API Win32 ancestrales pour Chevaux de Troie hyper furtifs

JSSI 2004

Eric DETOISIEN
Eyal DOTAN

Sommaire

- Introduction
- Modèle de Communication
- Injection de Code
- API Hooking
- Démo finale
- Evolutions Possibles
- Prévention
- Conclusion

Introduction

- Aujourd'hui il existe des méthodes plus ou moins fiables pour sécuriser les réseaux des menaces extérieures (firewalls, cryptage, authentification, …)
- Mais le maillon faible des systèmes d'information reste encore et toujours à l'intérieur du réseau, c'est à dire le poste client. Même si l'administrateur peut faire confiance à ses utilisateurs, comment faire confiance aux logiciels qui tournent sur leurs machines ?

Introduction

- Les chevaux de Troie sont des outils d'attaque de plus en plus utilisés. Ils sont redoutables pour ce qui est de l'attaque des ordinateurs des particuliers.
- Mais la plupart des chevaux de Troie sont facilement détectables et inefficaces dans des environnements professionnels.
- Dans cette présentation, nous démontrerons qu'il est parfaitement possible de mettre au point des chevaux de Troie extrêmement furtifs en utilisant des techniques de programmation Win32 connues depuis plus de 10 ans.

Introduction

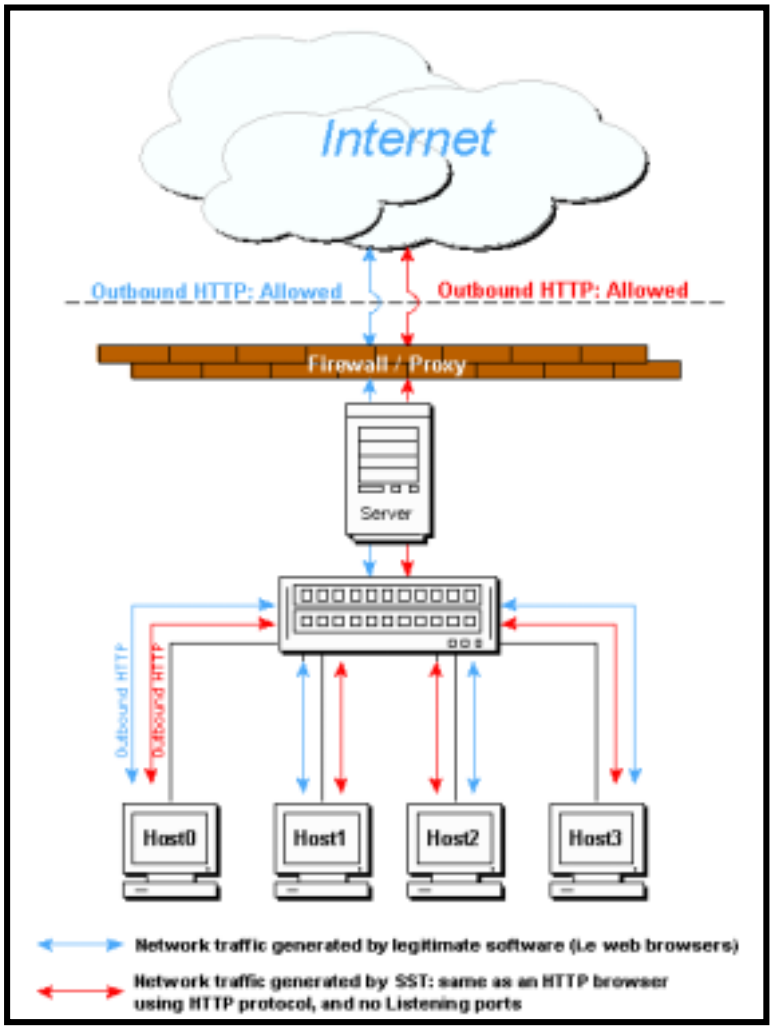
- Nous discuterons également des méthodes de prévention et de l'éventuelle existence de tels programmes d'attaque.
- Le but de cette présentation est de vous aider à mieux connaître vos ennemis et le danger de ce qui va devenir à notre avis la prochaine génération de chevaux de Troie (publics) visant les réseaux d'entreprise.

