



Audit d'Active Directory avec BTA

Philippe Biondi, Joffrey Czarny — Airbus Group Innovations

SSTIC — 4–6 juin 2014

Sommaire

- 1 Enjeux
- 2 BTA
 - Introduction
 - Import
 - Les miners
 - Le rapport
- 3 Méthodologie d'audit
- 4 Retour d'expérience

Sommaire

- 1 Enjeux
- 2 BTA
 - Introduction
 - Import
 - Les miners
 - Le rapport
- 3 Méthodologie d'audit
- 4 Retour d'expérience

Active Directory

Rôle de l'Active Directory

- Authentification et autorisation des utilisateurs et des machines
 - Politiques de sécurité
 - Base de référence
- ⇒ Pierre angulaire du SI Microsoft
- ⇒ Cible de choix

AD et sécurité

Objectifs

- Respect des bonnes pratiques
- Nettoyage de printemps
- État des lieux après une compromission
- Nettoyage après une compromission

Besoins

Besoins

- Identifier les mauvaises pratiques (affaiblissement du niveau de sécurité)
- Aider au nettoyage
- Trouver les anomalies

Démarche

- Lister
 - les administrateurs de domaine
 - les délégations
 - les comptes qui n'ont jamais servi
 - ...
- Vérifier avec les admins AD

Sommaire

- 1 Enjeux
- 2 BTA**
 - Introduction
 - Import
 - Les miners
 - Le rapport
- 3 Méthodologie d'audit
- 4 Retour d'expérience

BTA

- Airbus {Group {CERT|Innovations}|DS CyberSecurity}
- Open Source (GPLv2)
- <https://bitbucket.org/iwseclabs/bta>
- `pip install bta`

BTA permet de répondre aux problèmes suivants :

- Accès rapide et non filtré à l'ensemble des données AD
- Travail hors-ligne
- Points de contrôle pré-établis ($\neq$ outil exploratoire)
- Déterminisme
- Itérabilité, suivi dans le temps
- Modularité

Comparaison avec AD-perm

AD-perm¹

- Outil exploratoire
- Utile quand on ne sait pas à l'avance ce qu'on cherche (pentest, forensics, ...)

BTA vs AD-perm

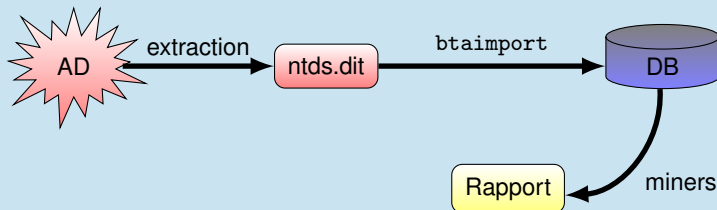
- AD-perm=*debugger*, BTA=*framework* de tests unitaires et de non régression
- AD-perm trouve les points de contrôle, BTA les cristallise

⇒ Approches complémentaires

¹Présenté au SSTIC 2012, <https://github.com/ANSSI-FR/AD-permissions>

Vue d'ensemble du fonctionnement de BTA

Architecture de BTA, vue d'ensemble



`btaimport` *NTDS.dit* → base mongo + postprocessing

`btamanage` Intendance sur les bases NTDS importées

`btaminer` Appel des *miners*

`btadiff` Différentiel entre deux instances de l'AD intégrées en base

Import

Structure du fichier ntds.dit

ntds.dit

- Format JetDB
- Une douzaine de tables
 - exploitées: `datatable`, `sd_table`, `link_table`
 - non-exploitées: `MSysObjects`, `hiddentable`, `quota_table`, ...

