SYNACKTIV

Analyse et contournement de la détection de préauthentification PKINIT de MDI

BARBHACK 2024

31/08/2024

Whoami

SYNACKTIV

- Guillaume André (@yaumn_)
- Pentesteur chez Synacktiv depuis 4 ans
- Intérêt pour les internals Windows et Active Directory

Plan

SYNACKTIV

1. Rappels et contexte
2. Investigation et contournement de la détection
3. Authentification PKINIT via l'API Windows

Rappels et contexte

MDI

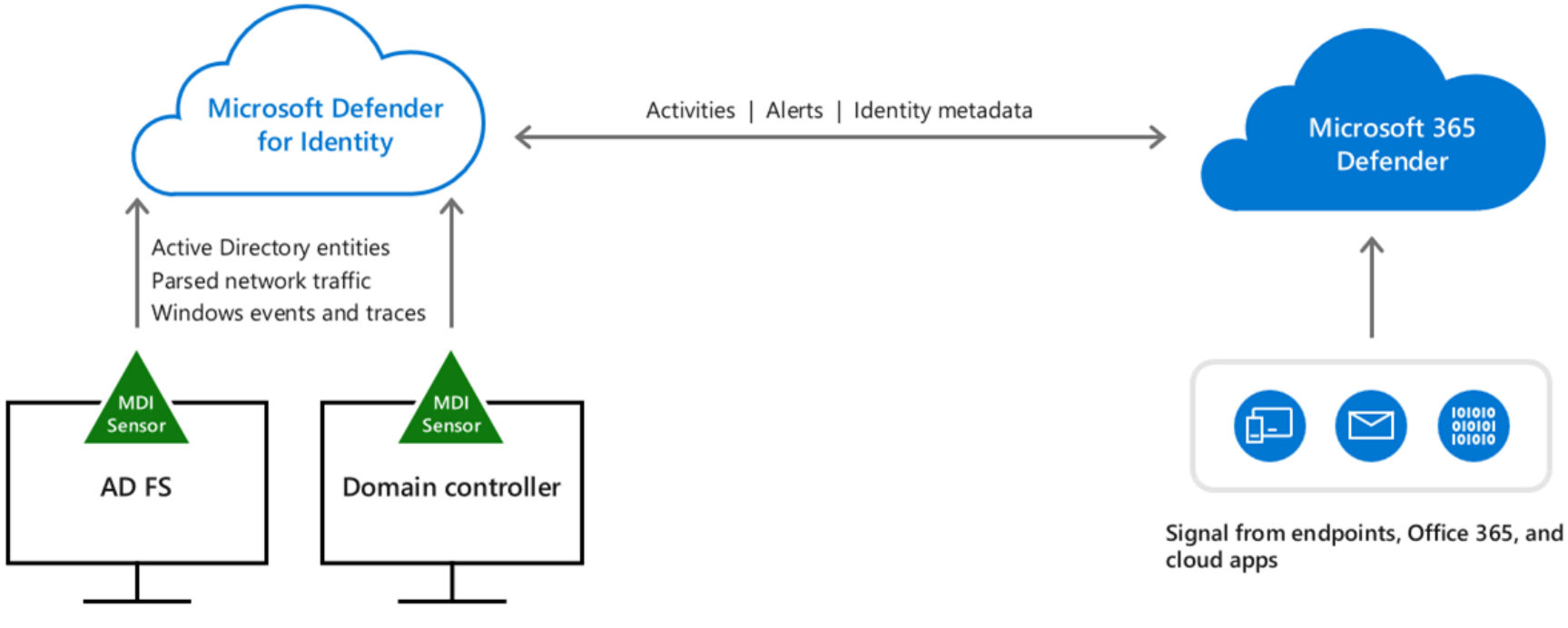
SYNACKTIV

- Microsoft Defender for Identity (ex Azure ATP, ex ATA)
- Composant de la suite Microsoft Defender XDR (ex Microsoft 365 Defender)
- Solution de détection d'attaques réseau et Active Directory
- Partie machine learning basée sur les habitudes des utilisateurs non analysée dans cette recherche

