Liberté • Égalité • Fraternité
RÉPUBLIQUE FRANÇAISE
PREMIER MINISTRE

S . G . D . S . N
Agence nationale de la sécurité des systèmes d'information
CERT-FR

Paris, le 13 avril 2015
N° CERTFR-2015-ACT-015

Affaire suivie par :
CERT-FR

BULLETIN D'ACTUALITÉ

Objet : Bulletin d'actualité CERTFR-2015-ACT-015

1 - Protection Control Flow Guard

La fonctionnalité *Control Flow Guard* (CFG) est une option de sécurité mise en place par Microsoft sur ses systèmes d'exploitation récents, comme Windows 8.1 et Windows 10. L'objectif de cette protection est de réduire au maximum les possibilités d'exploitation de vulnérabilités par appels indirects.

Lors de l'exploitation d'une vulnérabilité, il est fréquent qu'un attaquant réécrive des structures de données contenant des pointeurs de fonction. Ceux-ci sont ensuite utilisés par des instructions telles que `call esi`, notamment dans le cas d'un appel de méthode.

Dans le cas d'un code d'exploitation, le registre `esi` peut pointer sur une suite de « gadgets » pour mener une exploitation dite *Return Oriented Programing* (ROP).

Pendant une exécution normale, les appels sont destinés à des fonctions internes à l'application et peuvent donc être vérifiés. A l'opposé, les exploitations de type ROP exécutent souvent uniquement des fins de fonctions (appelées « gadgets ») comme par exemple :

```
pop esi
pop edi
ret
```

Ces instructions étant dans l'épilogue d'une fonction, il est possible de valider la cohérence d'un appel si la liste des adresses des fonctions est connue.

Visual Studio 2015 a ajouté une nouvelle option de compilation [1] permettant de référencer les fonctions dans un espace mémoire. Les binaires compilés avec cette option auront un prologue avant les appels indirects :

```
mov esi, [eax+8]
mov ecx, esi
call ds:___guard_check_icall_fptr
call esi
```

Dans un binaire compilé avec cette option, `___guard_check_icall_fptr` pointe sur la fonction `sub_635F4F70` :

```
.idata:646A5F34 ___guard_check_icall_fptr dd offset sub_635F4F70

sub_635F4F70 proc near
mov eax, eax
ret
sub_635F4F70 endp
```

Cette fonction ne fait donc aucune action à part rendre la main à l'appelant. Seulement, le binaire exécuté concerné possède le drapeau guard de l'entête optionnel PE `DllCharacteristics` à 1.

Quand ce drapeau est positionné, la bibliothèque `ntdll.dll` lira la valeur `PE.Optional_Header.Data_Directories[Configuration_Directory]` pour obtenir l'adresse du pointeur sur la routine de validation (offset 0x48).

NTDLL remplacera alors le pointeur sur la fonction inutile par un pointeur sur `LdrpValidateUserCallTarget`.

```
.text:6B2988C8  mov     esi, [esi+48h]
.text:6B2988CB  test    esi, esi
.text:6B2988CD  jz      loc_6B2D7993
[...]
.text:6B29891F  mov     edi, offset @LdrpValidateUserCallTarget@4~;
                                                    LdrpValidateUserCallTarget(x)
[...]
.text:6B298925  mov     [esi], edi
```

Ainsi, durant l'exécution du binaire, l'adresse à valider est placée dans le registre `ECX` et la fonction `LdrpValidateUserCallTarget` est appelée pour vérifier la pertinence du pointeur.

```
60AA296E    8B70 08         MOV ESI,DWORD PTR DS:[EAX+8]
60AA2971    8BCE            MOV ECX,ESI
60AA2973    FF15 345F5061   CALL DWORD PTR DS:[61505F34]~; LdrpValidateUserCallTarget
60AA2979    FFD6            CALL ESI
```

NTDLL possède une zone mémoire nommée `CFGBitmap` représentant les adresses de début de chacune des fonctions dans l'espace virtuel du processus. Cette zone est construite de la façon suivante : chaque bloc de 8 octets de l'espace mémoire du processus est transformé en un bit qui vaut 1 si le bloc contient une adresse de début d'une fonction, 0 sinon.

