Vulnérabilité des postes clients

SSTIC 2006

Ernst & Young

Technology & Security Risk Services

http://www.ey.com/global/content.nsf/France/home_isaas

Gaël Delalleau
prenom.nom@fr.ey.com

Renaud Feil
prenom.nom@fr.ey.com

Contexte de la présentation

- Cette présentation s'inscrit dans le cadre du thème du SSTIC 2006 « **les limites de la sécurité** » et présente notamment les limites fondamentales de certaines mesures de protection.
- Les attaques démontrées lors de cette conférence ont pour objectif d'alerter sur les risques que nous avons identifiés concernant la sécurité des postes clients.
- Les morceaux de code *proof of concept* présentés à titre de démonstration ne sont pas réutilisables tels quels pour réaliser ces attaques.
- Des propositions de solutions sont détaillées dans la dernière partie de cette présentation.

Contenu de la présentation

- Pourquoi s'intéresser à la sécurité des postes clients ?
- Limites des technologies protégeant les postes clients. Malgré certaines idées reçues, un attaquant peut :
  - Identifier avec précision les logiciels clients et leur environnement ;
  - Faire exécuter un code hostile dont la signature est connue par l'antivirus du poste client… sans qu'il ne soit détecté ;
  - Se servir d'un navigateur web pour rebondir vers les applications internes.
- Automatisation des attaques sur les postes clients : présentation d'un *framework* d'attaque des postes clients (démonstrateur).
- Protections contre la menace d'attaques sur les postes clients.

Pourquoi s'intéresser à la sécurité des postes clients ?

- Les *firewall* récents ne sont plus vulnérables aux attaques permettant de contourner leur filtrage de façon triviale.
- Les serveurs en *DMZ* font l'objet [illegible] procédures de sécurisation.

- Conclusion : la surface de vulnérabilité ne se limite plus aux adresses IP et ports visibles depuis Internet.

Pourquoi s'intéresser à la sécurité des postes clients ?

- Le poste de travail, fixe ou nomade, est le maillon faible de la sécurité du système d'information de l'entreprise.
- Les postes clients ont accès aux ressources sensibles de l'entreprise :
  - Les employés accèdent aux applications et aux données de l'entreprise depuis leur poste de travail ;
  - La compromission d'un poste client donne un accès au réseau interne.
- Les postes clients sont vulnérables :
  - Les mesures de protection « classiques » (firewall, IDS/IPS, proxy avec authentification, …) sont limitées ;
  - Les logiciels clients sont autant ou plus vulnérables que les logiciels serveurs ;
  - La sensibilisation des utilisateurs ne garantit pas que les attaques échoueront.

Ce qu'il faut retenir

- Les mécanismes de sécurité habituellement mis en place dans les entreprises protègent correctement les postes clients des attaques non ciblées…
  - Exemple : virus, vers, *spywares*.
- … mais, du fait de limites fondamentales dans ces technologies, ils ne protègent pas suffisamment contre des attaques ciblées, pouvant être en partie automatisées, visant les postes de travail des employés.
  - Exemple : attaque des logiciels clients pour déposer un cheval de Troie ou une *backdoor*, attaque des applications Web internes.
- Nous présentons plusieurs exemples d'idées reçues sur l'efficacité de ces protections donnant un faux sentiment de sécurité, les arguments techniques indiquant qu'un attaquant pourrait en réalité les contourner, et des pistes de solutions

Limites des technologies protégeant les postes clients

Exemple en phase de découverte :
Limites des protections associées à la prise d'empreinte des logiciels clients

« Je suis caché ! »

Un logiciel client anonyme

- Idées reçues :
  - J'utilise un logiciel client « alternatif » et/ou un système d'exploitation différent de Windows : les attaques ne peuvent pas me cibler avec précision.
  - Les bannières de mon logiciel client sont modifiées ou désactivées.
- Raisons fondamentales permettant une identification précise :
  - Différences dans le design du logiciel (choix du support des fonctionnalités prévues par les standards, présence de fonctionnalités propriétaires, etc.…).
  - Différences d'implémentation (similaires à celles utilisées dans le *fingerprinting* de pile TCP/IP pour déterminer le système d'exploitation).