MDI

SYNACKTIV

- **Capteurs installés sur les serveurs de gestion d'identité (DC, ADFS, ADCS)**
- **Lecture des logs et envoi dans le Cloud pour analyse**



PKINIT

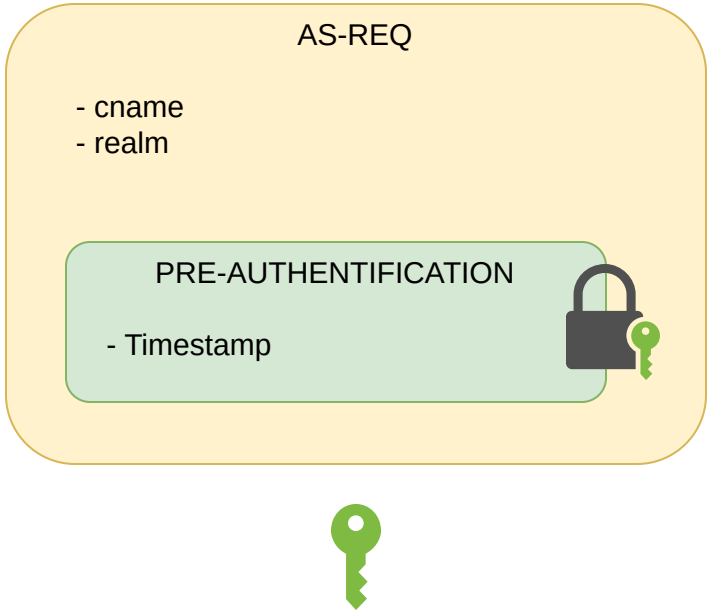
SYNACKTIV

- Public Key Cryptography for Initial Authentication
- Extension Kerberos pour réaliser la préauthentification de manière asymétrique
- Typiquement utilisé pour les authentifications par certificat

PKINIT

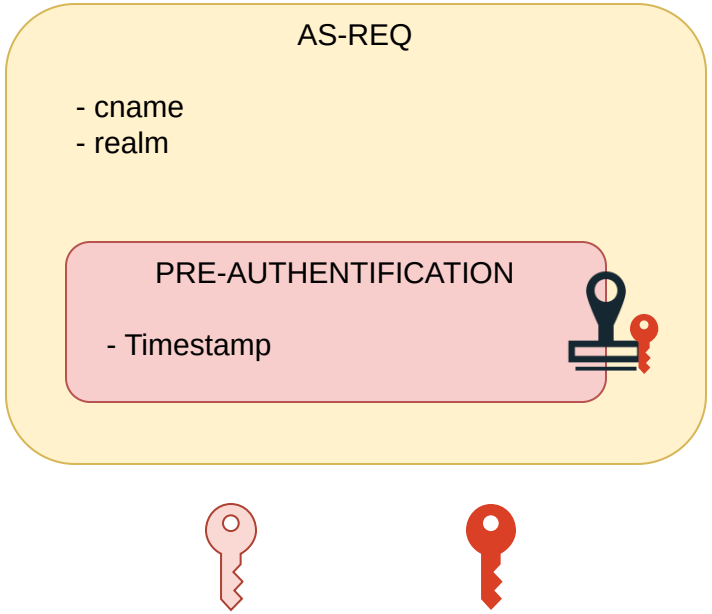
SYNACKTIV

Pré-authentification standard



clé symétrique dérivée du mot de passe de l'utilisateur

Pré-authentification PKINIT



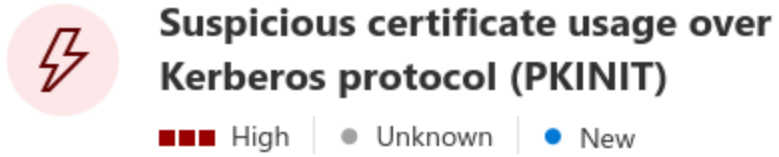
clé publique

clé privée

Contexte

SYNACKTIV

- Mission Red Team
- Obtention d'un certificat via ADCS
- Utilisation du certificat pour s'authentifier en Kerberos → alerte MDI



