WAVESTONE

CERT-W

CYBERSECURITY EXPERTS' NEWSLETTER

CERT-WAVESTONE NEWSLETTER

N°9

SUMMARY

EDITO

2017: a dark year...

... for the many companies, who were victims of destructive attacks (#NotPetya) and massive data leaks involving, often directly, cloud providers (#Yahoo, #Deloitte, #Equifax, #Uber, #USArmy, #NSA, #AWS and countless others).

The end of the year, and the explosion of Bitcoin's market price, saw renewed attacks on Bitcoin wallets and Bitcoin mining javascripts—something our CERT-W team has had direct experience of.

As well as these cases, below is a quick overview of some of the work our team has been involved in over the past three months:

/ Victims of brute force attacks on Office 365... resulting in accounts being compromised. Some of the impacts were:

- The misappropriation of bank transfers.
- The blocking of over 800 user accounts.

/ Two victims of ransomware spread by RDP (DHARMA/CRYSIS family).

/ A suspected compromise after the discovery of a username in sinograms (feedback from the experience is in this newsletter).

/ An attempt to destabilize a large corporation through a defamation campaign.

/ A workstation compromised by several types of malware... which were over ten years old (2001: Magistr.a@MM , 2004: Netsky - by the creator of Sasser, 2009: Kakworm...).

/ A suspicion that iOS devices had been compromised.

While no particular trend is emerging at present, we're ready to bet our Bitcoin wallets that attacks on cloud providers will continue to spread... unless Meltdown and Spectre trigger the end of the world, that is!

Vincent Nguyen, Manager, Head of CERT-WAVESTONE

USING REPLICATION METADATA WHEN LOGS FAIL

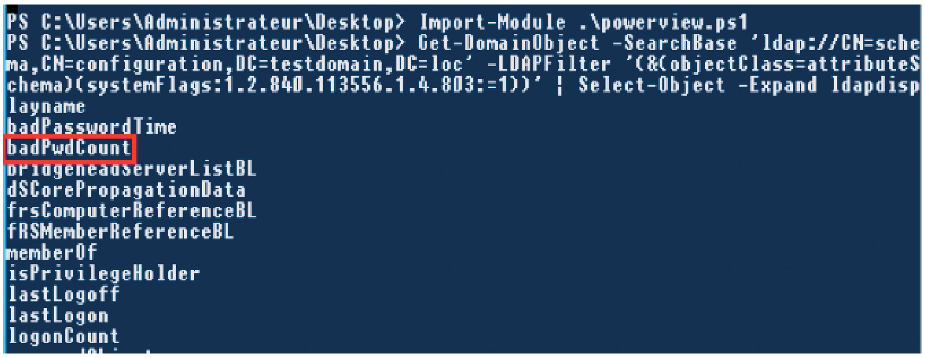
INTRODUCTION TO REPLICATION DATA IN THE ACTIVE DIRECTORY

In an Active Directory domain, there are usually several domain controllers that require the same information. To achieve this, the Active Directory has a replication mechanism that allows, among other things, changes to be propagated from one domain controller to the others.

In its replication process, the Active Directory uses USNs (Update Sequence Numbers) to determine the state of the domain controllers. These USNs represent a counter stored in the Active Directory database, which is incremented each time this database is changed at domain-controller level. Each domain controller then has a USN of its own.

When a change of information in the Active Directory occurs on a domain controller, there are two possibilities:

/ The modified information is not replicated between the different domain controllers. This is the case for all the attributes of the Active Directory that have the flag FLAG_ATTR_NOT_REPLICATED[1]; an example is the attribute «BadPwdCount», which keeps track of the number of unsuccessful connection attempts:



```
PS C:\Users\Administrateur\Desktop> Import-Module .\powerview.ps1
PS C:\Users\Administrateur\Desktop> Get-DomainObject -SearchBase 'ldap://CN=sche
ma,CN=configuration,DC=testdomain,DC=loc' -LDAPFilter '(&(objectClass=attributeS
chema)(systemFlags:1.2.840.113556.1.4.803:=1))' | Select-Object -Expand ldapdisp
layname
badPasswordTime
badPwdCount
bridgeheadServerListBL
dSCorePropagationData
frsComputerReferenceBL
fRSMemberReferenceBL
memberOf
isPrivilegeHolder
lastLogoff
lastLogon
logonCount
```

/ In this case, the domain controller makes the change in its own database, but transmits nothing to the other domain controllers. We should note, then, that it's possible to increase the maximum number of allowed erroneous connection attempts, before an account is blocked, by performing authentication attempts using separate domain controllers.

/ The modified information requires replication between the different domain controllers. In this case, the domain controller that received the change uses the Active Directory's replication model to pass the change to the other domain controllers. We don't discuss this replication model here, but it allows the transmission of the changes to all the domain controllers by limiting the traffic necessary and by ensuring the management of collisions (which occur when the same attribute is changed on different controllers within a limited time window).

The replication process uses metadata that is held in the form of two distinct attributes: msDS-ReplAttributeMetaData[2] and msDS-ReplValueMetaData[3]. msDS-ReplAttributeMetaData is used to make changes to non-linked attributes in the Active Directory, while msDS-ReplValueMetaData is reserved for linked attributes.

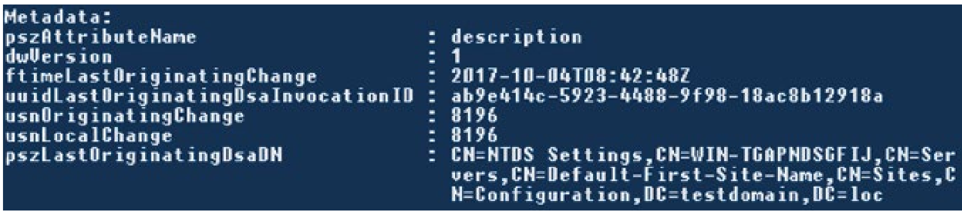
Linked attributes were introduced into the Active Directory from Windows Server 2003. They are, in fact, pairs of attributes, where the value of one is based on the other. This is the case, for example, with the member attributes of a group and the member of attributes of a user.

WHY DOES THIS MERIT INVESTIGATION?

As a forensic analyst who responds to security incidents, the first reaction—when it comes to identifying malicious activity on an Active Directory—is to use event logs. But what if they weren't activated at the time of the attack? Or, if the attackers managed to delete the logs they generated, using a tool like mimikatz [4]?

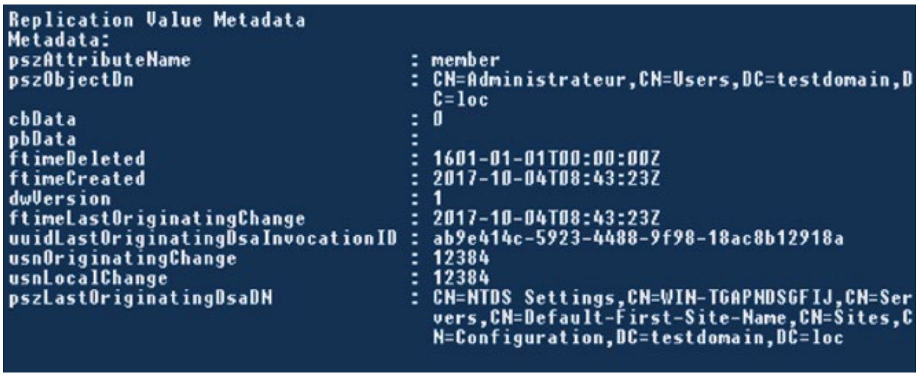
In such situations, you can use replication data to get a partial view of the attacker's actions. Because of the way in which replication data operates, any modification to an attribute in the Active Directory results in the creation of replication data containing information that can be useful for an investigation.

