Analyse de sécurité de NetBackup, logiciel de gestion de sauvegardes

Mouad Abouhali, Benoît Camredon, Nicolas Devillers,
Anaïs Gantet et Jean-Romain Garnier
prenom.nom(at)airbus.com

Airbus **

Résumé. NetBackup est le produit de gestion de sauvegardes le plus utilisé par les entreprises majeures. Des vulnérabilités avaient été identifiées par les auteurs et publiées précédemment, découlant de l'analyse des binaires les plus critiques du produit. Cependant, une vaste partie de la surface d'attaque exposée reste à découvrir.

Par ailleurs, la complexité du produit peut être un frein à de nouvelles analyses ou à l'estimation de la sécurité d'une infrastructure NetBackup déjà déployée. En effet, il emploie de multiples protocoles propriétaires, divers types de matériel cryptographique à protéger et plusieurs dizaines de binaires différents, reposant sur des technologies variées.

Cet article partage un résumé du fonctionnement interne des binaires étudiés tel que compris par les auteurs lors de leur analyse, accompagné d'une présentation d'outils de cartographie et reconnaissance avec pour objectif d'aider d'éventuels pentesteurs, chercheurs, architectes ou administrateurs réseau à étudier la sécurité de ce produit.

1 Introduction

1.1 Logiciels de gestion de sauvegardes : une cible intéressante

Que ce soit pour un souci de maintien de disponibilité d'informations critiques ou en prévision d'un besoin de récupération de données en cas de compromission d'un SI, l'utilisation de logiciels de gestion de sauvegardes facilite grandement le contrôle et le maintien de ces données et fait partie des dix règles d'or des recommandations de l'ANSSI [1]. Ces logiciels sont communément considérés comme cruciaux dans les « dernières lignes de défense ».

Étant donné la sensibilité des informations qu'un tel logiciel peut être amené à traiter et les privilèges que cela requiert sur les machines sauvegardées par ce biais, il est légitime de se demander quelle confiance accorder à ce type de logiciel.

** https://airbus-seclab.github.io

1.2 NetBackup, un produit phare

NetBackup se présente comme le premier produit de gestion de sauvegardes et de récupération des données dans le monde [10]. Il est par exemple classé leader dans la catégorie « *Enterprise Backup and Recovery Software Solutions* » en 2021 [9] par Gartner. Il serait également largement utilisé (87 % des entreprises du « Fortune Global 500 » [10]).

Selon la documentation officielle, c'est un produit qui se veut flexible, disponible pour sauvegarder une variété de plateformes et systèmes : Windows, Unix, Bases de données, Machines virtuelles, Cloud. Il se présente également comme « un produit de protection contre les rançongiciels de bout en bout » [10]. La probabilité de rencontrer NetBackup en mission d'audit d'un SI est donc élevée.

Il est à noter également que l'éditeur, Veritas, est le résultat de plusieurs rachats d'entreprises [3, 5]. De ce fait, NetBackup est l'agrégation de tout un ensemble de codes et technologies variés, ce qui apporte une probabilité importante de présence de bogues ou vulnérabilités.

Par ailleurs, il est possible de le configurer avec des fonctionnalités de sécurité à des degrés plus ou moins élevés, à la discrétion des administrateurs.

La nature de ce type de logiciel et la large utilisation de NetBackup ont motivé les auteurs à mener une analyse de sécurité de ce produit, d'une part pour comprendre quels sont les meilleurs usages et configuration à utiliser, d'autre part pour avoir une estimation de la confiance qu'on peut lui accorder.

1.3 Enjeux d'analyse de la sécurité de NetBackup

Des résultats d'étude de la sécurité de NetBackup avaient déjà été publiés auparavant sur une partie de la surface d'attaque exposée par NetBackup, laissant présager que d'autres résultats intéressants restaient encore à être découverts [6–8]. Les précédentes CVEs publiées sur NetBackup corroboraient cette idée [4].

Une première analyse par les auteurs a conduit à la découverte de nouvelles vulnérabilités, publiées selon un processus de divulgation coordonnée avec l'éditeur et ayant fait l'objet des bulletins de sécurité proposant notamment les correctifs associés [2]. Elle n'a cependant pas couvert l'intégralité de la surface d'attaque exposée par le produit NetBackup. Analyser le produit NetBackup est rendu difficile par les aspects suivants : NetBackup est un produit de plus de 100 binaires propriétaires. Par ailleurs, il existe peu de documentation technique sur le fonctionnement interne du produit,

sur le format des paquets applicatifs utilisés, l'architecture logicielle et le lien entre ces binaires. En outre, si les principes d'authentification et d'autorisation ainsi que leurs nombreuses options sont détaillés dans la documentation officielle du produit, il n'est pas évident de savoir quelles options utiliser et comment sont gérés les secrets liés à la mise en place de ces mesures de sécurité. Et enfin, bien que NetBackup embarque un grand nombre d'outils de débogage et d'administration du produit, il n'est pas trivial de savoir rapidement lequel utiliser pour effectuer une tâche donnée (par exemple : récupérer la sauvegarde d'un client depuis un serveur primaire).

Cet article se propose d'apporter quelques éléments de réponses à ces questions, sur la base des résultats de l'analyse réalisée par AirbusSeclab (environ 200 jours*personne) sur les versions 8.2 (publiée le 28/06/2019), 8.3 (publiée le 28/07/2020) et 9.0 (publiée le 04/01/2021), dans le but de partager un retour d'expérience sur l'approche utilisée pour analyser ce produit, à travers un regard offensif. Sa publication sera accompagnée de celle de plusieurs outils [1] développés par l'équipe visant à faciliter l'audit et l'analyse de NetBackup et de son infrastructure.

2 Cadre de l'étude

Au produit NetBackup sont associés un certain nombre de concepts clé qu'il est nécessaire de s'approprier avant d'entreprendre l'analyse à proprement parler. La première étape de l'analyse a donc été de s'informer sur la manière dont est architecturé le produit.

Cette partie résume dans un premier temps les composants clé de l'infrastructure NetBackup, puis effectue un premier bilan sur ce qui peut être déduit sur la sécurité de ces choix de conception, et enfin définit les questions de sécurité qui vont guider la suite de l'étude technique.

2.1 Notions clé de l'architecture NetBackup

La documentation officielle nous apprend que NetBackup s'architecture principalement autour des éléments suivants : un logiciel serveur pour gérer les sauvegardes, leur cycle, leur stockage et les hôtes ; et un logiciel client qui réside sur les hôtes disposant des données à sauvegarder. Ces fonctions sont réparties entre les différents composants présentés dans la figure 1 :

[1] `https://github.com/airbus-seclab/nbutools`

— **Les serveurs OpsCenter**, qui suivent les opérations de sauvegarde, génèrent des rapports et aident à gérer les serveurs primaires ;
— **Les serveurs primaires**, qui gèrent les sauvegardes, le stockage et les restaurations. Ils traitent la politique de sauvegarde mise en place et sont responsables de la sélection des serveurs médias et disques. Ils peuvent eux-mêmes disposer du rôle de serveur média ;
— **Les serveurs média**, qui exposent les périphériques de stockages qui leur sont attachés. Ils peuvent également augmenter les performances en distribuant la charge réseau ;
— **Les clients** qui contiennent les données à sauvegarder. Lors d'une sauvegarde, ils transmettent les données à sauvegarder à un périphérique de sauvegarde à travers un serveur média.

Une architecture classique de NetBackup est illustrée dans la figure 1.

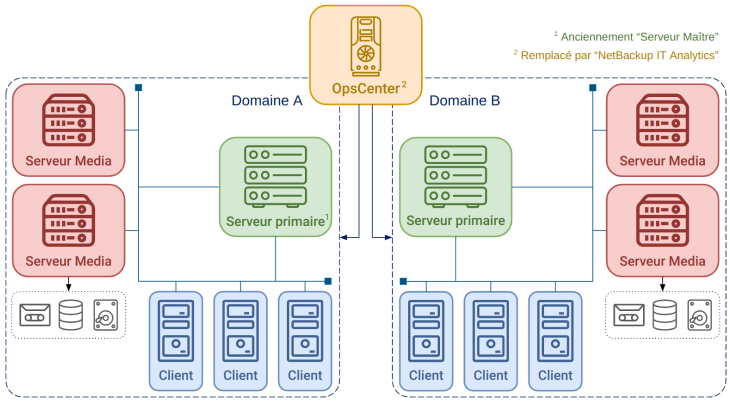

Fig. 1. Schéma d'une architecture NetBackup classique

Des concepts plus transverses sont également clés pour la compréhension de la suite de l'article : la notion de domaine NetBackup, de politique de sauvegarde et de catalogue.