- But de la recherche : comprendre la détection et la contourner

Investigation de la détection

Postulat et méthodologie

SYNACKTIV

- **Postulat**
  - L'authentification par certificat est souvent utilisée en entreprise
  - La détection ne peut pas être basée sur le simple fait d'utiliser PKINIT
  - Différence d'implémentation de PKINIT par les outils offensifs ?
- **Méthodologie**
  1. Reproduction de la détection
  2. Réalisation d'une authentification par certificat légitime
  3. Jeu des 7 erreurs

Reproduction de la détection

SYNACKTIV

- **Authentification PKINIT avec des outils offensifs**
  - Certipy

```
$ certipy auth -pfx odin.pfx -domain ASGARD.LOCAL -username odin
```

  - Rubeus

```
PS > Rubeus.exe asktgt /certificate:odin.pfx /user:odin /domain:ASGARD.LOCAL
```

- **Captures réseau des messages** `AS-REQ`

Authentification légitime par certificat

SYNACKTIV

- **Comment réaliser une authentification PKINIT autrement qu'avec des outils offensifs ?**
  - Authentification via carte à puce dans un réseau client → pas très pratique et pas d'accès au moment de la recherche
  - Utilisation de l'API Windows dans un lab de test → non trivial au premier abord
  - Utilisation des outils de la suite Samba → facile et rapide à mettre en place

Authentification PKINIT avec Samba

SYNACKTIV

- `/etc/krb5.conf`

```
[realms]
ASGARD.LOCAL = {
    kdc = DC01.asgard.local
    pkinit_anchors = FILE:/etc/krb5/cacert.pem
    pkinit_eku_checking = kpServerAuth
    pkinit_kdc_hostname = DC01.asgard.local
    pkinit_identities = FILE:/etc/krb5/clientcert.pem,/etc/krb5/clientkey.pem
}
```

```
# apt install krb5-pkinit

$ kinit odin@ASGARD.LOCAL
```

- **Aucune alerte MDI !**

Jeu des 7 erreurs

SYNACKTIV

- Comparaison côte à côte des captures réseau des différentes authentifications PKINIT
- Approche par tâtonnements pour trouver le(s) critère(s) de détection
- Modification du code de certipy pour les tests

Jeu des 7 erreurs

kdc-options

Certipy

```
kdc-options: 40800010
  0... .... = reserved: False
  .1.. .... = forwardable: True
  ..0. .... = forwarded: False
  ...0 .... = proxiable: False
  .... 0... = proxy: False
  .... .0.. = allow-postdate: False
  .... ..0. = postdated: False
  .... ...0 = unused7: False
  1... .... = renewable: True
  .0.. .... = unused9: False
  ..0. .... = unused10: False
  ...0 .... = opt-hardware-auth: False
  .... 0... = unused12: False
  .... .0.. = unused13: False
  .... ..0. = constrained-delegation: False
  .... ...0 = canonicalize: False
  0... .... = request-anonymous: False
  .0.. .... = unused17: False
  ..0. .... = unused18: False
  ...0 .... = unused19: False
  .... 0... = unused20: False
  .... .0.. = unused21: False
  .... ..0. = unused22: False
  .... ...0 = unused23: False
  0... .... = unused24: False
  .0.. .... = unused25: False
  ..0. .... = disable-transited-check: False
  ...1 .... = renewable-ok: True
  .... 0... = enc-tkt-in-skey: False
  .... .0.. = unused29: False
  .... ..0. = renew: False
  .... ...0 = validate: False
```

