

Nessus-Compliancechecks

Systemkonfigurationen und Inhalte prüfen

14. Januar 2014

(Revision 74)

Inhaltsverzeichnis

Einleitung	4
Voraussetzungen	4
Nessus- und SecurityCenter-Kunden	4
Regeln und Konventionen	4
Compliancestandards	5
Konfigurationsaudits, Datenlecks und Compliance	6
Was ist ein Audit?	6
Audit und Sicherheitslückenscans im Vergleich	6
Exemplarische Auditelemente.....	6
Windows.....	7
UNIX.....	7
Cisco	8
Palo Alto-Firewall	8
IBM iSeries.....	8
NetApp Data ONTAP	8
Datenbanken.....	9
Auditberichte.....	10
Erforderliche Technologie	10
.nbin-Plugins für Windows- und UNIX-Compliancetests.....	10
.nbin-Plugins für Windows-Inhaltscompliancetests.....	10
.nbin-Plugins für Datenbankcompliancetests	10
.nbin-Plugins für IBM iSeries-Compliancetests.....	10
.nbin-Plugins für Cisco-Compliancetests.....	11
Palo Alto .nbin Nessus-Plugin.....	11
VMware .nbin Nessus-Plugin	11
Citrix XenServer .nbin Nessus-Plugin	11
HP ProCurve .nbin Nessus-Plugin	11
FireEye .nbin Nessus-Plugin.....	11
Auditrichtlinien	12
Nützliche Utilitys	12
Nessus-Scanner für UNIX und Windows.....	12
Anmeldedaten für die zu prüfenden Geräte	12
„su“, „sudo“ und „su+sudo“ für Audits verwenden.....	13
Beispiel für „sudo“	14
Beispiel für „su+sudo“	14
Wichtiger Hinweis zu „sudo“.....	15
Beispiel für Cisco IOS:	16
Windows .inf-Dateien mit i2a in .audit-Dateien konvertieren	17
Das Tool beschaffen und installieren.....	17
.inf-Dateien in .audit-Dateien konvertieren	17
Konvertierung analysieren.....	18
Das korrekte .inf-Einstellungsformat	18
UNIX-Konfigurationsdateien mit c2a in .audit-Dateien konvertieren	20
Das Tool beschaffen und installieren.....	21
MD5-Auditdatei erstellen.....	21
Auditdateien auf der Basis einer oder mehrerer Konfigurationsdateien erstellen	22
MAP-Datei erstellen	22

Weitere Einsatzmöglichkeiten für das Tool c2a	23
Manuelle Optimierung von .audit-Dateien.....	24
UNIX-Paketlisten mit p2a in .audit-Dateien konvertieren.....	24
Das Tool beschaffen und installieren.....	24
Syntax.....	24
Ausgabedatei auf der Basis aller installierten Pakete erstellen	25
Ausgabedatei auf Basis der Paketliste erstellen und auf dem Bildschirm ausgeben	25
Auditdateien auf der Basis einer angegebenen Eingabedatei erstellen	25
Nessus-Benutzeroberfläche verwenden (Beispiel).....	26
Compliancetests abrufen	26
Scanrichtlinie konfigurieren	26
Scan durchführen	30
Ergebnisse (Beispiel)	30
Nessus-Befehlszeile für UNIX verwenden (Beispiel)	31
Compliancetests abrufen	31
.nessus-Dateien verwenden.....	31
.nessusrc-Dateien verwenden	31
Scan durchführen	32
Ergebnisse (Beispiel)	32
SecurityCenter verwenden.....	33
Compliancetests abrufen	33
Scanrichtlinie zur Durchführung eines Compliance-Audits konfigurieren.....	33
Anmeldedaten verwalten.....	35
Ergebnisse analysieren.....	36
Weitere Informationen	38
Wissenswertes zu Tenable Network Security	40

Einleitung

Das vorliegende Dokument beschreibt, wie Sie mithilfe von Nessus 5.x Audits nach einer Compiancerichtlinie für UNIX-, Windows-, Datenbank-, SCADA-, IBM iSeries- und Cisco-Systeme durchführen und verschiedene Systeme nach sensiblen Inhalten durchsuchen.



Die Begriffe „Richtliniencompliance“ und „Compliancetests“ werden in diesem Dokument synonym verwendet.



Zwar sind Audits für SCADA-Systeme mit Nessus möglich, doch ist diese Funktionalität nicht Gegenstand des vorliegenden Dokuments. Weitere Informationen hierzu entnehmen Sie der Tenable SCADA-Informationseite ([hier klicken](#)).

Trotz vorhandener Überschneidungen unterscheidet sich die Durchführung eines Compliance-Audits von der eines Sicherheitslückenscans. Beim Compliance-Audit wird festgestellt, ob das System entsprechend der geltenden Richtlinien konfiguriert wurde. Der Sicherheitslückenscan bestimmt, ob das System bekannte Sicherheitslücken aufweist. In diesem Dokument erfahren Sie, welche Arten von Konfigurationsparametern und sensiblen Daten Bestandteil eines Audits sein können, wie man Nessus für diese Audits konfiguriert und wie dieser Prozess mithilfe von Tenable SecurityCenter verwaltet und automatisiert werden kann.

Voraussetzungen

In diesem Dokument werden bestimmte Kenntnisse zum Nessus-Sicherheitslückenscanner vorausgesetzt. Weitere Informationen zur Konfiguration von Nessus für lokale Patchaudits unter UNIX und Windows entnehmen Sie dem Handbuch „Authentifizierte Nessus-Tests für UNIX und Windows“, das unter <http://www.tenable.com/products/nessus/documentation> verfügbar ist.

Nessus- und SecurityCenter-Kunden

Benutzer, die die in diesem Dokument beschriebenen Compliancetests durchführen möchten, müssen entweder eine kostenpflichtige Nessus-Version beziehen oder SecurityCenter für die Tests verwenden. Beide Komponenten sind bei Tenable Network Security erhältlich (<http://www.tenable.com/>). Eine ausführlichere Liste der technischen Anforderungen für die Ausführung von Audittests ist im weiteren Verlauf dieses Dokuments enthalten.

Regeln und Konventionen

In der gesamten Dokumentation werden Dateinamen, Daemons und ausführbare Dateien in einer Schriftart wie **courier bold** angezeigt.

Befehlszeichenoptionen und Schlüsselwörter werden ebenfalls in der Schriftart **courier bold** angezeigt. Die Befehlszeilen sind teils mit, teils ohne Befehlszeilen-Prompt und den Ausgabertext des betreffenden Befehls aufgeführt. In den Befehlszeilen erscheint der ausgeführte Befehl in der Schriftart **courier bold**, um zu verdeutlichen, was der Benutzer eingegeben hat. Die vom System generierte Beispielausgabe ist hingegen in der Schriftart **courier** (ohne Fettdruck) aufgeführt. Es folgt ein Beispiel für die Ausführung des UNIX-Befehls **pwd**:

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



Wichtige Hinweise und Aspekte werden durch dieses Symbol und graue Textfelder hervorgehoben.



Tipps, Beispiele und Best Practices (Empfehlungen) werden durch dieses Symbol und weißen Text auf blauem Grund hervorgehoben.

Compliancestandards

Es gibt viele verschiedene Anforderungen an gesetzliche und finanzielle Compliance. Hierbei ist zu beachten, dass es sich um Mindestanforderungen handelt, die je nach Geschäftszielen des Unternehmens unterschiedlich interpretiert werden können. Compliance-Anforderungen müssen den Geschäftszielen zugeordnet werden, um sicherzustellen, dass Risiken angemessen erkannt und beseitigt werden. Weitere Informationen zur Entwicklung eines solchen Prozesses entnehmen Sie dem von Tenable veröffentlichten Dokument „Maximizing ROI on Vulnerability Management“ („Renditemaximierung beim Sicherheitslückenmanagement“), das unter <http://www.tenable.com/whitepapers> verfügbar ist.

Angenommen, eine Richtlinie in einem Unternehmen sieht vor, dass auf allen Servern, auf denen personenbezogene Daten gespeichert sind, eine Protokollierung aktiviert ist und alle Kennwörter eine Mindestlänge von zehn Zeichen aufweisen. Eine solche Richtlinie könnte die Bemühungen des Unternehmens unterstützen, eine beliebige Anzahl verschiedener Vorschriften zu erfüllen.

Geläufige Compliancevorschriften und -handbücher behandeln unter anderem:

- BASEL II
- Center for Internet Security Benchmarks (CIS)
- Control Objectives for Information and related Technology (COBIT)
- Defense Information Systems Agency (DISA) STIGs
- Federal Information Security Management Act (FISMA)
- Federal Desktop Core Configuration (FDCC)
- Gramm-Leach-Bliley Act (GLBA)
- Health Insurance Portability and Accountability Act (HIPAA)
- ISO-Sicherheitsnormen 27002/17799
- Information Technology Information Library (ITIL)
- Konfigurationsempfehlungen des National Institute of Standards (NIST)
- Konfigurationsempfehlungen der National Security Agency (NSA)
- Payment Card Industry Data Security Standards (PCI DSS)
- Sarbanes-Oxley (SOX)
- Site Data Protection (SDP)
- Konfigurationsbaseline der Regierung der Vereinigten Staaten (United States Government Configuration Baseline, USGCB)
- Verschiedene nationalstaatliche Vorschriften

Diese Compliancetests beziehen auch eine Echtzeitüberwachung etwa bei der Durchführung einer Intrusion Detection und eine Zugriffssteuerung ein. Wenn Sie mehr dazu erfahren möchten, wie Tenable-Lösungen für Konfigurationsaudits, Sicherheitslückenmanagement, Datenlecks, Logdateianalyse und Netzwerküberwachung Sie bei der Einhaltung der

erwähnten Compliancevorschriften unterstützen, schicken Sie eine E-Mail an sales@tenable.com, um ein Exemplar des Ratgebers „Real-Time Compliance Monitoring“ („Compliance-Überwachung in Echtzeit“) zu bestellen.

Konfigurationsaudits, Datenlecks und Compliance

Was ist ein Audit?

Mithilfe von Nessus können Sie sich bei UNIX- und Windows-Servern, Cisco-Geräten, [SCADA](#)-Systemen, IBM iSeries-Servern und Datenbanken anmelden, um zu überprüfen, ob sie entsprechend der lokalen Standortsicherheitsrichtlinie konfiguriert wurden. Nessus kann zudem die gesamte Festplatte eines Windows- oder UNIX-Systems nach unzulässigen Inhalten durchsuchen.

Vor der Durchführung eines Audits muss in jedem Fall zunächst eine Standortsicherheitsrichtlinie erstellt werden, um sicherzustellen, dass Assets wie erforderlich geschützt sind. Mithilfe einer Sicherheitslückenbewertung wird festgestellt, ob die Systeme für bekannte Exploits anfällig sind; nicht herausgefunden werden kann hierbei allerdings etwa, ob personenbezogene Daten auf einem öffentlichen Server gespeichert sind.

Es gibt im Sicherheitsbereich keinen allgemeingültigen Standard, sondern es geht immer um die Verwaltung von Risiken, und diese unterscheiden sich von Organisation zu Organisation.

Sehen Sie sich beispielsweise einmal Kennwortanforderungen wie die kürzesten oder längsten Verfalldaten von Kennwörtern oder die Richtlinien für eine Kontosperrung an. Es gibt viele gute Gründe dafür, Kennwörter häufiger, aber auch weniger häufig zu ändern. Ebenso gibt es viele Gründe für die Sperrung eines Kontos nach fünf fehlgeschlagenen Anmeldeversuchen; wenn es aber um ein unternehmenskritisches System geht, kann es durchaus sinnvoll sein, hierfür einen höheren Wert festzulegen oder die Kontensperrung ganz zu unterbinden.

Diese Konfigurationseinstellungen haben eher mit Systemmanagement und Sicherheitsrichtlinien als mit Sicherheitslücken oder fehlenden Patches zu tun. Nessus kann Compliancetestes für UNIX- und Windows-Server durchführen. Richtlinien können sehr einfach, aber auch sehr komplex sein – je nach Anforderungen des einzelnen Compliancescans.

Audit und Sicherheitslückenscans im Vergleich

Nessus kann Sicherheitslückenscans von Netzwerkdiensten durchführen und sich bei Servern anmelden, um fehlende Patches zu ermitteln. Das Fehlen von Sicherheitslücken bedeutet aber nicht, dass die Server korrekt konfiguriert sind oder bestimmte Standards einhalten.

Der Vorteil der Verwendung von Nessus zur Durchführung von Sicherheitslückenscans und Compliance-Audits besteht darin, dass alle erforderlichen Daten in einem Arbeitsgang abgerufen werden können. Zur Ermittlung von Maßnahmen zur Risikovermeidung sollten Sie wissen, wie ein Server konfiguriert ist, welche Patches auf ihm vorhanden sind und wo es Sicherheitslücken gibt.

Von höherer Warte aus betrachtet können Sicherheit und Risiken global analysiert werden, wenn diese Informationen für ein gesamtes Netzwerk oder eine vollständige Assetklasse ermittelt werden (wie es mit Tenable SecurityCenter möglich ist). Auf diese Weise können Prüfer und Netzwerkmanager Trends in Systemen ausmachen, die Compliance-Anforderungen nicht einhalten, und Parameter so einstellen, dass Probleme im großen Umfang beseitigt werden.