Fingerprinting de logiciel client : l'exemple des navigateurs Web

- Différences dans les valeurs et l'ordre relatif des en-têtes HTTP :
  - En-têtes « Accept », « If-Modified-Since », « Referer », « Cookie »,...
  - Mais peuvent être modifiées par le *proxy* : technique peu fiable.
- Différences dans le support de certaines fonctionnalités.
- Utilisation des fonctions disponibles dans les moteurs de script et les différents *plugins* des navigateurs :
  - Exemples :
    - navigator.userAgent (exemple : code permettant de récupérer la valeur du userAgent, même si elle est filtré par le *proxy*)
    - navigator.appMinorVersion
    - navigator.plateform
    - navigator.cpuClass
    - navigator.oscpu
    - navigator.plugins
    - navigator.mimeTypes

```
<script language="javascript">
document.write("<img src='/userAgent_"
+ navigator.userAgent + ".jpg'>");
</script>
```

Fingerprinting de logiciel client : l'exemple des navigateurs Web

- Fonctionnalités spécifiques à Internet Explorer :
  - Version du moteur de script : ScriptEngineBuildVersion()

OS	Version IE	Script Engine Version
Windows 2000 sans SP	IE 5.0	5.1.4615
Windows 2000 SP 4	IE 5.0 SP4	5.1.8513
Windows XP SP2	IE 6.0 SP2	5.6.8820

  - Versions des composants installés : getComponentVersion().
    - Exemple : script permettant la récupération de la version du lecteur Windows Media Player (par exemple « 10,0,0,3646 »).

```
<p id=oCC style="behavior:url('#default#clientCaps')">
<script language="javascript">
document.write("<img src='/WMP_" +
oCC.getComponentVersion('{22d6f312-b0f6-11d0-94ab-
0080c74c7e95}', 'componentid') + ".jpg'>");
</script></p>
```

Fingerprinting de logiciel client : l'exemple des navigateurs Web

- Différences d'implémentation de certaines fonctionnalités, comme le parsing du code HTML.
  - Exemple : les tags correctement interprétés sont indiqués en vert

(IE : Internet Explorer / FF : Firefox / MZ : Mozilla / NS : Netscape / OP : Opéra / HT : WinHTTrack)

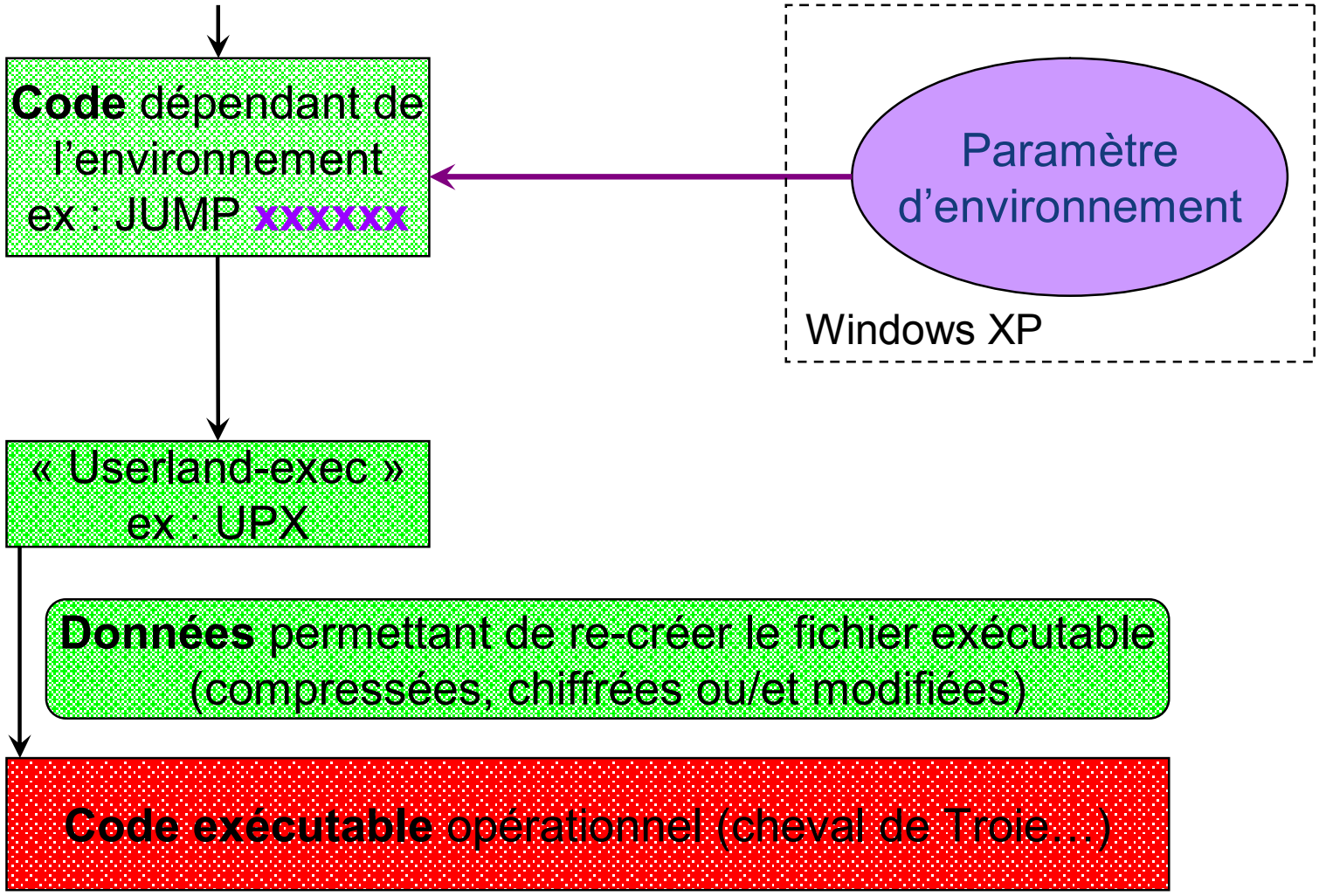
Balise HTML	IE 5 et 6	FF 1.0.x	FF 1.5.x	NS 7.1	OP 6.0	OP 8.53	HT 3.33
<img/src="/test0001">	x						
<im\x00g src="/test0002">	x						x
<img\x00src="/test0003">		x	x				x
<img\x01src="/test0004">							
<img\x0Asrc="/test0005">	x	x	x	x	x	x	x
<im\x0Ag src="/test0006">							x
<img"src="/test0007">		x		x			x
<img dynsrc="/test0008">	x				x		x
<img lowsrc="/test0009">	x						x