— **Domaine NetBackup :** NetBackup définit un « domaine » comme un serveur primaire et l'ensemble des serveurs média et clients qui y sont reliés. Par définition, il n'y a qu'un seul serveur primaire par domaine, mais il peut y avoir plusieurs serveurs média. Un serveur OpsCenter permet d'administrer et monitorer plusieurs domaines (et de gérer le serveur primaire de chaque domaine, entre autres).
— **Politiques de sauvegarde :** Au sein d'un domaine, des « politiques de sauvegarde » permettent de définir, pour une liste de

clients, les sauvegardes à effectuer. Ainsi, chaque politique peut préciser des fichiers à inclure dans les sauvegardes, l'endroit où ces dernières doivent être stockées, la récurrence des sauvegardes automatiques, etc.

— **Catalogue :** Ces politiques sont enregistrées dans un « catalogue NetBackup » se trouvant sur le serveur primaire. Le catalogue, qui correspond à un ensemble de bases de données et de fichiers de configuration, est absolument crucial au bon fonctionnement de la solution et en cas de restauration de données. En effet, sans ce catalogue, il serait presque mission impossible de retrouver les données sauvegardées d'un client, ces dernières étant dispersées, compressées et dé-dupliquées dans un souci d'optimisation.

2.2 Premières constations de sécurité du produit NetBackup

Ces considérations sur l'architecture de NetBackup permettent déjà de noter des faiblesses inhérentes à la conception. Par exemple, du point de vue des flux réseau : Le serveur primaire doit pouvoir se connecter à tous les clients de son domaine. Le client doit pouvoir se connecter au serveur média sur lequel ses données sont stockées. Le client doit pouvoir également se connecter à son serveur primaire. Lorsque ce n'est pas directement possible, le serveur média peut servir de proxy vers le serveur primaire (par exemple si le client est dans une DMZ). Et enfin, tous les composants exposent des services accessibles depuis le réseau.

À retenir : Il est difficile de gérer la segmentation et le filtrage réseau pour durcir la sécurité de NetBackup et du SI. Même lorsqu'une segmentation est implémentée, NetBackup peut offrir des moyens de contourner cette segmentation.

De plus, les composants étant fortement interdépendants, il est intéressant de noter que le serveur OpsCenter a accès aux composants des domaines qu'il contrôle (dont notamment les serveurs primaires et clients), que le serveur primaire peut lire et écrire des fichiers arbitraires à des chemins arbitraires sur tous les clients de son domaine et que le serveur média peut supprimer ou corrompre les données qu'il gère. Enfin, dans certains cas, le serveur OpsCenter peut également être un client s'il est sauvegardé par NetBackup.

À retenir : Les serveurs OpsCenter et serveurs primaires sont des cibles de choix pour les attaquants, surtout depuis un client ou un accès réseau non-authentifié.

2.3 Questions de sécurité à résoudre

Au-delà des constatations précédentes, ces premières informations d'architecture ont fait surgir les questions suivantes :

- **Un composant de plus haut privilège peut-il être compromis depuis un composant de moindre privilège (par exemple client vers serveur primaire, ou serveur primaire vers serveur OpsCenter) ?**
- **Quelles données de l'infrastructure NetBackup un attaquant devrait-il cibler pour empêcher le recouvrement de sauvegardes ?**
- **Le produit NetBackup dans sa globalité peut-il être utilisé comme pivot pour attaquer les systèmes collatéraux présent sur le SI ?**
- **De quels moyens (outillage, connaissance du produit, etc.) un attaquant a-t-il besoin pour compromettre NetBackup ?**

Tenter de répondre à ces questions a constitué le fil rouge de l'étude menée par AirbusSeclab et le sera également pour la suite de cet article. Il est à noter que les travaux publiés se focalisent uniquement sur l'infrastructure NetBackup la plus classique, telle que représentée sur la figure 1, constituée des clients, serveurs média, serveurs primaires et serveurs OpsCenter.

Notamment, l'étude ne couvre pas les utilisations spécifiques de NetBackup, telles que les sauvegardes de bases de données ou de machines virtuelles.

3 Zoom sur le fonctionnement interne de NetBackup

Cette section expose quelques points d'intérêts notables du fonctionnement interne du produit tel que compris par les auteurs. L'approche utilisée, à la fois pour la méthodologie d'analyse du produit et pour la rédaction de ce chapitre, est de commencer par se poser une question fonctionnelle simple et d'étudier le produit et sa documentation jusqu'à être capable d'y répondre.

Ce chapitre cherche donc à répondre à la question suivante : « que se passe-t-il lorsqu'une sauvegarde est réalisée ? ». Ce fil conducteur nous mènera à présenter divers composants, leurs interactions et leurs rôles. Sans être exhaustif, l'objectif est de permettre au public visé par cet article de mieux appréhender NetBackup et ses mécanismes internes.

Pour simplifier, les explications supposent que les différents composants sont installés sur des machines Linux ; le comportement sous Windows est toutefois très souvent analogue. De plus, les informations exposées ne représentent que la compréhension des auteurs, qui ne peut qu'être partielle, et peuvent donc diverger de la réalité.

3.1 Choix des binaires à analyser

Le produit NetBackup contient plus d'une soixantaine d'exécutables différents, en plus des bibliothèques propriétaires dont ils dépendent. Dans le temps imparti à l'analyse de sécurité, il a donc fallu faire le choix d'un sous-ensemble de binaires à analyser en priorité.

3.1.1 Approche 1 : Repérage des services disponibles Lister les services en écoute sur les différents composants permet d'avoir une première idée de la surface exposée à étudier. En effet, certains binaires sont exposés sur le réseau et leur nombre varie suivant les composants : certains sont toujours présents quelle que soit la configuration tandis que d'autres sont optionnels et dépendent de la configuration ou sont spécifiques à un composant. Les tableaux 1, 2, 3 et 4 recensent les services en écoute sur l'interface externe de chacune des entités dans leur configuration la plus classique.

`Local:Port` (en écoute)	`Distant:Port`	Processus	Utilisateur
`0.0.0.0:443`	`0.0.0.0:*`	`vnetd`	root
`0.0.0.0:1556`	`0.0.0.0:*`	`pbx_exchange`	root
`0.0.0.0:3652`	`0.0.0.0:*`	`java`	root
`0.0.0.0:8205`	`0.0.0.0:*`	`java`	root
`0.0.0.0:8443`	`0.0.0.0:*`	`java`	root
`0.0.0.0:13701`	`0.0.0.0:*`	`vmd`	root
`0.0.0.0:13720`	`0.0.0.0:*`	`bprd`	root
`0.0.0.0:13721`	`0.0.0.0:*`	`bpdbm`	root
`0.0.0.0:13723`	`0.0.0.0:*`	`bpjobd`	root
`0.0.0.0:13724`	`0.0.0.0:*`	`vnetd`	root
`0.0.0.0:13782`	`0.0.0.0:*`	`bpcd`	root
`0.0.0.0:13783`	`0.0.0.0:*`	`nbatd`	root
`0.0.0.0:13785`	`0.0.0.0:*`	`NB_dbsrv`	root
`0.0.0.0:34547`	`0.0.0.0:*`	`java`	root

Tableau 1. Ports NetBackup en écoute sur l'interface externe du serveur primaire

Local:Port (en écoute)	Distant:Port	Processus	Utilisateur
0.0.0.0:1556	0.0.0.0:*	pbx_exchange	root
0.0.0.0:13701	0.0.0.0:*	vmd	root
0.0.0.0:13723	0.0.0.0:*	bpjobd	root
0.0.0.0:13724	0.0.0.0:*	vnetd	root
0.0.0.0:13782	0.0.0.0:*	bpcd	root

Tableau 2. Ports NetBackup en écoute sur l'interface externe du serveur média

Local:Port (en écoute)	Distant:Port	Processus	Utilisateur
0.0.0.0:1556	0.0.0.0:*	pbx_exchange	root
0.0.0.0:13724	0.0.0.0:*	vnetd	root
0.0.0.0:13782	0.0.0.0:*	bpcd	root

Tableau 3. Ports NetBackup en écoute sur l'interface externe du client

Local:Port (en écoute)	Distant:Port	Processus	Utilisateur
0.0.0.0:443	0.0.0.0:*	java	root
0.0.0.0:1556	0.0.0.0:*	pbx_exchange	root

Tableau 4. Ports NetBackup en écoute sur l'interface externe du serveur OpsCenter

À noter : La totalité des binaires sont lancés par l'utilisateur `root` sous Linux ou `NT Authority\System` sous Windows.[2]

On peut remarquer que tous les composants de l'architecture NetBackup exécutent au moins le binaire `pbx_exchange` (pour « Private Branch Exchange »), ce qui constitue un bon point d'entrée d'analyse. En plus de `pbx_exchange`, pour des raisons historiques, de nombreux autres binaires acceptent des connexions entrantes directement. La liste de ces binaires et de ceux accessibles via `pbx_exchange` dépend à la fois du type de composant et de la configuration de NetBackup. La surface d'attaque est donc changeante, mais nos expériences montrent qu'elle est toujours élevée. Une approche plus fonctionnelle a été utile pour guider les auteurs dans le choix des autres binaires critiques à étudier.