Lecture

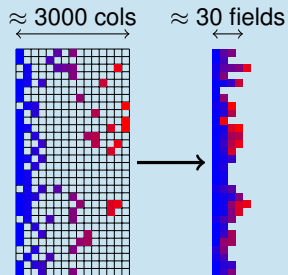
- Utilisation de la `libesedb`²

²<https://code.google.com/p/libesedb/>

Import

Principe de l'import

Import de la datatable du fichier ntds.dit



- impossible de stocker cette table dans une DB relationnelle classique
trop de colonnes
- table creuse → représentation compacte
- ntds.dit → 1 database dans MongoDB
document based DB
- 8 Go → 26 Go
index, champs décodés, données pré-calculées

Import

Commande d'import: `btaimport`

```
$ btaimport -C ::mabase /chemin/vers/mon/ntds.dit
```

```
$ btaimport --C-list ::mabase1,::mabase2 \  
          ad/base1.ntds.dit ad/base2.ntds.dit  
INFO : Going to import ::mabase1      <- ad/base1.ntds.dit  
INFO : Going to import ::mabase2      <- ad/base2.ntds.dit  
Can I carry on ? (y/n)
```

```
$ btaimport --C-from ::%s "basename rmallext" ad/*dit  
INFO : Going to import ::backdoor1    <- ad/backdoor1.ntds.dit  
INFO : Going to import ::backdoor2    <- ad/backdoor2.ntds.dit  
INFO : Going to import ::clean        <- ad/clean.ntds.dit  
Can I carry on ? (y/n)
```

Import

Aperçu des données en base Mongo: un champ de la datatable

```
{ "cn" : "ACS-Enable-ACS-Service",
  "LDAPDisplayName" : "aCSEnableACSService",
  "name" : "ACS-Enable-ACS-Service",
  "adminDescription" : "ACS-Enable-ACS-Service",
  "adminDisplayName" : "ACS-Enable-ACS-Service",
  "isVisibleInAB" : 42,
  "objectClass" : [ 196622, 65536 ],
  "schemaIDGUID" : "7f561287-5301-11d1-a9c5-0000f80367c1",
  "objectGUID" : "925af73d-e447-40c0-9655-b5a8603fb49f",
  "time_col" : ISODate("2009-02-11T18:37:08Z"),
  "distinguishedName" : 23,
  "systemFlags" : 16,
  "nTSecurityDescriptor" : 7,
  "RDNTyp_col" : 3,
  "isSingleValued" : 1,
  "instanceType" : 4,
  "oMSyntax" : 1,
  "uSNCreated" : 15,
  "recycle_time_col" : NumberLong("3038287259199220266"),
  "whenCreated" : ISODate("2009-02-11T18:37:08Z"),
  "replPropertyMetaData" : BinData(0,"AQAAAAAAAAATAAAAAAAAAAAAAABAAAAC+mLCAMAAAAvmvLtKEtaQqTKmYSWdi8vDwAAAAAAAAAPAAAAAAAAAAM..."),
  "whenChanged" : ISODate("2009-02-11T18:37:08Z"),
  "PDNT_col" : 1811,
  "objectCategory" : 14,
  "Ancestors_col" : BinData(0,"AgAAAPsGAAD8BgAA/QYAABMHAAAXAAAA"),
  "NCDNT_col" : 1811,
  "uSNChanged" : 15 }
```

Les *miners*

Rôle du *miner*

Les *miners*

- Vérifient la cohérence de la base MongoDB par rapport aux requêtes qu'ils vont effectuer
- Passent en revue un ou plusieurs points de contrôle
- Remplissent un document structuré avec leurs résultats

Les mineurs

Liste des mineurs

```
$ btaminer -h
usage: btaminer [-h] [-C CNX] [-B mongo,ldap] [--force-consistency]
               [-live-output] [-t csvzip,excel,ReST] [-o FILENAME]
               [-e ENCODING] [--ignore-version-mismatch] [--module MODULE]

Audit_UAC,Audit_Schema,Domains,DNGrep,DNTree,ListACE,CanCreate,skeleton,SIDHistory,Schema,Audit_ExtRights,WhoIs,
Membership,AdminCountCheck,ListObject,ListGroup,info,passwords,MailBoxRights,NewAdmin,Search4Rights,Audit_Groups,
TrustLink,Audit_Passwords,SDProp,Audit_Full,SDusers,Audit_SDProp,CheckUAC>PasswordPolicy,Nopassword,accounts,
SchemaModifs
...

positional arguments:
Audit_UAC,Audit_Schema,Domains,DNGrep,DNTree,ListACE,CanCreate,skeleton,SIDHistory,Schema,Audit_ExtRights,WhoIs,Membership,
AdminCountCheck,ListObject,ListGroup,info,passwords,MailBoxRights,NewAdmin,Search4Rights,Audit_Groups,TrustLink,
Audit_Passwords,SDProp,Audit_Full,SDusers,Audit_SDProp,CheckUAC>PasswordPolicy,Nopassword,accounts,SchemaModifs

Audit_UAC      Miners
Audit_UAC      Run all analyses on User Account Control
Audit_Schema    Run all analyses on schemas
Domains        Display Informations about domains
DNGrep         DN grepper
DNTree         DN Tree
ListACE        List ACE matching criteria
CanCreate      This miner list all user who possess the right to
               create or delete objects

[...]
```

