasser émerge en exploitant une vulnérabilité dans LSASS, provoque des problèmes dans les réseaux.
- Le ver Witty est un ver record dans de nombreux salutations.
  - Il a exploité des failles dans plusieurs systèmes de sécurité Internet (ISS) des produits.
  - c'était le premier ver Internet à transporter une charge utile destructrice et il s'est propagé rapidement à l'aide d'une liste pré-remplie d'hôtes Ground Zero.
- MyDoom émerge et détient actuellement le record pour le ver de messagerie de masse qui se propage le plus rapidement.

# Chronologie du célèbre malware (2005)

[wikipedia] • Ver Zotob , dont l'effet a été exagéré car plusieurs médias américains ont été infectés.

# Qu'est-ce que l'ingénierie inverse ?

- L'ingénierie inverse (RE) est le processus de extraire les connaissances ou les plans de conception de tout ce que l'homme a créé. •

La différence entre l'ER et la recherche scientifique est qu'avec l'ER, l'artefact étudié est fabriqué par l'homme. • L'objectif de RE est

d'obtenir les connaissances, les idées et la philosophie de conception manquantes lorsque ces informations ne sont pas disponibles.

# Logiciel inversé

## Ingénierie : inversion

- Inverser consiste à disséquer un programme et en examinant ses entrailles.
- Dans la plupart des industries, les énergies renouvelables sont utilisées pour développer produits concurrents, mais ce n'est pas le cas des industrie du logiciel.
  - L'ER des logiciels est jugée trop complexe pour avoir un sens

financièrement. • Applications courantes de l'ER dans l'industrie du logiciel :

- Sécurité
- Développement de logiciels

# Inversion liée à la sécurité

- Logiciel malveillant
- Algorithmes cryptographiques inversés •
- Gestion des droits numériques (DMR) •
- Fichiers binaires du programme d'audit

# Inversion liée à la sécurité : Logiciel malveillant (malware)

- Propagation de logiciels malveillants (par exemple, virus et vers)  
beaucoup plus rapide maintenant que les ordinateurs sont connectés au L'Internet.
- Le processus d'infection était autrefois lent (par exemple, les virus diffusé par partage de disquette). • Les vers d'aujourd'hui peuvent se propager automatiquement ordinateurs sans aucune intervention humaine. • Les développeurs de logiciels malveillants utilisent RE pour trouver des vulnérabilités dans le système d'exploitation et les logiciels d'application. • Les développeurs de logiciels antivirus utilisent RE pour développer des outils pour protéger les utilisateurs contre les logiciels malveillants.

# Inversion liée à la sécurité :

## Inverser les algorithmes cryptographiques

- Algorithmes restreints, où le cryptage est dans le l'algorithme (par exemple, le chiffre de César) peut être facilement déchiffré en utilisant RE.
- Algorithmes basés sur des clés, où les algorithmes sont publics mais la clé est secrète, elle est plus difficile à briser.
  - Les vulnérabilités incluent l'obtention de la clé (besoin de chance) – essayer toutes les combinaisons possibles jusqu'à ce que vous obteniez la clé (besoin de chance).  
ordinateur quantique)
  - Recherchez une faille dans l'algorithme qui expose la clé du message original (besoin de RE)

# Inversion liée à la sécurité :

## Gestion des droits numériques

- Les fournisseurs de contenu multimédia ont développé ou utilisent technologies (DRM) qui contrôlent la distribution des contenu numérique (par exemple, musique, films).
- Les technologies DMR déterminent si le contenu doit être mis à disposition ou non.
- Les crackers utilisent RE pour tenter de vaincre les DRM les technologies. RE peut aider à : –
  - révéler les secrets de la technologie DRM ; –
  - découvrez les modifications simples qui peuvent être apportées au DRM technologies pour désactiver la protection qu'elles offrent.