[2] Depuis la version 9.1 de NetBackup (publiée mi-2021), la plupart des binaires du serveur primaire peuvent être lancés avec un utilisateur avec des privilèges restreints. Ce n'est toutefois pas la configuration par défaut et ce n'est pas le cas pour les autres composants.

3.1.2 Approche 2 : Fonctionnement nominal de la solution Il peut ne pas être facile de répondre à la simple question « que se passe-t-il lorsqu'une sauvegarde est réalisée ? ». La figure 2 illustre le processus standard pour une sauvegarde programmée dans NetBackup : Soit une sauvegarde programmée pour le client C. Le serveur primaire P disposant de l'ensemble des informations de gestion de C, il sait quel serveur média M contacter. P envoie alors une demande de sauvegarde à M, qui envoie lui-même une demande de sauvegarde à C. Le client prépare les données à sauvegarder puis les envoie à M. Il notifie également P que la sauvegarde a été effectuée (métadonnées). Pas moins de 9 binaires de NetBackup entrent en jeu pour effectuer ce processus.

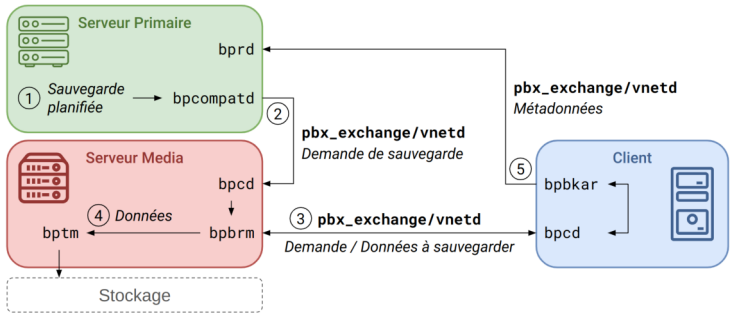

Fig. 2. Schéma d'un processus de sauvegarde NetBackup classique

Cette approche a permis de sélectionner les binaires clés suivants : `bpcd`, `bprd`, `pbx_exchange`, son alternative historique `vnetd`, ainsi que `nbatd`.

3.2 Dans les méandres de la gestion d'une sauvegarde par NetBackup

Dans cette partie, nous nous proposons d'aborder quelques composants plus en détails, tout en gardant en tête la question servant de fil conducteur de ce chapitre. L'objectif est de retracer, pas à pas, le chemin nous ayant permis de répondre à cette question en détails.

En effet, nombre de composants de NetBackup implémentent un protocole propriétaire propre à chacun, ce qui nécessite une implémentation particulière pour développer des outils permettant d'interagir avec eux. Les auteurs donnent ici quelques éléments importants de leur compréhension

du fonctionnement du produit, ainsi que quelques constatations, conseils et outils permettant de mieux appréhender et étudier NetBackup.

3.2.1 Fonctionnement de `pbx_exchange` (« Private Branch Exchange ») `pbx_exchange` est le point d'entrée privilégié pour les services NetBackup, et donc le premier à étudier pour comprendre comment l'ordre de réaliser une sauvegarde est transmis. Comme mentionné précédemment, le binaire `pbx_exchange` est exécuté sur tous les composants de l'infrastructure NetBackup et écoute sur le port TCP `1556`.

À noter que de nombreux services NetBackup écoutent également sur un port dédié pour des raisons de rétrocompatibilité.

Lors du lancement de `pbx_exchange`, se déroule une phase d'« enregistrement » des autres binaires. Cet enregistrement s'effectue dynamiquement via un port écoutant sur l'interface locale. Pour ce faire, un protocole propriétaire est utilisé, dont le format est décrit dans le tableau 5 et un exemple est présenté dans la figure 3.

Entête						Données
`0x00`	`0x01`	`0x03`	`0x04`	`0x08`	`0x09`	`0x10`
version	type	statut client	taille	erreur	aléatoire	données
`[0x0;0x4]`	`enum`	`enum`	`len(data)`	`bool`	-	-

Tableau 5. Format d'une requête `pbx_exchange`

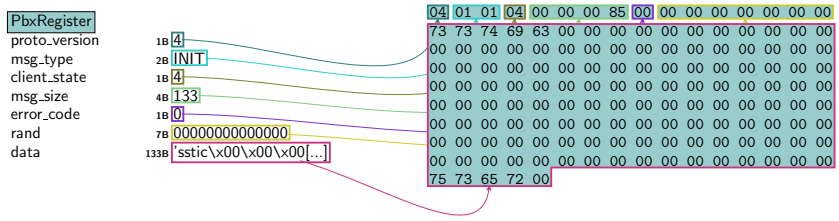

Fig. 3. Exemple de dissection Scapy d'un paquet d'enregistrement du service `sstic` par l'utilisateur `user` auprès de `pbx_exchange`

Lors de l'enregistrement auprès de `pbx_exchange`, chaque binaire doit préciser deux paramètres : le nom du service qu'il veut enregistrer (un binaire peut enregistrer plusieurs services) et l'utilisateur (au sens UNIX) exécutant le binaire. `pbx_exchange` vérifie cette information en s'assurant que le binaire peut accéder en écriture à une socket UNIX créée pour

l'occasion par `pbx_exchange` dans un dossier accessible uniquement à l'utilisateur (par exemple, le dossier `/var/VRTSpbx/user` créé avec les permissions `700` pour l'utilisateur `user`).

À l'issue de l'enregistrement, le service devient accessible via le port `1556`. Pour s'y connecter, il suffit d'envoyer un message comme celui du listing 1.

Listing 1: Connexion à un service « sstic » via `pbx_exchange`

```
1 $ echo -ne "ack=1\nextension=sstic\n\n" | nc <adresse> 1556
```

Des exemples de services ainsi accessibles via `pbx_exchange` sont les suivants :

- `TLSPROXY` et `HTTP TUNNEL`, pensés pour permettre à un client dans une DMZ de se connecter à un serveur primaire par l'intermédiaire d'un serveur média (permettant donc d'outrepasser les ségrégations réseau potentiellement mises en place) ;
- `vnetd`, un ancêtre de `pbx_exchange` décrit dans la section 3.2.3 ;
- `bpcd`, service présent sur tous les clients, serveurs média et serveurs primaires présenté dans la section 3.2.4.

Un serveur primaire enregistre généralement plus d'une trentaine de services via `pbx_exchange`, contre une quinzaine pour un serveur média, et une dizaine pour un client et un serveur OpsCenter.

Ainsi, la compréhension du fonctionnement de `pbx_exchange` est essentielle pour analyser les flux entre composants de NetBackup. De plus, son fonctionnement illustre la difficulté de filtrer ces flux, tout le trafic étant « masqué » derrière un port unique.

Au cours de cette analyse, les auteurs ont découvert plusieurs vulnérabilités exploitables localement avec un impact modéré à élevé.[3] De plus, il a été démontré que détourner le principe de `pbx_exchange` pour installer une « porte dérobée » peut efficacement masquer le trafic et passer outre les règles de filtrage existantes.

À retenir : `pbx_exchange` sert de porte d'entrée unique pour les services NetBackup, ce qui peut rendre difficile l'identification et le filtrage des flux.

3.2.2 `pbx_exchange` et le partage de sockets Un détail important à étudier pour dérouler le fil conducteur est la méthode utilisée par

[3] CVE-2022-42306, CVE-2022-42308

`pbx_exchange` pour passer une communication entrante (d'un client au sens large) au service ciblé. Pour cela, il utilise un mécanisme propre à Linux, permettant à deux processus de s'échanger un descripteur de fichier (ici une socket). Ainsi, le client qui communique initialement avec le processus `pbx_exchange` voit sa communication transférée à un autre processus, de façon totalement transparente.

Ici, le transfert du descripteur de fichier se fait par l'intermédiaire de la socket UNIX créée par `pbx_exchange` pour le client et de l'option `SCM_RIGHTS`. Ce mécanisme est résumé avec l'exemple du processus `bpcd` dans la figure 4. Il est à noter que `pbx_exchange` n'est pas le seul processus à utiliser ce mécanisme de transfert de socket au sein de NetBackup.

