

Preuve de chiffrement des flux de réplication Active Directory inter-sites

Analyse Wireshark — IPsec ESP / AES-256 — SRV-AD-01 ↔ SRV-AD-02

Objectif

Démontrer, capture Wireshark à l'appui, que les flux de réplication Active Directory entre [SRV-AD-01 \(10.11.3.18\)](#) (Site A) et [SRV-AD-02 \(10.11.4.18\)](#) (Site B) sont **totalelement chiffrés et illisibles** pour tout tiers qui intercepterait le trafic sur le réseau public. La preuve repose sur l'observation de paquets [ESP \(Encapsulating Security Payload\)](#) dont le contenu est mathématiquement opaque sans la clé de session IPsec.

Prérequis

- **Tunnel IPsec IKEv2 opérationnel** entre pfSense-01 et pfSense-02 (état « Established » visible dans [Status > IPsec](#))
- **SRV-AD-01** (10.11.3.18) et **SRV-AD-02** (10.11.4.18) — Réplication AD active (vérifiée via [repadmin /showrepl](#))
- **Wireshark** installé sur [CL-WIN11-01](#) (poste d'analyse)
- **Accès administrateur** à l'interface web pfSense-01 (<https://10.11.3.17>)
- Ouvrir [Services > Active Directory Sites and Services](#) sur SRV-AD-01 pour vérifier que le site « Site-B » existe bien

Rappel d'architecture IPsec

pfSense-01 WAN : [10.10.101.3](#) ↔ **pfSense-02 WAN** : [10.10.101.4](#)

Phase 1 IKEv2 : AES-256 / SHA-256 / DH Groupe 14 (2048 bits) — Phase 2 ESP : AES-256 / SHA-256 / PFS Groupe 14

Réseaux tunnelés : [10.11.3.16/28](#) ↔ [10.11.4.16/28](#)

Étape 1 — Initialisation du flux (Le « Trigger »)

Pour rendre le trafic AD visible dans Wireshark, il faut forcer des échanges réseau entre les deux sites. Un **ping continu** depuis SRV-AD-01 vers SRV-AD-02 génère du trafic ICMP encapsulé dans le tunnel IPsec — ce trafic sera capturé sous forme de paquets ESP opaques.

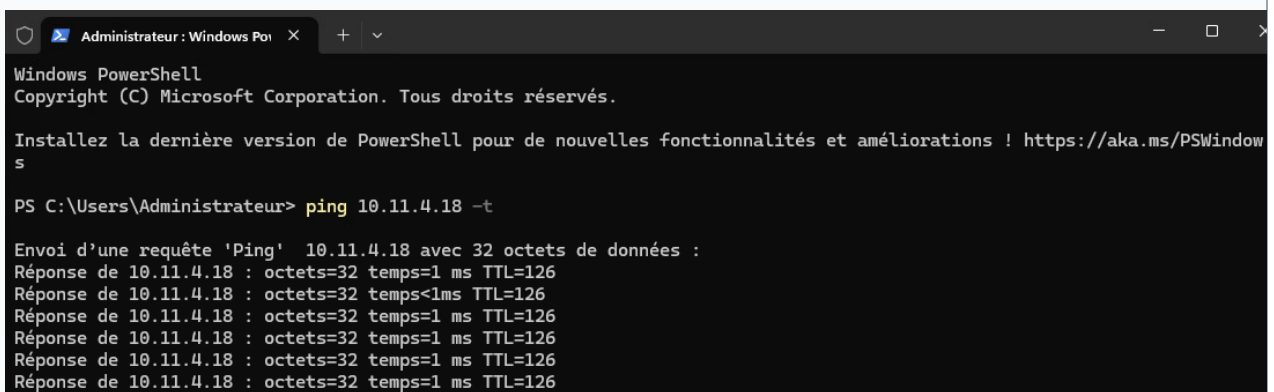
Action — Sur SRV-AD-01

Ouvrir une invite de commandes en tant qu'Administrateur et exécuter :

```
# Ping continu vers SRV-AD-02 (Site B) pour générer du trafic inter-sites
ping 10.11.4.18 -t
```

Le résultat attendu est une réponse `Reply from 10.11.4.18: bytes=32 time<1ms TTL=127` à chaque ligne. Ce retour prouve que le lien logique inter-sites fonctionne et que les paquets transitent bien par le tunnel.