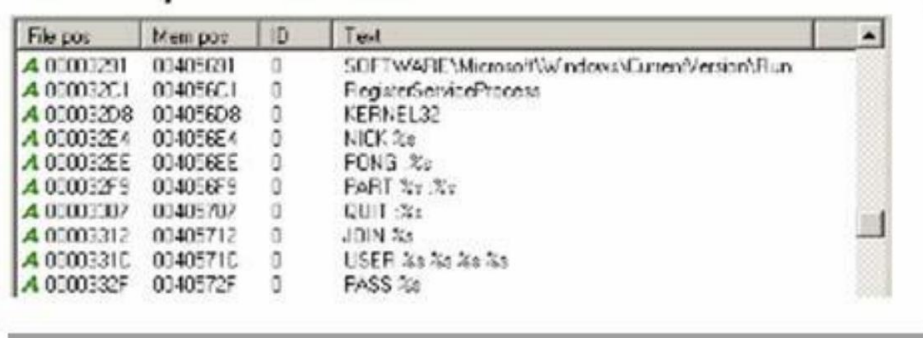
# Inversion liée à la sécurité :

## Fichiers binaires du programme d'audit

- Les logiciels open source semblent plus sûrs à utiliser car ils ont été inspectés et approuvés par des milliers d'ingénieurs logiciels impartiaux.
- RE est une alternative viable (quoique limitée) pour rechercher des failles de sécurité lorsque le code source du programme n'est pas disponible.

# Outils

- VMware
  - Isoler et restaurer des instantanés
- BinTexte
  - Extrait les chaînes des fichiers binaires (code)
  - Commandes IRC, SMTP, clés de registre



The screenshot shows the BinTexte application window with a table of extracted strings. The table has four columns: File pos, Mem pos, ID, and Text. The text column contains various strings, including file paths, process names, and IRC commands.

File pos	Mem pos	ID	Text
00003201	00405601	0	SOFTWARE\Microsoft\Windows\CurrentVersion\Run
00003201	00405601	0	RegisteredServiceProcess
00003208	00405608	0	KERNEL32
000032E4	004056E4	0	NICK %s
000032EE	004056EE	0	PONG %s
000032F9	004056F9	0	PART %s :%s
00003307	00405707	0	QUIT :%s
00003312	00405712	0	JOIN %s
0000331C	0040571C	0	USER %s %s %s %s
0000332F	0040572F	0	PASS %s

# Outils

- IDA Pro
- Désassemble les exécutables en assemblée



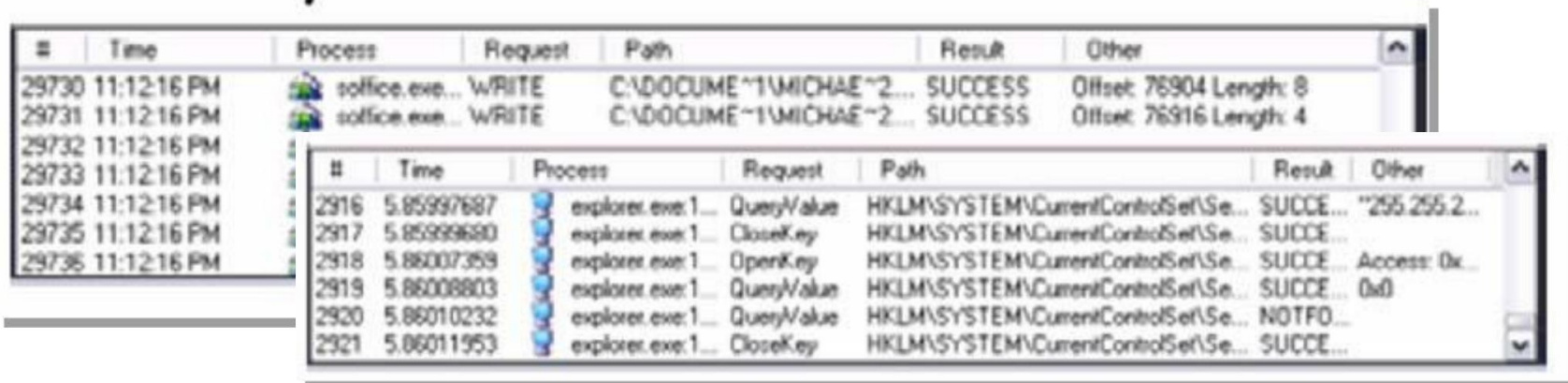
```
.text:004014D1  push    0                ; hTemplateFile
.text:004014D3  push    80h              ; dwFlagsAndAttributes
.text:004014D8  push    3                ; dwCreationDisposition
.text:004014DA  push    0                ; lpSecurityAttributes
.text:004014DC  push    1                ; dwShareMode
.text:004014DE  push    00000000h        ; dwDesiredAccess
.text:004014E3  mov     eax, [ebp+arg_4]
.text:004014E6  push    dword ptr [eax] ; lpFileName
.text:004014E8  call    CreateFileA
.text:004014ED  mov     edi, eax
```

# Outils

- Décompression UPX
  - Packeur exécutable
  - Déballer:  
`upx.exe -do dest.exesource.exe`

# Outils

- SysInternals.com
  - FileMon - surveille l'accès aux fichiers
  - RegMon - surveille l'accès au registre



The image shows two overlapping windows from the Sysinternals suite. The top window is FileMon, displaying file system activity. The bottom window is RegMon, displaying registry access activity.