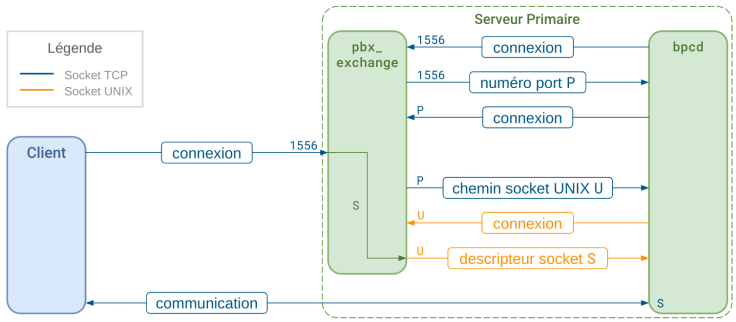

Fig. 4. Illustration du mécanisme d'enregistrement et de connexion de `pbx_exchange` (bleu : cf. 3.2.1, orange : cf. 3.2.2)

En termes de moyens d'analyse, ce fonctionnement peut rendre plus difficile l'interception des communications et l'analyse des interactions entre services NetBackup. Il peut pourtant être nécessaire de placer un « proxy » entre `pbx_exchange` et les services qui veulent s'y enregistrer. Par exemple, pour effectuer une interception du trafic en direction de `bpcd`, le mécanisme suivant a été mis en place par les auteurs (résumé en figure 5) : premièrement, la variable d'environnement `LD_PRELOAD` est utilisée pour forcer `bpcd` à se connecter sur un proxy (et non sur `pbx_exchange` directement). Ensuite, ce proxy change le chemin de la socket UNIX échangée, afin d'être capable de modifier la communication sur cette socket. Enfin, le descripteur de socket échangé via cette socket UNIX est intercepté et utilisé par le proxy qui renverra à `bpcd` un autre descripteur de socket qu'il contrôle.

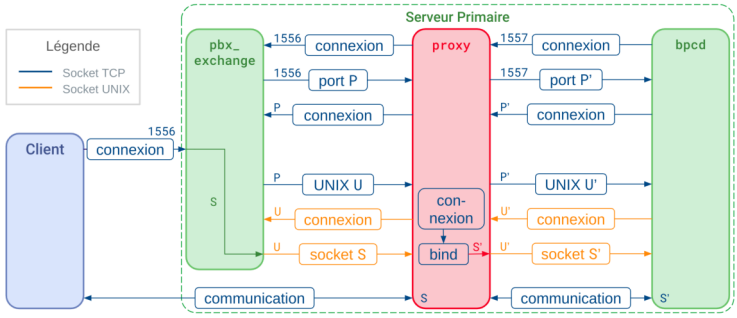

Fig. 5. Illustration du mécanisme d'**interception** des communications de `pbx_exchange`

3.2.3 Fonctionnement de `vnetd` (« Network Communication Service »)

Avant d'évoquer certains des services principaux accessibles via `pbx_exchange`, il convient de se pencher rapidement sur son ancêtre, `vnetd`. Ce dernier a un rôle très similaire à `pbx_exchange`, et reste utilisé pour des raisons de rétrocompatibilité avec d'anciennes versions de NetBackup. En plus d'être accessible via `pbx_exchange`, il écoute sur le port TCP 13724.

Contrairement à `pbx_exchange`, les services accessibles via `vnetd` ne sont pas définis dynamiquement, mais sont représentés par un fichier texte dans le dossier `/usr/openv/var/vnetd`. Ce fichier contient le nom du service ainsi que la commande à exécuter pour les démarrer (voir le listing 2).

Listing 2: Exemple de contenu d'un fichier de service `vnetd`

```
$ cat /usr/openv/var/vnetd/inetd_bpcd.txt
NAME=bpcd;PID=0;REGTIME=0;EXPTIME=0;ICMD=bpcd;
```

Il convient de noter qu'un accès en écriture à ce dossier confère automatiquement la possibilité d'exécuter du code arbitraire en root.

De plus, `vnetd` implémente des fonctionnalités additionnelles permettant, entre autres, d'obtenir des informations sur la machine exécutant ce service. Utiles à des fins de reconnaissance et d'identification de la cible, les informations incluent notamment : la version du service, le nom du serveur primaire le cas échéant, des informations sur les options de sécurité activées dans la configuration NetBackup de la machine cible. Pour ce faire, `vnetd` utilise un protocole propriétaire simple et purement textuel, dont un exemple d'échange est présenté dans le listing 3. Cette propriété

a été utilisée par les auteurs pour le développement d'un outil de scan de NetBackup, décrit dans la section 4.2.

Listing 3: Exemple d'échanges de paquets entre un client et le service `vnetd`

```
# "Handshake" de négociation de version
> 3400
< 3400
> 3400

# Envoie de la commande "VN_VERSION_GET"
> 3800
< 38323030303000  # Chaîne de caractères "820000" (version 8.2)
< 3000
```

3.2.4 Fonctionnement de `bpcd` (« BackuP Client Daemon »)

Comme présenté dans la figure 2, `bpcd` est l'un des premiers services intervenant dans le processus de sauvegarde. Il est accessible sur les clients, serveurs média et serveurs primaires via `pbx_exchange` ou directement sur le port `13782`. Il a le rôle important de recevoir des commandes pour, entre autres, démarrer une sauvegarde sur un serveur média ou un client. Dans ce sens, il sert de « point central » qui traite ou retransmet les commandes vers d'autres service NetBackup.

`bpcd` implémente de nombreuses fonctionnalités qui en font une cible de choix, dont notamment la lecture, l'écriture et la suppression arbitraire de fichiers, l'exécution arbitraire de commandes et l'obtention d'informations sur la machine et les utilisateurs.

Ainsi, ce service a fait l'objet de plusieurs travaux [6–8] qui ont mis en évidence de nombreuses vulnérabilités critiques. Ces dernières ont mené à de nombreux changements, dont une refonte du système d'authentification de NetBackup. Ces mécanismes n'étant pas propres à `bpcd` et étant intégrés dans un composant tiers, ils seront présentés en détail ultérieurement (cf. les sections 3.2.6 et 3.2.7).

Une fois la communication avec `bpcd` établie par l'un des services NetBackup (ou par un attaquant), un protocole propriétaire est utilisé pour établir la liaison et permettre l'exécution des commandes `bpcd`. Lorsque les communications ne sont pas chiffrées (par exemple pour les versions de NetBackup antérieures à 8.1), le format est majoritairement textuel. Au contraire, lorsque les communications sont chiffrées, le format utilisé est celui décrit dans le tableau 6, dont un exemple est présenté dans la figure 6.

Entête TLS			Données
0x00	0x03	0x04	0x08
magic TLS	statut	taille	données
0x5ec100	enum	len(data)	-

Tableau 6. Format d'un paquet `bpcd` pour les communications sécurisées en TLS

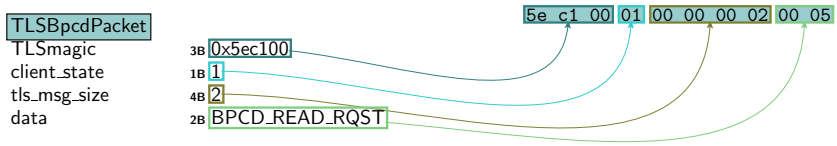

Fig. 6. Exemple de dissection Scapy d'un paquet `bpcd` avec communication sécurisée

La multitude de commandes implémentées par `bpcd` augmente considérablement sa surface d'attaque. De plus, ces commandes font généralement appel à des commandes systèmes ou des binaires NetBackup pour exécuter leur tâche. Ainsi, malgré la refonte du système d'authentification, il nous a été possible de démontrer l'existence d'une vulnérabilité permettant l'élévation de privilèges sous Windows.[4]

3.2.5 Fonctionnement de `bprd` (« BackuP Request Daemon »)

`bprd` a un fonctionnement analogue à `bpcd`, mais s'exécute sur le serveur primaire. Ainsi, il est chargé de recevoir et exécuter des commandes spécifiques venant d'autres machines de l'infrastructure NetBackup. Dans le contexte de notre fil conducteur, il reçoit des méta-données du client concernant la sauvegarde.

Tout comme `bpcd`, `bprd` implémente de nombreuses fonctionnalités intéressantes pour un attaquant, dont notamment :

- la lecture et l'écriture de fichiers sur le serveur primaire et sur les clients de son domaine ;
- le démarrage d'une sauvegarde sur un client ;
- le démarrage de la restauration d'une sauvegarde d'un client.

Les mécanismes d'authentification employés sont identiques à ceux évoqués dans la section 3.2.4 et sont décrits dans les sections 3.2.6 et 3.2.7.

[4] CVE-2022-36985

Le protocole propriétaire utilisé par `bprd` est majoritairement textuel mais varie légèrement en fonction de la version de NetBackup. Le tableau 7 et la figure 7 présentent le format utilisé lorsque les communications ne sont pas chiffrées (par exemple pour les versions de NetBackup antérieures à 8.1), tandis que le tableau 8 et la figure 8 sont leur pendant lorsque les communications sont chiffrées.

