

HoneyWISE : stratégie d'exploitation d'honeytokens en environnement Active Directory

Nathan FAEDDA, Augustin TOURNYOL DU CLOS

100-101 Terrasse Boieldieu, 92800 Puteaux, France

`nathan.faedda@wavestone.com`

`augustin.tournyol-du-clos@wavestone.com`

Abstract.

Vingt ans après les premiers projets de recherche d'envergure sur le sujet (HoneyNet project, Project HoneyPot), les stratégies de leurrage numérique sont encore peu adoptées dans le monde de l'entreprise. Pourtant, les avantages de ces systèmes de détection font figure d'exception pour les centres opérationnels de sécurité (SOC) saturés d'alertes : taux de faux positifs quasi inexistant, faible coût de déploiement et de maintenance... L'étude suivante, baptisée HoneyWISE, propose une stratégie concrète de leurrage contre plusieurs attaques de l'Active Directory emblématiques. Le but : permettre à toute organisation de tester simplement l'apport des leurres au sein de cet annuaire d'identités et mesurer leur efficacité sur une matrice de critères dédiés.

Keywords: Active Directory, honeytoken, fake credentials, traps, kerberoasting, deceptive cyber, detection

1 L'ambition du projet HoneyWISE

1.1 Contexte

Les illustrations historiques du concept de leurrage empruntent souvent à l'histoire militaire (l'*Art de la guerre* de Sun Tzu, l'opération Fortitude de 1944) et il faut reconnaître une vertu pédagogique et galvanisatrice à ces exemples. Toutefois ils peuvent également décourager nombre d'acteurs de la cybersécurité qui, en fait d'armées amphibies, disposent d'équipes et moyens techniques limités pour sécuriser les périmètres sous leur responsabilité. De plus, le risque de s'exposer davantage aux menaces, par méconnaissance du sujet, fait souvent craindre un investissement contreproductif aux responsables de la cybersécurité : la possible compromission de ces systèmes inquiète, alors même que la conformité juridique de ces opérations semble douteuse.

En deux décennies l'attirail de techniques, tactiques et procédures (TTPs) de leurrage numérique s'est pourtant considérablement développé, jusqu'à constituer une discipline particulière de la cybersécurité, désignée communément sous la terminologie *Deception* ou *Deceptive cyber*. A la suite des projets open-source nés de l'an 2000, un nombre croissant d'acteurs de tous bords (chercheurs, éditeurs, juristes, journalistes) s'est ainsi approprié cet arsenal, séduit par la promesse d'une détection plus fine des attaques ainsi qu'une compréhension approfondie des groupes d'attaquants. Reste que la relative jeunesse des éditeurs spécialisés, leur exposition médiatique limitée ou très récente¹ -comme leur intégration au reste de l'écosystème cyber (partenariats avec des SOC managés, fonctionnalités nouvelles d'EDR- concourent encore à leur méconnaissance.

Dès lors, le leurrage numérique serait-il réservé à la recherche de pointe et aux pionniers du secteur? Doit-il n'être considéré qu'en dernier recours par le reste des acteurs, une fois tous les autres aspects de la cybersécurité maîtrisés ?

C'est l'enjeu du projet HoneyWISE de montrer qu'une stratégie de leurrage est accessible et profitable à toute organisation. L'Active Directory fournit à cet égard un périmètre d'étude largement adopté par les entreprises, et l'approche *honeypot* des bénéfices de souplesse et robustesse particulièrement rares. L'étude suivante vise donc à présenter les étapes nécessaires pour déployer, maintenir et exploiter ces leurres au sein de l'annuaire d'entreprise, en démontrant leur apport conséquent pour les capacités de détection.

Définitions préalables

Les leurres peuvent varier par leur nature et leurs attributs, selon leur finalité. Ils empruntent l'apparence de ressources couramment exploitées au sein d'un système d'information : des serveurs (*honeypots*) parfois rassemblés en réseau (*honeynet*), des secrets (*honeypot*) ou même des documents (*honeypots* ou *breadcrumbs*). Leur

simple utilisation, ou même consultation, fournit dès lors une alerte de qualité, étant donné qu'aucune utilisation légitime ne leur est prévue dans la vie de l'organisation.

Certains travaux² ont œuvré à préciser la terminologie employée et permettent de justifier l'intérêt spécifique de chaque type de leurres. Lance Spitzner développait ainsi la notion clé d'*honeypot* en 2003 dans un article intitulé « Honeypot : the other honeypot »³, peu après l'invention du terme par son collègue Augusto Paes de Barros. Revenons à ces études pour adopter la définition suivante d'un *honeypot* : il s'agit d'une information numérique dont l'utilisation n'est pas supposée advenir. Ainsi, à la différence de la plupart des honeypots, les *honeypots* ne simulent pas des serveurs, des postes clients, mais représentent des valeurs, des bribes d'information. Les exemples sont donc nombreux : noms de domaine, identifiants de connexion, adresses IP, attribut d'un objet AD... Laissés visibles sur différents supports (identifiants en mémoire sur les terminaux, documents au sein de répertoires partagés, entrée dans une base de données publique), ils servent d'alarmes et parfois également d'intermédiaires « dormants » vers des services et systèmes eux-mêmes factices (*honeypots*).

L'ambition portée par le leurre numérique n'est pas raisonnable si la stratégie associée est trop diffuse. En effet, de la détection de *ransomwares* par la dispersion de *canaryfiles* (documents factices) sur les postes utilisateurs, au détournement d'attaquants vers des environnements hautement réalistes, les postures de *Deception* peuvent être extrêmement variables. La stratégie doit ainsi se proposer d'étudier la finalité du système de *Deception*, les précautions techniques et juridiques⁴ propres aux leurrages et les facteurs d'efficacité du système. Sans cela, l'option choisie ne peut que décevoir ou inquiéter l'équipe en charge de sa gestion : décevoir par la profusion de résultats inexploitable et inquiéter par l'exposition dangereuse d'une architecture défaillante. On peut alors se référer à la classification des leurres selon leur finalité :

- détecter le plus tôt possible les manœuvres suspectes (détection)
- ralentir l'attaquant avec des informations contradictoires (confusion)
- le conduire vers un environnement strictement cloisonné pour surveiller son activité (observation).

Notons que certains leurres (principalement *honeypots*) proposent une dernière option : le niveau d'interaction envisagé, de la simple ouverture de ports (faible) jusqu'à la présentation de véritables systèmes d'exploitation (fort).

Indépendamment de ces divers usages, l'absence de faux-positifs est donc une caractéristique commune de ces dispositifs, relativement élégante à l'heure de l'analyse massive de données. Cette force théorique est toutefois soumise en pratique à la configuration technique des leurres, la finesse de leur dissimulation et surtout leur positionnement vis à vis du SI d'entreprise. Ainsi, des *honeypots* dits « de recherche » seront volontairement exposés très largement (sur internet par exemple) pour observer les tendances et techniques contemporaines. De là, un simple service SSH vulnérable exposé sur internet n'attirera pas uniquement des attaquants ciblant l'organisation mais également une légion de bots opportunistes, des chercheurs en cybersécurité, voire des technophiles soucieux d'avertir son propriétaire ! Nous nous intéresserons donc dans la suite de cette étude aux seuls leurres présents « au sein » du SI d'entreprise (uniquement exposés aux collaborateurs de celle-ci), bien plus susceptibles d'atteindre l'absence de

faux-positifs. L'approche honeypot, moins adaptée à l'observation et la compréhension d'un attaquant, se prête en revanche parfaitement à l'amélioration de la détection. Contrairement au *honeypot* qui se doit de reconstituer un environnement complet et réagir aux sollicitations, le niveau d'interaction attendu avec un secret (*credential*) est minime, favorisant ainsi la crédibilité du leurre à moindre coût. A ceci s'ajoute enfin la robustesse d'un *honeypot* : l'intoxication de l'attaquant se fait via la fourniture d'une information factice qui n'offre aucune prise avec les ressources réelles, empêchant tout rebond sur des systèmes coexistants. Le honeypot n'offre pas cette ultime assurance : son interdépendance avec d'autres systèmes est non seulement gage de crédibilité mais également d'attractivité. S'il n'offre qu'un « cul-de-sac » au sein du SI, il attirera la défiance de l'attaquant, qui sera d'autant plus attentif aux incohérences et singularités du leurre. L'intégration du *honeypot* au sein d'un écosystème entièrement factice (*honeynet*) avec lequel il pourra alors librement communiquer permet de dépasser cet obstacle mais soulève d'autres défis en termes de crédibilité et d'économies.

