Pratique

Analyse de code malveillant

Hardik Shah, Anthony L. Williams

Degré de difficulté

L'Internet et les réseaux informatiques ont longtemps été infesté par du code malveillant et ses effets néfastes. Cet article vous donnera une introduction sur l'utilisation pratique et basique dans un environnement contrôlé de l'analyse des malwares.

On peut définir le code malveillant comme du *code ayant été développé pour exécuter diverses activités nocives sur un ordinateur.* Des exemples d'une telle activité peuvent être des actions comme : le vol de données utilisateur ou informations personnelles, infectant d'autres machines sur un réseau ou envoyant du Spam au travers des machines infectées. Il y a différents niveaux de code malveillant qui incluent les virus mais ne s'y limitent pas, notamment : vers, Chevaux de Troie et bots. Chacune de ces catégories a des caractéristiques différentes selon leur but prévu. En avançant, notre objectif est de discuter des diverses techniques que nous pouvons employer pour analyser de manière efficace un tel code malveillant.

Types de Code Malveillant

Discutons des définitions de base de quelques uns des divers types de codes malveillants :

- Virus : Les virus sont des programmes simples, qui sont écrits pour changer la manière dont un ordinateur travaille sans l'accord de l'utilisateur. Un virus ne peut infecter d'autres ordinateurs sur un Réseau jusqu'à ce qu'une personne exécute un fichier infecté.
- Cheval de Troie : Dans le milieu du logiciel, un Cheval de Troie est un programme qui, à la différence d'un virus possède ou installe un programme malveillant (parfois nommé payload ou *Trojan*) tout en ayant l'apparence d'être autre chose.
- Ver : Un ver est un programme d'ordinateur auto-réplicant. Il utilise le réseau pour envoyer des copies de lui-même à d'autres nœuds (terminaux d'ordinateurs sur le

Cet article explique ...

- Qu'est-ce que le code malveillant.
- Outils et techniques utilisés pour l'analyse de code malveillant.
- Comment analyser le ver : NetSky-P.

Ce qu'il faut savoir ...

- Techniques élémentaires de débogage binaire.
- Les bases de l'analyse de paquets.
- Environnement Windows.

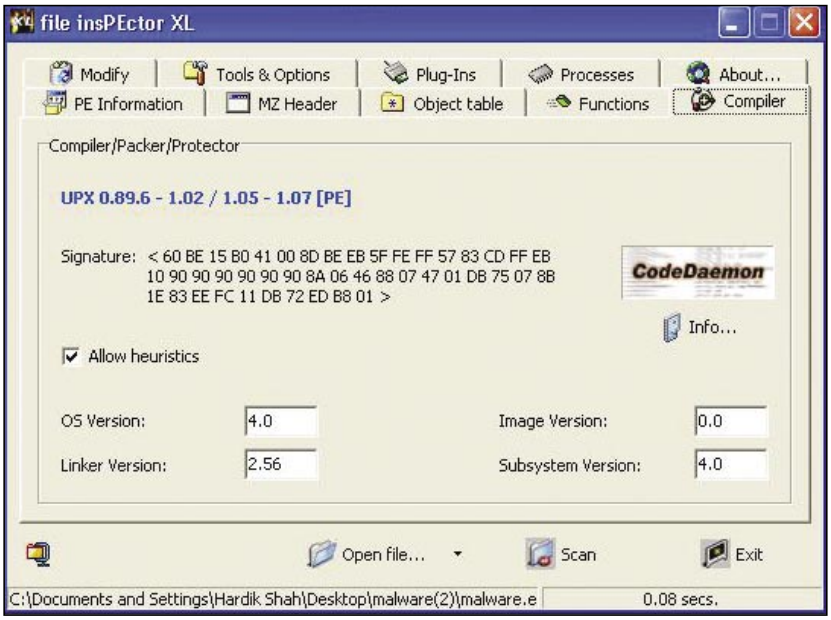

Figure 1. *File insPEctor*

réseau) et peut le faire ainsi sans l'intervention de l'utilisateur.

- Bots : un bot est un programme malveillant qui reçoit des instructions de celui qui le contrôle et effectue des opérations selon ses instructions. De part leur nature les bots se dupliqueront en utilisant différentes techniques comme l'exploitation de systèmes distants, en envoyant des e-mails utilisant le *social engineering* et créant par conséquent un réseau de bots désignés sous le nom de : *Botnets*. Ce réseau d'ordinateurs compromis peuvent être utilisés pour déclencher des attaques par Déni de Service distribué, installer des malwares ou faire d'autres activités néfastes. Les Bots deviennent de plus en plus populaires.

