Contrôles de conformité Nessus

Audit des configurations et du contenu du système

14 janvier 2014

(Révision 74)

Table des matières

Introduction

Ce document décrit comment Nessus 5.x peut être utilisé pour vérifier la configuration des systèmes Unix, Windows, bases de données, SCADA, IBM iSeries et Cisco par rapport à une stratégie de conformité, ainsi que pour rechercher un contenu sensible dans le contenu de divers systèmes.

Les expressions « Conformité des stratégies » et « Contrôles de conformité » sont interchangeables dans ce document.

L'audit du système SCADA est possible avec Nessus ; toutefois, cette fonctionnalité ne fait pas partie du cadre de ce document. Voir la page d'information Tenable SCADA ici pour plus d'informations.

L'exécution d'un audit de conformité est différente d'un scan des vulnérabilités, bien qu'il puisse exister des éléments communs. Un audit de conformité détermine si un système est configuré conformément à une stratégie établie. Un scan des vulnérabilités détermine si le système est ouvert à des vulnérabilités connues. Les lecteurs apprendront les différents types de paramètres de configuration et de données sensibles qui peuvent être vérifiés, comment configurer Nessus pour effectuer ces audits et comment SecurityCenter de Tenable peut être utilisé pour gérer et automatiser ce processus.

Prérequis

Ce document suppose un certain niveau de connaissances concernant le scanner de vulnérabilité Nessus. Pour de plus amples renseignements sur la façon dont Nessus peut être configuré pour effectuer des audits de correctifs Unix et Windows locaux, voir le document « Nessus Credentials Checks for Unix and Windows » (Contrôles des identifiants Nessus pour Unix et Windows) disponible sur http://www.tenable.com/products/nessus/documentation.

Clients de Nessus et de SecurityCenter

Les utilisateurs doivent avoir un abonnement commercial à Nessus ou utiliser SecurityCenter pour pouvoir effectuer les contrôles de conformité décrits dans ce document. Ces deux applications sont disponibles auprès de Tenable Network Security (http://www.tenable.com/). Une liste plus détaillée des exigences techniques nécessaires pour effectuer les contrôles de vérification est indiquée dans les chapitres suivants.

Normes et conventions

Dans l'ensemble de la documentation, les noms de fichiers, les démons (daemons) et les exécutables sont indiqués par la police **`courier bold`**.

Les options de ligne de commande et les mots clés sont aussi indiqués par la police **`courier bold`**. Les exemples de ligne de commande peuvent inclure ou non l'invite de ligne de commande et le texte provenant des résultats de la commande. Les exemples de ligne de commande sont affichés en **`courier bold`** dans la commande en cours d'exécution afin de montrer la saisie de l'utilisateur, tandis que l'exemple de sortie généré par le système utilisera la police `courier` (mais pas en gras). Voici ci-dessous un exemple d'exécution de la commande Unix **`pwd`** :

```
# pwd
/home/test/
#
```

Les remarques et considérations importantes sont mises en évidence avec ce symbole dans une boîte de texte grise.

Les conseils, exemples et meilleures pratiques sont mis en évidence avec ce symbole en texte blanc sur fond bleu.

Normes de conformité

Il existe bien des types d'exigences pour la conformité gouvernementale et financière. Il est important de bien comprendre que ces exigences de conformité servent de bases de référence minimales et peuvent être interprétées différemment selon les objectifs opérationnels de l'organisation. Les exigences de conformité doivent être établies en fonction des objectifs commerciaux pour garantir que les risques sont correctement identifiés et minimisés. Pour plus d'informations sur le développement d'un tel processus, voir le document de Tenable suivant : « Maximizing ROI on Vulnerability Management » (Maximiser le retour sur investissement pour la gestion des vulnérabilités) disponible sur http://www.tenable.com/whitepapers.

Par exemple, une entreprise peut avoir une stratégie exigeant qu'une opération d'ouverture de session, avec des mots de passe d'au moins 10 caractères, soit activée pour tous les serveurs qui contiennent des renseignements personnels sur les clients. Cette stratégie peut faciliter les efforts d'une organisation pour garantir le respect de différentes réglementations différentes.

Les réglementations et guides de conformité courants incluent notamment :

- BASEL II
- Normes du CIS (Center for Internet Security, Centre de sécurité Internet)
- COBIT (Control Objectives for Information and related Technology, Objectifs de contrôle pour la technologie de l'information et les technologies associées)
- STIG de DISA (Defense Information Systems Agency, Agence des systèmes de renseignement de la défense)
- FISMA (Federal Information Security Management Act, Loi fédérale sur la gestion de la sécurité de l'information)
- FDCC (Federal Desktop Core Configuration, Configuration de base du bureau fédéral)
- GLBA (Gramm-Leach-Bliley Act, Loi Gramm-Leach-Bliley)
- HIPAA (Health Insurance Portability and Accountability Act, Loi sur la capacité de transfert et la responsabilité de l'assurance maladie)
- Normes de sécurité ISO 27002/17799
- ITIL (Information Technology Information Library, Bibliothèque d'informations sur l'informatique)
- Consignes de configuration NIST (National Institute of Standards, Institut national des normes)
- Consignes de configuration NSA (National Security Agency, Agence de sécurité nationale)
- PCI DSS (Payment Card Industry Data Security Standards, Normes de sécurité des données de l'industrie des cartes de paiement)
- Sarbanes-Oxley (SOX)
- SDP (Site Data Protection, Protection des données de site)
- USGCB (United States Government Configuration Baseline, Ligne de base de configuration du gouvernement des États-Unis)
- Plusieurs lois d'état (par exemple, la loi sur la notification des violations de sécurité de Californie (Security Breach Notification Act) - SB 1386)

Ces contrôles de conformité assurent également la surveillance en temps réel comme la détection des intrusions et le contrôle des accès. Pour une analyse plus approfondie de la façon dont les vérifications de configuration et les solutions de gestion des vulnérabilités, fuites de données, analyse des journaux et surveillance des réseaux de Tenable peuvent

contribuer au respect des règlementations de conformité indiquées, envoyez un e-mail à sales@tenable.com pour obtenir une copie du document « Real-Time Compliance Monitoring » (Surveillance de la conformité en temps réel).

Audits de configuration, fuites de données et conformité

Qu'est-ce qu'un audit ?

Nessus peut être utilisé pour se connecter aux serveurs Unix et Windows, aux dispositifs Cisco, aux systèmes SCADA, aux serveurs IBM iSeries et aux bases de données afin de déterminer s'ils ont été configurés conformément à la stratégie de sécurité du site local. Nessus peut également rechercher des contenus non autorisés sur l'ensemble du disque dur des systèmes Windows et Unix.

Il est important que les organisations établissent une stratégie de sécurité de site avant d'effectuer un audit pour s'assurer que leurs actifs sont correctement protégés. Une évaluation des vulnérabilités déterminera si les systèmes sont vulnérables à des failles connues mais pas, par exemple, si les archives du personnel sont stockées sur un serveur public.

Il n'existe pas de norme absolue de sécurité : c'est une question de gestion des risques qui varie d'une organisation à l'autre.

Prenons l'exemple des exigences de mot de passe telles que l'antériorité minimale/maximale des mots de passe et les stratégies de verrouillage des comptes. Il peut exister d'excellentes raisons de changer de mot de passe fréquemment ou rarement. Il peut aussi exister d'excellentes raisons pour verrouiller un compte s'il a été sujet à plus de cinq échecs d'ouverture de session, mais, s'il s'agit d'un système essentiel pour la mission, il pourrait être plus prudent d'établir un nombre plus élevé d'échecs ou même de désactiver complètement les verrouillages.

Ces paramètres de configuration concernent dans une large mesure la gestion du système et la stratégie de sécurité, mais pas particulièrement les vulnérabilités du système ou les correctifs manquants. Nessus peut effectuer des contrôles de conformité pour les serveurs Unix et Windows. Les stratégies peuvent être très simples ou très complexes selon les exigences de chaque scan de conformité individuel.

Audit et scan des vulnérabilités

Nessus peut effectuer des scans des vulnérabilités des services réseau et se connecter à des serveurs pour détecter les correctifs manquants. Toutefois, une absence de vulnérabilités ne signifie pas que les serveurs sont configurés correctement ou qu'ils sont « conformes » à une norme particulière.

L'avantage de l'utilisation de Nessus pour effectuer des scans des vulnérabilités et des audits de conformité est que toutes ces données peuvent être obtenues en même temps. Savoir comment un serveur est configuré, comment il est corrigé et quelles vulnérabilités sont présentes peut aider à déterminer les mesures à prendre pour limiter les risques.

À un plus haut niveau, si cette information est agrégée pour un réseau entier ou une classe d'actifs complète (comme avec SecurityCenter de Tenable), la sécurité et le risque peuvent être analysés globalement. Ceci permet aux auditeurs et aux gestionnaires réseau d'identifier les tendances dans les systèmes non conformes et de régler les contrôles pour les corriger à plus grande échelle.

Exemples d'éléments d'audit

Les sections ci-dessous concernent les audits de configuration sur les systèmes Windows, Unix, bases de données, IBM iSeries et Cisco.

Le moteur regex (expression rationnelle) de Nessus 5 repose sur un dialecte Perl et est considéré comme « Extended POSIX » (POSIX étendu) en raison de sa flexibilité et de sa vitesse.

Les fichiers d'audit doivent toujours être codés au format ANSI. Les fichiers qui utilisent le codage Unicode, Unicode big endian ou UTF-8 ne sont pas utilisables.

Windows

Nessus peut tester tout paramètre qui peut être configuré comme une « stratégie » dans le cadre de Microsoft Windows. Il existe plusieurs centaines de paramètres de registre qui peuvent être vérifiés et les permissions des fichiers, répertoires

et objets peuvent également être analysées. Une liste non exhaustive d'exemples d'audits inclut les tests des paramètres suivants :

- Durée de verrouillage de comptes
- Conserver le journal de sécurité
- Permettre l'ouverture d'une session locale
- Appliquer l'historique des mots de passe

Voici un exemple d'élément d'« audit » pour les serveurs Windows :

```
<item>
 name: "Minimum password length"
 value: 7
</item>
```

Cet audit particulier recherche le paramètre « Minimum password length » (Longueur minimale du mot de passe) sur un serveur Windows et crée une alerte si la valeur est inférieure à sept caractères.

Nessus peut aussi rechercher des données sensibles sur les ordinateurs Windows. Voici un exemple de recherche des numéros de carte de crédit Visa dans plusieurs formats de fichier :

```
<item>
 type: FILE_CONTENT_CHECK
 description: "Determine if a file contains a valid VISA Credit Card Number"
 file_extension: "xls" | "pdf" | "txt"
 regex: "([^0-9-]|^)(4[0-9]{3}( |-|)([0-9]{4})( |-|)([0-9]{4})( |-|)([0-9]{4}))([^0-9-
       ]|$)"
 expect: "VISA" | "credit" | "Visa" | "CCN"
 max size: "50K"
 only_show: "4"
</item>
```

Ce contrôle examine les fichiers Excel, Adobe et texte afin de rechercher des motifs indiquant qu'un ou plusieurs numéros de carte de crédit Visa valides sont présents.

Unix
Nessus peut être généralement utilisé pour tester les permissions de fichiers et le contenu d'un fichier, pour exécuter des processus et pour contrôler l'accès des utilisateurs pour différents systèmes basés sur Unix. Les contrôles existants sont actuellement disponibles pour vérifier Solaris, Red Hat, AIX, HP-UX, SuSE, Gentoo et les dérivés FreeBSD d'Unix.

```
<item>
 name: "min_password_length"
 description: "Minimum password length"
 value: "14..MAX"
</item>
```

Cet audit contrôle si la longueur minimale du mot de passe sur un système Unix est 14 caractères.

Cisco

Nessus peut tester la configuration en cours pour les systèmes exécutant le système d'exploitation Cisco IOS et confirmer qu'elle est conforme aux normes des stratégies de sécurité. Les contrôles peuvent être effectués par une connexion non privilégiée ou une connexion utilisant le mot de passe privilégié « enable » (activer).

```
<item>
 type: CONFIG_CHECK
 description: "Require AAA service"
 info: "Verify centralized authentication, authorization and accounting"
 info: "(AAA)service (new-model) is enabled."
 item: "aaa new-model"
</item>
```

Pare-feu Palo Alto

Nessus utilise les transformations XSL (XSLT) et une API native pour demander des informations aux périphériques Palo Alto pilotés par le système d'exploitation PAN-OS. Les demandes sont acheminées via l'interface HTTP ou HTTPS du pare-feu ; elles exigent l'authentification administrateur `Superuser` (Super utilisateur) `Superuser (readonly)` (Super utilisateur (lecture seule)) sur PAN-OS >= 4.1.0, ou `Superuser` (Super utilisateur) sur PAN-OS < 4.1.0. Cette procédure permet d'effectuer les audits par rapport à une `operational config` (configuration opérationnelle) sur le périphérique.

```
<custom_item>
 type: AUDIT_XML
 description: "Palo Alto Security Settings - 'fips-mode = on'"
 info: "Fips-mode should be enabled."
 api_request_type: "op"
 request: "<show><fips-mode></fips-mode></show>"
 xsl_stmt: "<xsl:template match=\"/\">"
 xsl_stmt: " <xsl:apply-templates select=\"//result\"/>"
 xsl_stmt: "</xsl:template>"
 xsl_stmt: "<xsl:template match=\"//result\">"
 xsl_stmt: "fips-mode: <xsl:value-of select=\"text()\"/>"
 regex: "fips-mode:[\\s\\t]+"
 expect: "fips-mode:[\\s\\t]+on"
</custom_item>
```

IBM iSeries

Avec les identifiants fournis, Nessus peut tester la configuration des systèmes exécutant IBM iSeries et confirmer qu'elle est conforme aux normes des stratégies de sécurité.

```
<custom_item>
 type: AUDIT SYSTEMVAL
 systemvalue: "QALWUSRDMN"
 description: "Allow User Domain Objects (QALWUSRDMN) – '*all'"
 value_type: POLICY_TEXT
 value_data: "*all"
 info: "\nref :
       http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/books/sc415302.pdf
       pg. 21"
</custom_item>
```

NetApp Data ONTAP

Avec les identifiants fournis, Nessus peut tester la configuration des systèmes exécutant NetApp Data ONTAP et confirmer qu'elle est conforme aux normes des stratégies de sécurité.

```
<custom_item>
 type: CONFIG_CHECK
 description: "1.2 Secure Storage Design, Enable Kerberos with NFS –
       'nfs.kerberos.enable = on'"
 info: "NetApp recommends the use of security features in IP storage protocols to
       secure client access"
 solution: "Enable Kerberos with NFS"
 reference: "PCI|2.2.3"
 see_also: "http://media.netapp.com/documents/tr-3649.pdf"
 regex: "nfs.kerberos.enable[\\s\\t]+"
 expect: "nfs.kerberos.enable[\\s\\t]+on"
</custom_item>
```

Bases de données
Nessus peut être configuré pour se connecter aux types de base de données suivants et déterminer la conformité à la stratégie de sécurité locale :

- SQL Server
- Oracle
- MySQL
- PostgreSQL
- DB2
- Informix/DRDA

En général, Tenable recommande d'exécuter un scan de conformité de base de données avec un utilisateur possédant des privilèges SYSDBA pour Oracle, « **sa** » ou un compte avec un rôle de serveur sysadmin pour MS-SQL et un compte utilisateur d'instance DB2 pour DB2 afin de garantir l'exhaustivité du rapport ; en effet, certains paramètres et tables système ou cachés sont uniquement accessibles par un compte dotés de tels privilèges. Veuillez noter que pour Oracle, un utilisateur auquel est affecté le rôle DBA effectuera souvent la plupart des contrôles dans les audits Tenable, mais certains contrôles signaleront des erreurs imputables à des privilèges d'accès insuffisants. Ce même argument est également applicable aux autres bases de données. Un compte reposant sur des privilèges moindres peut être utilisé pour les audits de base de données mais avec un inconvénient : il ne sera pas possible de garantir un rapport exhaustif.