In the case of a non-linked attribute, and, therefore, metadata of the msDS-ReplAttributeMetaData type, the stored information is the version – which corresponds to the number of changes to the attribute since its creation, the date on which the change was performed, the USN corresponding to the change for the domain controller that initiated the replication, the USN corresponding to the change for the domain controller on which the metadata is retrieved, and the UUID and DN of the domain controller that initiated the change:



```
Metadata:
pszAttributeName                  : description
dwVersion                         : 1
ftimeLastOriginatingChange        : 2017-10-04T08:42:48Z
uuidLastOriginatingDsaInvocationID : ab9e414c-5923-4488-9f98-18ac8b12918a
usnOriginatingChange              : 8196
usnLocalChange                    : 8196
pszLastOriginatingDsaDN           : CN=NTDS Settings,CN=WIN-TGAPNDSGFIJ,CN=Ser
                                    vers,CN=Default-First-Site-Name,CN=Sites,C
                                    N=Configuration,DC=testdomain,DC=loc
```

For linked attributes, the replication metadata, this time of the msDS-ReplValueMetaData type, will also store information about the attributes related to the attribute in question. The replication metadata will then keep information about each of the properties of the linked attribute, including its previous values. Taking the example of the member attribute, the replication data will keep both the information about the current members of the group, but also about users who were, but are no longer, members:



```
Replication Value Metadata
Metadata:
pszAttributeName                   : member
pszObjectDn                        : CN=Administrateur,CN=Users,DC=testdomain,D
                                     C=loc
cbData                             : 0
pbData                             :
ftimeDeleted                       : 1601-01-01T00:00:00Z
ftimeCreated                       : 2017-10-04T08:43:23Z
dwVersion                          : 1
ftimeLastOriginatingChange         : 2017-10-04T08:43:23Z
uuidLastOriginatingDsaInvocationID : ab9e414c-5923-4488-9f98-18ac8b12918a
usnOriginatingChange               : 12384
usnLocalChange                     : 12384
pszLastOriginatingDsaDN            : CN=NTDS Settings,CN=WIN-TGAPNDSGFIJ,CN=Ser
                                     vers,CN=Default-First-Site-Name,CN=Sites,C
                                     N=Configuration,DC=testdomain,DC=loc
```

At any given moment, the date of the last modification to an attribute can be determined using this data, as well as the number of times it has been modified since it was created. This data, although apparently limited, can then be used to identify different attack scenarios.

INCREASING PRIVILEGES BY ADDING TO A GROUP

One of the situations where replication data delivers the best results is when identifying a scenario where attackers have added themselves to, and then removed themselves from, a group such as the "Domain Admins" group.

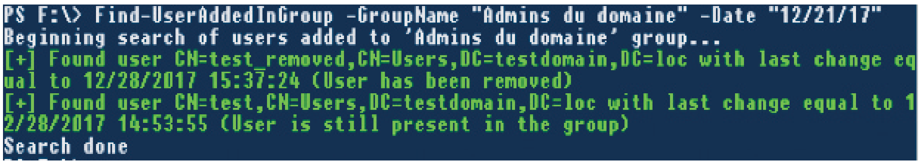
In an Active Directory, groups have a "member" property that lists the users who are members of the group. Adding a user to a group will then increment the USN of its "member" attribute by one, which represents the change. Likewise, removing a user from membership will also increment this USN by one.

Given these properties, there are two possible conclusions:

/ Users with an odd USN are members of the group (something that can be seen directly from the value of the «member» attribute), and the date the user was last added to the group is that of the USN;

/ Users with an even USN were members of the group, but have not been so since the date of the USN.

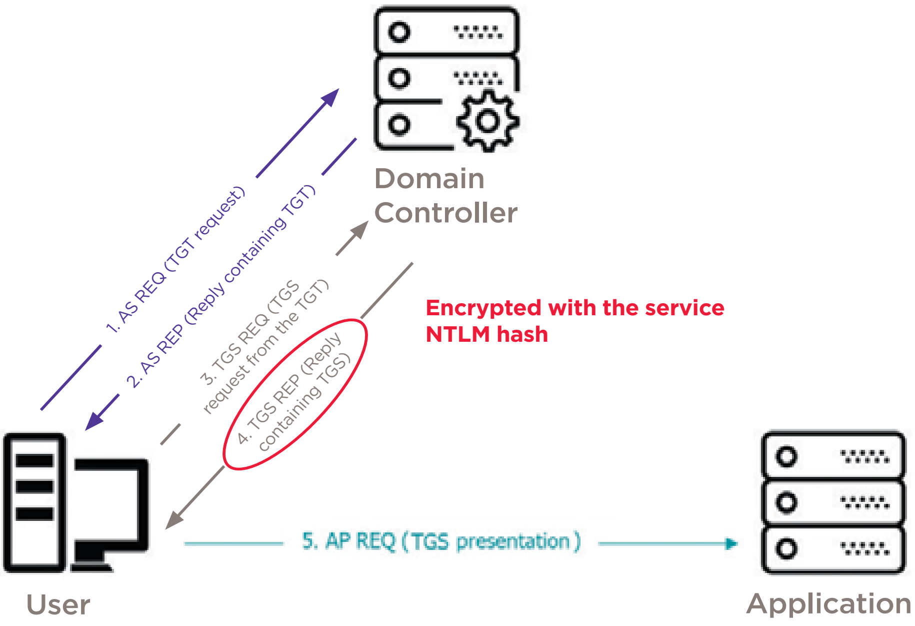
This is the second case, where an attacker has been added to the "Domain Admins" group, for malicious purposes, and then removed themself from it. It's possible, then, to create a script that retrieves the users who were added, or removed, from a group after a given date (with only the dates of the first and last changes being retained, limiting the search to a given period could be deceptive):



```
PS F:\> Find-UserAddedInGroup -GroupName "Admins du domaine" -Date "12/21/17"
Beginning search of users added to 'Admins du domaine' group...
[+] Found user CN=test_removed,CN=Users,DC=testdomain,DC=loc with last change eq
ual to 12/28/2017 15:37:24 (User has been removed)
[+] Found user CN=test,CN=Users,DC=testdomain,DC=loc with last change equal to 1
2/28/2017 14:53:55 (User is still present in the group)
Search done
```

TARGETED KERBEROASTING

Kerberoasting is a technique that uses the Kerberos authentication process to allow an attacker to retrieve the password for a service account (i.e. an "account with a Service Principal Name"). The principle of this attack is as shown in the following diagram: when requesting a service from a user, the KDC uses the NTLM hash of the service account to encrypt the TGS returned to the user. In this process, the legitimacy of the user to access the service is not verified, and, therefore, any user can obtain the TGS.


Domain Controller
1. AS REQ (TGT request)
2. AS REP (Reply containing TGT)
3. TGS REQ (TGS request from the TGT)
4. TGS REP (Reply containing TGS)
Encrypted with the service NTLM hash
User
5. AP REQ (TGS presentation)
Application

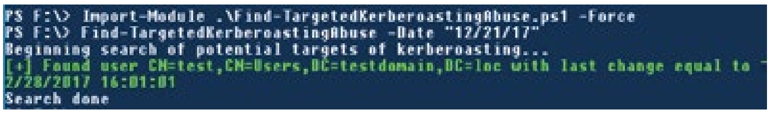
The attacker can then attempt to break the NTLM hash of the service account by trying to decrypt the TGS from successive hashes.

Now, suppose an attacker has managed to recover the maximum level of privileges on a user object, namely GenericAll type privileges [5], which, in particular, provide the right to change the account password or modify the properties of the Active Directory object associated with the account. Therefore, to misappropriate the account in question, the attacker could reset its password to whatever they choose, and then connect using this new password. Having said that, such an attack would be quickly detected by the account's legitimate user, who would not be able to connect anymore with their usual password.

A more attractive option for the attacker would then be to add a Service Principal Name (SPN) to the targeted account, and then carry out a kerberoasting attack. This is called targeted kerberoasting.

Since the majority of users in a domain are presumed never to have SPNs, such an attack can be detected quite simply—if this SPN has not been removed. However, if the SPN is deleted by the attacker after the attack, it is still possible to make use of the replication data.

