Referência das verificações de conformidade Nessus

12 de novembro de 2012

(Revisão 26)

Índice

Introdução

Este documento descreve a sintaxe usada para a criação de arquivos **.audit** personalizados, que podem ser usados para a auditoria da configuração do Unix, Windows, banco de dados e sistemas Scada, IBM iSeries, e Cisco com base em uma política de conformidade, bem como para a pesquisa de dados confidenciais no conteúdo de diversos sistemas.

A finalidade deste guia é auxiliar na criação do manual e a compreensão da sintaxe dos arquivos de auditoria. Consulte o documento em PDF Verificações de conformidade do Nessus, disponível no Tenable Support Portal para uma visão mais apronfudada do funcionamento das verificações de conformidade da Tenable.

O Nessus ofere suporte para auditoria no sistema SCADA. No entanto, esta funcionalidade não será discutida neste documento. Consulte a página de informações Nessus.org SCADA aqui para obter mais informações.

Pré-requisitos

Este documento requer algum nível de conhecimento sobre o scanner de vulnerabilidades Nessus, além da compreensão detalhada dos sistemas de destino a serem auditados. Para obter mais informações sobre como o Nessus pode ser configurado para executar auditorias de patches locais Unix e Windows, consulte o documento “Nessus Credentials Checks for Unix and Windows” (Verificações de credenciais do Nessus para Unix e Windows) disponível em http://www.nessus.org/documentation/.

Padrões e convenções

Este documento é a tradução de uma versão original em inglês. Algumas partes do texto permanecem em inglês para indicar a representação do próprio produto.

Em toda a documentação, nomes de arquivos, daemons e executáveis estão indicados com uma fonte **courier bold**.

As opções de linhas de comando e palavras-chave também são indicadas com a fonte **courier bold**. O exemplos de linhas de comando podem ou não conter o prompt da linha de comando e o texto gerado pelos resultados do comando. Os exemplos de linhas de comando exibirão o comando executado em **courier bold** para indicar o que o usuário digitou, enquanto que o exemplo de saída gerado pelo sistema será indicado em courier (sem negrito). Um exemplo da execução do comando **pwd** do Unix é apresentado a seguir:

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

As observações e considerações importantes são destacadas com este símbolo nas caixas de texto escurecidas.

As dicas, exemplos e práticas recomendadas são destacados com este símbolo em branco sobre fundo azul.

Referência de arquivo de conformidade de auditoria para configuração do Windows

A base para arquivos de conformidade **.audit** Windows é um arquivo de texto especialmente formatado. As entradas de dados no arquivo podem originar uma variedade de verificações de “item personalizado”, como verificações de configuração de registro, além de outras mais genéricas, como as verificações de configurações da política de segurança local. Em todo este guia são usados exemplos para esclarecimento.

Uso de aspas:

As aspas simples e aspas duplas são intercambiáveis quando estiverem delimitando campos de auditoria, exceto nos dois casos abaixo:

1. Nas verificações de conformidade do Windows em que os campos especiais como CRLF etc. devem ser interpretados literalmente, use aspas simples. Todos os campos incorporados interpretados como strings devem ser protegidos.

Por exemplo:

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

2. É necessário o uso de aspas ao usar “include_paths” e exclude_paths do WindowsFiles ”.

Se os strings forem usados em qualquer tipo de campo (descrição, value_data, regex etc.) que contém aspas simples ou aspas duplas, proceda da seguinte maneira:

a. Use o tipo oposto de aspa para as aspas delimitadoras mais afastadas.

Por exemplo:

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

b. Isole as aspas incorporadas com uma barra invertida (somente aspas duplas).

Por exemplo:

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

Tipo de verificação

Todas as verificações de conformidade do Windows devem estar entre colchetes angulares com o encapsulamento **`check_type`**, a designação “Windows” e a versão “2” especificada:

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

Um exemplo de verificação de conformidade do Windows pode ser visto no “Apêndice B”, que começa com a configuração **`check_type`** para “Windows” e versão “2” e termina com a tag “**`</check_type>`**”.

Isto é necessário para diferenciar os arquivos **`.audit`** do Windows daqueles destinados ao Unix (ou outra plataforma).

Dados de valor

A sintaxe do arquivo **`.audit`** contém palavras-chave às quais podem ser atribuídos variados tipos de valor para personalizar suas verificações. Esta seção descreve estas palavras-chave e o formato dos dados que podem ser inseridos.

Tipos de dados

Os seguintes tipos de dados podem ser inseridos para as verificações:

Tipo de dados	Descrição
DWORD	0 a 2.147.483.647
RANGE [X..Y]	Onde X é um DWORD ou MIN e Y é um DWORD ou MAX

Exemplos:

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

Além disso, os números podem ser especificados com o sinal de positivo (+) ou negativo (-) e podem ser especificados como valores hexadecimais. Os valores hexadecimais e sinais podem ser combinados. Os exemplos a seguir são válidos (sem o rótulo correspondente entre parênteses) dentro de uma auditoria de REGISTRY_SETTING para um POLICY_DWORD:

```
value_data: -1 (sem sinalcom sinal)
value_data: +10 (com sinal)
value_data: 10 (sem sinal)
value_data: 2401649476 (sem sinal)
value_data: [MIN..+10] (campo com sinal)
value_data: [20..MAX] (campo sem sinal)
value_data: 0x800010AB (hex sem sinal)
value_data: -0x10 (hex com sinal)
```

Expressões complexas

Expressões complexas podem ser usadas para o campo **`value_data`** com o uso de:

- ||: OU condicional
- &&: E condicional
- |: OU binário (operação de bits)
- &: E binário (operação de bits)
- (e): para delimitar expressões complexas

Exemplos:

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

Campo “check_type”

Este tipo de verificação é diferente do campo “**`check_type`**” acima especificado que é usado no início de cada arquivo de auditoria para indicar o tipo genérico de auditoria (Windows, WindowsFiles, Unix, Database, Cisco). O campo é opcional e pode ser executado em valores **`value_data`** do Windows para determinar o tipo de verificação a ser executada. Estão disponíveis as seguintes configurações:

- CHECK_EQUAL: compara o valor remoto ao valor da política (valor padrão se **`check_type`** estiver ausente).
- CHECK_EQUAL_ANY: verifica se cada elemento de **`value_data`** está presente pelo menos uma vez na lista do sistema.
- CHECK_NOT_EQUAL: verifica se o valor remoto é diferente do valor da política.
- CHECK_GREATER_THAN: verifica se o valor remoto é superior ao valor da política.

- CHECK_GREATER_THAN_OR_EQUAL: verifica se o valor remoto é maior ou igual ao valor da política.
- CHECK_LESS_THAN: verifica se o valor remoto é inferior ao valor da política.
- CHECK_LESS_THAN_OR_EQUAL: verifica se o valor remoto é menor ou igual ao valor da política.
- CHECK_REGEX: verifica se o valor remoto coincide com a regex no valor da política (funciona apenas com POLICY_TEXT e POLICY_MULTI_TEXT).
- CHECK_SUBSET: verifica se o ACL remoto é um subconjunto da política ACL (funciona apenas com ACLs).
- CHECK_SUPERSET: verifica se o ACL remoto é um superconjunto da política ACL (funciona apenas com ACLs de direitos de recusa).

O exemplo de auditoria para verificação para certificar-se de que não exista o nome de conta “Guest” para qualquer conta de convidado.

```
<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>
```

Se houver algum outro valor além de “Guest”, o teste será aprovado. Se “Guest” for detectado, a auditoria resultará em falha.

O campo “group_policy”

O campo “**group_policy**” pode ser usado para fornecer um string de texto curto que descreve a auditoria. A **group_policy** deve ser incluída em um arquivo de auditoria e deve ser inserida após o campo **check_type.**

```
<check_type: "Windows" version:"2">
<group_policy: "Audit file for Windows 2008">

…

</group_policy>
</check_type>
```

Campo “info”

O campo opcional “**info**” pode ser usado para rotular cada campo de auditoria com uma ou mais referências externas. Por exemplo: este campo será usado para a colocação de referências provenientes dos tags NIST CCE, além de requisitos específicos de auditoria CIS. As referências externas são impressas na auditoria final executada pelo Nessus e serão exibidas no relatório do Nessus ou por meio da interface com o usuário do SecurityCenter.

O exemplo de política de auditoria de senhas a seguir foi ampliado para relacionar referências a uma política corporativa fictícia:

```
<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>
```

Se forem necessárias várias referências de política para uma única auditoria, o string especificado pela palavra-chave "**info**" usará o separador "\n" para especificar diversos strings. Vejamos o exemplo de auditoria a seguir:

```
<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>
```

Quando executada com a ferramenta de linha de comando **nasl**, esta função de auditoria produz a seguinte saída:

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

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"
```

O campo "debug"

O campo "**debug**" pode ser usado para solucionar problemas de conformidade do conteúdo do Windows. A palavra-chave debug envia informações sobre a verificação do conteúdo que está sendo realizada, como o(s) arquivo(s) processado(s) e analisado(s), e se os resultados foram encontrados. Devido à grande quantidade de informações, esta palavra-chave deve ser usada apenas para fins de resolução de problemas. Por exemplo:

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

Formato ACL

Esta seção descreve a sintaxe usada para determinar se um arquivo ou pasta possui a configuração ACL desejada.

Verificações de controle de acesso a arquivos

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

Uma Lista de Controle de Acesso (ACL) é identificada pela palavra-chave **`file_acl`**. O nome da ACL deve ser exclusivo para ser usado com um item de permissões de arquivo. Uma ACL de arquivo pode conter uma ou várias entradas de dados pelo usuário.

Tipos associados	Tipos permitidos
`acl_inheritance`	not inherited inherited not used
`acl_apply`	• this folder only • this object only • this folder and files • this folder and subfolders • this folder, subfolders and files • files only • subfolders only • subfolders and files only
`acl_allow` **`acl_deny`**	Estas configurações são opcionais. Os direitos genéricos são: • full control • modify • read & execute • read • write • list folder contents Os direitos avançados são: • full control • traverse folder / execute file • list folder / read data • read attributes • read extended attributes • create files / write data

	• create folders / append data • write attributes • write extended attributes • delete subfolder and files • delete • read permissions • change permissions • take ownership

Exemplo de texto de controle de acesso a arquivo **`.audit`**:

```
 <file_acl: "ASU1">

  <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>
```

Verificações de controle de acesso a registros

Uso

```
<registry_acl: ["name"]>

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

</acl>
```

Uma ACL de registro é identificada pela palavra-chave **`registry_acl`**. O nome da ACL deve ser exclusivo para ser usado com um item de permissões de registro. A ACL de registro pode conter uma ou várias entradas de dados pelo usuário.

Tipos associados	Tipos permitidos
acl_inheritance	• not inherited • inherited • not used
acl_apply	• this key only • this key and subkeys • subkeys only
acl_allow **acl_deny**	Estas configurações são opcionais e são usadas para definir os direitos que um usuário tem sobre o objeto. Os direitos genéricos são: • full control • read Os direitos avançados são: • full control • query value • set value • create subkey • enumerate subkeys • notify • create link • delete • write dac • write owner • read control

Exemplo de texto de lista de controle de acesso ao registro **.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"
 acl_allow: "Full Control"
 </user>
```

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

 </acl>
```

Verificações de controle de acesso a serviços

Uso

```
<service_acl: ["name"]>

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

</acl>
```

A ACL de serviço é identificada pela palavra-chave **`service_acl`**. O nome da ACL deve ser exclusivo para ser usado com um item de permissões de serviço. A ACL de serviço pode conter uma ou várias entradas de dados pelo usuário.

Tipos associados	Tipos permitidos
`acl_inheritance`	• not inherited • inherited • not used
`acl_apply`	• this object only
`acl_allow` **`acl_deny`**	Estas configurações são opcionais e são usadas para definir os direitos que um usuário tem sobre o objeto. Os direitos genéricos são: • full control • read • start, stop and pause • write • delete Os direitos avançados são: • full control • delete • query template • change template • query status

	• enumerate dependents • start • stop • pause and continue • interrogate • user-defined control • read permissions • change permissions • take ownership

Exemplo de verificação de controle de acesso a serviço:

```
<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>
```

Verificações de controle de permissão de lançamento

Uso

```
<launch_acl: ["name"]>

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

</acl>
```

Uma launch ACL é identificada pela palavra-chave **`launch_acl`**. O nome da ACL deve ser exclusivo para ser usado com um item de permissões de lançamento DCOM. A launch ACL pode conter uma ou várias entradas de dados pelo usuário.

Tipos associados	Tipos permitidos
`acl_inheritance`	• not inherited • inherited
`acl_apply`	• this object only

acl_allow **acl_deny**	Estas configurações são opcionais e são usadas para definir os direitos que um usuário tem sobre o objeto. Os direitos genéricos são: • local launch • remote launch • local activation • remote activation

Esta ACL funciona apenas com Windows XP/2003/Vista (e parcialmente com Windows 2000).

Exemplo de verificação de controle de acesso a launch:

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

 </acl>
```

Verificações de controle de permissão de Launch2

Uso

```
<launch2_acl: ["name"]>

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

</acl>
```

Uma ACL launch2 é identificada pela palavra-chave **`launch2_acl`**. O nome da ACL deve ser exclusivo para ser usado com um item de permissões de lançamento DCOM. A ACL launch2 pode conter uma ou várias entradas de dados pelo usuário.

Tipos associados	Tipos permitidos
`acl_inheritance`	• not inherited • inherited
`acl_apply`	• this object only
`acl_allow` **`acl_deny`**	Estas configurações são opcionais e são usadas para definir os direitos que um usuário tem sobre o objeto. Os direitos genéricos são: • launch

Use o ACL launch2 com sistemas Windows 2000 e NT somente.

Exemplo de verificação de controle de acesso a launch:

```
 <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>
```

Verificações de controle de permissão de acesso

Uso

`<access_acl: ["name"]>`
 	
`<user: ["user_name"]>`

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

</acl>
```

Uma ACL de acesso é identificada pela palavra-chave **`access_acl`**. O nome da ACL deve ser exclusivo para ser usado com um item de permissões de acesso DCOM. A ACL de acesso pode conter uma ou várias entradas de dados pelo usuário.

Tipos associados	Tipos permitidos
`acl_inheritance`	• not inherited • inherited
`acl_apply`	• this object only
`acl_allow` **`acl_deny`**	Estas configurações são opcionais e são usadas para definir os direitos que um usuário tem sobre o objeto. Os direitos genéricos são: • local access • remote access

Exemplo de verificação de controle de acesso:

```
 <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>
```

Itens personalizados

Um item personalizado é uma verificação completa definida com base nas palavras-chave definidas anteriormente. A lista a seguir relaciona os tipos de item personalizado disponíveis. Cada verificação começa com um tag "**<custom_item>**" e termina com "**</custom_item>**". As tags contêm listas de uma ou mais palavras-chave que são interpretadas pelo analisador de verificação de conformidade para executar as verificações.

As verificações de auditoria personalizadas podem usar "**</custom_item>**" e "**</item>**" de forma intercambiável para o tag de encerramento.

PASSWORD_POLICY

Uso

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

Este item de política verifica os valores definidos em "Windows Settings -> Security Settings -> Account Policies -> Password Policy" (Configurações do Windows-> Configurações de segurança -> Diretivas de conta -> Diretivas de senha).

A verificação é executada ao acessar a função `NetUserModalsGet` com o nível 1.

Estes itens usam o campo **password_policy** para descrever qual elemento da política de senhas deve ser auditado. Os tipos permitidos são:

- ENFORCE_PASSWORD_HISTORY ("Aplicar histórico de senhas")
  `value_type: POLICY_DWORD`
  `value_data: DWORD or RANGE [number of remembered passwords]`
- MAXIMUM_PASSWORD_AGE ("Tempo de vida máximo da senha")
  `value_type: TIME_DAY`
  `value_data: DWORD or RANGE [time in days]`
- MINIMUM_PASSWORD_AGE ("Tempo de vida mínimo da senha")
  `value_type: TIME_DAY`
  `value_data: DWORD or RANGE [time in days]`
- MINIMUM_PASSWORD_LENGTH ("Comprimento mínimo da senha")
  `value_type: POLICY_DWORD`
  `value_data: DWORD or RANGE [minimum number of characters in the password]`
- COMPLEXITY_REQUIREMENTS ("A senha deve satisfazer a requisitos de complexidade")
  `value_type: POLICY_SET`
  `value_data: "Enabled" or "Disabled"`
- REVERSIBLE_ENCRYPTION ("Armazenar senhas usando criptografia reversível de todos os usuários no domínio")

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

- FORCE_LOGOFF (“Segurança da rede: Forçar logoff quando o horário de logon expirar”)
```
value_type: POLICY_SET
value_data: "Enabled" or "Disabled"
```

Atualmente, não há maneira de verificar a política “Armazenar senha usando criptografia reversível para todos os usuários no domínio”.

A política FORCE_LOGOFF está localizada em “Configurações de segurança -> Diretivas locais -> Opções de segurança”.

Exemplo de auditoria da política de senhas:

```
<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

Uso

```
<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>
```

Este item de política verifica os valores definidos em “Configurações de segurança -> Diretivas de conta -> Diretiva de bloqueio de conta”.

A verificação é executada ao acessar a função `NetUserModalsGet` com o nível 3.