Plusieurs critères permettent d'évaluer l'apport singulier d'un nouveau leurre, en particulier d'un *honeypot*. Les divers travaux dédiés à la fabrication de leurres partagent ainsi l'essentiel des exigences propres au leurrage (réalisme, attractivité...) mais l'ajout de critères spécifiques aux domaines d'étude⁵ (« les changements de configuration nécessaire sur les terminaux » comme l'ouverture de flux, propres à certains *honeypots*), l'orientation très technique du guide de réalisation⁶ et l'absence de comparaison aux techniques de défense existantes les rendent inadéquats. Cinq axes rassemblent toutefois l'essentiel de l'apport d'un leurre :

1. La **pertinence** (vis-à-vis des solutions de détection existantes)
2. L'**attractivité** (comparée aux ressources à disposition de l'attaquant)
3. Le **risque** (associé à son détournement en cas d'utilisation ou démasquage)
4. La **crédibilité** (associée à la fonction supposée dans son environnement)
5. La **scalabilité** (ou la facilité de déploiement et maintenance massifs)

Ces cinq critères forment donc la matrice d'évaluation de leurs « PARCS » dont les premiers examens porteront sur les *honeypot* de la présente étude. Conscients de la richesse sémantique de certains termes, il nous semblait également nécessaire d'explicitier les attendus de chaque axe afin de guider au mieux l'évaluation suivante :

Critère PARCS	Questionnement associé
Pertinente	« Le honeypot permet-il une détection moins complexe et coûteuse que les autres moyens traditionnels de supervision, tout en assurant un faible taux de faux-positifs ? »
Attractive	« Les gains envisageables sont-ils suffisamment importants pour que l'attaquant utilise ce honeypot, en préférence aux autres informations accessibles sur le support ? »
Risqué	« Le honeypot confère-t-il des privilèges importants à l'attaquant ? Quel usage malveillant pourrait-il en faire s'il déjoue l'artifice ? »
Crédible	« La probabilité de trouver une information de même importance en environnement de production est-elle raisonnable ? »
Scalable	« Quel est l'effort exigé pour l'utilisation du honeypot sur un large périmètre en termes de déploiement et de maintenance ? »

Table 1. Présentation des critères PARCS

1.2 Sélection du périmètre d'étude

L'annuaire Active Directory (AD) est l'outil crucial des systèmes d'information (SI) Microsoft où sont centralisées les informations relatives aux utilisateurs, aux ressources et aux permissions d'une organisation. Tombé entre les mains d'un attaquant, l'AD peut donc mener à la compromission complète d'un SI : en effet, s'il dispose des droits d'administrateur du domaine, l'attaquant est libre de mener toute opération d'exfiltration ou de sabotage au sein de ce dernier. Ainsi, à l'heure où les agences nationales publient des recommandations⁷ spécifiques à l'AD, un nombre grandissant d'acteurs focalise l'effort sur la détection de mouvement latéraux. Le postulat est le suivant : l'infection initiale est aujourd'hui inévitable vue l'étendue des ressources ; en revanche la propagation de la menace aux actifs les plus critiques doit être détectée et remédiée.

Certains travaux ont œuvré à pallier les faiblesses inhérentes de l'AD : Benjamin Delpy révélait ses premières découvertes avec *Mimikatz* en 2007 et Microsoft publiait plusieurs livres blancs sur les attaques « Pass-the-Hash » en 2012 et 2014, année également marquée par la note technique ad-hoc⁸ de l'ANSSI. La publication du « Tier-model »⁹ par Microsoft contribua également à la sécurisation des infrastructures Active Directory en isolant les comptes d'administration.

Enfin, la détection d’attaquants s’avère extrêmement complexe et imprécision dans l’Active Directory, notamment par le fort taux de faux-positifs occasionné. En effet, certaines attaques suivant une méthodologie « Living of the Land » consistant à utiliser les utilitaires déjà présents dans l’environnement pour réaliser les attaques, rendent la détection traditionnelle difficile en camouflant les offensives dans la masse de journaux d’activités légitimes.

En complément d’outils et solutions (PingCastle, Alsid...) dédiés à la cartographie et au durcissement de cet environnement, le déploiement de leurres permet d’aider à la détection en intervenant aux différentes étapes de la matrice ATT&CK¹⁰ illustrant la *kill chain*. Cette étude explicitera donc particulièrement les apports de l’approche honeypot dans la phase de « Credential Access » (T1003, T1081, T1208...) qui correspond au jalon annonceur d’une propagation latérale. L’étude et le démonstrateur HoneyWISE s’appuie sur les travaux précédents consacrés au leurrage dans l’Active Directory : sans ambition d’exhaustivité, il faut citer l’équipe SpecterOps (tout spécialement Will Schroeder, également connu sous le pseudonyme « Harmjoy »¹¹, pour ses travaux sur la détection du Kerberoasting) et le chercheur Nikhil Mittal¹² pour sa synthèse de l’approche des *honeypot*.

2 HoneyWISE

2.1 Démarche du projet

Présenté ci-dessous au sein d’un environnement de test dans le cloud Azure, le projet HoneyWISE est un démonstrateur automatisant le déploiement de trois *honeypots* (cf. 1.2 Définitions préalables) au sein de l’Active Directory afin de démontrer leur efficacité contre un trio d’attaques AD considérées comme « classiques » :

- l’attaque du Cpassword (extraction de secrets dans les Group Policy Preferences)
- l’AS-REP roasting
- le Kerberoasting

La détection de ces dernières est intéressante à plusieurs titres : il s’agit d’attaques désormais largement connues (leur première publication date d’une dizaine d’années environ), dont la prévention peut s’avérer fastidieuse¹³ et qui se situent suffisamment en amont de la *kill-chain* (figure ci-dessous) pour accorder du temps à leur remédiation en cas d’incident avéré.

Le Kerberoasting et le AS-REP roasting sont mutualisés au sein de la matrice ATT&CK dans la même catégorie « T1558.003 – Kerberoasting », alors que l’exploitation de l’attribut « cpassword » se trouve dans la catégorie « T1552.006 – Group Policy Preferences » :

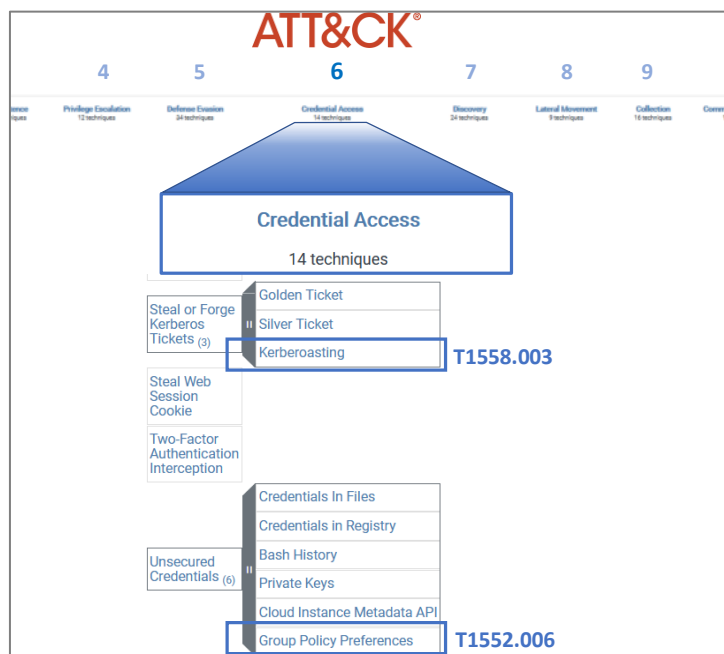


Fig. 1. Les attaques étudiées dans HoneyWISE se situent à mi-chemin des 12 étapes de la *kill-chain* MITRE ATT&CK, en amont du *Lateral Movement* (n°8)

Ce projet n’a donc pas vocation à démontrer les parades possibles contre de nouvelles tactiques, techniques et procédures (TTPs) affectant l’AD mais à fournir un socle expérimental et pédagogique pour éprouver l’apport distinctif du leurrage en détection. Dans une approche progressive, les différentes attaques étudiées correspondent à des niveaux de maturité croissants¹⁴¹⁵ dans les Points de contrôle Active Directory¹⁶, publiés par le CERT ANSSI en juin 2020. Le *Kerberoasting* est ainsi considéré comme un « problème critique », à résoudre dans les plus brefs délais tandis que l’AS-REP représente une « lacune de configuration » suffisante pour entreprendre une action correctrice à court terme. Le *Cpassword*, absent de ce recueil, est en revanche cité dans un rapport du CERT-FR en 2015¹⁷ et considéré comme un contrôle complémentaire dans plusieurs autres référentiels comme celui proposé par l’outil *Pingcastle*¹⁸.

En face de chaque attaque, le processus de création et déploiement d’un honeypot a été automatisé via un script *Powershell* lancé avec les droits d’administrateur du domaine AD étudié. Les caractéristiques propres de chaque honeypot, évaluées dans la matrice PARCS, seront détaillées par la suite. Enfin, chaque honeypot est

directement lié à la plateforme unifiée de détection, jouant le rôle de SIEM global, où a été configurée une règle propre au scénario d'attaque.

Les deux points précédents impliquent donc nécessairement un double questionnement avant d'étudier en détail l'adoption d'HoneyWISE. S'ils n'étaient pas des chantiers en perpétuel renouvellement ils pourraient d'ailleurs être considérés comme des prérequis.