Les miners

Le *miner* info

```
$ btaminer -C ::test -t ReST info
```

```
Analysis by miner [info]
```

```
=====
```

```
Data format version: 1
```

```
collections in this database
```

```
-----
```

name	number of records
system.indexes	50
metadata	1
log	3
sd_table	94
sd_table_meta	4
link_table	54
link_table_meta	10
datatable	3516
datatable_meta	1182
category	234
domains	1
dnames	3515

[...]

Les mineurs

Le *miner* passwords

```
$ btaminer -t ReST -C ::mabase passwords --never-logged
```

```
Analysis by miner: [passwords]
```

```
=====
```

name	userAccountControl	
Invité	GUEST of labz (s-1-5-\\	accountDisable:True
intru	intru (s-1-5-21-11546//	accountDisable:False
krbtgt	KRBGT of labz (s-1-5\\	accountDisable:True
SystemMailbox{1f05//7}	SystemMailbox{1f05a92//121}	accountDisable:True
SystemMailbox{e0dc\\9}	SystemMailbox{e0dc1c2\\122}	accountDisable:True
DiscoverySearchMai/E09334BB852}	DiscoverySearchMailbo//50385761-1123)	accountDisable:True
FederatedEmail.4c1\\42	FederatedEmail.4c1f4d\\125)	accountDisable:True
auditor	auditor (s-1-5-21-115//	accountDisable:False

Les mineurs

Exemple de miner

```
1  from bta.miner import Miner
2
3  @Miner.register
4  class Skel(Miner):
5      _name_ = "skeleton"
6      _desc_ = "skeleton, list SD id and hashes when id < 50"
7      @classmethod
8      def create_arg_subparser(cls, parser):
9          parser.add_argument("--dummy", help="dummy option")
10         parser.add_argument("--dummy_flag", help="dummy flag", action="store_true")
11
12     def run(self, options, doc):
13         doc.add("Option dummy is %s" % options.dummy)
14
15         table = doc.create_table("my table")
16         table.add(["id", "hash"])
17         table.add()
18
19         for r in self.sd_table.find({"sd_id": {"$lt": 50}}):
20             table.add([r["sd_id"], r["sd_hash"]])
21         table.finished()
22
23     def assert_consistency(self):
24         Miner.assert_consistency(self)
25         self.assert_field_exists(self.sd_table, "sd_id")
26         assert self.datatable.find({"cn":{"$exists":True}}).count() > 10, "less than 10 cn in datatable"
```

Les *miners*

Groupe de *miners*: plusieurs résultats dans le même rapport

Lancer plusieurs *miners* dans la même ligne de commande

```
$ btaminer -t ReST -C ::mabase \
           passwords --never-logged \
           -- passwords --last-logon 1215 \
           -- passwords --last-logon 302
```

Appeler un groupe de *miners*

```
$ btaminer -t ReST -C ::mabase Audit_Passwords
```

Les *miners*

Exemple de groupe de *miners*

Le groupe de *miners* audit_password

```
1  from bta.miner import Miner, MinerList
2
3  @Miner.register
4  class ExtendedRights_Audit(MinerList):
5      _name_ = "Audit_Passwords"
6      _desc_ = "Run all analyses on passwords"
7      _report_ = [
8          ("passwords", "--never-logged"),
9          ("passwords", "--last-logon", "1215"),
10         ("passwords", "--last-logon", "485"),
11         ("passwords", "--last-logon", "302"),
12         ("passwords", "--bad-password-count"),
13     ]
```

Le rapport

La structure de document

Objet docstruct

Permet de représenter un document structuré en :

- sections et sous-sections (profondeur arbitraire)
- listes et sous-listes (profondeur arbitraire)
- tables