Este item usa o campo **`lockout_policy`** para descrever qual elemento da política de senhas deve ser auditado. Os tipos permitidos são:

- LOCKOUT_DURATION (“Duração do bloqueio da conta”)
```
value_type: TIME_MINUTE
value_data: DWORD or RANGE [time in minutes]
```

- LOCKOUT_THRESHOLD (“Limite de bloqueio de conta”)
```
value_type: POLICY_DWORD
value_data: DWORD or RANGE [time in days]
```

- LOCKOUT_RESET (“Zerar contador de bloqueios de conta após”)
```
value_type: TIME_MINUTE
value_data: DWORD or RANGE [time in minutes]
```

Observe o exemplo a seguir:

```
<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

Uso

```
<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>
```

Este item de política verifica os valores definidos em “Security Settings -> Account Policies -> Kerberos Policy”. (Configurações de segurança -> Diretivas de conta -> Diretiva do Kerberos).

A verificação é executada ao acessar a função `NetUserModalsGet` com o nível 1.

Este item usa o campo **`kerberos_policy`** para descrever qual elemento da política de senhas deve ser auditado. Os tipos permitidos são:

- USER_LOGON_RESTRICTIONS (“Impor restrições ao logon do usuário”)
  `value_type: POLICY_SET`
  `value_data: "Enabled" or "Disabled"`
- SERVICE_TICKET_LIFETIME (“Tempo de vida máximo para o tíquete de serviço”)
  `value_type: TIME_MINUTE`
  `value_data: DWORD or RANGE [time in minutes]`
- USER_TICKET_LIFETIME (“Tempo de vida máximo para o tíquete de usuário”)
  `value_type: TIME_HOUR`
  `value_data: DWORD or RANGE [time in hours]`
- USER_TICKET_RENEWAL_LIFETIME (“Tempo de vida máximo para renovação do tíquete do usuário”)
  `value_type: TIME_DAY`
  `value_data: DWORD or RANGE [time in day]`
- CLOCK_SYNCHRONIZATION_TOLERANCE (“Tolerância máxima para a sincronização do relógio do computador”)
  `value_type: TIME_MINUTE`
  `value_data: DWORD or RANGE [time in minute]`

A política do Kerberos só pode ser verificada com base em KDC (Key Distribution Center) (Centro de Distribuição de Chaves), que, no Windows, é geralmente um controlador de domínio.

Exemplo:

```
 <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

Uso

```
<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>
```

Este item de política verifica os valores definidos em “Security Settings -> Local Policies -> Audit Policy”. (Configurações de segurança -> Diretivas locais -> Diretiva de auditoria).

A verificação é executada ao acessar a função `LsaQueryInformationPolicy` com o nível `PolicyAuditEventsInformation`.

Este item usa o campo **`audit_policy`** para descrever qual elemento da política de senhas deve ser auditado. Os tipos permitidos são:

- AUDIT_ACCOUNT_LOGON (“Auditoria de eventos de logon de conta”)
- AUDIT_ACCOUNT_MANAGER (“Auditoria de gerenciamento de conta”)
- AUDIT_DIRECTORY_SERVICE_ACCESS (“Acesso ao serviço de diretório de auditoria”)
- AUDIT_LOGON (“Auditoria de eventos de logon”)
- AUDIT_OBJECT_ACCESS (“Auditoria de acesso a objetos”)
- AUDIT_POLICY_CHANGE (“Auditoria de alteração de diretivas”)
- AUDIT_PRIVILEGE_USE (“Auditoria de uso de privilégios”)
- AUDIT_DETAILED_TRACKING (“Auditoria de acompanhamento de processos”)
- AUDIT_SYSTEM (“Auditoria de eventos de sistema”)

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

Observe que há um espaço exigido em “Success, Failure”.

Exemplo:

```
 <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

Uso

```
<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>
```

Este item de política verifica os valores relacionados em auditpol /get /category:*.

A verificação é realizada ao executar de cmd.exe auditpol /get /category:* por meio de WMI.

Este item usa o campo **`audit_policy_subcategory`** para determinar qual subcategoria precisa ser auditada. Os tipos de SUBCATEGORY_POLICY_TYPE permitidos são:

- Alteração no estado de segurança
- Extensão do sistema de segurança
- Integridade do sistema
- Driver IPsec
- Eventos do Sistema
- Logon
- Logoff
- Bloqueio de conta
- Modo principal do IPsec
- Modo rápido do IPsec
- Modo estendido do IPsec
- Logon especial
- Outros eventos de logon/logoff
- Servidor de Diretiva de Rede
- Sistema de arquivos
- Registro
- Objeto Kernel
- SAM
- Serviços de certificação

- Aplicativo gerado
- Manipulação de Identificador
- Compartilhamento de arquivos
- Descarte de Pacote de Plataforma de Filtragem
- Conexão de Plataforma de Filtragem
- Outros Eventos de Acesso a Objetos
- Uso de Privilégio Importante
- Uso de Privilégio Não Importante
- Outros Eventos de Uso de Privilégios
- Criação de processo
- Terminação de processo
- Atividade DPAPI
- Eventos RPC
- Auditoria de alteração de diretivas
- Alteração na diretiva de autenticação
- Alteração na Diretiva de Autorização
- Alteração na diretiva de nível de regra MPSSVC
- Alteração na Diretiva da Plataforma de Filtragem
- Outros eventos de alteração de diretiva
- Gerenciamento de conta de usuário
- Gerenciamento de conta de computador
- Gerenciamento de grupo de segurança
- Gerenciamento de grupo de distribuição
- Gerenciamento de grupo de aplicativos
- Outros Eventos de Gerenciamento de Contas
- Acesso ao Serviço de Diretórios
- Alterações no serviço de diretório
- Replicação do serviço de diretório
- Replicação detalhada do serviço de diretório
- Validação de Credenciais
- Operações do Tíquete de serviço Kerberos
- Eventos de Logon de Conta

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

Observe que há um espaço exigido em “Success, Failure”.

Esta verificação é aplicável apenas para Windows Vista/2008 Server e posteriores. Se houver um firewall ativado, além de adicionar o WMI como exceção nas configurações do firewall, “Firewall do Windows: permitir exceção de administração remota de entrada” também deve ser habilitado nas configurações de firewall com o uso de **`gpedit.msc`**. É provável que esta verificação não funcione em sistemas Vista/2008 em inglês ou sistemas que não tenham auditpol instalado.

Exemplo:

```
<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>
```

AUDIT_POWERSHELL

Uso

```
<custom_item>
 type: AUDIT_POWERSHELL
 description: "Powershell check"
 value_type: [value_type]
 value_data: [value]
 powershell_args: ["arguments for powershell.exe"]
 (optional) only_show_cmd_output: YES or NO
 (optional) check_type: [CHECK_TYPE]
 (optional) severity: ["HIGH" or "MEDIUM" or "LOW"]
 (optional) powershell_option: CAN_BE_NULL
 (optional) powershell_console_file: "C:\Program Files\Microsoft\Exchange
Server\ExShell.psc1"
</custom_item>
```

Esta verificação executa **`powershell.exe`** no servidor remoto juntamente com os argumentos fornecidos com "**`powershell_args`**" e retorna a saída de comando se "**`only_show_cmd_output`**" for definido como YES ou compara o resultado com "**`value_data`**" se **`value_data`** for especificado.

Tipos associados:

Este item usa o campo "**`powershell_args`**" para especificar os argumentos que devem ser fornecidos para **`powershell.exe`**. Se o local de powershell.exe não for padrão, é preciso usar a palavra-chave **`powershell_console_file`** para especificar o local. Atualmente, apenas "`get-`" cmdlets são aceitos. Por exemplo:

- get-hotfix | where-object {$_.hotfixid -ne 'File 1'} | select Description,HotFixID,InstalledBy | format-list
- get-wmiobject win32_service | select caption,name, state| format-list
- (get-WmiObject -namespace root\MicrosoftIISv2 -Class IIsWebService).ListWebServiceExtensions().Extensions
- get-wmiobject -namespace root\cimv2 -class win32_product | select Vendor,Name,Version | format-list
- get-wmiobject -namespace root\cimv2\power -class Win32_powerplan | select description,isactive | format-list

O item usará o campo opcional "**`only_show_cmd_output`**" se toda a saída de comando precisar ser informada:

- only_show_cmd_output : YES or NO

Outras considerações:

1. Se "**`only_show_cmd_output`**" for definido, mas a gravidade da saída tiver de ser definida, é possível usar a tag de gravidade para alterar a gravidade. O padrão é INFO.
2. Normalmente, o PowerShell não está instalado em alguns sistemas operacionais Windows (por exemplo: XP, 2003) e, em tais sistemas, a verificação não produzirá nenhum resultado. Portanto, certifique-se de que o PowerShell esteja instalado no destino remoto antes de usar esta opção.
3. Para que a verificação funcione corretamente, o serviço WMI deve ser habilitado. Além disso, configure o firewall para "Permitir exceção de administração remota de entrada".
4. Os alias cmdlet (por exemplo: "`gps`" em vez de "`Get-Process`") não são permitidos.

Exemplo:

Este exemplo executa o cmdlet “Get-Hotfix” do powershell e especifica um where-object para não selecionar correções com a ID “File 1” e, em seguida, informa Description, HotfixID e Installedby formatados como uma lista.

```
<custom_item>
 type: AUDIT_POWERSHELL
 description: "Show Installed Hotfix"
 value_type: POLICY_TEXT
 value_data: ""
 powershell_args: "get-hotfix | where-object {$_.hotfixid -ne 'File 1'} | select
       Description,HotFixID,InstalledBy | format-list"
 only_show_cmd_output: YES
</custom_item>
```

Exemplo:

Este exemplo verifica se o serviço do Windows “WinRM” está em execução.

```
<custom_item>
 type: AUDIT POWERSHELL
 description: "Check if WinRM service is running"
 value_type: POLICY_TEXT
 value_data: "Running"
 powershell_args: "get-wmiobject win32_service | where-object {$_.name -eq 'WinRM' -
       and $ .state -eq 'Running'} | select state"
 check_type: CHECK_REGEX
</custom_item>
```

AUDIT_FILEHASH_POWERSHELL

Uso

```
<custom_item>
 type: AUDIT_FILEHASH_POWERSHELL
 description: "Powershell FileHash Check"
 value_type: POLICY_TEXT
 file: "[FILE]"
 value_data: "[FILE HASH]"
</custom_item>
```

Essa verificação executa powershell.exe no servidor remoto junto com as informações fornecidas para comparar um hash de arquivo esperado com o hash do arquivo no sistema.

Outras considerações:

- Por padrão, um hash MD5 do arquivo é comparado. Porém, os usuários podem comparar hashes gerados com os algoritmos SHA1, SHA256, SHA384, SHA512 ou RIPEMD160.
- Para que a verificação funcione, o PowerShell deve estar instalado e o WMI deve estar habilitado no destino.

Exemplo:

Esse exemplo compara um hash MD5 fornecido com o hash de arquivo de C:\test\test2.zip.

```
<custom_item>
 type: AUDIT_FILEHASH_POWERSHELL
 description: "Audit FILEHASH - MD5"
 value_type: POLICY_TEXT
 file: "C:\test\test2.zip"
 value_data: "8E653F7040AC4EA8E315E838CEA83A04"
</custom_item>
```

Exemplo:

Esse exemplo compara um hash SHA1 fornecido com o hash de arquivo de C:\test\test3.zip.

```
<custom_item>
 type: AUDIT_FILEHASH_POWERSHELL
 description: "Audit FILEHASH - SHA1"
 value_type: POLICY_TEXT
 file: "C:\test\test3.zip"
 value_data: "0C4B0AF91F62ECCED3B16D35DE50F66746D6F48F"
 hash_algorithm: SHA1
</custom_item>
```

AUDIT_IIS_APPCMD

Uso

```
<custom_item>
 type: AUDIT_IIS_APPCMD
 description: "Test appcmd output"
 value_type: [value_type]
 value_data: [value]
 appcmd_args: ["arguments for appcmd.exe"]
 (optional) only_show_cmd_output: YES or NO
 (optional) check_type: [CHECK_TYPE]
 (optional) severity: ["HIGH" or "MEDIUM" or "LOW"]
</custom_item>
```

Esta verificação executa **`appcmd.exe`** em um servidor com IIS, juntamente com os argumentos que utilizam "**`appcmd_args`**" especificados, e determina a conformidade ao comparar a saída com **`value_data`**. Em alguns casos (por exemplo: configuração de listagem), pode ser desejável informar apenas a saída do comando. Nesse caso, "**`only_show_cmd_output`**" deve ser usado.

Essa verificação é aplicável somente para Internet Information Services (IIS) versão 7 do Windows.

Esse item usa o campo "**`appcmd_args`**" para especificar os argumentos que devem ser fornecidos para **`appcmd.exe`**. Atualmente, a única "lista" de comandos que pode ser especificada.

- `list sites`
- `list AppPools /processModel.identityType:ApplicationPoolIdentity`
- `list config`
- `list config -section:system.web/authentication`

- `list app`

O item usará o campo opcional "**`only_show_cmd_output`**" se toda a saída de comando precisar ser informada.

Exemplo:

Essa verificação compara o resultado de "`appcmd.exe list AppPools /processModel.identityType:ApplicationPoolIdentity`" com **`value_data`** e será aprovada somente se a saída contiver 'APPPOOL "DefaultAppPool"'

```
<custom_item>
 type: AUDIT_IIS_APPCMD
 description: "Set Default Application Pool Identity to Least Privilege Principal"
 value_type: POLICY_TEXT
 value_data: 'APPPOOL "DefaultAppPool"'
 appcmd_args: "list AppPools /processModel.identityType:ApplicationPoolIdentity"
 check_type: CHECK_REGEX
</custom_item>
```

AUDIT_ALLOWED_OPEN_PORTS

Uso

```
<custom_item>
 type: AUDIT_ALLOWED_OPEN_PORTS
 description: "Audit Open Ports"
 value_type: [value_type]
 value_data: [value]
 port_type: [port_type]
<item>
```

Essa verificação consulta a lista de portas TCP/UDP abertas no destino e as compara com uma lista de portas permitidas. A verificação baseia-se na saída de "netstat –ano" ou "netstat –an" para obter uma lista de portas abertas e, depois, verifica se as portas estão realmente abertas, verificando o estado da porta usando (get_port_state()/get_udp_port_state()).

Considerações:

- **`value_data`** também aceita um regex como um intervalo de portas; portanto, algo como 8[0-9]+ também funcionará.

Exemplos:

O seguinte exemplo compara "**`value_data`**" com uma lista de portas TCP abertas no destino:

```
<custom_item>
 type: AUDIT_ALLOWED_OPEN_PORTS
 description: "Audit TCP OPEN PORTS"
 value_type: POLICY_PORTS
 value_data: "80,135,445,902,912,1024,1025,3389,5900,8[0-
      9]+,18208,32111,38311,47001,139"
 port_type: TCP
</custom_item>
```

O seguinte exemplo compara "**`value_data`**" com uma lista de portas UDP abertas no destino:

```
<custom_item>
 type: AUDIT_ALLOWED_OPEN_PORTS
 description: "Audit UDP OPEN PORTS"
 value_type: POLICY_PORTS
 value_data: "161,445,500,1026,4501,123,137,138,5353"
 port_type: UDP
</custom_item>
```

AUDIT_DENIED_OPEN_PORTS

Uso

```
<custom_item>
 type: AUDIT_DENIED_OPEN_PORTS
 description: "Audit Denied Open Ports"
 value_type: [value_type]
 value_data: [value]
 port_type: [port_type]
<item>
```

Essa verificação consulta a lista de portas TCP/UDP abertas no destino e as compara com uma lista de portas negadas. A verificação baseia-se na saída de “netstat –ano” ou “netstat –an” para obter uma lista de portas abertas e, depois, verifica se as portas estão realmente abertas, verificando o estado da porta usando (get_port_state()/get_udp_port_state()).

Os tipos permitidos são:

- value_type: POLICY_PORTS
- value_data: "80,135,445,902,912,1024,1025,3389,5900,8[0-9]+,18208,32111,38311,47001,139"
- port_type: TCP ou UDP

Considerações:

- **value_data** também aceita um regex como um intervalo de portas; portanto, algo como 8[0-9]+ também funcionará.

Exemplos:

O seguinte exemplo compara “**value_data**” com uma lista de portas TCP abertas no destino.

```
<custom_item>
 type: AUDIT_DENIED_OPEN_PORTS
 description: "Audit TCP OPEN PORTS"
 value_type: POLICY_PORTS
 value_data: "80,443"
 port_type: TCP
</custom_item>
```

O seguinte exemplo compara “**value_data**” com uma lista de portas UDP abertas no destino.

```
<custom_item>
 type: AUDIT_DENIED_OPEN_PORTS
```

```
  description: "Audit UDP OPEN PORTS"
  value_type: POLICY_PORTS
  value_data: "161,5353"
  port_type: UDP
 </custom_item>
```

AUDIT_PROCESS_ON_PORT

Uso

```
<custom_item>
 type: AUDIT_PROCESS_ON_PORT
 description: "Audit Process on Port"
 value_type: [value_type]
 value_data: [value]
 port_type: [port_type]
 port_no: [port_no]
 port_option: [port_option]
 check_type: CHECK_TYPE
<item>
```

Essa verificação consulta o processo em execução em uma determinada porta. A verificação baseia-se na saída de “netstat -ano” e “tasklist /svc” para determinar qual processo está em execução em qual porta TCP/UDP.

Os tipos permitidos são:

- value_type: POLICY_TEXT
- value_data: String arbitrário, como por exemplo, "foo.exe"
- port_type: TCP ou UDP
- port_no: no. de porta, como por exemplo, 80, 445
- port_option: CAN_BE_CLOSED

Considerações:

- Se port_option estiver definida como CAN_BE_CLOSED, a verificação retorna um resultado PASS se a porta não está aberta no sistema remoto. Caso contrário, gera um erro.
- Windows 2000 e anteriores não são compatíveis com “netstat –ano”; portanto, essa verificação só funciona no Windows XP e mais recentes.

Exemplos:

O exemplo a seguir verifica se o processo em execução na porta tcp 5900 é “vss.exe” ou “vssrvc.exe”.

```
 <custom_item>
  type: AUDIT_PROCESS_ON_PORT
  description: "Audit OPEN PORT SERVICE"
  value_type: POLICY_TEXT
  value_data: "vssrvc.exe" || "vss.exe"
  port_type: TCP
```

```
  port_no: "5900"
  port_option: CAN_BE_CLOSED
 </custom_item>
```

O exemplo a seguir é semelhante ao primeiro exemplo, exceto que demonstra o uso de check_type.