En premier, la nécessaire connaissance de son propre Active Directory : si inventaire il existe, est-on capable d'énumérer les domaines de son AD, d'assurer l'existence d'un responsable pour la gestion de chacun d'entre eux, et le contrôle de leur exposition externe (en particulier sur Internet) ? La maîtrise du périmètre AD joue un rôle puisqu'elle conditionne en effet l'emplacement et la configuration des leurres : d'une part, la criticité du domaine-cible peut induire un déploiement très localisé de leurres ; d'autre part, le honeypot doit être cohérent avec les caractéristiques propres du domaine (type de services présents, taille et nature des groupes AD...). La cartographie de l'Active Directory est donc un facteur de succès dans l'utilisation de leurres, en assurant une configuration et un positionnement pertinents des leurres.

Le second point d'attention doit être à la gestion des vulnérabilités AD (présentées par exemple dans le corpus de l'ANSSI). En effet, la détection d'attaques relativement avancées ne doit pas donner l'illusion d'une protection suffisante de l'Active Directory. Certes, les manœuvres malveillantes étudiées dans ce travail demeurent relativement classiques et ne représentent pas un détournement démonstratif de technicité. Toutefois, à ressources humaines et temps limités, de nombreuses faiblesses « critiques » de l'AD méritent d'être étudiées avant de considérer la détection de TTPs plus fines. La détection de *mimikatz* est ainsi présentée par certains éditeurs de solutions comme un atout décisif pour la sécurité de l'AD à certains responsables de sécurité, alors même que leur organisation souffre parfois de failles bien plus basiques. La consultation des travaux d'inventaire précédemment mentionnés permet de prioriser justement les actions correctives, quitte à envisager le façonnement de leurres sur des scénarios différents de ceux abordés par HoneyWISE.

2.2 Présentation de l'infrastructure HoneyWISE

Une infrastructure a été mise en place afin de supporter la démonstration des trois scénarios retenus précédemment. Cette infrastructure est déployée dans le Cloud Azure en se basant sur le projet *Adaz*¹⁹. Elle se compose de quatre machines virtuelles, hébergeant deux postes de travail Windows 10, un contrôleur de domaine Windows Server 2019 et une dernière machine virtuelle hébergeant la pile ELK (Elasticsearch, Logstash, Kibana), afin de centraliser les événements Windows.

Cette architecture est schématisée ainsi :

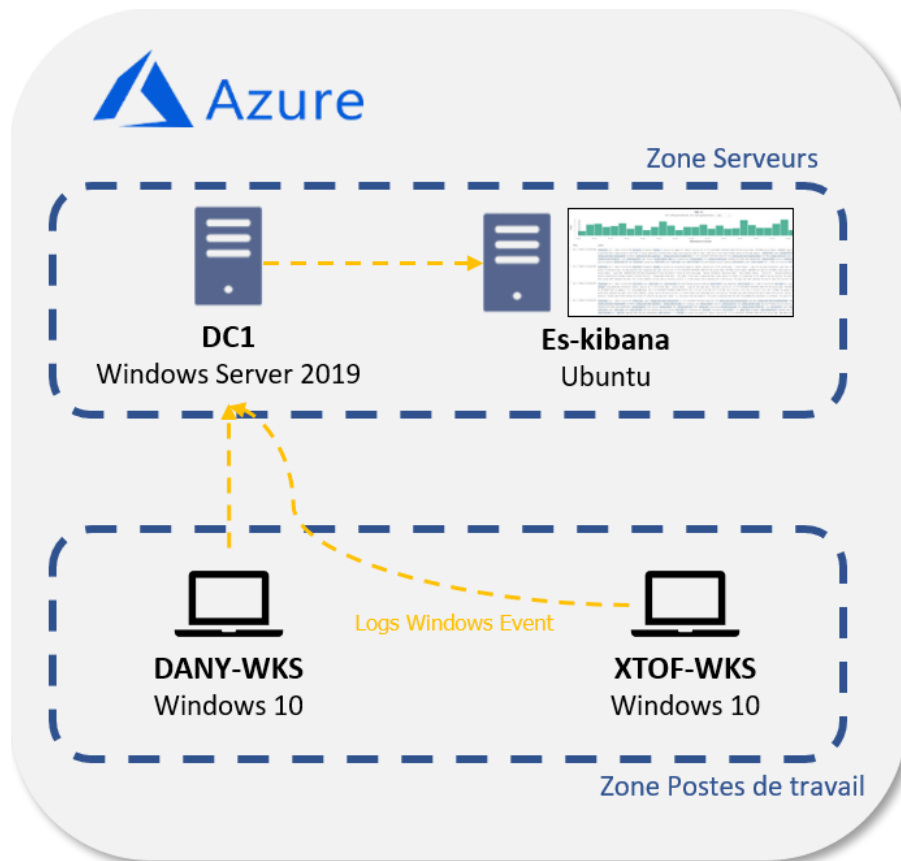


Fig. 2. Infrastructure du projet HoneyWISE

Les postes de travaux déployés sont configurés pour activer la journalisation des événements de sécurité Windows et les envoyer vers le contrôleur de domaine pour centralisation.

Le contrôleur de domaine (DC1) a pour fonction de porter l'Active Directory et est configuré pour activer et remonter les logs appropriés à la pile ELK pour analyse.

Enfin, la pile ELK est déployée sur une machine Ubuntu et sert à collecter, analyser et visualiser les différents événements Windows produits par les postes de travail et le contrôleur de domaine.

L'utilisation du projet Adaz pour le déploiement de cette architecture simplifie grandement le processus puisqu'il a suffi de reprendre l'architecture existante, de déployer une version légèrement modifiée avec Terraform, puis de créer les comptes et règles de détection détaillés dans la suite de ce papier.

2.3 Description des scénarios de détection

Scenario 1 : Cpassword

Objectifs de l'attaque

L'attaque Cpassword (ATT&CK id : T1552.006²⁰) est présentée pour la première fois en 2012 par Emilien Gauralt.²¹

Cette dernière attaque consiste à parcourir le partage réseau SYSVOL, accessible en lecture à l'ensemble du domaine, à la recherche d'identifiants stockés dans une GPO.

Outre certains scripts de connexion ainsi que les données relatives au domaine, le partage SYSVOL contient effectivement les Group Policy Objects (GPO). Ces Group policies sont notamment utilisées pour réaliser des changements distribués sur le domaine, tel que changer le mot de passe de l'administrateur local sur toutes les machines. Les « Group Policy Preferences » (GPP) introduites par Microsoft, permettent ainsi de changer le mot de passe de l'administrateur local d'un grand nombre de machines, ce qui en fait des cibles désignées pour un attaquant introduit sur un poste.

Déroulé de l'attaque

Lors de la création d'une GPP, un fichier XML associé est créé dans le SYSVOL afin de stocker la configuration de la GPP et lorsque des informations de connexion sont incluses dans ce fichier, elles sont chiffrées en AES256. Malheureusement, la clé servant à chiffrer ces informations et partagée par l'ensemble des domaines AD a été rendue publique par Microsoft²² ce qui permet à un attaquant de faire une recherche sur les fichiers XML du SYSVOL contenant le mot « cpassword » qui est l'attribut dans lequel sont stockés les mots de passe chiffrés en AES-256 afin de les déchiffrer et de réaliser ainsi une escalade de privilèges²³.

Prévention de l'attaque

Afin de se prémunir de cette attaque, le patch de sécurité du bulletin *MS14-025*²⁴ doit être appliqué sur tous les systèmes administrant les GPO. Ce patch empêche le remplissage de l'attribut « cpassword » dans une GPP, cependant les fichiers contenant un attribut « cpassword » ne seront pas retirés, il est donc nécessaire d'effectuer une recherche de ces fichiers et de les retirer manuellement. Certains répertoires du volume SYSVOL (*policies_nfrs*) conservent des instantanés de ces Group Policies, souvent ignorés des administrateurs, laissant toute son actualité²⁵ à l'attaque, 6 ans après le patch.

Détection par honeytoken de l'attaque

Pour détecter les tentatives d'exploitation de cette attaque, un leurre d'un genre particulier est créé. Il s'agit d'un nouveau fichier XML associé à une GPO inexistante qui contient un attribut « cpassword » et qui modifie les identifiants de connexions de notre compte piégé. Ce fichier peut être obtenu par exemple à l'aide d'une ancienne GPO

vulnérable qu'il suffira de supprimer en prenant soin de garder le fichier Groups.xml qui servira de leurre une fois l'attribut *cpassword* remplacé par une valeur incorrecte obtenue en réalisant l'inverse du déchiffrement de la clé, à l'aide de l'outil CyberChef par exemple (voir fig. 11).