Exemplarische Auditelemente

Die folgenden Abschnitte beschreiben Konfigurationsaudits auf Windows-, UNIX-, Datenbank-, IBM iSeries- und Cisco-Systemen.



Das RegEx-Modul (Modul zur Auswertung regulärer Ausdrücke) von Nessus 5 basiert auf einem Perl-Dialekt und gilt aufgrund seiner Flexibilität und Geschwindigkeit als eine Art „erweitertes POSIX“.



Alle Auditdateien müssen ANSI-kodiert sein. Unicode-, Unicode Big Endian- und UTF-8-kodierte Dateien können nicht funktionieren.

Windows

Nessus kann auf Vorhandensein beliebiger Einstellungen testen, deren Konfiguration als „Richtlinie“ im Microsoft Windows-Framework möglich ist. Mehrere hundert Registrierungseinstellungen lassen sich überwachen, und auch die Berechtigungen von Dateien, Verzeichnissen und Objekten können analysiert werden. Die folgende Liste enthält exemplarisch einige Audits, mit denen die folgenden Einstellungen getestet werden können:

- Kontosperrdauer
- Aufbewahrung von Sicherheitsprotokollen
- Zulassen lokaler Anmeldungen
- Erzwingen einer Kennwortchronik

Als Nächstes zeigen wir Ihnen ein „Beispielaudit“ für Windows-Server:

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

Dieses Audit sucht auf einem Windows-Server nach der Einstellung „Minimale Kennwortlänge“ und generiert eine Warnung, wenn der Wert weniger als sieben Zeichen umfasst.

Nessus kann auf Windows-Computern auch nach sensiblen Daten suchen. Das nächste Beispiel zeigt eine Suche nach Visa-Kreditkartennummern in Dateien unterschiedlichster Formate:

```
<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>
```

Dieser Test überprüft Excel-, Adobe- und Textdateien auf Muster, die vermuten lassen, dass mindestens eine Visa-Kreditkartennummer vorhanden ist.

UNIX

Nessus kann vielseitig zum Test auf Dateiberechtigungen, Dateiinhalte, laufende Prozesse und die Benutzerzugriffssteuerung auf unterschiedlichen UNIX-basierten Systemen verwendet werden. Gegenwärtig gibt es Tests für Audits von Solaris-, Red Hat-, AIX-, HP-UX-, SuSE-, Gentoo- und FreeBSD-Derivaten von UNIX.

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

Dieses Audit überprüft, ob die Mindestkennwortlänge auf einem UNIX-System 14 Zeichen beträgt.

Cisco

Nessus kann die laufende Konfiguration auf Systemen unter dem Betriebssystem Cisco IOS testen und überprüfen, ob dieses in Einklang mit den Standards einer Sicherheitsrichtlinie steht. Die Tests können über einen nichtprivilegierten Anmeldenamen oder aber einen Anmeldenamen erfolgen, der das privilegierte Enable-Kennwort verwendet.

```
<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>
```

Palo Alto-Firewall

Nessus ruft mithilfe von XSL-Transformationen (XSLTs) und einer nativen API Daten bei PAN-OS-basierten Palo Alto-Geräten ab. Die Abfragen erfolgen über die HTTP- oder HTTPS-Schnittstelle der Firewall. Hierzu werden ab PAN-OS 4.1.0 Superuser- oder Superuser (readonly)-Administratoranmeldedaten und bei früheren PAN-OS-Versionen Superuser-Administratoranmeldedaten benötigt. Auf diese Weise können Sie Audits der `operational config` auf dem Gerät durchführen.

```
<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

Nessus kann mithilfe der angegebenen Anmeldedaten die auf IBM iSeries-Systemen ausgeführte Konfiguration testen und überprüfen, ob diese in Einklang mit den Standards einer Sicherheitsrichtlinie steht.

```
<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

Nessus kann mithilfe der angegebenen Anmeldedaten die auf NetApp Data ONTAP-Systemen ausgeführte Konfiguration testen und überprüfen, ob diese in Einklang mit den Standards einer Sicherheitsrichtlinie steht.


```

<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>

```

Datenbanken

Nessus lässt sich so konfigurieren, dass es sich bei Datenbanken folgender Typen anmelden und die Einhaltung der lokalen Sicherheitsrichtlinie überprüfen kann:

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

Grundsätzlich empfiehlt Tenable die Ausführung eines Datenbankcompliance-scans unter Verwendung eines Benutzers mit SYSDBA-Berechtigungen für Oracle, „sa“ oder einem Konto mit der Sysadmin-Rolle für MS-SQL oder einem DB2-Instanzbenutzerkonto für DB2. Hierdurch soll die Vollständigkeit des Berichts sichergestellt werden, da der Zugriff auf bestimmte Systemtabellen oder ausgeblendete Tabellen und Parameter nur mit einem Konto möglich ist, das über solche Berechtigungen verfügt. Beachten Sie, dass bei Oracle in den meisten Fällen ein Benutzer mit der DBA-Rolle die Mehrzahl der Tests in Tenable-Audits ausführt, aber bei einigen Tests aufgrund nicht ausreichender Zugriffsrechte Fehler auftreten können. Gleiches gilt auch für die anderen Datenbanken: Man könnte zwar ein Konto mit niedrigeren Berechtigungen für das Datenbankaudit einsetzen, doch ist in diesem Fall die Vollständigkeit des Berichts unter Umständen nicht zu gewährleisten.

Datenbankaudits umfassen normalerweise SELECT-Anweisungen, die sicherheitsrelevante Details aus der Datenbank abrufen. Hierzu gehören beispielsweise das Vorhandensein oder der Status unsicherer gespeicherter Prozeduren. Das nächste Beispiel stellt fest, ob die potenziell gefährliche gespeicherte Prozedur „xp_cmdshell“ aktiviert ist:

```

<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>

```

Die Fähigkeit, Auditdateien für jede Organisation zu schreiben und nach sensiblen Daten zu suchen, ist ausgesprochen nützlich. Das vorliegende Dokument beschreibt, wie angepasste Richtlinien für die Suche nach verschiedenen Datentypen erstellt werden.

Auditberichte

Bei der Ausführung eines Audits versucht Nessus festzustellen, ob der Host die Vorgaben einhält oder nicht, oder ob die Ergebnisse keine Rückschlüsse zulassen.

Complianceergebnisse in Nessus werden als „Pass“ („Bestanden“), „Fail“ („Nicht bestanden“) oder „Warning“ („Warnung“) protokolliert. Die Nessus-Benutzeroberfläche und Tenable SecurityCenter protokollieren bestandene Tests als „Info“, fehlgeschlagene als „High“ und uneindeutige als „Medium“ (z. B. ein Zugriffsrechttest für Dateien, die nicht im System gefunden wurden).

Anders als ein Sicherheitslückentest, bei dem nur tatsächlich vorhandene Sicherheitslücken gemeldet werden, wird bei einem Compliantetest also immer etwas protokolliert. Auf diese Weise können die Daten als Grundlage für einen Auditbericht dienen, mit dem sich zeigen lässt, ob ein Host einen bestimmten Test bestanden oder nicht bestanden hat bzw. ob der Test nicht ordnungsgemäß durchgeführt werden konnte.

Erforderliche Technologie

.nbin-Plugins für Windows- und UNIX-Compliantetests

Tenable hat zwei Nessus-Plugins (IDs [21156](#) und [21157](#)) entwickelt, die die zur Durchführung von Audits von UNIX- und Windows-Systemen erforderlichen APIs implementieren. Diese Plugins wurden im Nessus-spezifischen „.nbin“-Format vorkompiliert.

Die Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen und SecurityCenter-Benutzern zur Verfügung. Ebenfalls im vorliegenden Dokument beschrieben werden zwei Windows-Tools, mit denen angepasste `.audit`-Dateien für Windows erstellt werden können, sowie ein UNIX-Tool zur Erstellung von `.audit` für UNIX.



Bei UNIX-Compliance-Audits wird nur die SSH-Authentifizierung unterstützt. Veraltete Protokolle wie Telnet sind aus Sicherheitsgründen unzulässig.

.nbin-Plugins für Windows-Inhaltscompliantetests

Tenable hat ein Nessus-Plugin (ID [24760](#)) namens „Windows File Contents Check“ („Compliantetests für Inhalte von Windows-Dateien“) entwickelt, das APIs implementiert, mit denen Audits von Windows-Systemen auf nichtkonforme Inhalte wie personenbezogene Daten oder geschützte Informationen zum Gesundheitszustand von Patienten durchgeführt werden können. Diese Plugins wurden im Nessus-spezifischen „.nbin“-Format vorkompiliert. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen und SecurityCenter-Benutzern zur Verfügung.



Bitte beachten Sie, dass UNIX-Systeme vom Plugin 24760 nicht gescannt werden.

.nbin-Plugins für Datenbankcompliantetests

Tenable hat ein Nessus-Plugin (ID [33814](#)) namens „Database Compliance Checks“ („Datenbankcompliantetests“) entwickelt, das die zur Durchführung von Audits verschiedener Datenbanksysteme erforderlichen APIs implementiert. Das Plugin wurde im Nessus-spezifischen „.nbin“-Format vorkompiliert. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen und SecurityCenter-Benutzern zur Verfügung.



Datenbankcompliantetests stehen für SecurityCenter 3.4.3 und ältere Versionen nicht zur Verfügung.

.nbin-Plugins für IBM iSeries-Compliantetests

Tenable hat ein Nessus-Plugin (ID [57860](#)) namens „IBM iSeries Compliance Checks“ („IBM iSeries-Compliantetests“) entwickelt, das die zur Durchführung von Audits von IBM iSeries-Systemen erforderlichen APIs implementiert. Das Plugin wurde im Nessus-spezifischen „.nbin“-Format vorkompiliert. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen zur Verfügung.

Zur Durchführung eines erfolgreichen Compliancescans für ein iSeries-System sind für authentifizierte Benutzer die folgenden Berechtigungen erforderlich:

1. Ein Benutzer mit der Berechtigung „(*ALLOBJ)“ oder „audit (*AUDIT)“ kann ein Audit aller Systemwerte durchführen. Solche Benutzer gehören normalerweise zur Klasse „(*SECOFR)“.
2. Benutzer der Klassen „(*USER)“ und „(*SYSOPR)“ können Audits der meisten Werte mit Ausnahme von QAUDCTL, QAUDENDACN, QAUDFRCLVL, QAUDLVL, QAUDLVL2 und QCRTOBJAUD durchführen.

Wenn ein Benutzer nicht über die Berechtigungen verfügt, um auf einen Wert zuzugreifen, wird der Wert „*NOTAVL“ zurückgegeben.

.nbin-Plugins für Cisco-Compliancetests

Tenable hat ein Nessus-Plugin (ID [46689](#)) namens „Cisco IOS Compliance Checks“ („Cisco IOS-Compliancetests“) entwickelt, das die erforderlichen APIs implementiert, um Audits auf Systemen mit dem Cisco IOS-Betriebssystem durchzuführen. Das Plugin wurde im Nessus-spezifischen „.nbin“-Format vorkompiliert. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen zur Verfügung. Der Compliancetest kann für die gespeicherte Konfiguration (Saved Config), die laufende Konfiguration (Running Config) oder die Startkonfiguration (Startup Config) ausgeführt werden.

Palo Alto .nbin Nessus-Plugin

Tenable hat ein Nessus-Plugin (ID [64095](#)) namens „Palo Alto Networks PAN-OS Compliance Checks“ („Palo Alto PAN-OS-Compliancetests“) entwickelt, das die erforderlichen APIs implementiert, um Audits auf Palo Alto-Geräten durchzuführen. Zudem wird ein Nessus-Plugin (ID 64286) namens „Palo Alto Networks Settings“ („Palo Alto Netzwerkeinstellungen“) zur Konfiguration der für die Auditdurchführung erforderlichen Authentifizierungsangaben verwendet. Diese Plugins wurden im Nessus-spezifischen „.nbin“-Format vorkompiliert. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen zur Verfügung. Der Compliancetest kann auch Betriebssystemkonfigurationen überprüfen.

VMware .nbin Nessus-Plugin

Tenable hat ein Nessus-Plugin (ID [64455](#)) namens „VMware vCenter/vSphere Compliance Checks“ („VMware vCenter/vSphere Compliancetests“) entwickelt, das die VMware SOAP-API zum Audit der ESX-, ESXi- und vCenter-Software implementiert. Anmeldeinformationen zum Audit können den „VMware vCenter SOAP API Settings“ („VMware vCenter SOAP API-Einstellungen“) im Abschnitt „Advanced“ („Erweitert“) der Richtlinie hinzugefügt werden. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen zur Verfügung. Weitere Informationen zur Durchführung eines Audits für VMware finden Sie im [betreffenden Blogpost](#).