```
 <custom_item>
  type: AUDIT_PROCESS_ON_PORT
  description: "Audit Process on Port - check_regex"
  value_type: POLICY_TEXT
  value_data: "foo.exe" || "vss.+"
  port_type: TCP
  port_no: "5900"
  check_type: CHECK_REGEX
 </custom_item>
```

CHECK_ACCOUNT

Uso

```
<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>
```

Este item de política verifica os seguintes valores definidos em “Configurações de segurança -> Diretivas locais -> Opções de segurança”.

- Contas: Status da conta de administrador
- Contas: Status da conta de convidado
- Contas: Renomear conta de administrador
- Contas: Renomear conta de convidado

A verificação é executada ao acessar a função `LsaQueryInformationPolicy` com o nível `PolicyAccountDomainInformation` para obter o SID do domínio/sistema, `LsaLookupSid` para obter os nomes do administrator e do convidado e `NetUserGetInfo` para obter informações sobre a conta.

Este item usa o campo **`account_type`** para descrever qual conta deve ser auditada. Os tipos permitidos são:

- ADMINISTRATOR_ACCOUNT (“Contas: Status da conta de administrador”)
  `value_type: POLICY_SET`
  `value_data: "Enabled" or "Disabled"`
- GUEST_ACCOUNT (“Contas: Status da conta de convidado”)
  `value_type: POLICY_SET`
  `value_data: "Enabled" or "Disabled"`

- ADMINISTRATOR_ACCOUNT (“Contas: Renomear conta de administrador”)
  ```
  value_type: POLICY_TEXT
  value_data: "TEXT HERE" [administrator name]
  check_type: [CHECK_TYPE] (any one of the possible check_type values)
  ```
- GUEST_ACCOUNT (“Contas: Renomear conta de convidado”)
  ```
  value_type: POLICY_TEXT
  value_data: "TEXT HERE" [guest name]
  check_type: [CHECK_TYPE] (any one of the possible check_type values)
  ```

Dependendo da parte de credenciais de domínio, as contas do sistema local ou as contas de domínio poderão ser verificadas.

Exemplos:

```
<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>

<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

Uso

```
<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>
```

Este item de política verifica nomes de grupos e o status de grupos relacionados em **`lusmgr.msc`**.

Este item usa o campo **`group_type`** para descrever qual conta deve ser auditada. Os tipos permitidos são:

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

Os tipos permitidos para o campo **`value_type`** são:

- POLICY_SET (é verificado o status do grupo)
  `value_type: POLICY_SET`
  `value_data: "Enabled" or "Disabled"`
- POLICY_TEXT (é verificado o nome do grupo)
  `value_type: POLICY_TEXT`
  `value_data: "Guests1"` (Neste caso, **`value_data`** pode ser qualquer string de texto)

Exemplos:

```
<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

Uso

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

Este item de política verifica o seguinte valor definido em “Configurações de segurança -> Diretivas locais -> Opções de segurança -> Acesso à rede: permitir SID anônimo/conversão de nomes”. A verificação é executada ao acessar a função `LsaQuerySecurityObject` no identificador diretiva de LSA.

Os tipos permitidos são:

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

Ao usar esta auditoria, observe que a política:

- é uma verificação de permissão no serviço LSA.
- verifica se o ANONYMOUS_USER possui o sinalizador POLICY_LOOKUP_NAMES configurado.
- não é aceita no Windows 2003 porque um usuário anônimo não consegue acessar o pipe LSA.

Exemplo:

```
 <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

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

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

Este item de política verifica os valores de inicialização definidos em “Serviços do Sistema”. A verificação é executada ao acessar a função **`RegQueryValueEx`** com as seguintes chaves:

- chave: "SYSTEM\CurrentControlSet\Services\" + nome_do_serviço
- item: "Iniciar"

Os tipos permitidos são:

```
value_type: SERVICE_SET
value_data: "Automatic", "Manual" or "Disabled"
svc_option: CAN_BE_NULL or CAN_NOT_BE_NULL
```

O campo **`service_name`** corresponde ao verdadeiro nome do serviço. O nome pode ser obtido por:

1. início do painel de controle de serviços (em Ferramentas Administrativas)
2. escolha do serviço desejado
3. abertura da caixa de diálogo Propriedades (ao clicar com o botão direito do mouse em Propriedades)
4. extração da parte “nome do serviço”

A configuração da permissão de serviço pode ser verificada com o item SERVICE_PERMISSIONS.

Exemplo:

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

GROUP_MEMBERS_POLICY

Uso

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

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

Este item da política verifica se há uma lista específica de usuários presente em um ou mais grupos.

O tipo permitido é:

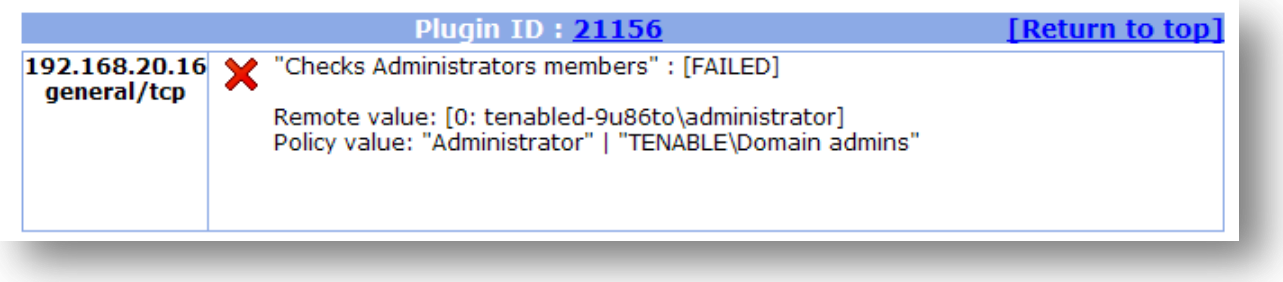
```
value_type: POLICY_TEXT or POLICY_MULTI_TEXT
value_data: "user1" && "user2" && ... && "usern"
```

Ao usar esta auditoria, observe que é possível especificar um nome de usuário com o nome de domínio como "MYDOMAIN\John Smith" e o campo **`group_name`** especifica um único grupo para auditoria.

Uma vez que um único arquivo Nessus **`.audit`** pode especificar vários itens diferentes do cliente, a auditoria das listas de usuários em vários grupos torna-se fácil. Exemplo de política **`.audit`** que permite localizar o grupo "Administrators" para conter somente o usuário "Administrator" e "TENABLE\Domain admins":

```
 <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>
```

Exemplo de imagem de tela da execução do arquivo **`.audit`** acima com um servidor Windows 2003:



USER_GROUPS_POLICY

Uso

```
<custom_item>
 type: USER_GROUPS_POLICY
 description: ["description"]
 value_type: [value type]
 value_data: [value]
 (optional) check_type: [value]
 user_name: ["user name"]
</custom_item>
```

Este item da política verifica se um usuário do Windows pertence aos grupos especificados em **value_data**. Ao usar esta auditoria, é possível verificar apenas usuários de domínio em relação a um controlador de domínio. Esta verificação não se aplica a usuários incorporados, como “Serviço Local”.

Exemplo:

```
 <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

Uso

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

Este item de política verifica os seguintes valores definidos em “Configurações de segurança -> Diretivas locais -> Atribuição de direitos do usuário”. A verificação é executada ao acessar a função **LsaEnumerateAccountsWithUserRight** no identificador da diretiva de LSA.

O campo **right_type** corresponde ao direito de teste. Os valores permitidos são:

```
right_type: RIGHT
```

Onde **RIGHT** pode ser:

```
SeAssignPrimaryTokenPrivilege
SeAuditPrivilege
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
```

O tipo permitido é:

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

Os testes de direitos do usuário executam muitas solicitações no controlador de domínio. Os testes devem ser incluídos em um arquivo de política separado e devem ser iniciados somente em relação ao controlador de domínio e apenas UM sistema no domínio.

Não use aspas para delimitar o tipo de **`right`**, pois a sintaxe é analisada como um token.

Exemplo:

```
<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_CHECK

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

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

Este item da política verifica se o arquivo (**`value_data`**) existe ou não (**`file_option`**). A verificação é executada ao acessar a função `CreateFile`.

Os tipos permitidos são:

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

Exemplos:

```
 <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

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<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>
```

Este item da política verifica se a versão do arquivo especificado pelo campo **`file`** é superior ou igual à versão do arquivo remoto padrão. A verificação pode também ser usada para determinar se a versão do arquivo remoto é inferior com o uso da opção **`check_type`**.

Os tipos permitidos são:

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

Exemplos:

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

FILE_PERMISSIONS

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<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>
```

Esta política verifica se a ACL FILE_PERMISSIONS está correta. A verificação é executada ao acessar a função `GetSecurityInfo` com o nível 7 na alça do arquivo.

O tipo permitido é:

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

Os seguintes caminhos predefinidos podem ser usados no nome do arquivo/pasta:

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

Ao usar esta auditoria, observe o seguinte:

- O campo **`file`** deve incluir o caminho completo até o nome do arquivo ou pasta (por exemplo: **`C:\WINDOWS\SYSTEM32`**) ou usar as palavras-chave de caminho acima. Se as palavras-chave do caminho forem usados, o registro remoto deverá ser ativado para permitir que o Nessus determine os valores de variável de caminho.
- O campo **`value_data`** é o nome de uma ACL definida no arquivo de política.
- O campo **`acl_option`** pode ser definido como CAN_BE_NULL ou CAN_NOT_BE_NULL para forçar sucesso/erro se o arquivo não existir.

Exemplos:

```
 <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>
```

Ao executar a verificação acima, o módulo de conformidade verificará se as permissões definidas para "**%SystemRoot%\SYSTEM32**" coincidem com as descritas no ACL1 file_acl.

FILE_AUDIT

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<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>
```

Este item da política é usado para verificar as propriedades da auditoria (Propriedades –> Segurança –> Avançado –> Auditoria) de um arquivo ou pasta com o uso do ACL especificado. Esta verificação é executada ao acessar a função `GetSecurityInfo` com o nível SACL_SECURITY_INFORMATION no identificado do arquivo.

O tipo permitido é:

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

Os seguintes caminhos predefinidos podem ser usados no nome do arquivo/pasta:

```
%allusersprofile%
%windir%
%systemroot%
```

```
%commonfiles%
%programfiles%
%systemdrive%
%systemdirectory%
```

Ao usar esta auditoria, observe o seguinte:

- O campo **`file`** deve incluir o caminho completo até o nome do arquivo ou pasta (por exemplo: **`C:\WINDOWS\SYSTEM32`**) ou usar as palavras-chave de caminho acima. Se as palavras-chave do caminho forem usados, o registro remoto deverá ser ativado para permitir que o Nessus determine os valores de variável de caminho.
- O campo **`value_data`** é o nome do ACL definido no arquivo de política.
- O campo **`acl_option`** pode ser definido como CAN_BE_NULL ou CAN_NOT_BE_NULL para forçar sucesso/erro se o arquivo não existir.
- Os campos **`acl_allow`** e **`acl_deny`** correspondem a eventos de auditoria “Successful” e “Failed”.

No exemplo a seguir, o arquivo **`.audit`** implementa a função FILE_AUDIT e inclui uma amostra de regra da lista de controle de acesso denominada “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

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<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]
 (optional) avoid_floppy_access
</custom_item>
```

Este item da política verifica se o arquivo contém a expressão regular **`regex`** e se esta expressão coincide com **`expect`**.

A verificação é executada ao acessar a função **`ReadFile`** no identificador do arquivo.

O arquivo é lido por meio de SMB em um buffer de memória no servidor Nessus e, em seguida, o buffer é processado para verificar se há conformidade ou não conformidade. Os arquivos não são salvos no disco do servidor Nessus, mas são copiados em um buffer de memória para análise.

O tipo permitido é:

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

Os seguintes caminhos predefinidos podem ser usados no nome do arquivo/pasta:

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

Ao usar este tipo de auditoria, observe o seguinte:

- O campo `value_data` deve incluir o caminho completo até o nome do arquivo ou pasta (por exemplo: **`C:\WINDOWS\SYSTEM32`**) ou usar as palavras-chave de caminho acima. Se as palavras-chave do caminho forem usados, o registro remoto deverá ser ativado para permitir que o Nessus determine os valores de variável de caminho.
- O campo **`regex`** verifica se há um item presente no arquivo.
- O campo **`expect`** verifica se o item coincide com a expressão regular.
- O campo **`file_option`** pode ser definido como CAN_BE_NULL para forçar um sucesso se o arquivo não existir.
- O campo **`file_option`** poderá ser definido como CAN_NOT_BE_NULL para forçar um erro se o arquivo existir e estiver vazio.

- O campo **avoid_floppy_access** pode ser definido para instruir a auditoria a não executar uma verificação que resulte no acesso à unidade de disquete. A opção deverá ser usada se uma auditoria estiver fazendo com que a unidade de disquete seja acessada quando não existir nenhum disco na unidade.

Exemplo:

```
<custom_item>
 avoid_floppy_access
 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>
```

FILE_CONTENT_CHECK_NOT

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<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>
```

Este item da política verifica se o arquivo contém a expressão regular regex e se a expressão coincide com o campo expect. A verificação é executada ao acessar a função **ReadFile** no identificador do arquivo.

O tipo permitido é:

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

Os seguintes caminhos predefinidos podem ser usados no nome do arquivo/pasta:

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

Ao usar este tipo de auditoria, observe o seguinte:

- O campo **value_data** deve incluir o caminho completo até o nome do arquivo ou pasta (por exemplo: **C:\WINDOWS\SYSTEM32**) ou usar as palavras-chave de caminho acima. Se as palavras-chave do caminho forem usadas, o registro remoto deverá ser ativado para permitir que o Nessus determine os valores de variável de caminho.
- O campo **regex** verifica se há um item presente no arquivo.
- O campo **expect** verifica se o item coincide com a expressão regular.
- O campo **file_option** pode ser definido como CAN_BE_NULL para forçar um sucesso se o arquivo não existir.
- O campo **file_option** poderá ser definido como CAN_NOT_BE_NULL para forçar um erro se o arquivo existir e estiver vazio.

Exemplo:

```
<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>
```

REG_CHECK

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<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>
```

Este item da política verifica se a chave (ou o item) do registro existe ou não. A verificação é executada ao acessar a função **RegOpenKeyEx** e **RegQueryValueEx**.

Os tipos permitidos são:

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

Se o campo **key_item** não for especificado, este item verificará se existe o caminho da chave. Do contrário, verificará se o item existe.

Exemplo:

```
 <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

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<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>
```

Este item da política é usado para a verificação do valor de uma chave do registro. Muitas verificações da política em “Configurações de segurança -> Diretivas locais -> Opções de segurança” usam este item da política. Esta verificação é executada ao acessar a função **RegQueryValueEx**.

O campo **reg_key** é o nome da chave do registro (por exemplo: “HKLM\SOFTWARE\Microsoft\Driver Signing”). A primeira parte de chave (HKLM) é usada para conexão à seção correta do registro. O caminho seguinte é uma designação onde está localizado o **reg_item** desejado.

A seção HKU (HKEY_USERS) é um caso especial. Não é possível especificar um SID para chaves HKU. Sabe-se que o **nbin** é repetido internamente para cada SID e só será aprovado se o valor em cada SID for válido.

Por exemplo:

```
<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>
```

entraria em loop:

```
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
...
```

e é aprovado se o item “ScreenSaveActive” for definido como 1 para todos os SIDs.

O campo opcional **reg_option** pode ser definido como CAN_BE_NULL para forçar a execução da verificação se a chave não existir ou como o CAN_NOT_BE_NULL oposto.

A opção **reg_enum** adicional com o argumento “ENUM_SUBKEYS” pode ser usada para enumerar um valor especificado para todas as subchaves da chave de um registro. Por exemplo, a chave: **HKLM\SOFTWARE\Microsoft\Windows\CurrentVersion\Uninstall** possui muitos pacotes de software listados. Para que o valor “CurrentVersion” coincida com todas as subchaves sob “Uninstall”, use **reg_enum**.

Exemplo:

```
<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>
```

Esta auditoria da seção do registro HKU não inclui o SID (identificador de segurança) no caminho de registro **reg_key**. Este exemplo pesquisará todo SID HKU quanto ao **reg_item** especificado.

Exemplo:

```
<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>
```

Estão disponíveis os seguintes tipos de campo **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]

Os seguintes tipos de campo **value_type** opcionais estão disponíveis e são usados nos itens predefinidos:

- 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
  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`

Para o campo **`custom_item`**, use o **`value_type`** principal. Foram criados tipos opcionais de itens predefinidos.

Se o **`value_type`** for uma ACL, o item do registro deve ser uma descrição de segurança em formato binário.

Exemplos:

```
<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>
```

REGISTRY_PERMISSIONS

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<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>
```

Esta política verifica se a ACL da chave do registro está correta. A verificação é executada ao acessar a função **`RegGetKeySecurity`** na chave do registro.

O tipo permitido é:

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

Podem ser usados os seguintes caminhos predefinidos para o campo **`reg_key`**:

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

Ao usar esta auditoria, observe o seguinte:

- O campo **`reg_key`** deve incluir o caminho completo até a chave do registro do arquivo.
- O campo **`value_data`** é o nome de uma ACL definida no arquivo de política.

- O campo **acl_option** pode ser definido como CAN_BE_NULL ou CAN_NOT_BE_NULL para forçar um sucesso/erro se a chave não existir.

Exemplo:

```
<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>
```

Ao executar a verificação acima, o módulo de conformidade verificará se as permissões definidas para "**HKLM\SOFTWARE\Microsoft**" coincidem com as descritas no ACL2 registry_acl.