Indeed, the addition or deletion of an SPN is a replicated event within the Active Directory, and, as such, generates replication metadata of the msDS-ReplAttributeMetaData type:



```
PS F:\> Import-Module .\Find-TargetedKerberoastingAbuse.ps1 -Force
PS F:\> Find-TargetedKerberoastingAbuse -Date "12/21/17"
Beginning search of potential targets of kerberoasting...
[+] Found user CN=test,CN=Users,DC=testdomain,DC=loc with last change equal to -
2/28/2017 16:01:01
Search done
```

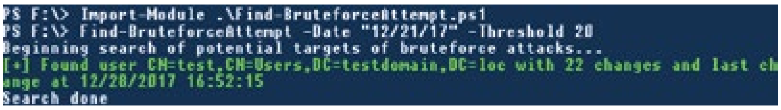
It is then possible to create a script that retrieves the accounts of the domain whose SPN attributes have been modified after a given date, and which are therefore potential victims of a targeted kerberoasting attack.

A BRUTE FORCE ATTACK ON AN ACCOUNT BY SUCCESSIVE BLOCKING

A brute force attack scenario that attackers can deploy on an Active Directory, and which does not result in any form of alert, is to make connection attempts outside the account's service hours, and to do this until the account is blocked.

When blocking an account, a LOCKOUT flag[6] is set on the user's userAccountControl attribute.

Because this attribute is replicated between the different domain controllers, replication data of the msDS-ReplAttributeMetaData type is generated. It is then possible to create a script to identify the domain accounts in the replication data that have high version numbers for this attribute, something that might signal such a brute force attack:



```
PS F:\> Import-Module .\Find-BruteforceAttempt.ps1
PS F:\> Find-BruteforceAttempt -Date "12/21/17" -Threshold 20
Beginning search of potential targets of bruteforce attacks...
[+] Found user CN=test,CN=Users,DC=testdomain,DC=loc with 22 changes and last ch
ange at 12/28/2017 16:52:15
Search done
```

However, it should be noted that the userAccountControl attribute has several other flags whose modification would also lead to the generation of replication data that are inseparable from the previous ones; the PASSWORD_EXPIRED flag, for example. However, as a general rule, this attribute is unlikely to change much, and, if there is a very large number of observed changes, it remains a fairly reliable indicator of a brute force attack.

Another point to note is that an attacker who limits login attempts to avoid an account being blocked would be invisible to this investigatory technique.

CONCLUSION

Although not providing as complete a view as event logs, replication data can be a significant source of information for the forensic investigation of an Active Directory.

It should be noted, however, that techniques for modifying replication data exist [7], so you should avoid placing blind faith in the information they yield.

Nicolas DAUBRESSE, Consultant

Sources

[1] See "systemFlags": https://msdn.microsoft.com/en-us/library/cc223202.aspx

[2] https://msdn.microsoft.com/en-us/library/cc220352.aspx

[3] https://msdn.microsoft.com/en-us/library/cc220356.aspx

[4] https://github.com/gentilkiwi/mimikatz/releases

[5] https://msdn.microsoft.com/en-us/library/aa772285(v=vs.85).aspx

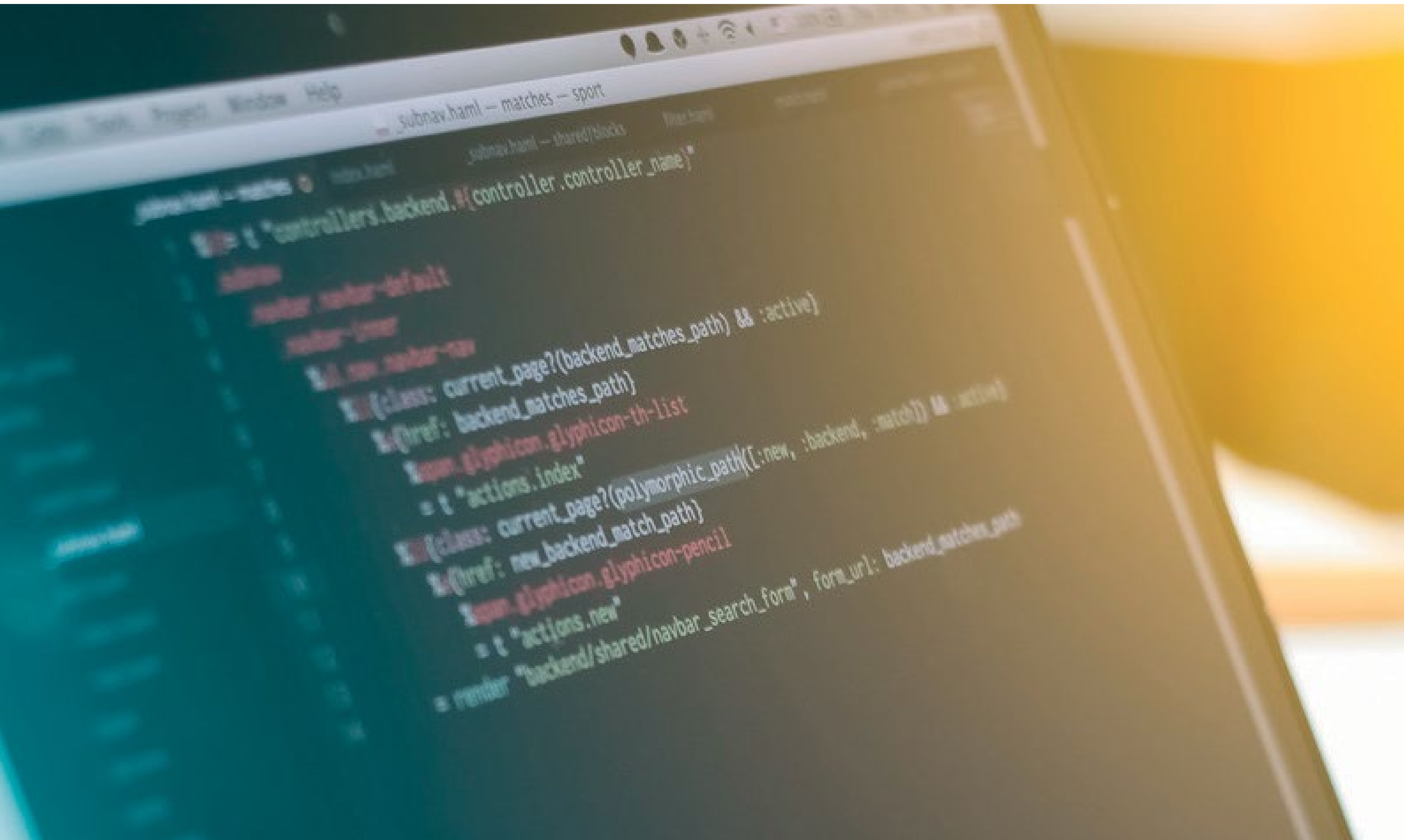
[6] https://support.microsoft.com/en-us/help/305144/how-to-use-the-useraccountcontrol-flags-to-manipulate-user-account-pro

[7] https://twitter.com/mysmartlogon/status/903166180889907200

https://www.harmj0y.net/blog/defense/hunting-with-active-directory-replication-metadata/

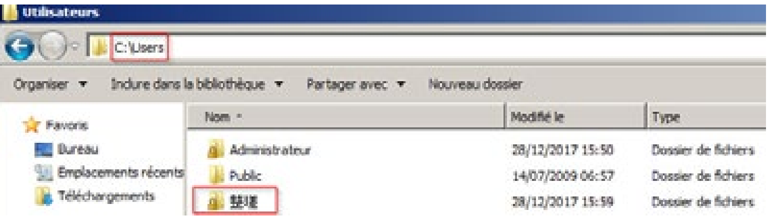
https://social.technet.microsoft.com/wiki/contents/articles/25946.metadata-de-replication-et-analyse-forensic-active-directory-fr-fr.aspx

https://blogs.technet.microsoft.com/pie/2014/08/25/metadata-2-the-ephemeral-admin-or-how-to-track-the-group-membership/



HOUSTON, WE’VE GOT A CHINESE ACCOUNT ON OUR SYSTEM

CERT-W was contacted to **investigate the origin** of a **Chinese user folder** on a **Windows** system :