Citrix XenServer .nbin Nessus-Plugin

Tenable hat ein Nessus-Plugin (ID [69512](#)) namens „Citrix XenServer Compliance Checks“ („Citrix XenServer-Compliancetests“) entwickelt, das die APIs implementiert, die zum Audit von Citrix XenServer-Systemen sowie von Systemen von Anbietern verwendet werden, die ihre eigenen, auf [Open-Source-Code](#) basierenden Versionen von XenServer entwickeln. Anmeldedaten zur Durchführung des Audits können den Voreinstellungen für „Citrix XenServer Compliance Checks“ im Abschnitt „Advanced“ („Erweitert“) der Richtlinie hinzugefügt werden. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen zur Verfügung. Weitere Informationen zur Durchführung eines Audits auf XenServer finden Sie im [betreffenden Blogpost](#).

HP ProCurve .nbin Nessus-Plugin

Tenable hat ein Nessus-Plugin (ID [70271](#)) namens „HP ProCurve Compliance Checks“ („HP ProCurve-Compliancetests“) entwickelt, das die zur Durchführung von Audits von HP ProCurve-Systemen erforderlichen APIs implementiert. Anmeldedaten zur Durchführung des Audits können den Voreinstellungen für „HP ProCurve Compliance Checks“ im Abschnitt „Advanced“ („Erweitert“) der Richtlinie hinzugefügt werden. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen zur Verfügung.

FireEye .nbin Nessus-Plugin

Tenable hat ein Nessus-Plugin (ID [70469](#)) namens „FireEye Compliance Checks“ („FireEye-Compliancetests“) entwickelt, das die zur Durchführung von Audits auf FireEye-Systemen erforderlichen APIs implementiert. Anmeldedaten zur Durchführung des Audits können den Voreinstellungen für „FireEye Compliance Checks“ im Abschnitt „Advanced“

(„Erweitert“) der Richtlinie hinzugefügt werden. Diese Plugins und die entsprechenden Auditrichtlinien stehen Kunden kostenpflichtiger Nessus-Versionen zur Verfügung.

Auditrichtlinien

Tenable hat eine Anzahl verschiedener Auditrichtlinien für die Plattformen UNIX, Windows, Palo Alto, IBM iSeries, VMware und Cisco entwickelt. Diese stehen als `.audit`-Textdateien für Benutzer kommerzieller Nessus-Versionen zur Verfügung und können vom Tenable Support Portal unter <https://support.tenable.com/> heruntergeladen werden. Aktuelle Neuigkeiten zu Tenables Auditfunktionalität und allen aktuellen `.audit`-Dateiveröffentlichungen finden Sie in den Diskussionsforen unter <https://discussions.nessus.org/>.

Viele Aspekte gängiger Compliance-Audits wie die Anforderungen von SOX, FISMA und PCI DSS wurden beim Schreiben dieser Auditrichtlinien berücksichtigt, auch wenn diese nicht als offizielle Auditdateien für diese Kriterien dargestellt werden. Benutzer sollten diese `.audit`-Richtlinien prüfen und die Tests an die lokalen Gegebenheiten anpassen. Die Namen der `.audit`-Dateien können an lokale Anforderungen angepasst werden. Andere `.audit`-Richtlinien basieren direkt auf den empfohlenen Konfigurationseinstellungen von [CERT](#), [CIS](#), [NSA](#) und [NIST](#).

Tenable wird auch weiterhin verschiedene Arten von `.audit`-Dateien basierend auf Kundenrückmeldungen und neue Best Practices entwickeln. Mehrere Consultingunternehmen und Tenable-Kunden haben ebenfalls bereits eigene `.audit`-Richtlinien implementiert und ihr Interesse daran geäußert, diese mit anderen Benutzern kostenpflichtiger Nessus-Versionen zu teilen. Eine ideale Gelegenheit zur Freigabe von `.audit`-Richtlinien oder Kommunikation mit der Nessus-Community bieten die Tenable Network Security-Diskussionsforen unter <https://discussions.nessus.org/>.

Nützliche Utilitys

Tenable hat ein Tool zur Konvertierung von `.inf`-Dateien in Nessus-`.audit` Dateien zur Durchführung von Windows-Audits entwickelt. Dieses Tool namens `i2a` wird im weiteren Verlauf dieses Dokuments noch behandelt.

Es gibt zwei UNIX-Tools, die zur Erstellung von `.audit`-Dateien für UNIX verwendet werden können. Das erste Tool namens `c2a` („Configuration to Audit“) kann zur Erstellung von `.audit`-Dateien in UNIX direkt aus bestehenden Konfigurationsdateien verwendet werden. Wenn beispielsweise Ihre Sendmail-Konfigurationsdatei korrekt und in Übereinstimmung mit Ihrer Standortrichtlinie konfiguriert ist, kann das Tool `c2a` eine Auditrichtlinie basierend auf der MD5-Prüfsumme der Datei oder bestimmten Wert-Argument-Paaren in der `sendmail.cf`-Datei erstellen. Das zweite Tool namens `p2a` („Package to Audit“) wird für die Erstellung von `.audit`-Dateien in UNIX wahlweise aus einem Basispaketsatz auf einem UNIX-System (RPM-basiertes Linux oder Solaris 10) oder aus einer einfachen Textdatei mit einer Liste mit Paketnamen verwendet.

Nessus-Scanner für UNIX und Windows

Zur Ausführung von Compliancetests kann eine Vielzahl unterschiedlicher Plattformen verwendet werden. Dabei spielt das Betriebssystem, unter dem Nessus ausgeführt wird, eigentlich keine Rolle. Sie können also Compliance-Audits eines Windows 2003-Servers über ein OS X-Laptop ebenso ausführen wie das Audit eines Solaris-Servers von einem Windows-Laptop aus.

Anmeldedaten für die zu prüfenden Geräte

In allen Fällen sind UNIX-SSH-, Windows-Domänen-, IBM iSeries-, Cisco IOS- oder Datenbank anmeldedaten erforderlich, damit Nessus auf dem Zielsystem angemeldet werden kann. Meistens muss der betreffende Benutzer ein „Superuser“, zumindest aber ein einfacher Benutzer mit der Möglichkeit der Berechtigungseskalation sein (z. B. `sudo`, `su` oder `su+sudo`). Verfügt der Benutzer, der das Audit ausführt, nicht über Superuser-Berechtigungen, dann können viele Remotesystembefehle nicht ausgeführt werden oder führen zu falschen Ergebnissen.

Das Windows-Konto, dessen Daten für die Anmeldung verwendet werden, benötigt die Berechtigung zum Lesen der lokalen Computerrichtlinie. Wenn ein Zielhost nicht Teil einer Windows-Domäne ist, muss das Konto Mitglied der Administratorgruppe des Hosts sein. Ist der Host hingegen Mitglied einer Domäne, dann ist die Administratorgruppe der Domäne auch Mitglied der Administratorgruppe des Hosts, und das Konto verfügt über Zugriff auf die lokale Computerrichtlinie, sofern es Mitglied der Administratorgruppe der Domäne ist.

Zur Ausführung von Windows-Inhaltscompliancetests muss zusätzlich zur Systemanmeldung mit Domänenberechtigungen auch der Zugriff auf die Windows-Verwaltungsinstrumentation (Windows Management

Instrumentation, WMI) erlaubt sein. Ist dieser Zugriff nicht vorhanden, dann meldet Nessus, dass der WMI-Zugriff für den Scan nicht verfügbar war.

Zur Durchführung eines vollständigen Datenbankcompliance-Audits sind nur die Anmeldedaten für die Datenbank erforderlich. Grund hierfür ist die Tatsache, dass die Datenbank – und nicht das Betriebssystem – auf Compliance geprüft wird.

Cisco IOS-Compliance-Tests erfordern in der Regel das Enable-Kennwort zur Durchführung eines vollständigen Compliance-Audits der Systemkonfiguration. Dies liegt daran, dass Nessus die Ausgabe des Befehls „`show config`“ überprüft, die nur berechtigten Benutzern zur Verfügung steht. Wenn der Nessus-Benutzer, unter dessen Konto das Audit ausgeführt wird, bereits über Enable-Berechtigungen verfügt, ist das Enable-Kennwort nicht erforderlich.

Weitere Informationen zur Konfiguration von Nessus oder SecurityCenter zur Durchführung lokaler authentifizierter Sicherheitslückenscans entnehmen Sie dem Handbuch „Authentifizierte Nessus-Tests für UNIX und Windows“, das unter <http://www.tenable.com/products/nessus/documentation> verfügbar ist.

„su“, „sudo“ und „su+sudo“ für Audits verwenden



Verwenden Sie „`su+sudo`“ in Fällen, in denen die Unternehmensrichtlinie die Anmeldung von Nessus an einem Remotehost mit Root-Privilegien oder einem Benutzer mit „`sudo`“-Berechtigungen untersagt. Bei Remote-Anmeldungen kann der nichtprivilegierte Nessus-Benutzer mit „`su`“ auf einen Benutzer mit `sudo`-Berechtigungen umgestellt werden.

Die wirksamsten authentifizierten Scans von UNIX sind solche, bei denen das angegebene Konto über Root-Berechtigungen verfügt. Da zahlreiche Sites eine Remoteanmeldung als Root nicht zulassen, können Nessus-Benutzer „`su`“, „`sudo`“ oder „`su+sudo`“ mit einem separaten Kennwort für ein Konto aufrufen, für das die passenden Berechtigungen konfiguriert wurden.

Außerdem wird Nessus sich, sofern möglich, nur bei den in dieser Datei genannten Hosts anmelden, wenn die SSH-Datei `known_hosts` vorhanden und als Teil der Scanrichtlinie angegeben ist. Auf diese Weise soll sichergestellt werden, dass die Kombination aus Benutzername und Kennwort, die Sie für Audits Ihrer bekannten SSH-Server verwenden, nicht zur Anmeldung auf einem System verwendet wird, das sich nicht unter ihrer Kontrolle befindet.

Beispiel für „sudo“

Es folgt eine Bildschirmabbildung für die Verwendung von „sudo“ in Verbindung mit SSH-Schlüsseln. In diesem Beispiel heißt das Benutzerkonto „audit“ und wurde der Datei `/etc/sudoers` auf dem zu scannenden System hinzugefügt. Angegeben wird das Kennwort für das Konto „audit“, nicht das Root-Kennwort. Die SSH-Schlüssel entsprechen Schlüsseln, die für das Konto „audit“ erstellt wurden:

The screenshot shows a configuration window titled "New Advanced Policy / Credentials / SSH settings". It contains the following fields and options:

- Credential Type:** SSH settings (dropdown menu)
- SSH user name:** audit (text input)
- SSH password (unsafe!):** (empty text input)
- SSH public key to use:** Add File (button)
- SSH private key to use:** Add File (button)
- Passphrase for SSH key:** (empty text input)
- Elevate privileges with:** sudo (dropdown menu)
- Privilege elevation binary path (directory):** (empty text input)
- su login:** (empty text input)
- Escalation account:** root (text input)
- Escalation password:** (password input field with 15 dots)

Beispiel für „su+sudo“

Mit der Veröffentlichung von Nessus 4.2.2 wurde eine neue Methode zum Hochstufen von Anmeldedaten für UNIX-basierte Hosts eingeführt, auf denen `sudo` installiert ist: „`su+sudo`“. Mithilfe dieser Methode können Sie Anmeldedaten für ein Konto angeben, das nicht über `sudo`-Berechtigungen verfügt, es mit `su` auf ein berechtigtes Benutzerkonto umstellen, und schließlich den Befehl `sudo` eingeben.

Diese Konfiguration bietet während des Scans mehr Sicherheit für Ihre Anmeldedaten und genügt den Compliance-Anforderungen zahlreicher Organisationen.

Wählen Sie zur Aktivierung der Funktion im Bereich „Elevate privileges with“ („Berechtigungen hochstufen mit“) der Anmeldedaten- und SSH-Einstellungen den Eintrag „su+sudo“ aus (siehe folgende Bildschirmabbildung):

New Advanced Policy / Credentials / SSH settings

Credential Type: SSH settings

SSH user name: raven

SSH password (unsafe):

SSH public key to use: Add File

SSH private key to use: Add File

Passphrase for SSH key:

Elevate privileges with: su+sudo

Privilege elevation binary path (directory):

su login:

Escalation account: root

Escalation password:

Geben Sie in die Felder „SSH user name“ („SSH-Benutzername“) und „SSH password“ („SSH-Kennwort“) die Anmeldedaten des Kontos ein, das nicht über `sudo`-Berechtigungen verfügt. Im obigen Beispiel heißt das Benutzerkonto „raven“. Wählen Sie im Pulldownmenü „Elevate privileges with“ den Eintrag „su+sudo“ aus. Geben Sie in die Felder „su login“ („su-Anmeldename“) und „Escalation password“ („Eskalationskennwort“) die Anmeldedaten des Kontos ein, das über Berechtigungen verfügt (in diesem Fall „sumi“). Andere Änderungen an der Scanrichtlinie sind nicht erforderlich.