Modèle de Communication

- La communication est la première raison d'être d'un cheval de Troie.
- Notre cheval de Troie communique avec le monde extérieur via le protocole HTTP. Le cheval de Troie vérifie régulièrement les ordres de l'attaquant en se connectant sur un site WEB maintenu par l'attaquant.

Modèle de Communication

- Les firewalls à l'entrée du réseau voient les requêtes HTTP initiées par le cheval de Troie. Mais ils ne peuvent pas faire la différence avec des requêtes légitimes d'un navigateur WEB.

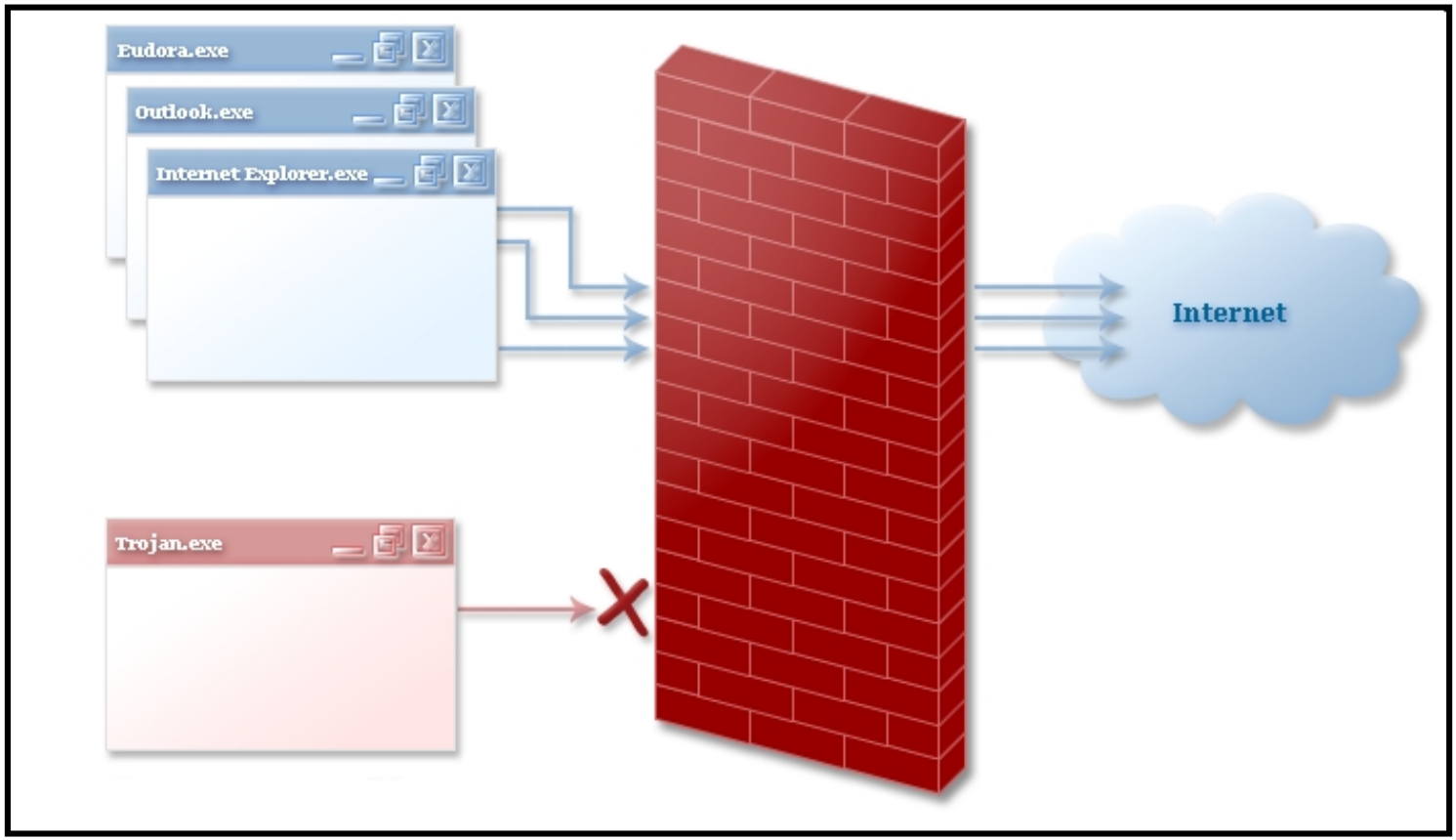


Modèle de Communication

- Ce mécanisme de communication est simple, mais efficace.
- Les tunnels HTTP ne sont pas des techniques nouvelles (cf. HTTPTunnel, Setiri, webdownloader, …), mais les chevaux de Troie fonctionnant ainsi sont plutôt rares. (Pourquoi?)
- Il existe des APIs très simples permettant de générer des requêtes HTTP. Ex: WININET.DLL