Les audits de base de données sont normalement constitués d'instructions sélectionnées qui récupèrent les détails associés à la sécurité de votre base de données comme l'existence ou l'état de procédures stockées non sécurisées. L'exemple ci-dessous détermine si la procédure stockée « **xp_cmdshell** » potentiellement dangereuse est activée :

```
<custom_item>
 type: SQL_POLICY
 description: "xp_cmdshell option"
 info: "The xp_cmdshell extended stored procedures allows execution of host
       executables outside the controls of database access permissions and may be
       exploited by malicious users."
 info: "Checking that the xp_cmdshell stored procedure is set to '0'"
 sql_request: "select value_in_use from sys.configurations where name = 'xp_cmdshell'"
 sql_types: POLICY_INTEGER
 sql_expect: "0"
</custom_item>
```

La capacité d'écrire des fichiers d'audit pour chaque organisation et de rechercher les données sensibles est très utile. Ce document décrit comment créer des stratégies personnalisées pour rechercher divers types de données.

Rapports d'audit

Lorsqu'un audit est effectué, Nessus essaie de déterminer si l'hôte est conforme, non conforme ou si les résultats ne sont pas concluants.

Dans Nessus, les résultats de conformité sont consignés sous la forme « Pass » (Succès), « Fail » (Échec) et « Warning » (Avertissement). L'interface utilisateur Nessus et SecurityCenter de Tenable consignent les résultats sous la forme « Info » pour les contrôles conformes, « High » (Élevé) pour les contrôles non conformes et « Medium » (Moyen) pour les contrôles peu concluants (par exemple, un contrôle de permission pour un fichier qui est introuvable sur le système).

À l'inverse d'un contrôle des vulnérabilités qui signale seulement si la vulnérabilité est réellement présente, un contrôle de conformité signale toujours quelque chose. De cette manière, les données peuvent être utilisées comme base d'un rapport d'audit pour montrer qu'un hôte a réussi ou échoué à un test spécifique ou qu'il n'a pas pu être testé correctement.

Technologie requise

Plugin .nbin Nessus pour la conformité de configuration Unix et Windows

Tenable a créé deux plugins Nessus (ID 21156 et 21157) qui déploient les API à utiliser pour effectuer les audits portant sur les systèmes Unix et Windows. Ces plugins ont été précompilés avec le format Nessus « **`.nbin`** ».

Ces plugins et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux et des utilisateurs de SecurityCenter. Ce document présente également deux outils Windows qui facilitent la création de fichiers **`.audit`** Windows personnalisés et un outil Unix pour créer des fichiers **`.audit`** Unix.

Pour les audits de conformité Unix, seule l'authentification SSH est prise en charge. Les protocoles hérités tels que Telnet ne sont pas autorisés pour des raisons de sécurité.

Plugin .nbin Nessus pour la conformité du contenu Windows

Tenable a créé un plugin Nessus (ID 24760) nommé « Windows File Contents Check » (Contrôle de contenu de fichier Windows) qui déploie les API utilisées pour rechercher les contenus non conformes sur les systèmes Windows, comme les renseignements personnels (PII - Personally Identifiable Information) ou les informations de santé protégées (PHI - Protected Health Information). Ces plugins sont précompilés avec le format Nessus « **`.nbin`** ». Les plugins et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux et des utilisateurs de SecurityCenter

Veuillez noter que les systèmes Unix ne sont pas scannés par le plugin 24760.

Plugin .nbin Nessus pour la conformité des bases de données

Tenable a créé un plugin Nessus (ID 33814) nommé « Database Compliance Checks » (Contrôles de conformité des bases de données) qui déploie les API utilisées pour vérifier divers systèmes de base de données. Ce plugin est précompilé avec le format Nessus « **`.nbin`** ». Le plugin et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux et des utilisateurs de SecurityCenter.

Les contrôles de conformité des bases de données ne sont pas disponibles pour une utilisation avec la version 3.4.3 de Security Center ou les versions plus anciennes.

Plugin .nbin Nessus pour la conformité IBM iSeries

Tenable a créé un plugin Nessus (ID 57860) nommé « IBM iSeries Compliance Checks » (Contrôles de conformité IBM iSeries) qui déploie les API utilisées pour vérifier divers systèmes exécutant IBM iSeries. Ce plugin est précompilé avec le format Nessus « **`.nbin`** ». Le plugin et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux.

Pour effectuer avec succès un scan de conformité sur un système iSeries, les utilisateurs authentifiés doivent posséder les privilèges définis ci-dessous :

1. Un utilisateur doté de l'autorité (*ALLOBJ) ou audit (*AUDIT) peut vérifier toutes les valeurs système. Un tel utilisateur appartient normalement à la classe (*SECOFR).

2. Les utilisateurs de la classe (*USER) ou (*SYSOPR) peuvent vérifier la plupart des valeurs, sauf QAUDCTL, QAUDENDACN, QAUDFRCLVL, QAUDLVL, QAUDLVL2 et QCRTOBJAUD.

Si un utilisateur ne dispose pas des privilèges lui permettant d'accéder à une valeur, la valeur renvoyée sera *NOTAVL.

Plugin .nbin Nessus pour la conformité Cisco

Tenable a créé un plugin Nessus (ID 46689) nommé « Cisco IOS Compliance Checks » (Contrôles de conformité Cisco IOS) qui déploie les API utilisées pour vérifier les systèmes exécutant le système d'exploitation CISCO IOS. Ce plugin est précompilé avec le format Nessus « **`.nbin`** ». Le plugin et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux. Ce contrôle de conformité peut être effectué avec une configuration Saved (Sauvegardée), Running (En cours d'exécution) ou Startup (Démarrage).

Plugin .nbin Nessus pour la conformité Palo Alto

Tenable a créé un plugin Nessus (ID 64095) nommé « Palo Alto Networks PAN-OS Compliance Checks » (Contrôles de conformité Palo Alto Networks PAN-OS) qui déploie les API utilisées pour vérifier les systèmes exécutant les périphériques Palo Alto. De plus, un plugin Nessus (ID 64286) nommé « Palo Alto Networks Settings » (Paramétrage Palo Alto Networks) permet de configurer les informations d'authentification requises pour effectuer l'audit. Ces plugins sont précompilés avec le format Nessus « **`.nbin`** ». Le plugin et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux. Ce contrôle de conformité peut être effectué par rapport à des configurations opérationnelles.

Plugin .nbin Nessus pour la conformité VMware

Tenable a créé un plugin Nessus (ID 64455) nommé « VMware vCenter/vSphere Compliance Checks » (Contrôles de conformité VMware vCenter/vSphere) qui déploie l'API SOAP VMware pour vérifier les logiciels ESX, ESXi et vCenter. Les informations d'authentification requises pour effectuer un audit peuvent être ajoutées sous « VMware vCenter SOAP API Settings » (Paramétrage de l'API SOAP VMware vCenter) dans la section « Advanced » (Configuration avancée) d'une stratégie. Le plugin et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux. Pour plus d'informations sur l'exécution d'un audit portant sur VMware, consultez l'article du blog associé.

Plugin .nbin Nessus pour la conformité Citrix XenServer

Tenable a créé un plugin Nessus (ID 69512) nommé « Citrix XenServer Compliance Checks » (Contrôles de conformité Citrix XenServer) qui déploie les API utilisées pour vérifier les systèmes exécutant Citrix XenServer, ainsi que les fournisseurs créant leurs propres versions de XenServer à partir de open sourced code (code open source). Les informations d'authentification requises pour effectuer un audit peuvent être ajoutées dans les préférences « Citrix XenServer Compliance Checks » (Contrôles de conformité Citrix XenServer) dans la section « Advanced » (Configuration avancée) d'une stratégie. Le plugin et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux. Pour plus d'informations sur l'exécution d'un audit portant sur XenServer, consultez l'article du blog associé.

Plugin .nbin Nessus pour la conformité HP ProCurve

Tenable a créé un plugin Nessus (ID 70271) nommé « HP ProCurve Compliance Checks » (Contrôles de conformité HP ProCurve) qui déploie les API utilisées pour vérifier les systèmes exécutant les solutions ProCurve de HP. Les informations d'authentification requises pour effectuer un audit peuvent être ajoutées dans les préférences « HP ProCurve Compliance Checks » (Contrôles de conformité HP ProCurve) dans la section « Advanced » (Configuration avancée) d'une stratégie. Le plugin et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux.

Plugin .nbin Nessus pour la conformité FireEye

Tenable a créé un plugin Nessus (ID 70469) nommé « FireEye Compliance Checks » (Contrôles de conformité FireEye) qui déploie les API utilisées pour vérifier les systèmes exécutant FireEye. Les informations d'authentification requises pour effectuer un audit peuvent être ajoutées dans les préférences « FireEye Compliance Checks » (Contrôles de conformité FireEye) dans la section « Advanced » (Configuration avancée) d'une stratégie. Le plugin et les stratégies d'audit correspondantes sont à la disposition des clients commerciaux.

Stratégies d'audit

Tenable a développé diverses stratégies d'audit pour les plateformes Unix, Windows, Palo Alto, IBM iSeries, VMware et Cisco. Elles sont disponibles sous forme de fichiers texte **`.audit`** au profit des abonnés commerciaux et peuvent être

téléchargées à partir du portail d'assistance de Tenable sur https://support.tenable.com/. Pour les dernières nouvelles concernant la fonctionnalité d'audit de Tenable et toutes les dernières versions des fichiers **.audit**, consultez les forums de discussion : https://discussions.nessus.org/.

La plupart des aspects des audits de conformité courants, comme les exigences de SOX, FISMA et PCI DSS, ont été pris en compte lors de la création de ces stratégies d'audit, bien qu'ils ne soient pas représentés sous la forme de fichiers d'audit officiels pour ces critères. Les utilisateurs sont encouragés à examiner ces stratégies **.audit** et à personnaliser ces contrôles pour leur environnement local. Les utilisateurs peuvent renommer les fichiers **.audit** en fonction des descriptions locales. D'autres stratégies **.audit** proviennent directement des paramètres de configuration recommandés par CERT, CIS, NSA et NIST.

Tenable a l'intention de créer plusieurs types de fichiers **.audit** différents en fonction des réactions des clients et de l'évolution des « meilleures pratiques ». Plusieurs organismes de conseil et clients de Tenable ont également commencé à mettre en place leurs propres stratégies **.audit** et ont exprimé leur intérêt à partager celles-ci avec les autres utilisateurs commerciaux Nessus. Les Tenable Network Security Discussion Forums constituent un moyen facile de partager les stratégies **.audit** ou simplement d'interagir avec la communauté Nessus : https://discussions.nessus.org/.

Utilitaires pratiques

Tenable a conçu un outil qui permet de convertir les fichiers **.inf** en fichiers **.audit** Nessus afin d'effectuer des audits Windows. Cet outil s'appelle **i2a** et il est également traité dans la suite de ce document.

Deux outils Unix peuvent être utilisés pour créer des fichiers **.audit** Unix. Le premier, **c2a** (pour « configuration to audit »), peut être utilisé pour créer des fichiers **.audit** Unix directement à partir de fichiers de configuration existants. Par exemple, si votre fichier de configuration Sendmail est configuré correctement selon la stratégie du site, l'outil **c2a** peut créer une stratégie d'audit basée sur la somme de contrôle MD5 du fichier ou sur des paires valeur/argument spécifiques dans le fichier **sendmail.cf**. Le second outil, **p2a** (pour « package to audit »), peut être utilisé pour créer des fichiers **.audit** Unix à partir du progiciel de base sur un système Unix (Linux basé sur RPM ou Solaris 10) ou à partir d'un fichier de texte plat contenant une liste de noms de progiciels.

Scanners Nessus Unix et Windows

De nombreuses plateformes peuvent être utilisées pour exécuter des contrôles de conformité et, de façon générale, le système d'exploitation sous-jacent sur lequel Nessus réside n'a aucune importance. Vous pouvez effectuer des audits de conformité sur un serveur Windows 2003 à partir d'un ordinateur portable OS X et vous pouvez aussi vérifier un serveur Solaris à partir d'un ordinateur portable Windows.

Identifiants pour les systèmes à vérifier

Dans tous les cas, des identifiants Unix SSH, domaine Windows, IBM iSeries, Cisco IOS ou bases de données sont nécessaires pour que Nessus entre en session sur les serveurs ciblés. Dans la plupart des cas, l'utilisateur doit être un « super-utilisateur » ou un utilisateur normal avec capacité d'élévation des privilèges (par exemple, **sudo**, **su** ou **su+sudo**). Si l'utilisateur qui effectue l'audit ne dispose pas des privilèges d'un « super-utilisateur », bien des commandes du système distant ne pourront pas être exécutées ou renverront des résultats incorrects.

Le compte Windows utilisé pour les identifiants de connexion doit avoir la permission de lire la stratégie de l'ordinateur de l'utilisateur. Si un hôte cible ne fait pas partie d'un domaine Windows, le compte doit être un membre du groupe des administrateurs de l'hôte. Si l'hôte fait partie d'un domaine, le groupe des administrateurs du domaine sera un membre du groupe des administrateurs de l'hôte et le compte aura accès à la stratégie de l'ordinateur de l'utilisateur s'il est membre du groupe des administrateurs du domaine.

Pour effectuer les contrôles de conformité du contenu Windows, en plus de la connexion au système avec des privilèges de domaine, l'accès à la WMI (Windows Management Instrumentation, Infrastructure de gestion Windows) doit aussi être autorisé. Si cet accès n'est pas disponible, Nessus indique que l'accès WMI n'était pas disponible pour le scan.

Les contrôles de conformité de base de données nécessitent seulement les identifiants de la base de données pour effectuer un audit de conformité complet d'une base de données. En effet, c'est la base de données, et non le système d'exploitation de l'hôte, qui est scannée pour vérifier la conformité.