Rubeus

```
kdc-options: 40800010
  0... .... = reserved: False
  .1.. .... = forwardable: True
  ..0. .... = forwarded: False
  ...0 .... = proxiable: False
  .... 0... = proxy: False
  .... .0.. = allow-postdate: False
  .... ..0. = postdated: False
  .... ...0 = unused7: False
  1... .... = renewable: True
  .0.. .... = unused9: False
  ..0. .... = unused10: False
  ...0 .... = opt-hardware-auth: False
  .... 0... = unused12: False
  .... .0.. = unused13: False
  .... ..0. = constrained-delegation: False
  .... ...0 = canonicalize: False
  0... .... = request-anonymous: False
  .0.. .... = unused17: False
  ..0. .... = unused18: False
  ...0 .... = unused19: False
  .... 0... = unused20: False
  .... .0.. = unused21: False
  .... ..0. = unused22: False
  .... ...0 = unused23: False
  0... .... = unused24: False
  .0.. .... = unused25: False
  ..0. .... = disable-transited-check: False
  ...1 .... = renewable-ok: True
  .... 0... = enc-tkt-in-skey: False
  .... .0.. = unused29: False
  .... ..0. = renew: False
  .... ...0 = validate: False
```

kinit

```
kdc-options: 50000010
  0... .... = reserved: False
  .1.. .... = forwardable: True
  ..0. .... = forwarded: False
  ...1 .... = proxiable: True
  .... 0... = proxy: False
  .... .0.. = allow-postdate: False
  .... ..0. = postdated: False
  .... ...0 = unused7: False
  0... .... = renewable: False
  .0.. .... = unused9: False
  ..0. .... = unused10: False
  ...0 .... = opt-hardware-auth: False
  .... 0... = unused12: False
  .... .0.. = unused13: False
  .... ..0. = constrained-delegation: False
  .... ...0 = canonicalize: False
  0... .... = request-anonymous: False
  .0.. .... = unused17: False
  ..0. .... = unused18: False
  ...0 .... = unused19: False
  .... 0... = unused20: False
  .... .0.. = unused21: False
  .... ..0. = unused22: False
  .... ...0 = unused23: False
  0... .... = unused24: False
  .0.. .... = unused25: False
  ..0. .... = disable-transited-check: False
  ...1 .... = renewable-ok: True
  .... 0... = enc-tkt-in-skey: False
  .... .0.. = unused29: False
  .... ..0. = renew: False
  .... ...0 = validate: False
```

Jeu des 7 erreurs

kdc-options

Certipy

```
kdc-options: 40800010
    0... .... = reserved: Fals
    .1.. .... = forwardable: True
    ..0. .... = forwarded: False
    ...0 .... = proxiable: False
    .... 0... = proxy: False
    .... .0.. = allow-postdate: False
    .... ..0. = postdated: False
    .... ...0 = unused7: False
    1... .... = renewable: True
    .0.. .... = unused9: False
    ..0. .... = unused10: False
    ...0 .... = opt-hardware-auth: False
    .... 0... = unused12: False
    .... .0.. = unused13: False
    .... ..0. = constrained-delegation: False
    .... ...0 = canonicalize: False
    0... .... = request-anonymous: False
    .0.. .... = unused17: False
    ..0. .... = unused18: False
    ...0 .... = unused19: False
    .... 0... = unused20: False
    .... .0.. = unused21: False
    .... ..0. = unused22: False
    .... ...0 = unused23: False
    0... .... = unused24: False
    .0.. .... = unused25: False
    ..0. .... = disable-transited-heck: False
    ...1 .... = renewable-ok: ue
    .... 0... = enc-tkt-in- key: False
    .... .0.. = unused2 False
    .... ..0. = ren False
    .... ...0 = validate: False
```