Exemple en phase d'attaque :
Limites des protections associées au filtrage de code malveillant

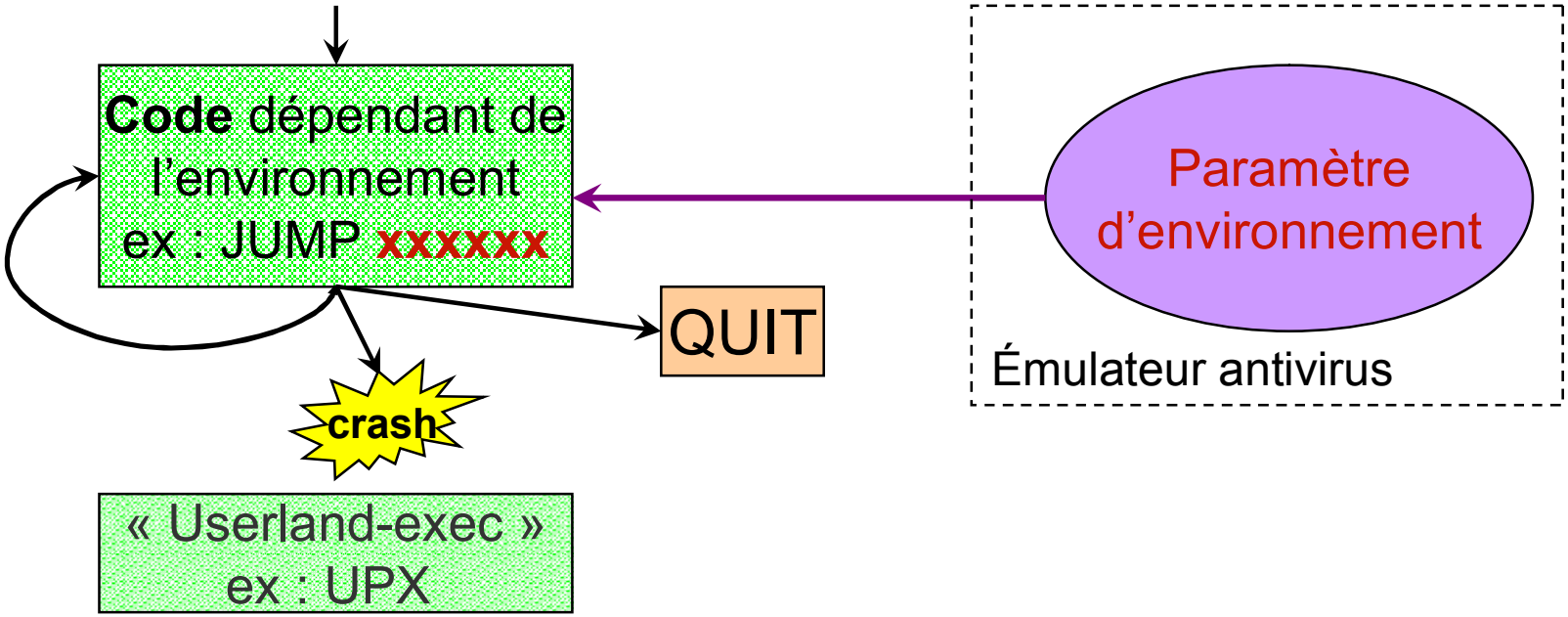
Limites des dispositifs de protection contre l'exécution de code malveillant

- Idées reçues :
  - Mon poste de travail, et ma passerelle Internet, disposent d'un antivirus à jour qui me protège contre tous les *malwares* connus.
  - Si ces *malwares* connus sont chiffrés ou compressés, il suffit à l'antivirus d'émuler la routine de déchiffrement pour détecter le code hostile.
- Limites fondamentales :
  - Un même code peut se comporter différemment en fonction de son environnement d'exécution (exécution sur un système Windows ou émulation dans un antivirus)
    - Données retournées par certains appels systèmes ;
    - Contenu de certaines zones de la mémoire ;
    - Temps d'exécution de certaines instructions ;
    - Contenu de certains fichiers ou d'une page Web…
  - Il est donc possible à tout programme d'exhiber un comportement inoffensif lorsqu'il est analysé par l'antivirus, et un comportement hostile lorsqu'il est exécuté normalement sur un système.