Les audits de conformité Cisco IOS nécessitent typiquement le mot de passe « enable » pour effectuer un audit de conformité complet de la configuration du système. En effet, Nessus procède à l'audit de la sortie de la commande « **`show config`** », mise uniquement à la disposition des utilisateurs privilégiés. Si l'utilisateur Nessus utilisé pour l'audit a déjà des privilèges « enable », le mot de passe « enable » n'est pas requis.

Pour de plus amples informations sur la configuration de Nessus ou de SecurityCenter afin d'effectuer des contrôles des vulnérabilités authentifiés locaux, voir le document « Nessus Credentials Checks for Unix and Windows » (Contrôles des identifiants Nessus pour Unix et Windows) disponible sur http://www.tenable.com/products/nessus/documentation.

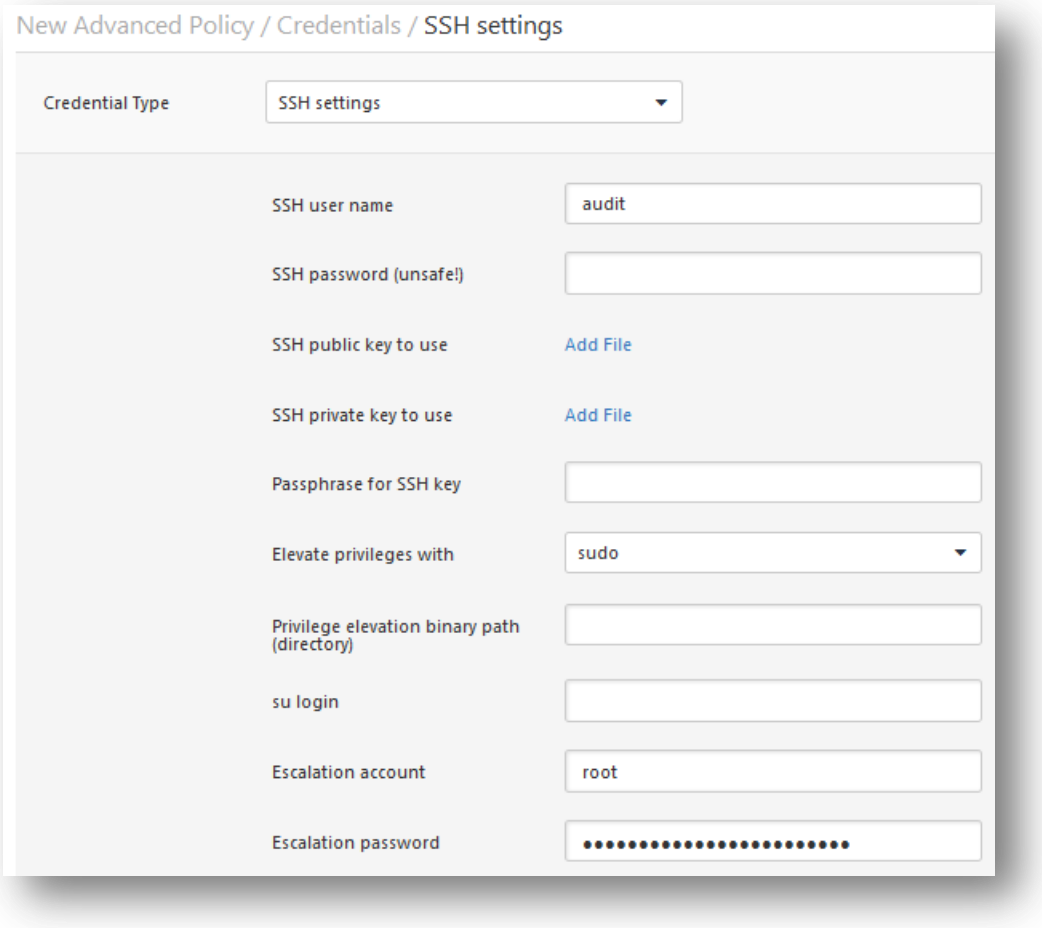
Utilisation de « su », « sudo » et « su+sudo » pour les audits

Utilisez « **`su+sudo`** » lorsque la stratégie de la société interdit à Nessus de se connecter à un hôte distant avec l'utilisateur root ou un utilisateur possédant des privilèges « **`sudo`** ». Sur une connexion à distance, l'utilisateur Nessus non privilégié peut « **`su`** » (remplacer l'utilisateur) par un autre disposant de privilèges **`sudo`**.

Les scans authentifiés Unix les plus efficaces sont ceux pour lesquels les identifiants fournis ont des privilèges « root ». Puisqu'un grand nombre de sites ne permettent pas de connexion à distance au niveau root, les utilisateurs Nessus peuvent invoquer « **`su`** », « **`sudo`** » ou « **`su+sudo`** » avec un mot de passe séparé pour un compte qui a été configuré pour disposer des privilèges appropriés.

En outre, si un fichier **`known_hosts`** SSH est disponible et fourni dans le cadre de la stratégie de scan, Nessus essaiera seulement de se connecter aux hôtes de ce fichier. Ceci garantit que la combinaison nom d'utilisateur-mot de passe utilisée pour vérifier les serveurs SSH connus ne sera pas utilisée pour une tentative de connexion à un système qui pourrait ne pas être sous votre contrôle.

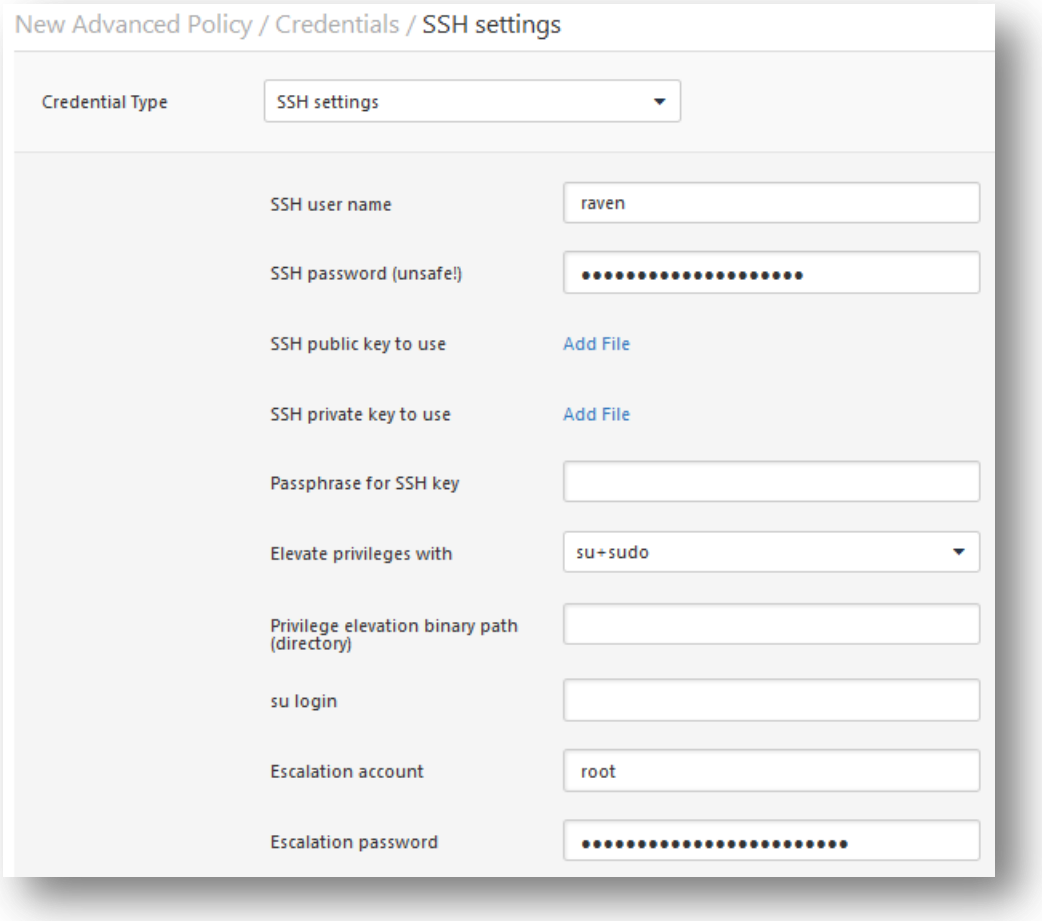
Exemple sudo
Cette capture d'écran illustre l'utilisation de « **sudo** » avec les clés SSH. Dans cet exemple, le compte de l'utilisateur est « audit », qui a été ajouté au fichier **/etc/sudoers** sur le système à scanner. Le mot de passe fourni est celui du compte « audit », et non le mot de passe root. Les clés SSH correspondent aux clés créées pour le compte « audit » :



Exemple su+sudo
Avec la sortie de Nessus 4.2.2, une nouvelle méthode d'élévation des identifiants a été ajoutée pour les hôtes basés sur Unix et pour lesquels **sudo** est installé : « **su+sudo** ». Cette méthode permet de fournir des identifiants pour un compte qui ne possède pas de permissions **sudo**, **su** à un compte d'utilisateur qui en possède puis d'utiliser la commande **sudo**.

Cette configuration assure une meilleure sécurité pour les identifiants pendant le scan et satisfait aux exigences de conformité de nombreuses organisations.

Pour activer cette fonction, sélectionnez simplement « su+sudo » dans la section « Elevate privileges with » (Élever les privilèges avec) sous « Credentials/SSH settings » (Identifiants/paramètres SSH) comme illustré par la capture d'écran ci-dessous :



Dans les champs « SSH user name » (Nom d'utilisateur SSH) et « SSH password » (Mot de passe SSH), saisissez les identifiants qui n'ont pas de privilèges **`sudo`**. Dans l'exemple ci-dessus, le compte d'utilisateur est « raven ». Dans le menu déroulant « Elevate privileges with » (Élever les privilèges avec), sélectionnez « su+sudo ». Dans les champs « su login » (connexion su) et « Escalation password » (Mot de passe d'élévation), saisissez le nom d'utilisateur et le mot de passe qui *possèdent* des identifiants privilégiés, soit « sumi » dans cet exemple. Aucun autre changement de stratégie de scan n'est requis.

Remarque importante concernant sudo
Lors de l'audit des systèmes Unix avec **`su`**, **`sudo`**, ou **`su+sudo`**, il faut se souvenir des points suivants :

- Si le système Unix a été renforcé pour limiter les commandes qui peuvent être exécutées par **`sudo`** ou les fichiers auxquels accèdent les utilisateurs distants, cela peut affecter l'audit. Si vous soupçonnez que la vérification est limitée par des mesures de sécurité, comparez les audits non-root à un audit root.

- La commande **`sudo`** n'est pas intégrée à Solaris et doit être téléchargée et installée si le système cible fonctionne sous Solaris. Assurez-vous que le binaire **`sudo`** est accessible en tant que « **`/usr/bin/sudo`** ».

- Lors d'un scan avec **`known_hosts`**, le scan de Nessus doit également toujours spécifier un hôte à scanner. Par exemple, si une catégorie C a été scannée mais qu'un fichier **`known_hosts`** contenant seulement 20 hôtes individuels dans cette catégorie C a été téléchargé, Nessus scannerait simplement ces hôtes dans le fichier.

- Certaines configurations basées sur Unix exigent que les commandes lancées par sudo soient effectuées à partir des sessions **tty**. Les scans des vulnérabilités Nessus effectués avec l'option « **su+sudo** » ne satisfont pas à cette exigence. Si vous utilisez l'option « **su+sudo** », il faut créer une exception sur le système cible. Pour déterminer si c'est le cas pour votre distribution Unix, saisissez la commande suivante en tant que root sur le système à scanner :

  ```
  # grep requiretty `locate sudoers` | grep -v "#" | grep /etc
  ```

  Si la ligne « **requiretty** » se trouve dans le fichier de configuration **sudoers**, une exception à cette règle doit être saisie dans le fichier **/etc/sudoers** comme suit :

  ```
  Defaults requiretty
  Defaults:{userid} !requiretty
  ```

  {userid} est le nom d'utilisateur qui sera utilisé pour exécuter la commande « **sudo** » (la page « su login » dans la section Credentials/SSH de la stratégie). Assurez-vous aussi que la ligne suivante se trouve dans le fichier **sudoers** :

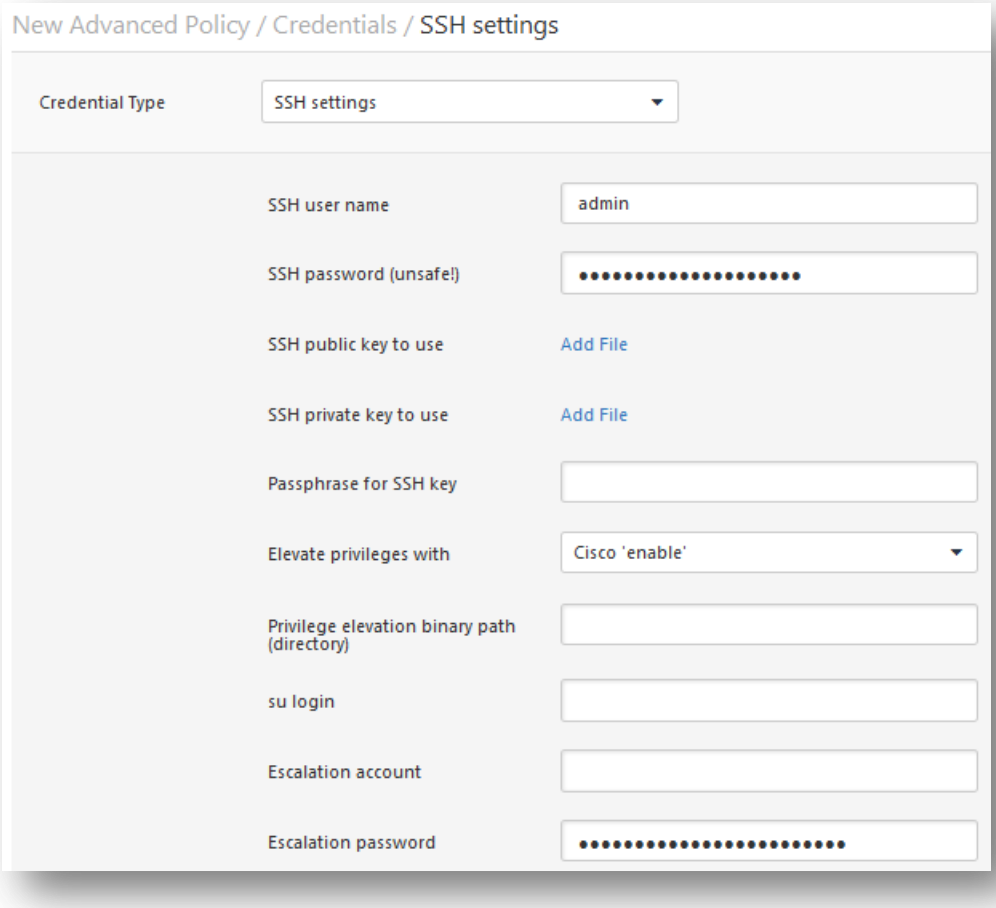
  ```
  {userid} ALL=(ALL) ALL
  ```

  {userid} est une nouvelle fois le nom d'utilisateur qui sera utilisé pour exécuter la commande « **sudo** » (la page « su login » dans la section Credentials/SSH de la stratégie).