Rubeus

```
kdc-options: 40800010
    0... .... = reserved: False
    .1.. .... = forwardable: True
    ..0. .... = forwarded: False
    ...0 .... = proxiable: False
    .... 0... = proxy: False
    .... .0.. = allow-postdate: False
    .... ..0. = postdated: False
    .... ...0 = unused7: False
    1... .... = renewable: True
    .0.. .... = unused9: False
    ..0. .... = unused10: False
    ...0 .... = opt-hardware-au False
    .... 0... = u sed12: Fa e
    .... .0.. = unus False
    .... ..0. = cons d-delegation: False
    .... ...0 = c onicaliz False
    0... .... = request-anonymo False
    .0.. .... = unused17: False
    ..0. .... = unused18: False
    ...0 .... = unused19: False
    .... 0... = unused20: False
    .... .0.. = unused21: False
    .... ..0. = unused22: False
    .... ...0 = unused23: False
    0... .... = unused24: False
    .0.. .... = unused25: False
    ..0. .... = disable-transited-check: False
    ...1 .... = renewable-ok: True
    .... 0... = enc-tkt-in-skey: False
    .... .0.. = unused29: False
    .... ..0. = renew: False
    .... ...0 = validate: False
```

kinit

```
kdc-options: 500 0010
    0... .... = reserved: False
    .1.. .... = forwardable: True
    ..0 .... = forwarded: False
    ...1 .... = proxiable: True
    .... 0... = proxy: False
    .... .0.. = allow-postdate: False
    .... ..0. = postdated: False
    .... ...0 = unused7: False
    0... .... = renewable: False
    .0.. .... = unused9: False
    ..0. .... = unused10: False
    ...0 .... = opt-hardware-auth: False
    .... 0... = unused12: False
    .... .0.. = unused13: False
    .... ..0. = constrained-delegation: False
    .... ...0 = canonicalize: False
    0... .... = request-anonymous: False
    .0.. .... = unused17: False
    ..0. .... = unused18: False
    ...0 .... = unused19: False
    .... 0... = unused20: False
    .... .0.. = unused21: False
    .... ..0. = unused22: False
    .... ...0 = unused23: False
    .... .... = unused24: False
    .0.. .... = unused25: False
    ..0. .... = disable-transited-check: False
    ...1 .... = renewable-ok: True
    .... 0... = en tkt-in-skey: False
    .... .0.. = unuse False
    .... ..0. = renew: Fa e
    .... ...0 = validate: False
```

Jeu des 7 erreurs

pA-PAC-REQUEST

SYNACKTIV

Certipy

```
padata: 2 items
  PA-DATA pA-PAC-REQUEST
  PA-DATA pA-PK-AS-REQ
```

Rubeus

```
padata: 2 items
  PA-DATA pA-PAC-REQUEST
  PA-DATA pA-PK-AS-REQ
```

kinit

```
padata: 3 items
  PA-DATA pA-PK-AS-REQ
  PA-DATA pA-AS-FRESHNESS
  PA-DATA pA-REQ-ENC-PA-REP
```

Jeu des 7 erreurs

pA-PAC-REQUEST

SYNACKTIV

Certipy

```
padata: 2 items
  PA-DATA pA-PAC-REQUEST
  PA-DATA pA-PK-AS-REQ
```

Rubeus

```
padata: 2 items
  PA-DATA pA-PAC-REQUEST
  PA-DATA pA-PK-AS-REQ
```

Kinit

```
padata: 3 items
  PA-DATA pA-PK-AS-REQ
  PA-DATA pA-AS-FRESHNESS
  PA-DATA pA-REQ-ENC-PA-REP
```

Jeu des 7 erreurs

etype

SYNACKTIV

Certipy

```
▾ etype: 2 items
    ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
    ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)
```

Rubeus

```
▾ etype: 1 item
    ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

kinit

```
▾ etype: 8 items
    ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
    ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)
    ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA384-192 (20)
    ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA256-128 (19)
    ENCTYPE: eTYPE-DES3-CBC-SHA1 (16)
    ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
    ENCTYPE: eTYPE-CAMELLIA128-CTS-CMAC (25)
    ENCTYPE: eTYPE-CAMELLIA256-CTS-CMAC (26)
```

Jeu des 7 erreurs

etype

SYNACKTIV

Certipy