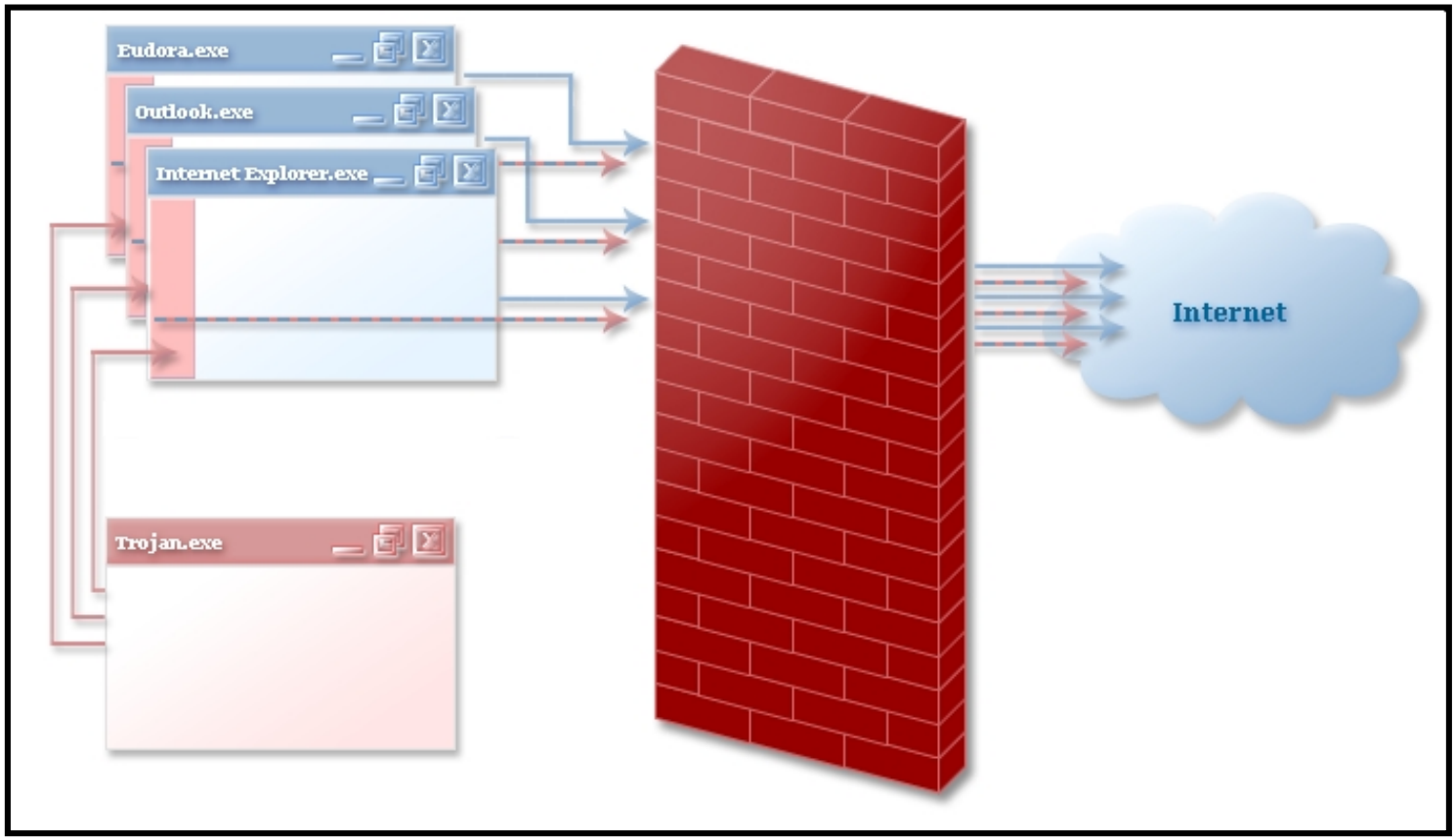
Injection de Code

Rappel sur les Desktop Firewalls :



Injection de Code

Technique de contournement par injection :



Injection de Code

- L’injection de code est une technique connue depuis 10 ans: "Load Your 32-bit DLL into Another Process's Address Space Using INJLIB" - Jeffrey Richter (Mai 1994).
- L’injection de code directe, sans passer par DLL, est plus difficile à programmer, mais plus furtive.
- Mais surtout : l’injection de code ne requiert pas de privilèges particuliers. Tous les processus appartenant à l’utilisateur peuvent être injectés.

Injection de Code

- **APIs d'injection de code:**
  - `OpenProcess` : Retourne un Handle vers le processus cible.
  - `VirtualAllocEx` : Alloue de la mémoire dans le processus cible.
  - `WriteProcessMemory` : Ecrit du code dans la mémoire allouée.
  - `CreateRemoteThread` : Exécute le code copié à partir du processus visé.

Injection de Code

Avantages pour un cheval de Troie

- Contournement de Desktop Firewalls en injectant une application autorisée à se connecter.
- Peut tromper d'autres logiciels de surveillance de comportement.
- Permet au cheval de Troie de modifier le comportement des programmes injectés et d'intercepter ses APIs (cf chapitre suivant).

Injection de Code

Utilisation dans les programmes malveillants

Aujourd'hui il existe plusieurs programmes malveillants utilisant la technique d'injection :

- BackStealth (proof of concept)
- Outils de tests Firewalls
- Optix, Beast et autres chevaux de Troie
- Keyloggers …

Injection de Code

L’avenir de l’injection de code

- « Inject and die »: une fois que le cheval de Troie a injecté son code, il peut se terminer et ainsi disparaître du gestionnaire des tâches.
- Survie du thread injecté: une fois que le cheval de Troie meurt, la survie du thread injecté dépend du processus injecté.
- Solution: injecter tous les processus utilisateur, ainsi que chaque nouveau processus créé.

Injection de Code

Démo:

