Справочник по проверкам соответствия Nessus

14 июня 2011 г.

(редакция 6)

Авторские права © 2011 г. Tenable Network Security, Inc. Все права защищены. Tenable Network Security и Nessus являются зарегистрированными товарными знаками компании Tenable Network Security, Inc. ProfessionalFeed является товарным знаком компании Tenable Network Security, Inc. Наименования всех прочих продуктов и услуг являются товарными знаками соответствующих владельцев.

Tenable Network Security, Inc. • 7063 Columbia Gateway Drive, Suite 100, Columbia, MD 21046 • 410.872.0555 • sales@tenable.com • www.tenable.com

Содержание

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

ВВЕДЕНИЕ

В этом документе описывается синтаксис, применяемый для создания пользовательских **.audit** файлов, которые можно использовать для проверки конфигураций ОС Unix, Windows, базы данных, систем SCADA и Cisco на соблюдение политики соответствия, а также выполнения поиска информации, которая может являться конфиденциальной, в содержимом различных систем.

Настоящее руководство предназначено для того, чтобы помочь научиться вручную создавать и понять синтаксис файлов проверки соответствия. Более высокоуровневое рассмотрение работы проверки соответствия Tenable см. в посвященном проверкам соответствия Nessus документе, размещенном в формате PDF на портале поддержки Tenable Support Portal.

Nessus поддерживает проверку системы SCADA, однако эта функциональная возможность выходит за рамки настоящего документа. Дополнительные сведения см. на информационной странице Nessus.org по SCADA, размещенной здесь.

Необходимые условия

Изучение настоящего документа предполагает определенный уровень знаний о сканере уязвимостей Nessus, а также глубокое понимание проверяемых систем. Для получения дополнительных сведений о том, как можно настроить Nessus для выполнения аудита исправлений локальных систем Unix и Windows см. документ «Nessus Credentials Checks for Unix and Windows» (Проверки Nessus с использованием учетных данных для Unix и Windows) http://www.nessus.org/documentation/.

Стандарты и условные обозначения

Этот документ является переводом оригинальной версии на английском языке. Часть текста остается на английском языке, чтобы показать, как этот текст представлен в программном продукте.

В рамках всей документации имена файлов, демонов и исполняемых модулей выделены шрифтом **courier bold**.

Параметры и ключевые слова командной строки также выделены шрифтом **courier bold**. Параметры командной строки могут включать или не включать приглашение командной строки и выводимый в результате выполнения команды текст. Часто выполняемая команда приводится **жирным шрифтом**, чтобы выделить набираемый пользователем текст. Ниже приведен пример выполнения команды Unix **pwd**.

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

Этим символом и рамкой с серым фоном выделены важные примечания и соображения.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Этим символом и рамкой с синим фоном и белым текстом выделены советы, примеры и оптимальные методы.

СПРАВКА ПО ФАЙЛАМ ПРОВЕРКИ СООТВЕТСТВИЯ КОНФИГУРАЦИИ ОС WINDOWS .AUDIT

Основой для файлов проверки соответствия для ОС Windows **`.audit`** является специально отформатированный текстовый файл. Включенные в такой файл записи могут вызывать выполнение проверок различных «пользовательских элементов», например настроек реестра, а также более общих проверок, например параметров локальных политик безопасности. Примеры в настоящем руководстве приведены для пояснения.

Использование кавычек

Одинарные и двойные кавычки при заключении в них полей аудита являются взаимозаменяемыми, за исключением следующих двух случаев:

1. В проверках соответствия операционной системы Windows, если специальные поля, например CRLF и т. д., должны интерпретироваться буквально, используйте одинарные кавычки. Любые оказавшиеся внутри этих кавычек поля, которые должны интерпретироваться как строки, следует выделять.

Например:

```
expect: 'First line\r\nSecond line\r\nJohn\'s Line'
```

2. Двойные кавычки требуются при использовании полей include_paths и exclude_paths аудита WindowsFiles.

При использовании в поле любого типа (description, value_data, regex и т. д.) строк, содержащих одинарные или двойные кавычки, существует два способа их обработки:

a. В качестве наружных кавычек можно использовать кавычки противоположного типа.

Например:

```
expect: "This is John's Line"
expect: 'We are looking for a double-quote-".*'
```

b. Можно выделять внутренние кавычки с помощью обратной наклонной черты (только двойные кавычки).

Например:

```
expect: "\"Text to be searched\""
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Тип проверки

Все проверки соответствия для ОС Windows должны быть инкапсулированы путем заключения между элементами **check_type** со значением Windows, а также указанием версии 2:

```
<check_type:"Windows" version:"2">
```

Пример проверки соответствия для ОС Windows можно посмотреть в разделе «Приложение B». Проверка начинается с параметра **check_type**, имеющего значение Windows и версию 2, и заканчивается тегом **</check_type>**.

Это необходимо для того, чтобы можно было отличать файлы для ОС Windows **.audit** от файлов, предназначенных для ОС Unix (или какой-либо другой платформы).

Данные значения

Синтаксис файла **.audit** содержит ключевые слова, которые могут быть присвоены различным типам значений с целью индивидуализации проверок. В настоящем разделе описываются эти ключевые слова и формат данных, которые могут быть введены.

Типы данных

Следующие типы данных могут быть введены для проведения проверок:

Типы данных	Описание
DWORD	От 0 до 2 147 483 647
RANGE [X..Y]	где X — это значение типа DWORD или MIN , а Y — это значение типа DWORD или MAX

Примеры:

```
value_data: 45
value_data: [11..9841]
value_data: [45..MAX]
```

Кроме того, числовые значения могут отмечаться символами плюс (+) и минус (-) для обозначения их «знака» и признания шестнадцатеричными значениями. Шестнадцатеричное значение и знаки могут быть объединены. Следующие примеры являются допустимыми (без соответствующей метки в скобках) в аудите REGISTRY_SETTING для POLICY_DWORD:

```
value_data: -1  (signed)
value_data: +10  (signed)
value_data: 10  (unsigned)
value data: 2401649476  (unsigned)
value_data: [MIN..+10]  (signed range)
value_data: [20..MAX]  (unsigned range)
value_data: 0x800010AB  (unsigned hex)
value_data: -0x10  (signed hex)
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Сложные выражения

Сложные выражения могут использоваться для поля `value_data` путем использования символов:

- ||: условное ИЛИ
- &&: условное И
- |: двоичное ИЛИ (побитовая операция)
- &: двоичное И (побитовая операция)
- (и): для разграничения сложных выражений

Примеры:

```
value_data: 45 || 10
value_data: (45 || 10) && ([9..12] || 37)
```

Поле check_type

Этот тип проверки отличается от описанного выше поля **`check_type`**, которое используется в начале каждого файла аудита для обозначения общего типа аудита (Windows, WindowsFiles, Unix, Database, Cisco). Это поле является необязательным и может применяться к значениям **`value_data`** ОС Windows для определения типа выполняемой проверки. Возможны следующие значения:

- CHECK_EQUAL: сравнить удаленное значение со значением из политики (в случае отсутствия **`check_type`** используется по умолчанию)
- CHECK_EQUAL_ANY: проверяет, присутствует ли каждый элемент **`value_data`** хотя бы раз в списке системы
- CHECK_NOT_EQUAL: проверяет, отличается ли удаленное значение от значения из политики
- CHECK_GREATER_THAN: проверяет, действительно ли удаленное значение больше, чем значение из политики
- CHECK_GREATER_THAN_OR_EQUAL: проверяет, действительно ли удаленное значение больше или равно значению из политики
- CHECK_LESS_THAN: проверяет, действительно ли удаленное значение меньше, чем значение из политики
- CHECK_LESS_THAN_OR_EQUAL: проверяет, действительно ли удаленное значение меньше или равно значению из политики
- CHECK_REGEX: проверяет, соответствует ли удаленное значение регулярному выражению, содержащемуся в значении из политики (действует только с полями POLICY_TEXT и POLICY_MULTI_TEXT)
- CHECK_SUBSET: проверяет, является ли удаленный список ACL составной частью списка ACL политики (эта проверка работает только со списками ACL)
- CHECK_SUPERSET: проверяет, включает ли удаленный список ACL список ACL политики (работает только с лишающими прав списками ACL)

Ниже приведен пример аудита, позволяющего убедиться, что имени учетной записи «Guest» (гость) нет среди учетных записей типа Guest.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
<custom_item>
 type: CHECK_ACCOUNT
 description: "Accounts: Rename guest account"
 value_type: POLICY_TEXT
 value_data: "Guest"
 account_type: GUEST_ACCOUNT
 check_type: CHECK_NOT_EQUAL
</custom_item>
```

В случае любого другого значения, кроме Guest (гость), тест успешно пройдет. Если будет найдено значение Guest, то аудит будет не пройден.

Поле info

Необязательное поле **info** может использоваться для маркировки каждого поля аудита одной или несколькими внешними ссылками. Например, это поле будет использоваться для включения ссылок из тегов CCE NIST, а также особых требований CIS к аудиту. Эти внешние ссылки распечатываются при окончательном аудите, выполняемом Nessus, и отображаются в отчете Nessus или через интерфейс SecurityCenter.

Ниже приведен пример политики аудита паролей, которая была дополнена включением ссылки на вымышленную корпоративную политику:

```
<custom_item>
 type: PASSWORD_POLICY
 description: "Password History: 24 passwords remembered"
 value_type: POLICY_DWORD
 value_data: [22..MAX] || 20
 password_policy: ENFORCE_PASSWORD_HISTORY
 info: "Corporate Policy 102-A"
</custom_item>
```

Если в один аудит необходимо включить несколько ссылок на политики, то в строке, обозначенной ключевым словом **info**, можно использовать разделитель \n для обозначения нескольких строк. Например, рассмотрите следующий аудит:

```
<custom_item>
 type: CHECK_ACCOUNT
 description: "Accounts: Rename Administrator account"
 value_type: POLICY_TEXT
 value_data: "Administrator"
 account_type: ADMINISTRATOR_ACCOUNT
 check_type: CHECK_NOT_EQUAL
 info: 'Ron Gula Mambo Number 5\nCCE-60\nTenable Best Practices Policy
       1005-a'
</custom_item>
```

При запуске с помощью средств командной строки **nasl** эта функция аудита выдает следующий результат:

```
# /opt/nessus/bin/nasl -t 192.168.20.16 ./compliance_check.nbin
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
Windows Compliance Checks, version 2.0.0

Which file contains your security policy : ./test_v2.audit
SMB login : Administrator
SMB password :
SMB domain (optional) :
"Accounts: Rename Administrator account": [FAILED]

Ron Gula Mambo Number 5
CCE-60
Tenable Best Practices Policy 1005-a

Remote value: "Administrator"
Policy value: "administrator"
```

Формат списка ACL

В этом разделе описывается синтаксис, используемый для определения, имеет ли файл или папка желаемые настройки в ACL.

Проверки контроля доступа к файлам

Использование

`<file_acl: ["name"]>`
 	
`<user: ["user_name"]>`
`acl_inheritance: ["value"]`
`acl_apply: ["value"]`
`(optional) acl_allow: ["rights value"]`
`(optional) acl_deny: ["rights value"]`
`</user>`
 	
`</acl>`

Список контроля доступа (ACL) для файлов определяется ключевым словом **`file_acl`**. Имя списка ACL должно быть уникальным, чтобы его можно было использовать с элементом разрешений для файлов. Список ACL для файлов может содержать одну или несколько пользовательских записей.

Связанные типы	Допустимые типы
`acl_inheritance`	> not inherited (не наследуется) > inherited (наследуется) > not used (не используется)
`acl_apply`	> this folder only (только эта папка) > this object only (только этот объект) > this folder and files (эта папка и файлы)

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	> this folder and subfolders (эта папка и подпапки) > this folder, subfolders and files (эта папка, подпапки и файлы) > files only (только файлы) > subfolders only (только подпапки) > subfolders and files only (только подпапки и файлы)
acl_allow **acl_deny**	Эти параметры являются необязательными. Общие права следующие: > full control (полный контроль) > modify (изменять) > read & execute (читать и исполнять) > read (читать) > write (записывать) > list folder contents (получать список содержимого папки) Расширенные права следующие: > full control (полный контроль) > traverse folder / execute file (проходить папку / исполнять файл) > list folder / read data (получать список содержимого папки / читать данные) > read attributes (читать атрибуты) > read extended attributes (читать дополнительные атрибуты) > create files / write data (создавать файлы / записывать данные) > create folders / append data (создавать папки / добавлять данные) > write attributes (записывать атрибуты) > write extended attributes (записывать дополнительные атрибуты) > delete subfolder and files (удалять подпапку и файлы) > delete (удалять) > read permissions (читать разрешения) > change permissions (изменять разрешения) > take ownership (становиться владельцем)

Ниже приведен пример текста контроля доступа к файлам из файла **.audit**:

```
<file_acl: "ASU1">
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
  <user: "Administrators">
   acl_inheritance: "not inherited"
   acl_apply: "This folder, subfolders and files"
   acl_allow: "Full Control"
  </user>

  <user: "System">
   acl_inheritance: "not inherited"
   acl_apply: "This folder, subfolders and files"
   acl_allow: "Full Control"
  </user>

  <user: "Users">
   acl_inheritance: "not inherited"
   acl_apply: "this folder only"
   acl_allow: "list folder / read data" | "read attributes" | "read
        extended attributes" | "create files / write data" | "create
        folders / append data" | "write attributes" | "write extended
        attributes" | "read permissions"
  </user>

 </acl>
```

Проверки контроля доступа к реестру

Использование

`<registry_acl: ["name"]>`
 	
` <user: ["user_name"]>`
`  acl_inheritance: ["value"]`
`  acl_apply: ["value"]`
`  (optional) acl allow: ["rights value"]`
`  (optional) acl_deny: ["rights value"]`
` </user>`
 	
`</acl>`

Список ACL для реестра определяется ключевым словом **`registry_acl`**. Имя списка ACL должно быть уникальным, чтобы его можно было использовать с элементом разрешений для реестра. Список ACL для реестра может содержать одну или несколько пользовательских записей.

Связанные типы	Допустимые типы
`acl_inheritance`	> not inherited (не наследуется) > inherited (наследуется) > not used (не используется)

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

acl_apply	> this key only (только этот раздел) > this key and subkeys (этот раздел и подразделы) > subkeys only (только подразделы)
acl_allow **acl_deny**	Эти параметры являются необязательными и используются для определения прав, которые пользователь имеет применительно к объекту. Общие права следующие: > full control (полный контроль) > read (читать) Расширенные права следующие: > full control (полный контроль) > query value (запрашивать значение) > set value (задавать значение) > create subkey (создавать подраздел) > enumerate subkeys (создавать перечисления подразделов) > notify (уведомлять) > create link (создавать ссылку) > delete (удалять) > write dac (записывать в список контроля доступа на уровне пользователей) > write owner (записывать владельца) > read control (читать список контроля)

Ниже приведен пример текста списка контроля доступа к реестру из файла **.audit**:

```
<registry_acl: "SOFTWARE ACL">

 <user: "Administrators">
  acl_inheritance: "not inherited"
  acl_apply: "This key and subkeys"
  acl_allow: "Full Control"
 </user>

 <user: "CREATOR OWNER">
  acl_inheritance: "not inherited"
  acl_apply: "Subkeys only"
  acl_allow: "Full Control"
 </user>

 <user: "SYSTEM">
  acl_inheritance: "not inherited"
  acl_apply: "This key and subkeys"
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
    acl_allow: "Full Control"
   </user>

   <user: "Users">
    acl_inheritance: "not inherited"
    acl_apply: "This key and subkeys"
    acl_allow: "Read"
   </user>

  </acl>
```

Проверки контроля доступа к службам

Использование

`<service_acl: ["name"]>`
 	
`<user: ["user_name"]>`
`acl_inheritance: ["value"]`
`acl_apply: ["value"]`
`(optional) acl_allow: ["rights value"]`
`(optional) acl_deny: ["rights value"]`
`</user>`
 	
`</acl>`

Список ACL для служб определяется ключевым словом **service_acl**. Имя списка ACL должно быть уникальным, чтобы его можно было использовать с элементом разрешений для служб. Список ACL для служб может содержать одну или несколько пользовательских записей.

Связанные типы	Допустимые типы
acl_inheritance	> not inherited (не наследуется) > inherited (наследуется) > not used (не используется)
acl_apply	> this object only (только этот объект)
acl_allow **acl_deny**	Эти параметры являются необязательными и используются для определения прав, которые пользователь имеет применительно к объекту. Общие права следующие: > full control (полный контроль) > read (читать) > start, stop and pause (запускать, останавливать и приостанавливать)

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	> write (записывать) > delete (удалять) Расширенные права следующие: > full control (полный контроль) > delete (удалять) > query template (отправлять запросы к шаблону) > change template (изменять шаблон) > query status (отправлять запросы статуса) > enumerate dependents (создавать перечисление зависимых) > start (запускать) > stop (останавливать) > pause and continue (приостанавливать и продолжать) > interrogate (опрашивать) > user-defined control (определенное пользователем управление) > read permissions (читать разрешения) > change permissions (изменять разрешения) > take ownership (становиться владельцем)

Пример проверки контроля доступа к службам приведен ниже:

```
 <service_acl: "ALERT ACL">

  <user: "Administrators">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "query template" | "change template" | "query status" |
        "enumerate dependents" | "start" | "stop" | "pause and continue" |
        "interrogate" | "user-defined control" | "delete" | "read
        permissions" | "change permissions" | "take ownership"
  </user>

 </acl>
```

Проверки контроля разрешения запуска Launch

Использование

`<launch_acl: ["name"]>`
 	
`<user: ["user_name"]>`
`acl_inheritance: ["value"]`
`acl_apply: ["value"]`
`(optional) acl_allow: ["rights value"]`

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
  (optional) acl_deny: ["rights value"]
 </user>

</acl>
```

Список launch ACL определяется ключевым словом `launch_acl`. Имя списка ACL должно быть уникальным, чтобы его можно было использовать с элементом прав запуска DCOM. Список launch ACL может содержать одну или несколько пользовательских записей.

Связанные типы	Допустимые типы
acl_inheritance	> not inherited (не наследуется) > inherited (наследуется)
acl_apply	> this object only (только этот объект)
acl_allow **acl_deny**	Эти параметры являются необязательными и используются для определения прав, которые пользователь имеет применительно к объекту. Общие права следующие: > local launch (локальный запуск) > remote launch (удаленный запуск) > local activation (локальная активация) > remote activation (удаленная активация)

Этот список ACL работает только для ОС Windows XP/2003/Vista (и частично для ОС Windows 2000).

Пример проверки контроля доступа для операций launch (запуск) приведен ниже:

```
 <launch_acl: "2">

  <user: "Administrators">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "Remote Activation"
  </user>

  <user: "INTERACTIVE">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "Local Activation" | "Local Launch"
  </user>

  <user: "SYSTEM">
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
    acl_inheritance: "not inherited"
    acl_apply: "This object only"
    acl_allow: "Local Activation" | "Local Launch"
   </user>

  </acl>
```

Проверки контроля разрешения запуска Launch2

Использование

`<launch2_acl: ["name"]>`
 	
`<user: ["user_name"]>`
`acl_inheritance: ["value"]`
`acl_apply: ["value"]`
`(optional) acl_allow: ["rights value"]`
`(optional) acl_deny: ["rights value"]`
`</user>`
 	
`</acl>`

Список launch2 ACL (список контроля доступа для запуска 2) определяется ключевым словом **`launch2_acl`**. Имя списка ACL должно быть уникальным, чтобы его можно было использовать с элементом прав запуска DCOM. Список launch2 ACL (список контроля доступа для запуска 2) может содержать одну или несколько пользовательских записей.

Связанные типы	Допустимые типы
`acl_inheritance`	> not inherited (не наследуется) > inherited (наследуется)
`acl_apply`	> this object only (только этот объект)
`acl_allow` **`acl_deny`**	Эти параметры являются необязательными и используются для определения прав, которые пользователь имеет применительно к объекту. Общие права следующие: > launch (запуск)

Список launch2 ACL (список контроля доступа для запуска 2) используйте только с системами Windows 2000 и NT.

Пример проверки контроля доступа для операций launch (запуск) приведен ниже:

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 <launch2_acl: "2">

  <user: "Administrators">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "Launch"
  </user>

  <user: "INTERACTIVE">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "Launch"
  </user>

  <user: "SYSTEM">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "Launch"
  </user>

 </acl>
```

Проверки контроля разрешения доступа

Использование

`<access_acl: ["name"]>`
 	
`<user: ["user_name"]>`
`acl_inheritance: ["value"]`
`acl_apply: ["value"]`
`(optional) acl_allow: ["rights value"]`
`(optional) acl_deny: ["rights value"]`
`</user>`
 	
`</acl>`

Список ACL для доступа определяется ключевым словом `access_acl`. Имя списка ACL должно быть уникальным, чтобы его можно было использовать с элементом прав доступа DCOM. Список ACL для доступа может содержать одну или несколько пользовательских записей.

Связанные типы	Допустимые типы
acl_inheritance	> not inherited (не наследуется) > inherited (наследуется)
acl_apply	> this object only (только этот объект)
acl_allow **acl_deny**	Эти параметры являются необязательными и используются для определения прав, которые пользователь имеет применительно к объекту.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	Общие права следующие: > local access (локальный доступ) > remote access (удаленный доступ)

Пример проверки контроля доступа приведен ниже:

```
<access_acl: "3">

 <user: "SELF">
  acl_inheritance: "not inherited"
  acl_apply: "This object only"
  acl_allow: "Local Access"
 </user>

 <user: "SYSTEM">
  acl_inheritance: "not inherited"
  acl_apply: "This object only"
  acl_allow: "Local Access"
 </user>

 <user: "Users">
  acl_inheritance: "not inherited"
  acl_apply: "This object only"
  acl_allow: "Local Access"
 </user>

</acl>
```

Пользовательские элементы

Пользовательский элемент — это полная проверка, определенная на основе ключевых слов, определения которых приведены ниже. Ниже приведен список доступных типов пользовательских элементов. Каждая проверка начинается с тега **<custom_item>** и заканчивается тегом **</custom_item>**. В данные теги заключатся списки из одного или нескольких ключевых слов, которые интерпретируются анализатором проверки соответствия для выполнения проверки.

Закрывающие теги **</custom_item>** и **</item>** для пользовательских аудиторских проверок являются взаимозаменяемыми.

PASSWORD_POLICY

Использование
<custom_item> type: PASSWORD_POLICY description: ["description"]

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
 password_policy: [PASSWORD_POLICY_TYPE]
</custom_item>
```

Этот элемент политик проверяет значения, определенные в разделе «Windows Settings -> Security Settings -> Account Policies -> Password Policy» (Настройки Windows -> Настройки безопасности -> Политики учетных записей -> Политика паролей).

Данная проверка проводится путем вызова функции **NetUserModalsGet** с 1 уровнем.

Для описания того, какой элемент политики в отношении паролей должен быть проверен, эти элементы используют поле **password_policy**. Допустимы следующие типы:

- ENFORCE_PASSWORD_HISTORY (вести журнал паролей)
  ```
  value_type: POLICY_DWORD
  value_data: DWORD or RANGE [number of remembered passwords]
  ```

- MAXIMUM_PASSWORD_AGE (максимальный срок действия пароля)
  ```
  value_type: TIME_DAY
  value_data: DWORD or RANGE [time in days]
  ```

- MINIMUM_PASSWORD_AGE (минимальный срок действия пароля)
  ```
  value_type: TIME_DAY
  value_data: DWORD or RANGE [time in days]
  ```

- MINIMUM_PASSWORD_LENGTH (минимальная длина пароля)
  ```
  value_type: POLICY_DWORD
  value_data: DWORD or RANGE [minimum number of characters in the password]
  ```

- COMPLEXITY_REQUIREMENTS (пароль должен соответствовать требованиям в отношении сложности)
  ```
  value_type: POLICY_SET
  value_data: "Enabled" or "Disabled"
  ```

- REVERSIBLE_ENCRYPTION (для всех пользователей этого домена хранить пароль, используя обратимое шифрование)
  ```
  value_type: POLICY_SET
  value_data: "Enabled" or "Disabled"
  ```

- FORCE_LOGOFF (сетевая безопасность: принудительный вывод из сеанса по истечении допустимых часов работы)
  ```
  value_type: POLICY_SET
  value_data: "Enabled" or "Disabled"
  ```

В настоящее время нет возможности проверки политики «Для всех пользователей этого домена хранить пароль, используя обратимое шифрование».

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Политика FORCE_LOGOFF размещается в разделе «Security Settings -> Local Policies -> Security Options» (Настройки безопасности -> Локальные политики -> Параметры безопасности).

Далее приведен пример аудиторской проверки политики паролей:

```
<custom_item>
 type: PASSWORD_POLICY
 description: "Minimum password length"
 value_type: POLICY_DWORD
 value_data: 7
 password_policy: MINIMUM_PASSWORD_LENGTH
</custom_item>
```

LOCKOUT_POLICY

Использование

```
<custom_item>
 type: LOCKOUT_POLICY
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
 lockout_policy: [LOCKOUT_POLICY_TYPE]
</custom_item>
```

Этот элемент политик проверяет значения, определенные в разделе «Security Settings -> Account Policies -> Account Lockout Policy» (Настройки безопасности -> Политики учетных записей -> Политика блокировки учетных записей).

Данная проверка проводится путем вызова функции **`NetUserModalsGet`** с 3 уровнем.

Для описания того, какой элемент политики в отношении паролей должен быть проверен, этот элемент использует поле **`lockout_policy`**. Допустимы следующие типы:

- LOCKOUT_DURATION (продолжительность блокировки учетной записи)
  `value_type: TIME_MINUTE`
  `value_data: DWORD or RANGE [time in minutes]`

- LOCKOUT_THRESHOLD (порог блокировки учетной записи)
  `value_type: POLICY_DWORD`
  `value_data: DWORD or RANGE [time in days]`

- LOCKOUT_RESET (выполнить сброс счетчика блокировки учетной записи через)
  `value_type: TIME_MINUTE`
  `value_data: DWORD or RANGE [time in minutes]`

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Приведем пример:

```
 <custom_item>
  type: LOCKOUT_POLICY
  description: "Reset lockout account counter after"
  value_type: TIME_MINUTE
  value_data: 120
  lockout_policy: LOCKOUT_RESET
 </custom_item>
```

KERBEROS_POLICY

Использование

```
<custom_item>
 type: KERBEROS_POLICY
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
 kerberos_policy: [KERBEROS_POLICY_TYPE]
</custom_item>
```

Этот элемент политик проверяет значения, определенные в разделе «Security Settings -> Account Policies -> Kerberos Policy» (Настройки безопасности -> Политики учетных записей -> Политика Kerberos).

Данная проверка проводится путем вызова функции **`NetUserModalsGet`** с 1 уровнем.