Le rapport

Utilisation par les *miners*

- Le *miner* reçoit sa racine de document en paramètre
- Il y crée les sections, listes et tables dont il a besoin
- Ce rapport peut faire partie d'un rapport plus gros

Le rapport

Exemple

```
1  def run(self, options, doc):
2      doc.add("Quelques lignes d'introduction du miner")
3      doc.add("Remarques pertinentes sur le contenu de la base")
4
5      s1 = doc.create_subsection("Premiere partie de l'analyse")
6      s2 = doc.create_subsection("Deuxieme partie de l'analyse")
7      table = s1.create_table("ma table")
8      lst = s2.create_list("ma liste")
9      table.add(["decimal", "hexadecimal"])
10     table.add()
11     for i in range(15):
12         table.add(["%i"%i, "%x"%i])
13     table.finished()
14     s1.finished()
15     s2.add("Je sais aussi compter a l'envers")
16     for i in range(20,0,-1):
17         lst.add("%i"%i)
18     lst.finished()
19     s2.finished()
```

Le rapport

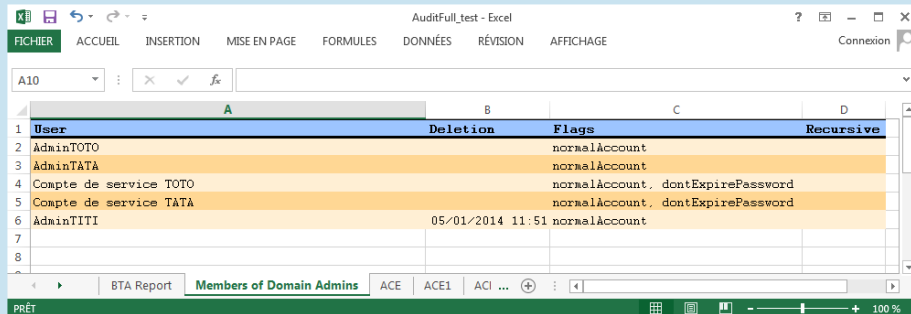
Formats de sortie

- Au fil de l'eau
- ReStructuredText
- Archive ZIP de CSV et de TXT
- Excel (XLSX)

Le rapport

Exemple: Sortie Excel

```
btaminer -C ::mabase -t excel -o mon_rapport.xlsx Audit_Full
```



The screenshot shows an Excel spreadsheet with the following data:

User	Deletion	Flags	Recursive
AdminTOTO		normalAccount	
AdminTATA		normalAccount	
Compte de service TOTO		normalAccount, dontExpirePassword	
Compte de service TATA		normalAccount, dontExpirePassword	
AdminTITI	05/01/2014 11:51	normalAccount	

Sommaire

- 1 Enjeux
- 2 BTA
 - Introduction
 - Import
 - Les miners
 - Le rapport
- 3 Méthodologie d'audit**
- 4 Retour d'expérience

Les grandes étapes

Extraction du fichier NTDS.dit d'un contrôleur de domaine

- Via *ntdsutil* sous les environnements 2008
- Via *vssadmin* sous les environnements 2003

Import du fichier NTDS.dit en base

- *btainport* se charge de l'import des données dans une base MongoDB
- Pré-traitement des données en base et ajout de nouvelles collections

Exécution de requêtes en base et corrélation des résultats

- *btaminer* permet de requêter les données en base
- Vérification des résultats avec un administrateur Active Directory

L'import du fichier NTDS.dit

Extraction en environnement Windows 2008

- *ntdsutil* et la méthode IFM permet d'effectuer une duplication du fichier NTDS.dit

```
C:\>ntdsutil
activate instance ntds
ifm
create full c:\NTDS_saved
quit
quit
```

Extraction en environnement Windows 2003

- *vssadmin* permet d'effectuer un *snapshot* du système de fichier.
- *esentutl* permet quant à lui de fermer et de mettre à jour de la base ESE.