Exécution « normale »



Émulation du code par un antivirus



Données permettant de re-créer le fichier exécutable (compressées, chiffrées ou/et modifiées)

Contournement d'antivirus : exemple

Mesure du temps d'exécution (1)

```
        CALL envdep_geteip
geteip:
        POP ECX
        RDTSC                   ; TIMESTAMP 1
        MOV EBX, EAX
        RDTSC                   ; TIMESTAMP 2
        SUB EAX, EBX            ; DIFF = TIMESTAMP 2 - TIMESTAMP 1
        SHR EAX, 4              ; OFFSET = DIFF / 2^4
        ADD EAX, 0x13           ; OFFSET = OFFSET + 0x13
        ADD EAX, ECX            ; ADDR = geteip + OFFSET
loop:
        JMP EAX                 ; On saute à l'adresse ADDR
        JMP SHORT loop          ; boucle infinie...
        NOP
        ...
        MOV DWORK ['ADR0], 'VALO' ; Modification du code compressé
        JMP userland_exec       ; Décompression et exécution du code
```

Mesure du temps d'exécution (2)

```
        CALL envdep_geteip
geteip:
        POP ECX
        RDTSC                   ; TIMESTAMP 1
        MOV EBX, EAX
        RDTSC                   ; TIMESTAMP 2
        SUB EAX, EBX            ; DIFF = TIMESTAMP 2 - TIMESTAMP 1
        SHR EAX, 4              ; OFFSET = DIFF / 2^4
        ADD EAX, 0x13           ; OFFSET = OFFSET + 0x13
        ADD EAX, ECX            ; ADDR = geteip + OFFSET
loop:
        JMP EAX                 ; On saute à l'adresse ADDR
        JMP SHORT loop          ; boucle infinie...
        NOP
        ...
        MOV DWORK ['ADR0], 'VALO' ; Modification du code compressé
        JMP userland_exec       ; Décompression et exécution du code
```

Mesure du temps d'exécution (3)

```
        CALL envdep_geteip
geteip:
        POP ECX
        RDTSC                   ; TIMESTAMP 1
        MOV EBX, EAX
        RDTSC                   ; TIMESTAMP 2
        SUB EAX, EBX            ; DIFF = TIMESTAMP 2 - TIMESTAMP 1
        SHR EAX, 4              ; OFFSET = DIFF / 2^4
        ADD EAX, 0x13           ; OFFSET = OFFSET + 0x13
        ADD EAX, ECX            ; ADDR = geteip + OFFSET
loop:
        JMP EAX                 ; On saute à l'adresse ADDR
        JMP SHORT loop          ; boucle infinie...
        NOP
        ...
        MOV DWORK ['ADR0], 'VALO' ; Modification du code compressé
        JMP userland_exec       ; Décompression et exécution du code
```

Résultat dans l'émulateur de l'antivirus

```
        CALL envdep_geteip
geteip:
        POP ECX
        RDTSC                   ; TIMESTAMP 1
        MOV EBX, EAX
        RDTSC                   ; TIMESTAMP 2
        SUB EAX, EBX            ; DIFF = TIMESTAMP 2 - TIMESTAMP 1
        SHR EAX, 4              ; OFFSET = DIFF / 2^4
        ADD EAX, 0x13           ; OFFSET = OFFSET + 0x13
        ADD EAX, ECX            ; ADDR = geteip + OFFSET
loop:
        JMP EAX                 ; On saute à l'adresse ADDR
        JMP SHORT loop          ; boucle infinie...
        NOP
        ...
        MOV DWORK ['ADR0], 'VALO' ; Modification du code compressé
        JMP userland_exec       ; Décompression et exécution du code
```

Résultat dans l'émulateur de l'antivirus

```
        CALL envdep_geteip
geteip:
        POP ECX
        RDTSC                       ; TIMESTAMP
        MOV EBX, EAX
        RDTSC                       ; TIME
        SUB EAX, EBX                ; DI      ESTAMP 2 - TIMESTAMP 1
        SHR EAX, 4                  ;       = DIFF / 2^4
        ADD EAX, 0x13               ;     T = OFFSET + 0x13
        ADD EAX, ECX                ;     DR = geteip + OFFSET
loop:
        JMP EAX                     ; On saute à l'adresse ADDR
        JMP SHORT l                 ; boucle infinie...
        NOP
        ...
        MOV DWORR   'ADR0], 'VALO'  ; Modification du code compressé
        JMP userland_exec           ; Décompression et exécution du code
```