 **CAPTURE D'ÉCRAN N°1** — Terminal SRV-AD-01 — ping 10.11.4.18 -t avec des réponses continues. La résolution et le TTL prouvent le transit via le tunnel IPsec.



```
Windows PowerShell
Copyright (C) Microsoft Corporation. Tous droits réservés.

Installez la dernière version de PowerShell pour de nouvelles fonctionnalités et améliorations ! https://aka.ms/PSWindows

PS C:\Users\Administrateur> ping 10.11.4.18 -t

Envoi d'une requête 'Ping' 10.11.4.18 avec 32 octets de données :
Réponse de 10.11.4.18 : octets=32 temps=1 ms TTL=126
Réponse de 10.11.4.18 : octets=32 temps<1ms TTL=126
Réponse de 10.11.4.18 : octets=32 temps=1 ms TTL=126
Réponse de 10.11.4.18 : octets=32 temps=1 ms TTL=126
Réponse de 10.11.4.18 : octets=32 temps=1 ms TTL=126
Réponse de 10.11.4.18 : octets=32 temps=1 ms TTL=126
```

Étape 2 — Capture de paquets sur pfSense-01 (Interface WAN)

On va intercepter les paquets au moment précis où ils quittent l'interface WAN de pfSense-01 vers Internet. À ce stade, si le chiffrement fonctionne, on ne doit voir **aucun protocole interne lisible** — uniquement des enveloppes ESP.

Action — Sur pfSense-01

Se connecter à l'interface web pfSense-01 : <https://10.11.3.17> puis naviguer dans **Diagnostics > Packet Capture**

Interface	WAN (interface publique, sortie vers Internet)
Protocol	ESP (Encapsulating Security Payload)
Host Address	10.10.101.4 (IP WAN de pfSense-02, Site B)
Count	0 (capture illimitée, on l'arrête manuellement)

Buffer Size

1 Mo (suffisant pour 10 secondes de trafic)

 CAPTURE D'ÉCRAN N°2 — Page Diagnostics > Packet Capture de pfSense-01 — configuration remplie (WAN, ESP, 10.10.101.4) avant de cliquer sur Start.

Packet Capture Output: /tmp/packetcapture-vmx1-20260506170358.pcap

```
15:03:59.391976 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1ee8), length 104
15:03:59.392620 IP 10.10.101.4 > 10.10.101.3: ESP(spi=0xcd261ac6,seq=0x1855), length 104
15:03:59.392700 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1ee9), length 104
15:03:59.392722 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1eea), length 136
15:03:59.393497 IP 10.10.101.4 > 10.10.101.3: ESP(spi=0xcd261ac6,seq=0x1856), length 120
15:03:59.393510 IP 10.10.101.4 > 10.10.101.3: ESP(spi=0xcd261ac6,seq=0x1857), length 104
15:03:59.393612 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1eeb), length 104
15:03:59.393641 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1eec), length 104
15:03:59.393868 IP 10.10.101.4 > 10.10.101.3: ESP(spi=0xcd261ac6,seq=0x1858), length 104
15:03:59.842443 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1eed), length 104
15:03:59.843047 IP 10.10.101.4 > 10.10.101.3: ESP(spi=0xcd261ac6,seq=0x1859), length 104
15:03:59.843297 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1eee), length 88
15:03:59.843549 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1eef), length 88
15:03:59.843566 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1ef0), length 120
15:03:59.843907 IP 10.10.101.4 > 10.10.101.3: ESP(spi=0xcd261ac6,seq=0x185a), length 88
15:04:00.391702 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1ef1), length 104
15:04:00.392742 IP 10.10.101.4 > 10.10.101.3: ESP(spi=0xcd261ac6,seq=0x185b), length 104
15:04:00.392864 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1ef2), length 104
15:04:00.392880 IP 10.10.101.3 > 10.10.101.4: ESP(spi=0xc6d2b689,seq=0x1ef3), length 152
15:04:00.393750 IP 10.10.101.4 > 10.10.101.3: ESP(spi=0xcd261ac6,seq=0x185c), length 120
```

Cliquer sur **Start**. Laisser la capture tourner **10 à 15 secondes** (le ping doit continuer sur SRV-AD-01 pendant ce temps). Cliquer sur **Stop**, puis sur **Download Capture** pour récupérer le fichier `.pcap` sur [SRV-AD-01](#).