Lors de l'appel à la fonction `LdrpValidateUserCallTarget`, la bibliothèque NTDLL va calculer l'entrée dans `CFGBitmap` correspondant à l'adresse à vérifier. La valeur obtenue, appelée `point`, est alors testée pour déterminer s'il s'agit d'un appel légitime de fonction ou non.

```
/ (fcn) LdrpValidateUserCallTarget 136
|           0x6b2aaa40  8b154032386b    mov edx, dword [0x6b383240] ; [:4]=0
|           0x6b2aaa46  8bc1            mov eax, ecx
|           0x6b2aaa48  c1e808          shr eax, 8
|           0x6b2aaa4b  8b1482          mov edx, dword [edx + eax*4]
|           0x6b2aaa4e  8bc1            mov eax, ecx
|           0x6b2aaa50  c1e803          shr eax, 3
|           0x6b2aaa53  f6c10f          test cl, 0xf
|       ,=< 0x6b2aaa56  7506            jne 0x6b2aaa5e
|       |   0x6b2aaa58  0fa3c2          bt edx, eax
|      ,==< 0x6b2aaa5b  730a            jae 0x6b2aaa67
|      ||   0x6b2aaa5d  c3              ret
|      ||
|      ||   ; JMP XREF from 0x6b2aaa56 (LdrpValidateUserCallTarget)
|      |`-> 0x6b2aaa5e  83c801          or eax, 1
|      |    0x6b2aaa61  0fa3c2          bt edx, eax
|     ,===< 0x6b2aaa64  7301            jae 0x6b2aaa67
|     ||    0x6b2aaa66  c3              ret
|     ||
|     ||    ; JMP XREF from 0x6b2aaa64 (LdrpValidateUserCallTarget)
|     ``--> 0x6b2aaa67  51              push ecx
|           0x6b2aaa68  8d642480        lea esp, dword [esp - 0x80]
[...]
|           0x6b2aaa93  e8d6ac0b00      call RtlpHandleInvalidUserCallTarget
|              RtlpHandleInvalidUserCallTarget(unk)
[...]
|           0x6b2aaabf  8da42480000.    lea esp, dword [esp + 0x80]
|           0x6b2aaac6  59              pop ecx
\
```

Pour ce faire, `LdrpValidateUserCallTarget` lit une valeur de 4 octets dans la zone mémoire tel que `point = *(unsigned int *)(CFGBitmap + 4 * (address » 8))` (lignes 1 à 4).

La valeur obtenue est ensuite vérifiée avec l'instruction `bittest(point, address » 3)` (ligne 9 et 14). Si le bit dans `point` à l'index `address » 3` est positionné à 1, la fonction retourne dans son chemin d'exécution légitime. Dans le cas contraire, la fonction `RtlpHandleInvalidUserCallTarget` est appelée et le processus se termine.

Il est important de noter que les fichiers DLL supportant CFG n'utiliseront cette fonctionnalité que s'ils sont chargés dans un processus le supportant également.

Le CERT-FR recommande de compiler les binaires sensibles en utilisant cette option. Elle permet de limiter significativement les tentatives de redirections de chemin d'exécution provoquées par un attaquant.

Documentation

– 1. Documentation sur le CFG :
http://blogs.msdn.com/b/vcblog/archive/2014/12/08/visual-studio-2015-preview-work-in-progress-security-feature.aspx
– 2. Article de TrendMicro :
http://blog.trendmicro.com/trendlabs-security-intelligence/exploring-control-flow-guard-in-windows-10/

2 - Sécurité Active Directory: détecter l'indétectable

L'objectif de cet article est de présenter certains événements potentiellement caractéristiques d'une intrusion d'un réseau basé sur Active Directory.

La compromission initiale commence le plus souvent par une machine mal administrée et/ou par un utilisateur peu sensibilisé aux menaces informatiques. Par la suite, un attaquant va généralement utiliser les moyens standards d'accès aux informations (SMB, RDP, OWA, etc.) avec un compte utilisateur légitime pour se déplacer ou accéder aux données de l'entreprise rendant de ce fait la détection plus difficile.

Le nombre d'évènements générés en environnement Active Directory étant très élevé, il est difficile de discerner l'information indiquant la compromission. L'approche retenue ici consiste à définir dans un premier temps un sous-ensemble d'objets que l'on désire surveiller tels que les contrôleurs de domaine, les serveurs sensibles ou les comptes privilégiés afin de relever tout comportement anormal.

La définition d'un comportement anormal implique de maîtriser auparavant l'ensemble des relations entre les différents objets de l'Active Directory. Cela consiste notamment à répertorier toutes celles que l'on considère comme normales au sein de l'environnement étudié. L'établissement d'un référentiel de comportements « normaux » permet d'identifier les scénarios pour lesquels il n'y a pas d'ambigüité, tels que :

– les comptes à forts privilèges s'authentifient exclusivement sur une liste définie de machines, telle que les contrôleurs de domaine ou stations d'administration ;
– les comptes privilégiés se connectent sur une plage horaire spécifique ;
– les comptes privilégiés doivent uniquement accéder à une liste spécifique de ressources (serveurs sensibles, contrôleur de domaine) ;
– les ressources sensibles sont accédées uniquement par une liste définie d'utilisateurs depuis une liste définie d'ordinateurs.