Résultat en exécution « normale »

```
        CALL envdep_geteip
geteip:
        POP ECX
        RDTSC                   ; TIMESTAMP 1
        MOV EBX, EAX
        RDTSC                   ; TIMESTAMP 2
        SUB EAX, EBX            ; DIFF = TIMESTAMP 2 - TIMESTAMP 1
        SH
        AD
        AD
loop:
        JMP EAX                 ; On saute à l'adresse ADDR
        JMP SHORT loop          ; boucle infinie...
        NOP
        ...
        MOV DWORK ['ADR0], 'VALO'  ; Modification du code compressé
        JMP userland_exec       ; Décompression et exécution du code
```

CODE HOSTILE !

Contournement d'antivirus : résultats

- Résultats des tests : (source www.virustotal.com)

Contenu de l'exécutable	Virus détecté	Programme « suspicieux »	Pas de détection
Cheval de Troie (CDT) seul	**24**	0	0
CDT compressé par UPX	16	8	0
CDT compressé par UPX et protégé par cette technique	0	2	**22**

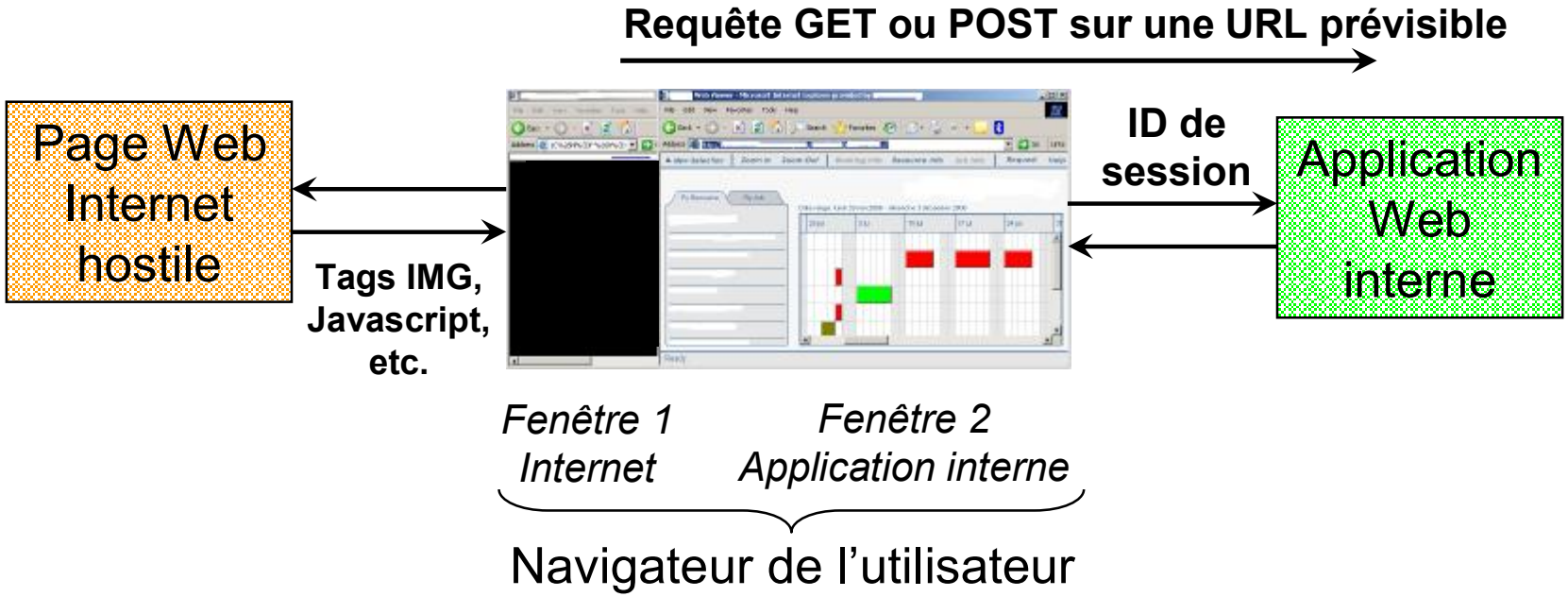
- **Conclusion** : il est possible d'exécuter un code dont la signature est connue sans qu'il soit bloqué par l'antivirus.