```
▾ etype: 2 items
    ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
    ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)
```

Rubeus

```
▾ etype: 1 item
    ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
```

kinit

```
▾ etype: 8 items
    ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA1-96 (18)
    ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA1-96 (17)
    ENCTYPE: eTYPE-AES256-CTS-HMAC-SHA384-192 (20)
    ENCTYPE: eTYPE-AES128-CTS-HMAC-SHA256-128 (19)
    ENCTYPE: eTYPE-DES3-CBC-SHA1 (16)
    ENCTYPE: eTYPE-ARCFOUR-HMAC-MD5 (23)
    ENCTYPE: eTYPE-CAMELLIA128-CTS-CMAC (25)
    ENCTYPE: eTYPE-CAMELLIA256-CTS-CMAC (26)
```

Bilan sur la détection

SYNACKTIV

- Basée sur les algorithmes de chiffrement supportés par le client
- Non déclenchée si on change la liste des algorithmes
- MDI pourrait se baser sur d'autres critères
- Peut-on faire mieux ?

Authentification PKINIT via l'API Windows

API Windows

SYNACKTIV

- **Idée : laisser Windows faire tout le travail**
  - Pas besoin de réimplémenter Kerberos
  - Pas de détection possible basée sur des différences avec l'implémentation de référence
- **Objectif : pouvoir faire l'équivalent d'un `runas /netonly` mais avec un certificat**
  - Doit être utilisable depuis une machine Windows non liée au domaine
- **Problème : les API classiques de création de logon session nécessitent un mot de passe**
  - `LogonUser`, `CreateProcessWithLogon`

API Windows

CredMarshalCredential

SYNACKTIV

CredMarshalCredentialW function (wincred.h)

Article • 02/09/2023 Feedback

The **CredMarshalCredential** function transforms a credential into a text string. Historically, many functions, such as NetUseAdd, take a domain name, user name, and password as credentials. These functions do not accept certificates as credentials. The **CredMarshalCredential** function converts such credentials into a form that can be passed into these APIs.

The marshaled credential should be passed as the user name string to any API that is currently passed credentials. The domain name, if applicable, passed to that API should be passed as **NULL** or empty. For certificate credentials, the PIN of the certificate should be passed to that API as the password.

The caller should not modify or print marshaled credentials. The returned value can be freely converted between the Unicode, ANSI, and OEM characters sets. The string is case sensitive.

Syntax

C++ Copy

```
BOOL CredMarshalCredentialW(
  [in]  CRED_MARSHAL_TYPE CredType,
  [in]  PVOID             Credential,
  [out] LPWSTR            *MarshaledCredential
);
```

API Windows

CERT_CREDENTIAL_INFO

SYNACKTIV

CERT_CREDENTIAL_INFO structure (wincred.h)

Article • 02/22/2024 Feedback

The **CERT_CREDENTIAL_INFO** structure contains a reference to a certificate.

Syntax

C++ Copy

```
typedef struct _CERT_CREDENTIAL_INFO {
  ULONG cbSize;
  UCHAR rgbHashOfCert[CERT_HASH_LENGTH];
} CERT_CREDENTIAL_INFO, *PCERT_CREDENTIAL_INFO;
```

Members

`cbSize`

Size of the structure in bytes. This member should be set to `sizeof(CERT_CREDENTIAL_INFO)`. This structure might be a larger value in the future, indicating a newer version of the structure.

`rgbHashOfCert[CERT_HASH_LENGTH]`

SHA-1 hash of the certificate referenced.

API Windows

Résumé

SYNACKTIV

- **Workflow**
  1. Ajout du certificat au magasin de certificat de l'utilisateur
  2. Création d'une instance de `CERT_CREDENTIAL_INFO` avec le champ `rgbHashOfCert` égal au hash SHA1 du certificat
  3. Appel à `CredMarshalCredential` pour convertir la structure en une chaîne de caractères
  4. Appel à `CreateProcessWithLogon` avec le paramètre `lpUsername` égal à la chaîne de caractères précédemment obtenue
- **Logique déjà implémentée par un employé de Microsoft**

API Windows

Problème

SYNACKTIV

Machine jointe au domaine