Vulnérabilités

Du code malveillant provenant de vers ou de bots exploite des vulnérabilités au niveau logiciel dans un ordinateur. Ces exploits peuvent avoir comme résultat le vol de données importantes comme des mots de passe et d'informations sur les cartes de crédit et le lancement d'attaques DDoS afin de menacer un individu et de lui extorquer de l'argent. Beaucoup d'auteurs de Botnets fournissent même leurs réseaux pirates de machines zombies pour le louer à d'autres. De tels logiciels impliquent beaucoup de difficultés liées à la sécurité des utilisateurs d'ordinateurs.

Plusieurs organismes ont perdu des millions de dollars en raison de la prolifération de ce type de logiciels au sein de leurs réseaux. Par exemple du code malveillant a détruit l'ensemble des programmes et des sources dans une firme nord américaine. Suite à cela cette entreprise a perdu des millions de dollars, a été déclassée de sa position dans l'industrie et a du licencier plus de 80 employés.

Le besoin de l'analyse

Tout comme l'écriture de code malveillant, il y a une myriade de raisons d'analyser les vers, virus et malwares. La raison principale derrière l'analyse de malwares est qu'il n'y a aucune source disponible pour de tels programmes. Le seul moyen de comprendre ces programmes est de les analyser et déterminer leur fonctionnement interne. Une autre raison pourrait être que beaucoup de chercheurs aiment explorer les fonctionnements cachés d'un programme en l'examinant avec un désassembleur et un débogueur. Il y a 2 techniques majeures pour analyser ce code :

- l'analyse statique (*analyse dead*),
- l'analyse dynamique (*analyse live*).

Nous discuterons de chacune de ces techniques dans les sections suivantes.

Pour cette analyse particulière nous avons choisi le vers : NetSky-P. Il se trouve parmi le TOP 10 des vers rapportés par Sophos Anti-Virus en mai 2007 (*http://www.sophos.com/security/top-10/*).

L'analyse statique (dead)

L'analyse statique est l'approche la plus sûre pour inspecter tout fichier binaire malveillant. En utilisant cette technique d'analyse nous n'exécutons

".data:00...	00000013	C	Re: Encrypted Mail
".data:00...	00000012	C	Re: Extended Mail
".data:00...	0000000B	C	Re: Status
".data:00...	0000000B	C	Re: Notify
".data:00...	00000010	C	Re: SMTP Server
".data:00...	00000010	C	Re: Mail Server
".data:00...	00000014	C	Re: Delivery Server
".data:00...	00000010	C	Re: Bad Request
".data:00...	0000000C	C	Re: Failure
".data:00...	0000001B	C	Re: Thank you for delivery
".data:00...	00000009	C	Re: Test
".data:00...	00000013	C	Re: Administration
".data:00...	00000012	C	Re: Message Error
".data:00...	0000000A	C	Re: Error
".data:00...	00000019	C	Re: Extended Mail System
".data:00...	00000018	C	Re: Secure SMTP Message
".data:00...	0000001B	C	Re: Protected Mail Request
".data:00...	0000001A	C	Re: Protected Mail System
".data:00...	0000001C	C	Re: Protected Mail Delivery

Figure 2. *Objet de l'e-mail*

Listing 1. *La décompression du fichier avec UPX*

```
C:\Documents and Settings\Hardik Shah\Desktop\upx300w\upx300w>upx -d
      malware.exe
Ultimate Packer for eXecutables
Copyright (C) 1996,1997,1998,1999,2000,2001,2002,2003,2004,2005,2006,2007
UPX 3.00w Markus Oberhumer, Laszlo Molnar & John Reiser Apr 27th 2007
File size Ratio Format Name
--------------------- ------ ----------- -----------
28160 <- 6384 58.18% win32/pe malware.exe
Unpacked 1 file.
```

Figure 3. *Messages diffusés*

jamais le programme mais utilisons divers désassembleurs comme : *Win32Dasm* ou IDA Pro pour étudier sans risque le contenu du fichier binaire. Nous utiliserons ces outils pour analyser le vers NetSky-p dans les sections suivantes.

Packers et unpackers

Il y a un format de fichier commun pour les exécutables sur la plate-forme MS Windows, appelé format PE. Chaque fichier exécutable sur un système MS Windows est à ce format de fichier. Habituellement l'auteur du code malveillant utilise différentes techniques pour rendre l'analyse plus compliquée en utilisant des techniques de base.

Une approche commune pour beaucoup d'auteurs de malwares est d'utiliser ce qui est mieux connu comme : des packers exécutables qui réduisent la taille de l'exécutable et changent le contenu en utilisant des algorithmes spécifiques d'obfuscation. Dans ce cas un désassemblage classique ne sera pas efficace. Parmi les packers de fichiers les plus utilisés citons : UPX et AS Pack.

Pour déterminer le packer de fichier nous pouvons utiliser l'outil : File insPector XL.