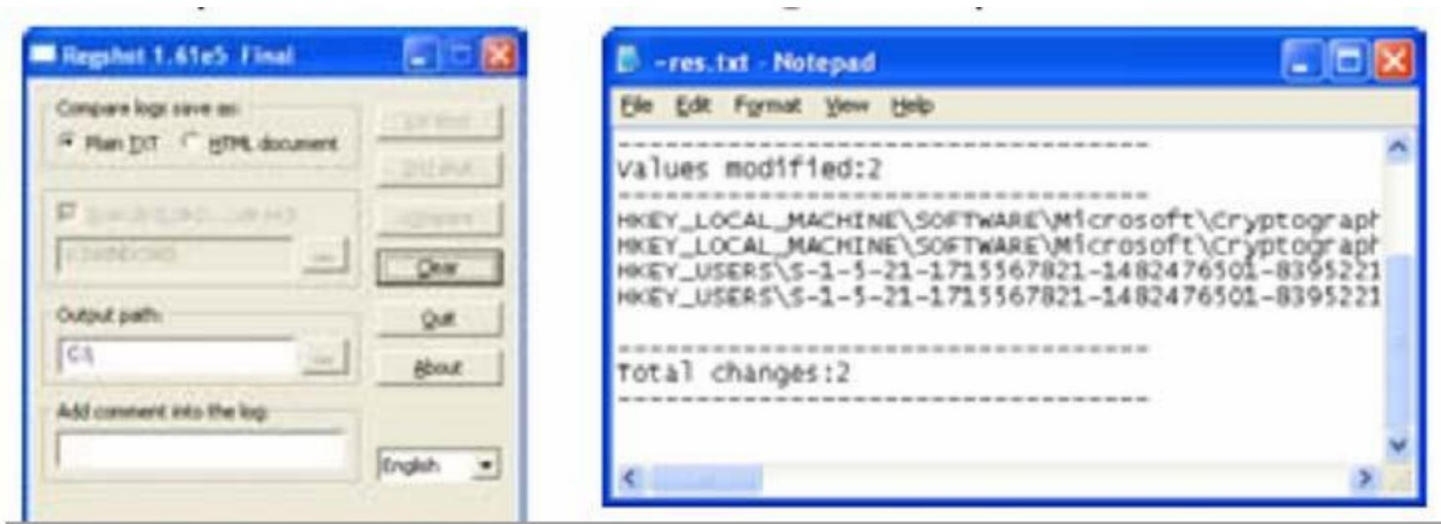
#	Time	Process	Request	Path	Result	Other
29730	11:12:16 PM	solrice.exe...	WRITE	C:\DOCUME~1\MICHAEL~2...	SUCCESS	Offset: 76904 Length: 8
29731	11:12:16 PM	solrice.exe...	WRITE	C:\DOCUME~1\MICHAEL~2...	SUCCESS	Offset: 76916 Length: 4
29732	11:12:16 PM					
29733	11:12:16 PM					
29734	11:12:16 PM					
29735	11:12:16 PM					
29736	11:12:16 PM					

#	Time	Process	Request	Path	Result	Other
2916	5.85997687	explorer.exe:1...	QueryValue	HKLM\SYSTEM\CurrentControlSet\Se...	SUCCE...	"255.255.2...
2917	5.85999680	explorer.exe:1...	CloseKey	HKLM\SYSTEM\CurrentControlSet\Se...	SUCCE...	
2918	5.86007359	explorer.exe:1...	OpenKey	HKLM\SYSTEM\CurrentControlSet\Se...	SUCCE...	Access: 0x...
2919	5.86008803	explorer.exe:1...	QueryValue	HKLM\SYSTEM\CurrentControlSet\Se...	SUCCE...	0x0
2920	5.86010232	explorer.exe:1...	QueryValue	HKLM\SYSTEM\CurrentControlSet\Se...	NOTFO...	
2921	5.86011953	explorer.exe:1...	CloseKey	HKLM\SYSTEM\CurrentControlSet\Se...	SUCCE...	