```
C:\> dir \\DC01.ASGARD.LOCAL\SYSVOL
    Directory: \\DC01.ASGARD.LOCAL\SYSVOL


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----l          9/5/2023   9:07 PM                ASGARD.LOCAL


C:\> klist
Current LogonId is 0:0x22a59a

Cached Tickets: (3)

#0>     Client: odin @ ASGARD.LOCAL
        Server: krbtgt/ASGARD.LOCAL @ ASGARD.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded
        renewable pre_authent name_canonicalize
        Start Time: 8/28/2024 0:06:41 (local)
        End Time:   8/28/2024 10:06:17 (local)
        Renew Time: 9/4/2024 0:06:17 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x2 -> DELEGATION
        Kdc Called: DC01.ASGARD.LOCAL
[...]
```

Machine hors domaine

```
C:\> dir \\DC01.ASGARD.LOCAL\SYSVOL
The Kerberos protocol encountered an error while
validating the KDC certificate during smartcard
logon. There is more information in the system
event log.

C:\> klist
Current LogonId is 0:0x1463c00
Cached Tickets: (0)
```

API Windows

kerberos.dll

SYNACKTIV

- **SSP Kerberos implémenté dans `kerberos.dll`, chargé par `lsass.exe`**
- **Idée : hooker avec frida toutes les fonctions liées à PKINIT et comparer les sorties pour les deux cas**
  - `KerbBuildPkinitPreauthData`
  - `KerbCheckKdcCertificate`
  - `KerbCheckKdcCertificateKeyUsage`
  - …

API Windows

Comparaison avec frida

SYNACKTIV

Machine jointe au domaine

```
PS C:\> frida -p (Get-Process lsass).Id -l hook.js
[...]
Enter KerbInitializePkCreds
Leave KerbInitializePkCreds : 0x0
Enter KerbInitializePkCreds
Leave KerbInitializePkCreds : 0x0
Enter KerbGetPKINITPreauthType
Leave KerbGetPKINITPreauthType : 0x0
Enter KerbBuildPkinitPreauthData
Leave KerbBuildPkinitPreauthData : 0x0
Enter KerbBuildPkinitPreauthData
  Enter KerbVerifyPkAsReply
    Enter KerbCheckKdcCertificate
      Enter KerbCheckKdcCertificateKeyUsage
      Leave KerbCheckKdcCertificateKeyUsage : 0x0
    Leave KerbCheckKdcCertificate : 0x0
  Leave KerbVerifyPkAsReply : 0x0
Leave KerbBuildPkinitPreauthData : 0x0
```

Machine hors-domaine

