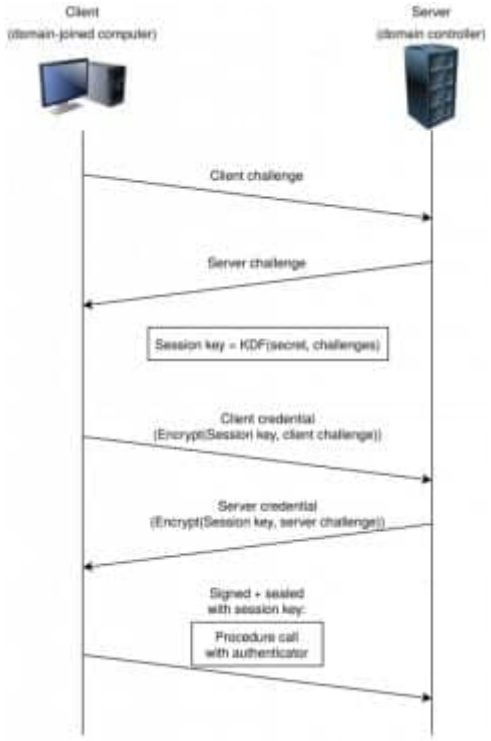
Zerologon : cette faille critique met à mal Active Directory

Devenir admin « en un clic » sur les domaines Active Directory ? C'est la menace que laisse planer Zerologon.

L'éditeur néerlandais Secura a donné ce nom à une faille (CVE-2020-1472) dotée du score de criticité maximal sur l'échelle CVSS.

Le problème se trouve dans le protocole Netlogon. Plus précisément au niveau de son schéma d'authentification. Celui-ci repose sur la capacité d'un client et d'un serveur à se prouver l'un l'autre qu'ils connaissent un même secret. En l'occurrence, un hash du mot de passe du client.

La session Netlogon s'amorce côté client. La première étape consiste à échanger, avec le serveur, des *nonces* de 8 octets. L'un et l'autre calculent une clé de session en mélangeant ces *nonces* avec le hash. Le client utilise cette clé pour créer un identifiant. Le serveur procède également ainsi et valide la connexion s'il obtient le même résultat.



Durant cette procédure, client et serveur peuvent négocier le chiffrement et la signature des messages ultérieurs. Lorsque le chiffrement est désactivé, tous les appels Netlogon qui sous-tendent des actions importantes doivent tout de même contenir une valeur d'authentification également calculée à partir de la clé de session.

Défaut de chiffrement

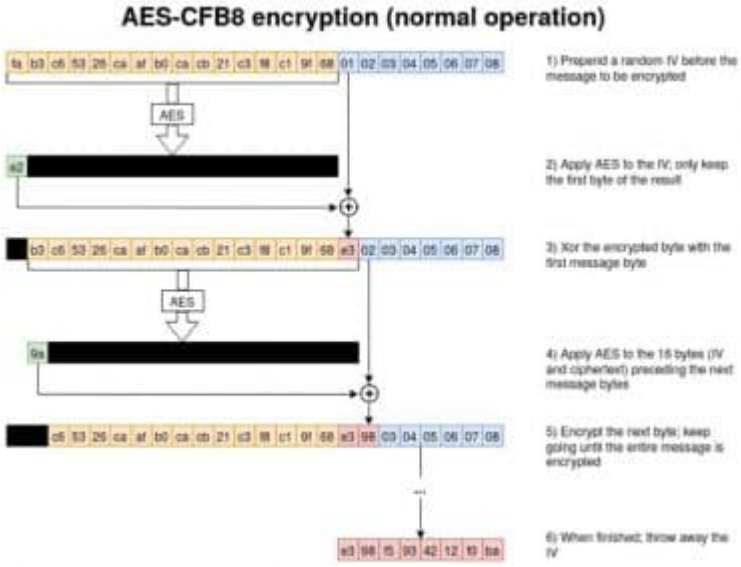
Les attaques sur Netlogon ne sont pas une nouveauté. Mais là où elles requièrent traditionnellement une capacité à intercepter le trafic client-serveur, la faille CVE-2020-1472 est **exploitable par quiconque peut établir une connexion TCP avec un contrôleur de domaine Active Directory vulnérable**.