En auditant les erreurs de connexion au compte exposé dans ce fichier (Windows event log 4625 – « failed logon ») et en filtrant sur le nom du compte leurre, il est alors possible de créer une alerte de sécurité fiable : en effet ce fichier n'est pas associé à une GPO existante et le mot de passe donné ne correspond pas à celui du compte. La recherche de mot de passe dans le SYSVOL renvoie alors le contenu du fichier leurre en résultat :

```
PS C:\Users\christophe> $SYSVOL_Path = "\\honeywise.lab\sysvol"
>>
>> Get-Childitem $SYSVOL_Path -Recurse -File | Select-String -Pattern "cpassword"

\\honeywise.lab\sysvol\honeywise.lab\Policies\{5AC5C2A3-B893-493E-B03A-D6F9E8BCC8CB}\Machine\Preferences\Groups\Groups.xml:4:4:Properties action="U" newName="TRDAdmin" fullName="" description="Standard Admin Account"
cpassword="S330PqYXxR1bjqt9KAUe1MQNt2V1vD/a9SrpHVBd+Sw3+eOM66LeNTUM/hedqHb8//SaMq1vvMsLB+q55Yx5Sw" changelogon="0"
noChange="0" neverExpires="1" acctDisabled="0" subAuthority="RID_ADMIN" userName="Administrator (built-in)" expires=""
/>
```

Fig. 3. Fichier leurre remonté par la recherche de mot de passe dans le SYSVOL

Un attaquant est ensuite en mesure de déchiffrer l'attribut « cpassword » grâce à la clé publiée par Microsoft, en utilisant par exemple cette recette CyberChef :

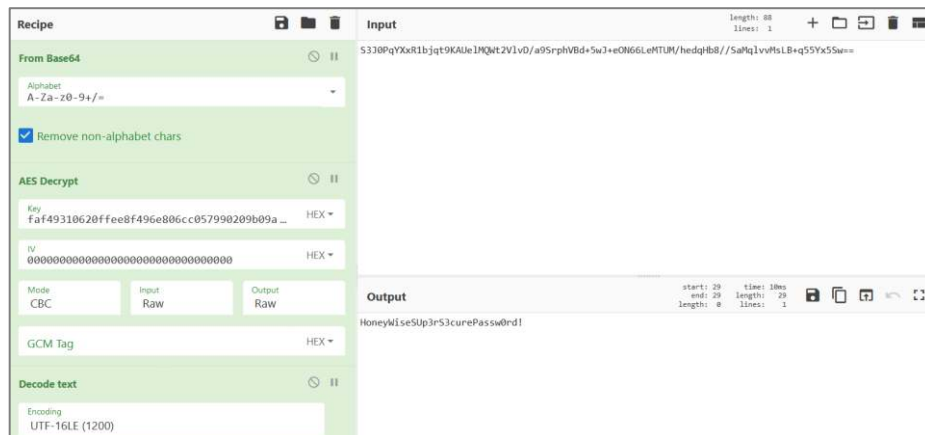


Fig. 4. Déchiffrement du cpassword avec CyberChef

La détection de cette attaque se fait en surveillant la production des événements de sécurité Windows 4625²⁶ afin de détecter un échec de connexion sur le compte leurre. Cette stratégie de détection peut aussi être utilisée pour d'autres leures exposant un faux mot de passe tel que le fait d'indiquer un faux mot de passe de compte dans l'attribut de description.

Evaluation du honeypot

Au regard de la matrice PARCS, ce leurre se démarque par une complexité de contextualisation plus avancée que les autres leures proposés (cela est dû au patch *MS14-025* qui impose donc de créer de toute pièce une fausse GPO sur une installation récente) mais récompensée par une plausibilité d'autant plus grande :

15/20	Critère PARCS pour le honeypot « Cpassword »
Pertinente 3/4	Les moyens actuels de détecter une inspection du répertoire SYSVOL, en quête de fichiers XML sont très laborieux. La création d'un <i>honeypot</i> est d'une grande valeur ajoutée.
Attractive 4/4	Lors d'une intrusion, tout identifiant est intéressant pour l'attaquant, que ce soit dans un contexte de gain de nouveaux privilèges ou tout simplement pour assurer de la résilience dans sa présence sur le réseau.
Risqué 4/4	Pour ce <i>token</i> il est possible d'empêcher le compte de se connecter en définissant les plages horaires de connexion, il est aussi possible d'attribuer un mot de passe robuste au compte et d'exposer de faux identifiants à travers les GPO. La seconde option est mise en œuvre pour ce <i>token</i> .
Crédible 2/4	La présence de mots de passes dans les GPO n'est plus si courante avec le patch <i>MS14-025</i> de Microsoft qui empêche la création de nouvelle GPO vulnérables. Il faut donc s'assurer que la GPO semble antérieure à cette remédiation. Sous réserve du choix du nom et des attribut du compte en fonction du contexte de production dans lequel il est déployé ce <i>honeypot</i> peut être très crédible, en effet dans un environnement chargé par l'existant il n'est pas rare de voir des identifiants dispersés ²⁷ dans des partages avec des accès permissifs ou dans ce cas dans le SYSVOL.
Scalable 2/4	Le déploiement du compte de leurrage peut se faire automatiquement sur plusieurs domaines grâce à des scripts. La contextualisation du compte est particulièrement importante sur ce scénario en particulier en fonction des mesures déjà mises en place. Par exemple si LAPS, permettant la gestion des mots de passes d'administrateurs locaux, est déployé sur le SI, il faudra trouver un scénario suffisamment convaincant pour justifier de la présence d'un tel compte amenant potentiellement à le déployer de manière plus manuelle.

Table 2. Critères PARCS du *honeypot* cpassword

Scenario 2 : AS-REP roasting

Objectifs de l'attaque

L'attaque AS-REP roasting a pour objectif de compromettre un compte de l'Active Directory, afin d'élever ses privilèges sur le domaine ou de renforcer simplement la présence d'un attaquant sur le SI. Cette attaque a la particularité d'utiliser des comportements légitimes rendant sa détection particulièrement ardue.

Déroulé de l'attaque

Afin de bien comprendre le déroulé technique de l'attaque AS-REP roasting, il est nécessaire de bien comprendre le fonctionnement de Kerberos. Ce schéma représente en résumé les 2 premières étapes de la communication Kerberos permettant de s'authentifier auprès d'un service :

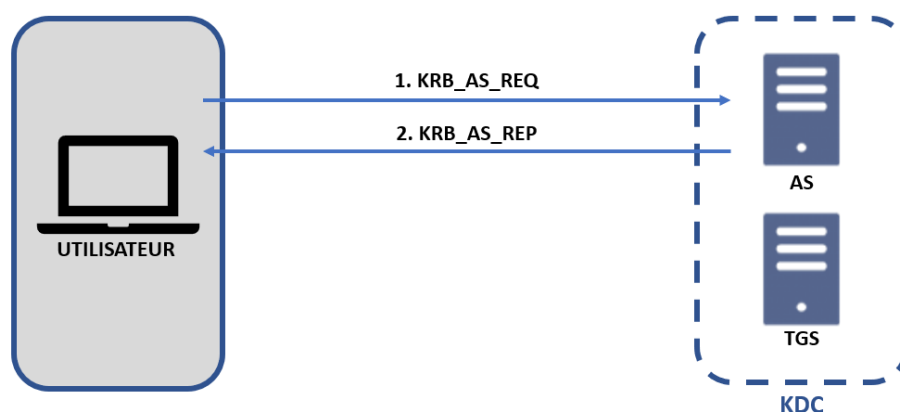


Fig. 5. Présentation des deux premières étapes de l'authentification Kerberos

Tout d'abord un utilisateur cherchant à joindre un service va demander un TGT (Ticket Granting Ticket) à l'AS (Authentication Service) contenant entre autres une clé de session. Pour cela il envoie une demande « KRB_AS_REQ » (étape 1 du schéma) constituée du nom de l'utilisateur et de l'heure précise de la demande, chiffrée avec pour secret le condensat du mot de passe de l'utilisateur. Le KDC (et plus précisément l'AS) va recevoir sa demande, récupérer le condensat du mot de passe de l'utilisateur indiqué dans la demande et enfin la déchiffrer avec ce condensat. En cas de succès, l'identité du demandeur est validée et l'étape 2 s'enclenche : l'AS envoie une réponse « KRB_AS_REP » contenant une clé de session chiffrée avec le condensat du mot de passe de l'utilisateur et un TGT chiffré avec le secret du KDC (Key Distribution Center). En possession d'une clé de session valide et du TGT associé l'utilisateur va pouvoir continuer sa démarche.

En l'état, seul le propriétaire légitime du compte est en mesure de recevoir une réponse « KRB_AS_REP » valide.

Cependant, il existe une option permettant de désactiver la pré-authentification, permettant alors de réaliser l'étape 1 sans connaissance du secret du compte. Ceci permet

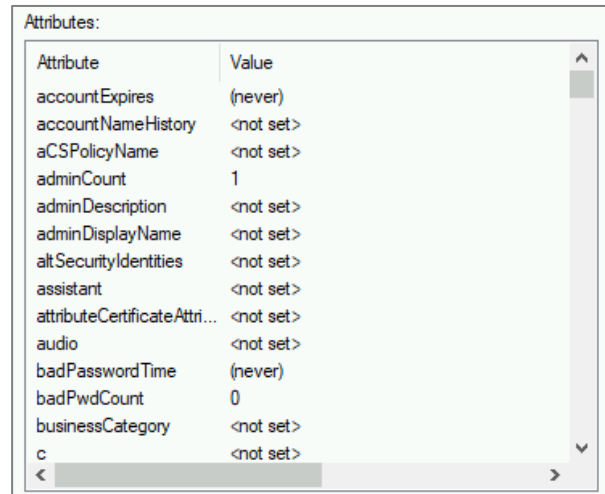
alors à un attaquant de récupérer la clé de session chiffrée avec le condensat du mot de passe de l'utilisateur et de lancer ainsi une tentative de bruteforce hors ligne afin de retrouver le mot de passe de l'utilisateur.