Exemple en phase d'attaque :
Limites des protections associées aux accès aux applications Web internes

Exposition des applications internes aux attaques provenant d'Internet

- Idées reçues
  - Mes applications Web internes ne sont pas accessibles depuis l'extérieur
  - Mes applications Web internes sont protégées par une authentification
    - Identifiant/mot de passe
    - Authentification forte
    - Authentification « two factors »
    - Single Sign On sur le réseau
  - Mes logiciels clients sur les postes de travail sont à jour des correctifs de sécurité, ils ne peuvent donc pas être exploités pour accéder au réseau interne
- Limite fondamentale
  - Les fonctionnalités « par défaut » des navigateurs Web permettent de s'en servir comme relais pour se connecter aux applications Web internes depuis Internet, avec les droits de l'utilisateur utilisant le navigateur

Exposition des applications internes aux attaques provenant d'Internet



• Par design, toute page Web peut effectuer une requête GET ou POST à destination d'un site Web arbitraire (interne ou externe)
• Les éléments d'authentification de l'utilisateur sur le site appelé (cookies de session, authentification HTTP…) sont transmis par le navigateur
• Possibilité de proxy Javascript entre Internet et les applications internes vulnérables au Cross Site Scripting

Automatisation des attaques sur les postes clients

Framework de démonstration

Gaël Delalleau - Renaud Feil

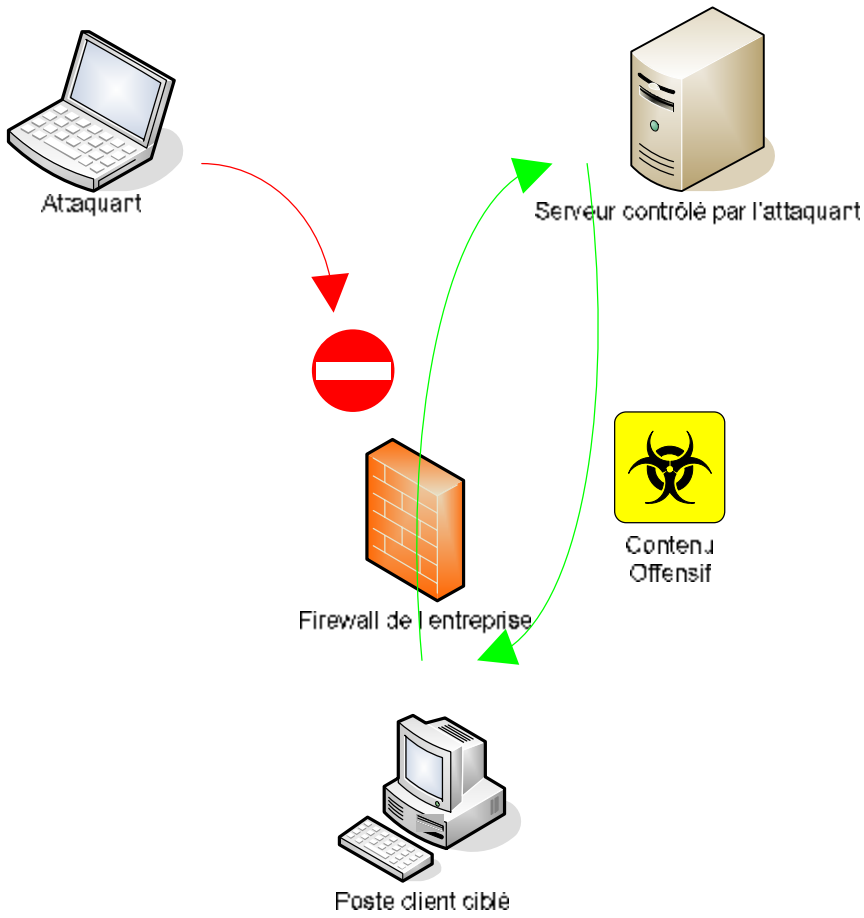
Un risque avéré : l'automatisation des attaques sur les postes clients

- Les attaques contre les logiciels clients se développent mais ne sont pas triviales.
- L'attaque doit se faire sur un laps de temps très court et l'attaquant ne peut savoir :
  - à quel moment les utilisateurs ciblés se connecteront,
  - quelles seront les logiciels clients utilisés et leurs versions,
  - quels seront les systèmes d'exploitations sous-jacents,
  - quels seront les filtrages et les mécanismes de sécurité pouvant bloquer certains types d'attaques.
- C'est le « sens de l'histoire » de voir apparaître des outils automatisant les attaques vers les postes clients pour :
  - détecter les caractéristiques de l'environnement ciblé,
  - envoyer les attaques les plus pertinentes… jusqu'à ce qu'une de ces attaques aboutisse,
  - maintenir un canal de communication avec les postes clients compromis.