REGISTRY_AUDIT

Esta verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

```
<custom_item>
 type: REGISTRY AUDIT
 description: ["description"]
 value_type: [value_type]
 value_data: [value]
 reg_key: ["regkeyname"]
 (optional) acl option: [acl option]
</custom_item>
```

Esta política verifica se a ACL da chave do registro está correta. A verificação é executada ao acessar a função **RegGetKeySecurity** na chave do registro.

O tipo permitido é:

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

Podem ser usados os seguintes caminhos predefinidos para o campo **reg_key**:

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

Ao usar esta auditoria, observe o seguinte:

- O campo **reg_key** deve incluir o caminho completo até a chave do registro do arquivo.
- O campo **value_data** é o nome do ACL definido no arquivo de política.
- O campo **acl_option** pode ser definido como CAN_BE_NULL ou CAN_NOT_BE_NULL para forçar um sucesso/erro se a chave não existir.
- Os campos **acl_allow** e **acl_deny** correspondem a eventos de auditoria “Successful” e “Failed”.

No exemplo a seguir, o arquivo **.audit** audita a chave do registro de “HKLM\SOFTWARE\Microsoft” ao compará-la com uma lista de controle de acesso denominada “ACL2” (não mostrada):

```
 <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>
```

REGISTRY_TYPE

Essa verificação requer acesso remoto ao registro para que o sistema Windows funcione corretamente.

Uso

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

Esse item da política é usado para a verificação do valor de um tipo de chave do registro. Essa verificação é executada acessando-se a função RegQueryValue.

O campo reg_key é o nome da chave do registro (“HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon”). A primeira parte de chave (HKLM, HKU, HKCU, ...) é usada para conexão à seção correta do registro. Na maioria dos casos, o campo reg_key requer uma entrada de registro estática, sem curingas. Porém, uma exceção é permitida ao pesquisar valores dentro de HKU (HKEY_USERS). Se um caminho for designado sob HKU, a pesquisa itera todos os valores de usuário em HKU para o valor no caminho designado. Por exemplo, se reg_key: "HKU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run" for especificado junto com reg_item “brastk”, todos os usuários em HKU serão pesquisados para o valor da chave de registro “brastk” sob o caminho relativo: “HKU\<user_id>\SOFTWARE\Microsoft\Windows\CurrentVersion\Run”. Por exemplo:

```
value_type: POLICY_TEXT
reg_key: "HKU\SOFTWARE\Microsoft\Windows\CurrentVersion\Run"
reg_item: "brastk"
value_data: "C:\WINDOWS\System32\brastk.exe"
```

Essa verificação pesquisa em:

HKU\S-1-5-18\SOFTWARE\Microsoft\Windows\CurrentVersion\Run
HKU\S-1-5-19\SOFTWARE\Microsoft\Windows\CurrentVersion\Run

O campo opcional reg_option pode ser definido como CAN_BE_NULL para forçar a execução da verificação se a chave não existir ou como o CAN_NOT_BE_NULL oposto.

Somente POLICY_TEXT value_type está disponível para essa verificação.

Aqui está um exemplo de arquivo .audit que audita o tipo de registro “HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon”:

```
<custom_item>
 type: REGISTRY_TYPE
 description: "Check type - reg_sz"
 value_type: POLICY_TEXT
 value_data: "reg_sz"
 reg_key: "HKLM\Software\Microsoft\Windows NT\CurrentVersion\Winlogon"
 reg_item: "ScreenSaverGracePeriod"
</item>
```

SERVICE_PERMISSIONS

Uso

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

Esta política verifica se a ACL de serviço está correta. A verificação é executada ao acessar a função `QueryServiceObjectSecurity` no identificador do serviço.

O tipo permitido é:

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

Ao usar esta auditoria, observe o seguinte:

- O campo **`value_data`** é o nome de uma ACL definida no arquivo de política.
- O campo **`acl_option`** pode ser definido como CAN_BE_NULL ou CAN_NOT_BE_NULL para forçar um sucesso/erro se a chave não existir.

Exemplo:

```
<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>

 <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>
```

Ao executar a verificação acima, o módulo de conformidade verificará se as permissões definidas para o serviço de alerta coincidem com as descritas no ACL3 service_acl.

SERVICE_AUDIT

Uso

```
<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>
```

Esta política verifica se a ACL de serviço está correta. A verificação é executada ao acessar a função `QueryServiceObjectSecurity` no identificador do serviço.

O tipo permitido é:

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

Ao usar este tipo de auditoria, observe o seguinte:

- O campo **`value_data`** é o nome do ACL definido no arquivo de política.
- O campo **`acl_option`** pode ser definido como CAN_BE_NULL ou CAN_NOT_BE_NULL para forçar um sucesso/erro se a chave não existir.
- Os campos **`acl_allow`** e **`acl_deny`** correspondem a eventos de auditoria “Successful” e “Failed”.

Aqui é apresentado um exemplo de arquivo **`.audit`** para auditoria do serviço “Alerter”:

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

WMI_POLICY

Uso

```
<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>
```

Esta verificação consulta o banco de dados WMI do Windows para localizar valores especificados dentro do intervalo de nomes/classes/atributos.

Qualquer um dos dois valores de chave pode ser extraído ou os nomes de atributo podem ser enumerados, dependendo da sintaxe usada.

Os tipos permitidos são:

```
wmi_namespace: "namespace"
wmi_request: "WMI Query"
wmi_attribute: "Name"
wmi_key: "Name"
wmi_option: option
wmi_exclude_result: "result"
only_show_query_output: YES
check_type : CHECK_NOT_REGEX
```

Se a configuração de serviço com valores duplicados no sistema for selecionada (por exemplo: MSFTPSVC/83207416" e "MSFTPSVC/2"), a solicitação extrairá o atributo de ambos. Caso um deles não corresponda ao valor da política, o campo **`wmi_key`** será adicionado ao relatório para indicar qual falhou. O campo **`wmi_enum`** permite enumerar os nomes de configuração dentro de um espaço de nomes para comparação ou para verificação do valor da política. Veja os exemplos abaixo.

Por padrão, se uma consulta WMI não retorna nenhuma saída, a verificação reporta um erro. Esse comportamento pode ser alterado e a verificação pode ser forçada a reportar PASS se wmi_option for definida como **`CAN_BE_NULL`**. Ao configurar **`only_show_query_output`** como YES, a saída da consulta WMI agora é incluída no relatório do Nessus. Usando a tag **`check_type`**, você pode ter um resultado PASS desde que um determinado string não exista na saída.

Outras considerações:

- Os atributos de WMI precisam ser especificados explicitamente. Por exemplo, **`select * from foo`** não funcionará.
- Os atributos que não têm valor definido não serão reportados.
- O caso dos atributos deve ser exatamente como aparece na documentação da Microsoft. Por exemplo, o atributo **`HandleCount`** não pode ser **`handlecount`** ou handlecount.
- Os valores de tipo de matriz não são incluídos no resultado.

Exemplo 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>
```

Se houver duas configurações de serviço FTP no sistema (“MSFTPSVC/83207416” e “MSFTPSVC/2”), a solicitação extrairá o atributo “UserIsolationMode” de ambas. Caso um deles não corresponda ao valor da política (0), neste caso, o campo **`wmi_key`** será adicionado ao relatório, indicando qual falhou.

Exemplo 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>
```

Este exemplo verifica se há dois nomes de configuração válidos, conforme especificado em **`value_data`**. Para mais informações sobre o espaço de nomes WMI e os respectivos atributos, o Microsoft WMI CIM Studio é uma valiosa ferramenta disponível no seguinte link: http://www.microsoft.com/downloads/details.aspx?FamilyID=6430f853-1120-48db-8cc5-f2abdc3ed314&displaylang=en.

Exemplo 3:

```
<custom_item>
 type: WMI_POLICY
 description: "List All Windows Processes - except svchost.exe and iPodService.exe"
 value_type: POLICY_TEXT
 value_data: ""
 wmi_namespace: "root/cimv2"
 wmi_exclude_result: "svchost.exe,iPodService.exe"
 wmi_request: "select Caption,HandleCount,ThreadCount from Win32_Process"
 only_show_query_output: YES
</custom_item>
```

Esse exemplo listará todos os processos do Windows, mas removerá instâncias de svchost.exe e **`iPodService.exe`**.

Itens

"Items" (itens) são tipos de verificação predefinidos no Windows Compliance Checks Engine. São usados para itens auditados com frequência e reduzem a sintaxe exigida para a criação de verificação de auditoria. Um item possui a seguinte estrutura:

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

O campo **name** deve ter um nome já definido (a tabela "Políticas predefinidas" abaixo relaciona os nomes predefinidos).

Todos os itens predefinidos correspondem à lista disponível no Editor de Políticas de Domínio do Windows 2003 SP1.

O exemplo abaixo verifica se o comprimento mínimo da senha está entre 8 e 14 caracteres:

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

O item personalizado correspondente é:

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

Políticas predefinidas

Política	Uso
Diretivas de senha	name: "Enforce password history" 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
Diretiva de bloqueio de conta	name: "Account lockout duration" value: TIME_MINUTE or name: "Account lockout duration"

	value: TIME_SECOND name: "Account lockout threshold" value: POLICY_DWORD name: "Reset lockout account counter after" value: TIME_MINUTE name: "Enforce user logon restrictions" value: POLICY_SET
Diretiva do 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
Diretiva de auditoria	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
Contas	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 name: "Accounts: Rename administrator account" value: POLICY_TEXT name: "Accounts: Rename guest account" value: POLICY_TEXT
Auditoria	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
Dispositivos	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
Controlador de domínio	name: "Domain controller: Allow server operators to schedule tasks" value: POLICY_SET name: "Domain controller: LDAP server signing requirements" value: LDAP_SET name: "Domain controller: Refuse machine account password changes" value: POLICY_SET

Membro do Domínio	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
Logon Interativo	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 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

Cliente de Rede Microsoft	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
Servidor de Rede Microsoft	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
Acesso à rede	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" 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
Segurança de rede	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
Console de Recuperação	name: "Recovery console: Allow automatic administrative logon" value: POLICY_SET name: "Recovery console: Allow floppy copy and access to all drives and all folders" value: POLICY_SET
Desligamento	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
Criptografia do Sistema	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
Objetos do Sistema	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
Configurações do sistema	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
Log de eventos	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" 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

Forçar geração de relatórios

As políticas de auditoria podem ser forçadas a produzir um resultado específico com o uso da palavra-chave “**report**”. Os tipos de relatório Passed (Aprovado), Failed (Reprovado) e Warning (Advertência) podem ser usados. Um exemplo de política é apresentado a seguir:

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

O texto dentro do campo “**description**” deve ser sempre exibido no relatório.

Este tipo de relatório permite informar a um auditor que uma auditoria real executada pelo Nessus não pode ser concluída. Por exemplo: pode haver uma exigência para determinar que um sistema específico está protegido fisicamente e que é preciso solicitar que o auditor execute a verificação manualmente. Este tipo de relatório também será usado se o tipo específico de auditoria necessária a ser executada pelo Nessus não for determinado com uma verificação OVAL.

Condições

É possível definir a lógica **if/then/else** na política do Windows para que seja iniciada uma verificação somente se as precondições forem válidas ou para agrupar vários testes em apenas um.

A sintaxe para executar as condições é a seguinte:

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

As condições podem ser do tipo “**and**” ou “**or**”.

Na auditoria da condição, as instruções “**then**” e “**else**” podem ser uma lista de itens (ou itens personalizados) ou uma instruções “**if**”. As instruções “**else**” e “**then**” também usam o tipo “**report**” para relatar um sucesso ou uma falha, dependendo do valor de retorno da condição:

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

O valor “**if**” retorna SUCCESS or FAILURE e é usado quando a instrução “**if**” estiver dentro de outra estrutura “**if**”. Por exemplo: se a estrutura **<then>** for executada, o valor de retorno será um dos seguintes:

- a auditoria contém somente itens: retornará SUCCESS se todos os itens “else” aprovados retornarem FAILURE
- a auditoria contém somente <**report**>: retornará o tipo de relatório
- a auditoria contém os dois itens e <**report**>: retornará o tipo de relatório

Se a expressão **<report>** for usada e o tipo for “FAILED”, o motivo da reprovação será exibido no relatório junto com um nível de severidade, se definido.

O exemplo de auditoria da política de senhas é mostrado a seguir. Uma vez que o tipo “**and**” é usado, a auditoria de ambos os itens personalizados deve ser aprovada para que a política também seja aprovada. Este exemplo verifica a combinação incomum de políticas de histórico de senhas para ilustrar o grau de sofisticação que uma lógica de teste pode atingir:

```
<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
 </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>
```

No exemplo acima, somente o novo tipo "**report**" é mostrado, mas a estrutura **if/then/else** aceita a execução de auditorias adicionais com as cláusulas "**else**". Com base em uma condição, as cláusulas **if/then/else** aninhadas também podem ser usadas. Observe um exemplo mais complexo a seguir:

```
<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>
   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>
```

Neste exemplo, se a conta Administrator não tiver sido renomeada, deve-se auditar se o tempo de vida mínimo da senha é de 30 dias ou menos. Este política de auditoria será aprovada se a conta do administrador for renomeada, independentemente da política de senhas, e verificará a política de tempo de vida da senha somente se a conta do administrador não for renomeada.

Referência de arquivo de conformidade de auditoria para conteúdo do Windows

As verificações **`.audit`** do Windows Content diferenciam-se das verificações **`.audit`** do Windows Configuration **`.audit`**, pois foram projetadas para buscar um sistema de arquivos do Windows para tipos específicos de arquivo que contenham dados confidenciais em vez de enumerar definições da configuração do sistema. Incluem uma variedade de opções para ajudar o auditor a limitar os parâmetros, localizar e exibir dados não conformes de forma mais eficaz.

Uso de aspas:

As aspas simples e aspas duplas são intercambiáveis quando estiverem delimitando campos de auditoria, exceto nos dois casos abaixo:

1. Nas verificações de conformidade do Windows em que os campos especiais como CRLF etc. devem ser interpretados literalmente, use aspas simples. Todos os campos incorporados interpretados como strings devem ser protegidos.

Por exemplo:

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

2. É necessário o uso de aspas ao usar “include_paths” e exclude_paths do WindowsFiles.

Se os strings forem usados em qualquer tipo de campo (descrição, value_data, regex etc.) que contém aspas simples ou aspas duplas, proceda da seguinte maneira:

a. Use o tipo oposto de aspa para as aspas delimitadoras mais afastadas.

Por exemplo:

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

b. Isole as aspas incorporadas com uma barra invertida (somente aspas duplas).

Por exemplo:

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

Tipo de verificação

Todas as verificações de conformidade do conteúdo do Windows devem ser delimitadas com o encapsulamento **`check_type`** e a designação "**`WindowsFiles`**". Isto é bastante parecido com todos os outros arquivos **`.audit`**. O formato básico de um arquivo de verificação de conteúdo é o seguinte:

```
<check_type: "WindowsFiles">
<item>
</item>
<item>
</item>
<item>
</item>
</check_type>
```

As verificações reais de cada item não estão mostradas. As seções abaixo mostram como várias palavras-chave e parâmetros podem ser usados para preencher uma auditoria de item de conteúdo específico.

Formato dos itens

Uso

```
<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>
```

Cada um desses itens pode ser usado para auditar uma grande variedade de formatos de arquivo, com uma grande variedade de tipos de dados. A tabela abaixo apresenta uma lista de tipos de dados aceitos. A seção seguinte possui vários exemplos de como as palavras-chave podem ser combinadas para auditar vários tipos de conteúdo de arquivo.

Palavra-chave	Descrição
type	Deverá ser sempre definida como FILE_CONTENT_CHECK
description	Esta informação deve usada como título de vulnerabilidades de conformidade exclusivo do SecurityCenter. Será também o primeiro conjunto de dados relatado pelo Nessus.
file_extension	Apresenta uma lista de todas as extensões desejadas a serem pesquisadas pelo Nessus. As extensões estão listadas com ".", entre aspas e separadas por barras verticais. Se as opções adicionais, como **regex** e **expect**, não forem incluídas na auditoria, os arquivos com file_extension especificado serão exibidos no resultado da auditoria.
regex	Esta palavra-chave contém a expressão regular usada para a pesquisa de tipos de dados complexos. Se a expressão regular for coincidente, o primeiro conteúdo coincidente será exibido no relatório de vulnerabilidade.  A palavra-chave **regex** deve ser executada com a palavra-chave **expect** descrita abaixo.  Ao contrário das Windows Compliance Checks, o Windows File Content Compliance Check **regex** e **expect** não possuem o(s) mesmo(s) string(s) de dados no arquivo pesquisado. As verificações do Windows File Content requerem que tanto a instrução **regex** como **expect** coincidam com os dados dentro dos bytes **<max_size>** do arquivo pesquisado.
expect	A instrução **expect** é usada para relacionar um ou mais padrões simples que devem estar contido no documento para que coincida. Por exemplo: ao pesquisar números de seguridade social, a palavra "SSN", "SS#" ou "Social" pode ser exigida. Vários padrões são mantidos entre aspas e separados com barras verticais. A correspondência de padrão simples também é aceita nesta palavra-chave com o ponto. Ao coincidir com o string "C.T", a declaração **expect** coincidirá com "CAT", "CaT", "COT", "C T" e assim por diante.  A palavra-chave **expect** pode ser executada de forma independente. No entanto, se a palavra-chave **regex** for usada, **expect** será obrigatório.  Ao contrário das Windows Compliance Checks, o Windows File Content Compliance Check **regex** e **expect** não possuem o(s) mesmo(s) string(s) de dados no arquivo pesquisado. As verificações do Windows File Content requerem que tanto a instrução **regex** como **expect**

	coincidam com os dados dentro dos bytes **<max_size>** do arquivo pesquisado.
file_name	Considerando que a palavra-chave **file_extension** é obrigatória, esta palavra-chave pode aprimorar ainda mais a lista de arquivos e serem analisados. Ao fornecer uma lista de padrões, os arquivos podem ser descartados ou coincidentes. Esta ação, por exemplo, facilita a pesquisa de qualquer tipo de nome de arquivo que possua termos como “funcionário”, “cliente” ou “salário”.
max_size	Para melhor desempenho, uma auditoria deve examinar apenas a primeira parte de cada arquivo. Isto pode ser especificado em bytes com esta palavra-chave. O número de bytes pode ser usado como um argumento. Além disso, aceita-se a extensão “K” ou “M” de kilobytes ou megabytes, respectivamente.
only_show	Ao coincidir dados confidenciais, como números de cartão de crédito, sua organização poderá exigir que apenas os quatro últimos dígitos fiquem visíveis no relatório. Esta palavra-chave aceita a divulgação de qualquer número de bytes especificado pela política.
regex_replace	Esta palavra-chave controla qual padrão na expressão regular será mostrado no relatório. Se padrões de dados complexos, como números de cartão de crédito, forem pesquisados, nem sempre será possível obter-se a primeira correspondência com os dados desejados. Esta palavra-chave proporciona mais flexibilidade para a captura dos dados desejados com maior exatidão.
include_paths	Esta palavra-chave permite a inclusão de diretório ou unidade dentro dos resultados da pesquisa. A palavra-chave pode ser usada em combinação com a palavra-chave “**exclude_paths**” ou independente dela. Seu uso é importante nos casos em que somente certas unidades ou pastas são pesquisadas em um sistema com várias unidades. Os caminhos são delimitados por aspas duplas e separados por barra vertical quando forem exigidos vários caminhos.  Somente letras de unidade ou nomes de pasta podem ser especificados com a palavra-chave “**include_paths**”. Os nomes de arquivos não podem ser incluídos na sequência de valores “**include_paths**”.
exclude_paths	Esta palavra-chave permite a exclusão de unidade, diretório ou arquivo dos resultados da pesquisa. A palavra-chave pode ser usada junto com a palavra-chave “**include_paths**” ou independente dela. Seu uso é importante nos casos em que uma unidade, diretório ou arquivo específico são excluídos dos resultados da pesquisa. Os caminhos são delimitados por aspas duplas e separados por barra vertical quando forem exigidos vários caminhos.