Wichtiger Hinweis zu „sudo“

Beachten Sie bei Audits auf UNIX-Systemen via `su`, `sudo` oder `su+sudo` bitte Folgendes:

- Wenn Ihr UNIX-System dahingehend eingeschränkt wurde, welche Befehle über `sudo` ausgeführt werden können und auf welche Dateien Remotebenutzer zugreifen dürfen, kann dies Auswirkungen auf Ihr Audit haben. Wenn Sie mutmaßen, dass das Audit aufgrund von Sicherheitsmaßnahmen nur beschränkt ausgeführt wurde, vergleichen Sie die Ergebnisse eines Audits ohne Root-Berechtigungen mit denen eines Audits, das mit Root-Berechtigungen ausgeführt wurde.
- Der Befehl `sudo` ist in Solaris nicht nativ, sondern muss, wenn ihr Zielsystem unter Solaris läuft, heruntergeladen und installiert werden. Vergewissern Sie sich, dass die `sudo`-Binärdatei als „`/usr/bin/sudo`“ zugänglich ist.
- Bei Nessus-Scans mit `known_hosts` muss noch ein zu scannender Host angegeben werden. Wenn Sie beispielsweise ein Klasse-C-Netzwerk scannen, aber eine `known_hosts`-Datei hochgeladen haben, die nur 20 Hosts in diesem Klasse-C-Netzwerk enthält, dann scannt Nessus auch nur die in dieser Datei genannten Hosts.

- Bei einigen UNIX-basierten Konfigurationen ist es erforderlich, dass durch „sudo“ eingeleitete Befehle in `tty`-Sitzungen ausgeführt werden. Nessus-Sicherheitslückenscans, die mit der Option „`su+sudo`“ ausgeführt werden, erfüllen diese Anforderungen nicht. Wenn Sie die Option „`su+sudo`“ verwenden, müssen Sie eine Ausnahme auf dem Zielsystem erstellen. Um festzustellen, ob dieser Fall auf Ihre UNIX-Distribution zutrifft, geben Sie als Root den folgenden Befehl auf dem System ein, das Sie scannen möchten:

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

Wenn die Zeile „`requiretty`“ in der `sudoers`-Konfigurationsdatei vorhanden ist, müssen Sie eine Ausnahme für diese Regel in der Datei `/etc/sudoers` erstellen:

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

Beachten Sie, dass `{userid}` der Benutzername ist, der zur Ausführung des „`sudo`“-Befehls verwendet wird (Seite „`su login`“ in den Anmeldedaten- und SSH-Einstellungen Ihrer Richtlinie). Vergewissern Sie sich außerdem, dass die folgende Zeile in Ihrer Datei `sudoers` vorhanden ist:

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

Beachten Sie auch hier, dass `{userid}` der Benutzername ist, der zur Ausführung des „`sudo`“-Befehls verwendet wird („`su login`“ in den Anmeldedaten- und SSH-Einstellungen Ihrer Richtlinie).

Beispiel für Cisco IOS:



Es wird nur die SSH-Authentifizierung unterstützt. Veraltete IOS-Geräte, die Telnet zur Authentifizierung benötigen, können mit Nessus Cisco-Compliancetests nicht gescannt werden.

Die Cisco IOS-Anmeldedaten werden im Fenster „**SSH settings**“ („SSH-Einstellungen“) der Nessus-Benutzeroberfläche erstellt. Geben Sie den für die Anmeldung am Cisco-Router erforderlichen SSH-Benutzernamen und das Kennwort ein. Um festzulegen, dass Berechtigungen mit „Enable“ hochgestuft werden müssen, wählen Sie „**Cisco 'enable'**“ neben der Einstellung „**Elevate privileges with**“ („Berechtigungen hochstufen mit“) aus und geben Sie dann das Enable-Kennwort in das Feld „**Escalation password**“ („Eskalationskennwort“) ein.

New Advanced Policy / Credentials / SSH settings

Credential Type: SSH settings

SSH user name: admin

SSH password (unsafe):

SSH public key to use: Add File

SSH private key to use: Add File

Passphrase for SSH key:

Elevate privileges with: Cisco 'enable'

Privilege elevation binary path (directory):

su login:

Escalation account:

Escalation password:

Windows .inf-Dateien mit i2a in .audit-Dateien konvertieren

Wenn Sie oder Ihre IT-Organisation über Windows-Richtliniendateien verfügen (diese weisen meist die Dateierweiterung „.inf“ auf), können diese für Nessus-Audits von Windows-Servern in .audit-Dateien konvertiert werden.

Das Tool beschaffen und installieren

Das Tool `i2a` steht als ZIP-Datei zur Verfügung und kann vom Tenable Support Portal unter <https://support.tenable.com/> heruntergeladen werden. Es verfügt nicht über eine grafische Benutzeroberfläche, sondern wird über die Befehlszeile ausgeführt.

Extrahieren Sie den Inhalt der Datei in ein Verzeichnis Ihrer Wahl und verschieben Sie die gewünschten .inf-Dateien in dasselbe Verzeichnis.

.inf-Dateien in .audit-Dateien konvertieren

Führen Sie das Konvertierungstool über die Befehlszeile aus. Hierzu geben Sie Folgendes ein:

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

In diesem Beispiel ist `yourfile.inf` die .inf-Quelldatei und `file.audit` die .audit-Zieldatei.

Konvertierung analysieren

Tenable hat versucht, eine möglichst hundertprozentige Konvertierung der Beschreibungen in einer `.inf`-Datei und der in einer `.audit`-Datei aufgeführten Auditelemente zu erzielen. Es gibt jedoch einige wenige Richtlinienelemente, die mithilfe der aktuellen Nessus 5-Technologie nicht getestet werden können.

Für jeden Konvertierungsdurchgang des Tools `i2a` wird eine Logdatei erstellt. Sie führt jeden Schritt des Konvertierungsvorgangs in einer einzelnen Zeile auf. Kann eine Zeile in der `.inf`-Datei nicht konvertiert werden, dann ist sie in dieser Logdatei enthalten.

Das korrekte .inf-Einstellungsformat

Bitte überprüfen Sie bei Tests, die laut Logdatei nicht verarbeitet werden konnten, ob sie den nachfolgend aufgeführten zulässigen Formaten entsprechen.

Die Einstellungen **System Access (Systemzugriff)**, **System Log (Systemprotokoll)**, **Security Log (Sicherheitsprotokoll)**, **Application Log (Anwendungsprotokoll)** und **Event Audit (Ereignisprotokoll)** weisen dasselbe Format auf. Jeder Eintrag wird durch den Schlüssel („**key**“) und einen nachfolgenden Wert („**value**“) beschrieben.

Syntax:

```
Key = value
```

Im obigen Fall ist „**key**“ das zu prüfende Element und „**value**“ der erwartete Wert für diesen Schlüssel auf dem Remotesystem.

Beispiel:

```
MinimumPasswordLength = 8
```

Das Format für die **Privilege Rights**-Einstellungen („**Berechtigungen**“) ähnelt dem oben erwähnten, nur kann der Wert bei dieser Einstellung leer sein.

Syntax:

```
PriviledgeRight = User1,User2...UserN
```

Beispiel:

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

Oder:

```
SeTcbPrivilege =
```

Eine **Registry Key**-Einstellung („Registrierungsschlüssel“) umfasst die folgenden vier Komponenten:

- „Registry Key“ (der zu prüfende Registrierungsschlüssel)
- „Inheritance Value“ (gibt an, ob die Berechtigungen für diesen Registrierungsschlüssel ererbt sind oder nicht; zulässig sind Werte zwischen 0 und 4)

- DACL (eine Zugriffssteuerungsliste – kurz ACL –, die vom Besitzer eines Objekts gesteuert wird und angibt, inwieweit bestimmte Benutzer oder Gruppen auf dieses Objekt zugreifen können)
- SACL (eine ACL, mit der die Erzeugung von Auditmeldungen für Zugriffsversuche auf ein sicherungsfähiges Objekt gesteuert wird)

Syntax:

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

Die Felder „DACL“ und „SACL“ können leer sein. In diesem Fall wird der Test ignoriert.

Beispiel:

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

Das Format der Einstellung **File Security** („Dateisicherheit“) ähnelt dem oben für „Registry Key“ („Registrierungsschlüssel“) beschriebenen.

Syntax:

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

Beispiel:

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

Die **Service General**-Einstellung („Allgemeine Diensteinstellungen“) umfasst die folgenden vier Komponenten:

- „Service Name“ („Dienstname“; der zu prüfende Dienst)
- „Service start type“ (Dienststarttyp, also „Manuell“, „Automatisch“ oder „Deaktiviert“; zulässig sind Werte zwischen 2 und 4)
- DACL (eine Zugriffssteuerungsliste – kurz ACL –, die vom Besitzer eines Objekts gesteuert wird und angibt, inwieweit bestimmte Benutzer oder Gruppen auf dieses Objekt zugreifen können)
- SACL (eine ACL, mit der die Erzeugung von Auditmeldungen für Zugriffsversuche auf ein sicherungsfähiges Objekt gesteuert wird)

Syntax:

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

Beispiel:

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

Wenn nur der Starttyp eines Diensts, nicht aber die Berechtigungen einer Diensteinstellung überprüft werden müssen, gehen Sie wie folgt vor.

Syntax:

```
Service Name, Start type
```

Beispiel:

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

Die **Registry Value**-Einstellung („Registrierungswert“) umfasst die folgenden drei Komponenten:

- „Registry Key“ (der zu prüfende Registrierungsschlüssel)
- „RegistryType“ (Registrierungstyp, also „REG_DWORD“, „REG_SZ“ usw.)
- „RegistryValue“ (Wert des Registrierungsschlüssels)



Der Registrierungswert („RegistryValue“) kann in doppelte oder einfache Anführungszeichen gesetzt oder ohne Anführungszeichen notiert werden.

Syntax:

```
RegistryKey, RegistryType, RegistryValue
```

Beispiel:

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

Wenn Sie eine bestimmte Zeile in der `.inf`-Datei auskommentieren möchten, stellen Sie Ihr ein Semikolon (;) voran. In diesem Fall wird die Zeile bei der Skriptausführung ignoriert.

UNIX-Konfigurationsdateien mit `c2a` in `.audit`-Dateien konvertieren

Das Tool `c2a.pl` soll Ihnen bei der Erstellung von `.audit`-Dateien helfen, mit denen Sie Audits für Anwendungskonfigurationen in einem Netzwerk durchführen können. Wenn beispielsweise gewünscht wird, dass alle Webserver in einem Netzwerk exakt so konfiguriert sein müssen wie der Masterhost X, dann würden Sie dieses Tool auf dem Host X ausführen und die `.audit`-Datei für `httpd` auf diesem System erstellen. Nachfolgend müssen Sie unter Verwendung dieser Datei als Eingabedatei für den Nessus-Daemon den Scan für alle anderen Webserver ausführen, um auf Compliance zu prüfen.

Optional kann dieses Tool auch verwendet werden, um MD5-Auditdateien für einen vollständigen Host zu erstellen. Erwartet wird eine Eingabedatei mit einer Liste aller zu prüfenden Dateien und Verzeichnisse. Durch rekursive Verarbeitung im Falle von Verzeichnissen wird dann eine `.audit`-Datei für das System erstellt. Anhand dieser Datei können später Scans auf Änderungen an Kerndateien und -verzeichnissen durchgeführt werden.

Das Tool beschaffen und installieren

Das Tool `c2a` ist ein komprimiertes `tar`-Archiv und kann vom Tenable Support Portal unter <https://support.tenable.com/> heruntergeladen werden.

Extrahieren Sie den Inhalt von `c2a-x.x.x.tar.gz` mithilfe des folgenden Befehls auf Ihren Computer:

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

Hierdurch wird im aktuellen Verzeichnis ein Unterverzeichnis „`c2a`“ erstellt, in das die Dateien automatisch extrahiert werden. Wenn Sie den Inhalt in ein Verzeichnis Ihrer Wahl extrahieren möchten, verwenden Sie den folgenden Befehl:

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

Nach der Dekomprimierung des Archivs sollten im Verzeichnis `~/c2a` die folgenden Dateien vorhanden sein:

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

MD5-Auditdatei erstellen

Führen Sie das Konvertierungstool mit der Option „`-md5`“ aus. Hierzu geben Sie Folgendes ein:

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

Das Tool erwartet eine Eingabedatei mit einer Liste von Dateien und Verzeichnissen, deren MD5-Werte beim Audit geprüft werden sollen, sowie einen Ausgabedateinamen für die Auditdatei.



Beachten Sie, dass Sie beim Hinzufügen von Dateien zur Eingabedatei das folgende Format verwenden müssen:

```
/path/to/file
```

Dieses Format verwenden Sie beim Hinzufügen von Verzeichnissen:

```
/path/to/file/
```

Wenn dieses Format verwendet wird, obwohl es sich nicht um ein Verzeichnis, sondern um eine Datei handelt, erscheint eine Warnmeldung des Tools `c2a`, derzufolge die Datei nicht vorhanden ist. Beim Hinzufügen von Verzeichnissen funktioniert der am Anfang stehende Schrägstrich („/“) einwandfrei.

Wenn der Eintrag in die Eingabedatei eine normale MD5-Datei ist, wird nur diese Datei verarbeitet und in das `.audit`-Format geschrieben. Handelt es sich hingegen um ein Verzeichnis, dann werden alle darin vorhandenen Dateien rekursiv abgearbeitet. Wenn keine Ausgabedatei angegeben ist, wird das Ergebnis in die Datei `~/c2a/op.audit` geschrieben.