Exemple Cisco IOS

Seule l'authentification SSH est prise en charge. Les périphériques IOS hérités dont l'authentification nécessite Telnet ne peuvent pas être scannés avec les contrôles de conformité Nessus Cisco.

Les identifiants Cisco IOS sont configurés depuis l'écran d'authentification « **SSH settings** » (Paramètres SSH) dans l'interface utilisateur Nessus. Saisissez le nom d'utilisateur SSH et le mot de passe requis pour vous connecter au routeur Cisco. Pour spécifier que les privilèges doivent être élevés avec « Enable » (Activer), choisissez « **Cisco 'enable'** » (Cisco – activer) à côté du paramètre « **Elevate privileges with** » (Élever les privilèges avec) et saisissez le mot de passe enable à côté de « **Escalation password** » (Mot de passe d'élévation).



Conversion des fichiers Windows .inf en fichiers .audit avec i2a

Si vous disposez des fichiers de stratégie Windows (qui ont souvent l'extension « **`.inf`** »), ceux-ci peuvent être convertis en fichiers **`.audit`** à utiliser pour les audits Nessus de serveurs Windows.

Obtention et installation de l'outil

L'outil **`i2a`** est disponible sous forme de fichier zip sur le portail d'assistance Tenable : https://support.tenable.com/. Cet outil n'utilise pas d'interface graphique et est exécuté à partir de la ligne de commande.

Extrayez le contenu du fichier dans le répertoire de votre choix, puis déplacez les fichiers Windows **`.inf`** dans le même répertoire.

Conversion de .inf à .audit

Exécutez l'outil de conversion à partir de l'invite de commande en saisissant simplement :

```
# i2a-x.x.x.exe yourfile.inf file.audit
```

Dans cet exemple, **`yourfile.inf`** est le fichier **`.inf`** source et **`file.audit`** est le fichier **`.audit`** cible.

Analyse de la conversion

Tenable a tenté d'obtenir une conversion aussi proche que possible de 100 % entre ce qui peut être décrit dans un fichier **`.inf`** et ce qui peut être vérifié dans un fichier **`.audit`**. Toutefois, il n'existe que peu d'éléments de stratégie qui ne peuvent pas être testés avec la technologie Nessus 5 actuelle.

Un journal du processus de conversion est créé pour chaque exécution de l'outil **`i2a`**. Il contient une vérification ligne par ligne du processus complet de conversion. Si une ligne du fichier **`.inf`** ne peut pas être convertie, elle sera incluse dans ce fichier journal.

Format de paramétrage .inf correct

Pour les contrôles indiqués comme n'ayant pas pu être effectués dans le fichier journal, il faut s'assurer qu'ils sont conformes aux formats acceptables répertoriés ci-dessous.

Les paramètres **System Access** (Accès système), **System Log** (Journal système), **Security Log** (Journal sécurité), **Application Log** (Journal application) et **Event Audit** (Audit des événements) partagent le même format. Chaque entrée est décrite par une clé « **`Key`** » suivie d'une valeur « **`value`** ».

Syntaxe :

```
Key = value
```

Dans le cas ci-dessus, **`Key`** est l'élément à vérifier et **`value`** est la valeur attendue pour cette clé sur le système distant.

Exemple :

```
MinimumPasswordLength = 8
```

Le format pour les paramètres **Privilege Rights** (Droits de privilèges) est similaire à celui mentionné ci-dessus, mais la valeur peut être vide dans ce paramètre.

Syntaxe :

```
PriviledgeRight = User1,User2…UserN
```

Exemple :

```
SeNetworkLogonRight = *S-1-5-32-545,*S-1-5-32-544
```

Ou :

```
SeTcbPrivilege =
```

Un paramétrage de **Registry Key** (clé de registre) comprend les quatre parties suivantes :

- Registry key (Clé de registre) : la clé de registre qui doit être vérifiée.
- Inheritance value (Valeur d'héritage) : identifie si les permissions pour cette clé de registre sont héritées ou non. La valeur peut être [0-4].
- DACL (DACL) : DACL est une ACL qui est contrôlée par le propriétaire d'un objet et qui spécifie l'accès particulier que des utilisateurs ou des groupes peuvent avoir à cet objet.

- SACL (SACL) : SACL est une ACL qui contrôle la création des messages d'audit pour les tentatives d'accès à un objet sécurisable.

Syntaxe :

```
"Registry Key",Inheritance value,
"D:dacl_flags(string_ace1)...(string_acen)S:sacl_flags(string_ace1)... (string_acen)"
```

Les champs DACL et SACL peuvent être vides, auquel cas le contrôle est ignoré.

Exemple :

```
"MACHINE\SYSTEM\CurrentControlSet\Control\Class",0,"D:PAR(A;CI;KA;;;BA)(A;CIIO;KA;;;CO)
      S:PAR(AU;OICIFA;CC;;;WD)"
```

Le format pour le paramètre **File Security** (Fichier de sécurité) est similaire au format de clé de registre décrit ci-dessus.

Syntaxe :

```
"File Object",Inheritance value,
      "D:dacl_flags(string_ace1)...(string_acen)S:sacl_flags(string_ace1)...
      (string_acen)"
```

Exemple :

```
"%SystemRoot%\system32\ciadv.msc",2,"D:PAR(A;OICI;FA;;;BA)(A;OICI;FA;;;SY)S:PAR(AU;OICI
      FA;CC;;;WD)"
```

Le paramètre **Service General** (Généralités sur le service) comprend les quatre parties suivantes :

- Service Name (Nom du service) : le service qui doit être vérifié.
- Service start type (Type de démarrage du service) : manuel, automatique ou désactivé. La valeur peut être [2-4].
- DACL : DACL est une ACL qui est contrôlée par le propriétaire d'un objet et qui spécifie l'accès particulier que des utilisateurs ou des groupes peuvent avoir à cet objet.
- SACL : SACL est une ACL qui contrôle la création des messages d'audit pour les tentatives d'accès à un objet sécurisable.

Syntaxe :

```
Service Name,Start type,
      "D:dacl_flags(string_ace1)...(string_acen)S:sacl_flags(string_ace1)...(string_a
      cen)"
```

Exemple :

```
kdc,3,"D:AR(A;;CCDCLCSWRPWPDTLOCRSDRCWDWO;;;BA)(A;;CCLCSWLOCRRC;;;AU)(A;;CCLCSWRPWPDTLO
      CRRC;;;SY)"
```

Si les permissions pour un paramètre de service ne doivent pas être contrôlées et que seul le type de démarrage doit être vérifié, vous pouvez procéder comme suit.

Syntaxe :

```
Service Name,Start type
```

Exemple :

```
kdc,3,""
```

Le paramètre **Registry Value** (Valeur du registre) comprend les trois parties suivantes :

- RegistryKey : la clé de registre qui doit être vérifiée.
- RegistryType : le type de registre, REG_DWORD, REG_SZ, etc.
- RegistryValue : la valeur pour la clé de registre.

RegistryValue (Valeur du registre) peut être défini avec des guillemets doubles ou simples ou sans guillemets.

Syntaxe :

```
RegistryKey,RegistryType,RegistryValue
```

Exemple :

```
MACHINE\SYSTEM\CurrentControlSet\Services\Tcpip\Parameters\EnableDeadGWDetect=4,0
```

Si vous souhaitez fournir un commentaire sur une ligne particulière du fichier **.inf**, ajoutez un point-virgule « ; » devant la ligne et le script ignorera cette ligne.

Conversion des fichiers de configuration Unix en fichiers .audit avec c2a

L'outil **c2a.pl** est conçu pour aider les auditeurs à créer des fichiers **.audit** afin de vérifier les configurations des applications sur un réseau donné. Par exemple, si tous les serveurs Web d'un réseau donné doivent être configurés exactement comme l'hôte maître X, exécutez cet outil sur l'hôte X, créez le fichier **.audit** pour **httpd** sur ce système, puis amenez ce fichier dans le démon Nessus et lancez à nouveau le scan sur tous les autres serveurs Web pour vérifier la conformité.

Cet outil peut également être utilisé pour créer des fichiers de vérification MD5 pour un hôte complet. Il attend une liste de fichiers/répertoires à vérifier dans un fichier d'entrée, qui est ensuite traité de façon récurrente, dans le cas de répertoires, afin de créer un fichier **.audit** pour le système. Ce fichier pourra alors être utilisé à une date ultérieure pour des scans afin de rechercher des changements apportés aux fichiers et répertoire principaux.

Obtention et installation de l'outil

L'outil **c2a** est une archive **tar** compressée qui être obtenue sur le portail d'assistance de Tenable, à l'adresse https://support.tenable.com/.

Extrayez le contenu de **c2a-x.x.x.tar.gz** vers l'ordinateur de l'utilisateur à l'aide de la commande suivante :

```
# tar xzf c2a-x.x.x.tar.gz
```

Cela créera un répertoire « **c2a** » sous le répertoire en cours d'utilisation et y extraira les fichiers. Pour extraire le contenu dans un répertoire choisi, utilisez la commande suivante :

```
# tar xzf c2a.x.x.x.tar.gz -C /path/to/directory
```

Une fois l'archive décompressée, les fichiers suivants doivent apparaître dans le répertoire **~/c2a** :

- **c2a.pl**
- **c2a.map**
- **c2a_regex.map**
- **cmv.pl**
- **ReadMe.txt**

Créer un fichier d'audit MD5

Exécutez l'outil de conversion avec l'option « **-md5** » en saisissant :

```
# ./c2a.pl -md5 -f /path/to/inputfile.txt -o outputfile.audit
```

L'outil attend un fichier d'entrée avec une liste de fichiers et de répertoires qui doivent être vérifiés par rapport aux valeurs MD5 ainsi qu'un nom de fichier de sortie pour le fichier d'audit.

Pour ajouter des fichiers au fichier d'entrée, veillez à utiliser ce format :

`/path/to/file`

Utilisez ce format pour ajouter des répertoires :

`/path/to/file/`

Si ce format est utilisé et que le fichier est en fait un fichier réel et non un répertoire, l'outil `c2a` signalera que ce fichier n'existe pas. La barre oblique « / » au début est totalement appropriée pour ajouter les répertoires.

Si l'élément dans le fichier d'entrée est un fichier MD5 normal, seul ce fichier sera calculé et écrit dans le format **.audit**. Dans le cas d'un répertoire, le script explorera de façon récurrente chaque fichier de ce répertoire. Si un fichier de sortie n'est pas spécifié, le résultat sera écrit dans **~/c2a/op.audit**.

Lors du traitement de la liste des fichiers spécifiés par le fichier d'entrée, tout lien symbolique rencontré sera ignoré. Un message d'avertissement s'affiche pour indiquer que le fichier n'existe pas ou qu'il est un lien symbolique. À partir de cette version, **c2a** ne prend pas en charge les liens symboliques.

Créer un fichier d'audit basé sur un ou plusieurs fichiers de configuration

L'outil **c2a** est idéal pour traiter les fichiers de configuration qui ont un contenu ligne par ligne unique. Si le fichier de configuration a une fonctionnalité multiligne, comme un fichier de configuration XML, **c2a** n'est pas l'option idéale.

Lancez l'outil de conversion avec l'option « **-audit** » en saisissant :

```
# ./c2a.pl -audit -f /path/to/input.txt -o outputfile.audit
```

L'outil attend un fichier d'entrée (**input.txt**) contenant une liste de fichiers de configuration qui doivent être vérifiés, ainsi qu'un nom de fichier de sortie pour le fichier d'audit.

Le script Perl **c2a.pl** est basé sur deux fichiers clés : **c2a.map** et **c2a_regex.map**. Il scanne chaque ligne d'un fichier de configuration en cours d'audit et contrôle si le premier mot de cette ligne correspond au « type » dans le fichier **c2a.map** (par exemple HTTP, SENDMAIL, etc.) et à la valeur qui lui est associée. Par exemple, s'il vérifie des paramètres HTTP, il contrôle si le mot correspond à l'un des mots clés HTTP du fichier **c2a.map**. Si c'est le cas, il applique l'expression regex de **c2a_regex.map** pour HTTP à cette ligne et extrait le paramètre et la valeur. Seuls les paramètres pour lesquels une entrée existe dans **c2a.map** seront vérifiés.

Les fichiers de configuration que vous ne souhaitez pas vérifier peuvent être commentés à l'aide du caractère « # ».

Si vous souhaitez convertir au format **.audit** les paramètres signalés par des commentaires dans le fichier de configuration, éditez **c2a.pl** et définissez « `$ENFORCE_COMMENT = 1;` ».

Comme dans le cas précédent, si le fichier de sortie n'est pas précisé, le résultat sera écrit dans **~/c2a/op.audit**.

Pour le moment, Tenable fournit des paramètres MAP pour HTTP, SENDMAIL, SYSCTL et NESSUS. Des paramètres d'applications supplémentaires peuvent facilement être ajoutés en utilisant un script Perl **cmv.pl**. Voir la section suivante pour plus d'informations.

Création d'un fichier MAP

La création d'un fichier MAP pour une application est simple. Il suffit d'exécuter le script **cmv.pl** comme suit :

```
# ./cmv.pl -r 'regex' -r tag -f config_file
```

Où :

- « **regex** » est la regex qui permet d'extraire le paramètre de configuration et la paire de valeurs. Généralement, le format est « <nom> = <valeur> ». Mais il pourrait être légèrement différent dans certains cas et « = » pourrait être remplacé par un espace, une tabulation ou autre.
- « **tag** » est essentiellement le mot clé souhaité pour baliser l'application vérifiée. Le mot clé **tag** lie **config_file** aux mots clés dans **c2a.map** et la regex dans **c2a_regex.map** ; il est donc important que la balise soit la même dans chacun de ces fichiers.
- « **config_file** » est le fichier pour lequel un fichier MAP est créé.

Par exemple, pour vérifier les paramètres de configuration pour VSFTPD, effectuez les étapes suivantes :

1. Utilisez d'abord **cmv.pl** comme suit :

   ```
   # ./cmv.pl -r '([A-Za-z0-9_]+)=([A-Za-z0-9]+)' -t VSFTPD -f /root/vsftpd-0.9.2/vsftpd.conf
   ```

   Cela crée le fichier **tag.map** (par exemple **VSFTPD.map**). Par défaut, toutes les lignes qui ont été commentées sont ignorées. Pour considérer toutes les variables, changez la valeur **$ENFORCE_COMMENT** de « 0 » à « 1 » puis exécutez à nouveau le script.