Comme son nom l'indique il inspectera le fichier à la recherche de signatures de packers communes à partir desquels il pourra facilement détecter le packer utilisé. Il est ensuite nécessaire d'unpacker les fichiers pour la phase d'analyse. Il y a de nombreux outils pour unpacker les fichiers sous environnement protégé. Un tel outil est PEID et un autre est ProcDump. Avec ces outils on peut unpacker la plupart des fichiers de type *packer.*

Parfois les auteurs de malwares rendent plus compliqué la décompression un fichier en particulier en obfuscant ses octets de signature dans l'exécutable, pour que les outils mentionnés ci-dessus ne puissent détecter les bons packers. Pour outrepasser ce problème des outils comme : ProcDump ont une option d'analyse heuristique, qui fera une analyse heuristique du packer employé. Dans certains cas il faut unpacker manuellement le fichier en question. Un unpacking manuel est un autre thème intéressant que nous n'aborderons pas ici par commodité (manque d'espace).

Dans le cadre de cet article on restera avec les outils mentionnés ci-dessus pour l'unpacking.

L'action initiale que nous allons entreprendre est de déterminer si le fichier en cours d'analyse est est de type packer ou non. Pour cela nous utiliserons : File insPEctor XL. Comme vous le voyez à la Figure 1.

Cet outil rapporte que le fichier est packagé en utilisant Ultimate Packer for Executables (UPX). UPX est un outil open source disponible gratuitement à téléchargement depuis *Sourceforge.net*. Après téléchargement et installation il peut être exécuté en ligne de commande avec le nom de fichier de notre malware comme argument générant l'affichage du Listing 1.

Désassembler et identifier des chaînes de données

Un fichier exécutable malveillant peut contenir plusieurs chaînes que le programmeur a intégrées durant la phase de développement. De telles chaînes peuvent apporter des messages d'erreurs ou peuvent être liés au code. Par exemple si un fichier exécutable envoie des mails alors il peut posséder plusieurs types de chaînes pour les différents objets comme : *RE:Voici la pièce*

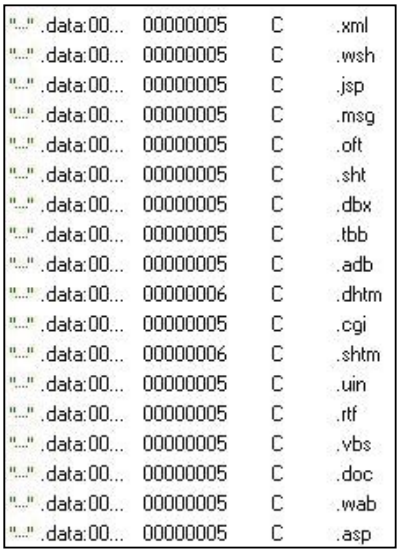

Figure 4. *Extensions de fichiers pièces jointes*

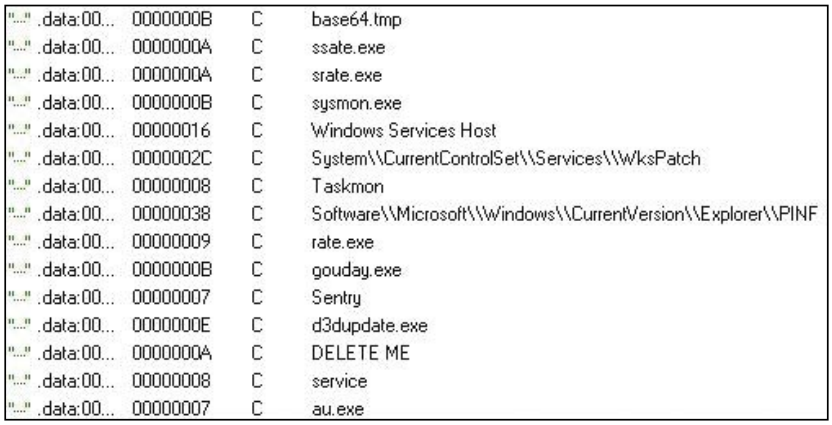

Figure 5. *fichier de contamination*

jointe,++Aucun virus trouvé++ etc. Après avoir unpacké le fichier on doit le désassembler en utilisant un outil comme : *Win32Dasm* ou IDA Pro pour analyser les chaînes de texte usuelles. Cette analyse nous donnera une idée globale du fonctionnement du fichier. Il y a plusieurs chaînes, que l'on peut déterminer par analyse. Ces chaînes peuvent contenir le corps des e-mails, objet, ou nom du fichier joint, qu'un vers envoie en pièce jointe etc.