Entête		Commande	
`0x00`	`0x04`	`0x0B`	-
taille	magic	commande	paramètres
`len(data)`	`"329199 "`	`"[0;283] "`	paramètres séparés par des espaces

Tableau 7. Format d'un paquet `bprd` pour les communications non sécurisées

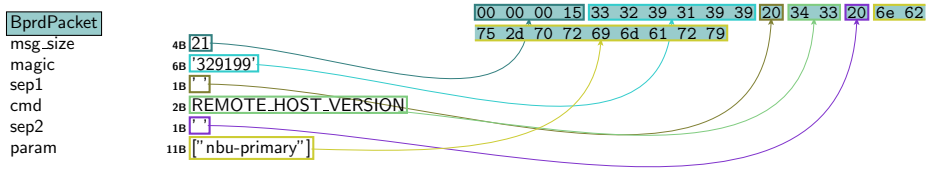

Fig. 7. Exemple de dissection Scapy d'un paquet `bprd`

Entête TLS			Entête textuel		Données	
`0x00`	`0x03`	`0x04`	`0x08`	`0x0C`	`0x13`	-
magic TLS	statut	taille TLS	taille	magic texte	commande	param.
`0x5ec100`	enum	`len(pkt)`	`len(data)`	`"329199 "`	`"[0;283] "`	param.

Tableau 8. Format d'un paquet `bprd` pour les communications sécurisées en TLS

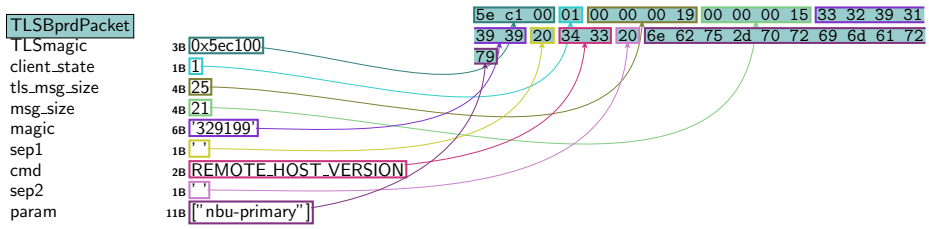

Fig. 8. Exemple de dissection Scapy d'un paquet `bprd` (communication TLS)

Chacune des commandes (près de 300 commandes existent) est identifiée par un entier (dans le champ *commande*). Si certaines sont accessibles sans authentification, la majorité requiert d'être reconnue en tant que client NetBackup authentifié.

Bien que similaire, ce service a fait l'objet de moins d'études détaillées que son homologue `bpcd`. Malgré tout, comme expliqué dans la section 2.2, la compromission de `bprd` entraîne la compromission du serveur primaire, et donc des conséquences importantes. Ce binaire mériterait d'être étudié plus en détail.

En outre, notre analyse de ce composant a montré l'existence de nombreuses vulnérabilités, permettant notamment à un client de lire et écrire des fichiers ainsi que d'exécuter des commandes arbitraires sur un serveur primaire.[5]

3.2.6 Fonctionnement de `nbatd` (« NetBackup AuthenTication Daemon ») Le service `nbatd` est en charge de l'authentification NetBackup, dont les mécanismes seront détaillés plus avant dans la section 3.2.7. Il communique en TLS, mais ne vérifie pas la validité des certificats qui lui sont transmis. Il utilise lui aussi un protocole de communication qui lui est propre, décrit dans le tableau 9, et dont un exemple de paquet est présenté dans la figure 9. Les champs de l'entête sont communs à tous les paquets de commande `nbatd`.

Entête					Données
`0x00`	`0x04`	`0x08`	`0x0C`	`0x10`	`0x11`
version	magic	commande	taille	unicode	données
`0x00000001`	`0xBAADF00D`	`[0x0-0x4B]`	`len(data)`	`bool`	-

Tableau 9. Format générique des paquets `nbatd`

Le champ `msg_type` encode un numéro de commande `nbatd`. Lors de l'étude, il existait plus d'une soixantaine de commandes, chacune ayant un format de données différent (texte, xml, binaire, etc.). Par exemple, la commande `PK_AUTH_DATA (0x05)` permet de s'authentifier et attend une demande d'authentification au format xml, alors que la commande `PK_AUTH_TYPE (0x06)` permet de choisir le mode d'authentification souhaitée et attend une chaîne de caractères (cf. listing 4).

[5] CVE-2022-36987, CVE-2022-36989, CVE-2022-36991, CVE-2022-36992, CVE-2022-36993

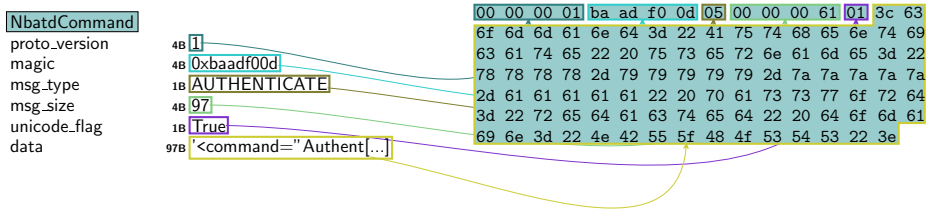

Fig. 9. Exemple de dissection Scapy d'un paquet `nbatd`

Listing 4: Format des données commande `PK_AUTH_TYPE (0x06)`

```
# Exemples de requêtes (simples chaînes de caractères)
"vx"
"pam"
"unixpwd"
# Exemples de réponses (format binaire)
00 00 00 01 ----> valid method requested.
00 00 00 09 ----> invalid method requested.
```

À noter : Bien que notre étude n'ait pas mené à la découverte de failles critiques (hormis des plantages dans le décodage d'entrées nulles), les mises à jour de `nbatd` sont à surveiller car il constitue une surface d'attaque exposée et très intéressante sur les serveurs primaires et serveurs OpsCenter.

3.2.7 Authentification Pour finir de comprendre le mécanisme de réalisation d'une sauvegarde, ainsi que mieux appréhender l'exposition des différents composants de NetBackup, il convient d'étudier les différents mécanismes d'authentification limitant l'accès aux services. Il en existe plusieurs en fonction de la version de NetBackup et de sa configuration :

- pour les versions antérieures à 8.1, une « identification » s'appuyant sur l'entrée DNS associée à l'adresse IP de la machine ;
- pour les versions plus récentes, une authentification s'appuyant sur des certificats X.509 (voir le paragraphe 3.2.7.2) ;
- lorsque le contrôle d'accès « NBAC » (NetBackup Access Control) est activé, une seconde couche d'authentification s'appuyant sur des certificats X.509 distincts (voir le paragraphe 3.2.7.4).

3.2.7.1 Gestion des certificats NetBackup utilise des certificats X.509 pour authentifier les différents composants. Pour ce faire, le serveur primaire joue le rôle d'autorité de certification et délivre des certificats

aux différents composants de son domaine. Il en existe deux types : les certificats s'appuyant sur le nom de domaine de la machine et ceux s'appuyant sur un identifiant unique propre à NetBackup. Ces premiers sont dépréciés, mais demeurent utilisés pour certaines fonctionnalités comme NBAC (voir 3.2.7.4). Il est également possible de configurer une autorité de certification externe pour signer ces certificats.

3.2.7.2 Secure Communication Lorsque des composants de NetBackup avec une version 8.1 ou ultérieure communiquent, ils utilisent obligatoirement le mode « Secure Communication », dans lequel les certificats mentionnés ci-dessus sont utilisés pour établir un canal TLS avec authentification mutuelle (à l'exception du serveur OpsCenter qui ne supporte pas cette fonctionnalité). Toutefois, pour rester compatible avec d'anciennes versions de NetBackup ou avec le serveur OpsCenter, l'option `allowInsecureBackLevelHost` (activée par défaut) autorise NetBackup à rétrograder les communications en clair si besoin.

3.2.7.3 Déploiement des certificats Lors de l'ajout d'un nouveau composant à un domaine NetBackup, celui-ci doit enregistrer l'autorité de certification du serveur primaire comme digne de confiance (i.e. ajouter son certificat dans son « truststore »). Il doit ensuite obtenir un certificat signé par cette autorité. Pour cela, il existe trois niveaux de sécurité :

- `moyen`, pour lequel le certificat est automatiquement fourni si le serveur primaire peut bien faire la correspondance entre l'adresse IP connectée et le nom du client via une requête DNS ;
- `élevé`, pour lequel le certificat est automatiquement renouvelé pour les machines connues uniquement (et doit être réalisé manuellement pour les autres) ;
- `très élevé`, pour lequel un jeton doit être généré sur le serveur primaire et fourni pour chaque nouvelle demande de certificat.