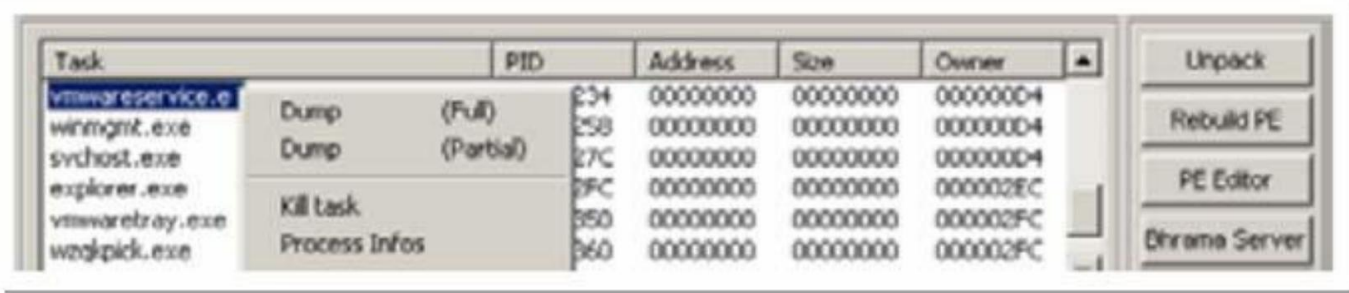
# Outils

- RegShot
- Enregistre les modifications au registre, mais ne lit pas



# Outils

- ProcDump
  - Vide le code d'un processus de la mémoire
  - Utile pour détecter une analyse virus polymorphes



# Outils

- OllyDbg
  - S'attache à un processus
  - Peut manipuler activement la mémoire et s'enregistre pendant le fonctionnement
  - Couteau suisse

The screenshot shows the OllyDbg interface with assembly code on the left and a function call on the right. The assembly code is for a function named `CTOS_44404000`. The highlighted instruction is `CALL [JMP, &KERNEL32.CreateFileA]`. The right pane shows the arguments for the `CreateFileA` function:

```

hTemplateFile = NULL
Attributes = 0x0000
Mode = OPEN_EXISTING
pSecurity = NULL
ShareMode = FILE_SHARE_READ
Access = GENERIC_READ

FileName
CreateFileA

pFileSizeHigh = NULL
hFile
GetFileSize
Origin = FILE_BEGIN
OffsetHi = NULL
  
```



# Outils

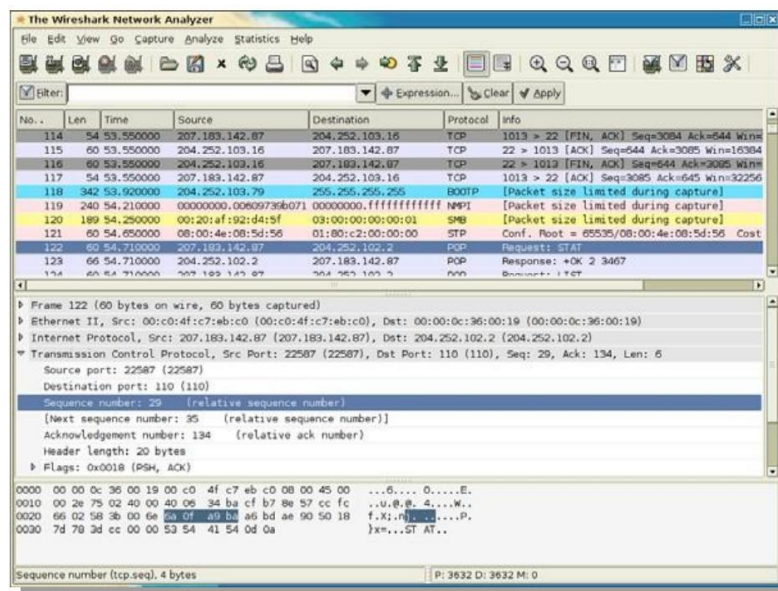
- **Activité réseau**
  - TCPView - affiche le réseau ouvert ports
  - TDIMon - surveille l'activité du réseau
  - Ethereal/Wireshark - Renifleur de paquets
  - Snort - IDS / Renifleur de paquets
  - netcat - Le couteau suisse du réseau

# Outils

- **SysInternals.com**
  - TCPView – Points de terminaison TCP et UDP et processus
  - TDIMon - Enregistre toute l'activité du réseau, mais pas le contenu du paquet