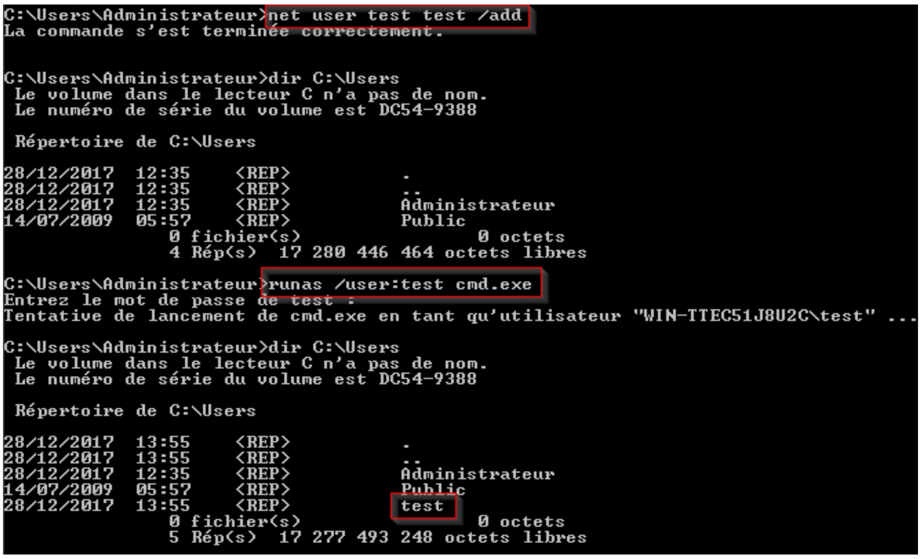
Utilisateurs

C:\Users

Nom	Modifié le	Type
Administrateur	28/12/2017 15:50	Dossier de fichiers
Public	14/07/2009 06:57	Dossier de fichiers
整瑳	28/12/2017 15:59	Dossier de fichiers

Presence of a Chinese folder in the «C:\Users» folder

A user folder is **generated** within the « C:\Users » directory after the user’s **first authentication, not** when the user is **created**.

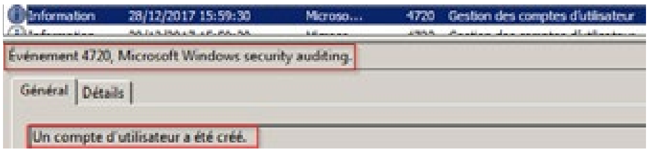


```
C:\Users\Administrateur>net user test test /add
La commande s’est terminée correctement.

C:\Users\Administrateur>dir C:\Users
 Le volume dans le lecteur C n’a pas de nom.
 Le numéro de série du volume est DC54-9388

 Répertoire de C:\Users

28/12/2017  12:35    <REP>          .
28/12/2017  12:35    <REP>          ..
28/12/2017  12:35    <REP>          Administrateur
14/07/2009  05:57    <REP>          Public
               0 fichier(s)                0 octets
               4 Rép(s)  17 280 446 464 octets libres

C:\Users\Administrateur>runas /user:test cmd.exe
Entrez le mot de passe de test :
Tentative de lancement de cmd.exe en tant qu’utilisateur "WIN-TTEC51J8U2C\test" ...

C:\Users\Administrateur>dir C:\Users
 Le volume dans le lecteur C n’a pas de nom.
 Le numéro de série du volume est DC54-9388

 Répertoire de C:\Users

28/12/2017  13:55    <REP>          .
28/12/2017  13:55    <REP>          ..
28/12/2017  12:35    <REP>          Administrateur
14/07/2009  05:57    <REP>          Public
28/12/2017  13:55    <REP>          test
               0 fichier(s)                0 octets
               5 Rép(s)  17 277 493 248 octets libres
```

Analysis of the generation of a «test» user file

The presence of this file reveals that a user named « 整瑳 » has **authenticated** on the machine being analyzed.

The **creation** of a user within a Windows system is registered in the **log** file, « **Security** », under the identifier « **4720**—a user account has been created. »



Information	28/12/2017 15:59:30	Microso...	4720	Gestion des comptes d'utilisateur

Événement 4720, Microsoft Windows security auditing.

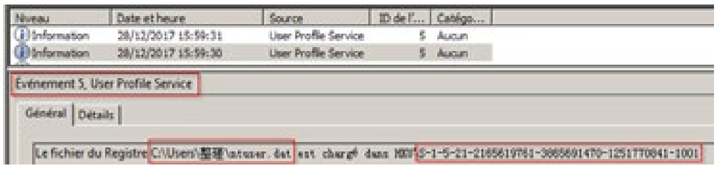
Un compte d'utilisateur a été créé.

An example of an event generated when a user is created on a Windows resource

However, during the investigation, the study of the « Security » log file on the resource being analyzed did not reveal **any information** about the creation of the user account, « 整瑳 ».

In fact, a **maximum size** is assigned to Windows logs; these are **rotated** if this limit is passed (i.e. the old logs are automatically overwritten, as and when required). « Security » logs are accessed frequently: so, it’s no surprise that the event is no longer present.

Nevertheless, the study of the « Microsoft-Windows-User Profile Operational » system log reveals that the « NTUSER.DAT» **user hive** present in the user folder « C:\Users\整瑳\ » is being **loaded** into the memory at **regular intervals**—which means the user is connecting regularly to the system.



Niveau	Date et heure	Source	ID de l'...	Catégo...
Information	28/12/2017 15:59:31	User Profile Service	5	Aucun
Information	28/12/2017 15:59:30	User Profile Service	5	Aucun

Événement 5, User Profile Service

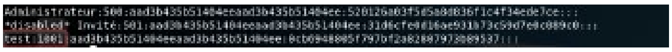
Le fichier du Registre C:\Users\整瑳\ntuser.dat est chargé dans HKU\S-1-5-21-2165619761-3865691470-1251770841-1001

Microsoft-Windows-User Operational-Profile Log Analysis

This event reveals the **SID** (« Security**ID**entifier ») associated with the user, « 整瑳 » :

/ S-1-5-21-2165619761-3865691470-1251770841-**1001**

The analysis of the system’s **SAM database** reveals that this **SID** is associated with a user named « **test**. » In addition, no user named « 整瑳 » appears to be configured on the system :



```
Administrateur:500:aad3b435b51404eeaad3b435b51404ee:520126a03f5d5a8d836f1c4f34ede7ce:::
*disabled* Invité:501:aad3b435b51404eeaad3b435b51404ee:31d6cfe0d16ae931b73c59d7e0c089c0:::
test:1001:aad3b435b51404eeaad3b435b51404ee:0cb6948805f797bf2a82807973b89537:::
```

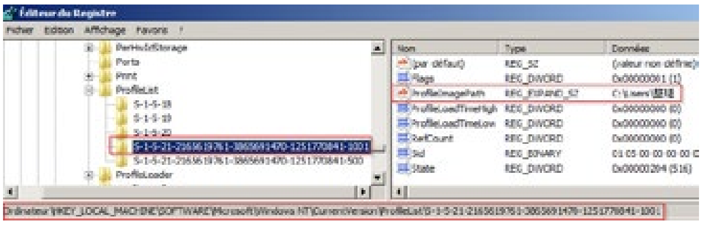
Excerpt from the SAM database

Microsoft documentation describes the following operation with respect to the allocation of SIDs on a Windows resource:

« For every local account and group, the SID is unique for the computer where it was created; no two accounts or groups on the computer ever share the same SID » (https://technet.microsoft.com/enus/ library/cc778824(v=ws.10).aspx).

This information suggests that **the users** « test » and « 整瑳 » are, in fact, the same, **single user**.