3.2.7.4 NetBackup Access Control (NBAC) NetBackup implémente également un contrôle d'accès plus fin. Ce mécanisme repose sur des agents gérant l'authentification et les autorisations de chaque composant de l'infrastructure. En pratique, ces agents s'exécutent sur le serveur primaire, les serveurs média et le serveur OpsCenter et s'appuient sur le protocole propriétaire « VxSS » utilisant des certificats X.509 et une authentification TLS. Encore une fois, le serveur primaire joue le rôle d'autorité de certification (différente de celle du mode « Secure Communication »). À noter que NBAC doit être configuré manuellement (via l'option `USE_VXSS` dans le fichier de configuration de chaque composant). Il peut être soit

négocié lors de l'établissement de la communication (valeur `AUTOMATIC`), soit obligatoire (valeur `REQUIRED`), soit désactivé (valeur `PROHIBITED`). De plus, cette option peut être configurée par « réseau » (i.e. machine, plage d'adresses IP ou suffix DNS).

Note : NBAC n'est pas supporté par les appliances NetBackup.

À noter : Dans la version étudiée, un grand nombre de vulnérabilités critiques ont pu être exploitées seulement lorsque l'option `VxSS` est activée. Si cette fonctionnalité est intéressante au niveau de la finesse du contrôle d'accès aux différentes entités de NetBackup, elle est donc à utiliser avec précaution.

3.2.7.5 Modèle de confiance d'authentification Le modèle de confiance pour l'authentification s'articule autour du serveur primaire. Lorsque « Secure Communication » est activé, chaque client et serveur média doit faire confiance à l'autorité de certification associée sur le serveur primaire et obtenir un certificat. Toutefois, si d'anciennes versions de NetBackup sont déployées dans le domaine ou si un serveur OpsCenter est utilisé, il faudra autoriser les communications à être rétrogradées en clair. Pour NBAC, il faut renouveler l'opération de déploiement pour chaque client et serveur média (et serveur OpsCenter le cas échéant) avec l'autorité de certification dédiée sur le serveur primaire. De plus, le serveur primaire doit faire confiance à l'autorité de certification du serveur OpsCenter pour des raisons opérationnelles. Le schéma 10 résume ces différents liens de confiance.

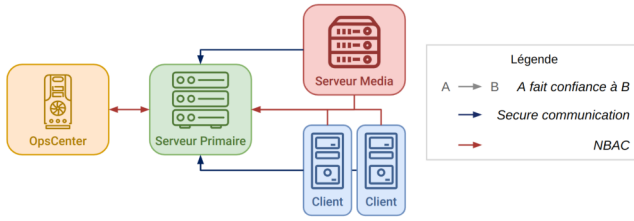

Fig. 10. Modèle de confiance des autorités de certification NetBackup

3.2.7.6 Modèle de confiance d'autorisation Le modèle de confiance pour l'autorisation diverge de celui pour l'authentification. En effet, celui-ci correspond plutôt à un système de « filtre » propre à chaque service (et éventuellement chacune de leurs commandes) prenant en compte la

nature du composant se connectant, le mode d'authentification utilisé et le fait qu'il soit connu ou non. Par exemple, le serveur primaire d'un domaine se connectant à l'un de ses clients pourra réaliser de nombreuses actions car ce dernier lui attribue une forme de confiance. Une sorte de « hiérarchie » entre composants peut alors se dégager, que nous avons traduite à haut niveau par un modèle de confiance entre les composants. Celui-ci est représenté dans la figure 11. À noter que l'autorisation s'appuie, en fonction des services, sur les enregistrements dans la base de données et/ou sur le système de fichier (e.g. `bprd` cherche un fichier correspondant au nom de domaine ou à l'adresse IP dans le dossier `/usr/openv/var/bprd/remote_ops`).

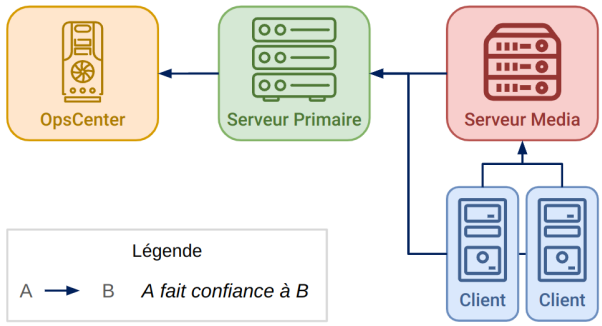

Fig. 11. Modèle de confiance entre composants de l'infrastructure NetBackup

3.2.7.7 Les certificats en pratique De manière générale, les certificats utilisés pour l'authentification sont enregistrés dans le dossier `/usr/openv/var/vxss/credentials`. La particularité du serveur primaire, du fait de son double rôle d'autorité de certification, est qu'il possède deux certificats spécifiques :

- Un certificat racine auto-signé (*Subject : Root Broker*) ;
- Un certificat pour l'authentification, signé par le certificat racine (*Subject : Authentication Broker*) ; ce certificat est utilisé pour :
  - Signer des certificats du serveur primaire utilisé dans le fonctionnement interne de NetBackup (cf. certificats en vert sur la figure 12) ;
  - Signer les certificats des clients (certificat en bleu pour un client Ubuntu par exemple, cf. la figure 12).

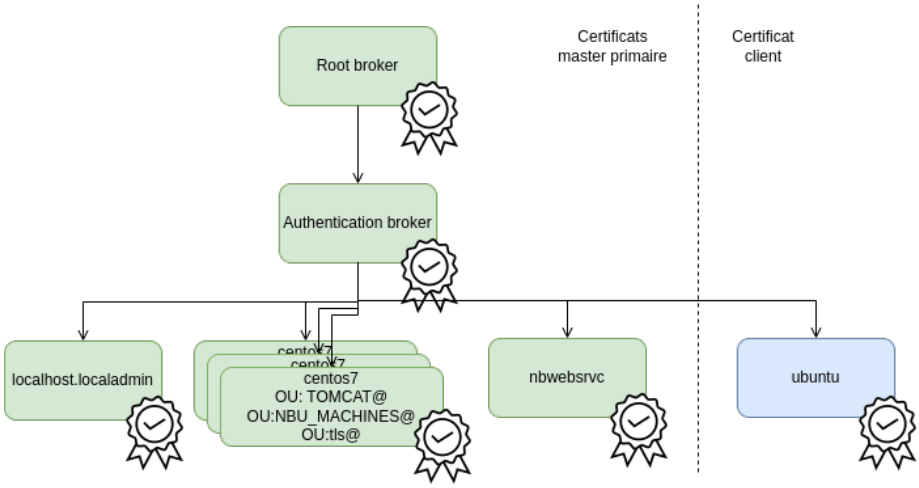

Fig. 12. Hiérarchie des certificats serveur primaire et client

Les secrets associés à ces certificats sont discutés plus en détail dans la section 3.3.2. À noter que posséder un certificat client (et sa clé privée) permet de s'authentifier auprès de nombreux services de NetBackup, et notre étude a montré que cette nouvelle surface d'attaque donne accès à un grand nombre de vulnérabilités. De plus, posséder la clé privée associée aux deux certificats principaux du serveur primaire permet de délivrer un certificat valide pour n'importe quel utilisateur.

À retenir : Il convient de s'assurer que le dossier `/usr/openv/var/vxss/credentials` du client ainsi que les clés privées du serveur (cf. paragraphe 3.3.2) sont correctement protégés.

3.2.7.8 Quelques fichiers intéressants Hormis les certificats, plusieurs fichiers, non documentés à notre connaissance, peuvent modifier le comportement des phases d'authentification et d'autorisation. De part l'aspect intrinsèquement partiel de notre étude, nous n'avons pas pu compiler une liste exhaustive ni étudier l'impact de chaque fichier, mais nous avons relevé les chemins suivants à surveiller :

— `/usr/openv/var/bprd/remote_ops/` ;
— `/usr/openv/var/vxss/credentials/dhcp_cred` ;
— `/usr/openv/var/vxss/credentials/match_required.txt` ;
— `/usr/openv/var/vxss/credentials/no_match_required.txt`.

3.2.8 Réflexions sur l'étude du mécanisme de sauvegarde Ainsi, chercher à répondre à la question purement fonctionnelle « que se passe-t-il lorsqu'une sauvegarde est réalisée ? » nous a mené à plonger dans les méandres de NetBackup et découvrir, couche par couche, certains de ses nombreux services, sans toutefois perdre de vue l'objectif initial. L'aspect stratiforme du produit, qui contribue largement à sa complexité, a néanmoins l'avantage de faciliter le découpage de l'analyse et ainsi permettre une meilleure parallélisation de l'étude. Cette approche a donc aidé à appréhender le produit et guider les choix réalisés lors de l'étude, et pourrait être appliquée à d'autres produits du même acabit.

3.3 Gestion des secrets au sein du produit

NetBackup utilise plusieurs types de matériel cryptographique, stockés sur différents composants du produit selon leur usage, qu'il est nécessaire de protéger correctement. Cette partie expose quels secrets sont utilisés par la solution, dans quel but et depuis quels composants ils sont accessibles.