Pourquoi filtrer sur ESP et non sur TCP/UDP ?

Le protocole ESP (IP Protocol 50) est le protocole de transport d'IPsec en mode tunnel. Tout paquet chiffré par IPsec arrive au niveau WAN sous cette forme. En filtrant sur ESP + IP WAN du Site B, on s'assure de ne capturer que le trafic IPsec sortant vers pfSense-02, sans bruit parasite.

Étape 3 — Analyse dans Wireshark : première preuve (colonne Protocol)

Ouvrir le fichier `.pcap` téléchargé avec Wireshark sur `CL-WIN11-01`. L'observation de la colonne **Protocol** constitue la première preuve formelle de chiffrement.

Observation attendue

- La colonne **Protocol** affiche **ESP** sur toutes les lignes — Wireshark est incapable d'identifier un protocole de couche applicative (pas de « ICMP », « RPC », « LDAP », « DNS »)
- La colonne **Info** affiche seulement `Encapsulated Security Payload`, sans aucun détail de contenu
- Les adresses `Source` et `Destination` affichent les IP WAN : `10.10.101.3` → `10.10.101.4` (et inversement pour les réponses)

CAPTURE D'ÉCRAN N°3 — Fenêtre principale de Wireshark — colonne Protocol remplie uniquement de lignes « ESP ». Aucun protocole applicatif visible. C'est la première preuve de chiffrement.

The screenshot displays the Wireshark network protocol analyzer interface. The main pane shows a list of captured packets. All packets in the list have a 'Protocol' column value of 'ESP' and an 'Info' column value of 'Encapsulated Security Payload'. The packet details pane on the right shows the structure of an ESP packet, including the SPI and sequence number. The packet bytes pane at the bottom shows the raw hex and ASCII data of the captured packet.

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)
2	0.000044	10.10.101.4	10.10.101.3	ESP	138	ESP (SPI=0xc6d2b689)
3	0.000724	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)
4	0.000746	10.10.101.3	10.10.101.4	ESP	170	ESP (SPI=0xc6d2b689)
5	0.001521	10.10.101.4	10.10.101.3	ESP	154	ESP (SPI=0xc6d2b689)
6	0.001534	10.10.101.4	10.10.101.3	ESP	138	ESP (SPI=0xc6d2b689)
7	0.001636	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)
8	0.001665	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)
9	0.001892	10.10.101.4	10.10.101.3	ESP	138	ESP (SPI=0xc6d2b689)
10	0.450467	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)
11	0.451071	10.10.101.4	10.10.101.3	ESP	138	ESP (SPI=0xc6d2b689)
12	0.451321	10.10.101.3	10.10.101.4	ESP	122	ESP (SPI=0xc6d2b689)
13	0.451573	10.10.101.3	10.10.101.4	ESP	122	ESP (SPI=0xc6d2b689)
14	0.451590	10.10.101.3	10.10.101.4	ESP	154	ESP (SPI=0xc6d2b689)
15	0.451931	10.10.101.4	10.10.101.3	ESP	122	ESP (SPI=0xc6d2b689)
16	0.999726	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)
17	1.000766	10.10.101.4	10.10.101.3	ESP	138	ESP (SPI=0xc6d2b689)
18	1.000888	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)
19	1.000904	10.10.101.3	10.10.101.4	ESP	186	ESP (SPI=0xc6d2b689)
20	1.001774	10.10.101.4	10.10.101.3	ESP	154	ESP (SPI=0xc6d2b689)
21	1.001792	10.10.101.4	10.10.101.3	ESP	138	ESP (SPI=0xc6d2b689)
22	1.001867	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)
23	1.001975	10.10.101.3	10.10.101.4	ESP	138	ESP (SPI=0xc6d2b689)

Packet details for the selected packet (No. 1):

- Frame 1: Packet, 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits)
- Ethernet II, Src: VMware_9e:09:94 (00:0c:29:9e:09:94), Dst: VMware_95:52:d6 (00:50:56:95:52:d6)
- Internet Protocol Version 4, Src: 10.10.101.3, Dst: 10.10.101.4
- Encapsulating Security Payload