Exemplos de linha de comando

Nesta seção, será criado um documento de texto falso com uma extensão **.tns** e, com base nele, vários arquivos **.audit** simples a complexos serão executados. Ao percorrer cada exemplo, analisaremos cada caso compatível com os parâmetros do Windows Content.

O binário da linha de comando **nasl** também será usado. Cada um dos arquivos **.audit** mostrados pode ser acrescentado facilmente às políticas de varredura do Nessus 4 ou do SecurityCenter, mas para auditorias rápidas de apenas um sistema, este método é bastante eficiente. O comando a ser executado sempre a partir do diretório **/opt/nessus/bin** será:

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

O <IP> é o endereço IP do sistema que a ser auditado.

Durante a execução do **.nbin** (ou qualquer outro plugin), o Nessus solicitará as credenciais do sistema de destino, além da localização do arquivo **.audit**.

Arquivo de teste de destino

O arquivo de destino a ser usado terá o seguinte conteúdo:

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

Copie os dados e envie-os a qualquer sistema Windows ao qual tiver acesso credenciado. Nomeie o arquivo como "Tenable_Content.tns".

Exemplo 1: Pesquisa de documentos .tns que contêm a palavra "Nessus"

No exemplo a seguir, o arquivo **.audit** simples busca qualquer arquivo **.tns** que contém a palavra "Nessus" em qualquer lugar do documento.

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

Ao ser executado este comando, espera-se a seguinte saída:

```
"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
```

Os resultados mostram que uma correspondência foi encontrada. O relatório informa uma "falha", pois foram encontramos dados que não eram procurados. Por exemplo: ao fazer uma auditoria de um número de seguridade social e houve uma correspondência positiva no computador público, embora a correspondência seja positiva, é registrada como falha por razões de conformidade.

Exemplo 2: Pesquisa de documentos .tns que contêm a palavra "França"

No exemplo a seguir, o arquivo **.audit** simples busca qualquer arquivo **.tns** que contém a palavra "França" em qualquer lugar do documento.

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

A saída obtida desta vez será:

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

A auditoria foi “aprovada” porque nenhum dos arquivos **.tns** auditados continha a palavra “França”.

Exemplo 3: Pesquisa de documentos .tns e .doc que contêm a palavra “Nessus”

O acréscimo de uma segunda extensão para buscas de arquivo de documentos Microsoft Word é bastante fácil e está exemplificado abaixo.

```
<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>
```

Os resultados (em nosso computador de testes) foram os seguintes:

```
"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
```

Ocorreu a mesma “reprovação” com nosso arquivo **.tns** de teste, porém, neste caso, havia um segundo arquivo **.doc**, que também continha a palavra “Nessus”. Se estiver executando os testes em seus próprios sistemas, é possível localizar ou não um arquivo Word com a palavra “Nessus”.

Exemplo 4: Pesquisa de documentos .tns e .doc que contêm a palavra “Nessus” e um número com 11 dígitos

Vamos adicionar a primeira expressão regular para coincidir com um número de 11 dígitos. É preciso adicionar apenas a expressão regular com a palavra-chave **regex** ao mesmo arquivo **.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>
```

Este procedimento resulta no seguinte:

```
"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)
```

O arquivo **.doc** com correspondência no último exemplo ainda está sendo pesquisado. Uma vez que não contém o número de 11 dígitos, não será mais exibido. Além disso, observe que, uma vez que a palavra-chave **regex** foi usada, uma correspondência também ocorreu nos dados.

E se for necessário localizar um número de 10 dígitos? O número de 11 dígitos acima contém dois números de 10 dígitos (0123456789 e 1234567890). Para escrever uma correspondência mais exata para apenas 11 dígitos, é preciso obter uma expressão normal que informe:

"Coincidir com qualquer número de 11 dígitos não precedido ou seguido por quaisquer outros números".

Para fazer isso em expressões regulares, podemos adicionar o operador "não", conforme abaixo:

```
<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>
```

A leitura da esquerda para a direita mostra também o caractere "^" e o cifrão algumas vezes. O caractere "^" pode significar, às vezes, o início de uma linha ou a correspondência com o valor negativo. O cifrão significa o fim de uma linha. A expressão regular acima significa basicamente a procura de quaisquer padrões que não iniciam com um número, mas, provavelmente iniciam em uma nova linha, que contenham 11 números e que não sejam seguidos por mais nenhum número ou que tenham um final de linha. As expressões regulares tratam o início e o final de uma linha como casos especiais e, por isso, requerem o uso dos caracteres "^" ou "$".

Exemplo 5: Pesquisa de documentos .tns e .doc que contêm a palavra "Nessus" e um número com 11 dígitos, mas exibem os últimos 4 bytes

O acréscimo da palavra-chave **only_show** ao arquivo **.audit** pode limitar a saída. Isto pode limitar o acesso dos auditores apenas aos dados confidenciais que estiverem procurando.

```
<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>
</check_type>
```

Em caso de correspondência, os dados são ocultados com caracteres “X”, conforme mostrado abaixo:

```
"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)
```

Exemplo 6: Busca de documentos .tns que contêm a palavra “Correlação” nos primeiros 50 bytes

Neste exemplo, examinaremos o uso da palavra-chave **`max_size`**. Em nosso teste, a palavra “Correlação” consumiu mais de 50 bytes no arquivo.

```
<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>
```

Ao executar o teste, obtém-se uma correspondência com aprovação:

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

Altere o valor `max_size` de “50” para “50K” e reexecute a varredura. Obtém-se agora uma mensagem de erro:

```
"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
```

Exemplo 7: Controle dos itens exibidos na saída

Neste exemplo, examinaremos o uso da palavra-chave **`regex_replace`**. Considere o arquivo **`.audit`** a seguir:

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

Esta verificação produz o seguinte:

```
"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)
```

No entanto, vejamos o que pode ocorrer se for necessário uma expressão regular para coincidir com as partes “Passive” e “Scanner”, mas o usuário deseja que somente a parte “Vulnerability” retorne. Uma nova expressão regular seria parecida com a seguinte:

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

A verificação também retorna a correspondência inteira de “Passive Vulnerability Scanner”, pois a declaração da expressão regular trata o string inteiro como a primeira correspondência. Para obtermos apenas a segunda correspondência, é preciso incluir a palavra-chave **`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>
```

A saída da varredura é a seguinte:

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

Usa-se “\3” para indicar o segundo item da correspondência, pois o primeiro (“\1”) é o string inteiro. Se “\2” fosse usado, “Passive” deveria retornar e, se “\4” fosse usado, “Scanner” deveria retornar.

Qual é a finalidade deste recurso? Se padrões de dados complexos, como números de cartão de crédito, forem pesquisados, nem sempre será possível obter-se a primeira correspondência com os dados desejados. Esta palavra-chave proporciona mais flexibilidade para capturar os dados desejados com maior exatidão.

Exemplo 8: Uso do nome do arquivo como filtro

Se levarmos em conta o arquivo **`.audit`** do terceiro exemplo, ele retorna um resultado de um arquivo **`.tns`** ou **`.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>
```

Os resultados (em nosso computador de testes) foram os seguintes:

```
"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
```

Pode-se usar também a palavra-chave **`file_name`** para filtrar arquivos desejado ou não. Ao adicioná-los ao arquivo **`.audit`** e solicitar que considere apenas arquivos com “tenable” no nome, a saída se parece com a seguinte:

```
<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>
```

A saída é a seguinte:

```
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
```

O arquivo **`.doc`** coincidente não está presente porque não tem a palavra “tenable” no caminho.

Uma vez que o string coincidente é uma expressão regular, pode ser flexível para corresponder a uma variedade de arquivos desejados ou não. Por exemplo: pode-se usar o string “[Tt]enable” para corresponder à palavra “Tenable” ou “tenable”. Da mesma forma, para existir uma correspondência com uma extensão ou extensão parcial, é preciso remover o ponto com uma barra invertida, como “\.t”, para pesquisar quaisquer extensões que comecem com “t”.

Exemplo 9: Uso das palavras-chave de inclusão/exclusão

As palavras-chave “**`include_paths`**” e “**`exclude_paths`**” para filtrar as buscas com base na letra da exclusão da unidade, do diretório ou do nome do arquivo.

```
<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>
```

A saída é a seguinte:

```
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
```

Observe que a saída não difere do resultado de uma busca padrão de conteúdo de arquivo Windows, porém exclui o caminho excluído. Se um único caminho com o uso de "**include_paths**" for incluído (por exemplo: "**c:**"), todos os outros caminhos serão automaticamente excluídos. Além disso, se uma letra de unidade for excluída (por exemplo: "**d:**"), mas se uma pasta nessa unidade estiver incluída (por exemplo: "**d:\users**"), a palavra-chave "**exclude_paths**" terá preferência e a unidade não será pesquisada. No entanto, é possível incluir uma unidade **C:**e excluir uma subpasta na unidade (por exemplo:**C:\users:**).

Auditoria de diferentes tipos de formatos de arquivo

Qualquer extensão de arquivo poderá ser auditada. No entanto, arquivos como **.zip** e **.gz** não são descompactados durante o processo. Caso o arquivo seja compactado ou contenha algum tipo de codificação de dados, a busca de padrão pode ser impossibilitada.

Para documentos que armazenam dados no formato Unicode, as rotinas de análise sintática do arquivo **.nbin** extrairão todos os bytes "NULL" encontrados.

Além disso, é compatível com todas as versões de documentos do Microsoft Office. Isto inclui as recentes versões codificadas adicionadas com o Office 2007, como **.xlsx** e **.docx.**

O suporte para vários tipos de formatos de arquivo PDF também foi acrescentado. A Tenable desenvolveu um extenso analisador de PDF que extrai strings não processados para correspondência. Os usuários devem se preocupar apenas com que tipo de dados que desejam visualizar em um arquivo PDF.

Considerações sobre desempenho

Uma organização pode enfrentar diversos dilemas ao modificar os arquivos **.audit** padrão e testá-los em redes ativas:

- Quais extensões devem ser pesquisadas?
- Qual é a quantidade de dados a ser examinada?

Os arquivos **.audit** não querem a palavra-chave **max_size**. Neste caso, o Nessus tenta recuperar o arquivo inteiro e prossegue, a menos que tenha uma correspondência em um padrão. Uma vez que os arquivos percorrem a rede, há mais tráfego de rede com essas auditorias do que com a auditoria de varredura ou configuração típica.

Se vários scanners Nessus forem gerenciados pelo SecurityCenter, os dados devem ser transferidos do host Windows ao scanner que estiver realizando a auditoria de vulnerabilidades.

Referência de arquivo de conformidade de auditoria para configuração do Cisco IOS

Esta seção descreve o formato e as funções das verificações de conformidade do Cisco IOS e o fundamento lógico por trás de cada configuração.

Uso de aspas:

As aspas simples e aspas duplas são intercambiáveis quando estiverem delimitando campos de auditoria, exceto nos dois casos abaixo:

1. Nas verificações de conformidade do Windows em que os campos especiais como CRLF etc. devem ser interpretados literalmente, use aspas simples. Todos os campos incorporados interpretados como strings devem ser protegidos.

Por exemplo:

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

2. É necessário o uso de aspas ao usar “include_paths” e exclude_paths do WindowsFiles ”.

Se os strings forem usados em qualquer tipo de campo (descrição, value_data, regex etc.) que contém aspas simples ou aspas duplas, proceda da seguinte maneira:

a. Use o tipo oposto de aspa para as aspas delimitadoras mais afastadas.

Por exemplo:

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

b. Isole as aspas incorporadas com uma barra invertida (somente aspas duplas).

Por exemplo:

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

Tipo de verificação

Todas as verificações de conformidade do conteúdo do Cisco IOS devem ser delimitadas por colchetes com o encapsulamento **`check_type`** e a designação “Cisco”. Isto é necessário para diferenciar arquivos **`.audit`** destinados a sistemas que executarem o sistema operacional Cisco IOS a partir de outros tipos de auditorias de conformidade.

Exemplo:

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

Ao contrário de outros tipos de auditoria de conformidade, nenhuma palavra-chave adicional de tipo ou versão está disponível.

Palavras-chave

A tabela a seguir indica como poderá ser usada cada palavra-chave nas verificações de conformidade do Cisco:

Palavra-chave	Exemplo de uso e configurações aceitas
type	CONFIG_CHECK, CONFIG_CHECK_NOT e RANDOMNESS_CHECK "CONFIG_CHECK" determina se o item especificado existe na saída "show config" do Cisco IOS. Do mesmo modo, "CONFIG_CHECK_NOT" determina se o item especificado existe ou não. "RANDOMNESS_CHECK" é usado para a realização de verificações da complexidade de strings (por exemplo: verificações de senha). Se um item de busca for especificado (por meio de regex), informará se o string é "aleatório" o bastante (pelo menos oito caracteres, caixa alta, caixa baixa, pelo menos um dígito e um caractere especial).  Atualmente, os parâmetros de aleatoriedade não são configuráveis.
description	A palavra-chave "**description**" permite adicionar uma breve descrição da verificação que estiver sendo realizada. Não podemos deixar de recomendar que o campo **description** seja exclusivo e que nenhuma verificação distinta tenha o mesmo campo **description**. O SecurityCenter da Tenable usa este campo para a geração automática de um número exclusivo de identificação do plugin com base no campo **description**. Exemplo: `description: "Forbid Remote Startup Configuration"`

feature_set	A palavra-chave "**feature_set**", da mesma forma que a palavra-chave "**system**" em verificações de conformidade do Unix, verifica a versão do Feature Set do Cisco IOS e executa a verificação resultante ou ignora a verificação devido a um regex com falha. É usada nos casos em que a verificação é aplicável apenas a sistemas com um Feature Set específico.
	
	Exemplo:
	`<item>`
	`type: CONFIG_CHECK`
	`description: "Version Check"`
	`info: "SSH Access Control Check."`
	`feature_set: "K8" context:"line .*"`
	`item: "access-class [0-9]+ in"`
	`</item>`
	
	A verificação acima executará apenas a verificação do "item" se a versão do Feature Set corresponder ao regex especificado: (K8)
	
	Caso haja uma falha de verificação da versão do Feature Set, será exibida uma mensagem de erro parecida com a seguinte:
	
	`"Version Check" : [SKIPPED]`
	`Test defined for the feature set 'K8' whereas we are running C850-ADVSECURITYK9-M`

ios_version	A palavra-chave "**ios_version**", da mesma forma que a palavra-chave "**system**" em verificações de conformidade do Unix, verifica a versão do Cisco IOS e executa a verificação resultante ou ignora a verificação devido a um regex com falha. É usada nos casos em que a verificação é aplicável apenas a sistemas com uma versão IOS

	específica.
	
	Exemplo:
	`<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>`
	
	A verificação acima executará apenas a verificação do "item" se a versão do IOS corresponder ao regex especificado: (12\.[5-9]).
	
	Caso haja uma falha de verificação da versão do IOS, será exibida uma mensagem de erro parecida com a seguinte:
	
	`"Version Check" : [SKIPPED]`
	`Test defined for 12.[5-9] whereas we are running 12.4(15)T10`

info	A palavra-chave "**info**" é usada para adicionar uma descrição mais detalhada à verificação que estiver sendo executada. O fundamento lógico para a verificação pode ser uma regulamentação, URL com mais informações, política corporativa etc. Podem ser adicionados vários campos **info** em linhas separadas para a formatação do texto como um parágrafo. Não há limite predefinido para o número de campos **info** que podem ser usados.
	
	  Cada tag "**info**" deve ser gravada em uma linha separada sem quebras de linha. Se for necessário mais de uma linha (por razões de formatação, por exemplo), adicione tags "**info**" adicionais.
	
	Exemplo:
	`info: "Verify at least one local user exists and ensure"`
	`info: "all locally defined user passwords are protected"`
	`info: "by encryption."`

item	A palavra-chave "**item**" especifica o item de configuração dentro da saída "show config" a ser auditada. Exemplo: `item:"transport input ssh"` Poderão ser usadas expressões regulares dentro desta palavra-chave para a filtragem dos resultados da correspondência. Observe a descrição da palavra-chave `regex` para obter mais detalhes da funcionalidade **regex**.
regex	A palavra-chave "**regex**" permite buscar a definição do item de configuração para correspondência com uma expressão regular específica. Exemplo: `regex: "snmp-server community ([^ ]*) .*"` Os metacaracteres a seguir requerem tratamento especial: + \ * ( ) ^

	Proteja os caracteres duas vezes com duas barras invertidas “\\” ou delimite-os entre colchetes “[]” se desejar que sejam interpretados de forma literal. Outros caracteres, como os abaixo, requerem apenas uma única barra invertida para serem interpretados de forma literal: . ? " ' Isto está relacionado com a maneira pela qual o compilador trata esses caracteres.
min_occurrences	A palavra-chave “**min_occurrences**” especifica o número mínimo de ocorrências do item de configuração necessárias para que seja aprovado pela auditoria. Exemplo: `min_occurrences: "3"`
max_occurrences	A palavra-chave “**max_occurrences**” especifica o número máximo de ocorrências do item de configuração necessárias para que seja aprovado pela auditoria. Exemplo: `max_occurrences: "1"`
required	A palavra-chave “**required**” é usada para especificar se o item auditado deve ou não estar presente no sistema remoto. Se **required**, por exemplo, for definido como “No” (Não) e o tipo de verificação for “CONFIG_CHECK”, a verificação será aprovada se o item de configuração existir ou se o item de configuração não existir. Por outro lado, se **required** for definido como “Yes” (Sim), a verificação acima será reprovada. Exemplo: `required: NO`

context	A palavra-chave “**context**” será usada em caso de mais de uma ocorrência de um item de configuração específico. Por exemplo: vejamos a seguinte configuração:
	
	`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`
	
	Para verificar o valor de uma linha serial específica, o uso da palavra-chave **item** com “line” não será suficiente, pois há mais de uma opção “line”. Se “**context**” for usado, destaca-se somente o item de seu interesse, por exemplo:
	
	`context: "con 0"`
	
	Pesquise somente o seguinte item de configuração:
	
	`line con 0`
	` no modem enable`
	
	Poderão ser usadas expressões regulares dentro desta palavra-chave para a filtragem dos resultados da correspondência. Observe a descrição da palavra-chave **regex** para obter mais detalhes da funcionalidade **regex**.

Exemplos de linha de comando

Esta seção fornece alguns exemplos de auditorias comuns usadas para verificações de conformidade do Cisco IOS. O binário da linha de comando **`nasl`** é usado como um meio rápido para o teste de auditorias durante o processo. Todos os arquivos **`.audit`** demonstrados a seguir podem fazer parte das políticas de varredura do Nessus. No entanto, para auditorias rápidas de um único sistema, os testes da linha de comando são mais eficientes. O comando será sempre executado do diretório **`/opt/nessus/bin`**, conforme exemplo a seguir:

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

O <IP> é o endereço IP do sistema a ser auditado.

É solicitada a senha “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 :
```