Injection,

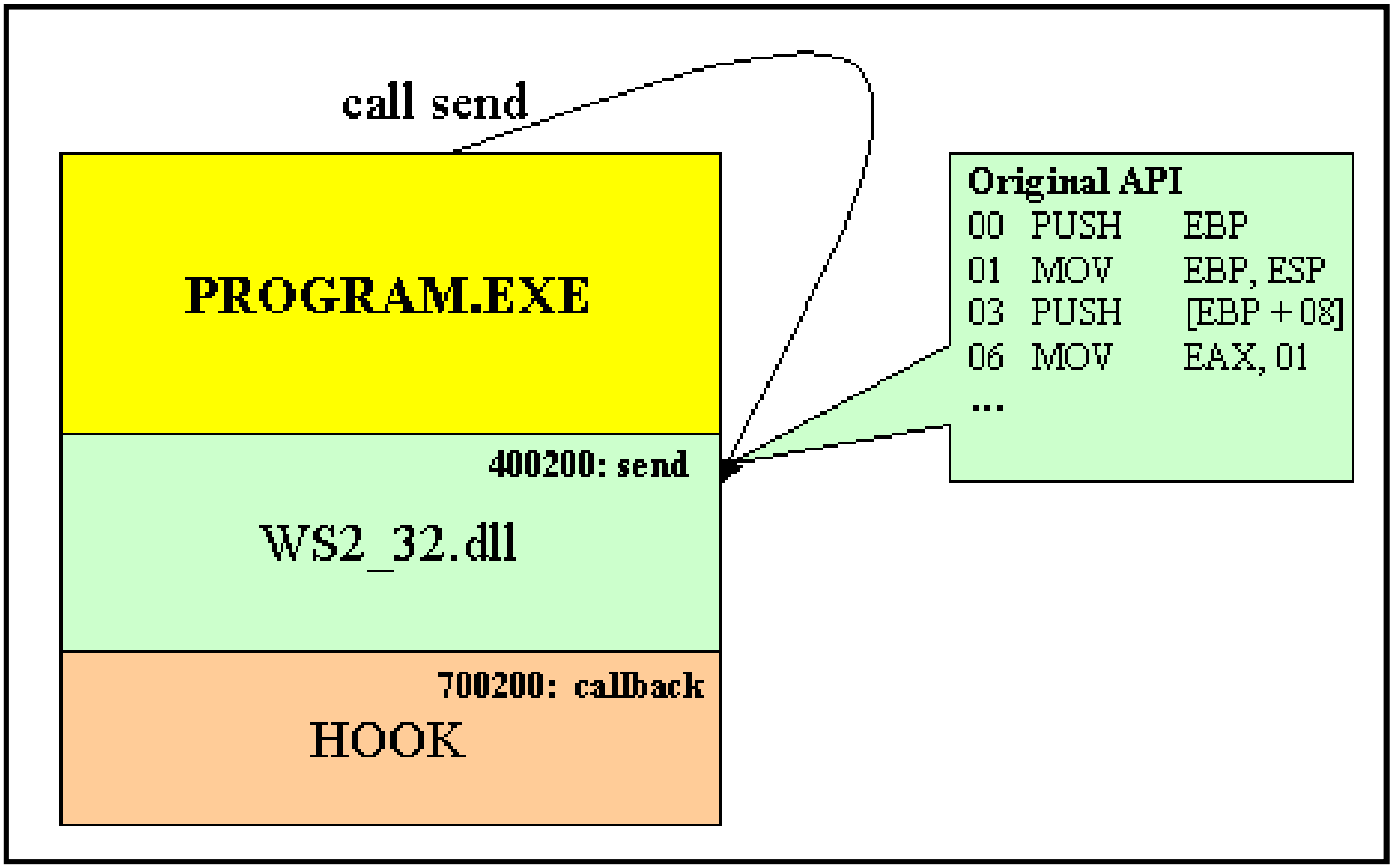
Inject-and-die,

Injection multiple

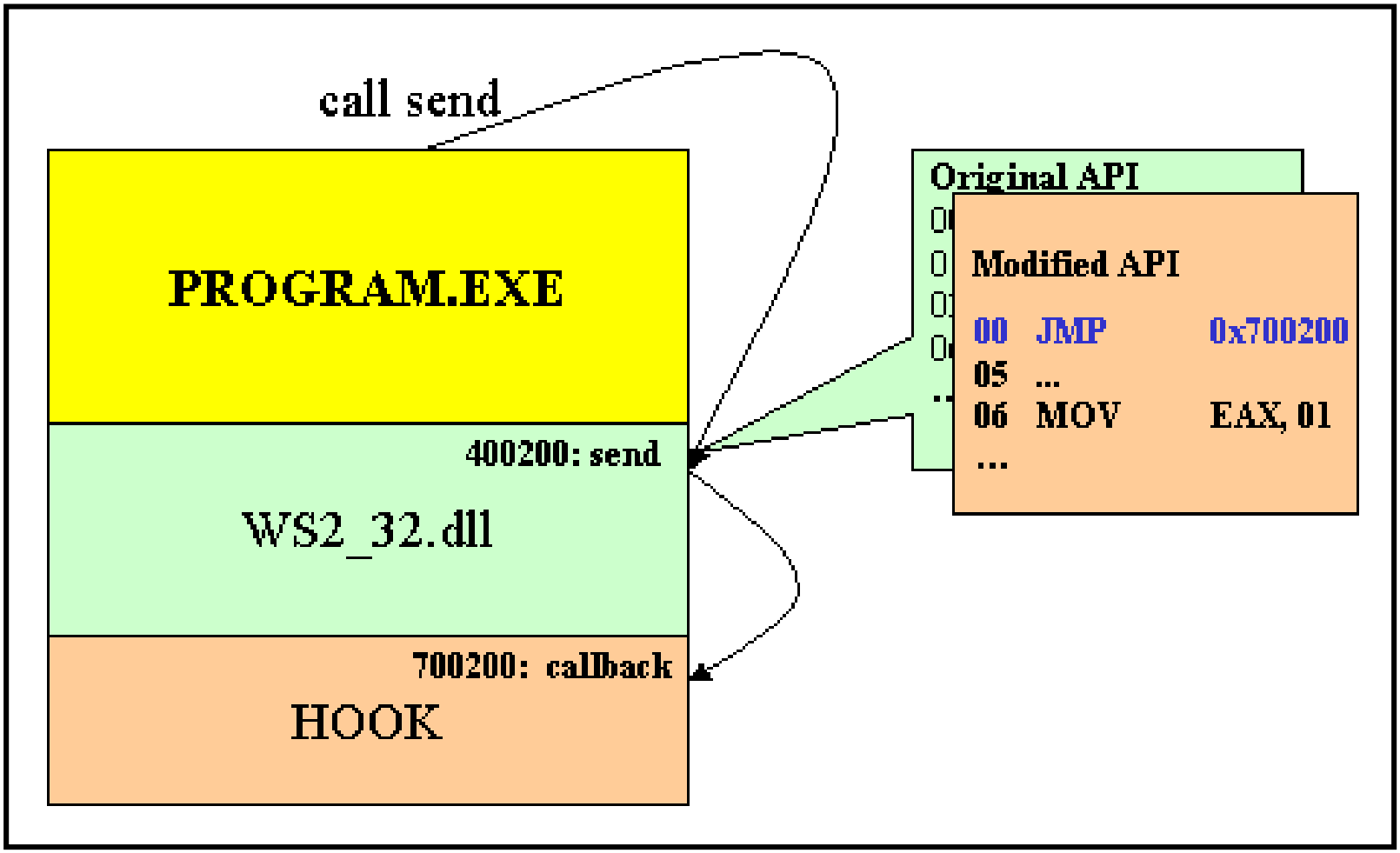
API Hooking

- But : intercepter les appels API des programmes de la machine.
- Technique connue depuis 10 ans: "Peering Inside the PE" - Matt Pietrek (Mars 1994)
- Méthode la plus utilisée: IAT hooking (Import Address Table) ⇨ Ne fonctionne pas bien dans un cheval de Troie.
- Solution la plus efficace: API hooking par JMP