Présentation du démonstrateur

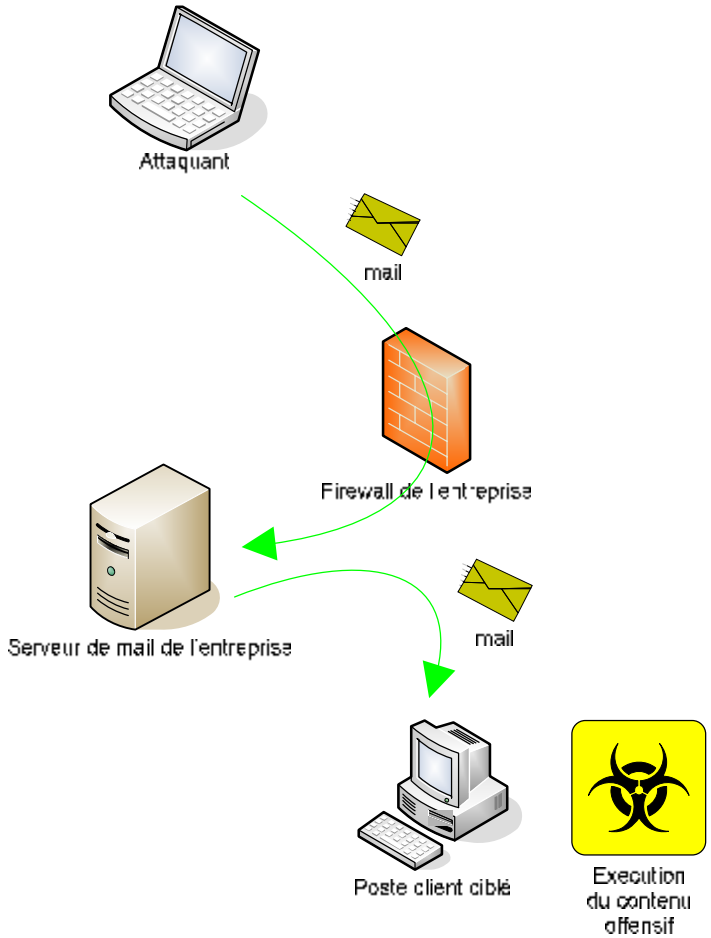
- *Framework* d'exploitation des postes clients :
  - Python,
  - SGBDR MySQL.
- Différents modules :
  - **« fakemail »** : génération des courriers électroniques avec un tag permettant de suivre les connexions des utilisateurs ciblés.
  - **« track »** : suivi des attaques en cours, des succès et des échecs, pour permettre une adaptation rapide.
  - **« core »** : gestion des connexions entrantes
    - pour l'instant : HTTP et HTTPS
    - possibilité d'ajouter d'autres protocoles : flux audio/vidéo, SMTP, POP3, IMAP4, telnet…
  - **« fingerprinting »** : identification du logiciel client se connectant et du système d'exploitation.
  - **« content »** : consolidation du contenu à renvoyer à l'utilisateur à partir d'une base de données d'exploits et de contenu.