3.3.1 Les secrets du catalogue Le catalogue NetBackup est une base de données Sybase divisée en plusieurs tables, dont les deux principales sont `NBDB.db` et `NBAZDB.db`.

Le mot de passe `dba` de `NBDB.db` est généré à l'installation du serveur primaire. Le chiffré `AES-256-CTR` de ce mot de passe est stocké dans le champ `VXDBMS_NB_PASSWORD` du fichier de configuration `vxdbms.conf` du serveur primaire. La clé de chiffrement `AES` est également stockée dans un autre fichier, `.yekcnedwssap`. Si ce principe n'apporte pas de sécurité en soi, il peut ralentir un utilisateur désirant connaître le mot de passe en clair.

À retenir : Ces deux fichiers étant accessibles en `root` seulement, ils permettent à un utilisateur `root` sur le serveur primaire de retrouver le mot de passe `dba` de la table `NBDB.db` et de lire ou écrire les données relatives à la gestion de la localisation des sauvegardes.

Note : l'outil `nbdb_admin` permet également à l'utilisateur `root` de changer ce mot de passe sans aucune connaissance préalable du mot de passe précédent.

De la même manière, le chiffré `3DES-EDE-CBC` du mot de passe `dba` de la table `NBAZDB.db` est stocké dans le fichier de configuration `vxdbms.conf` du serveur primaire. Deux constantes stockées dans le firmware servent de

vecteur d'initialisation et de clé de chiffrement. Le fichier de configuration n'est également accessible que par l'utilisateur `root` sur le serveur primaire.

À retenir : Avec les droits `root` sur le serveur primaire, il est possible de déchiffrer le mot de passe `dba` de `NBAZDB.db` et d'altérer la fonctionnalité d'autorisation de NetBackup.

3.3.2 La clé privée des certificats Un autre type de matériel cryptographique à protéger est celui sur lequel repose le modèle de confiance du produit, à savoir les clés privées correspondant aux certificats, en particulier pour ceux faisant office d'autorité de certification.

À noter : Ces clés privées sont stockées en clair sur le système de fichier de chaque entité, accessibles à l'utilisateur `root` seulement.

Les clés privées utilisées par le **Root Broker** et le **Authentication Broker** se trouvent dans le dossier `/usr/openv/var/global/vxss/eab/data/root/.VRTSat/profile/certstore/keystore`.

3.3.3 Clé de chiffrement en cas de backups chiffrés Enfin, un élément cryptographique important est la clé de chiffrement utilisée pour chiffrer les sauvegardes côté client avant de transiter vers les autres entités de NetBackup. La clé pour le « client-side encryption » est stockée chiffrée sur le système de fichier des clients (`/usr/openv/var/keyfile.dat`). La *passphrase* de déchiffrement de cette clé est basée sur une constante à la génération de la clé et donc commune à l'ensemble des fichiers `keyfile.dat`.

À retenir : Les secrets pour le « client-side encryption » sont stockés sur chaque client, et son utilisation (ou non) est paramétrée dans les politiques de sauvegarde. Le serveur primaire pouvant accéder à des fichiers arbitraires de ses clients, celui-ci est donc en mesure de déchiffrer toutes les données sauvegardées par tous les clients de son domaine, même si ces dernières sont chiffrées à l'aide du chiffrement côté client intégré à NetBackup.

4 Boîte à outil NetBackup

Pour toute personne s'intéressant à la sécurité de NetBackup, un certain nombre d'outils est intéressant à connaître, certains basés sur les

outils natifs de NetBackup, mais requérant souvent des droits particuliers sur les composants de l'infrastructure, d'autres développés sur la base de la connaissance acquise lors de l'analyse de sécurité à des fins de cartographie et reconnaissance d'une infrastructure NetBackup en place.

4.1 Tirer profit des outils natifs à NetBackup

En plus d'une interface graphique riche, NetBackup propose tout un ensemble d'outils en ligne de commande permettant d'effectuer des tâches d'administration diverses et variées. Ceux-ci peuvent être très intéressants pour comprendre l'état d'un système et son fonctionnement, mais il n'est pas trivial de savoir facilement et rapidement comment effectuer une tâche en particulier.

Pour illustrer ceci, on montre par la suite quelques exemples qui permettent d'accéder aux fichiers d'une sauvegarde d'un client depuis un serveur primaire sur lequel on a des privilèges élevés (e.g. `root`).

4.1.1 Obtenir une liste de clients Il est possible d'obtenir la liste des clients du serveur primaire courant avec l'outil `bpplclients`.[6] Par exemple, la commande `bpplclients -allunique -U` permettant d'avoir une liste exhaustive des clients. Cet outil permet également d'ajouter, supprimer ou modifier des clients.

4.1.2 Obtenir une liste de sauvegardes d'un client Une fois notre dévolu jeté sur un client, l'outil `bpimagelist`[7] permet de produire un rapport sur ses sauvegardes. Par exemple, la commande `bpimagelist -hoursago 48 -client client_victime` liste les sauvegardes effectuées pour `client_victime` dans les 48 dernières heures.

4.1.3 Inspecter une sauvegarde L'outil `bpflist`[8] permet de lister les fichiers sauvegardés par NetBackup. Ainsi, la commande `bpflist -U -client client_victime -rl 100` permet d'obtenir la liste des fichiers sauvegardés (avec une profondeur maximale de 100 dossiers).

[6] `https://www.veritas.com/support/fr_FR/doc/50047123-127736843-0/v14664482-127736843`

[7] `https://www.veritas.com/content/support/fr_FR/doc/50047123-127736843-0/v14662243-127736843`

[8] `https://www.veritas.com/content/support/fr_FR/doc/50047123-127736843-0/v93410403-127736843`

4.1.4 Accéder aux fichiers d'une sauvegarde Enfin, pour accéder aux fichiers de cette sauvegarde, il est possible de déclencher leur restauration vers un client que l'on contrôle avec l'outil `bprestore`.[9] La commande `bprestore -C client_victime -D client_attaquant /etc/passwd /etc/shadow` permettra d'obtenir les fichiers `/etc/passwd` et `/etc/shadow` sur le client `client_attaquant`.

Pour éviter d'écraser des fichiers, l'option `-R fichier_renommage` permet de pointer vers un fichier indiquant où déplacer les fichiers restaurés. Dans notre exemple, il pourrait contenir les lignes dans le listing 5.

Listing 5: Exemple de fichier de renommage pour `bprestore`

```
1 change /etc/passwd to /tmp/client_victime-passwd
2 change /etc/shadow to /tmp/client_victime-shadow
```

Note : Si aucun client n'est sous notre contrôle, il est possible d'en ajouter un via la commande `bpplclients`.

4.1.5 Autres finalités Cet exemple n'en est qu'un parmi d'autres, et de nombreuses autres commandes sont disponibles avec NetBackup. Elles peuvent permettre d'obtenir des informations précieuses, par exemple : les volumes utilisés, les serveurs média, les unités de stockage, les règles de durée de vie des différents stockage, la planification des sauvegardes, etc. Ces outils natifs étant toutefois plutôt pensés dans un but fonctionnel, et non à des fins d'analyse de sécurité du produit, les auteurs ont complété ces outils par les leurs, plus adaptés à leur besoin.

4.2 Outils dédiés publiés par l'équipe

L'étude du fonctionnement interne de NetBackup (cf. section 3) a mené les auteurs à réaliser plusieurs développements spécifiques pour faciliter l'audit et l'analyse de NetBackup et de son infrastructure. Cette « boîte à outils » est accessible publiquement [10] et automatise un certain nombre de tâches telles que : sans authentification, déterminer des éléments de configuration d'un composant et d'en déduire son type (client, serveur primaire, serveur OpsCenter, serveur média), ou construire une cartographie entre plusieurs composants d'une infrastructure NetBackup inconnue ; ou bien en tant qu'administrateur sur un serveur primaire,

[9] `https://www.veritas.com/content/support/fr_FR/doc/50047123-127736843-0/v14666184-127736843`

[10] `https://github.com/airbus-seclab/nbutools`

récupérer l'ensemble des hashs des mots de passe des utilisateurs de la base de données NetBackup.

Note : cette boite à outils se contente d'explorer les informations fonctionnelles disponibles et ne se base sur aucune vulnérabilité particulière.

4.2.1 Détection de la configuration d'un composant Les contraintes opérationnelles conduisent souvent les composants d'une infrastructure NetBackup à varier dans leur configuration. Pour évaluer ces divergences potentielles, AirbusSeclab a développé un outil cherchant à automatiquement obtenir ou déduire des informations de configuration pour un composant via différents accès réseau.

Ainsi, en s'appuyant sur les résultats présentés pour `pbx_exchange` dans la section 3.2.1, pour `vnetd` dans la section 3.2.3 et pour les mécanismes d'authentification dans la section 3.2.7, il est possible d'obtenir de nombreuses informations sur la configuration d'une machine, uniquement à partir d'un accès réseau non authentifié. Le listing 6 illustre par exemple la sortie de cet outil après avoir scanné un serveur primaire.