# Outils

- Wireshark (anciennement Ethereal)
- Capture et affiche tous les paquets  
Contenu



# Outils

- Netcat - lit et écrit connexions de données via TCP/IP
- Idéal pour sonder, écouter, débogage ou exploration d'un inconnu comportement du réseau



```
C:\WINDOWS\System32\cmd.exe
C:\>nc.exe -h
[1.11 NT www.vulnwatch.org/netcat/]
connect to somewhere: nc [-options] hostname [ports] [ports] ...
listen for inbound: nc -l [-p port] [options] [hostname] [port]
options:
-d          detach from console, background mode
-e prog     inbound program to exec [dangerous!!]
-g gateway  source-routing hop point[s], up to 8
-G num      source-routing pointer: 4, 8, 12, ...
-h          this cruff
-i secs     delay interval for lines sent, ports scanned
-l          listen mode, for inbound connects
-L          listen harder, re-listen on socket close
-n          numeric-only IP addresses, no DNS
-o file      hex dump of traffic
-p port      local port number
-r          randomize local and remote ports
-s addr      local source address
-t          answer TELNET negotiation
-u          UDP mode
-v          verbose [use twice to be more verbose]
-w secs     timeout for connects and final net reads
-z          zero-I/O mode [used for scanning]
port numbers can be individual or ranges: n-n [inclusive]
C:\>
```

# La tâche

- Beagle.J
- Analyse statique (BinText, IDA)
- Analyse dynamique
  - Côté hôte (registre, processus, fichiers)
  - Mise en réseau (Ports, connexions, trafic)
- Propagation, portes dérobées

# Étude de cas :

logiciel malveillant

Backdoor.Hackarmy.D

- Les prochaines diapositives présentent certains des principaux résultats de l'analyse.

# Porte dérobée.Hackarmy.D : Aperçu

- Backdoor.Hackarmy.D est un cheval de Troie dépourvu de tout mécanisme d'auto-réplication automatisé. • Il est distribué sous forme de fichier image innocent et a une extension .scr (économiseur d'écran). •

Les chevaux de Troie incitent l'utilisateur sans méfiance à ouvrir l'image et activez ainsi le porte arrière.

# Porte dérobée.Hackarmy.D :

## Déballage de l'exécutable

- Un packer exécutable est un programme qui compresse ou chiffre un programme exécutable. • Le programme est automatiquement restauré à l'original état en mémoire une fois le programme lancé. • Certains packers sont conçus comme des outils anti-retour qui chiffrent le programme et tentent de se défendre débogueurs et désassembleurs. •

Certains packers compressent simplement le programme pour diminuer sa taille.

- Backdoor.Hackarmy.D utilise le packer UPX pour diminuer simplement sa taille.



# Porte dérobée.Hackarmy.D : Initialisation

- Lorsque la porte dérobée est lancée, rien ne se passe du point de vue de l'utilisateur.
- Si la porte dérobée était plus intelligente, elle lancerait une application et afficher une image. •

Cependant, si vous vérifiez les processus sur la tâche Manager, vous verrez un processus appelé ZoneLockup.exe.

- Le nom est censé inciter l'utilisateur à réfléchir. que le processus est un élément de sécurité.

# Porte dérobée.Hackarmy.D :

## Un programme de discussion

- Le code assembleur révèle que le numéro de port 6667 est utilisé. • Ce numéro de port est compris entre 6665 et 6669, ce qui est généralement réservé au chat relais Internet (IRC) prestations de service.
- On dirait que le cheval de Troie cherche à discuter avec quelqu'un... l'attaquant très probablement.
- La chaîne USER est intégrée dans l'assembly : – NICK vsorpy USER vsorpy « X.COM » « X » : X
- Cela enregistre un nouvel utilisateur appelé vsorpy sur le Serveur IRC.

# Porte dérobée.Hackarmy.D :

## Communicant

- L'attaquant communique avec la porte dérobée grâce à l'utilisation de paquets de messages privés (PRIVMSG).
1. Trouvez le code pour analyser les commandes de porte dérobée en recherchant la partie du code qui traite les commandes PRIVMSG envoyées depuis le serveur.
  2. Inversez les chaînes de commande (c'est simple).
  3. Inversez les commandes en analysant le code qui suit l'analyse des chaînes de commande.

# Porte dérobée.Hackarmy.D :

## Résumé des commandes (1)

- !?dontuseme :

- Autodétruire le programme en supprimant son registre Autorun entrée et suppression de l'exécutable. • !