Для описания того, какой элемент политики в отношении паролей должен быть проверен, этот элемент использует поле **`kerberos_policy`**. Допустимы следующие типы:

- USER_LOGON_RESTRICTIONS (принудительные ограничения входа пользователей)
  ```
  value_type: POLICY_SET
  value_data: "Enabled" or "Disabled"
  ```
- SERVICE_TICKET_LIFETIME (максимальный срок жизни билета службы)
  ```
  value_type: TIME_MINUTE
  value_data: DWORD or RANGE [time in minutes]
  ```
- USER_TICKET_LIFETIME (максимальный срок жизни билета пользователя)
  ```
  value_type: TIME_HOUR
  value_data: DWORD or RANGE [time in hours]
  ```
- USER_TICKET_RENEWAL_LIFETIME (максимальный срок жизни для возобновления билета пользователя)
  ```
  value_type: TIME_DAY
  value_data: DWORD or RANGE [time in day]
  ```
- CLOCK_SYNCHRONIZATION_TOLERANCE (максимальная погрешность синхронизации часов компьютера)

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
value_type: TIME_MINUTE
value_data: DWORD or RANGE [time in minute]
```

Политику Kerberos можно проверять, только сравнивая с данными центра KDC (центра распространения ключей), который в случае ОС Windows обычно является контроллером домена.

Пример:

```
<custom_item>
 type: KERBEROS_POLICY
 description: "Maximum lifetime for user renewal ticket"
 value_type: TIME_DAY
 value_data: 12
 kerberos_policy: USER_TICKET_RENEWAL_LIFETIME
</custom_item>
```

AUDIT_POLICY

Использование

```
<custom_item>
 type: AUDIT_POLICY
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
 audit_policy: [PASSWORD_POLICY_TYPE]
</custom_item>
```

Этот элемент политик проверяет значения, определенные в разделе «Security Settings -> Local Policies -> Audit Policy» (Настройки безопасности -> Локальные политики -> Политика аудита).

Данная проверка проводится путем вызова функции **`LsaQueryInformationPolicy`** с уровнем **`PolicyAuditEventsInformation`**.

Для описания того, какой элемент политики в отношении паролей должен быть проверен, этот элемент использует поле **`audit_policy`**. Допустимы следующие типы:

- AUDIT_ACCOUNT_LOGON (аудит событий входа в систему)
- AUDIT_ACCOUNT_MANAGER (аудит управления учетными записями)
- AUDIT_DIRECTORY_SERVICE_ACCESS (аудит доступа к службе каталогов)
- AUDIT_LOGON (аудит входа в систему)
- AUDIT_OBJECT_ACCESS (аудит доступа к объектам)
- AUDIT_POLICY_CHANGE (аудит изменений политик)
- AUDIT_PRIVILEGE_USE (аудит использования привилегий)
- AUDIT_DETAILED_TRACKING (аудит отслеживания процессов)

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

- AUDIT_SYSTEM (аудит системных событий)

```
value_type: AUDIT_SET
value_data: "No auditing", "Success", "Failure", "Success, Failure"
```

Обратите внимание, что в выражении «Success, Failure» необходим пробел.

Пример:

```
 <custom_item>
  type: AUDIT_POLICY
  description: "Audit policy change"
  value_type: AUDIT_SET
  value_data: "Failure"
  audit_policy: AUDIT_POLICY_CHANGE
 </custom_item>
```

AUDIT_POLICY_SUBCATEGORY

Использование

```
<custom_item>
 type: AUDIT_POLICY_SUBCATEGORY
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
 audit_policy_subcategory: [SUBCATEGORY_POLICY_TYPE]
</custom_item>
```

Этот элемент политики проверяет значения, перечисленные в разделе auditpol /get /category:*.

Данная проверка выполняется путем выполнения команды cmd.exe auditpol /get /category:* через WMI.

Для определения, какую подкатегорию нужно проверить, этот элемент использует поле **audit_policy_subcategory**. Допустимые типы SUBCATEGORY_POLICY_TYPE:

- Security State Change (изменение состояния безопасности)
- Security System Extension (расширение системы безопасности)
- System Integrity (целостность системы)
- IPsec Driver (драйвер IPsec)
- Other System Events (другие системные события)
- Logon (вход в систему)
- Logoff (выход из системы)
- Account Lockout (блокировка учетных записей)

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

- IPsec Main Mode (основной режим IPsec)
- IPsec Quick Mode (быстрый режим IPsec)
- IPsec Extended Mode (расширенный режим IPsec)
- Special Logon (специальный вход)
- Other Logon/Logoff Events (другие события входа/выхода)
- Network Policy Server (сервер сетевых политик)
- File System (файловая система)
- Registry (реестр)
- Kernel Object (объект ядра)
- SAM
- Certification Services (службы сертификации)
- Application Generated (созданные приложениями)
- Handle Manipulation (работа с дескрипторами)
- File Share (общий доступ к файлам)
- Filtering Platform Packet Drop (отбрасывание пакетов платформой фильтрации)
- Filtering Platform Connection (подключение платформы фильтрации)
- Other Object Access Events (другие события доступа к объектам)
- Sensitive Privilege Use (использование прав, затрагивающее конфиденциальные данные)
- Non Sensitive Privilege Use (использование прав, не затрагивающее конфиденциальные данные)
- Other Privilege Use Events (другие события использования прав)
- Process Creation (создание процессов)
- Process Termination (завершение процессов)
- DPAPI Activity (активность DPAPI)
- RPC Events (события RPC)
- Audit Policy Change (изменение политики аудита)
- Authentication Policy Change (изменение политики проверки подлинности)
- Authorization Policy Change (изменение политики авторизации)
- MPSSVC Rule-Level Policy Change (изменение политики на уровне правил MPSSVC)
- Filtering Platform Policy Change (изменение политики платформы фильтрации)
- Other Policy Change Events (другие события изменения политики)
- User Account Management (управление учетными записями пользователей)
- Computer Account Management (управление учетными записями компьютера)
- Security Group Management (управление группами безопасности)
- Distribution Group Management (управление группами распространения)
- Application Group Management (управление группами приложений)
- Other Account Management Events (другие события управления учетными записями)
- Directory Service Access (доступ к службе каталогов)
- Directory Service Changes (изменения службы каталогов)
- Directory Service Replication (репликация службы каталогов)
- Directory Service Replication (подробная репликация службы каталогов)

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

- Credential Validation (проверка учетных данных)
- Kerberos Service Ticket Operations (операции с билетами службы Kerberos)
- Other Account Events (другие события с учетными записями)

```
value_type: AUDIT_SET
value_data: "No auditing", "Success", "Failure", "Success, Failure"
```

Обратите внимание, что в выражении «Success, Failure» необходим пробел.

Эта проверка применима только к операционным системам Windows Vista/2008 Server и более поздним. Если брандмауэр включен, то кроме добавления в его параметры в качестве исключения WMI, в параметрах брандмауэра необходимо также включить с помощью **gpedit.msc** параметр «Брандмауэр Windows: Разрешить исключение для входящих сообщений удаленного администрирования». Эта проверка может не работать на других системах Vista/2008, кроме английской, а также на системах, на которых не установлено средство auditpol.

Пример:

```
 <custom_item>
  type: AUDIT_POLICY_SUBCATEGORY
  description: "AUDIT Security State Change"
  value_type: AUDIT_SET
  value_data:  "success, failure"
  audit_policy_subcategory: "Security State Change"
 </custom_item>
```

CHECK_ACCOUNT

Использование

```
<custom_item>
 type: CHECK_ACCOUNT
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 account_type: [ACCOUNT_TYPE]
 (optional) check_type: [CHECK_TYPE]
</custom_item>
```

Этот элемент политик проверяет следующие значения, определенные в разделе «Security Settings -> Local Policies -> Security Options» (Настройки безопасности -> Локальные политики -> Параметры безопасности).

- Учетные записи: состояние учетной записи «Администратор»
- Учетные записи: состояние учетной записи «Гость»
- Учетные записи: переименование учетной записи администратора
- Учетные записи: переименование учетной записи гостя

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Данная проверка выполняется путем вызова функции **LsaQueryInformationPolicy** с уровнем **PolicyAccountDomainInformation**, чтобы получить идентификатор безопасности (SID) домена/системы, **LsaLookupSid** для получения имен администраторов и гостей, а **NetUserGetInfo** для получения информации об учетной записи.

Для определения того, какую учетную запись нужно проверить, этот элемент использует поле **account_type**. Допустимы следующие типы:

- ADMINISTRATOR_ACCOUNT (учетные записи: состояние учетной записи «Администратор»)
  ```
  value_type: POLICY_SET
  value_data: "Enabled" or "Disabled"
  ```
- GUEST_ACCOUNT (учетные записи: состояние учетной записи «Гость»)
  ```
  value_type: POLICY_SET
  value_data: "Enabled" or "Disabled"
  ```
- ADMINISTRATOR_ACCOUNT (учетные записи: переименование учетной записи администратора)
  ```
  value_type: POLICY_TEXT
  value_data: "TEXT HERE" [administrator name]
  check_type: [CHECK_TYPE] (any one of the possible check_type values)
  ```
- GUEST_ACCOUNT (учетные записи: переименование учетной записи гостя)
  ```
  value_type: POLICY_TEXT
  value_data: "TEXT HERE" [guest name]
  check_type: [CHECK_TYPE] (any one of the possible check_type values)
  ```

В зависимости от учетных данных «Домен» могут проверяться учетные записи локальной системы или домена.

Примеры:

```
<custom_item>
 type: CHECK_ACCOUNT
 description: "Accounts: Guest account status"
 value_type: POLICY_SET
 value_data: "Disabled"
 account_type: GUEST_ACCOUNT
</custom_item>

<custom_item>
 type: CHECK_ACCOUNT
 description: "Accounts: Rename administrator account"
 value_type: POLICY_TEXT
 value_data: "Dom_adm"
 account_type: ADMINISTRATOR_ACCOUNT
</custom_item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 <custom_item>
  type: CHECK_ACCOUNT
  description: "Accounts: Rename administrator account"
  value_type: POLICY_TEXT
  value_data: "Administrator"
  account_type: ADMINISTRATOR_ACCOUNT
  check_type: CHECK_NOT_EQUAL
 </custom_item>
```

CHECK_LOCAL_GROUP

Использование

```
<custom_item>
 type: CHECK_LOCAL_GROUP
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 group_type: [GROUP_TYPE]
 (optional) check_type : [CHECK_TYPE]
</custom_item>
```

Этот элемент политики проверяет имена и статус групп, перечисленных в **`lusmgr.msc`**.

Для определения того, какую учетную запись нужно проверить, этот элемент использует поле **`group_type`**. Допустимы следующие типы:

- ADMINISTRATORS_GROUP
- USERS_GROUP
- GUESTS_GROUP
- POWER_USERS_GROUP
- ACCOUNT_OPERATORS_GROUP
- SERVER_OPERATORS_GROUP
- PRINT_OPERATORS_GROUP
- BACKUP_OPERATORS_GROUP
- REPLICATORS_GROUP

Допустимыми для поля **`value_type`** типами являются:

- POLICY_SET (статус группы проверен)
  `value_type: POLICY_SET`
  `value_data: "Enabled" or "Disabled"`

- POLICY_TEXT (имя группы проверено)
  `value_type: POLICY_TEXT`
  `value_data: "Guests1"` (в этом случае **`value_data`** может быть любой текстовой строкой)

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Примеры:

```
<custom_item>
 type: CHECK_LOCAL_GROUP
 description: "Local Guest group must be enabled"
 value_type: POLICY_SET
 value_data: "enabled"
 group_type: GUESTS_GROUP
 check_type: CHECK_EQUAL
</custom_item>
```

```
<custom_item>
 type: CHECK_LOCAL_GROUP
 description: "Guests group account name should be Guests"
 value_type: POLICY_TEXT
 value_data: "Guests"
 group_type: GUESTS_GROUP
 check_type: CHECK_EQUAL
</custom_item>
```

```
<custom_item>
 type: CHECK_LOCAL_GROUP
 description: "Guests group account name should not be Guests"
 value_type: POLICY_TEXT
 value_data: "Guests"
 group_type: GUESTS_GROUP
 check_type: CHECK_NOT_EQUAL
</custom_item>
```

ANONYMOUS_SID_SETTING

Использование

```
<custom_item>
 type: ANONYMOUS_SID_SETTING
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
</custom_item>
```

Этот элемент политик проверяет следующие значения, определенные в разделе «Security Settings -> Local Policies -> Security Options -> Network access: Allow anonymous SID/Name translation» (Настройки безопасности -> Локальные политики -> Параметры безопасности -> Доступ к сети: разрешить трансляцию анонимного SID в имя). Данная проверка проводится путем вызова функции **`LsaQuerySecurityObject`** для дескриптора политики LSA.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Допустимы следующие типы:

```
value_type: POLICY_SET
value_data: "Enabled" or "Disabled"
```

При использовании этого аудита обратите внимание, что эта политика:

- является проверкой разрешений для службы LSA;
- проверяет, установлен ли для пользователя ANONYMOUS_USER флаг POLICY_LOOKUP_NAMES;
- не рекомендуется в ОС Windows 2003, потому что анонимный пользователь не может получить доступ к каналу LSA.

Пример:

```
 <custom_item>
  type: ANONYMOUS_SID_SETTING
  description: "Network access: Allow anonymous SID/Name translation"
  value_type: POLICY_SET
  value_data: "Disabled"
 </custom_item>
```

SERVICE_POLICY

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: SERVICE POLICY
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
 service_name: ["service name"]
</custom_item>
```

Этот элемент политики проверяет используемые при запуске значения, определенные в разделе «System Services» (системные службы). Данная проверка проводится путем вызова функции **`RegQueryValueEx`** для следующих разделов:

- раздел: "SYSTEM\CurrentControlSet\Services\" + имя_службы
- элемент: "Start"

Допустимы следующие типы:

```
value_type: SERVICE_SET
value_data: "Automatic", "Manual" or "Disabled"
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
svc_option: CAN_BE_NULL or CAN_NOT_BE_NULL
```

Поле **service_name** соответствует имени службы REAL. Это имя можно получить путем:

1) запуска панели управления «Службы» (в компоненте «Администрирование»);
2) выбора желаемой службы;
3) открытия диалогового окна «Свойства» (нажатие правой кнопки мыши -> пункт «Свойства»);
4) извлечения части «Имя службы».

Настройку разрешения службы можно проверить с помощью элемента SERVICE_PERMISSIONS.

Пример:

```
 <custom_item>
  type: SERVICE_POLICY
  description: "Background Intelligent Transfer Service"
  value_type: SERVICE_SET
  value_data: "Disabled"
  service_name: "BITS"
 </custom_item>
```

FILE_CHECK

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: FILE_CHECK
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
 file_option: [OPTION_TYPE]
</custom_item>
```

Этот элемент политики проверяет, существует файл (**value_data**) или нет (**file_option**). Данная проверка выполняется путем вызова функции **CreateFile**.

Допустимы следующие типы:

```
value_type: POLICY_TEXT
value_data: "file name"
file_option: MUST_EXIST or MUST_NOT_EXIST
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Примеры:

```
 <custom_item>
  type: FILE_CHECK
  description: "Check that win.ini exists in the system root"
  value_type: POLICY_TEXT
  value_data: "%SystemRoot%\win.ini"
  file_option: MUST_EXIST
 </custom_item>
```

```
 <custom_item>
  type: FILE_CHECK
  description: "Check that bad.exe does not exist in the system root"
  value_type: POLICY_TEXT
  value_data: "%SystemRoot%\bad.exe"
  file_option: MUST_NOT_EXIST
 </custom_item>
```

FILE_VERSION

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: FILE_VERSION
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 (optional) check_type: [value]
 file: PATH_TO_FILE
 file_option: [OPTION_TYPE]
 check_type: CHECK_TYPE
</custom_item>
```

По умолчанию этот элемент политики проверяет, является ли версия файла, указанного в поле **`file`**, более поздней или такой же, как версия удаленного файла. Эта проверка также может использоваться для определения того, является ли версия удаленного файла более ранней, с помощью опции **`check_type`**.

Допустимы следующие типы:

```
value_type: POLICY_TEXT
value_data: "file version"
file_option: MUST_EXIST or MUST_NOT_EXIST
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Примеры:

```
 <custom_item>
  type: FILE_VERSION
  description: "Audit for C:\WINDOWS\SYSTEM32\calc.exe"
  value_type: POLICY_TEXT
  value_data: "1.1.1.1"
  file: "C:\WINDOWS\SYSTEM32\calc.exe"
 </custom_item>
```

```
 <custom_item>
  type: FILE_VERSION
  description: "Audit for C:\WINDOWS\SYSTEM32\calc.exe"
  value_type: POLICY_TEXT
  value_data: "1.1.1.1"
  file: "C:\WINDOWS\SYSTEM32\calc.exe"
  check_type: CHECK_LESS_THAN
 </custom_item>
```

REG_CHECK

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: REG_CHECK
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 reg_option: [OPTION_TYPE]
 (optional) check_type: [value]
 (optional) key_item: [item value]
</custom_item>
```

Этот элемент политики проверяет, существует ли раздел (или элемент) реестра. Данная проверка выполняется путем вызова функций `RegOpenKeyEx` и `RegQueryValueEx`.

Допустимы следующие типы:

```
value_type: POLICY_TEXT
value_data: "key path"
reg_option: MUST_EXIST or MUST_NOT_EXIST
key_item: "item name"
```

Если поле **`key_item`** не определено, то этот элемент проверяет существование пути к разделу. Иначе проверяется существование элемента.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Пример:

```
 <custom_item>
  type: REG_CHECK
  description: "Check the key HKLM\SOFTWARE\Adobe\Acrobat
        Reader\7.0\AdobeViewer"
  value_type: POLICY_TEXT
  value_data: "HKLM\SOFTWARE\Adobe\Acrobat Reader\7.0\AdobeViewer"
  reg_option: MUST_NOT_EXIST
  key_item: "EULA"
 </custom_item>
```

REGISTRY_SETTING

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: REGISTRY_SETTING
 description: ["description"]
 value_type: [VALUE_TYPE]
 value_data: [value]
 reg_key: ["key name"]
 reg_item: ["key item"]
 (optional) check_type: [value]
 (optional) reg_option: [KEY_OPTIONS]
 (optional) reg_enum: ENUM_SUBKEYS
</custom_item>
```

Этот элемент политики используется для проверки значения раздела реестра. Этот элемент политики используется многими проверками политики в разделе «Security Settings -> Local Policies -> Security Options» (Настройки безопасности -> Локальные политики -> Параметры безопасности). Данная проверка выполняется путем вызова функции **RegQueryValueEx**.

Поле **reg_key** содержит имя раздела реестра (например: «HKLM\SOFTWARE\Microsoft\Driver Signing»). Первая часть раздела (HKLM) используется для подключения к правильному кусту реестра. Последующий путь является статическим путем к месту расположения нужного элемента **reg_item**.

Куст реестра HKU (HKEY_USERS) представляет собой особый случай. Указать идентификатор SID для разделов куста HKU невозможно. В этом случае функция **nbin** выполняет внутренний перебор всех идентификаторов SID, и проверка проходит только в том случае, если значения всех идентификаторов SID являются правильными.

Например:

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
<custom_item>
type: REGISTRY_SETTING
description: "HKU\Control Panel\Desktop\ScreenSaveActive"
value_type: POLICY_DWORD
value_data: 1
reg_key: "HKU\Control Panel\Desktop"
reg_item: "ScreenSaveActive"
</item>
```

выполнит цикл по:

```
HKU\S-1-5-18\Control Panel\Desktop\ScreenSaveActive
HKU\S-1-5-19\Control Panel\Desktop\ScreenSaveActive
HKU\S-1-5-20\Control Panel\Desktop\ScreenSaveActive
...
```

и проверка будет успешной, если элемент ScreenSaveActive имеет значение 1 для всех идентификаторов SID.

Необязательному полю **reg_option** может быть присвоено значение CAN_BE_NULL, чтобы проверка успешно завершалась в том случае, если раздела не существует, а также может быть присвоено противоположное значение CAN_NOT_BE_NULL.

Дополнительный параметр **reg_enum** с аргументом ENUM_SUBKEYS можно использовать для перечисления указанных значений для всех подразделов раздела реестра. Например, в разделе **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall** перечислено много пакетов программного обеспечения. Если требуется сравнить значение поля CurrentVersion для всех подразделов раздела Uninstall, воспользуйтесь параметром **reg_enum**.

Пример:

```
<custom_item>
 type: REGISTRY_SETTING
 description: "DBMS network port, protocol, and services (PPS) usage"
 info: "Checking whether TCPDynamicPorts key value is configured (should
       be blank)."
 value_type: POLICY_TEXT
 value_data: ""
 reg_key: "HKLM\SOFTWARE\Microsoft\Microsoft SQL
       Server\MSSQL.1\MSSQLServer\SuperSocketNetLib\Tcp"
 reg_item: "TCPDynamicPorts"
 reg_enum: ENUM_SUBKEYS
 reg_option: CAN_BE_NULL
</custom_item>
```

Этот аудит куста реестра HKU не включает идентификатор безопасности SID в путь реестра **reg_key**. Данный пример выполнит поиск по всем идентификаторам SID куста HKU указанного значения **reg_item**.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Пример:

```
<custom_item>
 type: REGISTRY_SETTING
 description: "FakeAlert.BG trojan check"
 value_type: POLICY_TEXT
 reg_key: "HKU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
 reg_item: "brastk"
 value_data: "C:\WINDOWS\System32\brastk.exe"
 reg_option: CAN_BE_NULL
 check_type: CHECK_NOT_EQUAL
 info: "A registry entry for FakeAlert.BG trojan/downloader was found."
 info: "The contents of this audit can be edited as desired."
</custom_item>
```

Возможны следующие основные типы поля `value_type`:

- POLICY_SET
  `value_data: "Enabled" or "Disabled"`
- POLICY_DWORD
  `value_data: DWORD or RANGE [same dword as in registry or range]`
- POLICY_TEXT
  `value_data: "TEXT" [same text as in registry]`
- POLICY_MULTI_TEXT
  `value_data: "TEXT1" && "TEXT2" && ... && "TEXTN" [same texts as in registry]`
- POLICY_BINARY
  `value_data: "0102ac0b...34fb" [same binary as in registry]`
- FILE_ACL, REG_ACL, SERVICE_ACL, LAUNCH_ACL, ACCESS_ACL
  `value_data: "acl_name" [name of the acl to use]`

Следующие дополнительные типы для поля **`value_type`** существуют и используются в предварительно определенных элементах:

- DRIVER_SET
  `value_data: "Silent Succeed", "Warn but allow installation", "Do not allow installation"`
- LDAP_SET
  `value_data: "None" or "Require Signing"`
- LOCKEDID_SET
  `value_data: "user display name, domain and user names", "user display name only", "do not display user information"`
- SMARTCARD_SET

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
value_data: "No action", "Lock workstation", "Force logoff", "Disconnect
if a remote terminal services session"
```

- LOCALACCOUNT_SET
  ```
  value_data: "Classic - local users authenticate as themselves", "Guest
  only - local users authenticate as guest"
  ```

- NTLMSSP_SET
  ```
  value_data: "No minimum", "Require message integrity", "Require message
  confidentiality", "Require ntlmv2 session security", "Require 128-bit
  encryption"
  ```

- CRYPTO_SET
  ```
  value_data: "User input is not required when new keys are stored and
  used", "User is prompted when the key is first used" or "User must enter a
  password each time they use a key"
  ```

- OBJECT_SET
  ```
  value_data: "Administrators group", "Object creator"
  ```

- DASD_SET
  ```
  value_data: "Administrators", "administrators and power users",
  "Administrators and interactive users"
  ```

- LANMAN_SET
  ```
  value_data: "Send LM & NTLM responses", "send lm & ntlm - use ntlmv2
  session security if negotiated", "send ntlm response only", "send ntlmv2
  response only", "send ntlmv2 response only\refuse lm" or "send ntlmv2
  response only\refuse lm & ntlm"
  ```

- LDAPCLIENT_SET
  ```
  value_data: "None", "Negotiate Signing" or "Require Signing"
  ```

- EVENT_METHOD
  ```
  value_data: "by days", "manually" or "as needed"
  ```

- POLICY_DAY
  ```
  value_data: DWORD or RANGE (time in days)
  ```

- POLICY_KBYTE
  ```
  value_data: DWORD or RANGE
  ```

Для поля **custom_item** используйте основные типы **value_type**. Дополнительные типы были созданы для предварительно определенных элементов.

Если тип **value_type** — ACL, то элемент реестра должен быть описанием безопасности в двоичном формате.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Примеры:

```
<custom_item>
 type: REGISTRY_SETTING
 description: "Network security: Do not store LAN Manager hash value on
       next password change"
 value_type: POLICY_SET
 value_data: "Enabled"
 reg_key: "HKLM\SYSTEM\CurrentControlSet\Control\Lsa"
 reg_item: "NoLMHash"
</custom_item>
```

```
<custom_item>
 type: REGISTRY_SETTING
 description: "Network access: Shares that can be accessed anonymously"
 value_type: POLICY_MULTI_TEXT
 value_data: "SHARE" && "EXAMPLE$"
 reg_key: "HKLM\SYSTEM\CurrentControlSet\Services\LanManServer\Parameters"
 reg_item: "NullSessionShares"
</custom_item>
```

```
<custom_item>
 type: REGISTRY_SETTING
 description: "DCOM: Network Provisioning Service - Launch permissions"
 value_type: LAUNCH_ACL
 value_data: "2"
 reg_key: "HKLM\SOFTWARE\Classes\AppID\{39ce474e-59c1-4b84-9be2-
       2600c335b5c6}"
 reg_item: "LaunchPermission"
</custom_item>
```

```
<custom_item>
 type: REGISTRY_SETTING
 description: "DCOM: Automatic Updates - Access permissions"
 value_type: ACCESS_ACL
 value_data: "3"
 reg_key: "HKLM\SOFTWARE\Classes\AppID\{653C5148-4DCE-4905-9CFD-
       1B23662D3D9E}"
 reg_item: "AccessPermission"
</custom_item>
```

GROUP_MEMBERS_POLICY

Использование

```
<custom_item>
 type: GROUP_MEMBERS_POLICY
 description: ["description"]
 value_type: [value type]
 value_data: [value]
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 (optional) check_type: [value]
 group_name: ["group name"]
</custom_item>
```

Этот элемент политики проверяет существование определенного списка пользователей в одной или нескольких группах.

Допустим следующий тип:

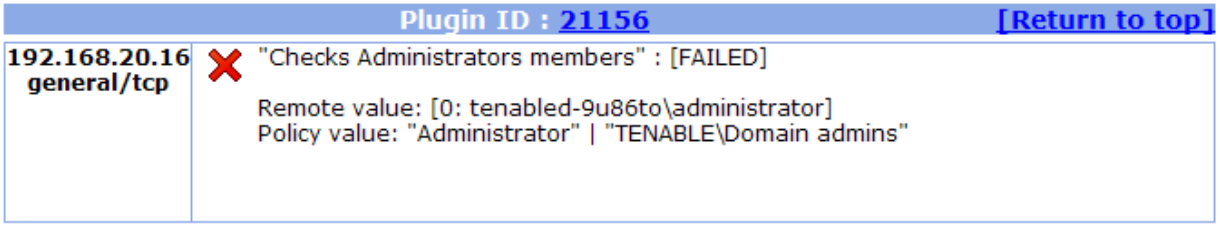
```
value_type: POLICY_TEXT or POLICY_MULTI_TEXT
value_data: "user1" && "user2" && ... && "usern"
```

При использовании этого аудита обратите внимание, что имя пользователя может быть указано с именем домена, например «МОЙ_ДОМЕН\Джон Смит», а поле **group_name** определяет единственную группу для проверки.