2. Inspectez le fichier MAP et ajoutez-le à **c2a.map**.

   Recherchez dans le fichier **VSFTPD.map** toute valeur indésirable qui aurait pu correspondre accidentellement à l'expression regex. Après avoir vérifié que tous les mots clés sont corrects, ajoutez-les à **c2a.map**.

3. Mettez à jour **c2a_regex.map** avec la même expression utilisée par **cmv.pl** comme suit :

   ```
   VSFTPD=([A-Za-z0-9_]+)=([A-Za-z0-9]+)
   ```

   Remarque : il s'agit de la même expression regex que celle utilisée par le script Perl **cmv.pl**.

4. Mettez à jour **input.txt** avec l'emplacement du fichier de configuration VSFTPD :

   ```
   VSFTPD=/root/vsftpd-0.9.2/vsftpd.conf
   ```

5. Exécutez le script **c2a.pl** :

   ```
   # ./c2a.pl -audit -f input.txt
   ```

6. Pour finir, vérifiez le fichier de sortie :

   ```
   # vi op.audit
   ```

Autres utilisation de l'outil c2a

Tenable a inclus plusieurs entrées dans les fichiers **c2a.map** et **c2a_regex.map** pour permettre de vérifier Sendmail, VSFTPD (Very Secure FTP Daemon, démon FTP très sécurisé), Apache, le fichier **/etc/sysctl.conf** Red Hat et Nessus. D'autres logiciels pourraient être ajoutés dans un avenir proche. Pour soumettre de nouveaux mappages à Tenable et les partager avec les autres utilisateurs de Nessus, envoyez-les à nessus-support@tenable.com.

Cela étant, le script **c2a.pl** peut être utilisé pour aider à créer des fichiers Nessus **.audit** pour plusieurs applications Unix existantes. Considérez les idées suivantes :

- Si une organisation utilise beaucoup de pare-feu basés sur Unix, un fichier **.audit** peut être créé pour vérifier les paramètres courants et requis que chaque pare-feu est supposé comporter. Par exemple, si tous les pare-feu sont supposés comporter un filtrage des adresses RFC 1918, les règles existantes des pare-feu peuvent être testées.

- Si beaucoup d'applications personnalisées différentes sont exécutées à partir de CRON, les différents CRONTAB peuvent être vérifiés pour s'assurer que les bonnes applications sont exécutées au bon moment.

- Pour les connexions centralisées, les configurations SYSLOG, SYSLOG-NG et LOGROTATE des systèmes Unix distants peuvent faire l'objet d'un contrôle.

Mise au point manuelle des fichiers .audit

Pour finir, la sortie du script **c2a.pl** peut également être éditée manuellement. Par exemple, vous pouvez envisager de combiner les règles de somme de contrôle MD5 avec les règles FILE_CONTENT_CHECK pour former une seule règle. La sortie créée par le script **c2a.pl** suppose aussi qu'un fichier de configuration se trouve toujours à un seul emplacement. Vous pouvez envisager de modifier le mot clé « **file** » pour spécifier d'autres emplacements pouvant contenir un fichier de configuration.

Si vous avez un contenu indésirable dans les configurations à distance, vous pouvez envisager d'ajouter manuellement des contrôles spécifiques avec le mot clé FILE_CONTENT_CHECK_NOT. Ceci peut aider à effectuer des audits pour les paramètres qui devraient être présents et pour ceux qui ne devraient pas être présents.

Conversion des listes de progiciel Unix en fichiers .audit avec p2a

L'outil **p2a.pl** est conçu pour aider les auditeurs à créer des fichiers **.audit** afin d'installer des configurations de progiciel sur les systèmes Linux et Solaris 10 basés sur RPM. Par exemple, s'il est souhaitable que tous les serveurs Internet Linux d'un réseau donné aient la même base RPM que l'hôte maître X, vous exécutez cet outil sur l'hôte X, ce qui crée un fichier **.audit** contenant tous les progiciels RPM sur ce système. Vous utilisez ensuite ce fichier **.audit** avec Nessus pour exécuter un scan des autres serveurs Internet afin de vérifier la conformité.

Cet outil peut éventuellement être utilisé pour créer un fichier d'audit à partir d'une liste textuelle de RPM ou de progiciels Solaris 10. Il attend une liste de progiciels, un par ligne, dans un fichier d'entrée, puis il formate correctement un fichier **.audit** pour le système cible. Le fichier **.audit** généré peut alors être utilisé à une date ultérieure pour des scans afin de rechercher des changements apportés aux progiciels d'installation principaux.

Obtention et installation de l'outil

L'outil **p2a** est une archive **tar** compressée constituée d'un seul script Perl et d'un fichier d'aide **ReadMe.txt**. Il peut être obtenu sur le portail d'assistance de Tenable, à l'adresse https://support.tenable.com/.

Extrayez le contenu de **p2a-x.x.x.tar.gz** sur l'ordinateur de l'utilisateur à l'aide de la commande suivante :

```
# tar xzf p2a-x.x.x.tar.gz
```

Cela créera un répertoire « **p2a** » sous le répertoire utilisé et y extraira les fichiers.

Pour extraire le contenu dans un répertoire choisi, utilisez la commande suivante :

```
# tar xzf p2a.x.x.x.tar.gz -C /path/to/directory
```

Une fois l'archive décompressée, les fichiers suivants doivent apparaître dans le répertoire **~/p2a** :

- **p2a.pl**
- **ReadMe.txt**

Rendez le script exécutable en exécutant :

```
# chmod 750 p2a.pl
```

Utilisation

Exécutez le script comme suit :

```
# ./p2a.pl [-h] -i inputfile.txt -o outputfile.audit
```

« **-h** » est un argument autonome facultatif qui permet d'afficher l'outil d'aide.

Création d'un fichier de sortie basé sur tous les progiciels installés

Si le script est exécuté uniquement avec l'option « **-o** », il exécute une commande système pour extraire tous les noms des progiciels système installés localement et le fichier **.audit** résultant est écrit dans **/path/to/outputfile.audit**.

```
# ./p2a.pl -o /path/to/outputfile.audit
```

Les fichiers de sortie doivent comporter l'extension **.audit** pour que le script soit exécuté. Sinon, une erreur indiquant une extension de fichier incorrecte est générée.

Création d'un fichier de sortie basé sur la liste des progiciels et envoi à l'écran

Exécutez **p2a** pour envoyer l'ensemble des résultats dans la fenêtre du terminal avec la syntaxe suivante :

```
# ./p2a.pl -i /path/to/inputfile.txt
```

Cette option nécessite un fichier d'entrée et crée sur la fenêtre du terminal une sortie (**stdout**) qui peut être copiée et collée dans le fichier **.audit**. Le fichier d'entrée doit être formaté avec un progiciel par ligne, sans ajouter de délimiteur.

Exemple :

```
mktemp-1.5-23.2.2
libattr-2.4.32-1.1
libIDL-0.8.7-1.fc6
pcsc-lite-libs-1.3.1-7
zip-2.31-1.2.2
```

Puisque les systèmes Unix peuvent souvent comporter plus d'un millier de progiciels, les résultats peuvent excéder le tampon de défilement et être difficiles à visualiser.

Création un fichier d'audit basé sur un fichier d'entrée spécifié

L'exécution de **p2a** avec des arguments d'entrée et de sortie utilise la liste de progiciels formatée et crée un fichier **.audit** à l'emplacement précisé.

```
# ./p2a.pl -i /path/to/input_file.txt -o /path/to/outputfile.audit
```

Les fichiers d'entrée doivent être formatés avec un progiciel par ligne, sans ajouter de délimiteur.

Exemple :

```
mktemp-1.5-23.2.2
libattr-2.4.32-1.1
libIDL-0.8.7-1.fc6
pcsc-lite-libs-1.3.1-7
zip-2.31-1.2.2
```

Les fichiers de sortie doivent comporter l'extension **.audit** pour que le script soit exécuté. Sinon, une erreur indiquant une extension de fichier incorrecte est générée.

Exemple d'utilisation de l'interface utilisateur Nessus

Obtention des contrôles de conformité

Les clients commerciaux disposent déjà des contrôles de conformité pour leur scanner Nessus et plusieurs fichiers **.audit** sont disponibles sur le portail d'assistance de Tenable, à l'adresse https://support.tenable.com/. Pour le confirmer, lancez l'interface utilisateur Nessus, identifiez-vous et gérez ou éditez une stratégie existante. Sous l'onglet « Plugins », recherchez la famille « Policy Compliance », cliquez sur le nom de la famille de plugin et confirmez que les plugins sont affichés :

- Cisco IOS Compliance Checks (Contrôles de conformité Cisco IOS)
- Database Compliance Checks (Contrôles de conformité des bases de données)
- IBM iSeries Compliance Checks (Contrôles de conformité IBM iSeries)
- PCI DSS Compliance (Conformité PCI DSS)
- PCI DSS Compliance: Database Reachable from the Internet (Conformité PCI DSS : base de données accessible par Internet)
- PCI DSS Compliance: Handling False Positives (Conformité PCI DSS : traitement des faux positifs)

- PCI DSS Compliance: Insecure Communication Has Been Detected (Conformité PCI DSS : communications non sécurisées détectées)
- PCI DSS Compliance: Remote Access Software Has Been Detected (Conformité PCI DSS : logiciel d'accès à distance détecté)
- PCI DSS Compliance: Passed (Conformité PCI DSS : réussite)
- PCI DSS Compliance: Tests Requirements (Conformité PCI DSS : exigences de tests)
- Unix Compliance Checks (Contrôles de conformité Unix)
- Windows Compliance Checks (Contrôles de conformité Windows)
- Windows File Contents Compliance Checks (Contrôles de conformité du contenu des fichiers Windows)

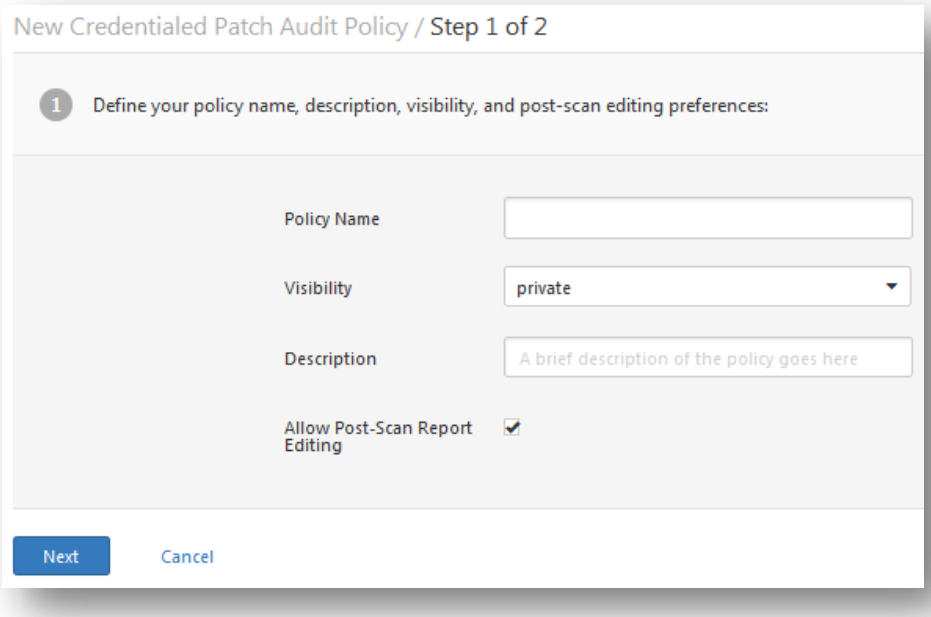
Configuration d'une stratégie de scan

Pour autoriser les contrôles de conformité dans Nessus, une stratégie de scan doit être créée avec les attributs suivants :

- Activer les plugins de contrôle de conformité qui sont dans la famille de plugin « Policy Compliance » (Conformité des stratégies)
- Préciser une ou plusieurs stratégies de conformité `.audit` comme préférence
- Préciser les identifiants permettant d'accéder au serveur cible y compris les identifiants de base de données sous l'onglet « Preferences » (Préférences) le cas échéant
- Activer les dépendances de plugin

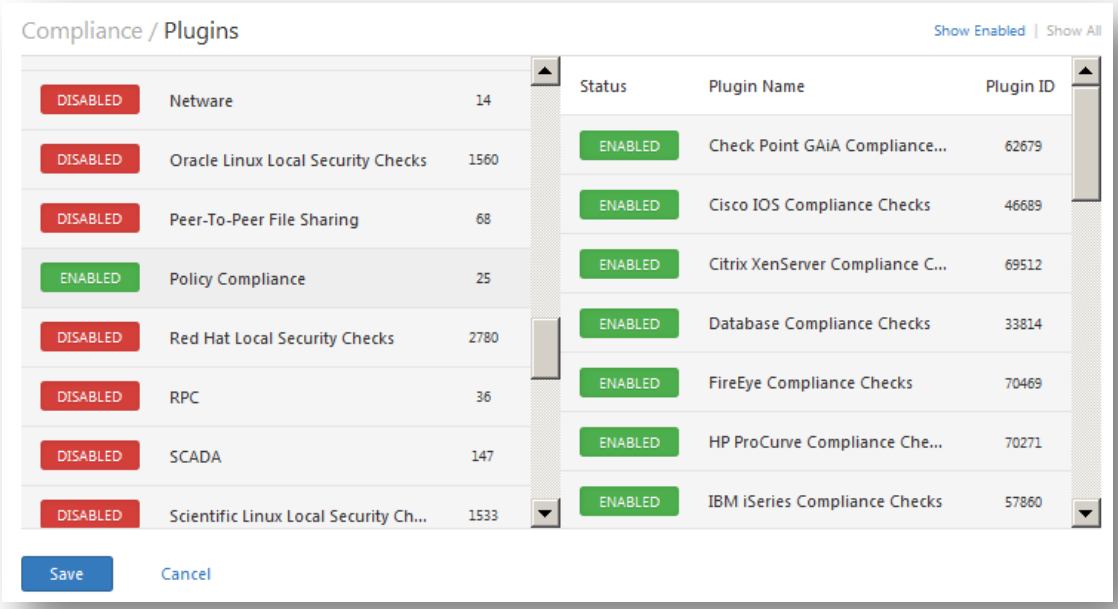
Cette opération peut être exécutée dans le Policy Wizard (Assistant de stratégies) en sélectionnant l'assistant « **Credentialed Patch Audit** » (Audit des correctifs avec identifiants) ou manuellement via l'option « **Advanced Policy** » (Stratégie avancée).

Il faut bien comprendre les contrôles dans les fichiers `.audit` sélectionnés, en particulier lorsque des fichiers personnalisés ont été créés. Lorsque vous utilisez deux fichiers `.audit` sur le même scan, les deux fichiers sont combinés pour produire les résultats de chaque fichier dans un seul scan. S'il existe des résultats incompatibles entre les fichiers, vous pourriez recevoir un résultat de réussite et un résultat d'échec. Assurez-vous de toujours vérifier les résultats dans les rapports.