- chaussettes4 :

- Transforme le système infecté en serveurs proxy. • !threads :

- Répertorie les threads de serveur actuellement actifs.

- !Info:

- Répertorie les informations générales sur l'hôte infecté (par exemple, nom, Adresse IP, modèle de CPU). • !?

- quit : – Ferme

le processus de porte dérobée sans désinstaller le programme.

- !?disconnect : – Provoque

la déconnexion du programme du serveur IRC, attendez, puis reconnectez-vous.

# Porte dérobée.Hackarmy.D :

## Résumé des commandes (2)

- !execute : –

Exécute un binaire local sur l'hôte. • !delete : –

Supprime un

fichier de l'hôte infecté. • !webfind64 :

- Demande à l'hôte infecté de télécharger un fichier à partir de un serveur distant utilisant http ou ftp.
- !killprocesses !listprocesses :
  - Code inaccessible, peut-être une future fonctionnalité.
  - Les noms suggèrent ce que feront ces fonctionnalités...

# Porte dérobée.Hackarmy.D :

## En savoir plus sur !?dontuseme

- La commande !?dontuseme désinstalle le programme du registre et supprime l'exécutable. • C'est difficile car un fichier programme exécutable ne peut pas être supprimé pendant l'exécution du programme. • Un fichier batch d'autodestruction est généré, qui supprime l'exécutable après que le programme existe. • Le code du fichier batch explique comment procéder. ...

# Backdoor.Hackarmy.D : En savoir plus sur !?dontuseme (rm.bat)

@Écho off

:commencer

s'il n'existe pas, "c:\WINNT\SYSTEM32\ZoneLockup.exe" allez à  
fait

du « c:\WINNT\SYSTEM32\ZoneLockup.exe »

je dois commencer

:fait

del rm.bat

# Porte dérobée.Hackarmy.D :

## En savoir plus sur !socks4

- La commande Backdoor.Hackarmy.D chaussettes4 établit un thread qui attend les connexions qui utilisent le protocole SOCKS4.
- SOCKS4 est un protocole de communication proxy qui peut être utilisé pour accéder indirectement à un réseau.
- En utilisant SOCKS4, on peut acheminer tout le trafic via un serveur unique. •

Permet aux attaquants de se connecter « de manière anonyme » (c'est-à-dire avec l'ID utilisateur de la victime sur l'hôte) aux serveurs sur le L'Internet.

- Difficile de retracer le système à partir duquel le trafic est originaire.



# Modélisation des menaces

- La modélisation des menaces est un processus d'évaluation d'un système logiciel pour les problèmes de sécurité. • Il

s'agit d'une variante du code et de la spécification processus d'inspection discutés plus tôt dans le cours.

- L'objectif est qu'une équipe d'examen recherche fonctionnalités logicielles vulnérables à partir d'un point de vue de la sécurité. •

La modélisation des menaces ne relève pas de la responsabilité d'un testeur de logiciels, bien que les testeurs puissent être impliqué dans l'équipe d'examen de sécurité.

# Processus de modélisation des menaces (1)

- Rassembler l'équipe de modélisation des menaces
  - Inclure des experts et des consultants en sécurité •

Identifier les actifs – Par

exemple, les numéros de carte de crédit, les numéros de sécurité sociale, les ressources informatiques, les secrets commerciaux,

les données financières • Créer une vue d'ensemble de l'architecture

- Définir l'architecture et identifier les limites de confiance et les mécanismes d'authentification

- Décomposer l'application

- Par exemple, identifier les flux de données, les processus de chiffrement, flux de mots de passe.

# Processus de modélisation des menaces (2)

- Identifier les menaces

- Par exemple, les données peuvent-elles être visualisées, modifiées ? Limiter l'accès à utilisateurs légitimes ? Accès non autorisé au système ?

- Documenter les menaces

- Par exemple, décrire la menace, la cible, la forme d'attaque, la contre-attaque. mesures pour prévenir une attaque, etc.

- Classer les menaces (échelle : faible, moyenne, élevée)

- Potentiel de dommages

- Par exemple, propriété, intégrité des données, perte financière

- Reproductibilité

- Par exemple, la probabilité qu'une tentative de compromettre le système réussisse –

## Exploitable/Découvertabilité

- Par exemple, est-il difficile de pirater le système ?

- Utilisateurs concernés

- Combien d'utilisateurs seront concernés ? Qui sont ces utilisateurs ? Sont-ils importants ?