Une fois ces scénarios établis, il est considéré comme « inhabituel » et nécessitant un traitement particulier les évènements signalant une déviance par rapport aux modèles établis. L'utilisation d'un collecteur d'événements Windows permet d'automatiser les traitements et la levée d'alertes afin de signaler au plus vite des scénarios tels que :

– l'émission de ticket de service pour une machine inhabituelle ou pour un compte non existant dans le domaine (afin de détecter l'usage d'un « Golden Ticket ») [évènement 4769] ;
– l'ajout de compte utilisateur étrange dans un groupe spécifique à hauts privilèges [évènement 4728];
– le transfert hors heures ouvrées d'une volumétrie anormale depuis le réseau interne à destination de l'Internet ;
– le nettoyage des journaux d'évènements ou la modification de la stratégie les régissant [évènement 4719] ;
– la création en tant qu'objet dynamique (à durée de vie limité) d'un compte d'administration [1] .

Pour compléter ces analyses, le CERT-FR rappelle qu'une autre méthode d'analyse de la sécurité d'Active Directory est fondée sur l'analyse des chemins de contrôle [3] [4]. En cartographiant les relations de chacun des objets du domaine, il est possible d'établir la liste des chemins permettant à un attaquant d'accéder à une ressource. Cette cartographie peut ensuite être utilisée pour contrôler l'étendue effective du pouvoir d'un compte et pour générer les alertes indiquant une déviance vis-à-vis du modèle de sécurité en place.

De manière générale, le CERT-FR recommande de porter une attention particulière aux événements suivants :

- Ticket Service : l'évènement 4769 sur les contrôleurs de Domaines permet le suivi de l'émission de ticket de services, cet évènement est généré à chaque demande d'accès ;
- Audit Special Logon : l'évènement 4964 identifie les nouvelles authentifications venant d'un utilisateur appartenant à un groupe ou compte jugé spécial ;
- Global Object Access Auditing : les évènements de ce type permettent, sur les serveurs sensibles ou l'audit est activé, de contrôler l'accès aux fichiers et aux clés de registres ;
- Audit Detailed File Share : l'évènement 5145, associé au « Global Object Access Auditing », permet d'identifier les accès des utilisateurs à un répertoire partagé identifié comme sensible ;
- Audit Security Group Management : l'évènement 4728 indique qu'un utilisateur est ajouté à un groupe privilégié.

Documentation

- 1. https://techdays.microsoft.fr/programmes/2015/fiche-session.aspx?ID=66ea4044-0534-4b5d-85d4-475643efdf63
- 2. http://blogs.technet.com/b/pie/archive/2014/08/25/metadata-2-the-ephemeral-admin-or-how-to-track-the-group-membership.aspx
- 3. http://www.ssi.gouv.fr/administration/guide/recommandations-de-securite-pour-la-mise-en-oeuvre-dun-systeme-de-journalisation/
- 4. https://www.sstic.org/2014/presentation/chemins_de_controle_active_directory/

3 - Rappel des avis émis

Dans la période du 06 au 11 avril 2015, le CERT-FR a émis les publications suivantes :

- CERTFR-2015-AVI-136 : Multiples vulnérabilités dans Red Hat Enterprise Linux Server
- CERTFR-2015-AVI-137 : Vulnérabilité dans IBM Tivoli Storage Manager FastBack
- CERTFR-2015-AVI-138 : Multiples vulnérabilités dans SCADA Siemens SIMATIC
- CERTFR-2015-AVI-139 : Multiples vulnérabilités dans Apple Macintosh OS X
- CERTFR-2015-AVI-140 : Multiples vulnérabilités dans le noyau Linux d'Ubuntu
- CERTFR-2015-AVI-141 : Multiples vulnérabilités dans Apple Safari
- CERTFR-2015-AVI-142 : Multiples vulnérabilités dans les produits Cisco
- CERTFR-2015-AVI-143 : Vulnérabilité dans les produits Cisco
- CERTFR-2015-AVI-144 : Multiples vulnérabilités dans le noyau Linux d'Ubuntu
- CERTFR-2015-AVI-145 : Multiples vulnérabilités dans IBM Systems Director Storage Control

Gestion détaillée du document

13 avril 2015 version initiale.