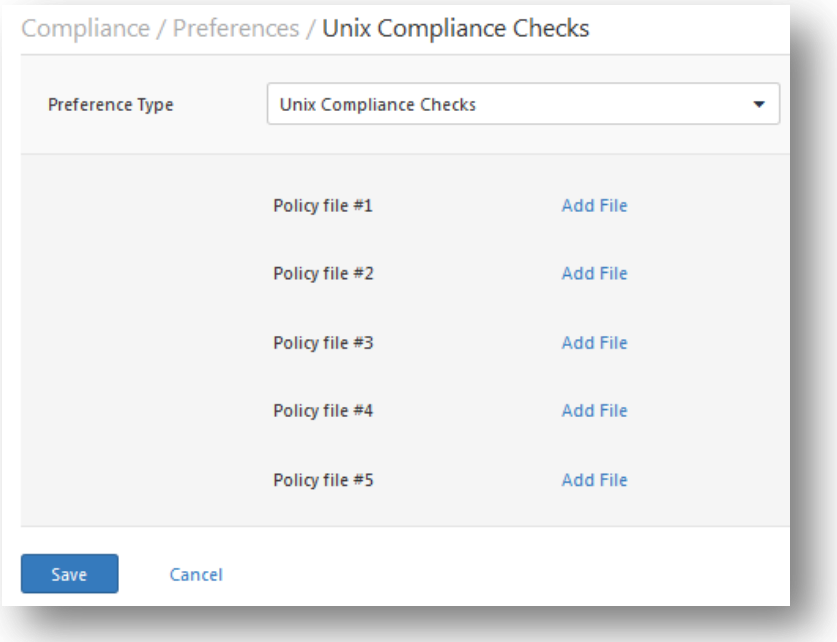
Pour créer une stratégie de scan, accédez à l'interface utilisateur Nessus, identifiez-vous et sélectionnez « Policies » (Stratégies). Éditez une stratégie existante ou créez-en une nouvelle. Vous pouvez préciser les identifiants pour accéder au serveur cible sous l'onglet « **Credentials** » (Identifiants) à gauche.

Sous l'onglet « **Plugins** » (Plugins), activez la famille de plugins « Policy Compliance » (Conformité des stratégies) et assurez-vous que « **`auto_enable_dependencies`** » est paramétré sur « **yes** » (oui) dans Advanced Settings (Paramètres avancés) ; il s'agit du paramètre par défaut :



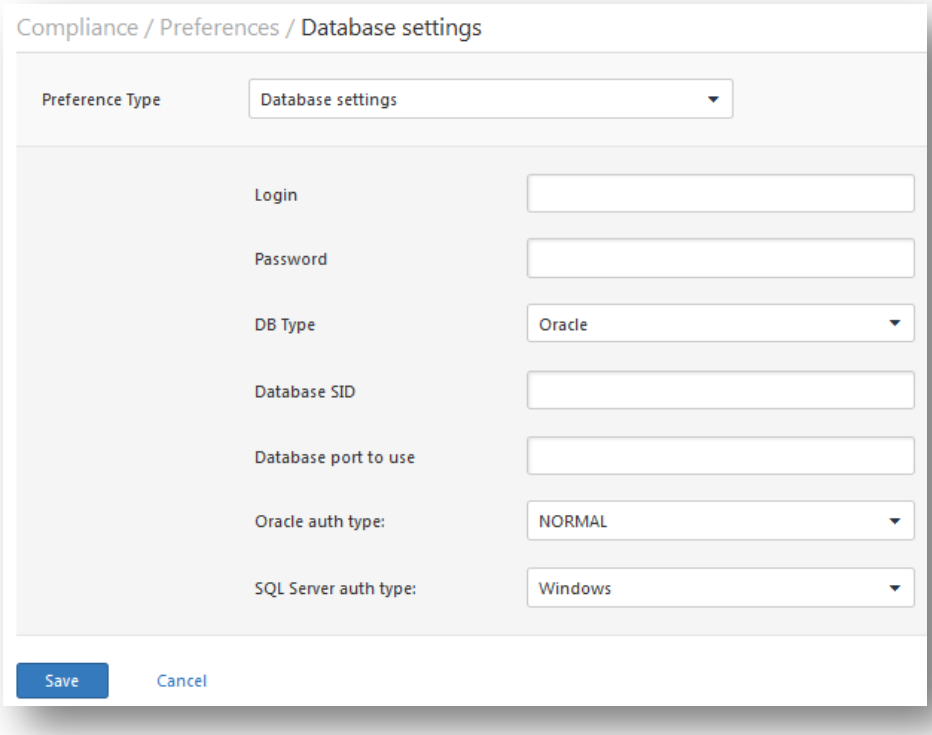
Édition d'une politique de scan pour déterminer si Policy Compliance (Conformité des stratégies) est disponible

Pour permettre l'utilisation d'un fichier **`.audit`**, sous l'onglet « **Preferences** » sélectionnez « Cisco IOS Compliance Checks » (Contrôles de conformité Cisco IOS), « Unix Compliance Checks » (Contrôles de conformité Unix), « Windows Compliance Checks » (Contrôles de conformité Windows), « Windows File Content Compliance Checks » (Contrôles de conformité du contenu des fichiers Windows), « IBM iSeries Compliance Checks » (Contrôles de conformité IBM iSeries) ou « Database Compliance Checks » (Contrôles de conformité des bases de données) dans le menu déroulant. Il y aura cinq champs pour chaque section pouvant spécifier des fichiers **`.audit`** séparés. Les fichiers spécifiés auront été précédemment téléchargés sur le système client local à partir du portail d'assistance Tenable.



Exemple de boîte de dialogue de l'interface utilisateur Nessus servant à spécifier les fichiers .audit Unix

Si « Database Compliance Checks » (Contrôles de conformité des bases de données) était sélectionné dans le menu déroulant précédent, les paramètres de connexion pour la base de données doivent être saisis sous « **Preferences** » -> « **Database Settings** » (Préférences -> Paramètres de base de données) :



Un certain nombre d'options sont disponibles sous « Database Settings » (Paramètres de base de données), notamment :

Option	Description
Login (Connexion)	Nom d'utilisateur pour la base de données.
Password (Mot de passe)	Mot de passe correspondant au nom d'utilisateur fourni.
DB Type (Type DB)	Oracle, SQL Server, MySQL, DB2, Informix/DRDA et PostgreSQL sont pris en charge.
Database SID (ID de système de la base de données)	ID système de la base de données à vérifier. Applicable à Oracle, DB2 et Informix uniquement.
Oracle auth type (Type d'auth. Oracle)	NORMAL, SYSOPER et SYSDBA sont pris en charge.
SQL Server auth type (Type d'auth. de serveur SQL)	Windows ou SQL Server sont pris en charge.

Consultez l'administrateur de base de données locale afin d'obtenir les valeurs correctes pour ces champs.

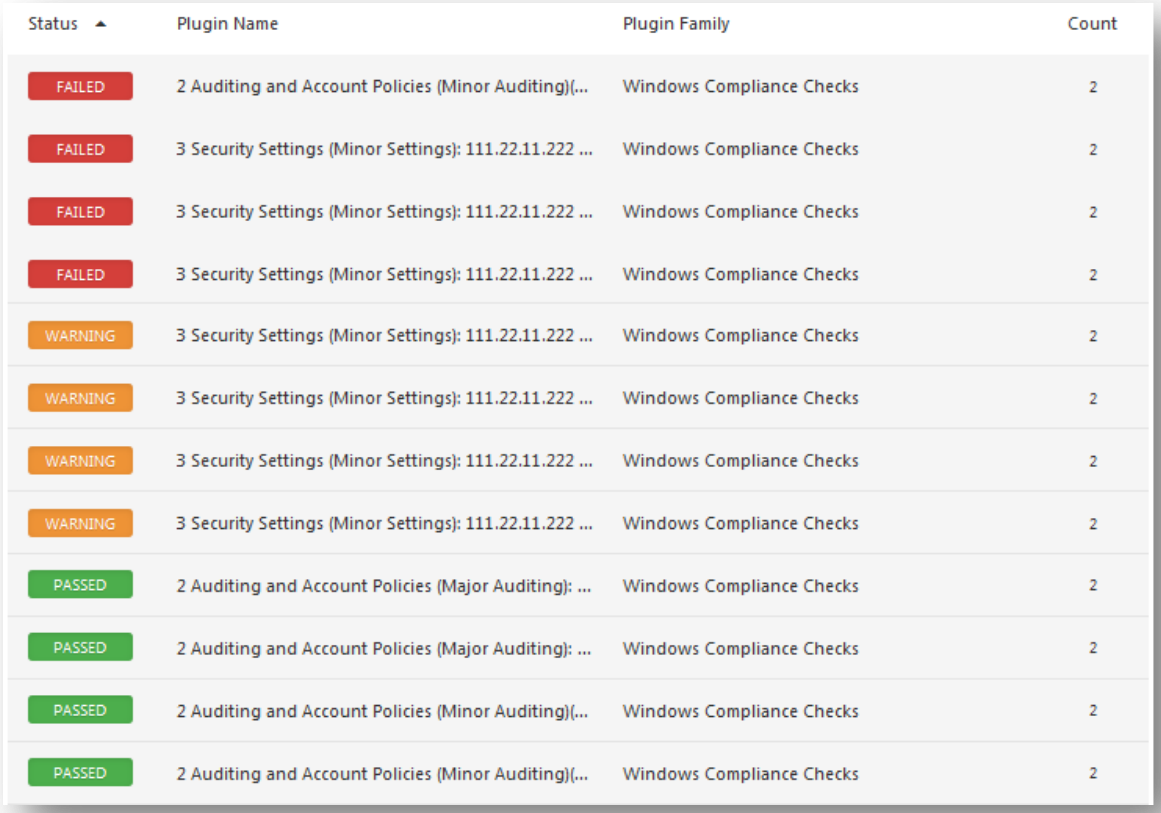
Vous pouvez alors cliquer sur « Save » (Enregistrer) au bas de la fenêtre pour terminer la configuration. La nouvelle stratégie de scan sera ajoutée à la liste des stratégies de scan gérées.

Lancement d'un scan

Le lancement d'un scan avec des contrôles de conformité activés est identique au lancement d'autres scans d'audit de correctif local ou même à celui de scans normaux de réseau. En fait, ceux-ci peuvent être mélangés et combinés pour être tous exécutés en même temps, si nécessaire.

Exemples de résultats

Dans Nessus, tous les résultats de conformité sont retournés avec l'ID de plugin effectuant le test. Dans l'exemple ci-dessous, toutes les données qui sont retournées pour un serveur Windows scanné proviendront du plugin de conformité `.nbin` Windows identifié comme le plugin 21156.



Status	Plugin Name	Plugin Family	Count
FAILED	2 Auditing and Account Policies (Minor Auditing)(...	Windows Compliance Checks	2
FAILED	3 Security Settings (Minor Settings): 111.22.11.222 ...	Windows Compliance Checks	2
FAILED	3 Security Settings (Minor Settings): 111.22.11.222 ...	Windows Compliance Checks	2
FAILED	3 Security Settings (Minor Settings): 111.22.11.222 ...	Windows Compliance Checks	2
WARNING	3 Security Settings (Minor Settings): 111.22.11.222 ...	Windows Compliance Checks	2
WARNING	3 Security Settings (Minor Settings): 111.22.11.222 ...	Windows Compliance Checks	2
WARNING	3 Security Settings (Minor Settings): 111.22.11.222 ...	Windows Compliance Checks	2
WARNING	3 Security Settings (Minor Settings): 111.22.11.222 ...	Windows Compliance Checks	2
PASSED	2 Auditing and Account Policies (Major Auditing): ...	Windows Compliance Checks	2
PASSED	2 Auditing and Account Policies (Major Auditing): ...	Windows Compliance Checks	2
PASSED	2 Auditing and Account Policies (Minor Auditing)(...	Windows Compliance Checks	2
PASSED	2 Auditing and Account Policies (Minor Auditing)(...	Windows Compliance Checks	2

Exemple de résultats de conformité pendant le scan d'un serveur Windows

Le compte-rendu HTML, qui peut être téléchargé à partir de l'onglet « Reports » (Rapports) dans l'interface utilisateur de Nessus, montre les tests de conformité réussis en bleu avec le message « PASSED » (SUCCÈS). Il montre également ceux qui ont échoué en rouge avec le message « FAILED » (ÉCHEC). Tous les éléments qui n'ont pas pu être vérifiés apparaissent en jaune avec le message « WARNING » (AVERTISSEMENT).

Dans l'exemple ci-dessus, seuls quatre éléments sont affichés. Chacun de ces éléments provient d'une stratégie de contrôle d'accès vérifiant la présence de services et de protocoles non nécessaires et non sécurisés. Certains de ces services n'ont pas été exécutés et ils ont satisfait aux attentes de la stratégie **`.audit`** tandis que d'autres (comme le service « remote registry » (registre à distance)) ont été exécutés et répertoriés comme « FAILED » (ÉCHEC). Il est vivement recommandé que les éléments répertoriés comme « FAILED » soient configurés pour respecter la stratégie conformément aux normes de sécurité.

Exemple d'utilisation de Nessus pour les lignes de commande Unix

Obtention des contrôles de conformité

Si l'installation Nessus commerciale a été configurée, il y aura cinq fichiers de conformité **`.nbin`** dans le répertoire des plugins.

Tout fichier **`.audit`** nécessaire peut être téléchargé à partir du portail d'assistance de Tenable à l'adresse https://support.tenable.com/ et placé dans le répertoire des plugins du scanner. Sur la plupart des distributions, l'emplacement par défaut est le répertoire suivant :

`/opt/nessus/lib/nessus/plugins`

Ces plugins figureront parmi les 40 000 fichiers de plugins **.nasl** utilisés par Nessus pour effectuer des scans des vulnérabilités. Ils peuvent être recherchés sur la base de l'extension **.nbin** comme illustré ci-dessous :

```
# ls compliance*nbin database*nbin unix*nbin cisco_compliance*nbin

cisco_compliance_check.nbin                 database_compliance_check.nbin
compliance_check.nbin                       unix_compliance_check.nbin
compliance_check_windows_file_content.nbin
```

D'autres fichiers **.nbin** qui n'ont rien à voir avec les contrôles de conformité peuvent être délivrés par Tenable, comme le plugin Skype.

Si vous n'avez pas l'accès local au démon Nessus mais que vous possédez un nom d'utilisateur et un mot de passe pour vous connecter au serveur, vous pouvez demander une liste de plugins en utilisant l'option « **-p** » du client de ligne de commande **nessus** comme indiqué ci-dessous :

```
# /opt/nessus/bin/nessus -xp 192.168.20.1 1241 username password | grep 21156
*** The plugins that have the ability to crash remote services or hosts
have been disabled. You should activate them if you want your security
audit to be complete
21156|Policy Compliance|Checks if the remote system is compliant with the
      policy|infos|This script is Copyright (C) 2006 Tenable Network Security|Check
      compliance policy|$Revision: 1.3 $|NOCVE|NOBID|NOXREF|\nSynopsis :\n\n
      Compliance checks\n\nDescription :\n\nUsing the supplied credentials this
      script perform a compliance\ncheck against the given policy.\n\nRisk factor
      :\n\nNone
```

L'exécution de l'interrogation peut nécessiter quelques minutes. Si l'interrogation est exécutée avec succès mais ne renvoie aucune donnée, les contrôles de conformité ne sont pas installés sur le scanner Nessus distant.