Prévention de l'attaque

Pour se prémunir de cette attaque, il faut veiller à ce que la pré-authentification ne soit désactivée sur aucun compte. Lorsqu'il est nécessaire de désactiver la pré-authentification, il faut alors s'assurer que le secret de ce compte soit robuste et régulièrement renouvelé.

Détection par honeytoken de l'attaque

Pour détecter les tentatives de AS_REP roasting, un compte leurre est créé. Ce compte dispose de droits pouvant attirer un attaquant, d'un mot de passe très robuste et la pré-authentification est désactivée :



Attribute	Value
accountExpires	(never)
accountNameHistory	<not set>
aCSPolicyName	<not set>
adminCount	1
adminDescription	<not set>
adminDisplayName	<not set>
altSecurityIdentities	<not set>
assistant	<not set>
attributeCertificateAttri...	<not set>
audio	<not set>
badPasswordTime	(never)
badPwdCount	0
businessCategory	<not set>
c	<not set>

Fig. 6. Aperçu du honeytoken as-rep au moment de sa création dans l'AD

Ainsi, en surveillant les événements d'authentification sur ce compte (et en particulier les demandes de TGT à travers l'événement de sécurité 4768²⁸), il est possible de générer des alertes très fiables :

event.code	4768
event.created	Oct 1, 2020 @ 16:32:59.415
event.kind	event
event.provider	Microsoft-Windows-Security-Auditing
host.name	DC-1.honeywise.lab
log.level	information
message	<p>> A Kerberos authentication ticket (TGT) was requested.</p> <p>Account Information:</p> <p>Account Name: asreptarget</p> <p>Supplied Realm Name: honeywise.lab</p> <p>User ID: S-1-5-21-3291268791-597879363-724021778-1601</p>
winlog.api	wineventlog
winlog.channel	Security
winlog.computer_name	DC-1.honeywise.lab
winlog.event_data.IpAddress	::ffff:10.0.11.10
winlog.event_data.IpPort	58549
winlog.event_data.PreAuthType	0
winlog.event_data.ServiceName	krbtgt
winlog.event_data.ServiceSid	S-1-5-21-3291268791-597879363-724021778-502
winlog.event_data.Status	0x0
winlog.event_data.TargetDomainName	honeywise.lab
winlog.event_data.TargetSid	S-1-5-21-3291268791-597879363-724021778-1601
winlog.event_data.TargetUserName	asreptarget
winlog.event_data.TicketEncryptionType	0x17
winlog.event_data.TicketOptions	0x40800010
winlog.event_id	4768

Fig. 7. Détection immédiate de la requête de TGT sans pré-authentification sur le compte leurre

Evaluation du honeytoken

Au regard de la matrice PARCS, un tel *honeytoken* dédié au AS-REP roasting s'avère un leurre de qualité :

14/20	Score PARCS du <i>honeypot</i> « As-reps roasting »
Pertinente 4/4	Les alertes générées par ce <i>honeypot</i> sont fiables. En effet, à partir du moment où un TGT est demandé pour accéder à un compte non utilisé ne servant aucun but de production, il apparaît clairement qu'une action malveillante est en cours.
Attractive 2/4	L'attractivité de ce <i>token</i> repose dans le fait que la réalisation de l'attaque ne nécessite pas de privilèges et permet potentiellement d'en gagner tout en étant silencieuse (génération de trafic jugé légitime). Cette attaque est néanmoins plus rare que le <i>kerberoasting</i> , l'attractivité du <i>honeypot</i> s'en trouvera donc légèrement diminuée. Sous réserve que le compte choisis pour leurrer l'attaquant paraisse privilégié et géré par un utilisateur (afin que le mot de passe soit vraisemblablement simple) ce <i>token</i> sera attractif pour un attaquant.
Risqué 4/4	Dans notre exemple un mot de passe de 64 caractères a été défini ce qui n'est pas cassable dans un temps raisonnable.
Crédible 2/4	Le <i>honeypot</i> as-rep est moins facilement crédible que le <i>honeypot</i> kerberoast à cause de la désactivation de la pré-authentification qu'il faut justifier par le contexte.
Scalable 2/4	Le déploiement du compte de leurrage peut se faire automatiquement sur plusieurs domaines grâce à des scripts. Néanmoins pour un leurre efficace, la contextualisation reste primordiale et constituera l'obstacle majeur à un déploiement de masse efficace. Il faudra donc prendre en compte le coût d'apporter cette contextualisation et de la maintenir à jour.

Table 3. Critères PARCS du *honeypot* AS-REP roasting

Scénario 3 : Kerberoasting

Objectifs de l'attaque

Le Kerberoasting (ATT&CK id : T1558.003) est une attaque sur l'Active Directory, qui permet à un attaquant de compromettre des comptes de service, qui possèdent très souvent des forts privilèges sur des serveurs ou l'AD, à partir d'un compte du domaine quelconque. Présentée en septembre 2014 par Tim Medin à l'occasion de la conférence DerbyCon²⁹, elle a été par la suite largement saluée pour ses atouts : en synthèse, le bruteforce offline (pas d'échec de logon) d'un ticket Kerberos recelant le secret d'un compte de service, sans devoir envoyer un seul paquet à ce service ni même être administrateur local du poste compromis.

Sa large adoption par la communauté cyber est encore sensible au moment de notre étude : après de nombreuses versions (dont une notable en novembre 2016³⁰, retirant la nécessité d'utiliser mimikatz avec les outils et framework Powersploit et Powershell Empire pour extraire les secrets et permettre le déchiffrement), son utilisation continue d'être démocratisée³¹³². Ainsi en juin 2020, son utilisation est facilitée sur CrackMapExec avec l'ajout du protocole LDAP, permettant la reconnaissance nécessaire au Kerberoasting.

Déroulé de l'attaque

Tout comme l'AS-Rep roasting, le Kerberoasting détourne le fonctionnement natif de Kerberos afin de réaliser une attaque. Ce détournement se fait sur les étapes 3 et 4 de l'authentification Kerberos présentées par le schéma suivant :

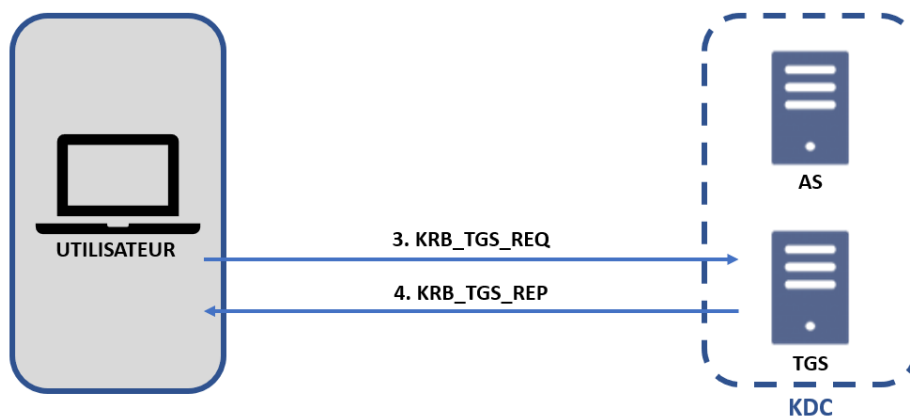


Fig. 8. Présentation des étapes 3 et 4 de l'authentification Kerberos

Le client, après s'être authentifié une première fois auprès du KDC lors des étapes 1 et 2, dispose d'un ticket initial baptisé « TGT » (Ticket Granting Ticket) avec lequel il peut désormais légitimement demander l'accès à tous les services du domaine.

La demande d'un service particulier via son alias dans l'AD (SPN pour Service Principal Name) se fait auprès du KDC à l'étape 3 avec la demande « KRB_TGS_REQ » pour laquelle le TGT est utilisé. Un second ticket correspondant à ce service, le « TGS » (Ticket Granting Service) est alors délivré avec la réponse 4 « KRB_TGS_REP ».

Seulement, afin que le service puisse reconnaître l'origine et la validité de ce ticket TGS, le DC en chiffre une partie avec le condensat NT du service, qu'il est seul à partager avec lui. A cette étape, l'utilisateur dispose donc d'un ticket chiffré (généralement en AES256 ou RC4/NTLM) avec le condensat du mot de passe du compte de service auquel le SPN est associé, le tout sans avoir eu à contacter un seul instant ce dernier grâce à la centralisation fonctionnelle de Kerberos. Le fonctionnement de Kerberos offre ainsi la possibilité à n'importe quel utilisateur de requêter un ticket pour n'importe quel service, la gestion des droits s'effectuant par la suite au niveau du service.