'...'	.data:00...	00000008	C service
'...'	.data:00...	00000007	C au.exe
'...'	.data:00...	00000009	C msgsvr32
'...'	.data:00...	00000036	C SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\RunServices
'...'	.data:00...	00000008	C system.
'...'	.data:00...	0000003C	C CLSID\\{E6FB5E20-DE35-11CF-9C87-00AA005127ED}\\InProcServer32
'...'	.data:00...	00000009	C Explorer
'...'	.data:00...	0000002E	C SOFTWARE\\Microsoft\\Windows\\CurrentVersion\\Run

Figure 6. *Entrées register utilisées par NetSky-P*

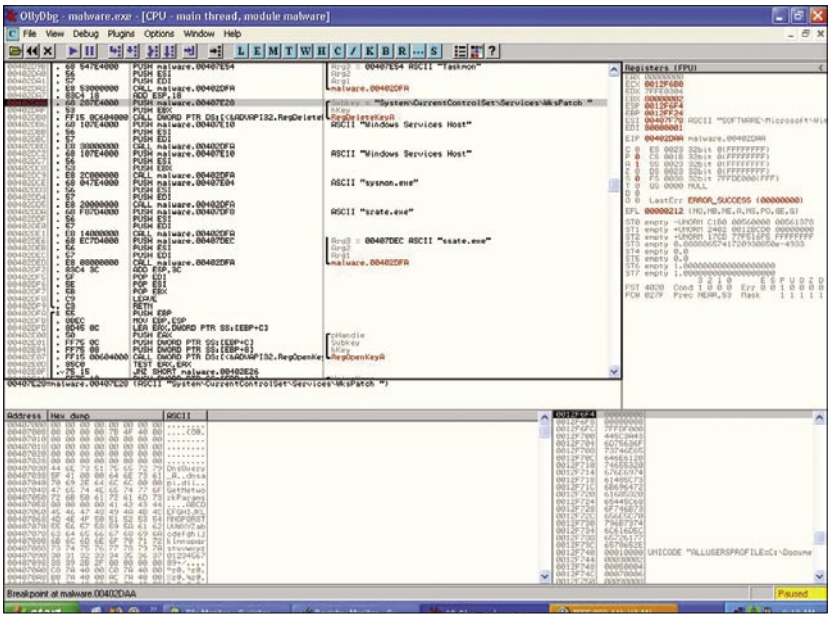

Figure 7. *Illustration breakpoint sur chaînes*

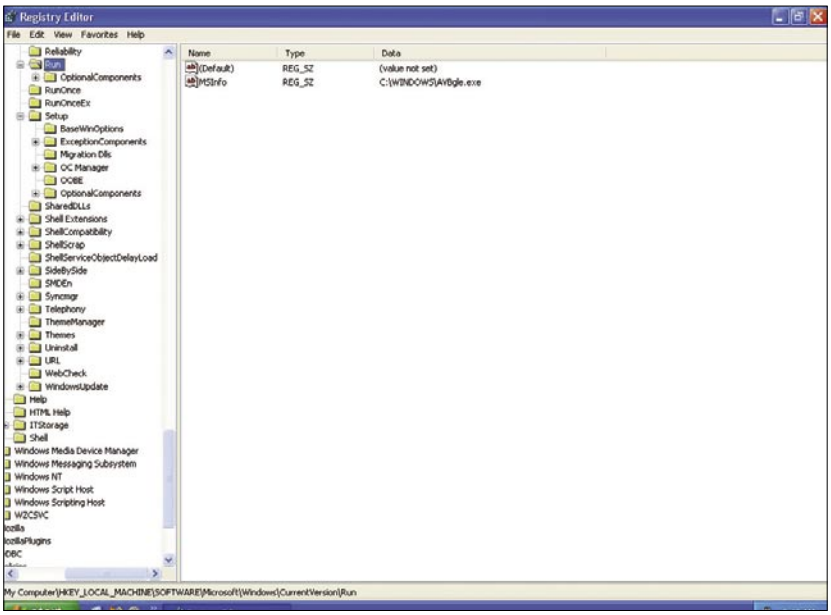

Figure 8. *Entrée Registre crée par le ver*

Maintenant que nous avons unpacké avec succès l'exécutable on peut procéder au désassemblage et effectuer un travail d'investigation approfondi. Effectuons une analyse statique de cet exécutable en utilisant le désassembleur IDA Pro. La première chose que nous rechercherons sont les chaînes. Les chaînes au sein d'un exécutable peuvent fournir un ensemble d'informations telles que : l'objet du mail, message, entrées du registre, extensions du fichier et leur nom.

L'exemple suivant montre l'objet de l'e-mail qu'utilise le vers NetSky-P lorsqu'il envoie des mails depuis la machine infectée (Figure 2).

Objet de l'e-mail

La Figure 3 montre les différentes chaînes qu'il possède pour les messages à diffuser. La Figure 4 affiche les différents types d'extensions de fichiers que le vers Netsky-P insère dans les pièces jointes envoyées.