Utilisation des fichiers .nessus

Nessus peut sauvegarder des stratégies de scan configurées, des cibles de réseau et des rapports sous forme de fichiers **.nessus**. La section « Exemple d'utilisation de l'interface utilisateur Nessus » explique la création d'un fichier **.nessus** contenant une stratégie de scan pour les contrôles de conformité. Pour plus d'instructions concernant l'exécution d'un scan de ligne de commande à l'aide du fichier **.nessus**, voir le « Nessus User Guide » (Guide de l'utilisateur de Nessus) disponible sur : http://www.tenable.com/products/nessus/documentation.

Utilisation des fichiers .nessusrc

Le client de ligne de commande Nessus peut également exporter des stratégies de scan configurées sous forme de fichiers **.nessusrc**. Ceci peut s'avérer pratique pour faciliter l'activation du scan de ligne de commande. La section « Exemple d'utilisation de l'interface utilisateur Nessus » décrit les étapes à suivre afin de créer une stratégie de scan pour les contrôles de conformité dans Nessus.

Pour invoquer un scan de ligne de commande avec Nessus, il faut préciser les éléments suivants :

- Plugins de contrôle de conformité Unix, Windows ou base de données
- Identifiants pour le ou les hôtes cibles scannés
- Un ou plusieurs fichiers **.audit** pour les plugins de contrôle de conformité à exécuter
- Vérifier que les dépendances ont été activées

Les éléments pertinents d'un fichier **.nessusrc** ont le format suivant (une partie du contenu a été omise) :

```
begin(SERVER_PREFS)
…
auto_enable_dependencies = yes
…
end(SERVER_PREFS)
begin(PLUGINS_PREFS)
…
Compliance policy file(s) : = federal_nsa_microsoft_xp_file_permissions.audit
…
end(PLUGINS_PREFS)
begin(PLUGIN_SET)
 21156 = yes
 21157 = yes
…
End(PLUGIN_SET)
```

L'exemple précédent a omis beaucoup d'autres données qui spécifient comment un scan peut être effectué. Le contenu omis inclut l'activation du fichier de stratégie **.audit** utilisé, l'activation des dépendances et les plugins de conformité.

Lancement d'un scan

Le lancement d'un scan avec des contrôles de conformité activés est identique au lancement d'autres scans d'audit de correctif local ou même à celui de scans normaux de réseau. En fait, ceux-ci peuvent être mélangés et combinés pour être tous exécutés en même temps, si nécessaire.

Exemples de résultats

Comme pour les clients de l'interface graphique, tous les résultats conformes ou non conformes décelés sont signalés au format suivant :

```
192.168.20.16|unknown (0/tcp)|21156|Security Hole|"Reset lockout account counter
      after" : [FAILED]\n\nRemote value: 30\nPolicy value: 20\n\n\n
192.168.20.16|unknown (0/tcp)|21156|Security Hole|"Minimum password length" :
      [FAILED]\n\nRemote value: 0\nPolicy value: 8\n\n\n
192.168.20.16|unknown (0/tcp)|21156|Security Hole|"Minimum password age" :
      [FAILED]\n\nRemote value: 0\nPolicy value: 1\n\n\n
192.168.20.16|unknown (0/tcp)|21156|Security Hole|"Maximum password age" :
      [FAILED]\n\nRemote value: 42\nPolicy value: 182\n\n\n
192.168.20.16|unknown (0/tcp)|21156|Security Hole|"Enforce password history" :
      [FAILED]\n\nRemote value: 0\nPolicy value: 5\n\n\n
192.168.20.16|unknown (0/tcp)|21156|Security Hole|"Account lockout threshold" :
      [FAILED]\n\nRemote value: 0\nPolicy value: 3\n\n\n
192.168.20.16|unknown (0/tcp)|21156|Security Hole|"Account lockout duration" :
      [FAILED]\n\nRemote value: 30\nPolicy value: 60\n\n\n
```

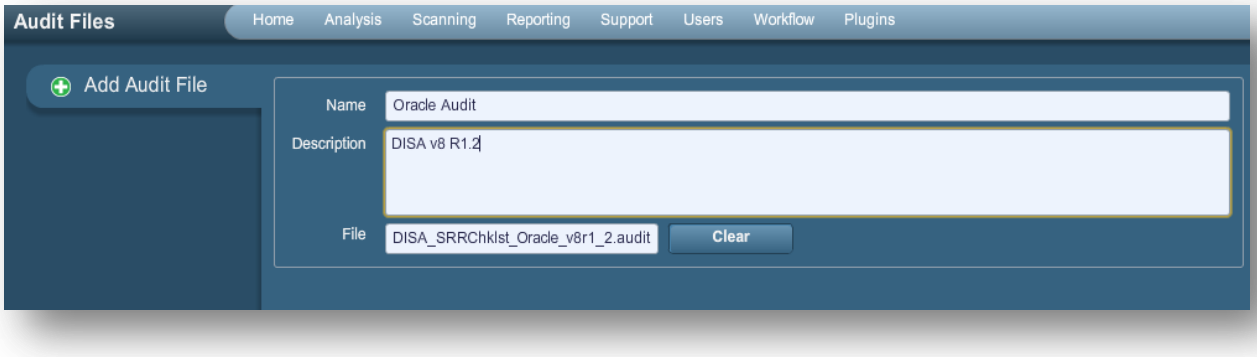
Ces données utilisent le format de rapports **.nsr** pour Nessus. Ce sont tous des événements non conformes.

Utilisation de SecurityCenter

Les informations ci-dessous sont basées sur l'exécution de scans de conformité avec SecurityCenter 4 ou une version plus récente. Les utilisateurs de Security Center 3.x doivent se reporter au document « Security Center 3.4 Documentation » (Documentation sur Security Center 3.4) disponible sur le Tenable Support Portal (Portail d'assistance de Tenable) : https://support.tenable.com/.

Obtention des contrôles de conformité

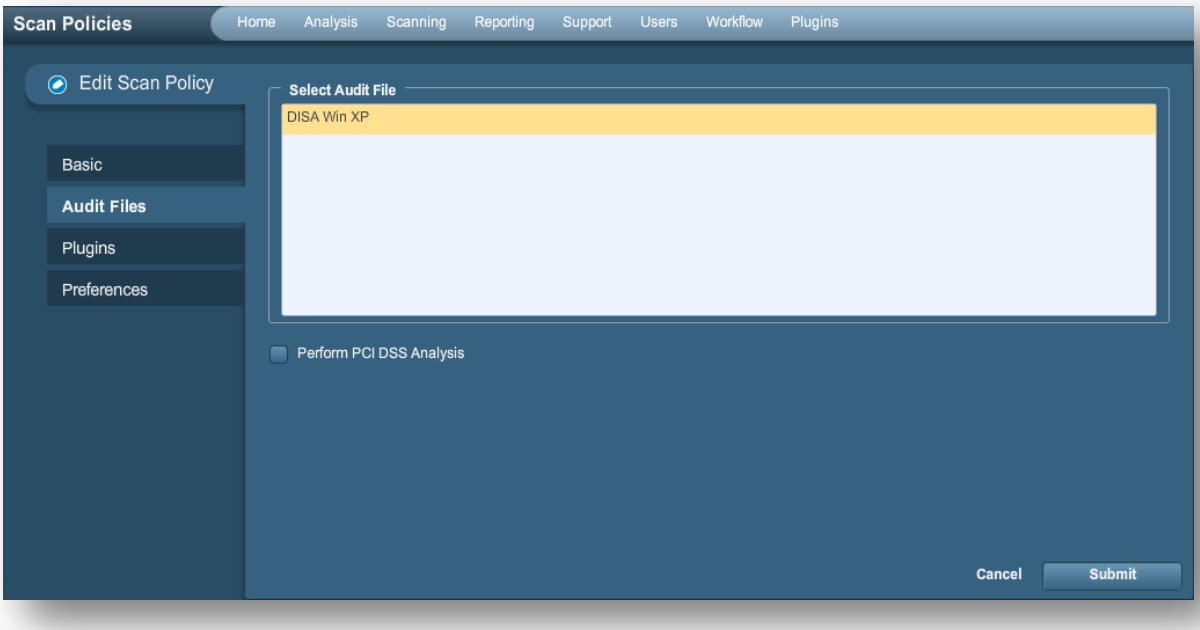
Tous les clients de SecurityCenter ont accès aux plugins commerciaux Nessus. Ceci inclut les plugins de contrôle de conformité Cisco, IBM iSeries, Unix, Windows, Windows File Contents et Database. Ces plugins permettent à l'utilisateur de télécharger et d'exécuter des scans de conformité utilisant des fichiers `.audit` prédéfinis et personnalisables fournis par Tenable. Tous les fichiers `.audit` requis sont disponibles sur le portail d'assistance de Tenable, à https://support.tenable.com/. Ces fichiers `.audit` peuvent être téléchargés vers SecurityCenter par tout utilisateur possédant la permission « Create Audit Files » (Créer des fichiers d'audit) en utilisant l'outil « Add Audit File » (Ajouter fichier d'audit) sous l'onglet « Support ».



Tout fichier `.audit` téléchargé vers SecurityCenter sera mis à la disposition de tout utilisateur de SecurityCenter possédant la permission « Create Policies » (Créer des stratégies). SecurityCenter gérera aussi la distribution des fichiers `.audit` nouveaux et mis à jour vers les scanners Nessus.

Configuration d'une stratégie de scan pour effectuer un audit de conformité

Pour effectuer un scan de conformité avec SecurityCenter, les utilisateurs doivent configurer une stratégie de scan avec les paramètres de conformité appropriés. Cette stratégie permet de préciser les options de scan, les fichiers d'audit, les plugins activés et les préférences avancées. La deuxième page de « Scan Policy » (Stratégie de scan) précise les fichiers `.audit` à utiliser pour l'audit de conformité.



Un ou plusieurs fichiers `.audit` peuvent être sélectionnés ici en mettant en surbrillance le fichier `.audit` et en cliquant sur « Submit » (Soumettre). Si vous voulez sélectionner plusieurs fichiers `.audit`, utilisez la touche « Ctrl » pour activer la sélection multiple. Si une analyse PCI DSS de base est requise, assurez-vous que la case « Perform PCI DSS Analysis » (Effectuer l'analyse PCI DSS) est cochée avant l'envoi.

PCI DSS (Payment Card Industry Data Security Standard, normes de sécurité des données de l'industrie des cartes de paiement) est un ensemble complet de normes de sécurité établies par les membres fondateurs du Conseil des normes de sécurité PCI, y compris Visa, American Express, Discover Financial Services et MasterCard. PCI DSS est conçu pour fournir une base commune afin de protéger les données sensibles des titulaires de carte pour toutes les marques de carte bancaire, et il est utilisé par un grand nombre de sites d'e-commerce qui acceptent et mémorisent les coordonnées des cartes bancaires.

Tenable fournit à tous les utilisateurs de SecurityCenter douze plugins qui automatisent le processus d'exécution des vérifications PCI DSS. Le tableau ci-dessous fournit la liste de ces plugins.

Ces plugins évaluent les résultats d'un scan et la configuration existante du scan pour déterminer si le serveur cible satisfait aux exigences de conformité PCI publiées. Les plugins n'effectuent pas le scan, ils examinent les résultats d'autres plugins. Pour activer les plugins PCI DSS, cochez simplement la case « Perform PCI DSS Analysis » (Effectuer l'analyse PCI DSS) dans l'écran « Compliance » (Conformité).

Après avoir sélectionné le ou les fichiers `.audit` souhaités et les paramètres PCI DSS, cliquez sur l'onglet « Plugins » pour confirmer les paramètres de plugin. Les éléments de la famille de plugins « Policy Compliance » (Conformité des stratégies) doivent être activés dans la stratégie pour pouvoir effectuer un scan de conformité.

Lorsque l'utilisateur sélectionne un ou plusieurs fichiers d'audit sous l'onglet « Audit Files » (Fichiers d'audit) de la stratégie de scan, le plugin correct est automatiquement activé sous l'onglet « Plugins ». SecurityCenter analyse le ou les fichiers `.audit` sélectionnés et, selon le type spécifié dans le fichier, le ou les bons plugins sont activés.

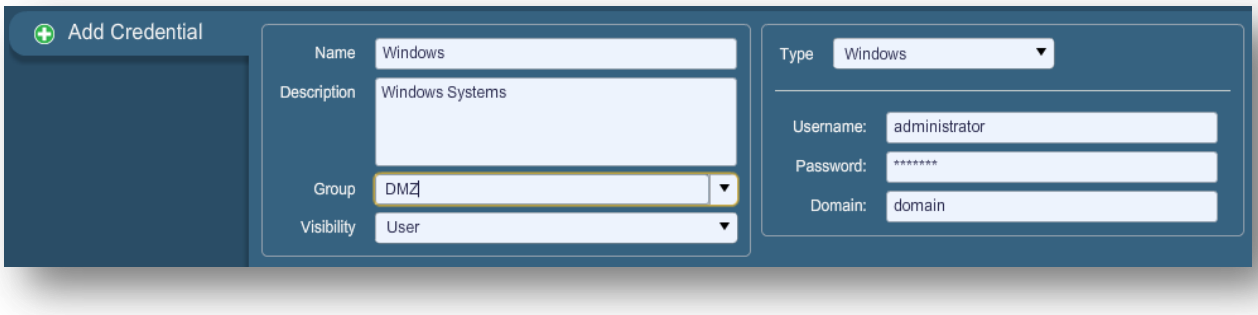
La famille « Policy Compliance » (Conformité des stratégies) propose treize plugins pour les audits de conformité, à savoir :

Plugin ID (ID du plugin)	Plugin Name (Nom du plugin)	Plugin Description (Description du plugin)
21156	Windows Compliance Checks (Contrôles de conformité Windows)	Sert à vérifier les paramètres de configuration Windows courants.
21157	Unix Compliance Checks (Contrôles de conformité Unix)	Sert à vérifier les paramètres de configuration Unix courants.
24760	Windows File Contents Compliance Checks (Contrôles de conformité du contenu des fichiers Windows)	Sert à vérifier le contenu sensible des fichiers sur les serveurs Windows.
33814	Database Compliance Checks (Contrôles de conformité des bases de données)	Sert à vérifier les paramètres de configuration de bases de données courants.
33929	PCI DSS compliance (Conformité PCI DSS)	Détermine si le serveur Internet distant est vulnérable aux attaques de script de site croisé (XSS), s'il met en œuvre l'ancienne cryptographie SSL2.0, s'il exécute un logiciel obsolète ou s'il est affecté par des vulnérabilités dangereuses (note de base CVSS >= 4).
57581	PCI DSS Compliance: Database Reachable from the Internet	Détecte la présence d'une base de données accessible par Internet, ce qui entraîne l'échec de l'audit de conformité.