В одном файле Nessus **.audit** могут быть определены элементы для нескольких клиентов, поэтому проверять списки пользователей в различных группах очень легко. Приведем пример политики **.audit**, проверяющей, содержит ли группа Administrators только пользователей «Administrator» и «TENABLE\Domain admins» (Tenable\администраторы домена):

```
 <custom_item>
  type: GROUP_MEMBERS_POLICY
  description: "Checks Administrators members"
  value_type: POLICY_MULTI_TEXT
  value_data: "Administrator" && "TENABLE\Domain admins"
  group_name: "Administrators"
 </custom_item>
```

Приведем образец снимка экрана в результате выполнения приведенного выше содержимого файла **.audit** для сервера под управлением ОС Windows 2003:



USER_GROUPS_POLICY

Использование

```
<custom_item>
 type: USER_GROUPS_POLICY
 description: ["description"]
 value_type: [value type]
 value_data: [value]
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 (optional) check_type: [value]
 user_name: ["user name"]
</custom_item>
```

Этот элемент политики проверяет, принадлежит ли пользователь ОС Windows к указанным в **value_data** группам. С помощью этого аудита можно проверять только пользователей домена по контроллеру домена. Эта проверка не применима к таким встроенным пользователям, как «Local Service» (локальная служба).

Пример:

```
 <custom_item>
  type: USER_GROUPS_POLICY
  description: "3.72 DG0005: DBMS administration OS accounts"
  info: "Checking that the 'dba' account is a member of required groups
        only."
  info: "Modify the account/groups in this audit to match your
        environment."
  value_type: POLICY_MULTI_TEXT
  value_data: "Users" && "SQL Server DBA" && "SQL Server Users"
  user_name: "dba"
 </custom_item>
```

USER_RIGHTS_POLICY

Использование

```
<custom_item>
 type: USER_RIGHTS_POLICY
 description: ["description"]
 value_type: [value type]
 value_data: [value]
 (optional) check_type: [value]
 right_type: [right]
</custom_item>
```

Этот элемент политик проверяет следующее значение, определенное в разделе «Security Settings -> Local Policies -> User Rights Assignment» (Настройки безопасности -> Локальные политики -> Предоставление прав пользователям). Данная проверка проводится путем вызова функции **LsaEnumerateAccountsWithUserRight** для дескриптора политики LSA.

Поле **right_type** соответствует испытываемому праву. Допустимы следующие значения:

```
right_type: RIGHT
```

где **RIGHT** может быть:

```
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
SeBackupPrivilege
SeBatchLogonRight
SeChangeNotifyPrivilege
SeCreateGlobalPrivilege
SeCreatePagefilePrivilege
SeCreatePermanentPrivilege
SeCreateTokenPrivilege
SeDenyBatchLogonRight
SeDenyInteractiveLogonRight
SeDenyNetworkLogonRight
SeDenyRemoteInteractiveLogonRight
SeDenyServiceLogonRight
SeDebugPrivilege
SeEnableDelegationPrivilege
SeImpersonatePrivilege
SeIncreaseBasePriorityPrivilege
SeIncreaseWorkingSetPrivilege
SeIncreaseQuotaPrivilege
SeInteractiveLogonRight
SeLoadDriverPrivilege
SeLockMemoryPrivilege
SeMachineAccountPrivilege
SeManageVolumePrivilege
SeNetworkLogonRight
SeProfileSingleProcessPrivilege
SeRemoteShutdownPrivilege
SeRemoteInteractiveLogonRight
SeReLabelPrivilege
SeRestorePrivilege
SeSecurityPrivilege
SeServiceLogonRight
SeShutdownPrivilege
SeSyncAgentPrivilege
SeSystemEnvironmentPrivilege
SeSystemProfilePrivilege
SeSystemtimePrivilege
SeTakeOwnershipPrivilege
SeTcbPrivilege
SeTimeZonePrivilege
SeUndockPrivilege
SeUnsolicitedInputPrivilege
```

Допустим следующий тип:

```
value_type: USER_RIGHT
value_data: "user1" && "user2" && "group1" && ... && "groupn"
```

Проверки прав пользователя выполняют многочисленные запросы к контроллеру домена. Эти проверки должны быть включены в отдельный файл политики и запускаться только в отношении контроллера домена и ОДНОЙ системы домена.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Тип **right** не должен быть заключен в кавычки, так как он анализируется в качестве лексемы.

Пример:

```
 <custom_item>
  type: USER_RIGHTS_POLICY
  description: "Create a token object"
  value_type: USER_RIGHT
  value_data: "Administrators" && "Backup Operators"
  right_type: SeCreateTokenPrivilege
 </custom_item>
```

FILE_PERMISSIONS

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: FILE_PERMISSIONS
 description: ["description"]
 value_type: [value_type]
 value data: [value]
 (optional) check_type: [value]
 file: ["filename"]
 (optional) acl_option: [acl_option]
</custom_item>
```

Этот элемент политики проверяет правильность FILE_PERMISSIONS списка ACL. Данная проверка проводится путем вызова функции **GetSecurityInfo** с 7 уровнем для дескриптора файла.

Допустим следующий тип:

```
value_type: FILE_ACL
value_data: "ACLname"
file: "PATH\Filename"
```

В качестве имени файла или папки можно использовать следующие предварительно определенные пути:

```
%allusersprofile%
%windir%
%systemroot%
%commonfiles%
%programfiles%
%systemdrive%
%systemdirectory%
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

При использовании этого аудита обратите внимание на следующее:

- Поле **file** должно содержать полный путь к файлу или имени папки (например, **C:\WINDOWS\SYSTEM32**) или в нем должны использоваться приведенные выше ключевые слова для определения путей. В случае использования ключевых слов должен быть включен удаленный реестр, чтобы сканер Nessus мог определить значения переменных путей.
- Поле **value_data** содержит имя списка ACL, определенного в файле политики.
- Поле **acl_option** может иметь значение CAN_BE_NULL или CAN_NOT_BE_NULL, чтобы установить успешное прохождение проверки или выдачу ошибки в случае отсутствия файла.

Примеры:

```
<file_acl: "ACL1">

 <user: "Administrators">
  acl_inheritance: "not inherited"
  acl_apply: "This object only"
  acl_allow: "Full Control"
 </user>

 <user: "System">
  acl_inheritance: "not inherited"
  acl_apply: "This object only"
  acl_allow: "Full Control"
 </user>

</acl>

<custom_item>
 type: FILE_PERMISSIONS
 description: "Permissions for C:\WINDOWS\SYSTEM32"
 value_type: FILE_ACL
 value_data: "ACL1"
 file: "C:\WINDOWS\SYSTEM32"
</custom_item>
```

```
<custom_item>
 type: FILE_PERMISSIONS
 description: "Permissions for C:\WINDOWS\SYSTEM32"
 value_type: FILE_ACL
 value_data: "ACL1"
 file: "%SystemRoot%\SYSTEM32"
</custom_item>
```

При выполнении приведенной выше проверки модуль проверки соответствия будет проверять, соответствуют ли определенные для **%SystemRoot%\SYSTEM32** разрешения описанным в списке file_acl ACL1.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

FILE_AUDIT

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: FILE_AUDIT
 description: ["description"]
 value_type: [value_type]
 value_data: [value]
 (optional) check_type: [value]
 file: ["filename"]
 (optional) acl_option: [acl_option]
</custom_item>
```

Этот элемент политики используется для проверки свойств аудита («Свойства –> Безопасность –> Дополнительно –> Аудит») файла или папки, используя указанный список ACL. Данная проверка проводится путем вызова функции **GetSecurityInfo** с уровнем SACL_SECURITY_INFORMATION (информация о безопасности списка SACL) для дескриптора файла.

Допустим следующий тип:

```
value_type: FILE_ACL
value_data: "ACLname"
file: "PATH\Filename"
```

В качестве имени файла или папки можно использовать следующие предварительно определенные пути:

```
%allusersprofile%
%windir%
%systemroot%
%commonfiles%
%programfiles%
%systemdrive%
%systemdirectory%
```

При использовании этого аудита обратите внимание на следующее:

- Поле **file** должно содержать полный путь к файлу или имени папки (например, **C:\WINDOWS\SYSTEM32**) или в нем должны использоваться приведенные выше ключевые слова для определения путей. В случае использования ключевых слов должен быть включен удаленный реестр, чтобы сканер Nessus мог определить значения переменных путей.
- Поле **value_data** содержит имя списка ACL, определенного в файле политики.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

- Поле **acl_option** может иметь значение CAN_BE_NULL или CAN_NOT_BE_NULL, чтобы установить успешное прохождение проверки или выдачу ошибки в случае отсутствия файла.
- Поля **acl_allow** и **acl_deny** соответствуют событиям аудита Successful (успешно пройден) и Failed (не пройден).

Приведем пример файла **.audit**, в котором реализована функция FILE_AUDIT, включая пример правила списка контроля доступа с именем ACL1.

```
 <check_type: "Windows" version:"2">
 <group_policy: "Audits SYSTEM32 directory for correct auditing
       permissions">

 <file_acl: "ACL1">
  <user: "Everyone">
   acl_inheritance: "not inherited"
   acl_apply: "This folder, subfolders and files"
   acl_deny: "full control"
   acl_allow: "full control"
  </user>
 </acl>

 <custom_item>
  type: FILE_AUDIT
  description: "Audit for C:\WINDOWS\SYSTEM32"
  value_type: FILE_ACL
  value_data: "ACL1"
  file: "%SystemRoot%\SYSTEM32"
 </custom_item>

 </group_policy>
 </check_type>
```

FILE_CONTENT_CHECK

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: FILE_CONTENT_CHECK
 description: ["description"]
 value_type: [value_type]
 value_data: ["filename"]
 (optional) check_type: [value]
 regex: ["regex"]
 expect: ["regex"]
(optional) file_option: [file_option]
</custom_item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Этот элемент политики проверяет, содержит ли файл регулярное выражение **regex** и соответствует ли оно значению **expect**.

Данная проверка проводится путем вызова функции **ReadFile** для дескриптора файла.

Допустим следующий тип:

```
value_type: POLICY_TEXT
value_data: "PATH\Filename"
regex: "regex"
expect: "regex"
```

В качестве имени файла или папки можно использовать следующие предварительно определенные пути:

```
%allusersprofile%
%windir%
%systemroot%
%commonfiles%
%programfiles%
%systemdrive%
```

При использовании этого типа аудита обратите внимание на следующее:

- Поле **value_data** должно содержать полный путь к файлу или имени папки (например, **C:\WINDOWS\SYSTEM32**), или в нем должны использоваться приведенные выше ключевые слова для определения путей. В случае использования ключевых слов должен быть включен удаленный реестр, чтобы сканер Nessus мог определить значения переменных путей.
- Поле **regex** проверяет наличие элемента в файле.
- Поле **expect** проверяет соответствие этого элемента регулярному выражению.
- Поле **file_option** может иметь значение CAN_BE_NULL, чтобы установить успешное прохождение проверки в случае отсутствия файла.
- Поле **file_option** может иметь значение CAN_NOT_BE_NULL, чтобы установить ошибку, если файл существует, но пустой.

Пример:

```
<custom_item>
 type: FILE_CONTENT_CHECK
 description: "File content for C:\WINDOWS\win.ini"
 value_type: POLICY_TEXT
 value_data: "C:\WINDOWS\win.ini"
 regex: "aif=.*"
 expect: "aif=MPEGVideo"
</custom_item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

FILE_CONTENT_CHECK_NOT

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: FILE_CONTENT_CHECK_NOT
 description: ["description"]
 value_type: [value_type]
 value_data: ["filename"]
 (optional) check_type: [value]
 regex: ["regex"]
 expect: ["regex"]
(optional) file_option: [file_option]
</custom_item>
```

Этот элемент политики проверяет, содержит ли файл регулярное выражение regex, а также, что это выражение не соответствует значению поля expect. Данная проверка проводится путем вызова функции **`ReadFile`** для дескриптора файла.

Допустим следующий тип:

```
value_type: POLICY_TEXT
value_data: "PATH\Filename"
regex: "regex"
expect: "regex"
```

В качестве имени файла или папки можно использовать следующие предварительно определенные пути:

```
%allusersprofile%
%windir%
%systemroot%
%commonfiles%
%programfiles%
%systemdrive%
```

При использовании этого типа аудита обратите внимание на следующее:

- Поле **`value_data`** должно содержать полный путь к файлу или имени папки (например, **`C:\WINDOWS\SYSTEM32`**), или в нем должны использоваться приведенные выше ключевые слова для определения путей. В случае использования ключевых слов должен быть включен удаленный реестр, чтобы сканер Nessus мог определить значения переменных путей.
- Поле **`regex`** проверяет наличие элемента в файле.
- Поле **`expect`** проверяет соответствие этого элемента регулярному выражению.
- Поле **`file_option`** может иметь значение CAN_BE_NULL, чтобы установить успешное прохождение проверки в случае отсутствия файла.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

- Поле **file_option** может иметь значение CAN_NOT_BE_NULL, чтобы установить ошибку, если файл существует, но пустой.

Пример:

```
<custom_item>
 type: FILE_CONTENT_CHECK_NOT
 description: "File content for C:\WINDOWS\win.ini"
 value_type: POLICY_TEXT
 value_data: "C:\WINDOWS\win.ini"
 (optional) check_type: [value]
 regex: "au=.*"
 expect: "au=MPEGVideo2"
 file_option: CAN_NOT_BE_NULL
</custom_item>
```

REGISTRY_PERMISSIONS

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: REGISTRY_PERMISSIONS
 description: ["description"]
 value_type: [value_type]
 value_data: [value]
 (optional) check_type: [value]
 reg_key: ["regkeyname"]
 (optional) acl_option: [acl_option]
</custom_item>
```

Этот элемент политики проверяет правильность списка ACL для раздела реестра. Данная проверка проводится путем вызова функции **RegGetKeySecurity** для дескриптора раздела реестра.

Допустим следующий тип:

```
value_type: REG_ACL
value_data: "ACLname"
reg_key: "RegistryKeyName"
```

В качестве значения поля **reg_key** можно использовать следующие предварительно определенные пути:

```
HKLM (HKEY_LOCAL_MACHINE)
HKU (HKEY_USERS)
HKCR (HKEY_CLASS_ROOT)
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

При использовании этого аудита обратите внимание на следующее:

- Поле **reg_key** должно содержать полный путь к разделу реестра.
- Поле **value_data** содержит имя списка ACL, определенного в файле политики.
- Поле **acl_option** может иметь значение CAN_BE_NULL или CAN_NOT_BE_NULL, чтобы установить успешное прохождение проверки или выдачу ошибки в случае отсутствия раздела.

Пример:

```
 <registry_acl: "ACL2">

  <user: "Administrators">
   acl_inheritance: "not inherited"
   acl_apply: "This key and subkeys"
   acl_allow: "Full Control"
  </user>

  <user: "SYSTEM">
   acl_inheritance: "not inherited"
   acl_apply: "This key and subkeys"
   acl_allow: "Full Control"
  </user>

 </acl>

 <custom_item>
  type: REGISTRY_PERMISSIONS
  description: "Permissions for HKLM\SOFTWARE\Microsoft"
  value_type: REG_ACL
  value_data: "ACL2"
  reg_key: "HKLM\SOFTWARE\Microsoft"
 </custom_item>
```

При выполнении приведенной выше проверки модуль проверки соответствия будет проверять, соответствуют ли определенные для **HKLM\SOFTWARE\Microsoft** разрешения описанным в списке registry_acl ACL2.

REGISTRY_AUDIT

Для надлежащего выполнения этой проверки требуется удаленный доступ к реестру для удаленной системы Windows.

Использование

```
<custom_item>
 type: REGISTRY_AUDIT
 description: ["description"]
 value_type: [value_type]
 value_data: [value]
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 reg_key: ["regkeyname"]
 (optional) acl_option: [acl_option]
</custom_item>
```

Этот элемент политики проверяет правильность списка ACL для раздела реестра. Данная проверка проводится путем вызова функции **RegGetKeySecurity** для дескриптора раздела реестра.

Допустим следующий тип:

```
value_type: REG_ACL
value_data: "ACLname"
reg_key: "RegistryKeyName"
```

В качестве значения поля **reg_key** можно использовать следующие предварительно определенные пути:

```
HKLM (HKEY_LOCAL_MACHINE)
HKU (HKEY_USERS)
HKCR (HKEY_CLASS_ROOT)
```

При использовании этого аудита обратите внимание на следующее:

- Поле **reg_key** должно содержать полный путь к разделу реестра.
- Поле **value_data** содержит имя списка ACL, определенного в файле политики.
- Поле **acl_option** может иметь значение CAN_BE_NULL или CAN_NOT_BE_NULL, чтобы установить успешное прохождение проверки или выдачу ошибки в случае отсутствия раздела.
- Поля **acl_allow** и **acl_deny** соответствуют событиям аудита Successful (успешно пройден) и Failed (не пройден).

Приведем пример файла **.audit**, выполняющего аудит раздела реестра «HKLM\SOFTWARE\Microsoft» по списку контроля доступа ACL2, который не показан:

```
 <custom_item>
  type: REGISTRY_AUDIT
  description: "Audit for HKLM\SOFTWARE\Microsoft"
  value_type: REG_ACL
  value_data: "ACL2"
  reg_key: "HKLM\SOFTWARE\Microsoft"
 </custom_item>
```

SERVICE_PERMISSIONS

Использование

```
<custom_item>
 type: SERVICE PERMISSIONS
 description: ["description"]
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 value_type: [value_type]
 value_data: [value]
 (optional) check_type: [value]
 service: ["servicename"]
 (optional) acl_option: [acl_option]
</custom_item>
```

Этот элемент политики проверяет правильность ACL службы. Данная проверка проводится путем вызова функции **QueryServiceObjectSecurity** для дескриптора службы.

Допустим следующий тип:

```
value_type: SERVICE_ACL
value_data: "ACLname"
service: "ServiceName"
```

При использовании этого аудита обратите внимание на следующее:

- Поле **value_data** содержит имя списка ACL, определенного в файле политики.
- Поле **acl_option** может иметь значение CAN_BE_NULL или CAN_NOT_BE_NULL, чтобы установить успешное прохождение проверки или выдачу ошибки в случае отсутствия раздела.

Пример:

```
 <service_acl: "ACL3">

  <user: "Administrators">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "query template" | "change template" | "query status" |
        "enumerate dependents" | "start" | "stop" | "pause and continue" |
        "interrogate" | "user-defined control" | "delete" | "read
        permissions" | "change permissions" | "take ownership"
  </user>

  <user: "SYSTEM">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "query template" | "change template" | "query status" |
        "enumerate dependents" | "start" | "stop" | "pause and continue" |
        "interrogate" | "user-defined control" | "delete" | "read
        permissions" | "change permissions" | "take ownership"
  </user>

  <user: "Interactive">
   acl inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "query template" | "query status" | "enumerate dependents" |
        "interrogate" | "user-defined control" | "read permissions"
  </user>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
  <user: "Everyone">
   acl_inheritance: "not inherited"
   acl_apply: "This object only"
   acl_allow: "query template" | "change template" | "query status" |
        "enumerate dependents" | "start" | "stop" | "pause and continue" |
        "interrogate" | "user-defined control" | "delete" | "read
        permissions" | "change permissions" | "take ownership"
  </user>

 </acl>

 <custom_item>
  type: SERVICE_PERMISSIONS
  description: "Permissions for Alerter Service"
  value_type: SERVICE_ACL
  value_data: "ACL3"
  service: "Alerter"
 </custom_item>
```

При выполнении приведенной выше проверки модуль проверки соответствия будет проверять, соответствуют ли определенные для службы оповещений разрешения описанным в списке service_acl ACL3.

SERVICE_AUDIT

Использование

```
<custom_item>
 type: SERVICE_AUDIT
 description: ["description"]
 value_type: [value_type]
 value_data: [value]
 (optional) check_type: [value]
 service: ["servicename"]
 (optional) acl option: [acl option]
</custom_item>
```

Этот элемент политики проверяет правильность ACL службы. Данная проверка проводится путем вызова функции **`QueryServiceObjectSecurity`** для дескриптора службы.

Допустим следующий тип:

```
value_type: SERVICE_ACL
value_data: "ACLname"
service: "ServiceName"
```

При использовании этого типа аудита обратите внимание на следующее:

- Поле **`value_data`** содержит имя списка ACL, определенного в файле политики.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

- Поле **acl_option** может иметь значение CAN_BE_NULL или CAN_NOT_BE_NULL, чтобы установить успешное прохождение проверки или выдачу ошибки в случае отсутствия раздела.
- Поля **acl_allow** и **acl_deny** соответствуют событиям аудита Successful (успешно пройден) и Failed (не пройден).

Приведем пример файла **.audit** для проведения аудита службы оповещений Alerter:

```
<custom_item>
 type: SERVICE_AUDIT
 description: "Audit for Alerter Service"
 value_type: SERVICE_ACL
 value_data: "ACL3"
 service: "Alerter"
</custom_item>
```

WMI_POLICY

Использование

```
<custom_item>
 type: WMI_POLICY
 description: "Test for WMI Value"
 value_type: [value_type]
 value_data: [value]
 (optional) check_type: [value]
 wmi_namespace: ["namespace"]
 wmi_request: ["request select statement"]
 wmi_attribute: ["attribute"]
 wmi_key: ["key"]
</custom_item>
```

Эта проверка запрашивает в базе данных WMI операционной системы Windows значения, указанные в пространстве имен/классе/атрибуте.

В зависимости от используемого синтаксиса могут быть извлечены значения реестра или получены в виде перечисления имена атрибутов.

Допустимы следующие типы:

```
wmi_namespace: "namespace"
wmi_request: "WMI Query"
wmi_attribute: "Name"
wmi_key: "Name"
wmi_option: option
```

Если выбрать конфигурацию службы с продублированными в системе значениями (например, MSFTPSVC/83207416 и MSFTPSVC/2), то запрос извлечет выбранный атрибут из обеих. Если одно из этих значений не соответствует значению политики, то в отчет будет добавлено значение **wmi_key**, чтобы отметить, какое именно. Поле

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

wmi_enum позволяет получить перечисление имен конфигураций в пространстве имен для сравнения или проверки значения политики. См. приведенные ниже примеры.

Пример 1:

```
<custom_item>
 type: WMI_POLICY
 description: "IIS test"
 value_type: POLICY_DWORD
 value_data: 0
 wmi_namespace: "root/MicrosoftIISv2"
 wmi_request: "SELECT Name, UserIsolationMode FROM IIsFtpServerSetting"
 wmi_attribute: "UserIsolationMode"
 wmi_key: "Name"
</custom_item>
```

При наличии в системе двух конфигураций службы FTP (MSFTPSVC/83207416 и MSFTPSVC/2) данный запрос извлечет атрибут UserIsolationMode для обеих служб. Если одно из значений этого атрибута не соответствует значению политики (0), то значение **wmi_key** (в данном случае) будет добавлено в отчет, чтобы отметить, какое именно.

Пример 2:

```
<custom_item>
 type: WMI_POLICY
 description: "IIS test2"
 value_type: POLICY_MULTI_TEXT
 value_data: "MSFTPSVC/83207416" && "MSFTPSVC/2"
 wmi_namespace: "root/MicrosoftIISv2"
 wmi_request: "SELECT Name FROM IIsFtpServerSetting"
 wmi_attribute: "Name"
 wmi_key: "Name"
 wmi_option: WMI_ENUM
</custom_item>
```

В этом примере выполняется проверка существования двух действительных имен конфигураций, указанных в поле **value_data**. Microsoft WMI CIM Studio – это ценное средство, позволяющее подробнее узнать о пространстве имен WMI и связанных с ним атрибутах, которое можно загрузить по ссылке: http://www.microsoft.com/downloads/details.aspx?FamilyID=6430f853-1120-48db-8cc5-f2abdc3ed314&displaylang=en

Элементы

Элементы (item) — это типы проверок, которые предварительно определены в Windows Compliance Checks Engine (система проверки соответствия Windows). Они используются для часто проверяемых элементов и для сокращения до минимума синтаксиса, необходимого для создания аудиторских проверок. Элемент имеет следующую структуру:

```
<item>
 name: ["predefined_entry"]
 value: [value]
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
</item>
```

Поле **name** должно иметь имя, которое уже определено (предварительно определенные имена перечислены в таблице «Предварительно определенные политики» ниже).

Все предварительно определенные элементы соответствуют списку, существующему в редакторе политик домена ОС Windows 2003 SP1.

Следующий пример проверяет, установлена ли минимальная длина пароля в диапазоне от 8 до 14 символов:

```
<item>
 name: "Minimum password length"
 value: [8..14]
</item>
```

Соответствующий пользовательский элемент следующий:

```
<custom_item>
 type: PASSWORD_POLICY
 description: "Minimum password length"
 value_type: POLICY_DWORD
 value_data: [8..14]
 password_policy: MINIMUM_PASSWORD_LENGTH
</custom_item>
```

Предварительно определенные политики

Политика	Использование
Политика в отношении паролей	name: "Enforce password policy" value: POLICY_DWORD name: "Maximum password age" value: TIME_DAY name: "Minimum password age" value: TIME_DAY name: "Minimum password length" value: POLICY_DWORD name: "Password must meet complexity requirements" value: POLICY_SET
Политика блокировки учетных записей	name: "Account lockout duration" value: TIME_MINUTE or name: "Account lockout duration" value: TIME_SECOND name: "Account lockout threshold"

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	value: POLICY_DWORD name: "Reset lockout account counter after" value: TIME_MINUTE name: "Enforce user logon restrictions" value: POLICY_SET
Политика Kerberos	name: "Maximum lifetime for service ticket" value: TIME_MINUTE name: "Maximum lifetime for user ticket" value: TIME_HOUR name: "Maximum lifetime for user renewal ticket" value: TIME_DAY name: "Maximum tolerance for computer clock synchronization" value: TIME_MINUTE
Политика аудита	name: "Audit account logon events" value: AUDIT_SET name: "Audit account management" value: AUDIT_SET name: "Audit directory service access" value: AUDIT_SET name: "Audit logon events" value: AUDIT_SET name: "Audit object access" value: AUDIT_SET name: "Audit policy change" value: AUDIT_SET name: "Audit privilege use" value: AUDIT_SET name: "Audit process tracking" value: AUDIT_SET name: "Audit system events" value: AUDIT_SET
Учетные записи	name: "Accounts: Administrator account status" value: POLICY_SET name: "Accounts: Guest account status" value: POLICY_SET name: "Accounts: Limit local account use of blank password to console logon only" value: POLICY_SET

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	name: "Accounts: Rename administrator account" value: POLICY_TEXT name: "Accounts: Rename guest account" value: POLICY_TEXT
Аудит	name: "Audit: Audit the access of global system objects" value: POLICY_SET name: "Audit: Audit the use of Backup and Restore privilege" value: POLICY_SET name: "Audit: Shut down system immediately if unable to log security audits" value: POLICY_SET
DCOM	name: "DCOM: Machine Launch Restrictions in Security Descriptor Definition Language (SDDL) syntax" value: POLICY_TEXT name: "DCOM: Machine Access Restrictions in Security Descriptor Definition Language (SDDL) syntax" value: POLICY_TEXT
Устройства	name: "Devices: Allow undock without having to log on" value: POLICY_SET name: "Devices: Allowed to format and eject removable media" value: DASD_SET name: "Devices: Prevent users from installing printer drivers" value: POLICY_SET name: "Devices: Restrict CD-ROM access to locally logged-on user only" value: POLICY_SET name: "Devices: Restrict floppy access to locally logged-on user only" value: POLICY_SET name: "Devices: Unsigned driver installation behavior" value: DRIVER_SET
Контроллер домена	name: "Domain controller: Allow server operators to schedule tasks" value: POLICY_SET name: "Domain controller: LDAP server signing requirements"

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	value: LDAP_SET name: "Domain controller: Refuse machine account password changes" value: POLICY_SET
Член домена	name: "Domain member: Digitally encrypt or sign secure channel data (always)" value: POLICY_SET name: "Domain member: Digitally encrypt secure channel data (when possible)" value: POLICY_SET name: "Domain member: Digitally sign secure channel data (when possible)" value: POLICY_SET name: "Domain member: Disable machine account password changes" value: POLICY_SET name: "Domain member: Maximum machine account password age" value: POLICY_DAY name: "Domain member: Require strong (Windows 2000 or later) session key" value: POLICY_SET
Интерактивный вход в систему	name: "Interactive logon: Display user information when the session is locked" value: LOCKEDID_SET name: "Interactive logon: Do not display last user name" value: POLICY_SET name: "Interactive logon: Do not require CTRL+ALT+DEL" value: POLICY_SET name: "Interactive logon: Message text for users attempting to log on" value: POLICY_TEXT name: "Interactive logon: Message title for users attempting to log on" value: POLICY_TEXT name: "Interactive logon: Number of previous logons to cache (in case domain controller is not available)" value: POLICY_DWORD name: "Interactive logon: Prompt user to change

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	password before expiration" value: POLICY_DWORD name: "Interactive logon: Require Domain Controller authentication to unlock workstation" value: POLICY_SET name: "Interactive logon: Require smart card" value: POLICY_SET name: "Interactive logon: Smart card removal behavior" value: SMARTCARD_SET
Клиент для сетей Майкрософт	name: "Microsoft network client: Digitally sign communications (always)" value: POLICY_SET name: "Microsoft network client: Digitally sign communications (if server agrees)" value: POLICY_SET name: "Microsoft network client: Send unencrypted password to third-party SMB servers" value: POLICY_SET
Сервер для сетей Майкрософт	name: "Microsoft network server: Amount of idle time required before suspending session" value: POLICY_DWORD name: "Microsoft network server: Digitally sign communications (always)" value: POLICY_SET name: "Microsoft network server: Digitally sign communications (if client agrees)" value: POLICY_SET name: "Microsoft network server: Disconnect clients when logon hours expire" value: POLICY_SET
Доступ к сети	name: "Network access: Allow anonymous SID/Name translation" value: POLICY_SET name: "Network access: Do not allow anonymous enumeration of SAM accounts" value: POLICY_SET name: "Network access: Do not allow anonymous enumeration of SAM accounts and shares" value: POLICY_SET name: "Network access: Do not allow storage of credentials or .NET Passports for network authentication"

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	value: POLICY_SET name: "Network access: Let Everyone permissions apply to anonymous users" value: POLICY_SET name: "Network access: Named Pipes that can be accessed anonymously" value: POLICY_MULTI_TEXT name: "Network access: Remotely accessible registry paths and sub-paths" value: POLICY_MULTI_TEXT name: "Network access: Remotely accessible registry paths" value: POLICY_MULTI_TEXT name: "Network access: Restrict anonymous access to Named Pipes and Shares" value: POLICY_SET name: "Network access: Shares that can be accessed anonymously" value: POLICY_MULTI_TEXT name: "Network access: Sharing and security model for local accounts" value: LOCALACCOUNT_SET
Сетевая безопасность	name: "Network security: Do not store LAN Manager hash value on next password change" value: POLICY_SET name: "Network security: Force logoff when logon hours expire" value: POLICY_SET name: "Network security: LAN Manager authentication level" value: LANMAN_SET name: "Network security: LDAP client signing requirements" value: LDAPCLIENT_SET name: "Network security: Minimum session security for NTLM SSP based (including secure RPC) clients" value: NTLMSSP_SET name: "Network security: Minimum session security for NTLM SSP based (including secure RPC) servers" value: NTLMSSP_SET
Консоль восстановления	name: "Recovery console: Allow automatic administrative logon"

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	value: POLICY_SET name: "Recovery console: Allow floppy copy and access to all drives and all folders" value: POLICY_SET
Отключение	name: "Shutdown: Allow system to be shut down without having to log on" value: POLICY_SET name: "Shutdown: Clear virtual memory pagefile" value: POLICY_SET
Системная криптография	name: "System cryptography: Force strong key protection for user keys stored on the computer" value: CRYPTO_SET name: "System cryptography: Use FIPS compliant algorithms for encryption, hashing, and signing" value: POLICY_SET
Системные объекты	name: "System objects: Default owner for objects created by members of the Administrators group" value: OBJECT_SET name: "System objects: Require case insensitivity for non-Windows subsystems" value: POLICY_SET name: "System objects: Strengthen default permissions of internal system objects (e.g. Symbolic Links)" value: POLICY_SET
Системные настройки	name: "System settings: Optional subsystems" value: POLICY_MULTI_TEXT name: "System settings: Use Certificate Rules on Windows Executables for Software Restriction Policies" value: POLICY_SET
Журнал событий	name: "Maximum application log size" value: POLICY_KBYTE name: "Maximum security log size" value: POLICY_KBYTE name: "Maximum system log size" value: POLICY_KBYTE name: "Prevent local guests group from accessing application log" value: POLICY_SET name: "Prevent local guests group from accessing security log"

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
value: POLICY_SET

name: "Prevent local guests group from accessing
system log"
value: POLICY_SET

name: "Retain application log"
value: POLICY_DAY

name: "Retain security log"
value: POLICY_DAY

name: "Retain system log"
value: POLICY_DAY

name: "Retention method for application log"
value: EVENT_METHOD

name: "Retention method for security log"
value: EVENT_METHOD

name: "Retention method for system log"
value: EVENT_METHOD
```

Принудительное информирование

Можно принудительно получать от политик аудита определенные результаты, используя ключевое слово **report**. Можно использовать типы отчетов PASSED (пройдено), FAILED (не пройдено) и WARNING (предупреждение). Ниже приведен пример политики:

```
<report type: "WARNING">
  description: "Audit 103-a requires a physical inspection of the pod bay
       doors Hal"
</report>
```

Текст внутри поля **description** будет всегда отображаться в отчете.

Это тип отчетности полезен, если нужно сообщить аудитору, что фактическая проверка, выполняемая Nessus, не может быть завершена. Например, возможно, существует требование определить, что определенная система была физически защищена, и мы хотим, чтобы аудитор выполнил проверку или инспекцию вручную. Этот тип отчета также полезен, если конкретный тип аудита, который должен провести сканер Nessus, проверкой OVAL определен не был.

Условия

Существует возможность определить логику **if/then/else** в политике ОС Windows только для запуска проверки, если предварительные условия выполняются, или для группировки нескольких тестов в один.

Синтаксис выполнения условий следующий:

```
<if>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 <condition type: "or">
  <Insert your audit here>
 </condition>
 <then>
  <Insert your audit here>
 </then>
 <else>
  <Insert your audit here>
 </else>
</if>
```

Условия могут быть типа **and** (и) либо **or** (или).

Аудит для операторов условия, **then** и **else** может быть списком элементов (или пользовательскими элементами) либо оператором **if**. Операторы **else** и **then** могут также использовать тип **report**, чтобы сообщать об успешном или неудачном выполнении в зависимости от возвращаемого условием значения:

```
 <report type:"PASSED|FAILED">
  description: "the test passed (or failed)"
  (optional) severity: INFO|MEDIUM|HIGH
 </report>
```

Значение **if** возвращает SUCCESS (успех) или FAILURE (неудача) и это значение используется, если оператор **if** располагается внутри другой конструкции **if**. Например, если выполняется конструкция **<then>**, то возвращаемое значение будет одним из следующих:

- аудит содержит только элементы: возвращается SUCCESS (успех), если элементы успешно выполнены, иначе возвращается FAILURE (неудача);
- аудит содержит только **<report>**: возвращается тип report;
- аудит содержит одновременно элементы и **<report>**: возвращается тип report.

Если используется оператор **<report>** и тип FAILED (неудача), то причина неудачного выполнения будет показана в отчете вместе с уровнем серьезности, если уровни были определены.

Далее приведен пример аудита политики в отношении паролей. Так как используется тип **and** (и), для этой политики успешное прохождение аудита предполагает успешное выполнение обоих пользовательских элементов. Этот пример испытывает очень странное сочетание действительных политик в отношении ведения журнала паролей, чтобы показать, насколько сложную логику проверки можно реализовать:

```
 <if>
  <condition type:"and">
   <custom_item>
    type: PASSWORD_POLICY
    description: "2.2.2.5 Password History: 24 passwords remembered"
    value_type: POLICY_DWORD
    value_data: [22..MAX] || 20
    password_policy: ENFORCE_PASSWORD_HISTORY
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
  </custom_item>
  <custom_item>
   type: PASSWORD_POLICY
   description: "2.2.2.5 Password History: 24 passwords remembered"
   value_type: POLICY_DWORD
   value_data: 18 || [4..24]
   password_policy: ENFORCE_PASSWORD_HISTORY
  </custom_item>
 </condition>

 <then>
  <report type:"PASSED">
   description: "Password policy passed"
  </report>
 </then>

 <else>
  <report type:"FAILED">
   description: "Password policy failed"
  </report>
 </else>
</if>
```

В приведенном выше примере показан только новый тип **report** (отчет), но структура **if/then/else** поддерживает выполнение дополнительных аудитов внутри инструкций **else**. Внутри условия также можно использовать вложенные инструкции **if/then/else**. Более сложный пример приведен ниже:

```
<if>
 <condition type:"and">
  <custom_item>
   type: CHECK_ACCOUNT
   description: "Accounts: Rename Administrator account"
   value_type: POLICY_TEXT
   value_data: "Administrator"
   account_type: ADMINISTRATOR_ACCOUNT
   check_type: CHECK_NOT_EQUAL
  </custom_item>
 </condition>

 <then>
  <report type:"PASSED">
   description: "Administrator account policy passed"
  </report>
 </then>

 <else>
  <if>
   <condition type:"or">
    <item>
     name: "Minimum password age"
     value: [1..30]
    </item>
    <custom_item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
      type: PASSWORD_POLICY
      description: "Password Policy setting"
      value_type: POLICY_SET
      value_data: "Enabled"
      password_policy: COMPLEXITY_REQUIREMENTS
     </custom_item>
    </condition>

    <then>
     <report type:"PASSED">
      description: "Administrator account policy passed"
     </report>
    </then>

    <else>
     <report type:"FAILED">
      description: "Administrator account policy failed"
     </report>
    </else>
   </if>

  </else>
 </if>
```

В этом примере, если учетная запись Administrator не была переименована, то проводится аудит правила, что минимальный срок действия пароля не должен превышать 30 дней. Эта политика аудита будет проверена успешно, если учетная запись администратора была переименована, независимо от политики в отношении паролей, а политика в отношении срока действия паролей будет проверяться только в том случае, если учетная запись администратора не была переименована.

СПРАВКА ПО ФАЙЛАМ ПРОВЕРКИ СООТВЕТСТВИЯ СОДЕРЖИМОГО ОС WINDOWS .AUDIT

Проверки **`.audit`** Windows Content (содержимое Windows) отличаются от проверок **`.audit`** Windows Configuration (конфигурация Windows) тем, что они предназначены для выполнения поиска в файловой системе ОС Windows файлов определенного типа, содержащих конфиденциальные данные, а не для перечисления параметров конфигурации системы. Они включают ряд дополнительных параметров, которые помогают аудитору сузить параметры поиска и более эффективно определить расположение несоответствующих требованиям данных и показать их.

Использование кавычек

Одинарные и двойные кавычки при заключении в них полей аудита являются взаимозаменяемыми, за исключением следующих двух случаев:

1. В проверках соответствия операционной системы Windows, если специальные поля, например CRLF и т. д., должны интерпретироваться буквально, используйте одинарные кавычки. Любые оказавшиеся внутри этих кавычек поля, которые должны интерпретироваться как строки, следует выделять.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Например:

```
expect: 'First line\r\nSecond line\r\nJohn\'s Line'
```

2. Двойные кавычки требуются при использовании полей include_paths и exclude_paths аудита WindowsFiles.

При использовании в поле любого типа (description, value_data, regex и т. д.) строк, содержащих одинарные или двойные кавычки, существует два способа их обработки:

a. В качестве наружных кавычек можно использовать кавычки противоположного типа.

Например:

```
expect: "This is John's Line"
expect: 'We are looking for a double-quote-".*'
```

b. Можно выделять внутренние кавычки с помощью обратной наклонной черты (только двойные кавычки).

Например:

```
expect: "\"Text to be searched\""
```

Тип проверки

Все проверки соответствия содержимого ОС Windows должны заключаться в теги **`check_type`** и иметь обозначение типа **`WindowsFiles`**. Это очень схоже со всеми остальными файлами **`.audit`**. Общий формат файла проверки содержимого следующий:

```
<check_type: "WindowsFiles">

<item>

</item>

<item>

</item>

<item>

</item>

</check_type>
```

Фактические проверки для каждого элемента не показаны. В следующих разделах показано, как можно использовать различные ключевые слова и параметры для составления конкретных проверок элементов содержимого.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Формат элементов

Использование
<item> type: FILE_CONTENT_CHECK description: ["value data"] file_extension: ["value data"] (optional) regex: ["value data"] (optional) expect: ["value data"] (optional) file_name: ["value data"] (optional) max_size: ["value data"] (optional) only_show: ["value data"] (optional) regex_replace: ["value data"] </item>

Каждый из этих элементов используется для проверки файлов очень многих форматов, содержащих данные различных типов. В следующей таблице приведен список поддерживаемых типов данных. В следующем разделе приведены многочисленные примеры того, как эти ключевые слова могут использоваться в совокупности для проведения аудита различных типов содержимого файлов.

Ключевое слово	Описание
type	Для этого ключевого слова во всех случаях должно быть установлено значение FILE_CONTENT_CHECK
description	Это информация, которая будет использоваться в качестве названия уникальных уязвимостей соответствия требованиям в SecurityCenter. Она также будет первой совокупностью данных, сообщаемой сканером Nessus.
file_extension	Это ключевое слово содержит перечисление всех расширений, которые должны разыскиваться сканером Nessus. Расширения перечисляются без точки (.) в кавычках и разделяются вертикальными линиями. Если дополнительные параметры, например **regex** и **expect**, в аудит не включены, в результатах аудита отображаются файлы с указанными в поле file_extension расширениями.
regex	В этом ключевом слове хранится регулярное выражение, используемое для поиска сложных типов данных. В случае совпадения с регулярным выражением, первый случай совпадения содержимого отображается в отчете уязвимостей.  Ключевое слово **regex** должно выполняться с ключевым словом **expect**, описанным ниже.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	 В отличие от проверок Windows Compliance Checks (проверки соответствия Windows), в проверке Windows File Content Compliance Check (проверка соответствия содержимого файлов Windows) ключевые слова **regex** и **expect** не требуют совпадения с одними и теми же строками данных в файле, по которому производится поиск. Проверки Windows File Content (содержимое файлов Windows) требуют просто наличия совпадения данных для обоих операторов, **regex** и **expect**, в пределах количества байт файла, по которому производится поиск, указанному в поле **<max_size>**.
expect	Оператор **expect** используется для перечисления одного или нескольких простых образцов текста, которые должны содержаться в документе, чтобы он соответствовал условию поиска. Например, при поиске номеров Social Security (номеров социального страхования) может требоваться наличие слов SSN, SS# или Social. Несколько образцов перечисляются в кавычках и разделяются символами вертикальной черты. Проверка соответствия простым образцам текста также поддерживается в этом ключевом слове с помощью точки. При сравнении со строкой «C.T» оператор **expect** будет проверять соответствие строкам «CAT», «CaT», «COT», «C T» и т. д.  Ключевое слово **expect** может выполняться отдельно для сравнения с одним образцом текста, однако если используется ключевое слово **regex** , то использование ключевого слова **expect** обязательно.  В отличие от проверок Windows Compliance Checks (проверки соответствия Windows), в проверке Windows File Content Compliance Check (проверка соответствия содержимого файлов Windows) ключевые слова **regex** и **expect** не требуют совпадения с одними и теми же строками данных в файле, по которому производится поиск. Проверки Windows File Content (содержимое файлов Windows) требуют просто наличия совпадения данных для обоих операторов, **regex** и **expect**, в пределах

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	количества байт файла, по которому производится поиск, указанному в поле **<max_size>**.
file_name	Если требуется ключевое слово **file_extension**, то оно может дополнительно уточнить список файлов, которые необходимо анализировать. Посредством списка образцов файлы могут отбрасываться или сравниваться. Например, это позволяет с легкостью искать имена файлов любого типа, включающие такие слова, как employee (работник), customer (заказчик) или salary (заработная плата).
max_size	Для повышения производительности аудит может проводиться только по началу каждого файла. Размер проверяемой части можно указать в байтах с помощью этого ключевого слова. Количество байт указывается в качестве аргумента. Также поддерживаются расширения «K» и «M» для килобайт и мегабайт, соответственно.
only_show	При проверке соответствия конфиденциальных данных, например номеров кредитных карт, организации может потребоваться отображение в отчете только последних четырех цифр. Это ключевое слово поддерживает отображение любого количества байт, определенного политикой.
regex_replace	Это ключевое слово позволяет контролировать, какой образец из регулярного выражения будет показан в отчете. При поиске сложных образцов данных, например номеров кредитных карт, не всегда искомыми данными оказывается первое совпадение. Это ключевое слово обеспечивает большую гибкость для нахождения необходимых данных с более высокой точностью.
include_paths	Это ключевое слово позволяет включать каталоги и диски в результаты поиска. Это ключевое слово может использоваться в сочетании с ключевым словом **exclude_paths** или самостоятельно. Это особенно полезно в случаях, когда необходимо выполнить поиск только по определенным дискам или папкам системы с несколькими дисками. Если требуется указать несколько путей, то они заключаются в двойные кавычки и разделяются символом вертикальной черты.  В качестве значений ключевого слова **include_paths** могут быть указаны только буквы дисков и имена папок. Имена файлов не могут быть включены в строку значений ключевого слова **include_paths**.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

exclude_paths	Это ключевое слово позволяет исключать диски, каталоги и файлы из результатов поиска. Это ключевое слово может использоваться в сочетании с ключевым словом **include_paths** или самостоятельно. Это особенно удобно в том случае, если какой-то конкретный диск, каталог или файл необходимо исключить из результатов поиска. Если требуется указать несколько путей, то они заключаются в двойные кавычки и разделяются символом вертикальной черты.

Примеры командных строк

В этом разделе мы создадим искусственный текстовый документ с расширением **.tns** и затем проверим его несколькими файлами **.audit**, от простого к сложному. Рассматривая каждый пример, мы испытаем каждый поддерживаемый случай применения параметров проверок Windows Content.

Мы также используем двоичный файл командной строки **nasl**. Каждый демонстрируемый нами файл **.audit** можно легко включить в политики сканирования Nessus 4 или SecurityCenter, но для быстрого аудита одной системы данный способ очень эффективен. Команда, которую мы будем каждый раз выполнять из каталога **/opt/nessus/bin**, будет следующей:

```
# ./nasl -t <IP>
       /opt/nessus/lib/nessus/plugins/compliance_check_windows_file_conten
       t.nbin
```

Здесь <IP> — это IP-адрес системы, аудит которой мы проводим.

В случае Nessus 4 при выполнении файла **.nbin** (или любого другого подключаемого модуля) будет выводиться запрос указать реквизиты входа для целевой системы, а также расположение файла **.audit**.

Целевой файл для испытания

Целевой файл, который мы будем использовать, содержит следующий текст:

```
abcdefghijklmnopqrstuvwxyz
01234567890
Tenable Network Security
SecurityCenter
Nessus
Passive Vulnerability Scanner
Log Correlation Engine
AB12CD34EF56
Nessus
```

Скопируйте эти данные в любую систему под управлением ОС Windows, для которой у вас есть реквизиты входа. Назовите файл Tenable_Content.tns.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Пример 1: поиск документов с расширением TNS, содержащих слово «Nessus»

Далее приведен простой файл **.audit**, выполняющий поиск любого файла с расширением **.tns**, содержащего слово «Nessus» в любой части документа.

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS File that Contains the word Nessus"
 file_extension: "tns"
 expect: "Nessus"
</item>
</check_type>
```

При выполнении этой команды ожидается следующий результат:

```
"TNS File that Contains the word Nessus" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns
```

Эти результаты показывают, что мы нашли совпадение. В отчете сказано «failed» (неудача), потому что мы нашли данные, которые не хотели находить. Например, если мы проводим аудит в списках номера Social Security (номера социального страхования) и получаем положительное соответствие номера Social Security на общедоступном компьютере, то, несмотря на положительный результат поиска соответствий, он записывается как выполнение требований соответствия.

Пример 2: поиск документов с расширением TNS, содержащих слово «France»

Далее приведен простой файл **.audit**, выполняющий поиск любого файла с расширением **.tns**, содержащего слово «France» в любой части документа.

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS File that Contains the word France"
 file_extension: "tns"
 expect: "France"
</item>
</check_type>
```

На этот раз мы получим следующий результат:

```
"TNS File that Contains the word France" : [PASSED]
```

Мы прошли аудиторскую проверку, потому что ни один из проверенных нами файлов с расширением **.tns** не содержит слова «France».

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Пример 3: поиск документов с расширением TNS и DOC, содержащих слово «Nessus»

Как показано ниже, добавить второе расширение для поиска по документам Microsoft Word очень просто:

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS or DOC File that Contains the word Nessus"
 file_extension: "tns" | "doc"
 expect: "Nessus"
</item>
</check_type>
```

Результаты (на нашем тестовом компьютере) были следующими:

```
"TNS or DOC File that Contains the word Nessus" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns
Share: C$, path: \documents and settings\jsmith\desktop\tns_roadmap.doc
```

Мы получили такую же «неудачу», как и прежде при испытании файла с расширением **.tns**, но в данном случае был найден второй файл с расширением **.doc**, который также содержал слово «Nessus». При выполнении этих тестов на своем собственном компьютере файл Word со словом «Nessus» может существовать, а может и отсутствовать.

Пример 4: поиск документов с расширением TNS и DOC, содержащих слово «Nessus» и 11-значное число

Теперь мы добавим наше первое регулярное выражение для поиска 11-значного числа. Необходимо только добавить регулярное выражение с ключевым словом **regex** в то же файл **.audit**.

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS or DOC File that Contains the word Nessus"
 file_extension: "tns" | "doc"
 regex: " ([0-9]{11})"
 expect: "Nessus"
</item>
</check_type>
```

Выполнение этих процедур приведет к следующему результату:

```
"TNS or DOC File that Contains the word Nessus" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns    (01234567890)
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Поиск файла с расширением **.doc**, который соответствует критериям поиска, продолжается. Так как в нем нет 11-значного числа, он больше не отображается. Кроме того, обратите внимание, что из-за использования ключевого слова **regex** мы также получаем показанное в данных совпадение.

А что если нам потребуется найти 10-значное число? Приведенное выше 11-значное число содержит два 10-значных числа (0123456789 и 1234567890). Если нам нужно написать более точное условие соответствия только для 11-цифр, нам необходимо регулярное выражение, говорящее:

«Искать любое 11-значное число, перед которым и после которого не следует каких-либо иных чисел».

Для этого в регулярные выражения можно добавить оператор not следующим образом:

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS or DOC File that Contains the word Nessus"
 file_extension: "tns" | "doc"
 regex: "([^0-9]|^)([0-9]{11})([^0-9]|$)"
 expect: "Nessus"
</item>
</check_type>
```

Читая слева направо, мы также видим несколько раз символ «^» и символ доллара. Символ «^» иногда означает начало строки, а в других случаях означает противоположное сравнение. Знак доллара означает конец строки. Приведенное выше регулярное выражение в общем случае означает: искать комбинации символов, не начинающиеся с числа, но, возможно, начинающиеся на новой строке, содержащие 11 цифр, а также не сопровождающиеся какими-либо еще цифрами или за которыми следует конец строки. Регулярные выражения обрабатывают начало строки и конец строки как особые случая, поэтому требуется использование символов «^» или «$».

Пример 5: поиск документов с расширением TNS и DOC, содержащих слово «Nessus» и 11-значное число, но отображать следует только последние 4 байта

Добавление ключевого слова **only_show** в наш файл **.audit** позволяет ограничить выводимые результаты. Это позволяет ограничить доступ проводящих аудит лиц к просматриваемым конфиденциальным данным.

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS or DOC File that Contains the word Nessus"
 file_extension: "tns" | "doc"
 regex: "([^0-9]|^)([0-9]{11})([^0-9]|$)"
 expect: "Nessus"
 only_show: "4"
</item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
</check_type>
```

При получении совпадения данные заменяются символами «X», как показано ниже:

```
"TNS or DOC File that Contains the word Nessus" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns    (XXXXXXX7890)
```

Пример 6: поиск документов с расширением TNS, содержащих слово «Correlation» в первых 50 байтах

В этом примере мы будем изучать использование ключевого слова **max_size**. В нашем тестовом файле слово Correlation расположено не в первых 50 байтах файла.

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS File that Contains the word Correlation"
 file_extension: "tns"
 expect: "Correlation"
 max_size: "50"
</item>
</check_type>
```

При выполнении этого кода мы получим успешную проверку на наличие соответствий:

```
"TNS File that Contains the word Correlation" : [PASSED]
```

Изменим значение **max_size** с «50» на «50K» и вновь выполним сканирование. Теперь получим ошибку:

```
"TNS File that Contains the word Correlation" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns
```

Пример 7: контроль отображаемых результатов

В этом примере мы будем изучать использование ключевого слова **regex_replace**. Представьте следующий файл **.audit**:

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "Seventh Example"
 file_extension: "tns"
 regex: "Passive Vulnerability Scanner"
 expect: "Nessus"
</item>
</check_type>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Результатом выполнения этой проверки будет:

```
"Seventh Example" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns    (Passive
       Vulnerability Scanner)
```

А теперь предположим, что нам нужно регулярное выражение, которое реагирует на части «Passive» и «Scanner», но возвращает только часть «Vulnerability». Новое регулярное выражение будет выглядеть следующим образом:

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "Seventh Example"
 file_extension: "tns"
 regex: "(Passive) (Vulnerability) (Scanner)"
 expect: "Nessus"
</item>
</check_type>
```

Результатом выполнения этой проверки все равно будет полное совпадение «Passive Vulnerability Scanner», потому что оператор регулярного выражения обрабатывает всю строку как первое совпадение. Чтобы получить только второе совпадение, необходимо добавить ключевое слово **`regex_replace`**.

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "Seventh Example"
 file_extension: "tns"
 regex: "(Passive) (Vulnerability) (Scanner)"
 regex_replace: "\3"
 expect: "Nessus"
</item>
</check_type>
```

Результатом данного сканирования будет:

```
"Seventh Example" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns    (Vulnerability)
```

Мы использовали «\3», чтобы указать на второй элемент нашего сопоставления, потому что первый элемент (\1) — это строка полностью. Если бы мы использовали «\2», то результатом бы стало слово «Passive», а если «\4», то результатом бы стало слово «Scanner».