Consulte o administrador do Cisco para obter os parâmetros corretos para o login com “enable”.

Exemplo 1: Busca de um SNMP ACL definido

O exemplo a seguir apresenta um arquivo **`.audit`** simples que pesquisa um SNMP ACL “deny” definido. Se nenhum arquivo for encontrado, a auditoria exibirá uma mensagem de falha. Esta verificação será executada somente se a versão IOS do roteador corresponder ao regex especificado. Do contrário, a verificação será ignorada.

```
<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>
```

Ao executar este comando, espera-se a seguinte saída de um sistema conforme:

```
"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.
```

A auditoria com reprovação retornaria o seguinte:

```
"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
```

Neste caso, a verificação foi reprovada porque o programa procurou uma regra “deny ip”, mas nenhuma foi encontrada.

Exemplo 2: Certifique-se de que o serviço “finger” esteja desativado

O exemplo a seguir apresenta um arquivo **`.audit`** simples que procura serviço “**`finger`**” sem segurança no roteador remoto. Esta verificação será executada somente se a versão IOS do roteador corresponder ao regex especificado. Do contrário, a verificação será ignorada. Se o serviço for encontrado, a auditoria exibirá uma mensagem de falha.

```
<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>
```

Ao executar este comando, espera-se a seguinte saída de um sistema conforme:

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

Disable finger server.
```

A auditoria com reprovação retornaria o seguinte:

```
"Forbid Finger Service" : [FAILED]
Disable finger server.
 - error message:
The following configuration line is set:
ip finger <----

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

Exemplo 3: Verificação de aleatoriedade para avaliar se os strings da comunidade SNMP e o controle de acesso são suficientemente aleatórios

No exemplo a seguir, o arquivo **`.audit`** simples procura strings da comunidade SNMP que não sejam aleatórios o suficiente. Se o string de comunidade encontrado não for considerado aleatório o bastante, a auditoria exibirá uma mensagem de falha. Uma vez que a opção “required” está definida como “NO” (Não), a verificação será aprovada se não existir nenhum string de comunidade “snmp-server”. Esta verificação será executada apenas se o roteador estiver usando o Feature Set: “K9”. Do contrário, a verificação será ignorada.

```
<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>
```

Ao executar este comando, espera-se a seguinte saída de um sistema conforme:

```
"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.
```

A auditoria com reprovação retornaria o seguinte:

```
"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:

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
```

No caso acima, havia dois strings: “foobar” e “public”, que não possuíam um token suficientemente aleatório e, portanto, resultaram na reprovação da verificação.

Exemplo 4: Verificação de contexto para avaliar o controle de acesso a SSH

No exemplo a seguir, o arquivo **`.audit`** simples localiza todos os itens de configuração “line” com a palavra-chave “**`context`**” e executa uma **`regex`** para avaliar se o controle de acesso a SSH está configurado.

```
<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>
```

Ao executar este comando, espera-se a seguinte saída de um sistema conforme:

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

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

A auditoria com reprovação retornaria o seguinte:

```
"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

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
```

No caso acima, há dois strings que correspondem ao regex da palavra-chave "**`context`**" de "**`line .*`**". Uma vez que nenhuma linha continha o regex "**`item`**", a auditoria retornou a mensagem "FAILED".

Condições

É possível definir a lógica **`if/then/else`** na política de auditoria Cisco. Isto possibilita ao usuário final retornar uma mensagem de advertência em vez de aprovado/reprovado caso uma auditoria resulte em aprovação.

A sintaxe para executar as condições é a seguinte:

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

Exemplo:

```
<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>
</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>
```

Independentemente de a condição ser reprovada ou aprovada, não aparecerá no relatório porque é uma verificação “silent” (silenciosa).

As condições podem ser do tipo “**and**” ou “**or**”.

Referência de arquivo de conformidade de auditoria para configuração do Juniper

Esta seção descreve o formato e as funções das verificações de conformidade do Juniper e o fundamento lógico por trás de cada configuração.

Uso de aspas:

As aspas simples e aspas duplas são intercambiáveis quando estiverem delimitando campos de auditoria, exceto nos dois casos abaixo:

1. Nas verificações de conformidade do Windows em que os campos especiais como CRLF etc. devem ser interpretados literalmente, use aspas simples. Todos os campos incorporados interpretados como strings devem ser protegidos.

Por exemplo:

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

2. É necessário o uso de aspas ao usar “include_paths” e “exclude_paths” do WindowsFiles.

Se os strings forem usados em qualquer tipo de campo (descrição, value_data, regex etc.) que contém aspas simples ou aspas duplas, proceda da seguinte maneira:

a. Use o tipo oposto de aspas para as aspas delimitadoras mais afastadas.

Por exemplo:

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

b. Isole as aspas incorporadas com uma barra invertida (somente aspas duplas).
Por exemplo:

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

Tipo de verificação: CONFIG_CHECK

As verificações de conformidade do Juniper são delimitadas por encapsulamento **custom_item** e CONFIG_CHECK ou SHOW_CONFIG_CHECK. São tratadas como quaisquer outros arquivos .audit e funcionam para sistemas que usam o sistema operacional do Juniper (Junos). A verificação CONFIG_CHECK consiste em duas ou mais palavras-chave. As palavras-chave **type** e **description** são obrigatórias e são seguidas por uma ou mais palavras-chave. A verificação funciona auditando a configuração no formato "set".

A configuração no formato "set" pode ser obtida acrescentando "display set" à solicitação "show configuration". Por exemplo:

```
show configuration | display set
```

```
admin> show configuration | display set
set version 10.2R3.10
set system time-zone GMT
set system no-ping-record-route
set system root-authentication encrypted-password "$1$hSGSlnwfdsdfdfsdfsdf43534"
.
.
```

Palavras-chave

A tabela a seguir indica como poderá ser usada cada palavra-chave nas verificações de conformidade do Juniper:

Palavra-chave	Exemplo de uso e configurações aceitas
type	CHECK_CONFIG e SHOW_CHECK_CONFIG "CONFIG_CHECK" determina se o item de configuração especificado existe na saída "show configuration" do Juniper em formato "set". Da mesma forma, "SHOW_CONFIG_CHECK" audita se o item de configuração existe na saída "show configuration" em formato padrão.
description	A palavra-chave "**description**" permite adicionar uma breve descrição da verificação que estiver sendo realizada. Não podemos deixar de recomendar que o campo description seja exclusivo e que nenhuma verificação distinta tenha o mesmo campo description. O SecurityCenter da Tenable usa este campo para a geração automática de um número exclusivo de identificação do plugin com base no campo description. Exemplo: `description: "3.1 Disable Unused Interfaces"`

	`(3.1 Desativar interfaces não usadas)`

info	A palavra-chave "**info**" é usada para adicionar uma descrição mais detalhada à verificação que estiver sendo executada. O fundamento lógico para a verificação pode ser uma regulamentação, URL com mais informações, política corporativa etc. Podem ser adicionados vários campos info em linhas separadas para a formatação do texto como um parágrafo. Não há limite predefinido para o número de campos info que podem ser usados.
	
	  Cada tag "**info**" deve ser gravada em uma linha separada sem quebras de linha. Se for necessário mais de uma linha (por razões de formatação, por exemplo), adicione tags "info" adicionais.
	
	Exemplo:
	`info: "Review the the list of interfaces" (Revisar a lista de interfaces)`
	`info: "Disable Unused Interfaces" (Desativar interfaces não usadas)`

severity	A palavra-chave "**severity**" especifica a gravidade da verificação que está sendo executada. Exemplo: `severity : MEDIUM` A gravidade pode ser definida como HIGH, MEDIUM ou LOW.
regex	A palavra-chave "**regex**" permite buscar a definição do item de configuração para correspondência com uma expressão regular específica. Exemplo: `regex: set system syslog .+"` Os metacaracteres a seguir requerem tratamento especial: + \ * ( ) ^ Isole os caracteres duas vezes com duas barras invertidas "`\\`" ou delimite-os entre colchetes "`[]`" se desejar que sejam interpretados de forma literal. Outros caracteres, como os abaixo, requerem apenas uma única barra invertida para serem interpretados de forma literal: . ? " ' Isto está relacionado com a maneira pela qual o compilador trata esses caracteres. Se uma verificação tem a tag "regex" definida, mas não tem definida a tag "expect" ou "not_expect" ou "number_of_lines", a verificação simplesmente reporta todas as linhas que correspondem ao regex.
expect	Essa palavra-chave permite auditar o item de configuração correspondido pela tag "**regex**" ou, se a tag "**regex**" não for usada, procura o string "expect" em toda a configuração. Exemplo: `expect: "syslog host 1.1.1.1"` A verificação é aprovada, desde que a linha de configuração encontrada por "regex"

	corresponda à tag “**expect**” ou, se “**regex**” não estiver definida, é aprovada se o string “expect” é encontrada na configuração.
	
	Exemplo:
	`regex: "syslog host [0-9\.]+"`
	`expect: "syslog host 1.1.1.1"`
	
	No caso acima, a tag “`expect`” garante que o host syslog seja definido como 1.1.1.1.

not_expect	Essa palavra-chave permite pesquisar os itens de configuração que não devem estar na configuração.
	
	Exemplo:
	`not_expect: "syslog host 1.1.1.1"`
	
	Atua como o oposto de “**expect**”. A verificação é aprovada, desde que a linha de configuração encontrada por “**regex**” não corresponda à tag “**not_expect**” ou, se a tag “**regex**” não estiver definida, é aprovada se o string “not_expect” não é encontrada na configuração.
	
	Exemplo:
	`regex: "syslog host [0-9\.]+"`
	`not_expect: "syslog host 1.1.1.1"`
	
	No caso acima, a tag “**not_expect**” garante que o host syslog seja definido como 1.1.1.1.

number_of_lines	Essa palavra-chave permite testar a conformidade de uma verificação .audit com base no número de linhas correspondentes retornadas pela configuração.
	
	`<custom_item>`
	` type: CONFIG_CHECK`
	` description: "Syslog"`
	` regex: "syslog host [0-9\.]+"`
	` number_of_lines: "^1$"`
	`</custom_item>`
	
	No caso acima, a verificação é aprovada desde que seja retornada apenas uma linha que corresponda a “**regex**”.

Exemplos de CONFIG_CHECK

Os seguintes são exemplos do uso de CONFIG_CHECK em um dispositivo Juniper:

```
<custom_item>
 type: CONFIG_CHECK
 description: "Audit Syslog host message severity"
 regex: "syslog host [0-9\.]+"
 expect: "syslog host [0-9\.]+ 6 .+"
</custom_item>
```

```
<custom_item>
 type: CONFIG_CHECK
 description: "Audit Syslog host"
```

```
 regex: "syslog host [0-9\.]+"
 number_of_lines: "^1$"
</custom_item>
```

```
<custom_item>
 type: CONFIG_CHECK
 description: "Audit Syslog host"
 regex: "syslog host [0-9\.]+"
 not_expect: "syslog host 1.2.3.4"
</custom_item>
```

```
<custom_item>
 type: CONFIG_CHECK
 description: "Audit Syslog settings"
 regex: "syslog .+"
</custom_item>
```

Tipo de verificação: SHOW_CONFIG_CHECK

Essa verificação, de muitas formas, audita as mesmas configurações auditadas pela verificação CONFIG_CHECK .audit. Porém, o formato da configuração auditada é diferente. SHOW_CONFIG_CHECK audita a configuração no formato padrão.

Por exemplo, aqui está a configuração no formato padrão:

```
admin> show configuration system syslog
user * {
    any emergency;
}
host 1.1.1.1 {
    any none;
}
file messages {
    any any;
    authorization info;
}
file interactive-commands {
    interactive-commands any;
}
```

Essa verificação não é recomendada, a não ser que você precise de maior flexibilidade sobre CONFIG_CHECK. Como cada verificação SHOW_CONFIG_CHECK .audit resulta na execução de um comando separado no dispositivo Juniper, o processo pode resultar em mais overhead de CPU e demorar mais para conclusão. Essa verificação existe para proporcionar flexibilidade para o auditor, e apoia um caso de uso futuro que pode não ser auditado com eficiência usando CONFIG_CHECK.

Palavras-chave

A tabela a seguir indica como poderá ser usada cada palavra-chave nas verificações de conformidade do Junos. Observe que a conformidade de uma verificação pode ser auditada comparando a saída da verificação com a tag “**expect**”, “**not_expect**” ou “**number_of_lines**”. Não pode haver mais de uma tag de teste de conformidade (ou seja, “**expect**”, “**not_expect**” ou “**number_of_lines**” pode existir, mas não “**expect**” e “**not_expect**”).

Palavra-chave	Exemplo de uso e configurações aceitas

hierarchy	Essa palavra-chave permite que os usuários naveguem para uma hierarquia específica na configuração do Junos.
	
	Exemplo:
	`hierarchy: "interfaces"`
	
	Internamente, a palavra-chave hierarchy é acrescentada ao comando “show configuration” em uma SHOW_CONFIG_CHECK. Por exemplo:
	
	`<custom_item>`
	` type: SHOW_CONFIG_CHECK`
	` description: "3.6 Forbid Multiple Loopback Addresses"`
	` hierarchy: "interfaces"`
	`</custom_item>`
	
	A verificação acima equivale a executar:
	
	`show configuration interfaces`

property	Essa palavra-chave permite que os usuários auditem uma “property” específica no dispositivo Junos. Por padrão, SHOW_CONFIG_CHECK audita o comando “show configuration” seguido por uma ou mais palavras-chave, como match, except e find. No caso em que a palavra-chave “property” está definida, audita a propriedade específica.
	
	Exemplo:
	`property: "ospf"`
	
	`<custom_item>`
	` type: SHOW_CONFIG_CHECK`
	` description: "4.3.1 Require MD5 Neighbor Authentication`
	`  (where OSPF is used)"`
	` info: "Level 2, Scorable"`
	` property: "ospf"`
	` hierarchy: "interface detail"`
	` match: "Auth type MD5"`
	`</custom_item>`
	
	A verificação acima equivale a executar:
	
	`show ospf interface detail`
	
	Observe que o exemplo acima não executou “show configuration”, como ocorreu em outros exemplos.

find	Essa palavra-chave encontra a hierarquia de configuração apropriada em uma verificação SHOW_CONFIG_CHECK .audit. `find: "chap"`

	A palavra-chave find é acrescentada à solicitação “show configuration”.
	
	`<custom_item>`
	` type: SHOW_CONFIG_CHECK`
	` description: "3.8.2 Require CHAP Authentication if Incoming Map is Used"`
	` hierarchy: "interfaces"`
	` find: "chap"`
	` match: "access-profile"`
	`</custom_item>`
	
	A verificação acima equivale a executar:
	
	`show configuration interfaces | find "chap" | match "access-profile"`

match	Essa palavra-chave procura linhas correspondentes em uma verificação SHOW_CONFIG_CHECK **.audit**.
	
	`match: "multihop"`
	
	A palavra-chave match é acrescentada à solicitação “**show configuration**”.
	