La Figure 5 montre le nom des fichiers utilisés sur le système infecté. La Figure 6 illustre quelques unes des entrées registres utilisées par le vers. Basé sur les informations collectées jusqu'à présent il est facile d'en déduire que le vers Net-Sky-P envoie des e-mails en utilisant diverses chaînes dans le champs objet de l'e-mail, de noms de fichiers et d'extensions. En plus de tout cela il stocke plusieurs entrées registres afin d'être exécuté à chaque démarrage de l'ordinateur.

Analyse Dynamique

Dans une analyse dynamique (*live*) nous devons vérifier le fonctionnement global et le fonctionnement interne du code actuellement en l'exécutant dans un environnement contrôlé. Ceci nous aide à éliminer les faux-positifs de la phase d'analyse statique.

Des auteurs de malwares incluent de façon intentionnelle plusieurs chaînes et fonctions afin d'empêcher l'analyse précise de leur malware (ou incluent du code pour détecter qu'il fonctionne dans les confins d'une machine virtuelle afin de changer son chemin d'exécution) ; de telles

tentatives d'obfuscation peuvent être détectées pendant l'Analyse Dynamique.

Pour cela on a mis en place 2 environnements de test s'exécutant sous MS Windows XP Professional SP2. Sur la première machine on a installé Ollydbg pour permettre le débogage du vers Net-Sky-P et le deuxième système était connecté au même réseau de sorte que nous puissions surveiller efficacement les diverses activités du vers en temps réel. On a ensuite lancé Wireshark sur les 2 ordinateurs et aussi RegMon, ainsi que FileMon sur le second système infecté.

Il est important de noter que vous devez prendre des précautions en manipulant les malwares afin de les conserver uniquement dans l'environnement de travail. Dans notre cas nous avons choisi un réseau air-gapped sans l'accès à nos réseaux de production ou Internet.

Beaucoup de gens choisissent le populaire VMWare Suite pour procéder à des tests dans les confins d'une machine virtuelle. Il s'agit d'un choix personnel pour l'environnement de test, mais nous vous invitons à en employer un sécurisé. Après avoir préparé *l'environnement,* on a lancé le débogueur OllyDbg et avons localisé le fichier *NetSky.exe*. Après cela nous avons mis le point d'arrêt (*breakpoint*) sur différentes chaînes comme montrées en Figure 7. Dans l'image 7 nous mettons un breakpoint sur la chaîne *System\ Current Control Set\ Services\ WksPatch* et exécutons OllyDebugger.

Il s'est arrêté sur celui-ci. Une examination attentive des chaînes confirme les résultats précédents déterminés pendant la phase statique de l'analyse. Maintenant nous allons remplacer les breakpoints de départ et utiliser les animations on fly et divers autres options de débogage comme : step in (pas à pas), step out (pas à pas aux appels) pour tracer les différents appels aux API de Windows comme : `GetInternetConnectionState()` et `RegCreateKeyEx()`.

De cette analyse on peut déterminer que le ver créé également diverses tâches pour envoyer des e-mails.

Clés de Registre

Pour se répandre, un code malveillant a besoin d'être lancé d'une façon ou d'une autre, soit en exécutant le fichier malveillant ou en cliquant sur un lien web malveillant ou depuis l'option autorun disponible dans le registre Windows. Les malwares modernes utilisent diverses techniques de social engineering afin de manipuler les utilisateurs à non seulement l'exécuter la première fois mais aussi pour l'exécuter à chaque fois que l'ordinateur démarre. Ils créent à cette fin des entrées registre.

Pour analyser un tel comportement on utilisera un outil appelé : RegMon de Sysinternals. Il affichera l'ensemble des entrées registre utilisées par un programme. Pour analyser le vers NetSky-P nous l'avons exécuté et avons vérifié les multiples accès au registre dans les logs de RegMon. Il essaye d'accéder à plusieurs clés mentionnées précédemment.