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Для чего существует эта функциональная возможность? При поиске сложных образцов данных, например номеров кредитных карт, не всегда искомыми данными оказывается первое совпадение. Это ключевое слово обеспечивает большую гибкость для нахождения необходимых данных с более высокой точностью.

Пример 8: использование имени файла в качестве фильтра

Если рассмотреть файл **.audit** из третьего примера, то в результате были приведены одновременно файлы **.tns** и **.doc**.

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS or DOC File that Contains the word Nessus"
 file_extension: "tns" | "doc"
 expect: "Nessus"
</item>
</check_type>
```

Результаты (на нашем тестовом компьютере) были следующими:

```
"TNS or DOC File that Contains the word Nessus" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns
Share: C$, path: \documents and settings\jsmith\desktop\tns_roadmap.doc
```

Ключевое слово **file_name** можно также использовать, чтобы отфильтровать файлы, которые нам нужны или не нужны. Добавление этого ключевого слова в файл **.audit** с указанием учитывать только файлы со словом «tenable» в имени будет выглядеть следующим образом:

```
<check_type:"WindowsFiles">
<item>
 type: FILE_CONTENT_CHECK
 description: "TNS or DOC File that Contains the word Nessus"
 file_extension: "tns" | "doc"
 file_name: "tenable"
 expect: "Nessus"
</item>
</check_type>
```

Результатом выполнения будет:

```
TNS or DOC File that Contains the word Nessus" : [FAILED]
 - error message:
The following files do not match your policy :
Share: C$, path: \share\new folder\tenable_content.tns
```

Содержащий совпадения файл **.doc** отсутствует, потому что в его пути нет слова «tenable».

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Строка сравнения является регулярным выражением, поэтому она может быть очень гибкой для сопоставления широкого спектра файлов, которые мы хотим или не хотим проверять. Например, мы может использовать строку «[Tt]enable» для сопоставления со словом «Tenable» или «tenable». Аналогично, если мы хотим сравнить расширение или часть расширения, то нужно добавить точку с обратной наклонной чертой, чтобы по выражению «\.t» выполнялся поиск всех файлов с расширениями, начинающимися с буквы «t».

Пример 9: использование ключевых слов включения и исключения

Ключевые слова **include_paths** и **exclude_paths** могут использоваться для фильтрации результатов поиска путем включения буквы диска, каталога или даже имени файла.

```
<item>
 type: FILE_CONTENT_CHECK
 description: "Does the file contain 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-]|$)"
 regex_replace: "\3"
 expect:"."
 max_size : "50K"
 only_show : "4"
 include_paths: "c:\" | "g:\" | "h:\"
 exclude_paths: "g:\dontscan"
</item>
```

Результатом выполнения будет:

```
Windows File Contents Compliance Checks
"Determine if a file contains a valid VISA Credit Card Number" : [FAILED]
- error message:
The following files do not match your policy :
Share: C$, path: \documents and settings\administrator\desktop\ccn.txt
       (XXXXXXXXXXXX0552)


Nessus ID : 24760
```

Обратите внимание, что данный результат не отличается от стандартного результата поиска по содержимому файлов ОС Windows, но не включает исключенный путь. Если с помощью ключевого слова **include_paths** включен единственный путь (например, **c:**), то все остальные пути будут исключены автоматически. Кроме того, если исключена буква диска (например, **d:**), но какая-то папка на этом диске включена (например, **d:\users**), то ключевое слово **exclude_paths** обладает преимуществом, и поиск на этом диске производиться не будет.

Аудит различных типов форматов файлов

Возможен аудит файлов с любыми расширениями, но файлы таких типов, как **.zip** и **.gz**, сходу не распаковываются. Если файл сжат или к данным применено какое-то шифрование, то поиск образцов может быть невозможен.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Для документов, в которых данные хранятся в формате Юникод, процедуры анализа файла **.nbin** будут исключать все встречающиеся байты, содержащие значение NULL.

Кроме того, поддерживаются все версии документов из пакета ПО Microsoft Office, в том числе и новые шифрованные версии, добавленные в пакете Office 2007, например файлы **.xlsx** и **.docx**.

И, наконец, включена поддержка различных типов форматов PDF. Компанией Tenable создан объемный анализатор файлов формата PDF, извлекающий необработанные строки для сравнения. Пользователи должны заботиться только о том, какие данные они хотят найти в файле формата PDF.

Соображения производительности

Существует несколько компромиссов, которые любая организация должна учитывать, внося изменения в используемые по умолчанию файлы **.audit** и испытывая их на рабочих сетях:

- Поиск каких расширений следует выполнять?
- Сколько данных следует сканировать?

Файлы **.audit** не требуют ключевого слова **max_size**. В этом случае программное обеспечение Nessus пытается извлечь файл целиком и продолжает до тех пор, пока не обнаружит совпадение с образцом. Так как эти файлы передаются по сети, при проведении аудита создается больший сетевой трафик, чем при обычном сканировании или аудите конфигурации.

При управлении несколькими сканерами Nessus с помощью консоли SecurityCenter требуется перемещение данных только от сканируемого компьютера под управлением ОС Windows до сканера, выполняющего аудит уязвимостей.

Справка по файлам проверки соответствия конфигурации ОС Cisco IOS .audit

В этом разделе описывается формат и функции проверок соответствия для ОС Cisco IOS, а также приводится обоснование каждой настройки.

Использование кавычек

Одинарные и двойные кавычки при заключении в них полей аудита являются взаимозаменяемыми, за исключением следующих двух случаев:

1. В проверках соответствия операционной системы Windows, если специальные поля, например CRLF и т. д., должны интерпретироваться буквально, используйте одинарные кавычки. Любые оказавшиеся внутри этих кавычек поля, которые должны интерпретироваться как строки, следует выделять.

Например:

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
expect: 'First line\r\nSecond line\r\nJohn\'s Line'
```

2. Двойные кавычки требуются при использовании полей include_paths и exclude_paths аудита WindowsFiles.

При использовании в поле любого типа (description, value_data, regex и т. д.) строк, содержащих одинарные или двойные кавычки, существует два способа их обработки:

a. В качестве наружных кавычек можно использовать кавычки противоположного типа.

Например:

```
expect: "This is John's Line"
expect: 'We are looking for a double-quote-".*'
```

b. Можно выделять внутренние кавычки с помощью обратной наклонной черты (только двойные кавычки).

Например:

```
expect: "\"Text to be searched\""
```

Тип проверки

Все проверки соответствия для ОС Cisco IOS должны заключаться в теги **check_type** и иметь обозначение типа Cisco. Это необходимо для того, чтобы можно было отличать файлы **.audit**, предназначенные специально для систем под управлением операционной системы Cisco IOS, от аудитов соответствия иного типа.

Пример:

```
<check_type:"Cisco">
```

В отличие от аудитов соответствия других типов, в данном случае дополнительные ключевые слова type или version не поддерживаются.

Ключевые слова

В следующей таблице показано, как может быть использовано каждое ключевое слово в проверках соответствия Cisco:

Ключевое слово	Пример использования и поддерживаемые параметры
type	CONFIG_CHECK, CONFIG_CHECK_NOT и RANDOMNESS_CHECK Проверка "CONFIG_CHECK" определяет, существует ли указанный элемент в результатах выполнения команды «show config» операционной системы CISCO IOS. Таким же

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	образом проверка CONFIG_CHECK_NOT″ определяет, не существует ли определенного элемента. Проверка RANDOMNESS_CHECK″ используется для осуществления проверки сложности строк (например, проверки паролей). Если указать элемент, который требуется искать (с помощью ключевого слова regex), то данная проверка сообщит, является ли указанная строка достаточно «произвольной» (имеет длину не менее восьми символов, содержит символы верхнего регистра, нижнего регистра, по крайней мере одну цифру и один специальный символ).  Параметры произвольности в данный момент не настраиваются.
description	Ключевое слово **description** позволяет добавлять краткое описание выполняемой проверки. Настоятельно рекомендуется обеспечивать уникальность значения поля **description** и отсутствие в разных проверках одинаковых значений в полях **description**. Поставляемое компанией Tenable программное обеспечение SecurityCenter использует это поле для автоматического генерирования уникального идентификационного номера подключаемого модуля на основе значения поля **description**. Пример: **description: "Forbid Remote Startup Configuration"**

feature_set	Ключевое слово **feature_set** аналогично ключевому слову **system** в проверках соответствия для ОС Unix и проверяет версию набора функций операционной системы Cisco IOS и либо проводит результирующую проверку, либо пропускает проверку из-за несоответствия регулярному выражению. Это полезно в тех случаях, когда проверка применима только к системам с каким-то конкретным набором функций.
	
	Пример:
	`<item>`
	` type: CONFIG_CHECK`
	` description: "Version Check"`
	` info: "SSH Access Control Check."`
	` feature_set: "K8" context:"line .*"`
	` item:"access-class [0-9]+ in"`
	`</item>`
	
	Приведенная выше проверка запустит проверку item только в том случае, если версия набора функций соответствует указанному регулярному выражению: (K8)
	
	В случае невыполнения проверки версии набора функций

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	на экран будет выведена ошибка, аналогичная приведенной ниже:
	
	`"Version Check" : [SKIPPED]`
	`Test defined for the feature set 'K8' whereas we are running C850-ADVSECURITYK9-M`

ios_version	Ключевое слово **ios_version** аналогично ключевому слову **system** в проверках соответствия для ОС Unix и проверяет версию операционной системы Cisco IOS и либо проводит результирующую проверку, либо пропускает проверку из-за несоответствия регулярному выражению. Это полезно в тех случаях, когда проверка применима только к системам под управлением операционной системы IOS какой-то определенной версии.
	
	Пример:
	`<item>`
	`type: CONFIG_CHECK`
	`description: "Version Check"`
	`info: "SSH Access Control Check."`
	`ios_version: "12\.[5-9]"`
	`context:"line .*"`
	`item:"access-class [0-9]+ in"`
	`</item>`
	
	Приведенная выше проверка запустит проверку item только в том случае, если версия операционной системы IOS соответствует указанному регулярному выражению: (12\.[5-9]).
	
	В случае невыполнения проверки версии ОС IOS на экран будет выведена ошибка, аналогичная приведенной ниже:
	
	`"Version Check" : [SKIPPED]`
	`Test defined for 12.[5-9] whereas we are running 12.4(15)T10`

info	Ключевое слово **info** позволяет добавлять более подробное описание выполняемой проверки. Основанием для проверки может служить нормативный акт, URL-адрес с дополнительными сведениями, корпоративная политика и прочее. Возможно добавление нескольких полей **info** на отдельных строках, чтобы обеспечить форматирование текста в виде параграфа. Количество полей **info** не ограничено.  Каждый тег **info** должен быть написан на отдельной строке без переносов. Если необходимо несколько строк (например, по соображениям формата), то следует добавить дополнительные теги **info**.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	Пример:
	`info: "Verify at least one local user exists and ensure”`
	`info: “all locally defined user passwords are protected”`
	`info: “by encryption."`

item	Ключевое слово **item** определяет элемент конфигурации в результатах выполнения команды show config, который нужно проверить. Пример: `item:"transport input ssh"` Регулярные выражения могут использоваться внутри этого ключевого слова для фильтрации результатов сопоставления. Дополнительные сведения о функциональных возможностях ключевого слова **regex** см. в описании ключевого слова **regex**.
regex	Ключевое слово **regex** позволяет выполнять поиск параметра элемента конфигурации, соответствующего определенному регулярному выражению. Пример: `regex: "snmp-server community ([^ ]*) .*"` Следующие специальные символы требуют особого обращения: + \ * ( ) ^ Если требуется, чтобы они интерпретировались буквально, то их следует дважды исключить с помощью двух символов обратной наклонной черты (\\) или заключить в квадратные скобки ([]). Другие символы, как показанные ниже, для буквальной интерпретации требуют только одного символа обратной наклонной черты: . ? " ' Это определяется порядком обработки этих символов компилятором.
min_occurrences	Ключевое слово **min_occurrences** определяет минимальное количество случаев обнаружения определенного элемента конфигурации для успешного прохождения аудита. Пример: `min_occurrences:"3"`
max_occurrences	Ключевое слово **max_occurrences** определяет максимальное количество случаев обнаружения определенного элемента конфигурации для успешного прохождения аудита.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	Пример: `max_occurrences : "1"`
required	Ключевое слово **required** позволяет указать, обязательно ли наличие проверяемого элемента на удаленной системе. Например, если ключевое слово **required** имеет значение NO, а тип проверки «CONFIG_CHECK», то эта проверка пройдет успешно независимо от того, существует ли элемент конфигурации. С другой стороны, если ключевому слову **required** присвоено значение YES, то приведенная выше проверка пройдена не будет. Пример: `required: NO`

context	Ключевое слово **context** полезно в том случае, если существует несколько экземпляров определенного элемента конфигурации. Например, рассмотрите следующую конфигурацию:
	
	`line con 0`
	` no modem enable`
	`line aux 0`
	` access-class 42 in`
	` exec-timeout 10 0`
	` no exec`
	`line vty 0 4`
	` exec-timeout 2 0`
	` password 7 15010X1C142222362G`
	` transport input ssh`
	
	Если вы хотите проверить значение какой-то конкретной последовательной линии, то использование ключевого слова **item** со значением line (линия) будет недостаточным, потому что существует несколько случаев использования значения line (линия). В случае использования ключевого слова **context** можно сосредоточиться только на интересующем элементе. Например:
	
	`context: "con 0"`
	
	Вы получите только следующий элемент конфигурации:
	
	`line con 0`
	` no modem enable`
	
	Регулярные выражения могут использоваться внутри этого ключевого слова для фильтрации результатов сопоставления. Дополнительные сведения о функциональных возможностях ключевого слова **regex** см. в описании ключевого слова **regex**.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Примеры командных строк

В этом разделе приводятся примеры распространенных аудитов, используемых для проведения проверок соответствия систем под управлением ОС Cisco IOS. Двоичный файл командной строки **nasl** используется в качестве средства для быстрого испытания аудитов на ходу. Каждый показанный ниже файл **.audit** можно легко включить в политики сканера Nessus. Однако для быстрой проверки одной системы выполнение тестов в командной строке более эффективно. Команда, которую мы будем каждый раз выполнять из каталога **/opt/nessus/bin**, будет следующей:

```
# ./nasl -t <IP>
       /opt/nessus/lib/nessus/plugins/cisco_compliance_check.nbin
```

Здесь <IP> — это IP-адрес системы, аудит которой проводится.

Запрашивается пароль enable:

```
Which file contains your security policy ? cisco_test.audit
SSH login to connect with : admin
How do you want to authenticate ? (key or password) [password]
SSH password :
Enter the 'enable' password to use :
```

Обратитесь к администратору ОС Cisco, чтобы получить правильные параметры входа enable.

Пример 1: поиск определенного списка SNMP ACL

Ниже приведен простой файл **.audit**, выполняющий поиск определенного запрещающего трафик (deny) списка SNMP ACL. Если списка не найдено, то этот аудит выдаст сообщение о неудачной попытке. Эта проверка будет выполнена только в том случае, если версия IOS маршрутизатора совпадает с указанной с помощью регулярного выражения. Иначе проверка будет пропущена.

```
<check_type: "Cisco">

<item>
Type          : CONFIG_CHECK
 Description : "Require a Defined SNMP ACL"
 info         : "Verify a defined simple network management protocol (SNMP)
       access control list (ACL) exists with rules for restricting SNMP
       access to the device."
 ios_version : "12\.[4-9]"
 item         :"deny ip any any"
</item>

</check_type>
```

При выполнении этой команды от соответствующей требованиям системы ожидается следующий результат:

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
"Require a Defined SNMP ACL" : [PASSED]

Verify a defined simple network management protocol (SNMP) access control
       list (ACL) exists with rules for restricting SNMP access to the
       device.
```

Закончившийся ошибкой аудит приведет к следующему результату:

```
"Require a Defined SNMP ACL" : [FAILED]

Verify a defined simple network management protocol (SNMP) access control
       list (ACL) exists with rules for restricting SNMP access to the
       device.

 - error message: deny ip any any not found in the configuration file
```

В этом случае проверка была неудачной, потому что мы искали правило deny ip и не нашли ни одного.

Пример 2: проверка того, что служба finger отключена

Ниже приведен текст простого файла **.audit**, выполняющего поиск небезопасной службы **finger** на удаленном маршрутизаторе. Эта проверка будет выполнена только в том случае, если версия IOS маршрутизатора совпадает с указанной с помощью регулярного выражения. Иначе проверка будет пропущена. Если служба не будет найдена, то этот аудит выдаст сообщение о неудачной попытке.

```
<check_type: "Cisco">

<item>
 type         : CONFIG_CHECK_NOT
 description  : "Forbid Finger Service"
 ios_version  : "12\.[4-9]"
 info         : "Disable finger server."
 item         : "(ip|service) finger"
</item>

</check_type>
```

При выполнении этой команды от соответствующей требованиям системы ожидается следующий результат:

```
"Forbid Finger Service" : [PASSED]

Disable finger server.
```

Закончившийся ошибкой аудит приведет к следующему результату:

```
"Forbid Finger Service" : [FAILED]
Disable finger server.
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 - error message:
The following configuration line is set:
ip finger <----

Policy value:
(ip|service) finger
```

Пример 3: проверка произвольности, выполняемая, чтобы убедиться, что строки доступа и средства контроля доступа протокола SNMP достаточно произвольные

Ниже приведен текст простого файла **.audit**, выполняющего поиск строк доступа SNMP, которые недостаточно произвольны. Если найдена строка доступа, которая недостаточно произвольна, то аудит выведет на экран сообщение о том, что проверка не пройдена. Так как параметр required имеет значение NO, проверка будет успешно выполнена, если ни одной строки доступа snmp-server не существует. Эта проверка будет выполнена только в том случае, если маршрутизатор использует набор функций: «K9». Иначе проверка будет пропущена.

```
<check_type: "Cisco">

<item>
type          : RANDOMNESS_CHECK
 description  : "Require Authorized Read SNMP Community Strings and Access
       Control"
 info         : "Verify an authorized community string and access control
       is configured to restrict read access to the device."
 feature_set  : "K9"
 regex        : "snmp-server community ([^ ]*) .*"
 required     : NO
</item>

</check_type>
```

При выполнении этой команды от соответствующей требованиям системы ожидается следующий результат:

```
"Require Authorized Read SNMP Community Strings and Access Control" :
       [PASSED]

Verify an authorized community string and access control is configured to
       restrict read access to the device.
```

Закончившийся ошибкой аудит приведет к следующему результату:

```
"Require Authorized Read SNMP Community Strings and Access Control" :
       [FAILED]

Verify an authorized community string and access control is configured to
       restrict read access to the device.
 - error message:
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
The following configuration line does not contain a token deemed random
      enough:
snmp-server community foobar RO

The following configuration line does not contain a token deemed random
      enough:
snmp-server community public RO
```

В приведенном выше случае было две строки: foobar и public, которые содержат недостаточно произвольные лексемы и, следовательно, не проходят проверку.

Пример 4: контекстная проверка контроля доступа SSH

Ниже приведен простой файл **.audit**, выполняющий поиск всех элементов конфигурации line с помощью ключевого слова **context** и сравнение с **regex**, чтобы убедиться, что контроль доступа SSH установлен.

```
<check_type: "Cisco">

<item>
 type          :CONFIG_CHECK
 description  :"Require SSH Access Control"
 info         :" Verify that management access to the device is restricted
      on all VTY lines."
 context      :"line .*"
 item         :"access-class [0-9]+ in"</item>

</check_type>
```

При выполнении этой команды от соответствующей требованиям системы ожидается следующий результат:

```
"Require SSH Access Control" : [PASSED]

 Verify that management access to the device is restricted on all VTY
      lines.
```

Закончившийся ошибкой аудит приведет к следующему результату:

```
"Require SSH Access Control" : [FAILED]

 Verify that management access to the device is restricted on all VTY
      lines.

 - error message:
The following configuration is set:
line con 0
 exec-timeout 5 0
 no modem enable

Missing configuration: access-class [0-9]+ in
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
The following configuration is set:
line vty 0 4
 exec-timeout 5 0
 password 7 15010A1C142222362D
 transport input ssh

Missing configuration: access-class [0-9]+ in
```

В приведенном выше примере было две строки, соответствующие заданному в поле **context** регулярному выражению **line .***. Так как ни одна строка не включала регулярное выражение **item**, результатом выполнения данного аудита будет сообщение FAILED (неудача).

Условия

В политике аудита Cisco можно определить логику **if/then/else**. Это позволяет конечному пользователю в случае успешного прохождения аудита возвращать предупреждающее сообщение, а не сообщение типа «пройдено» или «неудача».

Синтаксис выполнения условий следующий:

```
<if>
 <condition type: "or">
  <Insert your audit here>
 </condition>
 <then>
  <Insert your audit here>
 </then>
 <else>
  <Insert your audit here>
 </else>
</if>
```

Пример:

```
<if>
<condition type:"AND">
 <item>
  type:CONFIG_CHECK
  description:"Forbid Auxiliary Port"
  info:"Verify the EXEC process is disabled on the auxiliary (aux) port."
  context:"line aux "
  item:"no exec"
 </item>
 <item>
  type:CONFIG_CHECK_NOT
  description:"Forbid Auxiliary Port"
  info:"Verify the EXEC process is disabled on the auxiliary (aux) port."
  context:"line aux "
  item:"transport input [^n][^o]?[^n]?[^e]?$"
 </item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
</condition>
<then>
 <report type:"PASSED">
  description: "Forbid Auxiliary Port"
  info: "Verify the EXEC process is disabled on the auxiliary (aux) port."
 </report>
</then>
<else>
 <report type:"FAILED">
  description: "Forbid Auxiliary Port"
  info: "Verify the EXEC process is disabled on the auxiliary (aux) port."
</report>
</else>
</if>
```

Выполнено условие или нет, в этом отчете никогда не отображается, потому что это «тихая» проверка.

Условия могут быть типа **and** либо **or**.

СПРАВКА ПО ФАЙЛАМ ПРОВЕРКИ СООТВЕТСТВИЯ КОНФИГУРАЦИИ БАЗЫ ДАННЫХ .AUDIT

В этом разделе описывается формат и функции проверок соответствия для баз данных, а также приводится обоснование каждой настройки.

Использование кавычек

Одинарные и двойные кавычки при заключении в них полей аудита являются взаимозаменяемыми, за исключением следующих двух случаев:

1. В проверках соответствия операционной системы Windows, если специальные поля, например CRLF и т. д., должны интерпретироваться буквально, используйте одинарные кавычки. Любые оказавшиеся внутри этих кавычек поля, которые должны интерпретироваться как строки, следует выделять.

Например:

```
expect: 'First line\r\nSecond line\r\nJohn\'s Line'
```

2. Двойные кавычки требуются при использовании полей include_paths и exclude_paths аудита WindowsFiles.

При использовании в поле любого типа (description, value_data, regex и т. д.) строк, содержащих одинарные или двойные кавычки, существует два способа их обработки:

a. В качестве наружных кавычек можно использовать кавычки противоположного типа.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Например:

```
expect: "This is John's Line"
expect: 'We are looking for a double-quote-".*'
```

b. Можно выделять внутренние кавычки с помощью обратной наклонной черты (только двойные кавычки).

Например:

```
expect: "\"Text to be searched\""
```

Тип проверки

Все проверки соответствия базы данных должны заключаться в теги `check_type` и иметь обозначение типа Database. Это необходимо для того, чтобы иметь возможность отличать файлы **`.audit`**, предназначенные специально для баз данных, от аудитов соответствия иного типа. Поле **`check_type`** требует наличия двух дополнительных параметров:

- **`db_type`**
- **`version`**

Возможно проведение аудита баз данных в том числе следующих типов:

- SQLServer
- Oracle
- MySQL
- PostgreSQL
- DB2
- Informix

Параметр **`version`** в настоящее время всегда имеет значение «1».

Пример:

```
<check_type: "Database" db_type:"SQLServer" version:"1">
```

Ключевые слова

В следующей таблице показано, как может быть использовано каждое ключевое слово в проверках соответствия баз данных:

Ключевое слово	Пример использования и поддерживаемые параметры
`type`	`SQL_POLICY`
`description`	Это ключевое слово позволяет добавлять краткое описание выполняемой проверки. Настоятельно

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	рекомендуется обеспечивать уникальность значения поля **description** и отсутствие в разных проверках одинаковых значений в поле описания. Программное обеспечение SecurityCenter компании Tenable использует это поле для автоматического генерирования уникального идентификационного номера подключаемого модуля на основе значения поля **description**. Пример: `description: "DBMS Password Complexity"`
info	Это ключевое слово используется для добавления более подробного описания выполняемой проверки, например, сведений о нормативном акте, URL-адресе, корпоративной политике или иной причине ее необходимости. Возможно добавление нескольких полей **info** на отдельных строках, чтобы обеспечить форматирование текста в виде параграфа. Количество полей **info** не ограничено. Пример: `info: "Checking that \"password complexity\" requirements are enforced for systems using SQL Server authentication."`

sql_request	Это ключевое слово используется для определения фактического запроса **SQL**, передаваемого базе данных. Запрашиваться и возвращаться запросом **SQL** могут массивы данных, для чего требуется запрашиваемые и возвращаемые значения разделять запятыми.
	
	Пример:
	`sql_request: "select name from sys.sql_logins where type = 'S' and is_policy_checked <> '1'"`
	
	Пример:
	`sql_request: "select name, value_in_use from sys.configurations where name = 'clr enabled'"`

sql_types	Это ключевое слово может иметь два значения: **POLICY_VARCHAR** и **POLICY_INTEGER**. Используйте тип **POLICY_INTEGER** для числовых значений от 0 до 2147483647, а тип **POLICY_VARCHAR** для возвращаемых значений любых других типов.
	
	Пример:
	`sql_types: POLICY_VARCHAR`
	
	Пример:
	`sql_types: POLICY_VARCHAR,POLICY_INTEGER`
	
	При наличии нескольких возвращаемых элементов значение в поле **sql_types** должно иметь формат списка разделенных запятыми значений, позволяющего

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	принимать типы данных каждого возвращаемого базой данных **SQL** результата. В приведенном выше примере первое значение, получаемое в результате выполнения запроса **SQL**, имеет тип **varchar**, а второе возвращаемое значение является целым числом.

sql_expect	Это ключевое слово используется для определения ожидаемого возвращаемого по запросу **SQL** значения. Может потребоваться точное значение, включая **NULL** и **0**. Кроме того, для значений типа **POLICY_VARCHAR sql_types** могут требоваться регулярные выражения.
	
	Пример:
	`sql_expect: regex:"^.+Failure" || regex:"^.+ALL"`
	
	Пример:
	`sql_expect: NULL`
	
	Пример:
	`sql_expect: 0 || "0"`
	
	Для возвращаемых целочисленных значений двойные кавычки являются необязательными.
	
	Пример:
	`sql_expect: "clr enabled",0`
	
	По запросу **SQL** может возвращаться массив данных, который включается в поле **sql_expect** в формате разделенного запятыми списка.

Примеры командных строк

В этом разделе приводятся примеры распространенных аудитов, используемых для проведения проверок соответствия баз данных. Двоичный файл командной строки **nasl** используется в качестве средства для быстрого испытания аудитов на ходу. Каждый показанный ниже файл **.audit** можно легко включить в политики сканирования Nessus 4 или SecurityCenter. Однако для быстрой проверки одной системы выполнение тестов в командной строке более эффективно. Команда, которую мы будем каждый раз выполнять из каталога **/opt/nessus/bin**, будет следующей:

```
# ./nasl -t <IP>
      /opt/nessus/lib/nessus/plugins/database_compliance_check.nbin
```

Здесь <IP> — это IP-адрес системы, аудит которой проводится.

В зависимости от типа проверяемой базы данных могут запрашиваться другие параметры, кроме используемого файла аудита. Например, аудиты Oracle запрашивают идентификатор безопасности (SID) базы данных и тип входа Oracle:

```
Which file contains your security policy : oracle.audit
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
login : admin
Password :
Database type: ORACLE(0), SQL Server(1), MySQL(2), DB2(3),
       Informix/DRDA(4), PostgreSQL(5)
type : 0
sid: oracle
Oracle login type: NORMAL (0), SYSOPER (1), SYSDBA (2)
type: 2
```

Правильные параметры входа для базы данных узнавайте у своего администратора баз данных.

Пример 1: поиск реквизитов входа без даты окончания срока действия

Ниже приведен простой файл **.audit**, выполняющий поиск реквизитов входа сервера SQL Server, которые не имеют даты окончания срока действия. При обнаружении таких реквизитов аудит выдает сообщение об ошибке и показывает несоответствующие установленным требованиям реквизиты.