Quelques points de contrôles

btaminer

- Vérification sur les membres des groupes *Domain Admins*, *Enterprise Admins*...:

```
btaminer -C ::snktest ListGroup -match "Domain Admins"
```

- Liste des objets protégés par *AdminSDHolder*

```
btaminer -C::snktest SDProp -list
```

- Vérification des ACE liés à *AdminSDHolder*

```
btaminer -C ::snktest SDProp -checkACE
```

- Vérification de la santé du schéma

```
btaminer -C ::snktest Schema -owner
```

```
btaminer -C ::snktest Schema -timelineAS changed
```

```
btaminer -C ::snktest Schema -timelineCS created
```

Quelques points de contrôles

btaminer

- Vérification sur les droits étendus

```
btaminer -C ::snktest ListACE -type 00299570-246d-11d0-a768-00aa006e0529
```

- Liste des comptes qui ne se sont jamais connectés à l'AD

```
btaminer -C ::snktest passwords -never-logged
```

- Liste des comptes qui ne se sont pas authentifiés sur l'AD depuis 6 mois

```
btaminer -C ::snktest passwords -last-logon 182
```

- Nombre de tentatives infructueuses de connexion par compte

```
btaminer -C ::snktest passwords -bad-password-count
```

- Liste des comptes disposant d'un flag *UserAccountControl* particulier

```
btaminer -C ::snktest CheckUAC -check passwdCantChange
```

Contrôles des groupes sensibles

```
$ btaminer -C::snktest -t ReST ListGroup --match "Admins du domaine"
```

```
Analysis by miner [ListGroup]
```

```
:=====
```

```
List of groups matching [Admins du domaine]
```

```
Group Admins du domaine
```

```
-----
```

```
sid = S-1-5-21-1154669122-758131934-2550385761-512
```

```
guid = 8bff35e5-87ff-4d9f-b979-122adf32cdd9
```

```
dn = .intra.secu.labz.Users.Admins du domaine
```

```
+=====+
```

```
| snorky | | normalAccount | |
```

```
| Administrateur | | normalAccount | |
```

```
+-----+
```

```
User snorky (S-1-5-21-1154669122-758131934-2550385761-1154)
```

```
+-----+
```

```
| Trustee | Member | ACE Type | Object type |
```

```
+-----+
```

```
| Admins du domaine | snorky | AccessAllowedObject | (none) |
```

```
[...]
```

```
| Everyone | snorky | AccessAllowedObject | User-Change-Password |
```

```
| Jean Dupond | snorky | AccessAllowedObject | (none) |
```

```
| Self | snorky | AccessAllowedObject | User-Change-Password |
```

```
| Self | snorky | AccessAllowedObject | Private-Information |
```

```
| Admins du domaine | snorky | AccessAllowed | (none) |
```

```
| Administrateurs | snorky | AccessAllowed | (none) |
```

```
| System | snorky | AccessAllowed | (none) |
```

```
| Everyone | snorky | SystemAudit | (none) |
```

```
| Everyone | snorky | SystemAuditObject | GP-Link |
```

```
| Everyone | snorky | SystemAuditObject | GP-Options |
```

```
+-----+
```



Contrôles des droits étendus

Objectifs

- Lister les utilisateurs disposant des droits étendus spécifiques:
 - *User-Force-Change-Password* (type 00299570-246d-11d0-a768-00aa006e0529)
 - *Self-Membership* (type bf9679c0-0de6-11d0-a285-00aa003049e2)
 - ...

btaminer ListACE

```
$ btaminer -C::snktest -t ReST ListACE \  
    --type 00299570-246d-11d0-a768-00aa006e0529
```

Analysis by miner [ListACE]

=====

Trustee	Subjects	Object type
jean dupond	Administrateur	User-Force-Change-Password

Vérification des informations collectées

Échanges avec les équipes Active Directory

- Système Active Directory très vivant → modifications journalières
- Corrélation des éléments avec les administrateurs AD → explique les mauvaises pratiques

Différentiel entre deux instances d'un AD

Différentiel avec btadiff

- Permet de comparer un AD à deux moments dans le temps
- ⇒ Permet de surveiller un objet dans le temps
- ⇒ Permet de vérifier l'absence de modification suspecte

```
$ btadiff --CA ::clean --CB ::backdoor1 --ignore-defaults
```

```
=====
Starting diffing sd_table
-----
AB, 101: [] *sd_refcount['14'=>'15']
AB, 108: [] *sd_refcount['39'=>'41']
A , 229: []
A , 372: []
AB, 423: [] *sd_refcount['3'=>'2']
B, 424: []
B, 425: []
B, 428: []
-----
Table [sd_table]: 160 records checked, 2 disappeared, 3 appeared, 3 changed
=====
```