Concluons sur ce fonctionnement inhérent à Kerberos dont l'attaque naît : si le SPN est lié à un compte machine, comme c'est souvent le cas, la clé chiffrée est alors considérée comme inattaquable à cause de la gestion automatique des comptes machines par l'AD (secrets aux caractères aléatoires, changements réguliers). En revanche, si le SPN est lié à un compte utilisateur, le mot de passe a de fortes probabilités d'être nettement plus court, construit logiquement et donc attaquant. En outre, il n'est pas rare que l'utilisateur attaché au SPN soit administrateur de la machine spécifiée dans le SPN et soit également membre de groupes à forts privilèges. Si enfin la suite de chiffrement RC4 est utilisée (pour la compatibilité avec un logiciel ancien ou une version datée de Kerberos par exemple) alors son déchiffrement est considéré comme plus aisé³³.

En conclusion il s'agit du même principe d'attaque que AS-REP roasting mais ici c'est le contenu de la réponse « KRB_TGS_REP » qui est soumis à une tentative de bruteforce, afin de trouver le mot de passe du compte associé au service.

Prévention de l'attaque

Des méthodes de prévention existent pour lutter contre le Kerberoasting mais sont rarement mises en application. On peut toutefois citer les principales :

- Adopter une hygiène de mots de passe complexes (générés aléatoirement, minimum 30 caractères, régulièrement modifiés) pour les comptes auxquels sont assignés des SPN
- Recourir aux “*group Managed Services Accounts*” (gMSA)³⁴ dont les mots de passe sont générés et changés par l'Active Directory (comme pour les comptes-machines)
- Auditer régulièrement les SPNs et leur attribution à des comptes utilisateurs sensibles (les membres du groupe « *Domain Admin* » ne devraient ainsi pas être des comptes de service, et n'avoir aucun SPN attribué)
- Éliminer l'usage de protocoles de chiffrement non-sécurisés (les attributs de comptes permettent de limiter le recours au RC4 et favoriser AES128 et AES256).

Détection traditionnelle de l'attaque

Les méthodes de détection traditionnelles du Kerberoasting ne sont pas plus aisées. L'attaque repose en effet sur des processus et outils courants: un utilisateur requiert

quotidiennement plusieurs dizaines de TGS pour accéder à des services variés, ce qui noie toute tentative de Kerberoasting dans une masse de requêtes légitimes. Les « *Windows security event logs* » identifiant 4769 (qui correspond à l'évènement « *TGS requested* ») peuvent toutefois être filtrés selon :

- L'utilisation du chiffrement RC4 : les requêtes TGS incluant un « *encryptionType* » de 0x17 peuvent être considérées comme suspectes, compte tenu de la faiblesse de ce mécanisme. Certains systèmes legacy (serveurs inférieurs à Windows 2008R2) requièrent parfois encore le recours au chiffrement DES, tout aussi déconseillé : le recours aux *encryptionType* de '0x1', '0x2', '0x3' doivent donc être également supervisés.
- Leur fréquence dans un court laps de temps pour un même utilisateur, hautement improbable pour un utilisateur commun.
- Le type de compte supervisé : les noms de services incluant un « \$ » indiquent un compte machine et non utilisateur. Ceux-ci, raisonnablement inattaquables par Kerberoasting, peuvent être éliminés du champ de supervision.

Toutefois, force est de constater qu'en dépit de ces filtres successifs, la supervision traditionnelle du *Kerberoasting* déclenche encore de nombreux faux-positifs. Plusieurs travaux récents³⁵³⁶ soulignent l'imperfection de ces modes de détection tandis que la *Deception* n'apporte, en complément, aucune fausse alerte et prouve toute son efficacité lors des tests en conditions réelles³⁷. Le tableau suivant, tirée d'une étude³⁸ de la Czech Technical University de Prague, permet de comparer les différentes façons de détecter le *kerberoasting* :

Scenario	Total Detected Events	True positives		Fales positives	
		Count	%	Count	%
D01 - Possible Kerberoasting activity	13	7	58.85	6	46.15
D02 - Excessive service ticket requests from one source – filtering krbtgt account	326	7	2.15	319	97.85
D02 - Excessive service ticket requests from one source – add weak encryption types	10	7	70	3	30
D02 - Excessive service ticket requests from one source – filter \$ accounts	5	5	100	0	0
D04 - Detecting Kerberoasting with a honeypot	7	7	100	0	0

Fig. 9. Comparaison de l'efficacité entre différents scénarios de détection

Détection par honeytoken de l'attaque

Le *honeytoken* déployé contre le *Kerberoasting* consiste donc en un compte AD auquel on attribue un ServicePrincipalName (*MSSQLSvc/DC1.honeywise.lab*) pour l'exposer à la reconnaissance de l'attaquant.

La première partie du SPN -ici 'MSSQLSvc'- désigne d'ordinaire le type de service utilisé, tandis que la seconde reprend le nom du serveur hébergeant le service. Ce nom peut être le FQDN, comme ici 'DC1.honeywise.lab'.

Notons que le service utilisé (SQLServer) contribue à la crédibilité du compte, puisque ce genre de service est d'ordinaire lié à un compte utilisateur et non pas un compte machine.

Attribute	Value
accountExpires	(never)
adminCount	1
badPasswordTime	10/30/2020 4:25:21 PM Central European St
badPwdCount	3
cn	Admin_MSSQLSvc
codePage	0
countryCode	0
distinguishedName	CN=Admin_MSSQLSvc,CN=Users,DC=honeywise.lab
dSCorePropagationD...	10/30/2020 2:52:43 PM Central European St
instanceType	0x4 = (WRITE)
lastLogoff	(never)
lastLogon	10/30/2020 4:24:27 PM Central European St
lastLogonTimestamp	10/30/2020 2:54:07 PM Central European St
logonCount	9

Attribute	Value
name	Admin_MSSQLSvc
objectCategory	CN=Person,CN=Schema,CN=Configuration,
objectClass	top; person; organizationalPerson; user
objectGUID	65a64e9b-afce-44f8-83eb-9511abb93fcd
objectSid	S-1-5-21-3291268791-597879363-72402177
primaryGroupID	513 = (GROUP_RID_USERS)
pwdLastSet	10/30/2020 2:46:33 PM Central European St
replPropertyMetaData	AttID Ver Loc USN Org.DSA
sAMAccountName	Admin_MSSQLSvc
sAMAccountType	805306368 = (NORMAL_USER_ACCOUNT)
servicePrincipalName	MSSQLSvcDC1.honeywise.lab:1433
userAccountControl	0x10200 = (NORMAL_ACCOUNT DONT_
userPassword	SqlServer1!
userPrincipalName	Admin_MSSQLSvc@honeywise.lab

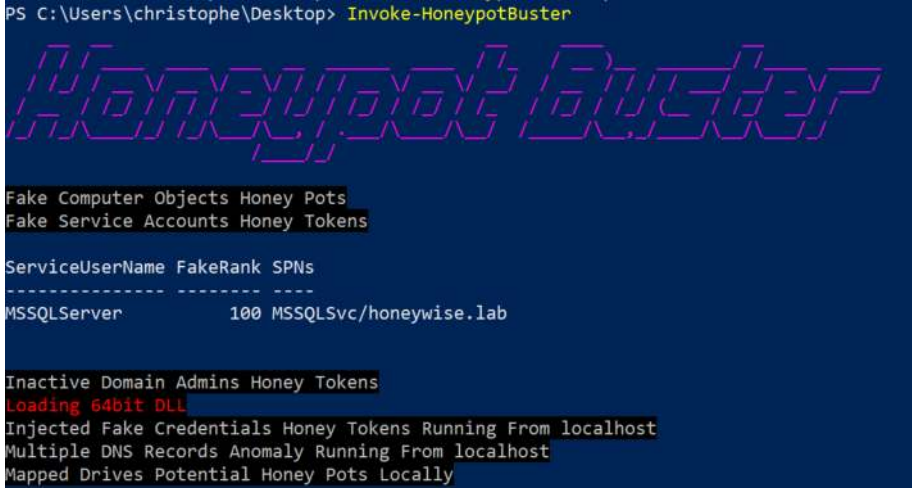
Fig. 10. Aperçu du honeytoken dédié au Kerberoasting.