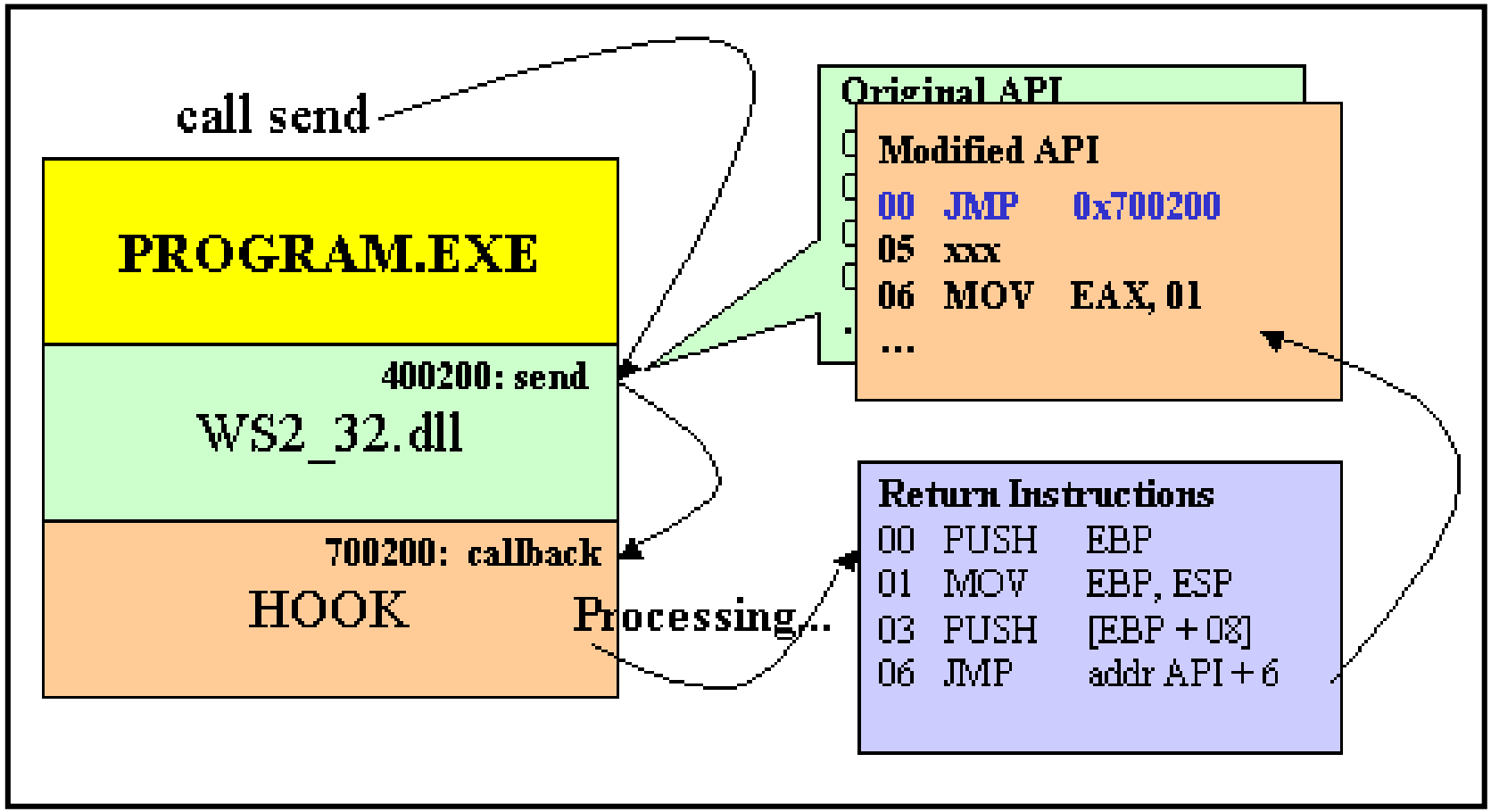
API Hooking


call send
PROGRAM.EXE
400200: send
WS2_32.dll
700200: callback
HOOK
Original API
00 PUSH EBP
01 MOV EBP, ESP
03 PUSH [EBP + 08]
06 MOV EAX, 01
...

API Hooking


call send
PROGRAM.EXE
400200: send
WS2_32.dll
700200: callback
HOOK
Original API
Modified API
00 JMP 0x700200
05 ...
06 MOV EAX, 01
...

API Hooking


call send
PROGRAM.EXE
400200: send
WS2_32.dll
700200: callback
HOOK
Processing...
Original API
Modified API
00 JMP 0x700200
05 xxx
06 MOV EAX, 01
...
Return Instructions
00 PUSH EBP
01 MOV EBP, ESP
03 PUSH [EBP + 08]
06 JMP addr API + 6

API Hooking

API hooking : Quel intérêt pour un « Trojan » ?