```
PS C:\> frida -p (Get-Process lsass).Id -l hook.js
[...]
Enter KerbInitializePkCreds
Leave KerbInitializePkCreds : 0x0
Enter KerbInitializePkCreds
Leave KerbInitializePkCreds : 0x0
Enter KerbGetPKINITPreauthType
Leave KerbGetPKINITPreauthType : 0x0
Enter KerbBuildPkinitPreauthData
Leave KerbBuildPkinitPreauthData : 0x0
Enter KerbBuildPkinitPreauthData
  Enter KerbVerifyPkAsReply
    Enter KerbCheckKdcCertificate
      Enter KerbCheckKdcCertificateKeyUsage
      Leave KerbCheckKdcCertificateKeyUsage : 0xc0000320
    Leave KerbCheckKdcCertificate : 0xc0000320
  Leave KerbVerifyPkAsReply : 0xc0000320
Leave KerbBuildPkinitPreauthData : 0xc0000320
```

API Windows

Patch LSASS

SYNACKTIV

- **Changement de la valeur de retour de `KerbCheckKdcCertificate` par `0` -> tout fonctionne !**
- **Nécessite de patcher LSASS et d'avoir les symboles Windows**
- **Peut-on faire mieux ?**

API Windows

KerbCheckKdcCertificateKeyUsage

SYNACKTIV

- **Réalise des vérifications sur les EKU du certificat du contrôleur de domaine**
- **Échoue si :**
  - Le certificat ne possède pas au moins l'une des EKU suivantes :
    - `KDC authentication`
    - `Smart Card Logon`
    - `Server Authentication`
    - `Any extended key usage`
  - OU que la vérification KDC est activée et que le certificat ne possède pas l'EKU `KDC authentication`
- **La variable globale `KerbGlobalStandaloneKdcValidation` permet de désactiver la vérification KDC**

API Windows

Problème n°2

SYNACKTIV

```
C:\> dir \\DC01.ASGARD.LOCAL\SYSVOL
The revocation status of the domain controller certificate used for smartcard
authentication could not be determined. There is additional information in the
system event log. Please contact your system administrator.

C:\> klist
Current LogonId is 0:0x1463c00
Cached Tickets: (0)
```

API Windows

KerbCheckKdcCertificate

SYNACKTIV

- **Vérifie (entre autres) la chaîne de confiance du certificat ainsi que sa révocation**
- **La vérification de la révocation peut-être désactivée via la variable globale**

```
KerbGlobalUseCachedCRLOnlyAndIgnoreRevocationUnknownErrors
```

API Windows

Victoire !

```
C:\> dir \\DC01.ASGARD.LOCAL\SYSVOL
    Directory: \\DC01.ASGARD.LOCAL\SYSVOL


Mode                 LastWriteTime         Length Name
----                 -------------         ------ ----
d----l          9/5/2023   9:07 PM                ASGARD.LOCAL


C:\> klist
Current LogonId is 0:0x22a59a

Cached Tickets: (3)

#0>     Client: odin @ ASGARD.LOCAL
        Server: krbtgt/ASGARD.LOCAL @ ASGARD.LOCAL
        KerbTicket Encryption Type: AES-256-CTS-HMAC-SHA1-96
        Ticket Flags 0x60a10000 -> forwardable forwarded
        renewable pre_authent name_canonicalize
        Start Time: 8/28/2024 0:06:41 (local)
        End Time:   8/28/2024 10:06:17 (local)
        Renew Time: 9/4/2024 0:06:17 (local)
        Session Key Type: AES-256-CTS-HMAC-SHA1-96
        Cache Flags: 0x2 -> DELEGATION
        Kdc Called: DC01.ASGARD.LOCAL
[...]
```

API Windows

Pas de patch LSASS ?

SYNACKTIV

- **Comment modifier les deux variables globales sans patcher LSASS ?**
- **Cross references des deux variables globales dans IDA ->** `KerbGetKerbRegParams`
  - Lis les valeurs de la clé de registre

    `HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Kerberos\Parameters`
  - Modifie à la volée les variables globales associées
- **Il suffit donc d'ajouter les valeurs de registre suivantes :**
  - `StandaloneKdcValidation` = 0 (DWORD)
  - `UseCachedCRLOnlyAndIgnoreRevocationUnknownErrors` = 0 (DWORD)
- **Pas de reboot nécessaire**

Conclusion

SYNACKTIV

- Détection MDI basée sur des différences d'implémentation de Kerberos
- Présentation d'une méthodologie simple et générique pour comprendre rapidement l'origine d'un problème
- Script PowerShell pour s'authentifier avec un certificat disponible sur le github de Synacktiv : Invoke-RunAsWithCert

SYNACKTIV

https://www.linkedin.com/company/synacktiv

https://twitter.com/synacktiv

https://synacktiv.com