```
<check_type: "Database" db_type:"SQLServer" version:"1">
<group_policy: "Login expiration check">
<custom_item>
 type: SQL_POLICY
 description: "Login expiration check"
 info: "Database logins with no expiration date pose a security threat. "
 sql_request: "select name from sys.sql_logins where type = 'S' and
       is_expiration_checked = 0"
 sql_types: POLICY_VARCHAR
 sql_expect: NULL
</custom_item>
</group_policy>
</check_type>
```

При выполнении этой команды от соответствующей требованиям системы ожидается следующий результат:

```
"Login expiration check": [PASSED]
```

Устанавливаемые требования обычно предполагают обязательность наличия у реквизитов входа базы данных даты окончания их срока действия.

Закончившийся ошибкой аудит приведет к следующему результату:

```
"Login expiration check": [FAILED]

Database logins with no expiration date pose a security threat.

Remote value:

"distributor_admin"

Policy value:
NULL
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Этот результат показывает, что учетная запись distributor_admin не имеет заданного окончания срока действия и требует проверки соответствия политике безопасности системы.

Пример 2: проверка включенного состояния неразрешенной хранимой процедуры

Этот аудит проверяет, включена ли хранимая процедура «SQL Mail XPs». Хранимые за пределами базы данных процедуры могут представлять для некоторых систем угрозу безопасности и часто требуют отключения.

```
<check_type: "Database" db_type:"SQLServer" version:"1">
<group_policy: "Unauthorized stored procedure check">
<custom_item>
 type: SQL_POLICY
 description: "SQL Mail XPs external stored procedure check"
 info: "Checking whether SQL Mail XPs is disabled. "
 sql_request: "select value_in_use from sys.configurations where name =
       'SQL Mail XPs'"
 sql_types: POLICY_INTEGER
 sql_expect: 0
</custom_item>
</group_policy>
</check_type>
```

Приведенная выше проверка возвращает результат успешного выполнения (Passed), если хранимая процедура «SQL Mail XPs» отключена (value_in_use = 0). Иначе возвращаемый результат выполнения будет отрицательным (Failed).

Пример 3: проверка состояния базы данных с результатами смешанного типа sql_types

В некоторых случаях для соответствия запросов базы данных установленным требованиям необходимо, чтобы для результатов, включающих данные разного типа, использовались различные запросы. Приведенный ниже аудит смешивает различные типы данных и показывает, как результат может быть проанализирован.

```
<check_type: "Database" db_type:"SQLServer" version:"1">
<group_policy: "Mixed result type check">
<custom_item>
 type: SQL_POLICY
 description: "Mixed result type check"
 info: "Checking values for the master database."
 sql_request: " select database_id,user_access_desc,is_read_only from
       sys.databases where is_trustworthy_on=0 and name = 'master'"
 sql_types: POLICY_INTEGER,POLICY_VARCHAR,POLICY_INTEGER
 sql_expect: 1,MULTI_USER,0
</custom_item>
</group_policy>
</check_type>
```

Обратите внимание, что значения полей **`sql_request`**, **`sql_types`** и **`sql_expect`** разделены запятыми.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Условия

В политике для баз данных можно определить логику **if/then/else**. Это позволяет конечному пользователю в случае успешного прохождения аудита возвращать предупреждающее сообщение, а не сообщение типа «пройдено» или «неудача».

Синтаксис выполнения условий следующий:

```
<if>
 <condition type: "or">
  <Insert your audit here>
 </condition>
 <then>
  <Insert your audit here>
 </then>
 <else>
  <Insert your audit here>
 </else>
</if>
```

Пример:

```
<if>
 <condition type : "or">
  <custom_item>
   type: SQL_POLICY
   description: "clr enabled option"
   info: "Is CLR enabled?"
   sql_request: "select value_in_use from sys.configurations where name =
       'clr enabled'"
   sql_types: POLICY_INTEGER
   sql_expect: "0"
  </custom_item>
 </condition>

 <then>
  <custom_item>
   type: SQL_POLICY
   description: "clr enabled option"
   info: "CLR is disabled?"
   sql_request: "select value_in_use from sys.configurations where name =
       'clr enabled'"
   sql_types: POLICY_INTEGER
   sql_expect: "0"
  </custom_item>
 </then>

 <else>
  <report type: "WARNING">
   description: "clr enabled option"
   info: "CLR(Command Language Runtime objects) is enabled"
   info: "Check system policy to confirm CLR requirements."
  </report>
 </else>
</if>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Выполнено условие или нет, в этом отчете никогда не отображается, потому что это «тихая» проверка.

Условия могут быть типа **and** либо **or**.

СПРАВКА ПО ФАЙЛАМ ПРОВЕРКИ СООТВЕТСТВИЯ КОНФИГУРАЦИИ ОС UNIX .AUDIT

В этом разделе описываются проверки соответствия встроенных функций ОС Unix, а также приводится обоснование каждой настройки.

Использование кавычек

Одинарные и двойные кавычки при заключении в них полей аудита являются взаимозаменяемыми, за исключением следующих двух случаев:

1. В проверках соответствия операционной системы Windows, если специальные поля, например CRLF и т. д., должны интерпретироваться буквально, используйте одинарные кавычки. Любые оказавшиеся внутри этих кавычек поля, которые должны интерпретироваться как строки, следует выделять.

Например:

```
expect: 'First line\r\nSecond line\r\nJohn\'s Line'
```

2. Двойные кавычки требуются при использовании полей include_paths и exclude_paths аудита WindowsFiles.

При использовании в поле любого типа (description, value_data, regex и т. д.) строк, содержащих одинарные или двойные кавычки, существует два способа их обработки:

a. В качестве наружных кавычек можно использовать кавычки противоположного типа.

Например:

```
expect: "This is John's Line"
expect: 'We are looking for a double-quote-".*'
```

b. Можно выделять внутренние кавычки с помощью обратной наклонной черты (только двойные кавычки).

Например:

```
expect: "\"Text to be searched\""
```

Тип проверки

Все проверки соответствия для ОС Unix должны заключаться в теги **check_type**" и иметь обозначение типа Unix. В приложении A приведен пример проверки соответствия

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

для ОС Unix, начинающийся с параметра **check_type**, имеющего значение «Unix» и заканчивающегося тегом **</check_type>**.

Это необходимо для того, чтобы можно было отличать файлы **.audit**, предназначенные для аудита соответствия ОС Windows (или других платформ).

Ключевые слова

В следующей таблице показано, как может быть использовано каждое ключевое слово в проверках соответствия Unix.

Ключевое слово	Пример использования и поддерживаемые параметры
attr	Это ключевое слово используется в совокупности с проверками типов FILE_CHECK и FILE_CHECK_NOT для проведения аудита атрибутов файла, связанных с каким-либо файлом. Подробные сведения о настройке атрибутов файлов см. на странице справочника man, посвященной chattr(1).
comment	Это поле используется для добавления дополнительной информации, которая не помещается в поле description. Пример: `comment: (CWD - Current working directory)`
description	Это ключевое слово содержит краткое описание выполняемой проверки. Рекомендуется обеспечивать уникальность значения поля **description** и отсутствие в разных проверках одинаковых значений в поле описания. Поставляемое компанией Tenable программное обеспечение SecurityCenter использует это поле для автоматического генерирования уникального идентификационного номера подключаемого модуля на основе значения поля **description**. Пример: `description: "Permission and ownership check for /etc/at.allow"`
dont_echo_cmd	Это ключевое слово используется с аудиторскими проверками соответствия CMD_EXEC (выполнение команд) для ОС Unix и указывает на то, что аудит не должен выводить фактические команды, выполняемые проверкой. Отображаются только результаты выполнения команд. Пример: **dont_echo_cmd: YES**
except	Это ключевое слово используется для исключения определенных пользователей, служб и файлов из проверки.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	Пример:
	`except: "guest"`
	
	Возможна передача нескольких учетных записей пользователей вместе.
	
	Пример:
	`except: "guest" | "guest1" | "guest2"`

expect	Это ключевое слово используется в совокупности **regex**. Оно обеспечивает возможность искать в файлах конкретные значения.
	
	Пример:
	`<custom_item>`
	` system: "Linux"`
	` type: FILE_CONTENT_CHECK`
	` description: "This check reports a problem when the log level setting in the sendmail.cf file is less than the value set in your security policy."`
	` file: "sendmail.cf"`
	` regex: ".*LogLevel=.*"`
	` expect: ".*LogLevel=9"`
	`</custom_item>`

file	Это ключевое слово используется для описания абсолютного и относительного пути к файлу с целью проверки настройки разрешений и принадлежности.
	
	Примеры:
	`file: "/etc/inet/inetd.conf"`
	`file: "~/inetd.conf"`
	
	Значением поля **file** также может быть вся область.
	
	Пример:
	`file: "/var/log/*"`
	
	Эта функция особенно полезна, если необходимо проверить разрешения или содержимое всех файлов, находящихся в определенном каталоге, используя аудиты FILE_CHECK, FILE_CONTENT_CHECK, FILE_CHECK_NOT или FILE_CONTENT_CHECK_NOT.

file_type	Это ключевое слово описывает тип файла, который разыскивается. Ниже приведен перечень поддерживаемых типов файлов. > b — специальный блочный (буферизованный) > c — специальный символьный (не буферизованный) > d — каталог > p — именованный канал (FIFO) > f — обычный файл

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	Пример:
	`file_type: "f"`
	
	Один или несколько типов файлов можно объединять в одной строке.
	
	Пример:
	`file_type: "c|b"`

group	Это ключевое слово используется для указания группы файла; оно всегда используется в сочетании с ключевым словом **file**. Ключевое слово **group** может иметь значение «none», с помощью которого можно найти файлы, не имеющие владельца.
	
	Пример:
	`group: "root"`
	
	Группа также может быть указана с логическим условием OR посредством следующего синтаксиса:
	
	`group : "root || "bin" || "sys"`

ignore	Это ключевое слово указывает проверке на то, что нужно игнорировать указанные файлы при поиске. Это ключевое слово можно использовать с типами проверок FILE_CHECK, FILE_CHECK_NOT, FILE_CONTENT_CHECK и FILE_CONTENT_CHECK_NOT.
	
	Примеры:
	`# ignore single file`
	`ignore : "/root/test/2"`
	
	`# ignore certain files from a directory`
	`ignore : "/root/test/foo*"`
	
	`# ignore all files in a directory`
	`ignore : "/root/test/*"`

info	Это ключевое слово используется для добавления более подробного описания выполняемой проверки, например сведений о нормативном акте, URL-адресе, корпоративной политике или иной причине ее необходимости. Возможно добавление нескольких полей **info** на отдельных строках, чтобы обеспечить форматирование текста в виде параграфа. Количество полей **info** не ограничено. Пример: `info: "ref. CIS_AIX_Benchmark_v1.0.1.pdf ch 1, pg 28-29."`
levels	Это ключевое слово используется в сочетании с CHKCONFIG и применяется для указания уровней

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	выполнения, для которых должна быть запущена какая-либо служба. Все уровни выполнения описываются в одной строке. Например, если служба «sendmail» должна выполняться на уровне 1, 2 и 3, то соответствующее значение параметра **levels** в проверке CHKCONFIG будет следующим: `levels: "123"`
mask	Это ключевое слово противоположно ключевому слову **mode**, позволяющему указать разрешения, которые должны быть **не** доступны определенному пользователю, группе или другому члену. В отличие от ключевого слова **mode**, проверяющего *определенное* значение разрешения, аудиты **mask** действуют шире и проверяют, имеет ли файл или каталог уровень, равный или более защищенный, чем указано с помощью ключевого слова **mask**. (В ситуации, когда аудит **mode** может пропустить файл с разрешением 640, так как оно не совпадает с ожидаемым аудитом значением 644, аудит **mask** обнаружит, что разрешение 640 «более безопасное», и сообщит об успешном выполнении проверки.) Пример: `mask: 022` Эта строка укажет, что допустимы любые разрешения для владельца, но право записи для группы и других участников недопустимо. Значение «7» параметра **mask** означает отсутствие каких-либо разрешений у определенного пользователя, группы или других членов.

md5	Это ключевое слово используется в проверках типа FILE_CHECK и FILE_CHECK_NOT, чтобы убедиться, что значение MD5 для файла фактически соответствует заданному в политике.
	
	Пример:
	`<custom_item>`
	`type: FILE_CHECK`
	`description: "/etc/passwd has the proper md5 set"`
	`required: YES`
	`file: "/etc/passwd"`
	`md5: "ce35dc081fd848763cab2cfd442f8c22"`
	`</custom_item>`

mode	Это ключевое слово описывает совокупность разрешений в отношении рассматриваемого файла или папки. Ключевое слово **mode** может быть представлено в строчном и восьмеричном формате. Примеры: `mode: "-rw-r--r--"`

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	`mode: "644"` `mode: "7644"`
name	Это ключевое слово используется для определения имени процесса в проверке PROCESS_CHECK. Пример: **`name: "syslogd"`**
operator	Это ключевое слово используется в сочетании с RPM_CHECK и PKG_CHECK, чтобы указать условие успешного или не успешного выполнения проверки в зависимости от версии установленного пакета RPM. Оно может принимать следующие значения: > **`lt`** (меньше) > **`lte`** (меньше или равно) > **`gte`** (больше или равно) > **`gt`** (больше) > **`eq`** (равно) Пример: `operator: "lt"`

owner	Это ключевое слово используется для указания владельца файла; оно всегда используется в сочетании с ключевым словом **`file`** (файл). Ключевое слово **`owner`** (владелец) может иметь значение «none», с помощью которого можно найти файлы, не имеющие владельца.
	
	Пример:
	`owner: "root"`
	
	Владение также может быть указано с логическим условием OR посредством следующего синтаксиса:
	
	`owner : "root" || "bin" || "adm"`

regex	Это ключевое слово позволяет выполнять поиск файла, соответствующего определенному регулярному выражению. Пример: `regex: ".*LogLevel=9$"` Следующие специальные символы требуют особого обращения: + \ * ( ) ^ Если требуется, чтобы они интерпретировались буквально, то их следует дважды исключить с помощью двух символов обратной наклонной черты («\\») или заключить в квадратные скобки («[]»). Другие символы,

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	как показанные ниже, для буквальной интерпретации требуют только одного символа обратной наклонной черты: . ? " ' Это определяется порядком обработки этих символов компилятором.
Required	Это ключевое слово позволяет указать, обязательно ли наличие проверяемого элемента на удаленной системе. Например, если параметр **required** имеет значение NO и проверка имеет тип **type** FILE_CHECK, то эта проверка будет успешно выполнена, если файл существует и касающиеся его разрешения соответствуют указанным в файле **.audit**, либо его вообще не существует. С другой стороны, если ключевому слову **required** присвоено значение YES, то приведенная выше проверка пройдена не будет.

Rpm	Это ключевое слово используется для указания пакета RPM, который нужно искать, и используется в сочетании с проверкой RPM_CHECK.
	
	Пример:
	`<custom_item>`
	`type: RPM_CHECK`
	`description: "Make sure that the Linux kernel is BELOW version 2.6.0"`
	`rpm: "kernel-2.6.0-0"`
	`operator: "lt"`
	`required: YES`
	`</custom_item>`

search_locations	Это ключевое слово может быть использовано для указания мест файловой системы, где может выполняться поиск.
	
	Пример:
	`search_locations: "/bin"`
	
	Возможно указание нескольких мест поиска вместе.
	
	Пример:
	`search_locations: "/bin" | "/etc/init.d" | "/etc/rc0.d"`

Service	Это ключевое слово используется в сочетании с ключевыми словами CHKCONFIG, XINETD_SVC и SVC_PROP и предназначено для указания проверяемой службы.
	
	Пример:
	`<custom_item>`
	`type: CHKCONFIG`
	`description: "2.1 Disable Standard Services - Check if cups is disabled"`

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	service: "cups" levels: "123456" status: OFF </custom_item>
Severity	Любая проверка, как **<item>**, так и **<custom_item>**, позволяет добавить флаг **severity** и установить уровень серьезности: низкий (LOW), средний (MEDIUM) или высокий (HIGH). По умолчанию результаты, не соответствующие установленным требованиям, имеют высокий уровень серьезности. Пример: severity: MEDIUM
status	Это ключевое слово используется с проверками PROCESS_CHECK, CHKCONFIG и XINETD_SVC для определения того, должна ли работающая на определенном хосте служба быть запущена или отключена. Ключевое слово **status** может принимать 2 значения: включено (ON) и выключено (OFF). Пример: status: ON status: OFF
system	Это ключевое слово указывает тип системы, в которой следует провести проверку.  Ключевое слово **system** применимо только к проверкам custom_item и не применимо к встроенным проверкам item. Доступными значениями являются возвращаемые командой **uname** в целевой операционной системе. Например, в системе Solaris это будет значение SunOS, в системе Mac OS X — Darwin, в системе FreeBSD — FreeBSD и т. д. Пример: system: "SunOS"
timeout	Это ключевое слово используется в сочетании с CMD_EXEC и указывает в секундах количество времени, в течение которого указанная команда может выполняться, пока время ее выполнения не истечет. Это ключевое слово полезно в тех случаях, когда определенная команда, например команда ОС Unix **find**, требует для завершения работы значительного периода времени. Если это ключевое слово не определено, то по умолчанию в аудитах типа CMD_EXEC устанавливается таймаут пять минут.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

	Пример: `timeout: “600”`

type	`CHKCONFIG`
	`CMD_EXEC`
	`FILE_CHECK`
	`FILE_CHECK_NOT`
	`FILE_CONTENT_CHECK`
	`FILE_CONTENT_CHECK_NOT`
	`GRAMMAR_CHECK`
	`PKG_CHECK`
	`PROCESS_CHECK`
	`RPM_CHECK`
	`SVC_PROP`
	`XINETD_SVC`

value	Ключевое слово **value** полезно для проверки, соответствует ли значение параметра системы предусмотренному политикой. Пример: `value: "90..max"` Ключевое слово **value** может быть равно диапазону [число .. максимум]. Если значение находится в диапазоне от указанного числового значения до максимального значения, то проверка успешно пройдена.

Пользовательские элементы

Пользовательский элемент — это полная проверка, определенная на основе ключевых слов, определения которых приведены ниже. Ниже приведен список пользовательских элементов. Каждая проверка начинается с тега **<custom_item>** и заканчивается тегом **</custom_item>**. В данные теги заключаются списки из одного или нескольких ключевых слов, которые интерпретируются анализатором проверки соответствия для выполнения проверки.

Закрывающие теги **</custom_item>** и **</item>** для пользовательских аудиторских проверок являются взаимозаменяемыми.

CHKCONFIG

Аудиторская проверка CHKCONFIG обеспечивает взаимодействие со служебной программой **chkconfig** проверяемой удаленной системы под управлением ОС Red Hat. Эта проверка состоит из пяти обязательных ключевых слов: **type**, **description**, **service**, **levels** и **status**.

Аудит CHKCONFIG работает только с системами под управлением ОС Red Hat или производными от Red Hat системами, например Fedora.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Пример:

```
<custom_item>
 type: CHKCONFIG
 description: "Make sure that xinetd is disabled"
 service: "xinetd"
 levels: "123456"
 status: OFF
</custom_item>
```

CMD_EXEC

Существует возможность выполнения команд на удаленном хосте и проверки соответствия результата ожидаемому. Такие проверки следует проводить с исключительной осторожностью, так как они не всегда переносимы между различными разновидностями ОС Unix.

Ключевое слово **quiet** указывает сканеру Nessus, что показывать результат выполнения команды, проверка которой дала отрицательный результат, **не** следует. Оно может иметь значения «YES» (да) или «NO» (нет). По умолчанию этот параметр имеет значение «NO», и результат выполнения команд отображается. Аналогично, ключевое слово **dont_echo_cmd** ограничивает результаты, выводя только сами результаты выполнения команды, а не ход ее выполнения.

Ключевое слово **nosudo** позволяет пользователю указать сканеру Nessus, что **не** следует использовать команду sudo для выполнения данной команды, установив для этого ключевого слова значение «YES» (да). По умолчанию этот параметр имеет значение «NO», и команда sudo всегда используется, если в настройках задано ее использование.

Пример:

```
<custom_item>
 type: CMD_EXEC
 description: "Make sure that we are running FreeBSD 4.9 or higher"
 cmd: "uname -a"
 timeout: 7200
 expect: "FreeBSD (4\.(9|[1-9][0-9])|[5-9]\.)"
 dont_echo_cmd : YES
</custom_item>
```

FILE_CHECK

Аудиты соответствия для ОС Unix обычно проверяют существование и настройки определенного файла. Аудит FILE_CHECK использует четыре или больше ключевых слов, чтобы определить эти проверки. Ключевые слова **type**, **description** и **file** являются обязательными, и за ними следует одна или несколько проверок. Текущий синтаксис позволяет проверять владельца, группу и разрешения файла.

В проверках FILE_CHECK можно указывать всю область (например, **/var/log/***). Однако надо помнить, что это касается только файлов, но не каталогов. Если указана вся область, но один или несколько совпадающих файлов должны быть исключены из поиска, то для игнорирования этих файлов нужно использовать ключевое слово **ignore**.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Приведем несколько примеров:

```
<custom_item>
 system: "Linux"
 type: FILE_CHECK
 description: "Permission and ownership check for /etc/default/cron"
 file: "/etc/default/cron"
 owner: "bin"
 group: "bin"
 mode: "-r--r--r--"
</custom_item>
```

```
<custom_item>
 system: "Linux"
 type: FILE_CHECK
 description: "Permission and ownership check for /etc/default/cron"
 file: "/etc/default/cron"
 owner: "bin"
 group: "bin"
 mode: "444"
</custom_item>
```

```
<custom_item>
 system: "Linux"
 type: FILE_CHECK
 description: "Make sure /tmp has its sticky bit set"
 file: "/tmp"
 mode: "1000"
</custom_item>
```

```
<custom_item>
 type: FILE_CHECK
 description: "/etc/passwd has the proper md5 set"
 required: YES
 file: "/etc/passwd"
 md5: "ce35dc081fd848763cab2cfd442f8c22"
</custom_item>
```

```
<custom_item>
 type: FILE_CHECK
 description: "Ignore maillog in the file mode check"
 required: YES
 file: "/var/log/m*"
 mode: "1000"
 ignore: "/var/log/maillog"
</custom_item>
```

FILE_CHECK_NOT

Аудит FILE_CHECK_NOT состоит из трех или более ключевых слов. Ключевые слова **`type`**, **`description`** и **`file`** являются обязательными, и за ними следует одна или

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

несколько проверок. Текущий синтаксис позволяет проверять владельца, группу и разрешения файла. Аналогично аудиту FILE_CHECK ключевое слово **ignore** позволяет пропускать один или несколько файлов, если указана вся область.

Эта функция имеет противоположное функции FILE_CHECK действие. Политика не выполняется, если файла не существует или если его режим соответствует требованиям, определенным в проверке.

В проверках FILE_CHECK_NOT можно указывать всю область (например, **/var/log/***). Однако учтите, что это касается только файлов, но не каталогов.

Приведем несколько примеров:

```
<custom_item>
 type: FILE_CHECK_NOT
 description: "Make sure /bin/bash does NOT belong to root"
 file: "/bin/bash"
 owner: "root"
</custom_item>
```

```
<custom_item>
 type: FILE_CHECK_NOT
 description: "Make sure that /usr/bin/ssh does NOT exist"
 file: "/usr/bin/ssh"
</custom_item>
```

```
<custom_item>
 type: FILE_CHECK_NOT
 description: "Make sure /root is NOT world writeable"
 file: "/root"
 mode: "0777"
</custom_item>
```

FILE_CONTENT_CHECK

Возможна не только проверка существования и параметров файла, но и проведение анализа содержимого текстовых файлов. Для поиска существующего содержимого в одном или нескольких местах можно использовать регулярные выражения. Используйте ключевое слово **ignore**, чтобы пропускать один или несколько файлов в указанных местах поиска.

Приведем несколько примеров:

```
<custom_item>
 system: "Linux"
 type: FILE_CONTENT_CHECK
 description: "This check reports a problem when the log level setting in
       the sendmail.cf file is less than the value set in your security
       policy."
 file: "sendmail.cf"
 regex: ".*LogLevel=.*$"
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 expect: ".*LogLevel=9"
</custom_item>
```

```
<custom_item>
 system: "Linux"
 type: FILE_CONTENT_CHECK
 file: "sendmail.cf"
 search_locations: "/etc:/etc/mail:/usr/local/etc/mail/"
 regex: ".*PrivacyOptions=".*"
 expect: ".*PrivacyOptions=.*,novrfy,.*"
</custom_item>
```

```
<custom_item>
 #System: "Linux"
 type: FILE_CONTENT_CHECK
 description: "FILE_CONTENT_CHECK"
 file: "/root/test2/foo*"
 # ignore single file
 ignore: "/root/test/2"
 # ignore all files in a directory
 ignore: "/root/test/*"
 #ignore certain files from a directory
 ignore: "/root/test/foo*"
 regex: "FOO"
 expect: "FOO1"
</custom_item>
```

FILE_CONTENT_CHECK_NOT

Этот аудит проверяет содержимое файлов на совпадение с регулярным выражением, указанным в поле **`regex`**. Эта функция действует противоположно функции FILE_CONTENT_CHECK. Т. е. политика не выполнена, если регулярное выражение **соответствует** содержимому файла. Используйте ключевое слово **`ignore`**, чтобы пропускать один или несколько файлов в указанных местах поиска.

Приведем пример:

```
<custom_item>
 type: FILE_CONTENT_CHECK_NOT
 description: "Make sure NIS is not enabled on the remote host by making
       sure that  +::' is not in /etc/passwd"
 file: "/etc/passwd"
 regex: "^\+::"
 expect: "^\+::"
</custom_item>
```

GRAMMAR_CHECK

Аудит GRAMMAR_CHECK проверяет содержимое файла и сравнивает с очень свободно определенными грамматическими правилами (состоящими из одного или нескольких регулярных выражений). Если **одна** строка в целевом файле не соответствует какому-либо регулярному выражению, то проверка не пройдена.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Пример:

```
<custom_item>
 type: GRAMMAR_CHECK
 description: "Check /etc/securetty contents are OK."
 file: "/etc/securetty"
 regex: "console"
 regex: "vc/1"
 regex: "vc/2"
 regex: "vc/3"
 regex: "vc/4"
 regex: "vc/5"
 regex: "vc/6"
 regex: "vc/7"
</custom_item>
```

PKG_CHECK

Аудит PKG_CHECK выполняет команду **pkgchk** для системы SunOS. Ключевое слово **pkg** используется для указания пакета, который нужно искать, а ключевое слово **operator** указывает условие прохождения или непрохождения проверки в зависимости от версии установленного пакета.

Примеры:

```
<custom_item>
 system: "SunOS"
 type: PKG_CHECK
 description: "Make sure SUNWcrman is installed"
 pkg: "SUNWcrman"
 required: YES
</custom_item>
```

```
<custom_item>
 system: "SunOS"
 type: PKG_CHECK
 description: "Make sure SUNWcrman is installed and is greater than 9.0.2"
 pkg: "SUNWcrman"
 version: "9.0.2"
 operator: "gt"
 required: YES
</custom_item>
```

PROCESS_CHECK

Как и в случае проверок файлов подлежащая аудиту платформа Unix может быть испытана на выполняющиеся процессы. Реализация данной проверки запускает команду **chkconfig –list** для получения списка выполняющихся процессов.

Примеры:

```
<custom_item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
 system: "Linux"
 type: PROCESS_CHECK
 name: "auditd"
 status: OFF
</custom_item>
```

```
<custom_item>
 system: "Linux"
 type: PROCESS_CHECK
 name: "syslogd"
 status: ON
</custom_item>
```

RPM_CHECK

Аудит RPM_CHECK используется для проверки номеров версий установленных на удаленной системе пакетов RPM. Эта проверка состоит из пяти обязательных ключевых слов: **type**, **description**, **rpm**, **operator** (оператор) и одного необязательного ключевого слова **required**. Ключевое слово **rpm** используется для указания пакета, который нужно искать, а ключевое слово **operator** указывает условие прохождения или не прохождения проверки в зависимости от версии установленного пакета RPM.

Использование проверок RPM не переносимо между различными версиями Linux. Поэтому использование аудита RPM_CHECK не считается переносимым.

Приведем некоторые примеры, предполагая, что пакет **iproute-2.4.7-10** установлен:

```
<custom_item>
 type: RPM_CHECK
 description: "RPM check for iproute-2.4.7-10 - should pass"
 rpm: "iproute-2.4.7-10"
 operator: "gte"
</custom_item>
```

```
<custom_item>
 type: RPM_CHECK
 description: "RPM check for iproute-2.4.7-10 should fail"
 rpm: "iproute-2.4.7-10"
 operator: "lt"
 required: YES
</custom_item>
```

```
<custom_item>
 type: RPM_CHECK
 description: "RPM check for iproute-2.4.7-10 should fail"
 rpm: "iproute-2.4.7-10"
 operator: "gt"
 required: NO
</custom_item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
<custom_item>
 type: RPM_CHECK
 description: "RPM check for iproute-2.4.7-10 should pass"
 rpm: "iproute-2.4.7-10"
 operator: "eq"
 required: NO
</custom_item>
```

SVC_PROP

Аудит SVC_PROP позволяет взаимодействовать со средством **svcprop -p** на системе под управлением ОС Solaris 10. Его можно использовать для запроса свойств, связанных с определенной службой. Ключевое слово **service** используется для указания службы, которая подлежит аудиту. Ключевое слово **property** указывает имя свойства, которое мы хотим запросить. Ключевое слово **value** указывает ожидаемое значение этого свойства. Ожидаемое значение также может быть регулярным выражением.

Примеры:

```
<custom_item>
 type: SVC_PROP
 description: "Check service status"
 service: "cde-ttdbserver:tcp"
 property: "general/enabled"
 value: "false"
</custom_item>
```

```
<custom_item>
 type: SVC_PROP
 description: "Make sure FTP logging is set"
 service: "svc:/network/frp:default"
 property: "inetd_start/exec"
 regex: ".*frpd.*-1"
</custom_item>
```

XINETD_SVC

Аудит XINETD_SVC используется для проверки статуса служб xinetd при запуске. Эта проверка состоит из четырех обязательных ключевых слов: **type**, **description**, **service** и **status**.

Она работает только в системах Red Hat или производных от Red Hat системах, например Fedora.

Пример:

```
<custom_item>
 type: XINETD_SVC
 description: "Make sure that telnet is disabled"
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
  service: "telnet"
  status: OFF
 </custom_item>
```

Встроенные проверки

Проверки, которые не могут быть обеспечены описанными выше проверками, должны быть написаны как пользовательские на языке NASL. Все подобные проверки относятся к категории встроенных. Каждая проверка начинается с тега **<item>** и заканчивается тегом **</item>**. В данные теги заключатся списки из одного или нескольких ключевых слов, которые интерпретируются анализатором проверки соответствия для выполнения проверки. Ниже приведен список доступных проверок.

Ключевое слово **system** недоступно для встроенных проверок и в случае использования приводит к синтаксической ошибке.

Управление паролями

В приведенных ниже примерах <min> (минимум) и <max> (максимум) используются для обозначения целочисленного значения, которое должно использоваться в качестве значения в аудите, а не строки.

В случаях, когда точное минимальное или максимальное значение неизвестны, целочисленные значения заменяются строками «Min» (минимум) или «Max» (максимум).

min_password_length

Использование

```
<item>
 name: "min_password_length"
 description: "This check examines the system configuration for the minimum
password length that the passwd program will accept. The check reports a
problem if the minimum length is less than the length specified in your
policy."
 except: "user1" | "user2" (list of users to be excluded)
 value: "<min>..<max>"
</item>
```

Эта встроенная проверка позволяет убедиться, что на удаленной системе принудительно установлена минимальная длина пароля в диапазоне **<min>..<max>**. Наличие минимальной длины пароля вынуждает пользователей выбирать более сложные пароли.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Операционная система	Реализация
Linux	Минимальная длина пароля определяется как PASS_MIN_LEN в **/etc/login.defs**.
Solaris	Минимальная длина пароля определяется как PASSLENGTH в **/etc/default/passwd**. Обратите внимание, что эта переменная также контролирует максимальную длину пароля.
HP-UX	Минимальная длина пароля определяется как MIN_PASSWORD_LENGTH в **/etc/default/security**.
Mac OS X	Минимальная длина пароля определяется как minChar в локальной политике, определяемой с помощью команды **pwpolicy**.

Пример:

```
 <item>
  name: "min_password_length"
  description: "Make sure that each password has a minimum length of 6
        chars or more"
  value: "6..65535"
 </item>
```

max_password_age

Использование

```
<item>
 name: "max_password_age"
 description: "This check reports agents that have a system default maximum
password age greater than the specified value and agents that do not have a
maximum password age setting."
 except: "user1" | "user2" (list of users to be excluded)
 value: "<min>..<max>"
</item>
```

Эта встроенная функция позволяет убедиться, что максимальный срок действия пароля (например, время, через которое пользователи вынуждены менять свои пароли) установлен в заданных пределах.

Наличие максимального срока действия пароля не позволяет пользователям сохранять один и тот же пароль на протяжении нескольких лет. Смена пароля часто помогает предотвратить использование пароля завладевшим им злоумышленником в течение неограниченного времени.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Операционная система	Реализация
Linux	Переменная PASS_MAX_DAYS определена в **/etc/login.defs**.
Solaris	Переменная MAXWEEKS, расположенная в **/etc/default/passwd**, определяет максимальное количество недель, в течение которого паролем можно пользоваться.
HP-UX	Это значение контролируется переменной PASSWORD_MAXDAYS, расположенной в **/etc/default/security**.
Mac OS X	Чтобы задать это значение, можно воспользоваться параметром maxMinutesUntilChangePassword политики в отношении паролей (устанавливаемым с помощью средства **pwpolicy**).

Пример:

```
 <item>
  name: "max_password_age"
  description: "Make sure a password can not be used for more than 21 days"
  value: "1..21"
 </item>
```

min_password_age

Использование

```
<item>
 name: "min_password_age"
 description: "This check reports agents and users with password history
settings that are less than a specified minimum number of passwords."
 except: "user1" | "user2" (list of users to be excluded)
 value: "<min>..<max>"
</item>
```

Эта встроенная функция позволяет убедиться, что минимальный срок действия пароля (например, время, по истечении которого пользователям разрешено менять свои пароли) установлен в заданных пределах.

Наличие минимального срока действия пароля не позволяет пользователям менять пароли слишком часто, пытаясь преодолеть ограничение максимального размера журнала паролей. Некоторые пользователи пытаются циклически вернуться к своему первоначальному паролю, обходя требования к изменению пароля.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Операционная система	Реализация
Linux	Переменная PASS_MIX_DAYS определена в **/etc/login.defs**.
Solaris	Переменная MINWEEKS, расположенная в **/etc/default/passwd**, определяет минимальное количество недель, в течение которого паролем можно пользоваться.
HP-UX	Это значение контролируется переменной PASSWORD_MINDAYS, расположенной в **/etc/default/security**.
Mac OS X	Этот параметр не поддерживается.

Пример:

```
 <item>7
  name: "min_password_age"
  description: "Make sure a password cannot be changed before 4 days while
        allowing the user to change at least after 21 days"
  value: "4..21"
 </item>
```

Доступ к учетной записи root

root_login_from_console)

Использование

```
<item>
 name: "root_login_from_console"
 description: "This check makes sure that root can only log in from the
system console (not remotely)."
</item>
```

Эта встроенная функция проверяет, что пользователь root может осуществлять непосредственный вход в удаленную систему только с физической консоли.

Эта проверка обосновывается тем, что администраторам настоятельно рекомендуется запрещать использовать учетную запись root напрямую, чтобы можно было отслеживать конкретного человека, осуществлявшего доступ. Вместо этого используется обычная учетная запись пользователя (члена группы wheel в системах под управлением ОС BSD) и затем используется команда «**su**» (или **sudo**) для расширения прав для выполнения административных задач.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Операционная система	Реализация
Linux и HP-UX	Убедитесь, что **/etc/securetty** существует и содержит только «console».
Solaris	Убедитесь, что **/etc/default/login** содержит строку «CONSOLE=/dev/console».
Mac OS X	Эта функция не поддерживается.

Управление разрешениями

accounts_bad_home_permissions

Использование

`<item>`
` name: "accounts bad home permissions"`
` description: "This check reports user accounts that have home directories with incorrect user or group ownerships."`
`</item>`

Эта встроенная функция позволяет убедиться, что начальный каталог каждого не обладающего привилегиями пользователя принадлежит этому пользователю и третьи лица (входящие в ту же группу или группу «everyone») не могут осуществлять запись в нее. Обычно рекомендуется, чтобы начальные каталоги пользователей имели режим 0755 или более жесткий (например, 0700). Эта проверка завершается успешно, если каждый начальный каталог правильно настроен, и возвращает ошибку в противном случае. Для указания желаемого уровня разрешений в отношении начального каталога можно использовать ключевые слова **mode** или **mask**. Ключевое слово **mode** допускает начальные каталоги, точно соответствующие указанному уровню, а ключевое слово **mask** допускает начальные каталоги указанного уровня или более безопасные.

Если посторонние лица будут иметь возможность записывать в начальном каталоге пользователя, то они могут вынудить пользователя исполнить произвольные команды, подделав файлы **~/.profile**, **~/.cshrc**, **~/.bashrc**.
Если пользователем одной группы необходимо совместно использовать файлы, то обычно рекомендуется использовать специально выделенный каталог, в котором будут иметь право записи члены группы, а не использовать начальный каталог какого-либо пользователя.

Для любых неправильно настроенных начальных каталогов следует выполнить команду «**chmod 0755 <user directory>**» и сменить право собственности соответствующим образом.

accounts_without_home_dir

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Использование

```
<item>
 name: "accounts_without_home_dir"
 description: "This check reports user accounts that do not have home
directories."
</item>
```

Эта встроенная функция проверяет наличие у каждого пользователя начального каталога. Проверка проходит успешно, если каждому пользователю присвоен допустимый каталог, и не проходит в ином случае. Обратите внимание, что владение и разрешения при выполнении этой проверки не проверяются.

Обычно рекомендуется всем пользователям системы иметь определенный начальный каталог, потому что некоторым средствам может быть необходимо что-то считывать из него или записывать в него (например, программа **sendmail** проверяет файл **~/.forward**). Если какому-то пользователю не нужно выполнять вход, то вместо этого следует определить несуществующую оболочку shell (например, **/bin/false**). На многих системах пользователи без начального каталога все равно получают привилегии входа в систему, но их фактический начальный каталог — /.

invalid_login_shells

Использование

```
<item>
 name: "invalid_login_shells"
 description: "This check reports user accounts with shells which do not
exist or is not listed in /etc/shells."
</item>
```

Эта встроенная функция позволяет убедиться, что каждый пользователь имеет действительную оболочку shell, определенную в **/etc/shells**.

С помощью файла **/etc/shells** такие приложения, как Sendmail и серверы FTP, определяют, является ли определенная оболочка (shell) действительной для этой системы. Если этот файл не используется программой входа, администраторы могут с его помощью определять, какие оболочки shell действительны в этой системе. Проверка **invalid_login_shells** позволяет установить, все ли пользователи, перечисленные в файле **/etc/passwd**, имеют действительные оболочки shell, определенные в файле **/etc/shells**.

Это позволяет избежать использования неразрешенных методов, например применения **/sbin/passwd** в качестве оболочки shell, чтобы предоставить пользователям возможность менять пароли. Если вы не хотите, чтобы пользователь мог войти в систему, создайте недействительную оболочку shell в файле **/etc/shells** (например, «**/nonexistent**») и установите ее для соответствующих пользователей.

Если у вас есть пользователи без действительной оболочки shell, определите для них какую-либо действительную оболочку shell.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

login_shells_with_suid

Использование

```
<item>
 name: "login_shells_with_suid"
 description: "This check reports user accounts with login shells that have
setuid or setgid privileges."
</item>
```

Эта встроенная функция проверяет, нет ли какой-то оболочки shell с правами «set-uid».

Оболочка shell с правами setuid означает, что при каждом запуске этой оболочки shell сам процесс будет иметь право устанавливать свои разрешения (например, оболочка setuid «root» shell предоставляет любому пользователю права суперпользователя).

Наличие оболочки shell с правами setuid противоречит цели наличия идентификаторов UID и GID и делает управление доступом намного более сложным.

Удалите бит SUID из каждой оболочки shell, обеспечивающей права setuid.

login_shells_writeable

Использование

```
<item>
 name: "login_shells_writeable"
 description: "This check reports user accounts with login shells that have
group or world write permissions."
</item>
```

Эта встроенная функция проверяет, нет ли какой-то оболочки shell, позволяющей выполнять запись всем пользователям или группе.

Если оболочка shell разрешает запись всем пользователям (или всей группе пользователей), то пользователи без привилегий могут заменить ее любой программой. Это позволяет злоумышленникам вынудить других пользователей этой оболочки shell выполнять после выполнения входа произвольные команды.

Убедитесь, что разрешения для каждой оболочки shell установлены надлежащим образом.

login_shells_bad_owner

Использование

```
<item>
 name: "login_shells_bad_owner"
 description: "This check reports user accounts with login shells that are
not owned by root or bin."
</item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Эта встроенная функция позволяет убедиться, что все оболочки shell принадлежат пользователям root и bin.

Что касается оболочек shell с недопустимыми разрешениями, то если пользователю принадлежит оболочка shell, используемая другими пользователями, они могут изменить ее, чтобы вынудить сторонних пользователей исполнять произвольные команды, когда они входят в систему.

Только пользователи root и bin должны иметь возможность изменять двоичные файлы, касающиеся всей системы.

Управление файлом паролей

passwd_file_consistency

Использование

```
<item>
 name: "passwd_file_consistency"
 description: "This check makes sure /etc/passwd is valid."
</item>
```

Эта встроенная функция проверяет, что каждая строка файла **/etc/passwd** имеет допустимый формат (например, семь разделенных двоеточием полей). Если какая-то строка имеет неправильный формат, то выдается соответствующее сообщение и проверка считается не пройденной.

Наличие файла **/etc/passwd** неправильного формата может нарушить работу нескольких средств управления пользователями. Это также может свидетельствовать о взломе или ошибке в пользовательской программе управления пользователями. Кроме того, это может означать, что кто-то пытался добавить пользователя с недопустимым именем (в прошлом часто создавался пользователь с именем «toor:0:0» для получения прав пользователя root).

Если тест показывает несоответствие требованиям, то администратор обязан удалить содержащие нарушения строки из файла **/etc/passwd** или исправить их.

passwd_zero_uid

Использование

```
<item>
 name: "passwd_zero_uid"
 description: "This check makes sure that only ONE account has a uid of 0."
</item>
```

Эта встроенная функция обеспечивает наличие идентификатора UID «0» в файле **/etc/passwd** только у одной учетной записи. Этот идентификатор зарезервирован для учетной записи root, но существует возможность добавить дополнительные учетные записи с идентификатором UID «0», которые будут иметь такой же привилегированный

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

доступ. Эта проверка выполняется успешно, если только одна учетная запись имеет идентификатор UID «0», иначе выдается сообщение об ошибке.

Идентификатор UID «0» предоставляет учетной записи root особые права в системе. Пользователь root может выполнять в системе все, что захочет, в том числе, обычно, просматривать память других процессов (или ядра), считывать и записывать любые файлы системы и т. д. Поскольку эта учетная запись наделена такими мощными возможностями, ее использование необходимо ограничивать до минимума и хорошо защищать.

Администраторам настоятельно рекомендуется обеспечивать уникальность каждого идентификатора UID (буква U в аббревиатуре UID означает: «уникальный» (англ. Unique)). Наличие двух (и более) учетных записей с правами учетной записи root сводит на нет контролируемость системы администратором. Кроме того, многие системы ограничивают возможность непосредственного входа под именем root только консолью, чтобы административное использование можно было отслеживать. Обычно системные администраторы должны сначала войти в систему с помощью своей личной учетной записи, а затем с помощью команды **su** стать пользователем root. Наличие дополнительной учетной записи с идентификатором UID «0» позволяет обойти это ограничение.

Если доступ с правами root требуется нескольким пользователям, то вместо этого следует применять такое средство, как **sudo** или **calife** (либо RBAC в ОС Solaris). Идентификатор UID «0» должен быть только у одной учетной записи.

passwd_duplicate_uid

Использование

```
<item>
 name: "passwd_duplicate_uid"
 description: "This check makes sure that every UID in /etc/passwd is
unique."
</item>
```

Эта встроенная функция позволяет проверить наличие у каждой перечисленной в файле **/etc/passwd** учетной записи уникального идентификатора UID. Этот тест проходит успешно, если каждый идентификатор UID уникален, иначе выдается сообщение об ошибке.

Каждый пользователь в системе Unix идентифицируется по идентификатору пользователя (UID) — числу от 0 до 65535. Если два пользователя пользуются одинаковым идентификатором UID, то они не только получают одинаковые права, но система также считает их одним лицом. Это сводит на нет какую-либо ответственность, поскольку становится невозможно определить, какие действия были выполнены каждым пользователем (обычно система выполняет обратный просмотр UID и использует при отображении журналов первое имя учетных записей, имеющих общий UID).

Стандарты безопасности, например стандарт CIS, запрещают использование одного идентификатора UID несколькими пользователями. Если пользователям необходимо совместно использовать файлы, то вместо этого следует воспользоваться группами.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Присваивайте каждому пользователю системы уникальный идентификатор.

passwd_duplicate_gid

Использование

```
<item>
 name: "passwd_duplicate_gid"
 description: "This check makes sure that every GID in /etc/passwd is
unique."
</item>
```

Эта встроенная функция позволяет проверить уникальность основного идентификатора группы (GID) каждого пользователя. Тест проходит успешно, если каждый пользователь имеет уникальный идентификатор GID, иначе выдается сообщение об ошибке.

Согласно стандартам безопасности рекомендуется создавать одну группу для каждого пользователя (обычно с таким же именем, что и имя пользователя). В этом случае созданные пользователем файлы обычно являются «защищенными по умолчанию», так как принадлежат его основной группе и, следовательно, могут изменяться только им самим. Если пользователь хочет, чтобы файл принадлежал другим членам группы, то он должен явным образом использовать команду **chgrp** для изменения права владения.

Еще одним преимуществом этого подхода является объединение управления членством в группах в один файл (**/etc/group**), а не одновременно в файлы **/etc/passwd** и **/etc/group**.

Для каждого пользователя создавайте группу с таким же именем. Управляйте правами владения группами только посредством файла **/etc/group**.

passwd_duplicate_username

Использование

```
<item>
 name: "passwd_duplicate_username"
 description: "This check makes sure that every username in /etc/passwd is
unique."
</item>
```

Эта встроенная функция позволяет проверить уникальность каждого имени пользователя в файле **/etc/passwd**. Проверка проходит успешно, если это так, иначе выдается сообщение об ошибке.

Повторяющиеся имена пользователей в файле **/etc/passwd** создают проблемы, так как неясно, права какой учетной записи следует использовать.

Команда **adduser** не позволяет создавать повторяющиеся имена пользователей. Такая настройка обычно означает, что создана какая-то угроза системе, используемые для

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

управления пользователями средства содержат ошибки или файл **/etc/passwd** редактировался вручную.

Удалите повторяющиеся имена пользователей или измените их на другие.

passwd_duplicate_home

Использование

```
<item>
 name: "passwd_duplicate_home"
 description: "(arbitrary user comment)"
</item>
```

Эта встроенная функция позволяет проверить уникальность начального каталога каждого несистемного пользователя (т. е. пользователей с идентификаторами UID больше 100) в файле **/etc/passwd**.

Каждое имя пользователя в файле **/etc/passwd** должно иметь уникальный начальный каталог. Если пользователи пользуются общим начальным каталогом, то один из них может вынудить другого выполнить произвольные команды, изменив файлы запуска (**.profile** и т. д.) или поместив мошеннические двоичные файлы в сам начальный каталог. Кроме того, общий начальный каталог сводит на нет ответственность пользователя.

Для обеспечения соответствия требованиям необходимо, чтобы каждый пользователь имел уникальный начальный каталог.

passwd_shadowed

Использование

```
<item>
 name: "passwd_shadowed"
 description: "(arbitrary user comment)"
</item>
```

Эта встроенная проверка позволяет убедиться, что все пароли, хранящиеся в файле **/etc/passwd**, являются «теневыми» (т. е. фактически хранятся в другом файле).

Так как файл **/etc/passwd** является общедоступным для чтения, хранение в нем хеш-кодов паролей позволяет каждому, кто имеет к нему доступ, запустить программу для взлома паролей. Попытки угадать пароль пользователя методом прямого перебора (многократные попытки входа путем ввода различных паролей) обычно обнаруживаются в файлах журналов системы. Если файл **/etc/passwd** содержит хеш-коды паролей, то этот файл можно скопировать и в автономном режиме использовать в качестве исходного материала для программы взлома паролей. Это позволяет совершающему атаку лицу получить пароли пользователей и не быть обнаруженным.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Большинство современных систем Unix используют теневые файлы паролей. Чтобы узнать, как включить хранение паролей в теневом файле в своей системе, обращайтесь к документации к системе.

passwd_invalid_gid

Использование

```
<item>
 name: "passwd_invalid_gid"
 description: "This check makes sure that every GID defined in /etc/passwd
exists in /etc/group."
</item>
```

Эта встроенная функция проверяет, все ли идентификаторы групп (GID), включенные в файл **/etc/passwd**, существуют в файле **/etc/group**. Если каждый идентификатор GID надлежащим образом определен, то проверка проходит успешно, иначе выдается сообщение об ошибке.

При каждом определении идентификатора группы в файле **/etc/passwd** он должен немедленно вноситься в файл **/etc/group**. Иначе система будет находиться в непоследовательном состоянии и возможно возникновение проблем.

Представьте следующий сценарий: пользователь bob имеет идентификатор UID 1000 и идентификатор GID 4000. Идентификатор GID не определен в файле **/etc/group**, что означает, что основная группа пользователя не предоставляет ему каких-либо прав сегодня. Через несколько месяцев системный администратор редактирует файл **/etc/group** и добавляет группу admin, выбрав для нее «неиспользуемый» идентификатор GID 4000. Теперь пользователь bob по умолчанию входит в группу admin, хотя это и не было предусмотрено.

Отредактируйте файл **/etc/group**, чтобы добавить в него недостающие идентификаторы GID.

Управление файлами групп

group_file_consistency

Использование

```
<item>
 name: "group_file_consistency"
 description: "This check makes sure /etc/group is valid."
</item>
```

Эта встроенная функция проверяет, что каждая строка файла **/etc/group** имеет допустимый формат (например, три разделенных двоеточием элемента и список пользователей). Если какая-то строка имеет неправильный формат, то выдается соответствующее сообщение и проверка считается не пройденной.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Наличие файла **/etc/group** неправильного формата может нарушить работу нескольких средств управления пользователями. Это также может свидетельствовать о взломе или ошибке в пользовательской программе управления пользователями. Кроме того, это может указывать на попытку добавить пользователя с недопустимым именем группы.

Отредактируйте файл **/etc/group**, чтобы исправить неправильно сформированные строки.

group_zero_gid

Использование

```
<item>
 name: "group_zero_gid"
 description: "This check makes sure that only ONE group has a gid of 0."
</item>
```

Эта встроенная функция позволяет убедиться, что только одна группа имеет идентификатор группы (GID) «0». Если только одна группа имеет идентификатор GID «0», то проверка проходит успешно, иначе выдается сообщение об ошибке.

Идентификатор GID «0» означает, что пользователи, входящие в эту группу, также являются членами основной группы пользователя root. Это обеспечивает им права пользователя root в отношении любых файлов, для которых установлены разрешения для группы root.

При желании определить группу администраторов создайте вместо этого группу admin.

group_duplicate_name

Использование

```
<item>
 name: "group_duplicate_name"
 description: "This check makes sure that every group name in /etc/group is
unique."
</item>
```

Эта встроенная проверка позволяет проверить уникальность каждого имени группы. Проверка проходит успешно, если это так, иначе выдается сообщение об ошибке.

Повторяющиеся имена групп в файле **/etc/group** создают проблемы, так как неясно, права какой группы следует использовать. Это означает, что наличие повторяющегося имени группы может привести в первую очередь к наличию членов или привилегий у членов, которых у них не должно было быть.

Удалите или переименуйте повторяющиеся имена групп.

group_duplicate_gid

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Использование

```
<item>
 name: "group_duplicate_gid"
 description: "(arbitrary user comment)"
</item>
```

Каждая группа в системе Unix идентифицируется по идентификатору группы (GID) — числу от 0 до 65535. Если две группы пользуются одинаковым идентификатором GID, то они не только получают одинаковые права, но система также считает их одной группой. Это сводит на нет цель использования групп, заключающуюся в разделении прав пользователей.