A registry scan reveals that **the profile path associated with the user** having the previously identified **SID** is the « C:\Users\整瑳 » **folder :**



Éditeur du Registre

Nom	Type	Données
(par défaut)	REG_SZ	(valeur non définie)
Flags	REG_DWORD	0x00000001 (1)
ProfileImagePath	REG_EXPAND_SZ	C:\Users\整瑳
ProfileLoadTimeHigh	REG_DWORD	0x00000000 (0)
ProfileLoadTimeLow	REG_DWORD	0x00000000 (0)
RefCount	REG_DWORD	0x00000000 (0)
Sid	REG_BINARY	01 05 00 00 00 00 0
State	REG_DWORD	0x00000204 (516)

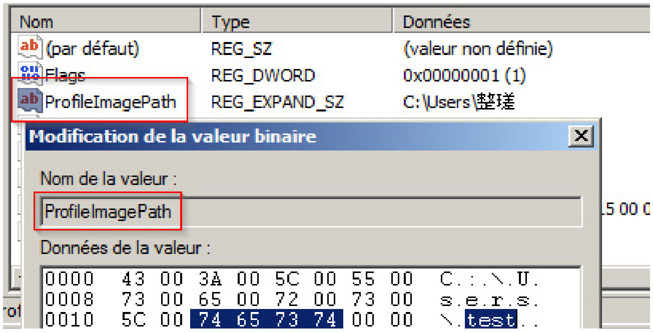
Ordinateur\HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\ProfileList\S-1-5-21-2165619761-3865691470-1251770841-1001

Données de la valeur « ProfileImagePath »

The user **account,** « **test** » therefore has the **folder** « C:\Users\整瑳 » as its personal folder.

An analysis of the **binary value** of the « **ProfileImagePath** » key reveals the presence of the character string, « **test** ». However, this also reveals that the « test » characters have **not** been correctly **encoded** in Unicode (which defines a character using two bytes).

In Unicode, the bytes “te” and “st” represent the two characters : 整瑳.

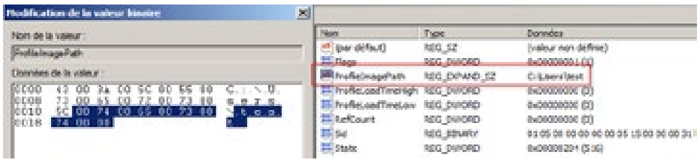


Nom	Type	Données
(par défaut)	REG_SZ	(valeur non définie)
Flags	REG_DWORD	0x00000001 (1)
ProfileImagePath	REG_EXPAND_SZ	C:\Users\整瑳

```
0000   43  00  3A  00  5C  00  55  00   C.:.\.U.
0008   73  00  65  00  72  00  73  00   s.e.r.s.
0010   5C  00  74  65  73  74  00  00   \.test..
```

Mauvais encodage de la chaîne de caractère Incorrect encoding of the string, « test », within the value, « ProfileImagePath »

Indeed, this value should be set to « test » (i.e. « 00 74 00 65 00 73 00 74 00 00 ») such that the « test » string is displayed correctly by the system:



```
0000   43  00  3A  00  5C  00  55  00   C.:.\.U.
0008   73  00  65  00  72  00  73  00   s.e.r.s.
0010   5C  00  74  00  65  00  73  00   \.t.e.s.
0018   74  00  00                       t.
```

Nom	Type	Données
(par défaut)	REG_SZ	(valeur non définie)
Flags	REG_DWORD	0x00000001 (1)
ProfileImagePath	REG_EXPAND_SZ	C:\Users\test
ProfileLoadTimeHigh	REG_DWORD	0x00000000 (0)
ProfileLoadTimeLow	REG_DWORD	0x00000000 (0)
RefCount	REG_DWORD	0x00000000 (0)
Sid	REG_BINARY	01 05 00 00 00 00 00 05 15 00 00 00 31
State	REG_DWORD	0x00000204 (516)

Correction de la valeur « ProfileImagePath » en unicode

Thus, it confirms that :

/ The user « 整瑳 » and « test » are one and the same user ;

/ The folder « C:\Users\整瑳 » is the personal folder of the user, « test ».

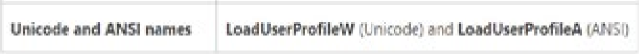
After discussions with the in-house team, the resource’s administrators tell us that the « test » account is operated as a **service account**, using a **third-party software**.

So, it’s highly likely that the cause of this bug is **an implementation error** within the **function that generates** the **service account’s personal folder**, and was created during **the installation of the software**.

A timeline of system events generated using the plaso tool, shows that the « test » account’s registration in the SAM database and the creation of the user folder « C:\Users\整瑳 » are linked: the two events occurred **simultaneously**. Nevertheless, no trace of the installation of the software in question could be recovered.

Lastly, it was possible to **reproduce** this **bug** by slightly modifying the source code, which is available via the following link : https://support.microsoft.com/en-us/help/196070/how-to-programmatically-cause-the-cre ation-of-a-users-profile.

Indeed, a **similar behavior** (the creation of a user having a non-encoded file in Unicode format) could be **obtained** by replacing the call to the LoadUserProfileA() function by one to the LoadUserProfileW() function :



Unicode and ANSI names	LoadUserProfileW (Unicode) and LoadUserProfileA (ANSI)

Excerpt from LoadUserProfile function documentation

Therefore, it is very possible that **an implementation error** has crept into the third-party software deployed on the resource.

In fact, an error occurred when calling the LoadUserProfile function: the LoadUserProfileW function was called instead of the LoadUserProfileA function. However, the program probably had to handle ANSI strings, thus generating an **encoding error** when creating the user folder associated with the service account configured by the software.

CONCLUSION

The presence of a Chinese file on a system does not necessarily mean that it has been compromised; having said that, such an occurrence should still be **investigated** thoroughly. In fact, this particular event can be linked to incorrect implementation of the software deployed on the system.

What’s more, special attention must be paid when using functions sensitive to **encoding**, such that the introduction of **bugs,** and potential **edge effects**, can be avoided when executing such programs. These kinds of behaviors can be avoided by making use of the latest features offered by **Windows APIs** ; in particular, the CreateUserProfile function can be used to create user profiles on Windows systems more recent than Windows Vista. The latter does not differ according to the encoding required (ANSI or Unicode).

Lastly, it’s important to **centralize** real-time system logs on a **dedicated resource**, and then **store them externally**, to **avoid** any **loss of the records**. This facilitates **investigations** if an incident occurs.

François LELIEVRE,
Consultant CERT-W

HITB2017 - WHEN TWO-FACTOR AUTHENTICATION IS A FOE: BREAKING APPLE'S ICLOUD KEYCHAIN

A short presentation (30 minutes) on this was given at the Hack In The Box 2017 conference in Amsterdam by Vladimir Katalov, CEO of ElcomSoft, a Moscow-based company specializing in password recovery, and analysis/restoration of encrypted, protected systems. ElcomSoft works for the following bodies:

- The NSA (The US National Security Agency);
- EUROPOL;
- INTERPOL;
- And others.

In the course of this half hour, Vladimir Katalov presented the encryption model implemented in Apple devices, focusing on the recovery of secrets via iCloud mechanisms.

INTRODUCTION

Given that large amounts of data are present on smartphones (contacts, calendars, call history, etc.), there is a need, at times, to recover it, especially during an investigation.

Acquiring the data (whether encrypted or not) is always the first step, before secrets can be extracted. This can be done using a number of different methods :

- **Acquisition via JTAG/Chip-Off/Hex dump :** these procedures make it possible to extract the memory from the device using physical access ; they cannot be used when data is encrypted. Also, errors can occur during capture, resulting in the recovery of a corrupted image, or damage to the memory being extracted.
- **Acquisition using software :** these procedures aim to use a software tool to extract a device's memory. As well as offering more limited possibilities than physical extraction, there are often limitations due to the compatibility of software. In addition, using such software means that the screen lock pattern has to be bypassed.
- **Using "cloud" functionality :** this method enables data recovery without the need for physical access to the device. It retrieves the data saved on cloud-type interfaces (iCloud, Google Drive, etc.).