Un détail que nous avons observé était que le vers créait une nouvelle

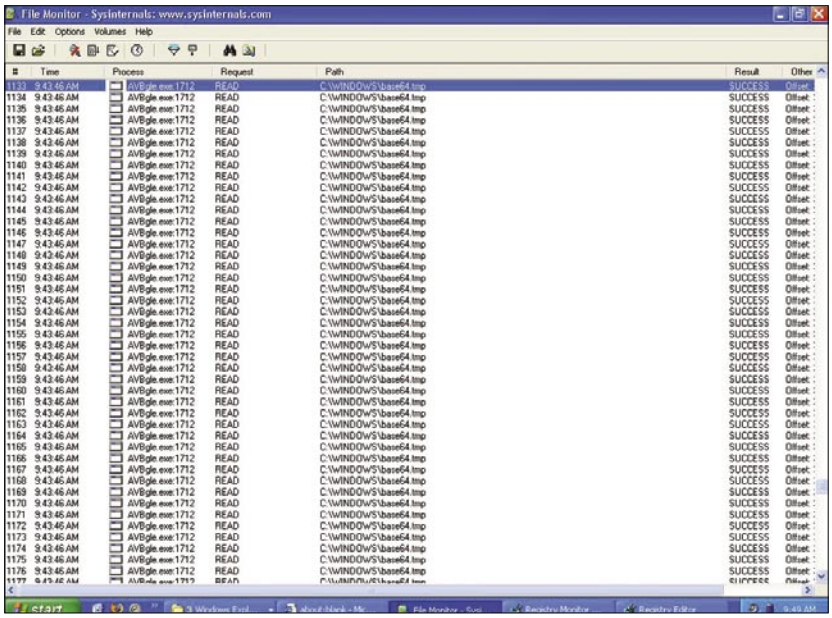

Figure 9. *Base64 FileMon*

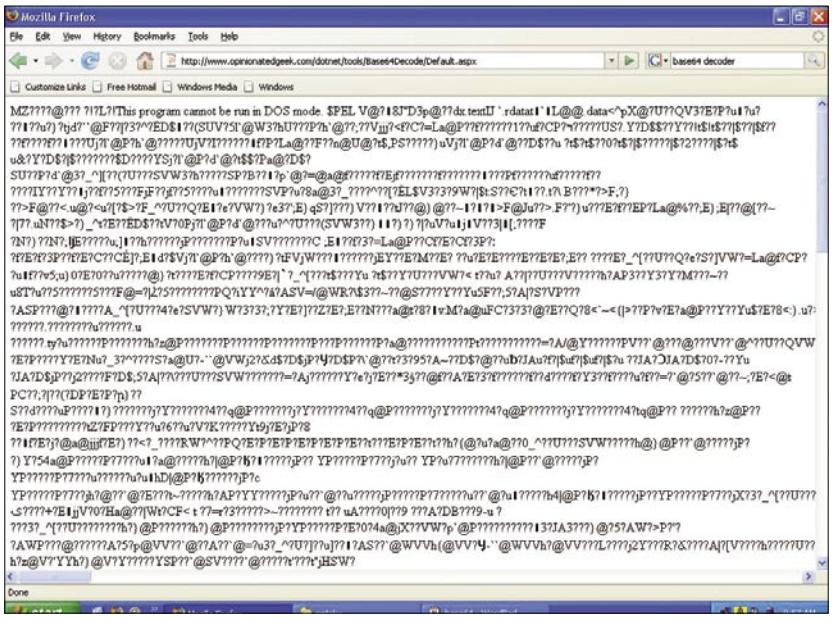

Figure 10. *Fichier décodé*

entrée au registre via *HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows\CurrentVersion\Run* comme affiché à la Figure 8. Nous avons ensuite analysé le dossier Windows et y avons trouvé 2 nouveaux fichiers *AVBgle.exe* et *Base64.tmp*.

FileMon

Du code malveillant peut se modifier ou se répliquer en utilisant différents noms dans plusieurs endroits.

Il peut également télécharger et exécuter d'autres fichiers comme des : backdoors depuis un lieu distant et le placer sur le système infecté. Afin d'étudier ce comportement, on peut utiliser un outil nommé FileMon également disponible chez Sysinternals.

Pour continuer l'analyse nous avons redémarré le système de test infecté et avons lancé RegMon, FileMon et Wireshark de nouveau. Nous avons vérifié les logs de FileMon et le point commun que nous avons trouvé était qu'il accédait continuellement à un fichier nommé *Base64.tmp*. Comme son nom le suggerait, on pouvait supposer que le fichier a été crypté avec l'algorithme *Base64*. Par conséquent nous avons utilisé un décodeur *base64* pour déterminer que notre fichier était bien : NetSky-P.

Base64 FileMon

La figure 10 montre le fichier décodé qui était au format *base64*. En regardant le contenu il est clair qu'il s'agit d'un fichier exécutable étant donné qu'il possède l'en-tête MZ qui est un en-tête standard pour les fichiers exécutables sur la plateforme Windows.

Capture de Paquets et Analyse

La plupart des malwares dans la nature actuellement essayient de contaminer d'autres machines sur le réseau ou bien font partie des botnets et envoient ainsi beaucoup de spam depuis des machines infectées ou bien peuvent également envoyer beaucoup d'informations depuis des systèmes compromis comme les habitudes de navigation des utilisateurs, mots de passe, détails de comptes bancaires etc. Les malwares peuvent également être utilises pour lancer des attaques DDoS sur Internet.