Bei der Verarbeitung der durch die Eingabedatei spezifizierten Liste werden gegebenenfalls vorhandene symbolische Links ignoriert. Eine Warnmeldung wird angezeigt, derzufolge die Datei entweder nicht vorhanden oder ein symbolischer Link ist. Mit dieser neuen Version entfällt die Unterstützung symbolischer Links durch `c2a`.

Auditdateien auf der Basis einer oder mehrerer Konfigurationsdateien erstellen

Das Tool `c2a` ist ideal zum Verarbeiten von Konfigurationsdateien, die in separaten Zeilen vorhandene Inhalte aufweisen. Wenn Ihre Konfigurationsdatei jedoch zeilenübergreifende Inhalte enthält (wie es etwa bei XML-Konfigurationsdateien der Fall ist), dann ist `c2a` eher nicht geeignet.

Führen Sie das Konvertierungstool mit der Option „`-audit`“ aus. Hierzu geben Sie Folgendes ein:

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

Das Tool erwartet eine Eingabedatei (`input.txt`) mit einer Liste der zu prüfenden Konfigurationsdateien sowie einen Ausgabedateinamen für die Auditdatei.

Das Perl-Skript `c2a.pl` benötigt zwei Schlüsseldateien: `c2a.map` und `c2a_regex.map`. Es scannt die zu prüfende Konfigurationsdatei zeilenweise und überprüft dabei, ob das erste Wort der Zeile dem „Typ“ in der `c2a.map`-Datei (z. B. HTTP, SENDMAIL usw.) und dem ihm zugeordneten Wert entspricht. Werden beispielsweise Audits für HTTP-Einstellungen ausgeführt, dann wird überprüft, ob das Wort mit einem der HTTP-Schlüsselwörter in der Datei `c2a.map` übereinstimmt. Ist dies der Fall, dann wird der reguläre Ausdruck für HTTP aus `c2a_regex.map` auf diese Zeile angewendet, und die Einstellung sowie der Wert werden extrahiert. Es werden nur solche Einstellungen geprüft, für die in `c2a.map` ein Eintrag vorhanden ist.

Konfigurationsdateien, die nicht geprüft werden sollen, können mit dem Rautezeichen („`#`“) auskommentiert werden.



Wenn Sie die in der Konfigurationsdatei auskommentierten Einstellungen in das `.audit`-Format konvertieren möchten, bearbeiten Sie das Skript `c2a.pl` und legen Sie dort „`,$ENFORCE_COMMENT = 1;`“ fest.

Ist keine Ausgabedatei angegeben, wird ähnlich wie weiter oben beschrieben das Ergebnis in die Datei `~/c2a/op.audit` geschrieben.

Gegenwärtig bietet Tenable MAP-Einstellungen für HTTP, SENDMAIL, SYSCTL und NESSUS. Weitere Anwendungseinstellungen lassen sich durch Verwendung des Perl-Skripts `cmv.pl` problemlos hinzufügen. Weitere Informationen finden Sie im nächsten Abschnitt.

MAP-Datei erstellen

Die Erstellung einer MAP-Datei für eine Anwendung ist nicht schwierig. Sie müssen lediglich wie folgt das Skript `cmv.pl` ausführen:

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

Hierbei gilt:

- „`regex`“ ist der reguläre Ausdruck zum Extrahieren des Paares aus Konfigurationsparameter und Wert. Normalerweise hat er das Format „`<Name> = <Wert>`“. In einigen Fällen kann es geringfügige Unterschiede geben, beispielsweise durch Ersetzung des Gleichheitszeichens durch ein Leerzeichen, ein Tabulatorzeichen usw.
- „`tag`“ ist im Wesentlichen das Schlüsselwort, mit dem Sie die zu prüfende Anwendung auszeichnen werden. Das Schlüsselwort „`tag`“ verknüpft die Konfigurationsdatei `config_file` mit den Schlüsselwörtern in `c2a.map` und dem regulären Ausdruck in `c2a_regex.map`. Deswegen müssen die Tags in allen diesen Dateien identisch sein.
- „`config_file`“ ist die Datei, für die eine MAP-Datei erstellt wird.

Wenn Sie beispielsweise ein Audit der Konfigurationseinstellungen für VSFTPD durchführen möchten, führen Sie die folgenden Schritte aus:

1. Zunächst verwenden Sie `cmv.p1` wie folgt:

```
# ./cmv.p1 -r '([A-Za-z0-9_]+)=([A-Za-z0-9_]+)' -t VSFTPD -f /root/vsftpd-0.9.2/vsftpd.conf
```

Hierdurch wird die Datei `tag.map` erstellt (z. B. `VSFTPD.map`). Standardmäßig werden alle auskommentierten Zeilen ignoriert. Möchten Sie alle Variablen berücksichtigen, dann setzen Sie den Wert `$ENFORCE_COMMENT` von „0“ auf „1“ und führen Sie das Skript erneut aus.

2. Überprüfen Sie die MAP-Datei und hängen Sie sie an `c2a.map` an.

Prüfen Sie die Datei `VSFTPD.map` auf unerwünschte Werte, die unbeabsichtigt Ihrem regulären Ausdruck entsprechen könnten. Wenn Sie sich vergewissert haben, dass alle Schlüsselwörter korrekt sind, hängen Sie sie an `c2a.map` an.

3. Aktualisieren Sie `c2a_regex.map` mit demselben Ausdruck, der auch von `cmv.p1` verwendet wird. Hierzu gehen Sie wie folgt vor:

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

Hinweis: Es handelt sich um denselben regulären Ausdruck, der auch vom Perl-Skript `cmv.p1` verwendet wird.

4. Aktualisieren Sie `input.txt` mit dem Speicherort der VSFTPD-Konfigurationsdatei:

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

5. Führen Sie das Skript `c2a.p1` aus:

```
# ./c2a.p1 -audit -f input.txt
```

6. Überprüfen Sie abschließend die Ausgabedatei:

```
# vi op.audit
```

Weitere Einsatzmöglichkeiten für das Tool c2a

Tenable hat verschiedene Einträge in den Dateien `c2a.map` und `c2a_regex.map` vorgenommen, um Audits von Sendmail, VSFTPD (Very Secure FTP Daemon), Apache, der Red Hat-Datei `/etc/sysctl.conf` und Nessus durchzuführen. Weitere Softwareprogramme werden wahrscheinlich in absehbarer Zeit hinzugefügt. Wenn Sie Tenable Ihre eigenen Zuordnungen zukommen lassen möchten, damit auch andere Nessus-Benutzer davon profitieren können, senden Sie sie an die Adresse nessus-support@tenable.com.

Unter Berücksichtigung der genannten Gesichtspunkte kann das Skript `c2a.p1` verwendet werden, um die Erstellung von `.audit`-Dateien für verschiedene UNIX-Onlineanwendungen zu erleichtern. Erwägen Sie die folgenden Ansätze:

- Verfügt ihre Organisation über zahlreiche UNIX-basierte Firewalls, dann kann eine `.audit`-Datei für die Überprüfung der allgemeinen und obligatorischen Einstellungen der einzelnen Firewalls erstellt werden. Wenn beispielsweise alle Firewalls RFC 1918-Adressen ausfiltern sollen, können die tatsächlichen Firewallregeln darauf geprüft werden.
- Wenn viele unterschiedliche angepasste Anwendungen aus CRON heraus ausgeführt werden, können die verschiedenen CRONTABs geprüft werden, um sicherzustellen, dass alle Anwendungen zum jeweils korrekten Zeitpunkt ausgeführt werden.
- Für eine zentralisierte Protokollierung können die SYSLOG-, SYSLOG-NG- und LOGROTATE-Konfigurationen von UNIX-Remotesystemen getestet werden.

Manuelle Optimierung von .audit-Dateien

Abschließend kann die Ausgabe des Skripts `c2a.pl` auch manuell bearbeitet werden. Man könnte beispielsweise überlegen, die MD5-Prüfsummenregeln mit den `FILE_CONTENT_CHECK`-Regeln zu einer Regel zusammenzufassen. Die Ausgabe des Skripts `c2a.pl` setzt zudem voraus, dass sich eine Konfigurationsdatei stets am selben Ort befindet. Ziehen Sie in Betracht, das Schlüsselwort „`file`“ („Datei“) so abzuändern, dass weitere Speicherorte angegeben werden können, an denen sich eine Konfigurationsdatei befinden kann.

Wenn Sie bestimmte Inhalte nicht in Ihren Remotedateikonfigurationen wünschen, ziehen Sie in Betracht, entsprechende Tests mit dem Schlüsselwort `FILE_CONTENT_CHECK_NOT` manuell hinzuzufügen. Auf diese Weise können Sie Audits für Einstellungen durchführen, die vorhanden sein müssen bzw. nicht vorhanden sein dürfen.

UNIX-Paketlisten mit p2a in .audit-Dateien konvertieren

Das Tool `p2a.pl` soll Ihnen bei der Erstellung von `.audit`-Dateien für Installationspaketkonfigurationen auf RPM-basierten Linux- und Solaris 10-Systemen helfen. Sollen beispielsweise alle Linux-Webserver in einem Netzwerk dieselbe RPM-Basis wie der Masterhost X aufweisen, dann müssen Sie dieses Tool auf dem Host X ausführen, um eine `.audit`-Datei zu erstellen, die alle RPM-Pakete auf diesem System enthält. Danach könnten Sie unter Verwendung dieser `.audit`-Datei mit Nessus einen Scan mit einem Compliancetest für alle anderen Webserver ausführen.

Optional kann mithilfe dieses Tools auch eine Auditdatei aus einem Text erstellt werden, der RPM- oder Solaris 10-Pakete auflistet. Erwartet wird eine Eingabedatei, in der alle Pakete zeilenweise aufgelistet sind; ausgegeben wird dann eine korrekt formatierte `.audit`-Datei für das Zielsystem. Anhand der generierten `.audit`-Datei können später Scans auf Änderungen an Kerninstallationspaketen durchgeführt werden.

Das Tool beschaffen und installieren

Das Tool `p2a` ist ein komprimiertes `tar`-Archiv, das ein einzelnes Perl-Skript und eine Hilfedatei namens `ReadMe.txt` umfasst. Es kann vom Tenable Support Portal unter <https://support.tenable.com/> heruntergeladen werden.

Extrahieren Sie den Inhalt von `p2a-x.x.x.tar.gz` mithilfe des folgenden Befehls auf Ihren Computer:

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

Hierdurch wird im aktuellen Verzeichnis ein Unterverzeichnis „`p2a`“ erstellt, in das die Dateien automatisch extrahiert werden.

Wenn Sie den Inhalt in ein Verzeichnis Ihrer Wahl extrahieren möchten, verwenden Sie den folgenden Befehl:

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

Nach der Dekomprimierung des Archivs sollten im Verzeichnis `~/p2a` die folgenden Dateien vorhanden sein:

- `p2a.pl`
- `ReadMe.txt`

Machen Sie das Skript wie folgt ausführbar:

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

Syntax

Führen Sie das Perl-Skript wie folgt aus:

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




„-h“ ist ein optionales Standalone-Argument, mit dem das Hilfetool angezeigt wird.

Ausgabedatei auf der Basis aller installierten Pakete erstellen

Wird das Skript ausschließlich mit der Option „-o“ ausgeführt, dann führt es seinerseits einen Systembefehl aus, mit dem die Namen aller lokal installierten Systempakete extrahiert werden. Die resultierende `.audit`-Datei wird dann als `/path/to/outputfile.audit` gespeichert.

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



Ausgabedateinamen müssen die Dateierweiterung `.audit` enthalten, damit das Skript ausgeführt werden kann. Andernfalls wird eine Fehlermeldung generiert, die auf die unpassende Dateierweiterung hinweist.

Ausgabedatei auf Basis der Paketliste erstellen und auf dem Bildschirm ausgeben

Führen Sie `p2a` mit der folgenden Syntax aus, um die gesamte Ausgabe im Terminalfenster anzuzeigen:

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

Diese Option erfordert eine Eingabedatei und generiert eine Ausgabe im Terminalfenster (`stdout`), die Sie kopieren und in Ihre `.audit`-Datei einfügen können. Die Eingabedatei muss jeweils ein Paket pro Zeile enthalten. Trennzeichen sind unzulässig.

Beispiel:

```
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
```



Da viele UNIX-basierte Systeme mehr als tausend installierte Pakete aufweisen, kann der Bildlaufpuffer diese unter Umständen nicht alle aufnehmen. Hierdurch wird die Anzeige der Ausgabe erschwert.

Auditdateien auf der Basis einer angegebenen Eingabedatei erstellen

Wenn Sie bei der Ausführung von `p2a` sowohl Eingabe- als auch Ausgabeargumente angeben, wird aus Ihrer formatierten Paketliste eine `.audit`-Datei am angegebenen Speicherort erstellt.

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

Die Eingabedateien müssen ein Paket pro Zeile enthalten. Trennzeichen sind unzulässig.

Beispiel:

```
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
```



Ausgabedateinamen müssen die Dateierweiterung `.audit` enthalten, damit das Skript ausgeführt werden kann. Andernfalls wird eine Fehlermeldung generiert, die auf die unpassende Dateierweiterung hinweist.