- Identifier les processus communiquants: hooker les APIs de communication telles que `connect`. But : effectuer des communications vers l'extérieur à partir de ces programmes. (hook `connect`)
- Vol des e-mails entrants et sortants. Vol des mots de passe POP3, proxy et serveur Socks. (hook de `recv` et `send`)
- Espionnage et log des URL visitées et formulaires d'authentification remplis sur le Web. (hook `send`)

API Hooking

API hooking: Quel intérêt pour un « Trojan » ?

- Fonctionnalités « Rootkit » : cacher les fichiers et clés de registre du cheval de Troie. Le tout en mode user et… sans privilèges d'administrateur!
- Intercepter `CreateProcess` pour rendre la survie du thread plus efficace. Tout processus créé est directement injecté.
- Interdire ou simuler les connexions vers les sites de mise à jour de signature des anti-virus (attaque spécifique qui varie selon l'anti-virus ciblé).

API Hooking

API hooking et « malware » aujourd'hui

- Certains programmes malveillants utilisent l'API hooking, mais uniquement pour des fonctionnalités de rootkit : Hacker Defender, Vanquish et autres rootkits.

L'avenir de l'API hooking dans les « malware »

- Hook d'APIs importantes : WinSock. Egalement `CreateProcess` et `LoadLibraryW`, afin d'effectuer les actions d'espionnage décrites.
- Pas d'injection de DLL

Démo

Démo:

Cheval de Troie complet

Anti-Virus à Signatures

Anti anti-virus

- Ce genre de chevaux de Troie n'est pas conçu pour une distribution à grande échelle. Ainsi, les anti-virus à signature ne sont pas efficaces pour le contrer.
- Tout de même, si l'on part du principe qu'un anti-virus connaîtra un jour la signature du cheval de Troie, celui-ci peut lui échapper en introduisant une fonction d'auto-update qui permet de recompiler un binaire différent et le redéployer sur les postes infectés. Les signatures d'anti-virus y seront alors inefficaces.

Evolutions Possibles

Injection & API Hooking

- Il est possible d'injecter du code sans utiliser `CreateRemoteThread`.
- Inclure un calculateur de taille d'instruction dynamique pour l'API Hooking.

Communication et protocoles

- Nous avons utilisé le protocole HTTP, mais bien entendu il est possible de faire la même chose avec d'autres protocoles tels que : DNS, FTP, SMTP.
- Espionnage de trafic crypté en SSL : Faisable!

Prévention

- Théoriquement, ce genre de chevaux de Troie n'est pas détectable à partir du moment où il y a exécution. Ainsi, le meilleur moyen de lutter contre reste la prévention.
- La protection contre les chevaux de Troie, virus et vers passe par les étapes suivantes :

Prévention

1. Formation des utilisateurs contre le syndrome du « je-clique-partout ».

2. Mise en place régulière de correctifs de sécurité.

3. Implémentation d'une politique de sécurité pour les téléchargements et pièces jointes des e-mails.

4. Sécurisation du démarrage système (« startup »).

5. Protection des processus en mémoire.

6. Lorsque cela est possible, sécuriser les postes clients contre les programmes non validés.

Conclusion

- Une fois exécuté, les possibilités d'un programme malveillant sont quasiment illimitées. Et ce, en dépit des firewalls réseau, firewall personnels et autres anti-virus.
- Question : puisque ces techniques existent depuis si longtemps, pourquoi ne voit-on pas plus de tels chevaux de Troie furtifs ?
  - 1) Le fait que les plateformes Windows ont seulement été unifiées récemment depuis Windows 2000.

Conclusion

- 2) Au vue de leur furtivité, il est difficile de savoir s'ils se trouvent déjà sur des postes en production ou pas.

- Quoi qu'il en soit, il faut s'attendre à une forte augmentation dans le développement et l'utilisation de ce genre de chevaux de Troie dans les mois et années à venir.

- La probabilité de l'introduction d'un tel cheval de Troie est forte avec le manque de vigilance actuelle (cf la propagation des vers/virus)