Packet bytes (hex and ASCII):

```

0000  00 50 56 95 52 d6 00 0c 29 9e 09 94 00 00 45 00  PV-R... )...E-
0010  00 7c eb bd 00 00 40 32 b0 77 0a 0a 65 03 0a 0a  e.....).....
0020  65 04 c6 d2 b6 89 00 00 1e e8 3d f1 b5 90 98 86  e.....).....
0030  46 e9 d8 9f e6 cc 1d 7b 62 0a c6 71 0f c5 cd 4c  F.....( b.....L
0040  7d 49 77 24 c6 ef a1 69 1b e9 74 71 7e 46 42 ec  }lw.....i tq~P8
0050  92 8c e1 6f 30 40 06 37 01 d3 c7 1b 19 69 00 6c  ...o@7.....i l
0060  6d 71 e5 03 21 e6 6f 4c ab 51 11 e6 a8 66 8d 23  mq~l ol Q...f#
0070  cd 6e 5c fc a8 9c 55 7d 12 d0 80 93 c2 ea c3 1c  n\..U).....
0080  47 a8 bb 5e 9f 4f 99 e6 c9 a4  G..A.O...

```

Interprétation pour l'oral

Wireshark ne peut pas disséquer le contenu de ces paquets car il n'a pas accès à la clé de session IKEv2 négociée entre les deux pfSense. Sans cette clé, les octets du payload sont mathématiquement indistinguable d'une suite aléatoire.

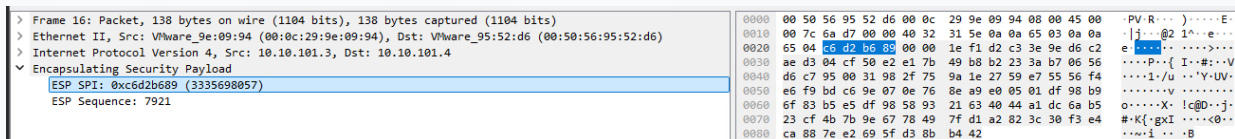
Étape 4 — La preuve par l'échec : analyse du payload (Data hexadécimal)

C'est l'étape qui apporte la preuve **irréfutable**. En cliquant sur un paquet ESP et en observant la zone **Data / Payload** dans le panneau inférieur de Wireshark, on visualise directement l'impossibilité de lire le contenu.

Procédure

1. Sélectionner un paquet ESP au hasard dans la liste
2. Dans le panneau du milieu (arbre de dissection), déplier la section **Encapsulating Security Payload**
3. Observer la ligne **Data:** — elle affiche une suite hexadécimale sans structure identifiable
4. Dans le panneau du bas (dump hexadécimal), aucune chaîne ASCII lisible n'apparaît — pas de **ICMP**, pas de **ping**, pas de **LDAP**, pas de **Kerberos**

 **CAPTURE D'ÉCRAN N°4 — Wireshark — zoom sur la zone Data/Payload d'un paquet ESP sélectionné. La suite hexadécimale opaque prouve que le contenu est chiffré AES-256. Aucun protocole interne identifiable.**



The screenshot shows the Wireshark interface. The packet list on the left shows 'Frame 16: Packet, 138 bytes on wire (1104 bits), 138 bytes captured (1104 bits)'. The packet details pane shows 'Ethernet II', 'Internet Protocol Version 4', and 'Encapsulating Security Payload'. The 'Encapsulating Security Payload' section is expanded, showing 'ESP SPI: 0xc6d2b689 (3335698057)' and 'ESP Sequence: 7921'. The packet bytes pane shows a hex dump of the payload, which is a random sequence of bytes, confirming it is encrypted.

Commentaire technique pour l'oral

"Ici, on voit que Wireshark est **incapable d'identifier le protocole** transporté à l'intérieur du paquet (ICMP, Kerberos, RPC de réplique AD...).

Les données sont **encapsulées et chiffrées en AES-256 bits**. Sans la clé de session IKEv2, qui n'existe que sur les deux pfSense, le contenu est **mathématiquement illisible**.

Un attaquant interceptant ce trafic sur le réseau public ne verrait que des blocs de données aléatoires — il ne pourrait même pas déterminer qu'il s'agit de trafic Active Directory. C'est exactement l'objectif de l'architecture IPsec."