Nessus-Benutzeroberfläche verwenden (Beispiel)

Compliancetests abrufen

Kunden kostenpflichtiger Nessus-Versionen verfügen bereits über die Compliancetests für ihren Nessus-Scanner. Zudem können verschiedene `.audit`-Dateien vom Tenable Support Portal unter <https://support.tenable.com/> heruntergeladen werden. Führen Sie zur Überprüfung die Nessus-Benutzeroberfläche aus, authentifizieren Sie sich und verwalten oder bearbeiten Sie dann eine vorhandene Richtlinie. Suchen Sie auf der Registerkarte „Plugins“ nach der Familie „Policy Compliance“ („Richtliniencompliance“) und klicken Sie auf den Namen der Plugin-Familie. Nun sollten die folgenden Plugins angezeigt werden:

- Cisco IOS Compliance Checks (Cisco IOS-Compliancetests)
- Database Compliance Checks (Datenbankcompliancetests)
- IBM iSeries Compliance Checks (Compliancetests für IBM iSeries-Systeme)
- PCI DSS Compliance (PCI-DSS-Compliance)
- PCI DSS Compliance: Database Reachable from the Internet (PCI-DSS-Compliance: Datenbank über das Internet erreichbar)
- PCI DSS Compliance: Handling False Positives (PCI-DSS-Compliance: Umgang mit Fehlalarmen)
- PCI DSS Compliance: Insecure Communication Has Been Detected (PCI-DSS-Compliance: Unsichere Kommunikation erkannt)
- PCI DSS Compliance: Remote Access Software Has Been Detected (PCI-DSS-Compliance: Software mit Remotezugriff erkannt)
- PCI DSS Compliance: Passed (PCI-DSS-Compliance: bestanden)
- PCI DSS Compliance: Tests Requirements (PCI-DSS-Compliance: Anforderungstest)
- Unix Compliance Checks (UNIX-Compliancetests)
- Windows Compliance Checks (Windows-Compliancetests)
- Windows File Contents Compliance Checks (Compliancetests für Inhalte von Windows-Dateien)

Scanrichtlinie konfigurieren

Zum Aktivieren der Compliancetests in Nessus muss eine Scanrichtlinie mit den folgenden Attributen erstellt werden:

- Aktivieren Sie die Compliancetest-Plugins der Familie „Policy Compliance“ („Richtliniencompliance“).
- Geben Sie mindestens eine `.audit`-Compliancerichtlinie als Voreinstellung an.
- Geben Sie auf der Registerkarte „Preferences“ („Voreinstellungen“) die Anmeldedaten für den Zugriff auf den Zielservers an.
- Hierzu gehören ggf. auch die Datenbank anmeldedaten.

Diese Schritte können Sie mit dem Richtlinien-Assistenten und durch Auswählen des Assistenten „**Credentialed Patch Audit**“ („Authentifizierter Patch-Audit“) oder manuell über „**Advanced Policy**“ (Erweiterte Richtlinie) ausführen.



Es ist wichtig zu wissen, wie die in den ausgewählten `.audit`-Dateien vorhandenen Tests funktionieren, insbesondere wenn angepasste Dateien erstellt wurden. Wenn Sie zwei `.audit`-Dateien beim selben Scan verwenden, werden beide Dateien kombiniert, d. h., die Ergebnisse beider Dateien sind im selben Scan enthalten. Sollten Ergebnisse beider Dateien im Widerspruch zueinander stehen, dann erhalten Sie jeweils einen erfolgreichen und einen fehlgeschlagenen Test. Überprüfen Sie die Ergebnisse in Ihren Berichten stets sehr genau.

New Credentialed Patch Audit Policy / Step 1 of 2

1 Define your policy name, description, visibility, and post-scan editing preferences:

Policy Name

Visibility

Description

Allow Post-Scan Report Editing

[Next](#) [Cancel](#)

Rufen Sie zur Erstellung einer Scanrichtlinie die Nessus-Benutzeroberfläche auf, authentifizieren Sie sich und wählen dann „Policies“ („Richtlinien“) aus. Sie können dann eine vorhandene Richtlinie bearbeiten oder eine neue erstellen. Die Anmeldedaten für den Zugriff auf den Zielsystem können Sie links auf der Registerkarte „**Credentials**“ („Anmeldedaten“) angeben.

Aktivieren Sie die Plugin-Familie „Policy Compliance“ auf der Registerkarte „**Plugins**“ und vergewissern Sie sich, dass „`auto_enable_dependencies`“ in den erweiterten Einstellungen den Wert „**yes**“ („ja“) hat (dies ist die Standardeinstellung):

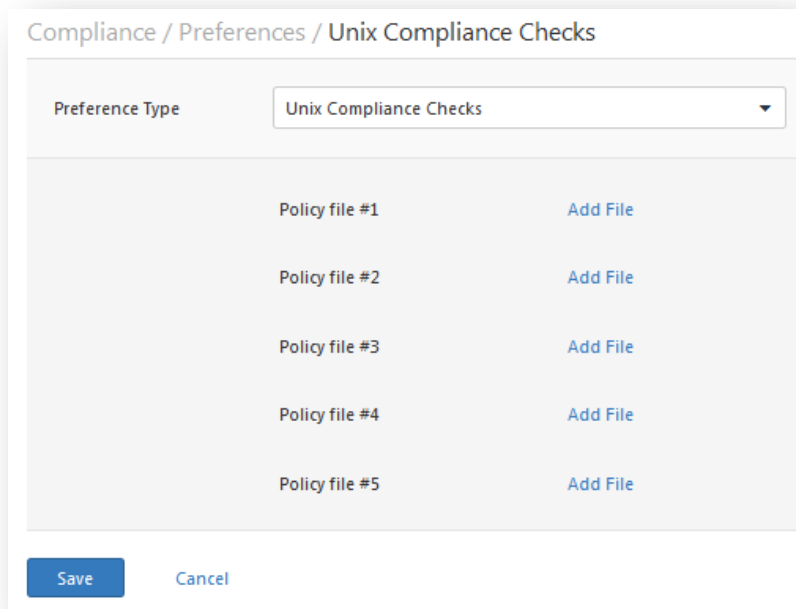
Compliance / Plugins [Show Enabled](#) | [Show All](#)

Status	Plugin Name	Plugin ID
DISABLED	Netware	14
DISABLED	Oracle Linux Local Security Checks	1560
DISABLED	Peer-To-Peer File Sharing	68
ENABLED	Policy Compliance	25
DISABLED	Red Hat Local Security Checks	2780
DISABLED	RPC	36
DISABLED	SCADA	147
DISABLED	Scientific Linux Local Security Ch...	1533
ENABLED	Check Point GAiA Compliance...	62679
ENABLED	Cisco IOS Compliance Checks	46689
ENABLED	Citrix XenServer Compliance C...	69512
ENABLED	Database Compliance Checks	33814
ENABLED	FireEye Compliance Checks	70469
ENABLED	HP ProCurve Compliance Che...	70271
ENABLED	IBM iSeries Compliance Checks	57860

[Save](#) [Cancel](#)

Eine Scanrichtlinie bearbeiten, um festzustellen, ob „Policy Compliance“ („Richtlinien-Compliance“) vorhanden ist

Um die Verwendung einer `.audit`-Datei zu ermöglichen, wählen Sie auf der Registerkarte „**Preferences**“ im Dropdownmenü einen der Einträge „Cisco IOS Compliance Checks“, „UNIX Compliance Checks“, „Windows Compliance Checks“, „Windows File Content Compliance Checks“, „IBM iSeries Compliance Checks“ oder „Database Compliance Checks“ aus. In jedem Bereich sind fünf Felder vorhanden, die die Angabe jeweils einer `.audit`-Datei ermöglichen. Die angegebenen Dateien müssen zuvor vom Tenable Support Portal auf das lokale Clientsystem heruntergeladen worden sein.



Compliance / Preferences / Unix Compliance Checks

Preference Type	
Unix Compliance Checks	
Policy file #1	Add File
Policy file #2	Add File
Policy file #3	Add File
Policy file #4	Add File
Policy file #5	Add File

[Save](#) [Cancel](#)

Dialogfeld zur Angabe von `.audit`-Dateien für UNIX auf der Nessus-Benutzeroberfläche (Beispiel)

Wenn „Database Compliance Checks“ im obigen Dropdownmenü ausgewählt wurde, müssen die Anmeldeparameter für die Datenbank unter „**Preferences**“ > „**Database Settings**“ („Datenbankeinstellungen“) angegeben werden:

Compliance / Preferences / Database settings

Preference Type: Database settings

Login:

Password:

DB Type: Oracle

Database SID:

Database port to use:

Oracle auth type: NORMAL

SQL Server auth type: Windows

Save Cancel

Es gibt unter „Database Settings“ („Datenbankeinstellungen“) eine Reihe von Optionen. Hierzu gehören:

Option	Beschreibung
Login (Anmeldename)	Der Benutzername für die Datenbank.
Password (Kennwort)	Das Kennwort zum angegebenen Benutzernamen.
DB Type (Datenbanktyp)	Oracle, SQL Server, MySQL, DB2, Informix/DRDA und PostgreSQL werden unterstützt.
Database SID (System-ID der Datenbank)	System-ID der zu überwachenden Datenbank. Nur gültig für Oracle, DB2 und Informix.
Oracle auth type (Oracle-Authentifizierungstyp)	NORMAL, SYSOPER und SYSDBA werden unterstützt.
SQL Server auth type (SQL Server-Authentifizierungstyp)	Windows und SQL Server werden unterstützt.

Wenden Sie sich an Ihren lokalen Datenbankadministrator, um die korrekten Werte für diese Felder zu erhalten.

Klicken Sie unten im Fenster auf „Save“ („Speichern“), um die Konfiguration abzuschließen. Die neue Scanrichtlinie wird nun der Liste der verwalteten Scanrichtlinien hinzugefügt.

Scan durchführen

Die Ausführung eines Scans, bei dem Compliancetests aktiviert sind, unterscheidet sich nicht von der Ausführung anderer lokaler Scans zum Prüfen von Patches oder auch normaler Netzwerkscans. Solche Scans lassen sich vielmehr kombinieren und abgleichen, um sie ggf. gleichzeitig ausführen zu können.

Ergebnisse (Beispiel)

In Nessus werden alle Compliance-Ergebnisse unter Angabe der ID des Plugins zurückgegeben, mit dem der Test ausgeführt wurde. Im nachfolgenden Beispiel stammen alle für einen gescannten Windows-Server zurückgegebenen Daten vom `.nbin`-Plugin „Windows Compliance“ (Plugin-ID 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

Compliance-Ergebnisse beim Scannen eines Windows-Servers (Beispiel)

Im HTML-Bericht, der über die Registerkarte „Reports“ („Berichte“) auf der Nessus-Benutzeroberfläche heruntergeladen werden kann, werden erfolgreiche Compliancetests blau und mit der Meldung „PASSED“ („Bestanden“) hervorgehoben. Nicht bestandene Tests hingegen werden rot und mit der Meldung „FAILED“ („Fehlgeschlagen“) hervorgehoben. Jegliche Elemente, für die kein Audit möglich war, sind gelb mit der Meldung „WARNING“ („Warnung“) hervorgehoben.

Im obigen Beispiel sind nur vier Elemente angezeigt. Alle diese Elemente entstammen einer Zugriffssteuerungsrichtlinie, die auf Vorhandensein nicht erforderlicher und unsicherer Dienste und Protokolle prüft. Einige dieser Dienste wurden nicht ausgeführt, was den Anforderungen der `.audit`-Richtlinie entsprach; andere hingegen (wie etwa der Dienst „Remote-Registrierung“) wurden ausgeführt, weswegen der Test hier den Status „FAILED“ erhalten hat. Wir empfehlen dringend, Elemente, die als „FAILED“ aufgeführt sind, so zu konfigurieren, dass sie die Anforderungen der Richtlinie erfüllen und folglich Ihren Sicherheitsstandards entsprechen.

Nessus-Befehlszeile für UNIX verwenden (Beispiel)

Compliancetests abrufen

Wenn Ihr Nessus-Daemon für den Empfang des ProfessionalFeed konfiguriert ist, befinden sich fünf `.nbin`-Dateien für Compliancetests in Ihrem Plugin-Verzeichnis.

Laden Sie sich ggf. benötigte `.audit`-Dateien vom Tenable Support Portal unter <https://support.tenable.com/> herunter und speichern Sie sie im Plugin-Verzeichnis Ihres Scanners. Bei den meisten Distributionen ist das folgende Verzeichnis der vorgegebene Speicherort:

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

Diese Plugins zählen zu den insgesamt mehr als 40.000 `.nas1`-Plugin-Dateien, die von Nessus zur Durchführung von Sicherheitslückenscans verwendet werden. Sie können wie nachfolgend gezeigt nach der Erweiterung `.nbin` suchen:

```
# 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
```

Unter Umständen sind weitere von Tenable bereitgestellte `.nbin`-Dateien vorhanden, die nicht für die Durchführung von Compliancetests verwendet werden (hierzu gehört etwa das Skype-Plugin).