Listing 6: Extrait de sortie de `nbuscan` ciblant un serveur primaire

```
$ nbuscan.py nb-primary
--- VNETD Scan Results:
Running NetBackup version 1010000
Assigned Primary Server: nb-primary
--- VXSS Scan Results:
USE_VXSS = AUTOMATIC
VXSS_NETWORK = 10.0.0.37 PROHIBITED
VXSS_NETWORK = 10.0.0. REQUIRED
USE_AUTHENTICATION = ON
AUTHENTICATION_DOMAIN = TOMCAT@nb-primary "AUTO" VXPD nb-primary 0
AUTHENTICATION_DOMAIN = nb-primary "AUTO" PASSWD nb-primary 0
--- PBX_EXCHANGE Scan Results:
Guessed role: Primary Server
--- NETBACKUP_API Scan Results:
Server Name: nb-primary
Host ID: dd6c0f1b-52f3-4a45-b92a-cf11d96f0771
NetBackup version: NetBackup_10.1.1
SSO enabled: False
Secure Communications: Enabled
Certificate auto-deployment level: Medium
```

4.2.2 Cartographie d'une liste de composants En audit, il n'est pas rare d'identifier une liste d'adresses IP pour lesquelles le port `pbx_exchange` (`1556`) est en écoute (par un scan `nmap` par exemple),

sans pour autant connaître a priori le lien NetBackup logique entre chaque machine. Basé sur les informations de configuration de chaque composant, il est possible d'automatiser la construction d'une cartographie indiquant le lien entre les composants, leur type respectif et leur version. L'outil `nbumap.py` implémente cette automatisation. Le listing 7 illustre un exemple d'informations reconstruite par `nbumap.py` pour une liste d'adresses IP donnée et la figure 13 est l'exemple de cartographie visuelle reconstruite associée.

Listing 7: Exemple de sortie de `nbumap`

```
$ nbumap.py listening_1556_IPlist.txt --plot carto.png
Machines      Type             Version   Master          Vnetd State
172.16.142.49 OpsCenter        820000     -               -
172.16.142.50 Primary Server   820000    nb-primary-a    up
172.16.142.51 Media Server     820000    nb-primary-a    up
172.16.142.52 Client           820000    nb-primary-a    up
172.16.142.53 Client           820000    nb-primary-a    up
172.16.142.60 Primary Server   760000    nb-primary-b    up
nb-primary-a  Unknown          Unknown   Unknown         DNS
nb-primary-b  Unknown          Unknown   Unknown         DNS
```

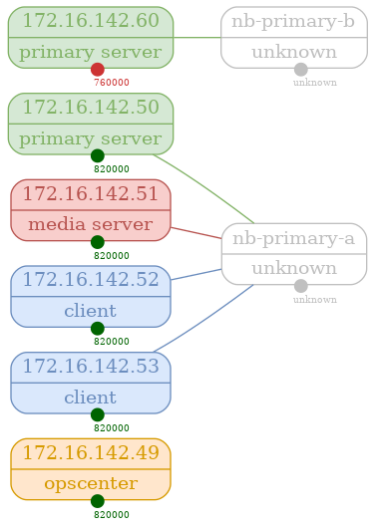

Fig. 13. Exemple de cartographie reconstruite par `nbumap.py`

Cette cartographie peut également permettre à un mainteneur de parc NetBackup d'avoir une rapide vue d'ensemble de l'état des versions de son parc.

4.2.3 Dump de la base de données `NBDB.db` La base de données `NBDB.db` contient des informations intéressantes, mais est stockée chiffrée sur les serveurs primaires. Néanmoins, en tant qu'administrateur d'un serveur primaire, il est possible d'accéder aux fichiers permettant de retrouver le mot de passe `dba` de cette base de données comme indiqué en section 3.3.1. Ce mot de passe permet par exemple d'en extraire la liste des utilisateurs et les hashs des mots de passe associés. L'outil `nbudbdump.py` automatise le déchiffrement du mot de passe `dba` à l'aide d'un fichier de « clé » (`-k`) et du fichier de configuration (`-p`) contenant le chiffré de ce mot de passe. Le listing 8 illustre un exemple de sortie de cet outil.

Listing 8: Exemple de sortie de `nbudbdump`

```
$ nbudbdump.py -k files/.yekcnedwssap -p files/vxd.conf  -H
↪  172.16.142.50
[DEBUG] TAG found. corresponding key:
↪  d2a3ee736aafa29bf997f1c355c8b2da279fb00ca879997bc69d31acc2bb9f23
[DEBUG] Sybase driver found.
[DEBUG] Connection to host: 172.16.142.50 with DBA password aaaaaa
↪  successful.
Username: DBA Hash: 01dcxxxxxxxx...xxxxxxxc4a0
Username: EMM_MAIN Hash: 01c0xxxxxxxx...xxxxxxx40f2
...
Username: NBWEBSVC Hash: 01f5xxxxxxxx...xxxxxxxefde
Username: joe Hash: 0154xxxxxxxx...xxxxxxx44d8
```

5 Conclusion et perspectives

L'analyse de cinq binaires de NetBackup a permis de relever des points d'intérêts pour le maintien et déploiement sécurisé du produit. En outre, certains de ces binaires, accessibles depuis le réseau, peuvent être utilisés à des fins d'identification et caractérisation des composants de l'infrastructure NetBackup. Pour cela, de nouveaux outils ont été publiés dans l'optique d'aider des pentesters rencontrant ce produit lors de missions ou des architectes désirant se renseigner sur les points critiques à protéger lors du déploiement de NetBackup sur un SI.

Par ailleurs, une liste de fichiers et données qu'il convient de surveiller a été établie, contenant notamment les clés privées des serveurs primaires

et serveurs OpsCenter et les base de données des serveurs primaires. Leur disponibilité, intégrité et confidentialité sont en effet critiques pour que NetBackup puisse continuer d'opérer en minimisant la surface d'attaque du SI qu'il sauvegarde. Cette surface exposée est d'autant plus essentielle que le filtrage des flux est rendu difficile par `pbx_exchange`, point d'entrée pour de nombreux services NetBackup, et que les services s'exécutent avec des privilèges élevés.

Au cours de cette étude, la priorisation de la surface à étudier s'est appuyée sur l'exposition de chaque binaire et sur leur implication fonctionnelle dans certains processus clés de l'utilisation et administration de NetBackup. Un nombre important de binaires restant à étudier, cet article pallie en partie le manque de documentation technique bas-niveau sur le sujet, facilitant ainsi le début de l'analyse à des personnes souhaitant aller plus loin dans l'étude et l'amélioration de la sécurité de ce logiciel.

Références

1. 10 règles d'or pour la conception et la mise en œuvre de services numériques. `https://www.ssi.gouv.fr/uploads/2022/08/plaquette_10_regles_or_concepteurs_services_numeriques.pdf`.
2. Bulletins de sécurité de veritas. `https://www.veritas.com/content/support/en_US/security/VTS22-004`, `https://www.veritas.com/content/support/en_US/security/VTS22-008`, `https://www.veritas.com/content/support/en_US/security/VTS22-010`, `https://www.veritas.com/content/support/en_US/security/VTS22-011`, `https://www.veritas.com/content/support/en_US/security/VTS22-012`, `https://www.veritas.com/content/support/en_US/security/VTS22-013`.
3. Historique netbackup - wikipédia. `https://fr.wikipedia.org/wiki/NetBackup#Historique`.
4. Veritas netbackup : List of security vulnerabilities. `https://cve.mitre.org/cgi-bin/cvekey.cgi?keyword=Netbackup`.
5. History of netbackup - good to know. `https://vox.veritas.com/t5/Enterprise-Data-Services/History-of-NetBackup-Good-to-Know/ba-p/782767`, 2009.
6. Apt cyber-numérique sur sauvegardiciel connecté. `https://www.rump.beer/2016/slides/APT_cyber-numerique_sur_sauvegardiciel_connecte.pdf`, 2016.
7. Veritas netbackup v6.x, v7.x, v8.0 and netbackup appliances v2.x, v3.0 - multiple critical vulnerabilities. `https://seclists.org/fulldisclosure/2017/Feb/101`, 2017.
8. Veritas netbackup v8.0 - multiple vulnerabilities. `https://seclists.org/fulldisclosure/2017/May/27`, 2017.
9. Veritas. Veritas is proud to be named a leader in the 2021 gartner magic quadrant report for enterprise backup and recovery software solutions for the 16th time. `https://www.veritas.com/form/whitepaper/gartner-mq-data-center-backup`, 2021.
10. Veritas. Netbackup : n°1 des solutions de sauvergarde d'entreprise. `https://www.veritas.com/fr/fr/protection/netbackup`, 2022.