Pour détecter cela on doit utiliser un Sniffer de paquets comme Wireshark qui peut capturer le trafic réseau passant par la machine infectée. Basée sur l'analyse des données capturée, on peut déterminer une variété de détails. Par exemple s'il s'agit d'un botnet : quels sont les instructions de contrôle, depuis quels serveurs il télécharge les fichiers et quel type de spam il envoie.

Nous avons ensuite décidé de sauvegarder le fichier décodé sous : *decoded.exe* et de l'ouvrir avec IDA Pro pour investigation. Depuis notre poste de travail d'analyse, nous avons remarqué que : *AVBgle.exe* scannait le fichier index.dat dans le dossier *Temporary Internet Files* sur le système infecté. Ceci est intéressant et permet de déduire que celui-ci envoyait aléatoirement beaucoup d'e-mails grâce aux adresses mails trouvées dans ce répertoire.

Ce comportement est exhibé dans le paquet de type : dump montré à la Figure 11. Une analyse des paquets plus précise est illustrée à la Figure 12. Dans les circonstances actuelles nous avons décidé d'exécuter une analyse des paquets du ver. Nous avons noté qu'au début il essayait d'exécuter plusieurs requêtes DNS pour des serveurs externes tels que Yahoo!, AOL, et Hotmail.

Après ceci, il envoyait des mails avec divers objets, noms de fichiers, comme présenté précédemment. La Figure 13 en est une illustration.

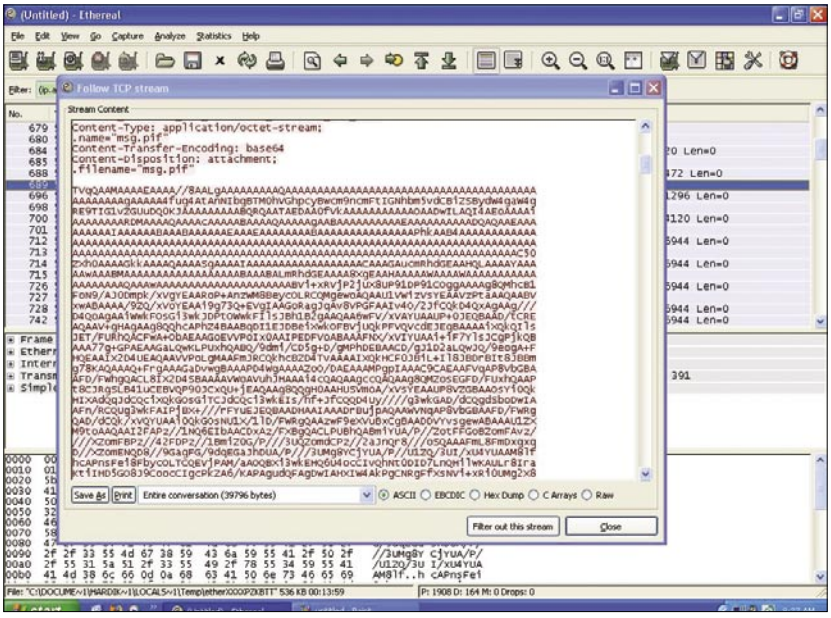

Figure 11. *Paquet Dump*

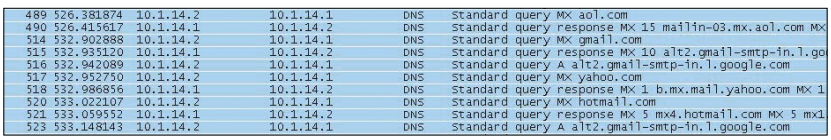

Figure 12. *Analyse des paquets plus précise*

Identifier les Algorithmes de Réplication

Les malwares ne fonctionnent pas dans le vide. Pour s'accroître ils doivent engendrer plusieurs instances du même code, pouvant travailler conjointement sous le contrôle d'une personne pour réaliser des activités nocives. Par conséquent il essaie sans interruption d'infecter (ou réinfecter selon les cas) les autres machines sur réseau local ou sur Internet. Les malwares utilisent une variété de

techniques pour y parvenir. En voici les exemples :

- l'envoi de mails avec une pièce jointe possédant du code malveillant,
- exploiter les logiciels de l'ordinateur en utilisant des vulnérabilités ou failles de type : *0day,*
- exploiter les vulnérabilités du système d'exploitation lui-même.

Afin d'identifier avec exactitude l'algorithme de réplication utilisé on doit exécuter le code malveillant dans un environnement contrôlé et tracer le code au sein d'un débogueur. Pour ce type d'analyses on utilisera Ollydbg pour identifier l'algorithme de réplication.

Dans certains cas il n'est pas possible d'identifier l'algorithme avec le débogueur seul. Dans ces scénarios on doit combiner d'autres techniques telles que la capture de paquets afin que nous puissions déterminer si le malware utilise un exploit connu ou non ou d'autres comportements visibles. Vu l'analyse précédente il est évident que le ver NetSky-P est un mail bomber qui se replique au sein d'un e-mail, en attente d'utilisateurs confiants pour l'ouvrir.