Wenn Sie keinen lokalen Zugriff auf den eigentlichen Nessus-Daemon haben, sich aber mit einem Benutzernamen und einem Kennwort beim Server anmelden können, können Sie mithilfe der Option „-p“ des Befehlszeilenclients `nessus` wie folgt eine Liste mit den Plugins abrufen:

```
# /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
```

Die Ausführung der Abfrage kann mehrere Minuten dauern. Wenn Ihre Abfrage erfolgreich ausgeführt wird, aber keine Daten zurückgibt, dann sind die Compliancetests auf dem entfernten Nessus-Scanner nicht installiert.

.nessus-Dateien verwenden

Nessus kann konfigurierte Scanrichtlinien, Netzwerkziele und Berichte als `.nessus`-Datei speichern. Die Erstellung einer `.nessus`-Datei, die eine Scanrichtlinie für Compliancetests enthält, ist weiter oben im Abschnitt „[Nessus-Benutzeroberfläche verwenden \(Beispiel\)](#)“ beschrieben. Hinweise zur Ausführung eines Befehlszeilenscans mithilfe der `.nessus`-Datei entnehmen Sie dem Nessus-Benutzerhandbuch. Dieses ist verfügbar unter: <http://www.tenable.com/products/nessus/documentation>.

.nessusrc-Dateien verwenden

Der Nessus-Befehlszeilenclient bietet zudem die Möglichkeit, konfigurierte Scanrichtlinien als `.nessusrc`-Dateien zu exportieren. Dies kann nützlich sein, um das Scannen über die Befehlszeile zu ermöglichen. Die Erstellung einer Scanrichtlinie für Compliancetests in Nessus ist im Abschnitt „[Nessus-Benutzeroberfläche verwenden \(Beispiel\)](#)“ beschrieben.

Zum Aufruf eines Befehlszeilenscans mit Nessus müssen Sie folgende Angaben machen:

- Compliancetest-Plugins für UNIX, Windows oder Datenbanken
- Anmeldedaten für den oder die zu scannenden Zielhosts
- Mindestens eine `.audit`-Datei, damit die Compliancetest-Plugins ausgeführt werden können
- Eine Bestätigung, dass die Abhängigkeiten aktiviert wurden

Relevante Einträge in einer `.nessusrc`-Datei haben das folgende Format (einige Elemente wurden weggelassen):

```
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)
```

Im obigen Beispiel wurden viele weitere Datenelemente ausgelassen, mit denen angegeben werden kann, wie ein Scan durchgeführt wird. Zu den ausgelassenen Inhalten gehören die verwendete `.audit`-Richtliniendatei, die Aktivierung der Abhängigkeiten und die eigentlichen Compliancetest-Plugins.

Scan durchführen

Die Ausführung eines Scans, bei dem Compliancetests aktiviert sind, unterscheidet sich nicht von der Ausführung anderer lokaler Scans zum Prüfen von Patches oder auch normaler Netzwerkskans. Solche Scans lassen sich vielmehr kombinieren und abgleichen, um sie ggf. gleichzeitig ausführen zu können.

Ergebnisse (Beispiel)

Wie bei den GUI-Clients erscheinen alle erkannten Ergebnisse, die die Compliance-Anforderungen erfüllten bzw. nicht erfüllten, im folgenden Format im Bericht:

```
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\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\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\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\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\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\n\nRemote value: 30\nPolicy value: 60\n\n\n
```


Die Daten sind im `.nsr`-Berichtsformat für Nessus angegeben. Es handelt sich ausschließlich um Ereignisse, die die Compliancevorgaben nicht erfüllen.

SecurityCenter verwenden



Die nachfolgenden Angaben basieren auf der Ausführung von Compliancescans mit SecurityCenter 4 oder höher. Benutzer von Security Center 3.x finden weitere Informationen im Dokument „Security Center 3.4 Documentation“ („Dokumentation zu Security Center 3.4“). Dieses kann vom Tenable Support Portal heruntergeladen werden: <https://support.tenable.com/>.

Compliancetests abrufen

Alle SecurityCenter-Kunden haben Zugriff auf die Plugins kostenpflichtiger Nessus-Versionen. Hierzu gehören die Compliancetest-Plugins für Cisco, UNIX, Windows, Windows-Dateiinhalte und Datenbanken. Mithilfe dieser Plugins kann der Benutzer unter Verwendung der vorprogrammierten und anpassbaren `.audit`-Dateien, die von Tenable bereitgestellt werden, Compliancescans hochladen und ausführen. Laden Sie ggf. benötigte `.audit`-Dateien vom Tenable Support Portal unter <https://support.tenable.com/> herunter. Diese `.audit`-Dateien können dann von jedem Benutzer mit der Berechtigung „Create Audit Files“ („Auditdateien erstellen“) in SecurityCenter geladen werden. Hierzu dient die Funktion „Add Audit File“ („Auditdatei hinzufügen“) auf der Registerkarte „Support“.

Audit Files

Home Analysis Scanning Reporting Support Users Workflow Plugins

+ Add Audit File

Name Oracle Audit

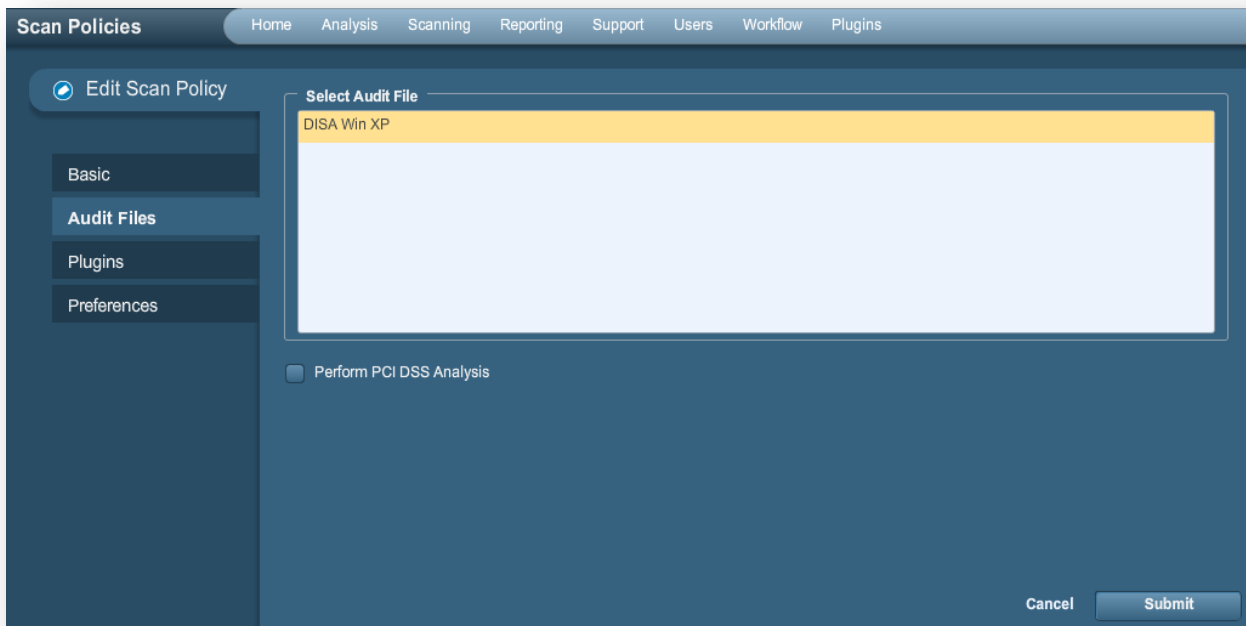
Description DISA v8 R1.2

File DISA_SRRchklist_Oracle_v8r1_2.audit Clear

Alle in SecurityCenter geladenen `.audit`-Dateien stehen jedem SecurityCenter-Benutzer zur Verfügung, der die Berechtigung „Create Policies“ („Richtlinien erstellen“) hat. SecurityCenter verwaltet auch die Verteilung neuer und geänderter `.audit`-Dateien an die Nessus-Scanner.

Scanrichtlinie zur Durchführung eines Compliance-Audits konfigurieren

Zur Durchführung eines Compliancescans mit SecurityCenter muss zunächst eine Scanrichtlinie mit geeigneten Compliance-spezifischen Einstellungen erstellt werden. Diese Richtlinie gibt die Scanoptionen, Auditdateien, aktivierten Plugins und erweiterten Einstellungen an. Auf der zweiten Seite von „Scan Policy“ („Scanrichtlinie“) werden dann die `.audit`-Dateien festgelegt, die für den Compliance-Audit verwendet werden sollen.



Hier muss mindestens eine `.audit`-Datei ausgewählt werden. Klicken Sie auf die gewünschten `.audit`-Dateien und abschließend auf „Submit“ („Senden“). Halten Sie zur Auswahl mehrerer `.audit`-Dateien die STRG-Taste gedrückt. Wenn eine einfache PCI-DSS-Analyse erforderlich ist, stellen Sie vor dem Versand sicher, dass das Kontrollkästchen „Perform PCI DSS Analysis“ („PCI-DSS-Analyse durchführen“) aktiviert ist.

PCI DSS (Payment Card Industry Data Security Standard) ist ein umfangreicher Satz mit Sicherheitsstandards, die von den Gründungsmitgliedern des PCI Security Standards Council (Visa, American Express, Discover Financial Services und MasterCard) definiert wurden. PCI DSS ist als allgemeine Mindestanforderung für den Schutz von Kreditkartendaten aller Marken gedacht und wird von vielen E-Commerce-Anbietern verwendet, die Kreditkartenzahlungen anbieten und entsprechende Daten speichern.

Tenable stellt für alle SecurityCenter-Benutzer zwölf Plugins bereit, die die Abläufe bei einem PCI-DSS-Audit automatisieren. Eine Liste dieser Plugins finden Sie in folgender Tabelle.

Mit diesen Plugins werden die Ergebnisse und die tatsächliche Konfiguration des Scans ausgewertet, um festzustellen, ob der Zielsystem die publizierten PCI-Compliance-Anforderungen erfüllt. Die Plugins führen den Scan nicht selbst aus, sondern überprüfen die Ergebnisse anderer Plugins. Markieren Sie zur Aktivierung der PCI-DSS-Plugins einfach das Kontrollkästchen „Perform PCI DSS Analysis“ im Fenster „Compliance“.

Klicken Sie nach Auswahl der gewünschten `.audit`-Datei(en) und PCI-DSS-Einstellungen auf die Registerkarte „Plugins“, um die Plugin-Einstellungen zu bestätigen. Elemente der Plugin-Familie „Policy Compliance“ müssen in der Richtlinie aktiviert sein, damit ein Compliancescan ausgeführt wird.



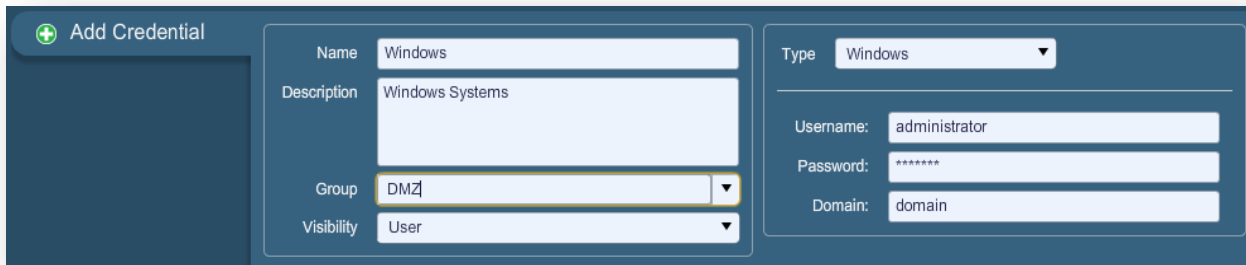
Wenn der Benutzer mindestens eine Auditdatei auf der Registerkarte „Audit Files“ („Auditdateien“) der Scanrichtlinie ausgewählt hat, wird das korrekte Plugin auf der Registerkarte „Plugins“ automatisch aktiviert. SecurityCenter analysiert die ausgewählten `.audit`-Dateien. Basierend auf dem in der jeweiligen Datei angegebenen Typ werden dann das oder die entsprechenden Plugins aktiviert.