Attaque 1 : Dépôt d'un contenu offensif sur un serveur de l'Internet



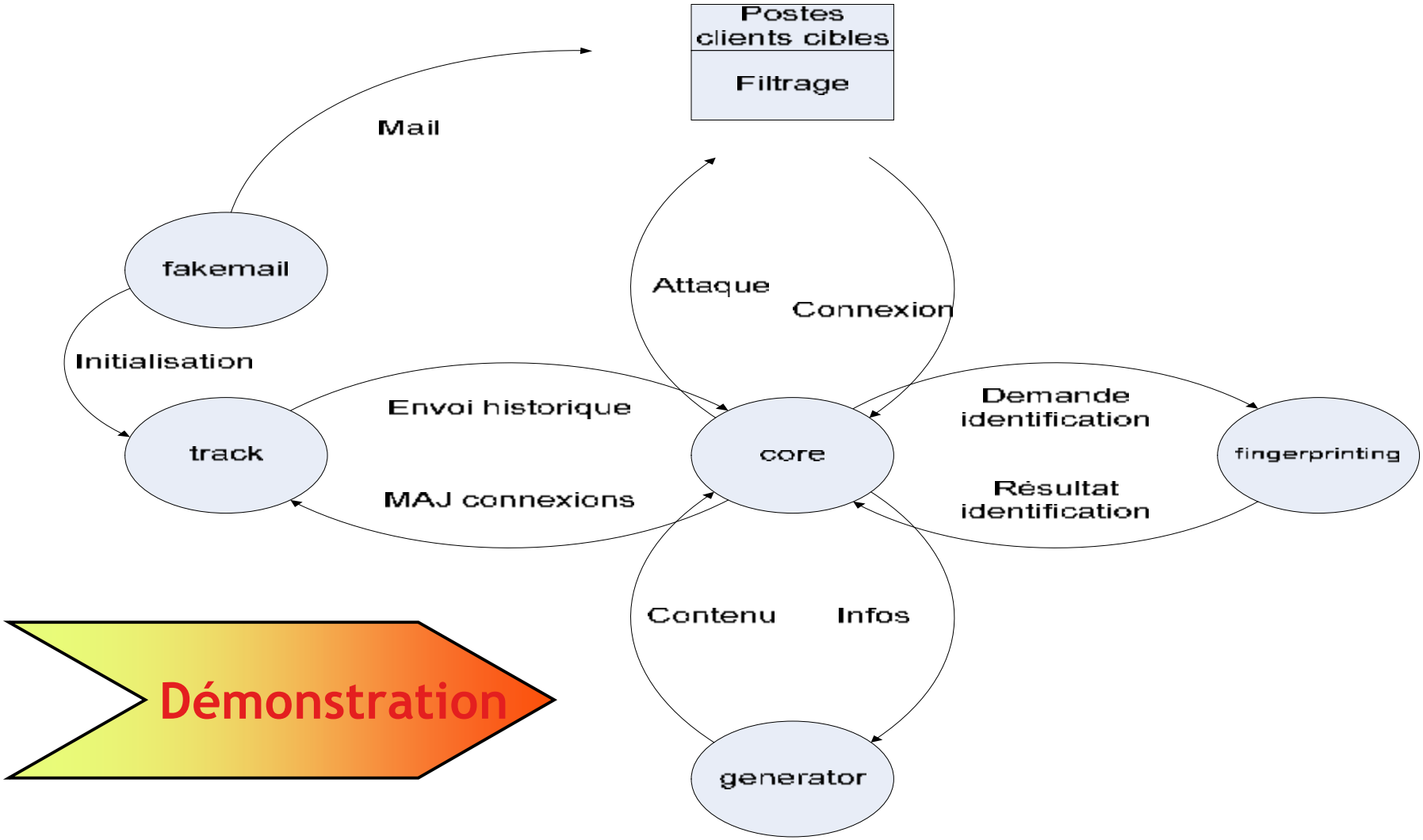
- Envoi d'un mail contenant un lien vers le contenu offensif (et un tag pour identifier l'utilisateur ciblé).
- Lors de la connexion du poste client, *fingerprinting* du logiciel client utilisé.
- A partir des objectifs définis pour cet utilisateur, envoi d'un contenu pour exploiter une faille sur le logiciel client.
- Adaptation de l'attaque jusqu'au succès (… ou échec !).
- Création d'un tunnel de communication avec le réseau interne de l'entreprise.

Attaque 2 : Dépôt du contenu offensif sur un serveur mail



- Le contenu offensif est déposé dans la boite mail des utilisateurs ciblés.
- Contournement du filtrage antivirus :
  - Code offensif « inconnu ».
  - Code offensif « connu », mais avec une technique de contournement de l'antivirus.
- Exploitation d'une faille sur un logiciel du poste de travail (ou lancement exécutable).
- Création d'un tunnel de communication avec le réseau interne de l'entreprise.

Présentation du démonstrateur



Protections contre la menace d'attaques automatisées des postes clients

Prévention des attaques

- Renforcement de la configuration du poste de travail :
  - « End-point protection »,
  - NX,
  - Paramétrage de sécurité des logiciels clients (par exemple : zones de sécurité d'Internet Explorer).
- Renforcement du filtrage de périmètre :
  - Filtrage selon un mode « liste blanche » si possible (au moins pour les sites accédés en HTTPS),
  - Interdire la récupération d'exécutables depuis Internet.
- Renforcement du suivi des connexions dans les applications Web
  - Exemple : adresses URL non prévisibles contenant un ID de session en complément du cookie
- Utilisation d'un navigateur dédié pour Internet passant par un proxy n'ayant pas accès au réseau interne
- Sensibilisation des utilisateurs

Détection et réaction aux attaques

- Ajouter une logique de détection et de réaction des incidents de sécurité à la logique actuelle de prévention, en s'inspirant pour cela de ce qui existe déjà pour les serveurs :
  - Détection de canaux de communication cachés,
  - IDS/IPS notamment sur les postes de travail,
  - Honeypots servant d'alerte,
  - Honeytokens placés dans les applications internes et sur les postes clients.
- Mettre en place une procédure de réaction aux incidents adaptée au cas d'une attaque ciblée sur un ou plusieurs postes de travail.
  - Procédure de réaction adaptée pour la *hotline*.

Résilience aux attaques

- Cloisonnement des segments réseau contenant des postes clients :
  - Vision pragmatique : un poste client sera compromis « tôt ou tard ».
  - Considérer le réseau bureautique comme « semi-hostile »…
  - … ou même « hostile » (vision du forum Jericho).
  - Protège de plus des malveillances d'un utilisateur connecté au réseau bureautique (employés, intérimaires ou personnes s'étant introduite par effraction dans les locaux).
  - Mais suppose des efforts pour la mise en place et la maintenance de règles de filtrages efficaces (c'est-à-dire restrictives).

Merci pour votre attention

… des questions ?