Стандарты безопасности запрещают использование одного идентификатора GID несколькими группами. Если двум группам требуются одинаковые права, то в них должны входить одинаковые пользователи.

Удалите повторяющиеся группы или назначьте одной из них новый уникальный идентификатор GID.

group_duplicate_members

Использование

```
<item>
 name: "group_duplicate_members"
 description: "This check makes sure that every member of a group is listed
once."
</item>
```

Эта встроенная функция позволяет убедиться, что каждый член группы включен в нее только один раз. Если каждый пользователь уникален, то проверка проходит успешно, иначе выдается сообщение об ошибке.

Каждый член группы должен быть включен в список только один раз. Хотя включение в список несколько раз не создает проблем для лежащей в основе операционной системы, это усложняет жизнь системного администратора, так как отмена предоставленных прав становится более сложной. Например, если в группу admin входят члены alice, bob, charles, daniel, bob, то при отмене предоставленных прав пользователя bob его необходимо удалять дважды.

Убедитесь, что каждый член группы включен в нее только один раз.

group_nonexistant_users

Использование

```
<item>
 name: "group_nonexistant_users"
 description: "This check makes sure that every member of a group actually
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
exists."
</item>
```

Эта проверка позволяет убедиться, что каждый член группы фактически существует в файле **/etc/passwd**.

Наличие несуществующих пользователей в файле **/etc/group** предполагает несовершенную методику администрирования. Пользователя не существует, потому что его имя было введено с ошибкой или потому что он не был удален из группы при удалении из системы.

Не рекомендуется иметь пользователей-приведений в файле **/etc/group**. Если позднее будет добавлен пользователь с таким же именем, он может получить права группы, которые ему не должны были предоставляться.

Удалите несуществующих пользователей из файла **/etc/group**.

Среда пользователя root

dot_in_root_path_variable

Использование

```
<item>
 name: "dot_in_root_path_variable"
 description: "This check makes sure that root's $PATH variable does not
contain any relative path."
</item>
```

Эта проверка позволяет убедиться в том, что текущий рабочий каталог («.») не включен в пути для исполняемых файлов пользователя root. Это позволит избежать повышения прав злоумышленника до супер-пользователя путем вынуждения администратора, работающего под учетной записью root, запустить программу-троян, которая может быть установлена в текущем рабочем каталоге.

writeable_dirs_in_root_path_variable

Использование

```
<item>
 name: "writeable_dirs_in_root_path_variable"
 description: "This check makes sure that root's $PATH variable does not
contain any writeable directory."
</item>
```

Эта проверка сообщает все каталоги, разрешающе запись всем пользователям или всей группе, в переменной PATH пользователей root. Все каталоги, включенные в список этой проверкой, следует внимательно проверить и удалить те права записи для всех пользователей или всех пользователей группы в отношении каталогов, в которых нет необходимости, следующим образом:

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
# chmod go-w path/to/directory
```

Разрешения в отношении файлов

find_orphan_files

Использование

```
<item>
 name: "find_orphan_files"
 description: "This check finds all the files which are 'orphaned' (ie:
whose owner is an invalid UID or GID)."
 # Globs allowed (? and *)
 (optional) basedir: "<directory>"
 (optional) ignore: "<directory>"
 (optional) dir: "<directory>"
</item>
```

Эта проверка сообщает обо всех файлах в системе, которые не имеют владельца.

По умолчанию поиск выполняется рекурсивно из каталога «/». Это может привести к тому, что проверка будет выполняться крайне медленно в зависимости от количества файлов на удаленной системе. Однако при необходимости установленный по умолчанию начальный каталог поиска может быть изменен с помощью необязательного ключевого слова **basedir**. Также можно при поиске пропускать определенные файлы в базовом каталоге с помощью еще одного необязательного ключевого слова **ignore**. При выполнении поиска по файловой системе по умолчанию пропускаются все каталоги, смонтированные в системе NFS, если только они не были отмечены с помощью необязательного ключевого слова **dir**.

Эта проверка, как правило, длится несколько часов в зависимости от типа сканируемой системы. По умолчанию значение таймаута, по истечении которого сканер Nessus перестанет обрабатывать результаты этой проверки, установлено равным пяти часам, и изменить это значение невозможно.

Пример:

```
 <item>
  name: "find_orphan_files"
  description: "This check finds all the files which are 'orphaned' (ie:
        whose owner is an invalid UID or GID)."
  # Globs allowed (? and *)
  basedir: "/tmp"
  ignore: "/tmp/foo"
  ignore: "/tmp/b*"
 </item>
```

find_world_writeable_files

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Использование

```
<item>
 name: "find_world_writeable_files"
 description: "This check finds all the files which are world writeable and
whose sticky bit is not set."
 # Globs allowed (? and *)
 (optional) basedir: "<directory>"
 (optional) ignore: "<directory>"
 (optional) dir: "<directory>"
</item>
```

Эта проверка сообщает обо всех файлах на удаленной системе, запись в которые разрешена всем пользователям. В идеале, на удаленной системе не должно быть файлов, запись в которые разрешена всем пользователям, и эта проверка должна давать пустой результат. Однако в некоторых случаях могут потребоваться файлы, запись в которые разрешена всем пользователям. Все получаемые в результате этой проверки файлы необходимо внимательно проверить, и те из них, которые не требуют наличия атрибутов возможности записи для всех пользователей, удалить следующим образом:

```
# chmod o-w world_writeable_file
```

По умолчанию поиск выполняется рекурсивно из каталога «/». Это может привести к тому, что проверка будет выполняться крайне медленно в зависимости от количества файлов на удаленной системе. Однако при необходимости заданный по умолчанию начальный каталог поиска может быть изменен с помощью необязательного ключевого слова **basedir**. Также можно при поиске пропускать определенные файлы в базовом каталоге с помощью еще одного необязательного ключевого слова **ignore**. При выполнении поиска по файловой системе по умолчанию пропускаются все каталоги, смонтированные в системе NFS, если только они не были отмечены с помощью необязательного ключевого слова **dir**.

Эта проверка, как правило, длится несколько часов в зависимости от типа сканируемой системы. По умолчанию значение таймаута, по истечении которого сканер Nessus перестанет обрабатывать результаты этой проверки, установлено равным пяти часам, и изменить это значение невозможно.

Пример:

```
<item>
 name: "find_world_writeable_files"
 description: "Search for world-writable files"
 # Globs allowed (? and *)
 basedir: "/tmp"
 ignore: "/tmp/foo"
 ignore: "/tmp/bar"
</item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

find_world_writeable_directories

Использование

```
<item>
 name: "find_world_writeable_directories"
 description: "This check finds all the directories which are world
writeable and whose sticky bit is not set."
 # Globs allowed (? and *)
 (optional) basedir: "<directory>"
 (optional) ignore: "<directory>"
 (optional) dir: "<directory>"
</item>
```

Эта проверка сообщает все каталоги удаленной системы, запись в которые разрешена всем пользователям, и бит закрепления которых не установлен. Проверка того, что бит закрепления установлен для всех каталогов, позволяющих запись всем пользователям, обеспечивает возможность удаления файлов только их владельцам. Это позволяет избежать случайного или намеренного удаления какого-то файла другим пользователем.

По умолчанию поиск выполняется рекурсивно из каталога «/». Это может привести к тому, что проверка будет выполняться крайне медленно в зависимости от количества файлов на удаленной системе. Однако при необходимости заданный по умолчанию начальный каталог поиска может быть изменен с помощью необязательного ключевого слова **basedir**. Также можно при поиске пропускать определенные файлы в базовом каталоге с помощью еще одного необязательного ключевого слова **ignore**. При выполнении поиска по файловой системе по умолчанию пропускаются все каталоги, смонтированные в системе NFS, если только они не были отмечены с помощью необязательного ключевого слова **dir**.

Эта проверка, как правило, длится несколько часов в зависимости от типа сканируемой системы. По умолчанию значение таймаута, по истечении которого сканер Nessus перестанет обрабатывать результаты этой проверки, установлено равным пяти часам, и изменить это значение невозможно.

Пример:

```
 <item>
  name: "find_world_writeable_directories"
  description: "This check finds all the directories which are world
        writeable and whose sticky bit is not set."
  # Globs allowed (? and *)
  basedir: "/tmp"
  ignore: "/tmp/foo"
  ignore: "/tmp/b*"
 </item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

find_suid_sgid_files

Использование

```
<item>
 name: "find_suid_sgid_files"
 description: "This check finds all the files which have their SUID or SGID
bit set."
 # Globs allowed (? and *)
 (optional) basedir: "<directory>"
 (optional) ignore: "<directory>"
 (optional) dir: "<directory>"
</item>
```

Эта проверка сообщает обо всех файлах, у которых установлен бит SUID/SGID. Все включенные этой проверкой в отчет файлы следует внимательно проверить, особенно скрипты командного процессора и исполняемые файлы собственной разработки, например исполняемые файлы, которые не поставлялись с системой. Файлы с установленным битом SUID/SGID создают угрозу повышения прав обычного пользователя до прав владельца этого файла или соответствующей группы. Если в существовании таких файлов или скриптов есть необходимость, то их следует специально обследовать, не позволяют ли они создавать файл с повышенными правами.

По умолчанию поиск выполняется рекурсивно из каталога «/». Это может привести к тому, что проверка будет выполняться крайне медленно в зависимости от количества файлов на удаленной системе. Однако при необходимости заданный по умолчанию начальный каталог поиска может быть изменен с помощью необязательного ключевого слова **basedir**. Также можно при поиске пропускать определенные файлы в базовом каталоге с помощью еще одного необязательного ключевого слова **ignore**. При выполнении поиска по файловой системе по умолчанию пропускаются все каталоги, смонтированные в системе NFS, если только они не были отмечены с помощью необязательного ключевого слова **dir**.

Эта проверка, как правило, длится несколько часов в зависимости от типа сканируемой системы. По умолчанию значение таймаута, по истечении которого сканер Nessus перестанет обрабатывать результаты этой проверки, установлено равным пяти часам, и изменить это значение невозможно.

Пример:

```
 <item>
  name: "find_suid_sgid_files"
  description: "Search for SUID/SGID files"
  # Globs allowed (? and *)
  basedir: "/"
  ignore: "/usr/sbin/ping"
 </item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Подозрительное содержимое файлов

admin_accounts_in_ftpusers

Использование

```
<item>
 name: "admin_accounts_in_ftpusers"
 description: "This check makes sure every account whose UID is below 500 is
present in /etc/ftpusers."
</item>
```

Эта проверка позволяет установить, все ли административные учетные записи, т. е. пользователи с идентификаторами UID меньше 500, включены в **/etc/ftpusers**, **/etc/ftpd/ftpusers** или **/etc/vsftpd.ftpusers**.

Необязательные файлы

find_pre-CIS_files

Использование

```
<item>
 name: "find_preCIS_files"
 description: "Find and list all files created by CIS backup script."
 # Globs allowed (? and *)
 (optional) basedir: "<directory>"
 (optional) ignore: "<directory>"
</item>
```

Эта проверка специально предназначена для обеспечения соблюдения определенного требования центра интернет-безопасности (CIS) для прохождения сертификации соответствия стандарту Red Hat CIS. Эта проверка, в частности, полезна для тех, кто настроил и повысил защищенность системы Red Hat в соответствии со стандартом центра CIS корпорации Red Hat. Средство стандартизации центра CIS включает скрипт резервного копирования, который выполняет резервное копирование всех системных файлов, которые могут быть изменены в процессе повышения защищенности системы, и этим файлам добавляется суффикс «**-preCIS**». После успешного применения всех рекомендаций стандарта и восстановления рабочего состояния системы эти файлы следует удалить. Эта проверка позволяет убедиться, что на удаленной системе не осталось файлов «**preCIS**».

По умолчанию поиск выполняется рекурсивно из каталога «/». Это может привести к тому, что проверка будет выполняться крайне медленно в зависимости от количества файлов на удаленной системе. Однако при необходимости заданный по умолчанию начальный каталог поиска может быть изменен с помощью необязательного ключевого слова **basedir**. Также можно при поиске пропускать определенные файлы в базовом каталоге с помощью еще одного необязательного ключевого слова **ignore**.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

Эта проверка, как правило, длится несколько часов в зависимости от типа сканируемой системы. По умолчанию значение таймаута, по истечении которого сканер Nessus перестанет обрабатывать результаты этой проверки, установлено равным пяти часам, и изменить это значение невозможно.

Условия

Существует возможность определить логику **if/then/else** в политике для ОС Unix. Это позволяет конечному пользователю использовать один файл, позволяющий обрабатывать различные конфигурации. Например, один файл политики может проверять настройки программ Postfix и Sendmail благодаря надлежащему синтаксису логики **if/then/else**.

Синтаксис выполнения условий следующий:

```
<if>
 <condition type: "or">
  <Insert your audit here>
 </condition>
 <then>
  <Insert your audit here>
 </then>
 <else>
  <Insert your audit here>
 </else>
</if>
```

Пример:

```
<if>
 <condition type: "or">
  <custom_item>
   type: FILE_CHECK
   description: "Make sure /etc/passwd contains root"
   file: "/etc/passwd"
   owner: "root"
  </custom_item>
 </condition>

 <then>
  <custom_item>
   type: FILE_CONTENT_CHECK
   description: "Make sure /etc/passwd contains root (then)"
   file: "/etc/passwd"
   regex: "^root"
   expect: "^root"
  </custom_item>
 </then>

 <else>
  <custom_item>
   type: FILE_CONTENT_CHECK
   description: "Make sure /etc/passwd contains root (else)"
   file: "/etc/passwd"
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
    regex: "^root"
    expect: "^root"
   </custom_item>
  </else>
 </if>
```

Выполнено условие или нет, в этом отчете никогда не отображается, потому что это «тихая» проверка.

Условия могут быть типа **and** (и) либо **or** (или).

ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ

Компания Tenable подготовила различные документы, содержащие подробные сведения об установке, развертывании, настройке, пользовательской эксплуатации и общем тестировании сканера Nessus:

- **Руководство по установке Nessus** — пошаговое руководство по установке.
- **Руководство пользователя Nessus** — настройка и работа с интерфейсом пользователя Nessus.
- **Проверки Nessus с использованием учетных данных для Unix и Windows —** сведения о порядке выполнения сканирования сетей с проверкой подлинности при помощи сканера уязвимостей Nessus.
- **Проверки соответствия Nessus** — руководство высокого уровня для понимания и выполнения проверок соответствия с помощью сканера Nessus и консоли SecurityCenter.
- **Формат файлов Nessus v2** — содержит описание структуры формата файлов **.nessus** который был введен с версиями Nessus 3.2 и NessusClient 3.2.
- **Спецификация протокола Nessus XML-RPC** — содержит описание протокола XML-RPC и интерфейса в Nessus.
- **Контроль соответствия в режиме реального времени** — содержит обзор того, как решения компании Tenable могут использоваться для обеспечения выполнения разных типов государственных и финансовых норм.

Без колебаний пишите нам по адресам электронной почты support@tenable.com, sales@tenable.com или посетите наш веб-сайт по адресу http://www.tenable.com/.

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

О КОМПАНИИ TENABLE NETWORK SECURITY

Компания Tenable Network Security, ведущая компания в области комплексного мониторинга безопасности, является разработчиком сканера уязвимостей Nessus, а также создателем решения корпоративного класса, не требующего агентов, для непрерывного мониторинга уязвимостей, слабых мест конфигураций, утечек данных, управления журналами и обнаружения взломов с целью обеспечения безопасности сетей и соответствия требованиям FDCC, FISMA, SANS CAG и PCI. Продукты компании Tenable, заслужившие различные награды, используются организациями из списка Global 2000 и государственными учреждениями для упреждающего понижения связанных с сетями рисков до минимума. Дополнительные сведения см. на веб-сайте http://www.tenable.com/.

Tenable Network Security, Inc.
7063 Columbia Gateway Drive
Suite 100
Columbia, MD 21046
410.872.0555
www.tenable.com

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

ПРИЛОЖЕНИЕ A. ПРИМЕР ФАЙЛА ПРОВЕРКИ СООТВЕТСТВИЯ ДЛЯ ОС UNIX

Примечание. Следующий файл (**tenable_unix_compliance_template.audit**) можно получить через портал поддержки Tenable Support Portal, расположенный по адресу https://support.tenable.com/support-center/. В этом файле перечислены проверки соответствия для ОС Unix различного типа, которые могут быть выполнены с помощью модуля проверки соответствия для ОС Unix компании Tenable. В фактический файл могут быть внесены изменения, не отраженные в этом документе.

```
#
# (C) 2008-2010 Tenable Network Security, Inc.
#
# This script is released under the Tenable Subscription License and
# may not be used from within scripts released under another license
# without authorization from Tenable Network Security, Inc.
#
# See the following licenses for details:
#
# http://cgi.tenablesecurity.com/Nessus_3_SLA_and_Subscription_Agreement.pdf
# http://cgi.tenablesecurity.com/Subscription_Agreement.pdf
#
# @PROFESSIONALFEED@
#
# $Revision: 1.11 $
# $Date: 2010/11/04 15:54:36 $
#
# NAME            : Cert UNIX Security Checklist v2.0
#
#
# Description     : This file is used to demonstrate the wide range of
#             checks that can be performed using Tenable's Unix
#             compliance module. It consists of all the currently
#             implemented built-in checks along with examples of all
#             the other Customizable checks. See:
#             https://plugins-customers.nessus.org/support-
center/nessus_compliance_checks.pdf
#                 For more information.
#
#
###################################
#                         #
# File permission related checks  #
#                         #
###################################


<check_type:"Unix">

# Example 1.
# File check example with owner and group
# fields set and mode field set in Numeric
# format

<custom_item>
      #system           : "Linux"
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
      type              : FILE_CHECK
      description       : "Permission and ownership check /etc/inetd.conf"
      info        : "Checking that /etc/inetd.conf has owner/group of root
and is mode '600'"
      file              : "/etc/inetd.conf"
      owner             : "root"
      group             : "root"
      mode              : "600"
</custom_item>


# Example 2.
# File check example with just owner field set
# and mode set.

<custom_item>
      #system           : "Linux"
      type              : FILE_CHECK
      description       : "Permission and ownership check /etc/hosts.equiv"
      info        : "Checking that /etc/hosts.equiv is owned by root and mode
'500'"
      file              : "/etc/hosts.equiv"
      owner             : "root"
      mode              : "-r-x------"
</custom_item>

# Example 3.
# File check example with just file field set
# starting with "~". This check will search
# and audit the file ".rhosts" in home directories
# of all accounts listed in /etc/passwd.

<custom_item>
      #system           : "Linux"
      type              : FILE_CHECK
      description       : "Permission and ownership check ~/.rhosts"
      info        : "Checking that .rhosts in home directories have the
specified ownership/mode"
      file              : "~/.rhosts"
      owner             : "root"
      mode              : "600"
</custom_item>


# Example 4.
# File check example with mode field having
# sticky bit set. Notice the first integer in
# the mode field 1 indicates that sticky bit is
# set. The first integer can be modified to check
# for SUID and SGUID fields. Use the table below
# to determine the first integer field.
#
# 0   000   setuid, setgid, sticky bits are cleared
# 1   001   sticky bit is set
# 2   010   setgid bit is set
# 3   011   setgid and sticky bits are set
# 4   100   setuid bit is set
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
# 5   101   setuid and sticky bits are set
# 6   110   setuid and setgid bits are set
# 7   111   setuid, setgid, sticky bits are set

<custom_item>
      #system           : "Linux"
      type              : FILE_CHECK
      description       : "Permission and ownership check /var/tmp"
      info        : "Checking that /var/tmp is owned by root and mode '1777'"
      file              : "/var/tmp"
      owner       : "root"
      mode        : "1777"
</custom_item>

# Example 5.
# File check example with mode field having
# sticky bit set in textual form and is owned by root.

<custom_item>
      #system           : "Linux"
      type              : FILE_CHECK
      description       : "Permission and ownership check /tmp"
      info        : "Checking that the /tmp mode has the sticky bit set in
textual form and is owned by root"
      file              : "/tmp"
      owner       : "root"
      mode        : "-rwxrwxrwt"
</custom_item>

###################################
#                                 #
# Service/Process related checks  #
#                                 #
###################################

# Example 6.
# Process check to audit if fingerd is turned
# OFF on a given host.

<custom_item>
      #system           : "Linux"
      type        : PROCESS_CHECK
      description       : "Check fingerd process status"
      info        : "This check looks for the finger daemon to be 'OFF'"
      name        : "fingerd"
      status            : OFF
</custom_item>

# Example 7.
# Process check to audit if sshd is turned
# ON on a given host.

<custom_item>
      #system           : "Linux"
      type        : PROCESS_CHECK
      description       : "Check sshd process status"
      info        : "This check looks for the ssh daemon to be 'ON'"
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
      name        : "sshd"
      status            : ON
</custom_item>

##############################
#                            #
# File Content related checks #
#                            #
##############################

# Example 8
# File content check to audit if file /etc/host.conf
# contains the string described in the regex field.
#

<custom_item>
      #System            : "Linux"
      type         : FILE_CONTENT_CHECK
      description  : "This check reports a problem if the order is not 'order
hosts,bind' in /etc/host.conf"
      file         : "/etc/host.conf"
      search_locations : "/etc"
      regex        : "order hosts,bind"
      expect             : "order hosts,bind"
</custom_item>

# Example 9
# This is a better example of a file content check. It first looks
# for the string ".*LogLevel=.*" and if it matches it checks whether
# it matches .*LogLevel=9. For example, if the file was to have LogLevel=8
# this check will fail since the expected value is set to 9.
#

<custom_item>
      #System                  : "Linux"
      type              : FILE_CONTENT_CHECK
      description             : "This check reports a problem when the log
level setting in the sendmail.cf file is less than the value set in your
security policy."
      file              : "sendmail.cf"
      search_locations  : "/etc:/etc/mail:/usr/local/etc/mail"
      regex             : ".*LogLevel=.*"
        expect                : ".*LogLevel=9"
</custom_item>

# Example 10
# With compliance checks you can cause the shell to execute a command
# and parse the result to determine compliance. The check below determines
# whether the version of FreeBSD on the remote system is compliant with
# corporate standards. Note that since we determine the system type using
# the "system" tag, the check will skip if the remote OS doesn't match
# the one specified.

<custom_item>
      system            : "FreeBSD"
      type        : CMD_EXEC
      description : "Make sure that we are running FreeBSD 4.9 or higher"
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
      cmd         : "uname -a"
      expect            : "FreeBSD (4\.(9|[1-9][0-9])|[5-9]\.)"
</custom_item>

##################
#            #
# Builtin Checks #
#            #
##################

# Checks that are not customizable are built
# into the Unix compliance check module. Given below
# are the list of all the checks are the performed
# using the builtin functions. Please refer to the
# the Unix compliance checks documentation for more
# details about each check.
#


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

<item>
name: "max_password_age"
description : "Maximum password age"
value: "1..90"
</item>

<item>
name: "min_password_age"
description : "Minimum password age"
value: "6..21"
</item>

<item>
name: "accounts_bad_home_permissions"
description : "Account with bad home permissions"
</item>

<item>
name: "accounts_without_home_dir"
description : "Accounts without home directory"
</item>

<item>
name: "invalid_login_shells"
description: "Accounts with invalid login shells"
</item>

<item>
name: "login_shells_with_suid"
description : "Accounts with suid login shells"
</item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
<item>
name: "login_shells_writeable"
description : "Accounts with writeable shells"
</item>

<item>
name: "login_shells_bad_owner"
description : "Shells with bad owner"
</item>

<item>
name: "passwd_file_consistency"
description : "Check passwd file consistency"
</item>

<item>
name: "passwd_zero_uid"
description : "Check zero UID account in /etc/passwd"
</item>

<item>
name : "passwd_duplicate_uid"
description : "Check duplicate accounts in /etc/passwd"
</item>

<item>
name : "passwd_duplicate_gid"
description : "Check duplicate gid in /etc/passwd"
</item>


<item>
name : "passwd_duplicate_username"
description : "Check duplicate username in /etc/passwd"
</item>

<item>
name : "passwd_duplicate_home"
description : "Check duplicate home in /etc/passwd"
</item>

<item>
name : "passwd_shadowed"
description : "Check every passwd is shadowed in /etc/passwd"
</item>

<item>
name: "passwd_invalid_gid"
description : "Check every GID in /etc/passwd resides in /etc/group"
</item>

<item>
name : "group_file_consistency"
description : "Check /etc/group file consistency"
</item>

<item>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
name: "group_zero_gid"
description : "Check zero GUID in /etc/group"
</item>

<item>
name: "group_duplicate_name"
description : "Check duplicate group names in /etc/group"
</item>

<item>
name: "group_duplicate_gid"
description : "Check duplicate gid in /etc/group"
</item>

<item>
name : "group_duplicate_members"
description : "Check duplicate members in /etc/group"
</item>

<item>
name: "group_nonexistant_users"
description : "Check for nonexistent users in /etc/group"
</item>

</check_type>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

ПРИЛОЖЕНИЕ A. ПРИМЕР ФАЙЛА ПРОВЕРКИ СООТВЕТСТВИЯ ДЛЯ ОС WINDOWS

Примечание. Следующий файл можно получить через портал поддержки Tenable Support Portal, расположенный по адресу https://support.tenable.com/support-center/. В фактический файл могут быть внесены изменения, не отраженные в этом документе. Этот конкретный скрипт называется **financial_microsoft_windows_user_audit_guideline_v2.audit** и основывается на общем руководстве по повышению защищенности при администрировании пользователей. Эта политика рассчитана на разумную политику в отношении паролей, политику блокировки учетных записей и обеспечивает регистрацию событий входа в систему в журнале событий ОС Windows.

```
# (C) 2008 Tenable Network Security
#
# This script is released under the Tenable Subscription License and
# may not be used from within scripts released under another license
# without authorization from Tenable Network Security Inc.
#
# See the following licenses for details:
#
# http://cgi.tenablesecurity.com/Nessus_3_SLA_and_Subscription_Agreement.pdf
# http://cgi.tenablesecurity.com/Subscription_Agreement.pdf
#
# @PROFESSIONALFEED@
#
# $Revision: 1.2 $
# $Date: 2008/10/07 15:48:17 $
#
# Synopsis: This file will be read by compliance_check.nbin
#         to check compliance of a Windows host to
#           typical financial institution audit policy
#

<check_type:"Windows" version:"2">
<group_policy:"User audit guideline">

      <item>
      name: "Enforce password history"
      value: 24
      </item>

      <item>
      name: "Maximum password age"
      value: 90
      </item>

      <item>
      name: "Minimum password age"
      value: 1
      </item>

      <item>
      name: "Minimum password length"
      value: [12..14]
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.

```
        </item>

        <item>
        name: "Account lockout duration"
        value: [15..30]
        </item>

        <item>
        name: "Account lockout threshold"
        value: [3..5]
        </item>

        <item>
        name: "Reset lockout account counter after"
        value: [15..30]
        </item>

        <item>
                  name: "Audit account logon events"
                  value: "Success, Failure"
        </item>

          <item>
                  name: "Audit logon events"
                  value: "Success, Failure"
        </item>

</group_policy>
</check_type>
```

Авторские права © 2002 - 2011 гг. Tenable Network Security, Inc.