Die Familie „Policy Compliance“ enthält dreizehn Plugins für Compliance-Audits. Hierzu gehören:

Plugin-ID	Plugin-Name	Plugin-Beschreibung
21156	Windows Compliance Checks (Windows-Compliancetests)	Wird für Audits allgemeiner Windows-Konfigurationseinstellungen verwendet.
21157	Unix Compliance Checks (UNIX-Compliancetests)	Wird für Audits allgemeiner UNIX-Konfigurationseinstellungen verwendet.
24760	Windows File Contents Compliance Checks (Compliancetests für Inhalte von Windows-Dateien)	Wird für Audits sensibler Dateiinhalte auf Windows-Servern verwendet.
33814	Database Compliance Checks (Datenbank-Compliancetests)	Wird für Audits allgemeiner Datenbankkonfigurationseinstellungen verwendet.
33929	PCI DSS Compliance (PCI-DSS-Compliance)	Ermittelt, ob der Remotewebserver anfällig für XSS-Angriffe (Cross-Site-Scripting) ist, die veraltete SSL 2.0-Verschlüsselung implementiert, veraltete Software ausführt oder gefährliche Sicherheitslücken aufweist (CVSS-Basiswert ≥ 4).
57581	PCI DSS Compliance: Database Reachable from the Internet (PCI-DSS-Compliance: Datenbank über das Internet erreichbar)	Erkennt das Vorhandensein einer Datenbank, die über das Internet aufgerufen werden kann, was zum Fehlschlagen des Compliance-Audits führt.
60020	PCI DSS Compliance: Handling False Positives (PCI-DSS-Compliance: Umgang mit Fehlalarmen)	Beschreibt die sachgemäße Handhabung von Fehlalarmen in PCI-DSS-Scans.
56208	PCI DSS Compliance: Insecure Communication Has Been Detected (PCI-DSS-Compliance: Unsichere Kommunikation erkannt)	Ermittelt, ob ein nicht gesicherter Port oder Service oder ein nicht gesichertes Protokoll erkannt wurden, was das Fehlschlagen der Compliance bewirken würde.
56209	PCI DSS Compliance: Remote Access Software Has Been Detected (PCI-DSS-Compliance: Software mit Remotezugriff erkannt)	Erkennt das Vorhandensein von Software mit Remotezugriff, was das Fehlschlagen der Compliance bewirken würde.
33930	PCI DSS Compliance: Passed (PCI-DSS-Compliance: bestanden)	Auf der Basis der vorhandenen Scaninformationen konnte Nessus keine disqualifizierenden Schwachstellen auf diesem Host feststellen.
33931	PCI DSS Compliance: Tests Requirements (PCI-DSS-Compliance: Anforderungstest)	Analysiert, ob der Nessus-Scan die Anforderungen des PCI-Tests erfüllt. Auch bei bestandenem Test ist dieser Bericht für eine Zertifizierung des Servers nicht ausreichend.
46689	Cisco IOS Compliance Checks (Cisco IOS-Compliancetests)	Wird für Audits allgemeiner Konfigurationseinstellungen von Cisco-Geräten verwendet.
57860	IBM iSeries Compliance Checks (Compliancetests für IBM iSeries-Systeme)	Wird für Audits allgemeiner IBM iSeries-Konfigurationseinstellungen verwendet.

Anmeldedaten verwalten

Ein Vorteil von SecurityCenter bei der Durchführung authentifizierter Scans besteht in der Möglichkeit, die verwendeten Anmeldedaten zu verwalten. Anmeldedaten werden in SecurityCenter auf der Registerkarte „Support“ erstellt. Klicken Sie dort auf „Credentials“ („Anmeldedaten“) und dann auf „Add“ („Hinzufügen“).



Add Credential

Name: Windows

Description: Windows Systems

Group: DMZ

Visibility: User

Type: Windows

Username: administrator

Password: *****

Domain: domain

UNIX-, Windows- und Cisco-Anmeldedaten werden getrennt von der Scanrichtlinie gespeichert und verwaltet. Anmeldedaten können auf der Ebene „User“ („Benutzer“) für den aktuellen Benutzer oder auf der Ebene „Organizational“ („Organisation“) erstellt werden, auf der sie auch von anderen SecurityCenter-Benutzern verwendet werden können. Auf diese Weise können Benutzer mit den Scanergebnissen arbeiten und neue Scans durchführen, ohne selbst wissen zu müssen, wie die für Scans verwendeten Anmeldedaten lauten.

Für das Scannen von Datenbanksystemen sind zusätzliche Anmeldedaten erforderlich. Diese werden in der Scanrichtlinie gespeichert und in den Voreinstellungen für die Scanrichtlinie über das Plugin „Database settings“ („Datenbankeinstellungen“, Plugin-ID 33815) konfiguriert. Die Konfiguration erfolgt getrennt von den Anmeldedaten, die im vorigen Absatz beschrieben wurden.

Ergebnisse analysieren

SecurityCenter kann in vielfältiger Weise zur Analyse von Compiancedaten, die von Nessus-Scans zurückgegeben wurden, und zur Erstellung entsprechender Berichte verwendet werden. Die folgenden Berichte treten häufig auf:

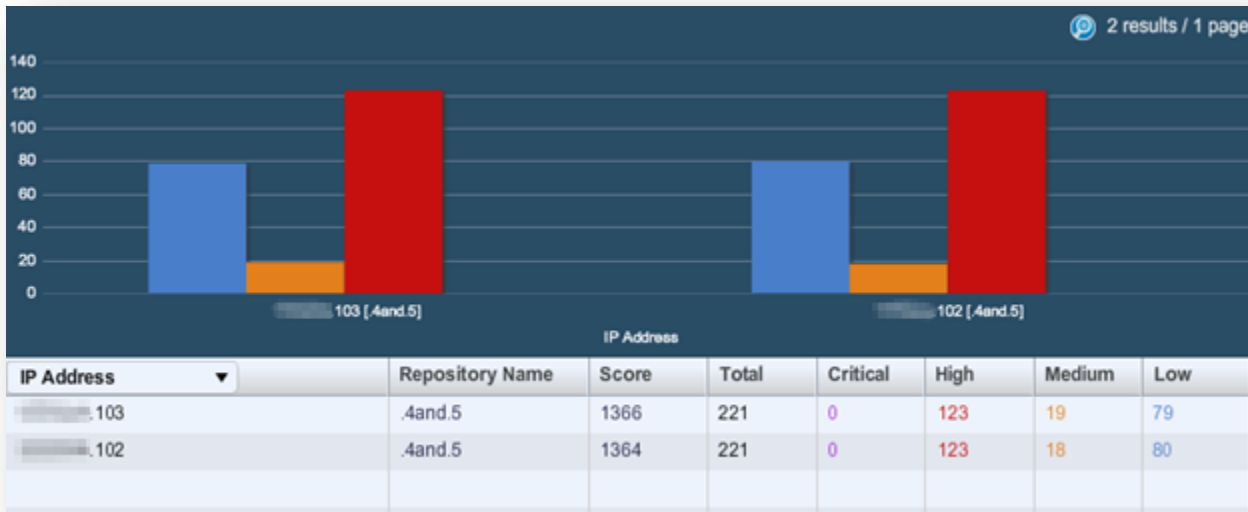
- Auflistung aller Sicherheitslücken, die die Compliance-Anforderungen erfüllen bzw. nicht erfüllen, nach Assetgruppe
- Auflistung aller Sicherheitslücken, die die Compliance-Anforderungen erfüllen bzw. nicht erfüllen, nach Host oder Netzwerk
- Zusammenfassung aller Elemente, die die Compliancevorgaben nicht erfüllen
- Prüfung von Datenbankeinstellungen auf häufige Konfigurationsfehler
- Benutzer- oder Softwarestatus nach IT-Anforderungen

Wenn die Compiancedaten durch SecurityCenter ermittelt wurden, kann mithilfe der Ticket-, Berichterstellungs- und Analysetools die optimale Vorgehensweise für die Neukonfiguration der geprüften Geräte bestimmt werden. Diese Daten lassen sich parallel mit anderen ermittelten Informationen zu Sicherheitslücken, Patches und passiv ermittelten Daten analysieren.

Nachfolgend gezeigt sind einige SecurityCenter-Beispielabbildungen, die bei der Analyse von Compiancedaten zu gescannten Hosts erstellt wurden:

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\NfsDisable8dot3NameCreation
1000281	4	High	HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon\ScRemoveOption
1000280	4	High	HKLM\System\CurrentControlSet\Control\Lsa\CompatibilityLevel
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\EnableCMPRedirect

Mit SecurityCenter aufgezeichnete Compliance-Auditdaten (Beispiel)



Mit SecurityCenter aufgezeichnete Compliance-Auditdaten, nach Server sortiert (Beispiel)

Weitere Informationen zur Verwendung von SecurityCenter finden Sie in der zugehörigen Dokumentation. Diese ist unter <https://support.tenable.com/> erhältlich.

Weitere Informationen

Tenable hat eine Reihe von Dokumenten erstellt, in denen die Bereitstellung, Installation, Konfiguration, der Betrieb und die Testmethoden von Nessus ausführlich beschrieben werden.

- **Nessus 5.2 Installation and Configuration Guide** („Nessus 5.2-Installations- und Konfigurationshandbuch“; Schrittanleitung zur Nessus-Installation und -Konfiguration)
- **Nessus 5.2 User Guide** („Nessus 5.2-Benutzerhandbuch“; beschreibt den Einsatz des Nessus Nessus-Sicherheitslückenscanners einschließlich Konfiguration und Berichterstellung)
- **Nessus Credential Checks for Unix and Windows** („Authentifizierte Nessus-Tests für UNIX und Windows“; enthält Informationen zur Durchführung authentifizierter Netzwerkskans mit dem Nessus-Sicherheitslückenscanner)
- **Nessus Compliance Checks Reference** („Nessus-Referenzhandbuch für Compliancetests“; umfassender Leitfaden zur Syntax von Nessus-Compliancetests)
- **Nessus v2 File Format** („Nessus V2-Dateiformat“; beschreibt die Struktur des `.nessus`-Dateiformats, das mit Nessus 3.2 und NessusClient 3.2 eingeführt wurde)
- **Nessus 5.0 REST Protocol Specification** („Nessus 5.0 REST-Protokollspezifikation“; beschreibt das REST-Protokoll und die Schnittstelle in Nessus)
- **Nessus 5 and Antivirus** („Nessus 5 und Virenschutz“; beschreibt die Funktion verschiedener gängiger Sicherheitssoftwarepakete in Nessus und enthält Tipps und Lösungsvorschläge für eine verbesserte Funktionsweise der Software ohne Einschränkung der Sicherheit oder Verhinderung Ihrer Sicherheitslückenscans)

- **Nessus 5 and Mobile Device Scanning** („Nessus 5 und Scans von Mobilgeräten“; beschreibt die Integration von Nessus in Microsoft Active Directory und Verwaltungsserver für Mobilgeräte zur Bestimmung von im Netzwerk eingesetzten Mobilgeräten)
- **Nessus 5.0 and Scanning Virtual Machines** („Nessus 5.0 und Scans virtueller Maschinen“; beschreibt den Einsatz des Sicherheitslückenscanners von Tenable Network Security Nessus für Audits der Konfiguration virtueller Plattformen sowie der darauf ausgeführten Software)
- **Strategic Anti-malware Monitoring with Nessus, PVS, and LCE** („Strategische Malwareüberwachung mit Nessus, PVS und LCE“; beschreibt, wie mithilfe der Tenable USM-Plattform zahlreiche bösartige Softwareprogramme erkannt werden können und das Ausmaß der Malware-Infizierung bestimmt werden kann)
- **Patch Management Integration** („Integration des Patchmanagements“; beschreibt, wie Nessus und SecurityCenter mithilfe von Berechtigungen auf die IBM TEM-, Microsoft WSUS- und SCCM-, VMware Go- und Red Hat Network Satellite-Patchmanagementsysteme Patch-Audits auf Systemen ausführen, für die dem Nessus-Scanner möglicherweise keine Berechtigungen zur Verfügung stehen)
- **Real-Time Compliance Monitoring** („Compliance-Überwachung in Echtzeit“; erläutert, wie die Lösungen von Tenable Sie bei der Erfüllung zahlreicher gesetzlicher Vorschriften und Finanzstandards unterstützen)
- **Tenable Products Plugin Families** („Tenable Produkt-Plugin-Familien“; stellt eine Beschreibung und Zusammenfassung der Plugin-Serien für Nessus, Log Correlation Engine und den Passive Vulnerability Scanner bereit)
- **SecurityCenter Administration Guide** („SecurityCenter-Administratorhandbuch“)

Weitere Onlineresourcen sind nachfolgend aufgeführt:

- Nessus-Diskussionsforum: <https://discussions.nessus.org/>
- Tenable-Blog: <http://www.tenable.com/blog>
- Tenable-Podcast: <http://www.tenable.com/podcast>
- Beispielvideos zum Gebrauch: <http://www.youtube.com/user/tenablesecurity>
- Tenable-Twitterfeed: <http://twitter.com/tenablesecurity>

Setzen Sie sich mit uns in Verbindung – via E-Mail (support@tenable.com, sales@tenable.com) oder über unsere Website unter <http://www.tenable.com/>.

Wissenswertes zu Tenable Network Security

Wenn es um die frühzeitige Erkennung neu entwickelter Sicherheitslücken, Bedrohungen und Compliance-relevanter Risiken geht, verlassen sich mehr als 20.000 Organisationen auf Tenable Network Security. Hierzu gehören neben dem gesamten US-Verteidigungsministerium eine Reihe von Großunternehmen und Regierungsbehörden weltweit. Die Nessus- und SecurityCenter-Lösungen sind nach wie vor branchenführend beim Ermitteln von Sicherheitslücken, beim Verhindern von Angriffen und bei der Erfüllung einer Vielzahl gesetzlicher Vorschriften. Weitere Informationen finden Sie unter www.tenable.com.

Globale Unternehmenszentrale

Tenable Network Security

7021 Columbia Gateway Drive

Suite 500

Columbia, MD 21046, USA

+1.410.872.0555

www.tenable.com