	`<custom_item>`
	` type: SHOW_CONFIG_CHECK`
	` description: "3.6 Forbid Multiple Loopback Addresses"`
	` hierarchy: "interfaces"`
	` match: "lo[0-9]"`
	`</custom_item>`
	
	A verificação acima equivale a executar:
	
	`show configuration interfaces | match "lo[0-9]"`

except	Essa palavra-chave exclui determinadas linhas da configuração em uma verificação SHOW_CONFIG_CHECK .audit.
	
	`except: "multihop"`
	
	A palavra-chave except é acrescentada à solicitação “**show configuration**”.
	
	`<custom_item>`
	` type: SHOW_CONFIG_CHECK`
	` description: "6.8.1 Require External Time Sources"`
	` hierarchy: "system ntp"`
	` match: "server"`
	` except: "boot-server"`

	`</custom_item>`
	
	A verificação acima equivale a executar:
	
	`show configuration system ntp | match "server" | except "boot-server"`

expect	Essa palavra-chave permite auditar o item de configuração correspondido pela tag "**regex**" ou, se a tag "**regex**" não for usada, procura o string "expect" em toda a configuração. A verificação é aprovada, desde que a linha de configuração encontrada por "**regex**" corresponda à tag "**expect**" ou, se "**regex**" não estiver definida, é aprovada se o string "expect" é encontrado na configuração.
	
	`regex: "syslog host [0-9\.]+"`
	`expect: "syslog host 1.2.4.5"`
	
	No caso acima, a tag "**expect**" garante que a complexidade seja definida como um valor entre 1 e 4.
	
	`expect: "syslog host"`
	
	No caso acima, a tag "**expect**" garante que a complexidade seja definida como 4.

not_expect	Essa palavra-chave permite pesquisar os itens de configuração que não devem estar na configuração.
	
	Atua como o oposto de "**expect**". A verificação é aprovada, desde que a linha de configuração encontrada por "**regex**" não corresponda à tag "**not_expect**" ou, se a tag "**regex**" não estiver definida, é aprovada se o string "not_expect" não é encontrado na configuração.
	
	`regex: "syslog host [0-9\.]+"`
	`not_expect: "syslog host 1.2.3.4"`
	
	`not_expect: "syslog host"`

number_of_lines	Essa palavra-chave permite testar a conformidade de uma verificação .audit com base no número de linhas correspondentes retornadas pela configuração.
	
	`<custom_item>`
	`type: CONFIG_CHECK`
	`description: "Syslog"`
	`regex: "syslog host [0-9\.]+"`
	`number_of_lines: "^1$"`
	`</custom_item>`
	
	No caso acima, a verificação é aprovada desde que seja retornada apenas uma linha que corresponda a "**regex**".

Exemplos de SHOW_CONFIG_CHECK

Os seguintes são exemplos do uso de SHOW_CONFIG_CHECK em um dispositivo Juniper:

```
<custom_item>
 type: SHOW_CONFIG_CHECK
 description: "6.1.2 Require Accounting of Logins & Configuration Changes"
 hierarchy: "system accounting"
 find: "accounting"
 expect: "events [change-log login];"
</custom_item>
```

```
<custom_item>
 type: SHOW_CONFIG_CHECK
 description: "6.2.2 Require Archive Site"
 hierarchy: "system archival configuration archive-sites"
 match: "scp://"
 number_of_lines: "^([1-9]|[0-9][0-9]+)+$"
</custom_item>
```

```
<custom_item>
 type: SHOW_CONFIG_CHECK
 description: "4.7.1 Require BFD Authentication (where BFD is used)"
 hierarchy: "protocols"
 match: "authentication"
 except: "loose"
 number_of_lines: "^2$"
 check_option: CAN_BE_NULL
</custom_item>
```

```
<custom_item>
 type: SHOW_CONFIG_CHECK
 description: "4.3.1 Require MD5 Neighbor Authentication (where OSPF is used)"
 property: "ospf"
 hierarchy: "interface detail"
 match: "Auth type MD5"
 number_of_lines: "^([1-9]|[0-9][0-9]+)+$"
 check_option: CAN_BE_NULL
</custom_item>
```

Condições

É possível definir a lógica **`if/then/else`** na política de auditoria Juniper. Isto permite ao usuário final usar um único arquivo que seja capaz de tratar diversas configurações.

A sintaxe para executar as condições é a seguinte:

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

```
    < Insert your audit here >
 </then>
 <else>
    < Insert your audit here >
 </else>
</if>
```

Exemplo:

```
<if>
 <condition type: "OR">

<custom_item>
 type: CONFIG_CHECK
 description: "Configure Syslog Host"
 regex: "syslog host [0-9\.]+"
 not_expect: "syslog host 1.2.3.4"
</custom_item>

 </condition>
 <then>
  <report type: "PASSED">
   description: "Configure Syslog Host."
  </report>
 </then>
 <else>
<custom_item>
 type: CONFIG_CHECK
 description: "Configure Syslog Host"
 regex: "syslog host [0-9\.]+"
 not_expect: "syslog host 1.2.3.4"
</custom_item>

 </else>
</if>
```

Independentemente de a condição ser reprovada ou aprovada, ela não aparecerá no relatório (porque é uma verificação “silent” - silenciosa).

As condições podem ser do tipo “**and**” ou “**or**”.

Relatórios

Podem ser executados em <then> ou <else> para alcançar uma condição PASSED/FAILED desejada.

```
<if>
 <condition type: "OR">
 <custom_item>
  type: CONFIG_CHECK
  description: "Configure Syslog Host"
  regex: "syslog host [0-9\.]+"
  not_expect: "syslog host 1.2.3.4"
</custom_item>
 </condition>
 <then>
```

```
   <report type: "PASSED">
    description: "Configure Syslog host"
   </report>
  </then>
  <else>
  <report type: "FAILED">
    description: "Configure Syslog host"
   </report>
  </else>
 </if>
```

PASSED, WARNING e FAILED são valores aceitáveis para “**report type**”.

Referência de arquivo de conformidade de auditoria para configuração de Check Point GAiA

Esta seção descreve o formato e as funções das verificações de conformidade do Check Point GAiA e o fundamento lógico por trás de cada configuração.

Uso de aspas:

As aspas simples e aspas duplas são intercambiáveis quando estiverem delimitando campos de auditoria, exceto nos dois casos abaixo:

1. Nas verificações de conformidade do Windows em que os campos especiais como CRLF etc. devem ser interpretados literalmente, use aspas simples. Todos os campos incorporados interpretados como strings devem ser protegidos.

Por exemplo:

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

2. É necessário o uso de aspas ao usar “**include_paths**” e “**exclude_paths**” do WindowsFiles.

Se os strings forem usados em qualquer tipo de campo (descrição, value_data, regex etc.) que contém aspas simples ou aspas duplas, proceda da seguinte maneira:

a. Use o tipo oposto de aspas para as aspas delimitadoras mais afastadas.

Por exemplo:

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

b. Isole as aspas incorporadas com uma barra invertida (somente aspas duplas).

Por exemplo:

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

Tipo de verificação: CONFIG_CHECK

As verificações de conformidade do Check Point são delimitadas por encapsulamentocustom_item e CONFIG_CHECK. São tratadas como quaisquer outros arquivos .audit e funcionam para sistemas que usam o sistema operacional Check Point GAiA. A verificação CONFIG_CHECK consiste em duas ou mais palavras-chave. As palavras-chave type e

description são obrigatórias e são seguidas por uma ou mais palavras-chave. A verificação funciona auditando a saída do comando “show config”, que está em formato “set” por padrão.

Palavras-chave

A tabela a seguir indica como poderá ser usada cada palavra-chave nas verificações de conformidade do GAiA:

Palavra-chave	Exemplo de uso e configurações aceitas
type	“CONFIG_CHECK” determina se o item de configuração especificado existe na saída “show config” do GAiA.
description	A palavra-chave “**description**” permite adicionar uma breve descrição da verificação que estiver sendo realizada. Não podemos deixar de recomendar que o campo description seja exclusivo e que nenhuma verificação distinta tenha o mesmo campo description. O SecurityCenter da Tenable usa este campo para a geração automática de um número exclusivo de identificação do plugin com base no campo description. Exemplo: `description: "1.0 Require strong Password Controls - 'min-password-length >= 8'"`
info	A palavra-chave “**info**” é usada para adicionar uma descrição mais detalhada à verificação que estiver sendo executada. O fundamento lógico para a verificação pode ser uma regulamentação, URL com mais informações, política corporativa etc. Podem ser adicionados vários campos info em linhas separadas para a formatação do texto como um parágrafo. Não há limite predefinido para o número de campos info que podem ser usados.  Cada tag “**info**” deve ser gravada em uma linha separada sem quebras de linha. Se for necessário mais de uma linha (por razões de formatação, por exemplo), adicione tags “info” adicionais. Exemplo: `info: "Enable palindrome-check on passwords"`
severity	A palavra-chave “**severity**” especifica a gravidade da verificação que está sendo executada. Exemplo: `severity: MEDIUM` A gravidade pode ser definida como HIGH, MEDIUM ou LOW.
regex	A palavra-chave “**regex**” permite buscar a definição do item de configuração para correspondência com uma expressão regular específica. Exemplo: `regex: "set snmp .+"` Os metacaracteres a seguir requerem tratamento especial: + \ * ( ) ^ Isole os caracteres duas vezes com duas barras invertidas “\\” ou delimite-os entre colchetes “[]” se desejar que sejam interpretados de forma literal. Outros caracteres, como os abaixo, requerem apenas uma única barra invertida para serem interpretados

	de forma literal: . ? " ' Isto está relacionado com a maneira pela qual o compilador trata esses caracteres. Se uma verificação tem a tag "**regex**" definida, mas não tem definida a tag "**expect**" ou "**not_expect**" ou "**number_of_lines**", a verificação simplesmente reporta todas as linhas que correspondem ao regex.

expect	Essa palavra-chave permite auditar o item de configuração correspondido pela tag "**regex**" ou, se a tag "**regex**" não for usada, procura o string "**expect**" em toda a configuração.
	
	A verificação é aprovada, desde que a linha de configuração encontrada por "**regex**" corresponda à tag "**expect**" ou, se "**regex**" não estiver definida, é aprovada se o string "**expect**" é encontrado na configuração.
	
	Exemplo:
	`regex: "set password-controls complexity"`
	`expect: "set password-controls complexity [1-4]"`
	
	No caso acima, a tag "**expect**" garante que a complexidade seja definida como um valor entre 1 e 4.

not_expect	Essa palavra-chave permite pesquisar os itens de configuração que não devem estar na configuração.
	
	Atua como o oposto de "**expect**". A verificação é aprovada, desde que a linha de configuração encontrada por "**regex**" não corresponda à tag "**not_expect**" ou, se a tag "**regex**" não estiver definida, é aprovada se o string "not_expect" não é encontrado na configuração.
	
	Exemplo:
	`regex: "set password-controls password-expiration"`
	`not_expect: "set password-controls password-expiration never"`
	
	No caso acima, a tag "**not_expect**" garante que os controles de senha não sejam definidos como "**never**".

Exemplos de SHOW_CONFIG_CHECK

Os seguintes são exemplos do uso de CONFIG_CHECK em um dispositivo Check Point:

```
<custom_item>
 type: CONFIG_CHECK
 description: "1.0 Require strong Password Controls - 'min-password-length >= 8'"
 regex: "set password-controls min-password-length"
 expect: "set password-controls min-password-length ([8-9]|[0-9][0-9]+)"
 info: "Require Password Lengths greater than or equal to 8."
</custom_item>
```

```
<custom_item>
 type: CONFIG_CHECK
 description: "1.0 Require strong Password Controls - 'password-expiration != never'"
 regex: "set password-controls password-expiration"
```

```
 not_expect: "set password-controls password-expiration never"
 info: "Allow passwords to expire"
</custom_item>
```

```
<custom_item>
 type: CONFIG_CHECK
 description: "2.13 Secure SNMP"
 regex: "set snmp .+"
 severity: MEDIUM
 info: "Manually review SNMP settings."
</custom_item>
```

Condições

É possível definir a lógica **`if/then/else`** na política de auditoria Check Point. Isto permite ao usuário final usar um único arquivo que seja capaz de tratar diversas configurações.

A sintaxe para executar as condições é a seguinte:

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

Exemplo:

```
<if>
 <condition type: "OR">
 <custom_item>
  type: CONFIG_CHECK
  description: "2.6 Install and configure Encrypted Connections to devices - 'telnet'"
  regex: "set net-access telnet"
  expect: "set net-access telnet off"
  info: "Do not use plain-text protocols."
</custom_item>
 </condition>
 <then>
  <report type: "PASSED">
   description: "Telnet is disabled"
  </report>
 </then>
 <else>
  <custom_item>
  type: CONFIG_CHECK
  description: "2.6 Install and configure Encrypted Connections to devices - 'telnet'"
  regex: "set net-access telnet"
  expect: "set net-access telnet off"
```

```
  info: "Do not use plain-text protocols."
</custom_item>
 </else>
</if> </else>
</if>
```

Independentemente de a condição ser reprovada ou aprovada, ela não aparecerá no relatório (porque é uma verificação “silent” - silenciosa).

As condições podem ser do tipo “**and**” ou “**or**”.

Relatórios

Podem ser executados em <then> ou <else> para alcançar uma condição PASSED/FAILED desejada.

```
<if>
 <condition type: "OR">
 <custom_item>
  type: CONFIG_CHECK
  description: "2.6 Install and configure Encrypted Connections to devices - 'telnet'"
  regex: "set net-access telnet"
  expect: "set net-access telnet off"
  info: "Do not use plain-text protocols."
</custom_item>
 </condition>
 <then>
  <report type: "PASSED">
   description: "Telnet is disabled"
  </report>
 </then>
 <else>
 <report type: "FAILED">
   description: "Telnet is disabled"
  </report>
 </else>
</if>
```

PASSED, WARNING e FAILED são valores aceitáveis para “**report type**”.

Referência de arquivo de conformidade de auditoria para configuração de banco de dados

Esta seção descreve o formato e as funções das verificações de conformidade do banco de dados e o fundamento lógico por trás de cada configuração.

Uso de aspas:

As aspas simples e aspas duplas são intercambiáveis quando estiverem delimitando campos de auditoria, exceto nos dois casos abaixo:

1. Nas verificações de conformidade do Windows em que os campos especiais como CRLF etc. devem ser interpretados literalmente, use aspas simples. Todos os campos incorporados interpretados como strings devem ser protegidos.

Por exemplo:

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

2. É necessário o uso de aspas ao usar “include_paths” e exclude_paths do WindowsFiles.

Se os strings forem usados em qualquer tipo de campo (descrição, value_data, regex etc.) que contenha aspas simples ou aspas duplas, proceda da seguinte maneira:

a. Use o tipo oposto de aspa para as aspas delimitadoras mais afastadas.

Por exemplo:

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

b. Isole as aspas incorporadas com uma barra invertida (somente aspas duplas).

Por exemplo:

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

Tipo de verificação

Todas as verificações de conformidade devem estar entre colchetes com o encapsulamento **`check_type`** e a designação “Database”. Isto é necessário para diferenciar arquivos **`.audit`** destinados especificamente a bancos de dados de outros tipos de auditorias de conformidade. O campo **`check_type`** requer dois parâmetros adicionais:

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

Os tipos de banco de dados disponíveis são:

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

A **`version`** (versão) atualmente está definida como “1”.

Exemplo:

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

Palavras-chave

A tabela a seguir indica como poderá ser usada cada palavra-chave nas verificações de conformidade do banco de dados:

Palavra-chave	Exemplo de uso e configurações aceitas
type	SQL_POLICY
description	Esta palavra-chave permite adicionar uma breve descrição da verificação que estiver sendo realizada. Recomenda-se que o campo **description** seja exclusivo e que nenhuma verificação distinta tenha o mesmo campo de descrição. O SecurityCenter da Tenable usa este campo para a geração automática de um número exclusivo de identificação do plugin com base no campo **description**. Exemplo: description: "DBMS Password Complexity"
info	Esta palavra-chave é usada para adicionar uma descrição detalhada à verificação que está sendo executada, como regulamentação, URL, política corporativa ou outro motivo pelo qual a definição é necessária. Podem ser adicionados vários campos **info** em linhas separadas para a formatação do texto como um parágrafo. Não há limite predefinido para o número de campos **info** que podem ser usados. Exemplo: info: "Checking that \"password complexity\" requirements are enforced for systems using SQL Server authentication."
sql_request	Esta palavra-chave é usada para determinar a solicitação SQL real a ser enviada ao banco de dados. As matrizes de dados podem ser solicitadas e retornadas de uma solicitação SQL pelo uso de valores de solicitação/retorno delimitados por vírgula. Exemplo: sql_request: "select name from sys.sql_logins where type = 'S' and is_policy_checked <> '1'" Exemplo: sql_request: "select name, value_in_use from sys.configurations where name = 'clr enabled'"
sql_types	Esta palavra-chave possui duas opções disponíveis: POLICY_VARCHAR e POLICY_INTEGER. Use POLICY_INTEGER para valores numéricos de 0 a 2147483647 e POLICY_VARCHAR para qualquer outro tipo de valor de retorno. Exemplo: sql_types: POLICY_VARCHAR Exemplo: sql_types: POLICY_VARCHAR,POLICY_INTEGER Para vários itens de retorno, configure SQL em uma lista separada por vírgula para aceitar os tipos de dados de cada resultado de retorno sql_types. O exemplo acima indica que o primeiro valor de retorno proveniente da consulta SQL é varchar e que o segundo valor de retorno é um número inteiro.
sql_expect	Esta palavra-chave é usada para determinar o valor de retorno esperado proveniente da solicitação SQL. Poderá ser solicitado um valor exato, incluindo NULL ou "0". Além disso, podem ser necessárias expressões regulares para POLICY_VARCHAR **sql_types**.