La primitive que client et serveur utilisent pour générer les identifiants est implémentée dans la fonction ComputeNetlogonCredential. Cette fonction accepte une séquence de 8 octets en entrée et la transforme en une séquence de même longueur, à partir de la clé de session.

Il existe deux versions de ComputeNetlogonCredential : l'une basée sur 2DES et l'autre, plus récente, fondée sur AES. Le client signale au serveur laquelle utiliser. Mais sur les dernières versions de Windows Server, 2DES est bloqué. Or, c'est la version AES qui pose problème.

Par défaut, AES accepte des entrées de 16 octets. Pour faire varier cette taille, il faut choisir un mode de fonctionnement. Avec ComputeNetlogonCredential, c'est le mode CFB8 (8-bit *cipher feedback*).

AES-CFB8 accole à la séquence à chiffrer un vecteur d'initialisation (IV) de 16 octets. Il applique AES aux 16 premiers octets de cet ensemble, récupère le premier octet en sortie et lui applique la fonction XOR (OU exclusif) en combinaison avec le prochain octet de la séquence en clair.



Une histoire de zéros

Pour des raisons de sécurité, le vecteur d'initialisation doit être généré aléatoirement pour chaque élément chiffré avec la même clé. Problème : ComputeNetlogonCredential définit le vecteur comme étant fixe et toujours composé de 16 octets de valeur zéro.

Il existe, selon Secura, 1 chance sur 256 que le valeur chiffrée d'un bloc de zéros avec un tel IV commence par un octet à zéro. Dans ce cas, avec la fonction OU exclusif, tout s'enchaîne (0 XOR 0 = 0 ; cf. schéma ci-dessous).

De manière générale, si un vecteur d'initialisation se compose uniquement de zéros, alors il existe un entier x entre 0 et 255 pour lequel une séquence en clair commençant par n octets de valeur x aura un équivalent chiffré commençant par n octets de valeur 0.

Pas besoin de connaître cette propriété pour attaquer Netlogon. Il suffit de savoir qu'une série de zéros en entrée peut engendrer une série de zéros en sortie.

À partir de là, la première étape consiste à usurper l'identifiant du client. On s'appuie pour cela sur la possibilité de définir un *nonce* arbitraire à 8 zéros côté client. En sachant, d'une part, que statistiquement, l'identifiant devrait être le même au bout de 256 tentatives en moyenne. Et de l'autre, que les mauvais logins ne déclenchent pas de blocage.

Retour en 1970

À ce stade, on ne connaît toujours pas la clé de session. Cela pose problème sur Netlogon avec le mécanisme de chiffrement en transit, qui utilise cette clé. Sauf qu'on peut, comme dit précédemment, désactiver chiffrement et signature côté client… sans que le serveur rejette les connexions.

Pour ce qui est de la valeur d'authentification requise dans le cas où les appels ne sont pas chiffrés, il est possible de faire en sorte qu'elle ait une valeur de 0. Cela implique plusieurs manipulations, dont l'annonce, côté serveur, d'une date correspondant au 1er janvier 1970 (plus d'explications en page 6 du livre blanc de Secura).

On obtient alors la possibilité d'adresser des appels Netlogon depuis toute machine jointe au domaine. Parmi ces appels, il y a NetServerPasswordSet2. Son rôle : définir un nouveau mot de passe pour le client – y compris un mot de passe… vide (Netlogon l'autorise).

Le changement se fait au niveau de l'Active Directory. En local, le client conserve son mot de passe d'origine. On ne pourra donc le resynchroniser au domaine que manuellement. Une forme de déni de service susceptible d'éjecter tout ordinateur du domaine.

L'un des ordinateurs dont on peut changer le mot de passe n'est autre que le contrôleur de domaine. La porte ouverte vers l'obtention des identifiants administrateur.

Microsoft a diffusé un correctif le mois dernier, à l'occasion du Patch Tuesday. Il force l'usage de la signature pour tous les serveurs et clients d'un domaine. Les machines anciennes qui ne prennent la pas en charge font l'objet d'une exception temporaire, jusqu'en février 2021.

Illustration principale © Kentoh – shutterstock.com