Tests et Validation finale

Critères de succès — Checklist jury

- ✓ **Critère 1** : Le ping depuis SRV-AD-01 vers SRV-AD-02 répond avec succès (preuve de connectivité via tunnel)
- ✓ **Critère 2** : La capture pfSense sur l'interface WAN ne contient que des paquets ESP — aucun protocole interne en clair
- ✓ **Critère 3** : Wireshark affiche uniquement « ESP » en colonne Protocol — preuve que le chiffrement est actif
- ✓ **Critère 4** : Le payload hexadécimal est opaque — aucune donnée Active Directory lisible en clair sur le réseau public

Test complémentaire — Vérification de la réplication AD

Sur **SRV-AD-01**, vérifier que la réplication fonctionne correctement :

```
# Vérification complète de la réplication inter-sites
repadmin /showrepl

# Résultat attendu :
# DC=sio,DC=local
#   Default-First-Site-Name\SRV-AD-02 via RPC
#   GUID du domaine : <GUID>
#   La dernière tentative le <date> a réussi.
```

 **CAPTURE D'ÉCRAN N°5** — Résultat de repadmin /showrepl sur SRV-AD-01 — La dernière réplication avec SRV-AD-02 est marquée « a réussi ». Preuve que les données AD transitent correctement via le tunnel chiffré.

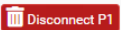
```
À partir de : CN=NTDS Settings,CN=SRV-AD-01,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=sio
C=local
Vers..... : CN=NTDS Settings,CN=SRV-AD-02,CN=Servers,CN=Default-First-Site-Name,CN=Sites,CN=Configuration,DC=sio
C=local
MESSAGE DE RAPPEL : Synchronisation totale terminée.
Synchronisation totale effectuée sans erreur.
```

Test complémentaire — Statut IPsec sur pfSense-01

Dans l'interface web pfSense-01, naviguer dans [Status](#) > [IPsec](#). Le tunnel doit afficher :

État	ESTABLISHED
Local	10.10.101.3 / 10.11.3.16/28
Remote	10.10.101.4 / 10.11.4.16/28
Algorithme	AES-CBC (256) / HMAC-SHA2-256 / MODP-2048
Bytes-In	Valeur > 0 (trafic reçu depuis Site B)
Bytes-Out	Valeur > 0 (trafic envoyé vers Site B)

 CAPTURE D'ÉCRAN N°6 — Interface Status > IPsec de pfSense-01 — Tunnel en état ESTABLISHED avec les compteurs Bytes-In et Bytes-Out non nuls, confirmant un trafic actif et chiffré.

IPsec Status							
ID	Description	Local	Remote	Role	Timers	Algo	Status
con1 #3	IPsec Bât A vers Bât B	ID: 10.10.101.3 Host: 10.10.101.3:500 SPI: a9c47006c816833c	ID: 10.10.101.4 Host: 10.10.101.4:500 SPI: 5575705c20947763	IKEv2 Initiator	Rekey: 10042s (02:47:22) Reauth: Disabled	AES_CBC (256) HMAC_SHA2_256_128 PRF_HMAC_SHA2_256 MODP_2048	Established 15106 seconds (04:11:46) ago 
ID	Description	Local	SPI(s)	Remote	Times	Algo	Stats
con1: #23	Multiple	10.11.3.0/28 10.11.3.16/28	Local: cd261ac6 Remote: c6d2b689	10.11.4.16/28	Rekey: 1633s (00:27:13) Life: 2145s (00:35:45) Install: 1455s (00:24:15)	AES_CBC (256) HMAC_SHA2_256_128 MODP_2048 IPComp: None	Bytes-In: 1,015,593 (992 KiB) Packets-In: 11,653 Bytes-Out: 2,026,552 (1.93 MiB) Packets-Out: 14,826 Installed 

Conclusion technique

La démonstration prouve, de manière empirique et non théorique, que l'architecture IPsec IKEv2 déployée sur la PME Sio-sisr garantit la **confidentialité totale** des flux de réplication Active Directory. Toute interception du trafic sur le réseau public entre les IPs WAN 10.10.101.3 et 10.10.101.4 ne révèle que des paquets ESP dont le contenu est **mathématiquement indéchiffrable** sans la clé de session — conformément aux spécifications RFC 4303 (ESP) et RFC 7296 (IKEv2).