DATA BACKED UP IN THE CLOUD

More and more data, on smartphones and other mobile devices, is being synchronized on cloud platforms (which are offered by market leaders such as Google, Apple and Microsoft). The data transmitted can be of different types :

- **Backup data :** this data corresponds to elements saved to restore an entire system. For this type of backup, almost all data is transmitted to the cloud platform.
- **Synchronization data :** this data corresponds to information shared between the different devices synchronized on a cloud-based account. It can also be of different types (which are detailed below).
- **Shared data :** this data corresponds to files shared on the cloud ; only files selected by the user are saved (the Google Drive model).

BACKUP DATA

Backup data mainly contains the complete system images for the device (especially on Apple). However, the following data is not commonly available :

- Third-party application data;
- Passwords (or those stored in additional backed up files).

However, in the majority of cases, there is no native functionality that allows the recovery of the saved images (in particular via the iCloud-type web interfaces); third-party tools (such as the one developed by ElcomSoft) may be required to do this.

SYNCHRONIZED DATA

In addition to backups, other data is automatically backed up on cloud-based systems through synchronization functions ; this includes :

- Contacts ;
- Call history ;
- Messages (SMS, iMessage, Hangouts, etc.) ;
- Emails;
- Internet activity (browsing history, etc.) ;
- Passwords ;
- And others.

Although the data saved by this method is more limited than that for full backups, a good deal of sensitive and potentially useful data can be retrieved. Moreover, such data is often replicated on other equipment which can then provide a further means of attack.

APPLE KEYCHAIN PROTECTION

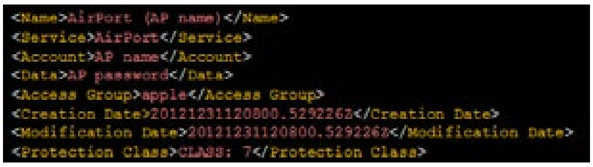
Apple's Keychain feature allows user secrets to be kept secure, in particular by limiting application password entries (on Safari, Wi-Fi, etc.). The passwords and secrets are then stored in the Keychain and users can access them on request. The Keychain management methods differ slightly between OSs : ▼

Platform	Encryption	Visualization	Extraction
iOS	According to user parameters	No native options	No option to export
macOS	Default encryption by the logon password used to open the session	Using the Keychain management tool	No option to export
iCloud	Encrypted	Need to synchronize a device (no visualization via web interface)	No option to export

Apple Keychain protection as a function of platforms

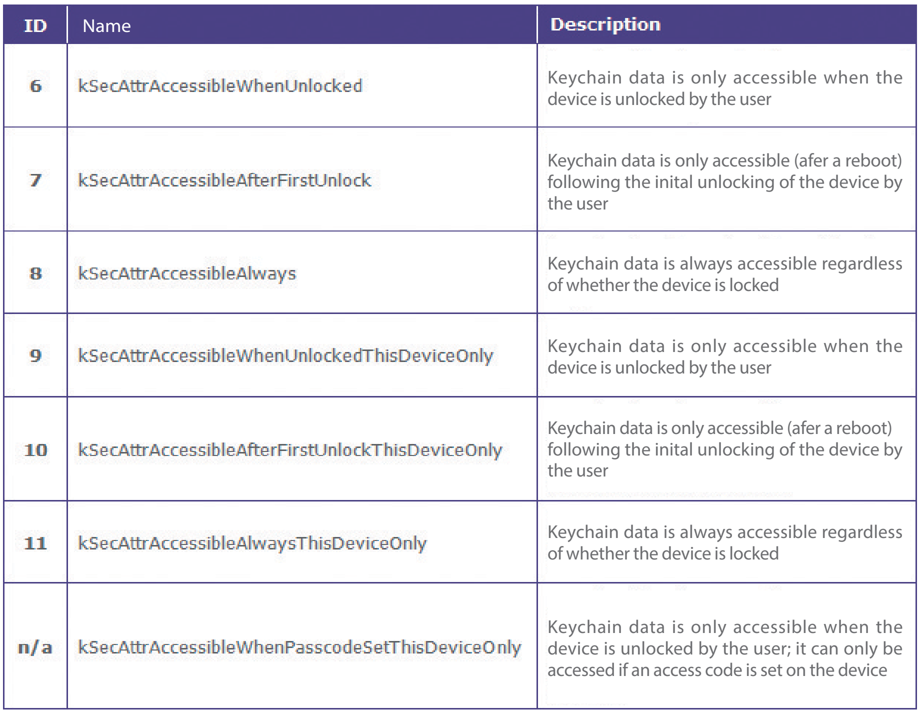
KEYCHAIN IOS

On the iOS platform, the Keychain is an XML file, and has the following structure: ▼



```
<Name>AirPort (AP name)</Name>
<Service>AirPort</Service>
<Account>AP name</Account>
<Data>AP password</Data>
<Access Group>apple</Access Group>
<Creation Date>20121231120800.5292262</Creation Date>
<Modification Date>20121231120800.5292262</Modification Date>
<Protection Class>CLASS: 7</Protection Class>
```

Different classes of protection can be implemented, including:



ID	Name	Description
6	kSecAttrAccessibleWhenUnlocked	Keychain data is only accessible when the device is unlocked by the user
7	kSecAttrAccessibleAfterFirstUnlock	Keychain data is only accessible (afer a reboot) following the inital unlocking of the device by the user
8	kSecAttrAccessibleAlways	Keychain data is always accessible regardless of whether the device is locked
9	kSecAttrAccessibleWhenUnlockedThisDeviceOnly	Keychain data is only accessible when the device is unlocked by the user
10	kSecAttrAccessibleAfterFirstUnlockThisDeviceOnly	Keychain data is only accessible (afer a reboot) following the inital unlocking of the device by the user
11	kSecAttrAccessibleAlwaysThisDeviceOnly	Keychain data is always accessible regardless of whether the device is locked
n/a	kSecAttrAccessibleWhenPasscodeSetThisDeviceOnly	Keychain data is only accessible when the device is unlocked by the user; it can only be accessed if an access code is set on the device

Data protection attributes

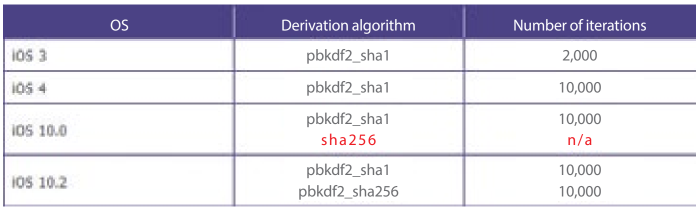
Note: The ThisDeviceOnly attribute is used to encrypt data using a device-specific hardware key. This can be extracted from 32-bit devices only.

ITUNES BACKUP FILES

The iTunes tool, developed by Apple, allows users to create local backups (on a computer) of data from mobile devices such as iPhones, iPods and iPads. Large amounts of data are saved in this way, in particular, certain user secrets (from Keychain).

In order to limit the risks of being recovered, the secrets are saved in an encrypted fashion (using the AES encryption mechanism) and kept in files named« manifest.plist » ; the encryption key is, in turn, present in the « BackupKeyBag » attribute, itself protected by a key derived from the password set up by the user.

The keyBag encryption/decryption key is generated as follows :



OS	Derivation algorithm	Number of iterations
iOS 3	pbkdf2_sha1	2,000
iOS 4	pbkdf2_sha1	10,000
iOS 10.0	pbkdf2_sha1 sha256	10,000 n/a
iOS 10.2	pbkdf2_sha1 pbkdf2_sha256	10,000 10,000

Cryptographic mechanisms used as a function of the iOS

Note: A weakness introduced on iOS 10.0 allowed the storage, in an unprotected database, of the cryptographic condensate (sha256) from the encryption password.

ICLOUD DATA PROTECTION

Most of the data transmitted to the iCloud platform is protected through the AES encryption mechanism, using a 128-bit key. The iCloud Keychain uses a 256-bit key to store sensitive information (bank details, passwords, etc.). The encryption key is stored, unencrypted, in the block containing the backed-up data.

In order to secure the stored data, a number of security measures have been put in place, including :

- Email notifications, which are sent when the data is accessed (except when access is via an authentication token) ;
- An account locking mechanism to limit fraudulent access (with blocking in the case of suspicious activity) ;
- The option of using two-factor authentication mode ;
- And others.