Différentiel entre deux instances d'un AD

La datatable

```
=====
Starting diffing datatable
-----
AB, 3586: [DC001] *logonCount['116'=>'117'], *lastLogon['datetime.datetime(20'=>'datetime.datetime(20']
AB, 3639: [RID Set] *rIDNextRID['1153'=>'1154']
AB, 8784: [A:[gc]/B:[gc DEL:346bf199-8567-4375-ac15-79ec4b42b270]]
        -showInAdvancedViewOnly, -objectCategory, +lastKnownParent, +isRecycled, +isDeleted,
        *name["u'gc'"=>"u'gc\\nDEL:346bf199-8"],
        *dc["u'gc'"=>"u'gc\\nDEL:346bf199-8"]
AB, 8785: [A:[DomainDnsZones]/B:[DomainDnsZones
        DEL:58b2962b-708c-4c93-99ff-0b7e163131f9]]
        -showInAdvancedViewOnly, -objectCategory, +lastKnownParent, +isRecycled, +isDeleted,
        *name["u'DomainDnsZones'"=>"u'DomainDnsZones\\nDE"],
        *dc["u'DomainDnsZones'"=>"u'DomainDnsZones\\nDE"]
AB, 8786: [A:[ForestDnsZones]/B:[ForestDnsZones
        DEL:87f7d8a2-4d05-48d0-8283-9ab084584470]]
        -showInAdvancedViewOnly, -objectCategory, +lastKnownParent, +isRecycled, +isDeleted,
        *name["u'ForestDnsZones'"=>"u'ForestDnsZones\\nDE"],
        *dc["u'ForestDnsZones'"=>"u'ForestDnsZones\\nDE"]
B, 8789: [snorky insomnihack]
B, 8790: [gc]
B, 8791: [DomainDnsZones]
B, 8792: [ForestDnsZones]
-----
Table [datatable]: 7636 records checked, 0 disappeared, 4 appeared, 5 changed
=====
```

Sommaire

- 1 Enjeux
- 2 BTA
 - Introduction
 - Import
 - Les miners
 - Le rapport
- 3 Méthodologie d'audit
- 4 Retour d'expérience**

Exigences matérielles

Machine d'analyse

- Xeon 3GHz 4 cœurs
- 12Go RAM
- disque SSD

Performances à l'import

- base NTDS de 8Go (=très gros)
- importée dans MongoDB: 26Go
- 8h30

Performances à l'analyse

- Temps généralement d'analyse négligeable

Problèmes rencontrés

L'import de la base NTDS.dit

- Mauvaise extraction du fichier NTDS.dit
- Méthode d'extraction non suivie par les administrateurs

Cohérence des objets en base

- Objets toujours référencés dans une ACE mais plus présents dans l'AD
- Migration d'un environnement en langue française vers l'anglais

Résultats d'audit

Constat sur la réalisation de l'audit

- Autonome dans la réalisation de l'audit, très peu d'interaction nécessaire au démarrage de l'audit
 - une fois le fichier NTDS.dit récupéré
- Environnement unique à chaque audit, paramétré avec des manières de travail propres à chaque entité
- Impossibilité de préjuger de la justesse de l'attribution de droits dans l'AD
 - ⇒ importance de vérifier les informations avec un administrateur

Résultats d'audit

Résultats communs entre les différents audits

- Souvent des mauvaises pratiques
 - Sur-utilisation de comptes d'administration génériques
 - Présence de nombreux comptes dont le mot de passe n'expire jamais
 - ...
- Absence d'homogénéité sur les templates de création d'utilisateurs
- Forte présence de comptes actifs mais ne s'étant jamais authentifiés sur l'AD

Conclusion

BTA

- donne en temps contraint des résultats déterministes
- aide à nettoyer les AD des mauvaises pratiques
- permet un audit récurrent
reproductibilité $\Rightarrow$ comparaison possible des résultats de deux audits

Prochains développements

- Accès LDAP
- Tests unitaires sur les *miners*
- Comparaison améliorée