Parmi les nombreux champs d'attribut disponibles, quelques-uns sont susceptibles d'assurer la crédibilité et l'attractivité du leurre :

- le *DomainSID-RID* doit être respectueux de celui conventionnellement utilisé dans l'entreprise
- l'attribut *whenCreated* doit être assez ancien pour ne pas éveiller les soupçons
- les attributs *lastLogon*, *lastlogontimestamp* peuvent ne pas être très récents pour faire croire à un compte dormant
- les attributs *badPwdCount*, *badPasswordTime* doivent alors être mis à jour en conséquence : s'il s'agit d'un compte utilisateur, l'absence complète d'erreur d'authentification serait étrange : l'attribut « *badPasswordTime* » ne doit pas être vide.
- le champ *AdminCount* à '1' peut indiquer son appartenance à un groupe à forts privilèges : Domain Admin, DnsAdmins, DHCP Admin, Entreprise Admin, Schema Admin
- une revue finale de la cohérence entre les attributs eux-mêmes : un « *pwdlastset* » ne devrait pas être antérieur à l'ancienneté du domaine de l'entreprise, ni du champ « *whencreated* »³⁹

Le recours à l'outil *HoneyPotbuster* permet d'évaluer la crédibilité générale du *honeytoken*, par l'examen détaillé de ses attributs : s'il parvient à détecter le leurre, le détail de ses critères d'évaluation⁴⁰ peut s'avérer instructif pour achever sa configuration.

Le précédent *honeytoken* peu après sa création est immédiatement détecté par Honey-potBuster :



```

PS C:\Users\christophe\Desktop> Invoke-HoneypotBuster

Honey-potBuster

Fake Computer Objects Honey Pots
Fake Service Accounts Honey Tokens

ServiceUserName FakeRank SPNs
-----
MSSQLServer          100 MSSQLSvc/honeywise.lab

Inactive Domain Admins Honey Tokens
Loading 64bit DLL
Injected Fake Credentials Honey Tokens Running From localhost
Multiple DNS Records Anomaly Running From localhost
Mapped Drives Potential Honey Pots Locally
  
```

Fig. 11. Le *honeytoken* est détecté par Honey-potBuster

Le « FakeRank » est dépendant de quelques variables aisément manipulables (Logon-count, *Lastlogontimestamp*, appartenance à des « Powerfulgroups », *AdminCount*...) : quelques retouches d'attributs permettent d'échapper au radar d'Honey-potBuster. Après ces modifications le *honeytoken* est bien plus furtif :



```

PS C:\Users\christophe\Desktop> Invoke-HoneypotBuster

Honey-potBuster

Fake Computer Objects Honey Pots
Fake Service Accounts Honey Tokens
Inactive Domain Admins Honey Tokens
Loading 64bit DLL
Injected Fake Credentials Honey Tokens Running From localhost
Multiple DNS Records Anomaly Running From localhost
Mapped Drives Potential Honey Pots Locally
  
```

Fig. 12. Après modifications des attributs la détection du *honeytoken* échoue

Du côté de l'attaquant, l'outil « Invoke-Kerberoast » disponible comme module sous le *framework Empire* ou encore comme script indépendant, permet de réaliser l'attaque

de façon automatisée en réalisant la recherche de SPN et le dump des condensats récupérables.

Dans la capture suivante, un exemple sur l'infrastructure du projet est réalisé :

```
PS C:\Users\christophe> Invoke-Kerberoast -Verbose
VERBOSE: [Get-DomainSearcher] search string: LDAP://DC=1.honeywise.lab/DC=honeywise,DC=lab
VERBOSE: [Get-DomainUser] Searching for non-null service principal names
VERBOSE: [Get-DomainUser] filter string: (&(samAccountType=805306368)(servicePrincipalName=*))

TicketByteHexStream :
Hash : $krb5tgt$MSSQLSvc/honeywise.lab:B902383ACB48E0E17A0CC103984D13055833CB226708768DF06F69F503914CE
2A600CCA0A428767F4E3585E0600458F17F84C10CAC9B36CD5242714FAF2CB67E1D98805E9080FCA592EDF8C1C988AC5
F86D53C0157588382F8CD4A0A12A968FE8B28D8811A490FC03626AA44A75E261CF140418040288E0D04EDE57816007EDE
898EAC1991F2F84B32D5F96089089A08B11DAAB874855D42A28463995688152C0FAB4F54C8EABAE563AFF64480470B4
384622580A065A00239B21E95E8FD7F5269C9A92A2898F20AF1E9D4FC9DE49780163CA9C835FDB3A4C656FB298FE3A5A
69EA3D11E3F70D8F86DA5A428AE3D6CA56A771A5520896B8BCED63CF32D16A6CA710842A90028012B49FBF02B787D47C
E6F60CF511E7A1AD363658F4E5C8639B384F004DBED4AD2F76D8484F32A4E1141C53E9098351B1C41E00CEFC3489525
535DC525E251E91EBD2CD1DD36F8536431AA32A36269409868FD0A8821C4A2FA4F956455C8DE77516987CC3CB870CDDF
8373E9AF97CADD66E76C445ED0C44014309E58B80F014A8D645C83190FF618131289118E4F6BD686169AE628907C09EF
E415F337AEDA9875F84E3B6243BC981036EF284922F8BD657C0A373937B8C83879AE18AC28F14ADD219561666030677
7A650389753202A1954908C436C69D735E251E6E2D7A580D4327A6813531B1F1588EB61158682903A89590783081B8F01
3F326685A6C2F33473863F6BEAD54CD9D47E3D5256CA3ABD2A2F0241F8279221C4EA90CD953FD09F9BCD58CD9176BAD3
D608466046458C7E9DCDD6582EEA888CC5C46F5DB427D4AC4D9326EA3393FD561AB087AD09898B11844E10624A6A7B
231F858681791D0F3B70EF46AB6C246D128B002461539C515745F9A2F4138B2E48606584614041F202862247D3AADFD
E88610E2989F955AC6096628344F4A9521A2EB5FDD084626280A5737189A5D88D77283FD28B0808A084DA820F810524622
822076A6278B7F3430BFF9E3C69729ACD61F4D9EE36DAE37B8F8638D3A9B2A8831EF78DF3C0724078A59DC682C9E6B
CF63FE60009C0D72F0A1F999A5B15E0AF168283987048304FAFB0F1255F9AD877AE9FE310E0BF43383C0A4E3FEF03A
558B54335E8F205176A7A8F5FC363F63718560D538E8004DBFCE3E32709FF2234668DA251F928A02386A7E4390CCEBF
9ED4AD42CD77DFC54D6344D082A4E8A25A018D296FC46743ED20FE04A140D64F67298AF0164E35A8E4682C9DA74E798
C7D89A80F47E401E019A82287F9026532E6A6B79F9350004A1D7F28EB2758712DA076E10C958040F30102F36960273
85BA0FA705D847881F92B1330F45E373DCF6F142C7D0ED573C28BA080B70F3E951A73543079C6D68CE9B1089B94047F3
7641963863811B449F1CB8968D080335CA32B147C4A12110A87CE3DDC4A3F5A15FCB888ABA25BA2F0B89F84F418D68FD
24FF7969BA054CEA0F14DE206DCC6EF11909F42B0AEB36369A9AF8EB377FA9F084663EABFA235769A42114B4849602DD
BF95779921E9D5C78C7E02953FF6AC45C8961A7C33A694908AA6516C05E4E643BB33F226F3DE906FD9E44

SamAccountName : kerberoast-target
DistinguishedName : CN=kerberoast-target,CN=Users,DC=honeywise,DC=lab
ServicePrincipalName : MSSQLSvc/honeywise.lab

PS C:\Users\christophe>
```

Fig. 13. Récupération du condensat Kerberos 5 TGS-REP associé au compte de service *honey-token*

Enfin, la détection d'un attaquant tentant d'exploiter le service leurre se fait grâce à l'événement de sécurité Windows 4769⁴¹. En effet cet événement se déclenche lors d'une demande de ticket de service Kerberos, il suffit alors de filtrer ces demandes et de sonner l'alerte si une demande est faite vers le service leurre, puisque pour rappel il n'est lié à aucun service réel et ne devrait donc jamais recevoir de requête d'accès légitime.

Evaluation du honeypot

Au regard de la matrice PARCS, un tel *honeypot* dédié au Kerberoasting s'avère un leurre de très bonne facture :

16/20	Score PARCS du <i>honeypot</i> « Kerberoasting »
Pertinente 4/4	Les alertes générées par ce <i>honeypot</i> sont fiables. En effet, à partir du moment où un ticket TGS est demandé pour accéder à service non-utilisé et inexistant, il apparaît clairement qu'une action malveillante est en cours.
Attractive 3/4	L'attractivité de ce <i>token</i> repose dans le fait que la réalisation de l'attaque ne nécessite pas de privilèges et permet potentiellement d'en gagner tout en étant silencieuse (génération de trafic jugé légitime). Sous réserve que le compte choisi pour leurrer l'attaquant paraisse privilégié et géré par un utilisateur (afin que le mot de passe soit vraisemblablement simple) ce <i>honeypot</i> est donc fortement attractif.
Risqué 4/4	Dans notre exemple un mot de passe de 64 caractères a été défini ce qui n'est pas cassable dans un temps raisonnable.
Crédible 3/4	Sous réserve du choix du nom et des attributs du compte en fonction du contexte de production dans lequel il est déployé, l'attaque se basant sur un fonctionnement normal de Kerberos, il ne sera pas étonnant de pouvoir la réaliser. La crédibilité est donc forte.
Scalable 2/4	Le déploiement du compte de leurrage peut se faire automatiquement sur plusieurs domaines grâce à des scripts. Néanmoins pour un leurre efficace, la contextualisation reste primordiale et constituera l'obstacle majeur à un déploiement de masse efficace. Il faudra donc prendre en compte le coût d'apporter cette contextualisation et de la maintenir à jour.