Note: The authentication token can be retrieved from other equipment on which the same account has been used.

DATA PROTECTION WHEN TWO-FACTOR AUTHENTICATION IS NOT ACTIVATED

Apple offers a two-factor authentication mode but it can be disabled, and an account access password can then be set. The security code is, in fact, optional, but if the user doesn't set such a code, they have to use a device that is already authorized in order to confirm the addition of a new device.

The main advantage of using a security code lies in the fact that the keychain can be sent to the "Apple Escrow" service (discussed in the next section), which allows the recovery of data (as well as the associated phone numbers) if the device is lost.

Note: Although this mode of operation has no inherent vulnerabilities, Vladimir Katalov recommends not using it because errors frequently occur during the parameterization phase.

DATA PROTECTION WITH TWO-FACTOR AUTHENTICATION ENABLED

Two-factor authentication mode is easier than the method described above; users simply need to choose a phone number on which they will receive confirmation notifications. This cannot be done on the iCloud web interface: only from an Apple device.

In addition, when adding a new device to the same iCloud account, users have to enter the unlock code of a previously authorized device.

HOW ICLOUD KEYCHAIN WORKS

iCloud Keychain enables users to synchronize their passwords securely between multiple iOS devices and Mac computers (without disclosing this information to Apple). This protocol was designed to protect against the following scenarios :

- The user's iCloud account being compromised ;
- The iCloud service being compromised (by an employee or external attacker) ;
- Third-party access to the user's accounts.

THE CIRCLE OF TRUST

To allow data sharing, a circle of trust is established between the different devices (provided two-factor authentication is enabled). This is done using the following mechanism :

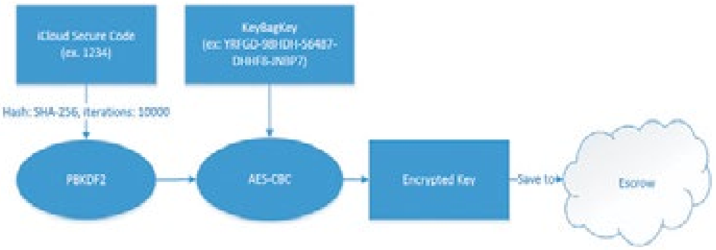
- Initializing the circle of trust :
    - The first device establishes the circle of trust ;
    - It creates a synchronization identity (a public/private key pair) ;
    - The public key is placed in the circle and signed by :
        - The private key of the synchronization authorization itself ;
        - An elliptic-curve asymmetric key (via P256) generated from the iCloud account password.

        The parameters used for generation are also stored in the circle.
    - The signed circle is placed in a storage zone (KVS) on iCloud.
- Activating the circle on another device :
    - The device creates its pair of synchronization identification keys ;
    - The device creates an application ticket (consisting of the public key from the device's synchronization identity) ;
    - The user authenticates themself via the submission of their iCloud password ;
    - If authentication is successful, the ticket is placed in iCloud.
    - When the first device notices the presence of the ticket, it notifies the user to confirm that a request has been made ;
    - Following confirmation, the user must enter their iCloud password to verify the application ticket ;
    - If this is successful, the first device adds the new member's key to the circle of trust by signing it with :
        - Its synchronization key ;
        - The secret key generated from the iCloud password.
- Addition of further devices :
    - The procedure used for the second device is repeated (and all previously authorized devices can then be used for verification).

Conversely, only items with the kSecAttrSyncronizable attribute are synchronized by this mechanism, in order to limit the replication of some information where there is a greater need for privacy. Implementing this protocol then allows the transfer of secrets between devices without Apple having the option to recover the data (at least not directly).

THE "ESCROW PROXY" SERVICE

The « escrow proxy » service allows users to recover passwords; however, it can only be used after a security code has been established (which is, by default, a series of four digits). The keychain encryption key is transmitted to the escrow service and encrypted with the following security code :



Transmission of the encryption key to the escrow service

The "escrow" service API can then carry out various actions, such as :

- Adding a record ;
- Retrieving a record ;
- Retrieving the list of trusted phone numbers ;
- Recovering the data (in a case of loss) ;
- And others.

When a user wants to recover saved data, the following procedure is used :

- The (authenticated) user retrieves a token through a GetAccountSettings call ;
- The user makes use of this recovered token to synchronize with the service (in order to retrieve only the necessary data) ;
- The user authenticates via the SRP protocol: this step is complex but it does not trigger a notification, and does not require any of the following:
    - An authorized device (although the security code of one of the authorized devices is necessary) ;
    - The iCloud security code ;
- The user retrieves the data and performs the necessary decryption steps.

Thus, an attacker who can recover an iCloud service access token can then recover the stored data without triggering a notification to the legitimate user.

CONCLUSION

Currently, there are a range of methods to recover stored data :

- Adding new equipment to the « circle of trust » : the two-factor authentication step then has to be bypassed or validated, which triggers notifications on the other devices.
- To recover the iCloud backups (provided they exist), it's then necessary to:
    - Pass two-factor authentication (as above);
    - Retrieve the encryption key of one of the devices (recoverable only on 32-bit mobiles).
- Accessing a local backup: however, it's then necessary to:
    - Have physical access to a device (a PC or Mac);
    - Possess the decryption key (if the backup is protected).
- Finding a vulnerability in the management protocol of the circle of trust: the way in which this last point was presented suggests that the protocol may be vulnerable, but no additional information was given during the conference. The vulnerability in question may be of a cryptographic nature and related to the choice of NIST P-256 elliptic-curve parameters.

Mahdi BRAIK, Senior Consultant

Sources

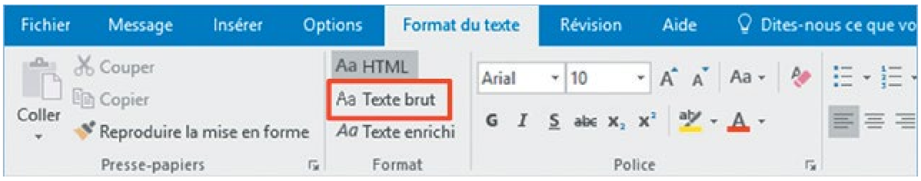
- http://conference.hitb.org/hitbsecconf2017ams/materials/D1T4%20-%20Vladamir%20Katalov%20-%20Breaking%20Apple%e2%80%99s%20iCloud%20Keychain.pdf
- https://images.apple.com/fr/business/docs/iOS_Security_Guide.pdf
- http://ogryb.blogspot.fr/2014/11/why-i-dont-trust-nist-p-256.html
- http://safecurves.cr.yp.to/rigid.html

CVE-2017-11776 : OUTLOOK VS S/MIME

This article discusses the topic of CVE 2017-11776 – a vulnerability in Microsoft Outlook – its impacts, and the corrective actions that can be taken after the event.

A VULNERABILITY RELATED TO S/MIME ENCRYPTION IN OUTLOOK

A security vulnerability in Outlook related to S/MIME encryption of emails has been identified (CVE-2017-11776). It concerns encrypted emails sent by Outlook in "plain text" format :



For about six months, encrypted mails sent in « plain text » format were sent in both encrypted and unencrypted form by Outlook in the body of the email sent. Meanwhile, emails sent in « HTML » format were correctly encrypted in their entirety.

THE EXTENT OF THE VULNERABILITY

This vulnerability is only relevant when Outlook has sent an email in plain text format with S/MIME encryption. If another mail client (Thunderbird, Webmail, etc.) has sent an encrypted email, this email will be correctly encrypted.

The scope for the transmission of the unencrypted message varies:

- In Outlook using Exchange:
  - When the recipient is not in the same mail domain, an unencrypted message is sent to the first mail
  - relay (MTA) and then deleted
  - When the recipient is in the same mail domain, the unencrypted message is sent to the first mail
  - relay (MTA) and then forwarded to the recipient's inbox.

Where Outlook uses SMTP, the unencrypted message is sent along the entire mail transmission chain to the recipient's inbox.