	(Conformité PCI DSS : base de données accessible par Internet)	
60020	PCI DSS Compliance: Handling False Positives (Conformité PCI DSS : traitement des faux positifs)	Consigne le traitement approprié des faux positifs dans les scans PCI DSS.
56208	PCI DSS Compliance: Insecure Communication Has Been Detected (Conformité PCI DSS : communications non sécurisées détectées)	Détermine si un port, protocole ou service non sécurisé a été détecté, ce qui entraînerait un échec de la conformité.
56209	PCI DSS Compliance: Remote Access Software Has Been Detected (Conformité PCI DSS : logiciel d'accès à distance détecté)	Détecte la présence d'un logiciel d'accès à distance, ce qui entraînerait un échec de la conformité.
33930	PCI DSS Compliance: Passed (Conformité PCI DSS : réussite)	En utilisant les informations disponibles sur le scan, Nessus n'a pas trouvé de défaut disqualifiant pour cet hôte.
33931	PCI DSS Compliance: Tests Requirements (Conformité PCI DSS : exigences de tests)	Analyse si le scan Nessus satisfait aux exigences de test PCI. Même si les tests techniques ont réussi, ce rapport peut être insuffisant pour certifier le serveur.
46689	Cisco IOS Compliance Checks (Contrôles de conformité Cisco IOS)	Sert à vérifier les paramètres de configuration courants des dispositifs Cisco.
57860	IBM iSeries Compliance Checks (Contrôles de conformité IBM iSeries)	Sert à vérifier les paramètres de configuration IBM iSeries courants.

Gestion des identifiants

L'un des avantages de SecurityCenter lors de l'exécution des scans basés sur authentification est le fait qu'il facilite la gestion des identifiants utilisés. Les identifiants sont créés dans SecurityCenter en sélectionnant l'onglet « Support » (Support), en cliquant sur « Credentials » (Identifiants), puis en cliquant sur « Add » (Ajouter).



Les identifiants Unix, Windows et Cisco sont mémorisés et gérés indépendamment de la stratégie de scan. Des identifiants peuvent être créés avec visibilité « User » (Utilisateur) pour l'utilisateur existant ou visibilité « Organizational » (Organisationnel) lorsqu'ils peuvent être utilisés par d'autres utilisateurs de SecurityCenter. Ceci permet aux utilisateurs d'utiliser les résultats des scans pour effectuer de nouveaux scans sans avoir réellement besoin de connaître les identifiants associé au scan.

Des identifiants supplémentaires sont nécessaires pour scanner les systèmes de base de données. Ces identifiants sont mémorisés dans la stratégie de scan et configurés grâce aux « Database settings » (Paramètres de base de données)

(plugin 33815) dans les préférences de la politique de scan. Ces identifiants sont configurés indépendamment des identifiants mentionnés au paragraphe précédent.

Analyse des résultats

SecurityCenter peut être utilisé de plusieurs manières pour analyser et signaler les données de conformité résultant des scans Nessus. Les rapports courants incluent :

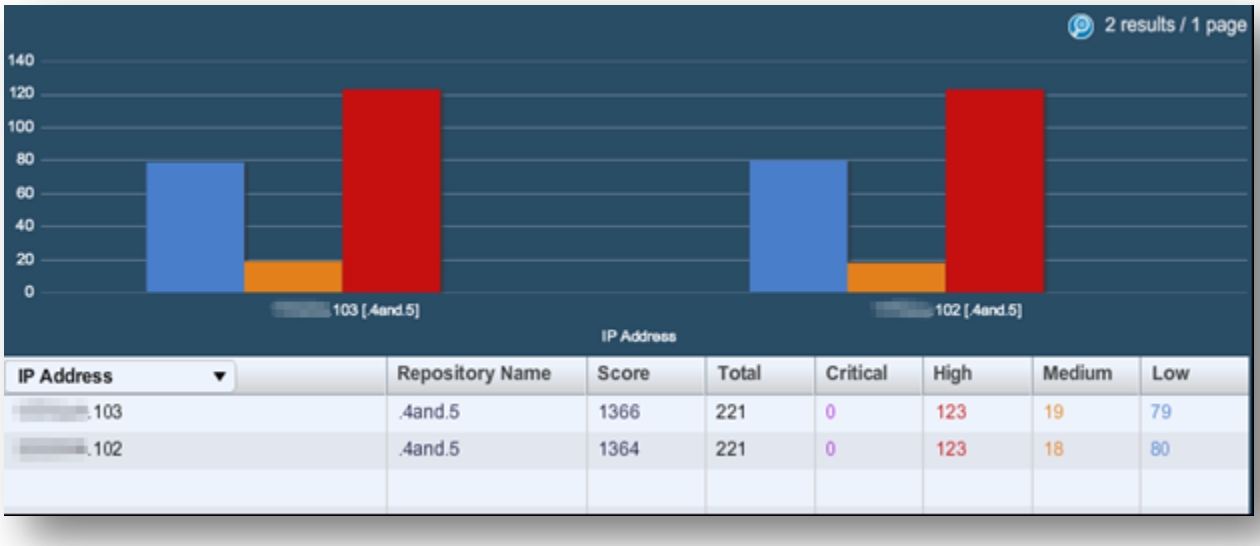
- Liste de toutes les vulnérabilités conformes ou non conformes par groupe d'actif
- Liste de toutes les vulnérabilités conformes ou non conformes par hôte ou réseau
- Récapitulatif de tous les problèmes non conformes
- Vérification des paramètres des bases de données pour les configurations défectueuses courantes
- Rapport concernant l'état des utilisateurs ou des logiciels en fonction des besoins informatiques

Une fois que les données de conformité ont été détectées par SecurityCenter, les outils de signalisation, de rapport et d'analyse peuvent être utilisés afin de déterminer les mesures à prendre pour reconfigurer les appareils vérifiés. Ces données peuvent être analysées en parallèle avec d'autres vulnérabilités, correctifs de sécurité ou informations découvertes passivement.

Voici quelques exemples de captures d'écran où SecurityCenter est utilisé pour analyser les informations de conformité concernant les hôtes scannés :

Plugin ID	Total ▼	Severity	Name
1000282	4	Low	HKLM\\software\\microsoft\\windows nt\\currentversion\\winlogon\\allocatedasd
1000295	4	Medium	HKLM\\Software\\Microsoft\\Windows NT\\CurrentVersion\\Winlog\\AutoAdminLogon
1000294	4	Low	HKLM\\System\\CurrentControlSet\\Control\\SecurePipeServers\\Winreg\\AllowedPaths\\Machine
1000293	4	Low	HKLM\\System\\CurrentControlSet\\Services\\LanManServer\\Parameters\\NullSessionPipes
1000292	4	Low	HKLM\\System\\CurrentControlSet\\Services\\LanManServer\\Parameters\\NullSessionShares
1000291	4	Medium	HKLM\\Software\\Policies\\Microsoft\\Cryptography\\ForceKeyProtection
1000290	4	Low	HKLM\\System\\CurrentControlSet\\Control\\Lsa\\ForceGuest
1000289	4	Low	HKLM\\System\\CurrentControlSet\\Control\\Lsa\\LimitBlankPasswordUse
1000288	4	High	HKLM\\System\\CurrentControlSet\\Control\\Lsa\\MSV1_0\\NTLMMinClientSec
1000287	4	High	HKLM\\System\\CurrentControlSet\\Control\\Lsa\\MSV1_0\\NTLMMinServerSec
1000286	4	Low	HKLM\\System\\CurrentControlSet\\Control\\Lsa\\NoDefaultAdminOwner
1000285	4	Low	HKLM\\System\\CurrentControlSet\\Services\\LDAP\\LDAPClientIntegrity
1000284	4	Low	HKLM\\Software\\Microsoft\\Driver Signing\\Policy
1000283	4	High	HKLM\\software\\microsoft\\non-driver signing\\policy
1000296	4	Low	HKLM\\System\\CurrentControlSet\\Control\\FileSystem\\NtfsDisable8dot3NameCreation
1000281	4	High	HKLM\\software\\microsoft\\windows nt\\currentversion\\winlogon\\scremoveoption
1000280	4	High	HKLM\\system\\currentcontrolset\\control\\lsa\\lmcompatibilitylevel
1000279	4	High	HKLM\\system\\currentcontrolset\\control\\print\\providers\\lanman print services\\servers\\addprinterdrive
1000278	4	Medium	HKLM\\Software\\Microsoft\\Windows NT\\CurrentVersion\\Winlogon\\AutoAdminLogon
1000277	4	Medium	HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\NetworkNoDialIn
1000276	4	Medium	HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\NetworkHideSharePwds
1000275	4	Medium	HKLM\\Software\\Microsoft\\Windows\\CurrentVersion\\Policies\\Explorer\\NoDriveTypeAutoRun
1000274	4	Medium	HKLM\\system\\CurrentControlSet\\Services\\Tcpip\\Parameters\\PerformRouterDiscovery
1000273	4	Medium	HKLM\\system\\CurrentControlSet\\Services\\Tcpip\\Parameters\\SynAttackProtect
1000272	4	Medium	HKLM\\system\\CurrentControlSet\\Services\\Tcpip\\Parameters\\DisableIPSourceRouting
1000271	4	Medium	HKLM\\system\\CurrentControlSet\\Services\\Tcpip\\Parameters\\KeepAliveTime
1000270	4	Medium	HKLM\\system\\CurrentControlSet\\Services\\Tcpip\\Parameters\\EnableDeadGWDetect
1000269	4	High	HKLM\\system\\CurrentControlSet\\Services\\Tcpip\\Parameters\\EnableICMPRedirect

Exemple de liste de données d'audit de conformité avec SecurityCenter



IP Address	Repository Name	Score	Total	Critical	High	Medium	Low
.103	.4and.5	1366	221	0	123	19	79
.102	.4and.5	1364	221	0	123	18	80

Exemple de liste de données d'audit de conformité par serveur avec SecurityCenter

Pour plus d'informations sur l'utilisation de SecurityCenter, consultez la documentation de SecurityCenter disponible sur https://support.tenable.com/.

Pour plus d'informations

Tenable a créé plusieurs autres documents expliquant en détail l'installation, le déploiement, la configuration, l'utilisation et les tests d'ensemble de Nessus.

- **Nessus 5.2 Installation and Configuration Guide** (Guide d'installation et de configuration Nessus 5.2) : explication pas à pas des étapes d'installation et de configuration
- **Nessus 5.2 User Guide** (Guide de l'utilisateur Nessus 5.2) : configuration et utilisation de l'interface utilisateur Nessus
- **Nessus Credential Checks for Unix and Windows** (Contrôles des identifiants Nessus pour Unix et Windows) : informations sur la façon d'effectuer des scans de réseau authentifiés avec le scanner de vulnérabilité Nessus
- **Nessus Compliance Checks Reference** (Référence pour les contrôles de conformité Nessus) : guide complet sur la syntaxe des contrôles de conformité Nessus
- **Nessus v2 File Format** (Format de fichier Nessus v2) : décrit la structure du format de fichier `.nessus`, qui a été introduit avec Nessus 3.2 et NessusClient 3.2
- **Nessus 5.0 REST Protocol Specification** (Caractéristique du protocole Nessus 5.0 REST) : décrit le protocole REST et l'interface dans Nessus
- **Nessus 5 and Antivirus** (Nessus 5 et les antivirus) : présente le mode d'interaction des progiciels de sécurité les plus courants avec Nessus et fournit des conseils et des solutions qui favoriseront une meilleure coexistence des logiciels, sans compromettre la sécurité ou faire obstacle à vos opérations de scan des vulnérabilités

- **Nessus 5 and Mobile Device Scanning** (Nessus 5 et scan des périphériques mobiles) : décrit comment Nessus s'intègre à Microsoft Active Directory et aux serveurs MDM (serveurs de gestion des périphériques mobiles) afin d'identifier les périphériques mobiles utilisés sur le réseau

- **Nessus 5.0 and Scanning Virtual Machines** (Nessus 5.0 et scan des machines virtuelles) : présente comment utiliser le scanner de vulnérabilité Nessus de Tenable Network Security pour effectuer l'audit de la configuration des plateformes virtuelles et des logiciels exécutés sur ces plateformes

- **Strategic Anti-malware Monitoring with Nessus, PVS, and LCE** (Surveillance stratégique des programmes malveillants avec Nessus, PVS et LCE) : décrit comment la plateforme USM de Tenable peut détecter divers logiciels malveillants, identifier les infections de ces programmes malveillants et en déterminer l'étendue

- **Patch Management Integration** (Intégration de la gestion de correctifs) : décrit comment Nessus et SecurityCenter peuvent tirer parti des identifiants pour les systèmes de gestion de correctif IBM TEM, Microsoft WSUS et SCCM, VMware Go et Red Hat Network Satellite afin d'effectuer l'audit de correctifs sur les systèmes pour lesquels les identifiants peuvent ne pas être mis à la disposition du scanner Nessus

- **Real-Time Compliance Monitoring** (Surveillance de conformité en temps réel) : décrit comment les solutions de Tenable peuvent être utilisées pour faciliter le respect d'un grand nombre de types de règlements gouvernementaux et financiers

- **Tenable Products Plugin Families** (Familles de plugins des produits Tenable) : fournit la description et le resumé des familles de plugins pour Nessus, Log Correlation Engine et Passive Vulnerability Scanner

- **SecurityCenter Administration Guide** (Guide d'administration de SecurityCenter)

D'autres ressources en ligne sont répertoriées ci-dessous :

- Forum de discussions Nessus : https://discussions.nessus.org/

- Blog Tenable : http://www.tenable.com/blog

- Podcast Tenable : http://www.tenable.com/podcast

- Vidéos d'exemples d'utilisation : http://www.youtube.com/user/tenablesecurity

- Feed Twitter Tenable : http://twitter.com/tenablesecurity

N'hésitez pas à contacter Tenable aux adresses support@tenable.com et sales@tenable.com, ou consultez notre site internet sur http://www.tenable.com/.

À propos de Tenable Network Security

Les solutions de Tenable Network Security sont utilisées par plus de 20 000 organisations, dont tous les services du Département de la Défense des États-Unis (DOD, Department of Defense) ainsi que de nombreuses grandes entreprises internationales et agences gouvernementales, pour prendre une longueur d'avance sur les vulnérabilités, menaces et risques de conformité émergents. Ses solutions Nessus et SecurityCenter continuent de définir la norme pour l'identification des vulnérabilités, la prévention des attaques et le respect de nombreuses exigences règlementaires. Pour plus d'informations, veuillez consulter www.tenable.com.

SIÈGE MONDIAL

Tenable Network Security
7021 Columbia Gateway Drive
Suite 500
Columbia, Maryland 21046
410.872.0555
www.tenable.com