Table 4. Critères PARCS du *honeypot* Kerberoasting

Notons que certaines manipulations plus avancées permettraient de gagner davantage en attractivité et crédibilité :

- Afin de présenter un attribut *whenCreated* suffisamment ancien, on peut considérer le réemploi d'un compte existant. Toutefois cette solution risque de porter atteinte à la précision du *honeypot* en déclenchant des faux positifs, si ce compte est lié à une inspection ou une utilisation ancienne ou intermittente.
- L'utilisation de la technique *DCShadow*, présentée par Vincent Letoux et Benjamin Delpy en 2018⁴² permet la modification d'un certain nombre d'attributs et données liés aux objets du domaine. On peut ainsi modifier des champs intéressants, comme le « pwdlastset »⁴³.

3 Conclusion

Lance Spitzner écrivait en 2003 pour dénoncer le relatif abandon de la détection au profit des projets de connaissance des attaquants (*Threat intelligence*) : « A titre personnel, le concept de *Deception* me semble avoir éclipsé la valeur des honeypots, et leur vraie valeur tient dans la détection »⁴⁴. Plus d'une quinzaine d'années après, cet effort semble reprendre : un nombre croissant d'éditeurs (start-ups spécialisées, SIEM et XDR...) intègrent des fonctionnalités *honeypot* dans leurs solutions, s'associant par-là aux efforts continus des chercheurs pour infliger aux attaquants des leurres toujours plus subtils et chronophages (mise en valeur de faux comptes sur Bloodhound⁴⁵, déploiement d'*honeypots* polyvalents⁴⁶...).

Un vaste travail de pédagogie est toutefois nécessaire pour initier les organisations à ces stratégies de détection et convaincre les SOC de leur grande maniabilité. Parmi la kyriade de prototypes et laboratoires dédiés à la *Deception* il semblait utile de fournir une maquette rapidement opérationnelle à des organisations de toute taille : c'est à cet objectif que s'est attaché le projet HoneyWISE en synthétisant l'apport des leurres contre un trio d'attaques AD classiques. Les résultats de cette étude sont désormais disponibles à l'analyse et nous nourrissons l'espoir que la matrice d'évaluation servira à l'examen d'autres leurres.

Alors que de nombreuses entreprises se tournent résolument vers le Cloud pour y déployer leurs infrastructures, il semble évident que la *Deception* trouvera également dans ces écosystèmes un nouveau théâtre d'opération. La sécurité des ressources y repose fondamentalement sur l'identité des utilisateurs et les systèmes de détection natifs des fournisseurs cloud, bien plus que sur les remparts périmétriques. Ainsi, les futures ramifications du projet HoneyWISE iront certainement en ce sens, où s'engagent déjà de multiples initiatives prometteuses.

4 Références

- 1 Anna Belak, Augusto Barros, <https://www.gartner.com/en/documents/3939890/solution-comparison-for-six-threat-deception-platforms>
- 2 Fabien Pouget, Hervé Debar, Marc Dancier : Research Report RR-03-081 “Honeypot, Honeytoken, Honeytoken: Terminological issues”, Institut Eurecom (2003)
- 3 Lance Spitzner : « Honeytoken, the other honeypot », 2003, <https://community.broadcom.com/symantec-enterprise/communities/community-home/librarydocuments/viewdocument?DocumentKey=74450cf5-2f11-48c5-8d92-4687f5978988&CommunityKey=1ecf5f55-9545-44d6-b0f4-4e4a7f5f5e68&tab=librarydocuments>
- 4 Marie Barel : Conférence « Honeypots, un pot-pourri juridique ». SSTIC04 (2004).
- 5 Nikhil MITTAL : https://fr.slideshare.net/nikhil_mittal/forging-trusts-for-deception-in-active-directory
- 6 Sean METCALF : <https://www.hub.trimarcsecurity.com/post/the-art-of-the-honeypot-account-making-the-unusual-look-normal>
- 7 ANSSI : « Recommandations de sécurité relatives à Active Directory », 10/09/2014
- 8 ANSSI CERT-FR : « Points de contrôle Active Directory », 02/02/2020
- 9 Microsoft website : <https://docs.microsoft.com/en-us/windows-server/identity/securing-privileged-access/securing-privileged-access-reference-material>
- 10 Matrice ATT&CK, <https://mitre-attack.github.io/attack-navigator/enterprise/>
- 11 <https://adsecurity.org/?p=3513>
- 12 <http://www.labofapenetrationtester.com/2018/10/deploy-deception.html>
- 13 <https://attack.stealthbits.com/cracking-kerberos-tgs-tickets-using-kerberoasting>
- 14 https://www.cert.ssi.gouv.fr/uploads/guide-ad.html#spn_priv
- 15 https://www.cert.ssi.gouv.fr/uploads/guide-ad.html#kerberos_properties_preauth
- 16 <https://www.cert.ssi.gouv.fr/dur/CERTFR-2020-DUR-001/>
- 17 <https://www.cert.ssi.gouv.fr/actualite/CERTFR-2015-ACT-046/#SECTION00020000000000000000>
- 18 https://www.pingcastle.com/PingCastleFiles/ad_hc_rules_list.html - “Find Password GPO”
- 19 <https://github.com/christophetd/Adaz>
- 20 <https://attack.mitre.org/techniques/T1552/006/>
- 21 esec-pentest.sogeti.com/post/Exploiting-Windows-2008-Group-Policy-Preferences
- 22 https://docs.microsoft.com/en-us/openspecs/windows_protocols/ms-gppref/2c15cbf0-f086-4c74-8b70-1f2fa45dd4be
- 23 <https://attack.stealthbits.com/plaintext-passwords-sysvol-group-policy-preferences>
- 24 <https://support.microsoft.com/en-au/help/2962486/ms14-025-vulnerability-in-group-policy-preferences-could-allow-elevati>
- 25 <https://twitter.com/mikeloss/status/1322317351065284608?s=20>
- 26 <https://docs.microsoft.com/fr-fr/windows/security/threat-protection/auditing/event-4625>
- 27 [https://github.com/wavestone-cdt/AD-security-workshop/tree/master/8%20\(very\)%20low%20hang-ing%20fruits%20and%20how%20to%20smash%20those%20attack%20paths](https://github.com/wavestone-cdt/AD-security-workshop/tree/master/8%20(very)%20low%20hang-ing%20fruits%20and%20how%20to%20smash%20those%20attack%20paths)
- 28 <https://docs.microsoft.com/fr-fr/windows/security/threat-protection/auditing/event-4768>
- 29 <https://www.irongeek.com/i.php?page=videos/derbycon4/t120-attacking-microsoft-kerberos-kicking-the-guard-dog-of-hades-tim-medin>
- 30 <https://www.harmj0y.net/blog/powershell/kerberoasting-without-mimikatz/>
- 31 <https://www.slideshare.net/harmj0y/derbycon-2019-kerberoasting-revisited>
- 32 <https://swarm.ptsecurity.com/kerberoasting-without-spns/>
- 33 https://www.researchgate.net/publication/221429668_Extracting_Kerberos_passwords_through_RC4-HMAC_encryption_type_analysis
- 34 <https://docs.microsoft.com/en-us/windows-server/security/group-managed-service-accounts/group-managed-service-accounts-overview>
- 35 <https://dspace.cvut.cz/bitstream/handle/10467/83217/F8-BP-2019-Kotlaba-Lukas-thesis.pdf?sequence=-1&isAllowed=y>
- 36 <https://www.insticc.org/Primoris/Resources/PaperPdf.ashx?idPaper=89550>
- 37 https://orange.cyberdefense.com/global/wp-content/uploads/sites/12/2020/04/Orange_Cyber_Lessons_learned_from_hacking_ourselves_Threat_detection_Whitepaper.pdf

-
- 38 <https://www.insticc.org/node/TechnicalProgram/icissp/2020/presentationDetails/89550>
 - 39 <https://apt29a.blogspot.com/2019/11/deploying-honeytokens-in-active.html>
 - 40 <https://github.com/JavelinNetworks/HoneypotBuster/blob/master/Invoke-HoneypotBuster.ps1>
 - 41 <https://docs.microsoft.com/fr-fr/windows/security/threat-protection/auditing/event-4769>
 - 42 <http://lib.21h.io/library/VRU3GD7Z/download/DYERTWJV/Buehat%20IL%20v2.3.pdf>
 - 43 <https://www.labofapenetrationtester.com/2018/04/dcshadow.html>
 - 44 <https://www.labofapenetrationtester.com/2018/04/dcshadow.html>
 - 45 <https://apt29a.blogspot.com/2019/11/deploying-honeytokens-in-active.html>
 - 46 <https://docs.canarytokens.org/guide/#what-are-canarytokens>