Il utilise plusieurs techniques de social engineering pouvant tromper les utilisateurs débutant en employant des termes comme : *Aucun Virus Trouvé!!* dans le contenu du mail.

Si les utilisateurs ne sont pas conscients de ce type de méfaits alors il est possible d'infecter la machine en question.

Conclusion

Le code malveillant a toujours été une menace pour les utilisateurs. De nos jours, avec Internet, les malwares sont intensément employés pour générer du trafic sur les sites, générer des liens invalides qui conduisent l'utilisateur à des sites infectés, lancer des attaques de Deni de Service (DDoS) pour voler des données personnelles et confidentielles. Ils emploient maintenant une variété de techniques telle que les exploits dit 0days pour permettre leur réplication plus rapidement.

En utilisant ces techniques, on peut analyser le fonctionnement interne du code malveillant. Acquérir de telles compétences demande du temps, de l'intuition, de la patience et du dévouement. On s'aperçoit finalement que cette analyse n'est pas complète, notre but étant de donner une vision globale sur l'utilisation de différents outils pour l'analyse de malwares et des techniques pour analyser le code malveillant d'aujourd'hui. ●

Références et Autres Lectures

- *http://www.smallbiztrends.com/2007/06/top-five-small-business-internet-security-threats.html*
- *http://www.offensivecomputing.net/*
- *http://www.viruslist.com/*
- *http://vx.netlux.org/*
- *http://hexblog.com/*

Outils

- VMWare (Virtualization Software) – *http://www.vmware.com/*
- IDA Pro/Freeware (Dissembler) – *http://www.datarescue.com/*
- Ollydbg (Popular Ring 3 Debugger) – *http://www.ollydbg.de/download.htm*
- UPX (Ultimate Packer for Executables) – *http://upx.sourceforge.net/*
- ImpREC(Import Reconstruction for PE files) – *http://securityxploded.com/download.php#imprec*
- Windows Sysinternals(FileMon,RegMon) – *http://www.microsoft.com/technet/sysinternals/default.mspx*

À propos des auteurs

Hardik Shah se spécialise dans la Sécurité des Réseaux, le Reverse Engineering et l'Analyse de code malveillant. Il s'intéresse à la sécurité web et à celle des applications. On peut le joindre à l'adresse : *hardik05@gmail.com* Anthony L. Williams est Architecte en Sécurité Informatique pour IRON::Guard Security, LLC où il conduit des test d'intrusion (Pen Test), évaluations de vulnérabilités, audits et réponse en cas d'incident. On peut le joindre à l'adresse : *awilliams@ironguard.net*.

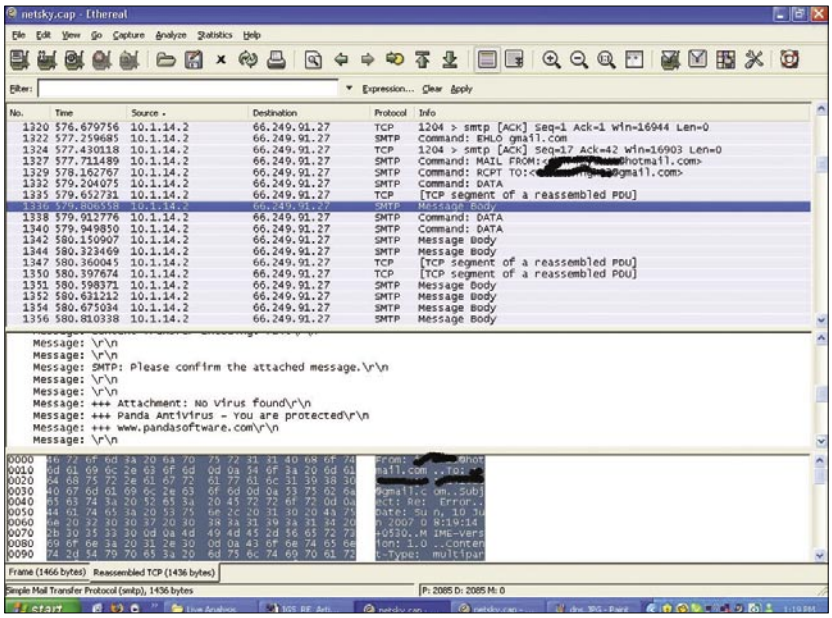

Figure 13. *Action du ver*

ARCHIVES hakin9 2006 !

Avez-vous raté un numéro en 2006 ?
Rien de plus simple

Abonnez-vous au hakin9 et choisissez ARCHIVES 2006 sur CD !

Pour recevoir plus d'informations
visitez notre boutique en ligne

www.buyitpress.com/fr