A PowerShell script has been developed by Wavestone to enable an Outlook mailbox, containing mails encrypted with S/MIME in « plain text » format, to be searched. It is available at this address and requires Outlook to be opened when executed.

FIXES

This vulnerability has been corrected in the following Outlook updates:

- Deferred Channel : Version 1705 (Build 8201.2200) - published on October 10, 2017
- Monthly Channel : Version 1708 (Build 8431.2107) - published on October 10, 2017

Cyprien OGER, Senior Consultant

Sources

https://www.sec-consult.com/en/blog/2017/10/fake-crypto-microsoft-outlook-smime-cleartext-disclosure-cve-2017-11776/index.html

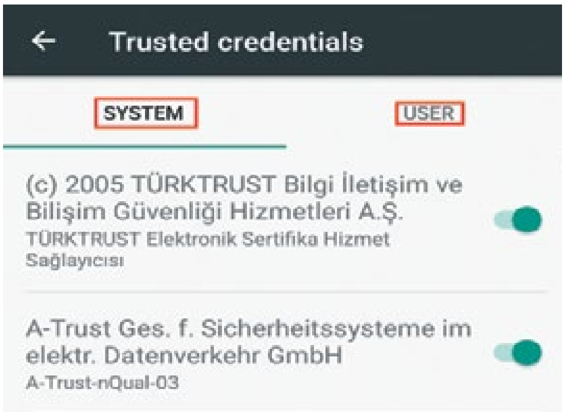
WHAT HAS CHANGED FOR CERTIFICATE PINNING AFTER ANDROID 7 NOUGAT

The seventh version of Android, « Nougat », classed as « API level 24 » and released in August 2016, introduced significant changes in the security of SSL/TLS communication streams and, in particular, around certificate pinning.

To recap, certificate pinning is a security measure aimed at limiting the impact of a Certification Authority (CA) compromise by defining, precisely, on the client side, which certificate, or certification chain, is expected (https://www.riskinsight-wavestone.com/2013/04/epinglez-vos-certificats).

WHAT CHANGES FOR DEVELOPERS

Put simply, up to this point, any Android application was blindly trusting the certificates present in the two stores available on a terminal, namely the « system » database, provided by the AOSP project (https://android.googlesource.com/platform/system/ca-certificates/), and the « user » database, containing custom certificates, and editable by a terminal user.



← Trusted credentials

SYSTEM USER

(c) 2005 TÜRKTRUST Bilgi İletişim ve Bilişim Güvenliği Hizmetleri A.Ş.
TÜRKTRUST Elektronik Sertifika Hizmet Sağlayıcısı

A-Trust Ges. f. Sicherheitssysteme im elektr. Datenverkehr GmbH
A-Trust-nQual-03

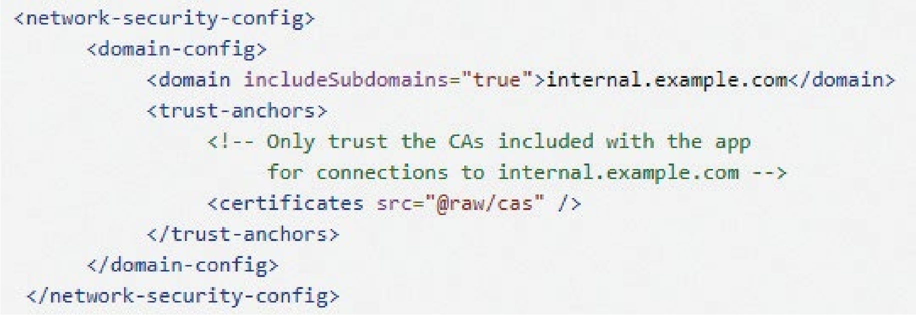
A developer had to apply a specific section of code in their application to perform pinning: without experience

in this area, a specific development could prove to be totally ineffective (https://www.synopsys.com/blogs/software-security/ineffective-certificate-pinning-implementations/). We advise developers who want to implement such a measure to use the examples provided by OWASP (https://www.owasp.org/index.php/Certificate_and_Public_Key_Pinning).

From now on, when an application is compiled using Android 7 as target version, it will no longer trust the « user » store by default, and must explicitly define the CAs recognized as being valid, for which there are several options:

- / Debugging only : in this case, the application must be compiled with the instruction, « debuggable=true », in the manifest
- / By domain
- / For a domain list
- / For all areas without exception

The following example, taken from the following blog post (https://android-developers.googleblog.com/2016/07/changes-to-trusted-certificate.html), illustrates how to declare a valid CA or the « internal example.com » domain:



```
<network-security-config>
    <domain-config>
        <domain includeSubdomains="true">internal.example.com</domain>
        <trust-anchors>
            <!-- Only trust the CAs included with the app
                 for connections to internal.example.com -->
            <certificates src="@raw/cas" />
        </trust-anchors>
    </domain-config>
 </network-security-config>
```

WHAT IS CHANGING FOR AUDITORS/PENTESTERS

Historically, anyone who wanted to analyze/audit an application's web streams, and was not able to compile the target application, had to:

1. Use a web proxy; Burp Suite, for example
2. Add the CA of this proxy to the « user » store in order to be able to intercept encrypted SSL/TLS streams
3. Disable the certificate pinning feature within the application, in particular by patching and recompiling the application (https://medium.com/@felipecsl/bypassing-certificate-pinning-on-android-for-fun-and-profit-1b0d14beab2b)
4. Redirect device streams to the proxy ; for example, through Wi-Fi connectivity settings

After Android 7, Step 2 has now become ineffective, and auditors have several options :

- / The simplest one: continue to use the old method, with an Android 6 device, provided that the target mobile application allows this version.
- / A slightly more complex one : add the Web proxy CA in the « system » certificate store ; this method is explained here (https://nvisium.com/blog/2017/07/12/advantages-and-disadvantages-of-android-n-network-security-configuration/) and here (https://blog.jeroenhd.nl/article/android-7-nougat-and-certificate-authorities) and consists of modifying the file contained in «/system/etc/security/cacerts/» of the system partition. The TEMPLATE FOCUS important prerequisite is to have root rights on the target device in order to be able to remount the partition in both read and write.
- / Or, an even more complex one—but the most technically advanced : using the dynamic-instrumentation framework, « Frida » (https://www.frida.re/) with a generic script for deactivation of the pinning (https://techblog.mediaservice.net/2017/07/universal-android-ssl-pinning-bypass-with-frida/).

A detailed example of the application of this method is available here : (https://blog.it-securityguard.com/the-stony-path-of-android-%F0%9F%A4%96-bug-bounty-bypassing-certificate-pinning/)

As a result of well-intentioned auditors complicating the interception procedure of the streams, these modifications to certificate management after Android 7 also increase interception options for attackers while facilitating the work of the developers, with the overall goal of building trust in the world's most widely used mobile platform (https://www.idc.com/promo/smartphone-market-share/os).

Thomas DEBIZE, Senior Consultant

CERT-Wavestone combines a range of expertise and business lines to provide a comprehensive response to security incidents. More than 45 experienced profiles can be mobilized within CERT-Wavestone.

Find the publications of our experts on www.securityinsider-wavestone.com

twitter@secuinsider

REACTION TO AN ATTACK, OR SUSPECTED ATTACK

- Digital/forensic investigation
- IS and business-function crisis management
- Development of remediation plans

THREAT INTELLIGENCE

- Evaluation of the attractiveness of the business as a target
- Analysis and understanding of attacks
- Watch and learn: cyber crime monitoring

DEFENSE PREPARATIONS AND CRISIS MANAGEMENT

- Definition and animation of CERT and SOC processes
- Red Team and Purple Team
- Crisis simulations

www.wavestone.com

Wavestone is a consultancy firm, formed by the combination of Solucom and the European businesses of Kurt Salmon (excluding consultancy for in the retail & consumer goods sectors).

Wavestone is one of the leading independent consultancy firms in Europe.

Wavestone's mission is to enlighten and guide its customers in their most strategic decisions through relying on its threefold functional, sectoral and technological expertise.