	Exemplo:
	`sql_expect: regex:"^.+Failure" || regex:"^.+ALL"`
	
	Exemplo:
	`sql_expect: NULL`
	
	Exemplo:
	`sql_expect: 0 || "0"`
	
	Aspas duplas são opcionais para valores de retorno inteiros.
	Exemplo:
	`sql_expect: "clr enabled",0`
	
	Uma matriz de dados pode retornar de uma solicitação `SQL` incluída em um formato separado por vírgula no campo **`sql_expect`**.

Exemplos de linha de comando

Esta seção fornece alguns exemplos de auditorias comuns usadas para verificações de conformidade do banco de dados. O binário da linha de comando **`nasl`** é usado como um meio rápido para o teste de auditorias durante o processo. Todos os arquivos **`.audit`** demonstrados a seguir podem fazer parte das políticas de varredura do Nessus 4 ou do SecurityCenter. No entanto, para auditorias rápidas de um único sistema, os testes da linha de comando são mais eficientes. O comando será sempre executado do diretório **`/opt/nessus/bin`**, conforme exemplo a seguir:

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

O <IP> é o endereço IP do sistema a ser auditado.

Dependendo do tipo de banco de dados que estiver sendo auditado, você pode ser solicitado a informar outros tipos de parâmetro além do arquivo de auditoria para serem usados. Por exemplo: as auditorias do Oracle solicitarão o banco de dados SID e o tipo de login Oracle:

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

Consulte o administrador do banco de dados para os parâmetros corretos de login no banco de dados.

Exemplo 1: Busca de logins sem data de expiração

No exemplo a seguir, o arquivo **`.audit`** simples procura qualquer login no SQL Server sem data de expiração. Se algum arquivo for encontrado, a auditoria exibirá uma mensagem de falha junto com o(s) login(s) ofensivo(s).

```
<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>
```

Ao executar este comando, espera-se a seguinte saída de um sistema conforme:

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

Os requisitos de conformidade geralmente obrigam que os logins para o banco de dados tenha uma data de expiração.

A auditoria com reprovação retornaria o seguinte:

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

Database logins with no expiration date pose a security threat.

Remote value:

"distributor_admin"

Policy value:

NULL
```

Esta saída indica que a conta “distributor_admin” não tem data de expiração e precisa ser conferida com base na política de segurança do sistema.

Exemplo 2: Verificação do estado de ativação do procedimento armazenado não autorizado

Esta auditoria verifica se o procedimento armazenado “SQL Mail XPs” está ativado. Os procedimentos armazenados externos podem constituir uma ameaça à segurança de alguns sistemas e muitas vezes precisam ser desabilitados.

```
<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>
```

A verificação acima retornará um resultado “passed” (Aprovado) se o procedimento armazenado “SQL Mail XPs” estiver desativado (value_in_use = 0). Do contrário, retornará um resultado “failed” (Reprovado).

Exemplo 3: Verificar estado do banco de dados com resultado sql_types misturado

Em alguns casos, as consultas ao banco de dados de conformidade requerem várias solicitações de dados com vários resultados do tipo de dados. O exemplo de auditoria a seguir combina vários tipos de dados e demonstra como realizar a análise sintática da saída.

```
<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>
```

Observe que todos os valores **`sql_request`**, **`sql_types`** e **`sql_expect`** contêm valores separados por vírgula.

Condições

É possível definir a lógica `if/then/else` na política do banco de dados. Isto possibilita ao usuário final retornar uma mensagem de advertência em vez de aprovado/reprovado caso uma auditoria resulte em aprovação.

A sintaxe para executar as condições é a seguinte:

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

Exemplo:

```
<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>
```

Independentemente de a condição ser reprovada ou aprovada, não aparecerá no relatório porque é uma verificação “silent” (silenciosa).

As condições podem ser do tipo “**and**” ou “**or**”.

Referência de arquivo de conformidade de auditoria para configuração do Unix

Esta seção descreve as funções incorporadas das verificações de conformidade do Unix e o fundamento lógico por trás de cada configuração.

Uso de aspas:

As aspas simples e aspas duplas são intercambiáveis quando estiverem delimitando campos de auditoria, exceto nos dois casos abaixo:

1. Nas verificações de conformidade do Windows em que os campos especiais como CRLF etc. devem ser interpretados literalmente, use aspas simples. Todos os campos incorporados interpretados como strings devem ser protegidos.

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

2. É necessário o uso de aspas ao usar “include_paths” e exclude_paths do WindowsFiles.

Se os strings forem usados em qualquer tipo de campo (descrição, value_data, regex etc.) que contém aspas simples ou aspas duplas, proceda da seguinte maneira:

a. Use o tipo oposto de aspa para as aspas delimitadoras mais afastadas.

Por exemplo:

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

b. Isole as aspas incorporadas com uma barra invertida (somente aspas duplas).

Por exemplo:

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

Tipo de verificação

Todas as verificações de conformidade do Unix devem estar entre colchetes com o encapsulamento "**check_type**" e a designação "Unix". O Apêndice A contém uma verificação de conformidade do Unix começando com a configuração **check_type** para "Unix" e finalizando com a tag "**</check_type>**".

Isto é necessário para diferenciar arquivos **.audit** destinados a auditorias de conformidade para Windows (ou outras plataformas).

O arquivo é lido por meio de SSH em um buffer de memória no servidor Nessus e, em seguida, o buffer é processado para verificar se há conformidade ou não conformidade.

Palavras-chave

A tabela abaixo indica como usar cada palavra-chave nas verificações de conformidade do Unix:

Palavra-chave	Exemplo de uso e configurações aceitas
attr	Esta palavra-chave é usada junto com FILE_CHECK e FILE_CHECK_NOT para auditar os atributos do arquivo associados a um arquivo. Consulte a página principal chattr(1) para obter mais detalhes sobre a configuração dos atributos de um arquivo.
comment	Este campo é usado para o acréscimo de qualquer informação adicional que não caiba no campo de descrição. Exemplo: `comment: (CWD - Current working directory)`
description	Esta palavra-chave fornece uma breve descrição da verificação que estiver sendo realizada. O campo **description** deve ser exclusivo e que nenhuma de duas verificações tenha o mesmo campo de descrição. O SecurityCenter da Tenable usa este campo para a geração automática de um número exclusivo de identificação do plugin com base no campo **description**. Exemplo: `description: "Permission and ownership check for /etc/at.allow"`
dont_echo_cmd	Esta palavra-chave é usada com auditorias de verificação de conformidade do Unix "CMD_EXEC" e solicita que a auditoria substitua a execução de comandos pela verificação realizada na saída. São exibidos apenas os resultados do comando. Exemplo: `dont_echo_cmd: YES`
except	Esta palavra-chave é usada para a exclusão de determinados usuários, serviços e arquivos da verificação. Exemplo: `except: "guest"` Várias contas de usuário podem ser agrupadas com barra vertical.

	Exemplo: `except: "guest" | "guest1" | "guest2"`

expect	Esta palavra-chave é usada em combinação com **`regex`**. Ela permite pesquisar valores específicos dentro de arquivos.
	
	Exemplo:
	`<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	Esta palavra-chave é usada para descrever o caminho absoluto ou relativo de um arquivo nos qual as definições de permissões e propriedade são verificadas.
	
	Exemplos:
	`file: "/etc/inet/inetd.conf"`
	`file: "~/inetd.conf"`
	
	O valor **`file`** também pode ser glob.
	
	Exemplo:
	`file: "/var/log/*"`
	
	Este recurso será usado se todos os arquivos dentro de um determinado diretório forem auditados quanto a permissões ou conteúdos com FILE_CHECK, FILE_CONTENT_CHECK, FILE_CHECK_NOT ou FILE_CONTENT_CHECK_NOT.

file_type	Esta palavra-chave descreve o tipo de arquivo pesquisado. A lista dos tipos de arquivo aceitos é mostrada abaixo.
	
	• b - block (bloco especial (com buffer))
	• c - character (caractere especial (sem buffer))
	• d - directory (diretório)
	• p - named pipe (pipe nomeado (FIFO))
	• f – regular file (arquivo normal)
	
	Exemplo:
	`file_type: "f"`
	
	Um ou mais tipos de arquivo podem ser agrupados com barra vertical na mesma sequência.
	
	Exemplo:
	`file_type: "c|b"`

group	Esta palavra-chave é usada para especificar o grupo de um arquivo e vem sempre junto à palavra-chave **`file`**. A palavra-chave `group` pode ter um valor “none”, o que facilita a busca de arquivos sem proprietário. Exemplo:

	`group: "root"`
	
	O grupo também pode ser especificado com uma condição “OR” lógica por meio da seguinte sintaxe:
	
	`group : "root" || "bin" || "sys"`

ignore	Esta palavra-chave solicita que a verificação ignore os arquivos designados pela pesquisa. Esta palavra-chave está disponível para os tipos de verificação FILE_CHECK, FILE_CHECK_NOT, FILE_CONTENT_CHECK e FILE_CONTENT_CHECK_NOT.
	
	Exemplos:
	`# 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	Esta palavra-chave é usada para adicionar uma descrição mais detalhada à verificação que estiver sendo executada, como regulamentação, URL, política corporativa ou um motivo pelo qual a definição é necessária. Podem ser adicionados vários campos **`info`** em linhas separadas para a formatação do texto como um parágrafo. Não há limite predefinido para o número de campos **`info`** que podem ser usados. Exemplo: `info: "ref. CIS_AIX_Benchmark_v1.0.1.pdf ch 1, pg 28-29."`
levels	Esta palavra-chave é usada junto com CHKCONFIG para especificar os níveis de execução (runlevel) aos quais o serviço deve obedecer. Todos os níveis de execução devem ser descritos em uma única sequência. Por exemplo: para executar o serviço “sendmail” nos níveis 1, 2 e 3, o valor de **`levels`** correspondente na verificação CHKCONFIG será: `levels: "123"`
mask	Esta palavra-chave é o oposto de **`mode`**, em que é possível especificar permissões que **não** estejam disponíveis para um usuário, grupo ou outro membro específicos. Ao contrário de **`mode`**, que verifica um valor de permissão *exato*, as auditorias são mais abrangentes e verificam se um arquivo ou diretório se encontra em um nível igual ou mais seguro que o especificado pela auditoria **`mask`** (neste caso, **`mode`** pode reprovar um arquivo com permissão 640 por não corresponder a uma auditoria que espera um valor 644 e **`mask`** aceitará 640 como “mais seguro” e aprovará a auditoria). Exemplo: `mask: 022` Isto especifica qualquer permissão como OK para o proprietário e nenhuma permissão de gravação para o grupo ou outro membro. O valor **`mask`** “7” significa que não há permissão para usuários específicos, como o proprietário, grupo ou outro membro específico.
md5	Esta palavra-chave é usada em FILE_CHECK and FILE_CHECK_NOT para assegurar que o MD5 de um arquivo seja definido para qualquer conjunto de política.

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

mode	Esta palavra-chave descreve o conjunto de permissões para um arquivo/pasta a ser considerado. A palavra-chave `mode` pode ser representada no formato de string ou octal.
	
	Exemplos:
	`mode: "-rw-r--r--"`
	`mode: "644"`
	`mode: "7644"`

name	Esta palavra-chave é usada para a identificação do nome do processo em PROCESS_CHECK. Exemplo: `name: "syslogd"`
operator	Esta palavra-chave é usada junto com RPM_CHECK e PKG_CHECK para especificar a condição de aprovação ou reprovação de uma verificação com base na versão do pacote RPM instalado. Pode ter os seguintes valores: • **`lt`** (menor que) • **`lte`** (menor que ou igual a) • **`gte`** (maior que ou igual a) • **`gt`** (maior que) • **`eq`** (igual a) Exemplo: `operator: "lt"`

owner	Esta palavra-chave é usada para especificar o proprietário de um arquivo sempre combinada à palavra-chave **`file`**. A palavra-chave **`owner`** pode ter um valor “none”, o que facilita a busca de arquivos sem proprietário.
	
	Exemplo:
	`owner: "root"`
	
	A propriedade pode ser especificada também com uma condição “OR” lógica com a seguinte sintaxe:
	
	`owner: "root" || "bin" || "adm"`

regex	Esta palavra-chave possibilita a busca de um arquivo que corresponda a uma expressão regex específica. Exemplo: `regex: ".*LogLevel=9$"` Os metacaracteres a seguir requerem tratamento especial: + \ * ( ) ^

	Proteja os caracteres duas vezes com duas barras invertidas “\\” ou delimite-os entre colchetes “[]” se desejar que sejam interpretados de forma literal. Outros caracteres, como os abaixo, requerem apenas uma única barra invertida para serem interpretados de forma literal: . ? " ' Isto está relacionado com a maneira pela qual o compilador trata esses caracteres.
required	Esta palavra-chave é usada para especificar se o item auditado deve ou não estar presente no sistema remoto. Por exemplo: se **`required`** for definido como “No” (Não) e o **`type`** de verificação for “FILE_CHECK”, a verificação será aprovada se o arquivo existir e as permissões forem as especificadas no arquivo **`.audit`** ou se o arquivo não existir. Por outro lado, se **`required`** for definido como “Yes” (Sim), a verificação acima será reprovada.

rpm	Esta palavra-chave é usada para especificar a pesquisa de RPM quando usado junto com RPM_CHECK.
	
	Exemplo:
	`<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	Esta palavra-chave pode ser usada para especificar os locais investigáveis em um sistema de arquivos.
	
	Exemplo:
	`search_locations: "/bin"`
	
	Vários locais de busca podem ser agrupados com barra vertical.
	
	Exemplo:
	`search_locations: "/bin" | "/etc/init.d" | "/etc/rc0.d"`

service	Esta palavra-chave é usada junto com CHKCONFIG, XINETD_SVC and SVC_PROP e é usada para especificar o serviço que estiver sendo auditado.
	
	Exemplo:
	`<custom_item>`
	`type: CHKCONFIG`
	`description: "2.1 Disable Standard Services - Check if cups is disabled"`
	`service: "cups"`
	`levels: "123456"`
	`status: OFF`
	`</custom_item>`

severity	Em qualquer teste, **`<item>`** ou **`<custom_item>`**, um flag “**`severity`**” pode ser adicionado e definido como baixo (“LOW”), médio (“MEDIUM”) ou alto (“HIGH”). Por padrão, os resultados não conformes serão mostrados como “high”. Exemplo: `severity: MEDIUM`

status	Esta palavra-chave é usada em PROCESS_CHECK, CHKCONFIG e XINETD_SVC para se determinar se um serviço executado em um determinado host deve ser executado ou desativado. A palavra-chave **`status`** pode ter dois valores: ativado (“ON”) ou desativado (“OFF”).
	
	Exemplo:
	`status: ON`
	`status: OFF`

system	Esta palavra-chave especifica o tipo de sistema no qual a verificação deve ser executada.  A palavra-chave “**`system`**” é aplicável apenas a verificações “custom_item”, não a verificações de “item” incorporadas. Os valores disponíveis são aqueles que retornam com o comando “**`uname`**” no sistema operacional de destino. Por exemplo: no Solaris, o valor é “SunOS”, no Mac OS X é “Darwin”, no FreeBSD é “FreeBSD” etc. Exemplo: `system: "SunOS"`
timeout	Esta palavra-chave é usada junto com CMD_EXEC e especifica o intervalo de tempo, em segundos, que o comando especificado poderá ser executado antes de esgotar-se. Esta palavra-chave é usada nos casos em que um comando específico, como o comando Unix “**`find`**”, requer intervalos de tempo prolongados para a conclusão. Se esta palavra-chave não for especificada, o limite de tempo padrão para auditorias CMD_EXEC será de cinco minutos. Exemplo: `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	A palavra-chave **`value`** é usada para verificar se uma definição no sistema confirma o valor da política. Exemplo: `value: "90..max"` A palavra-chave **`value`** pode ser especificada como um intervalo [number..max]. Se o valor estiver entre o número especificado e “max”, a verificação será aprovada.

Itens personalizados

Um item personalizado é uma verificação completa definida com base nas palavras-chave definidas anteriormente. O exemplo a seguir consiste em uma lista de itens personalizados. Cada verificação começa com um tag "<**custom_item**>" e termina com "**</custom_item**>". As tags contêm listas de uma ou mais palavras-chave que são interpretadas pelo analisador de verificação de conformidade para executar as verificações.

As verificações de auditoria personalizadas usam "**</custom_item>**" e "**</item>**" de forma intercambiável para o tag de encerramento.

CHKCONFIG

A verificação de auditoria "CHKCONFIG" permite a interação com o utilitário "**chkconfig**" no sistema Red Hat remoto que estiver sendo auditado. Esta verificação consiste de cinco palavras-chave obrigatórias: **type**, **description**, **service**, **levels** e **status**.

A auditoria CHKCONFIG funciona apenas em sistemas Red Hat ou em um derivado de um sistema Red Hat, como o Fedora.

Exemplo:

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

CMD_EXEC

É possível executar comandos no host remoto e verificar se a saída corresponde ao esperado. Este tipo de verificação deve ser usado com extrema cautela, pois nem sempre é portável com todas as versões do Unix.

A palavra-chave **quiet** instrui o Nessus a **não** mostrar a saída do comando reprovado. Pode ser definida como "YES" (Sim) ou "NO" (Não). Normalmente, é definida como "NO" (Não) e o resultado do comando é exibido. Da mesma maneira, a palavra-chave "**dont_echo_cmd**" limita os resultados pela saída dos resultados do comando, mas não o comando propriamente dito.

A palavra-chave nosudo permite ao usuário instruir o Nessus a **não** usar sudo quando executar o comando ao defini-lo como "YES"(Sim). Normalmente, é definida como "NO"(Não) e sudo é sempre usado quando configurado para a ação.

Exemplo:

```
<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

As auditorias de conformidade do Unix normalmente verificam a existência e as configurações de um determinado arquivo. A auditoria “FILE CHECK” usa quatro ou mais palavras-chave para permitir a especificação destas verificações. As palavras-chave **`type`**, **`description`** e **`file`** são obrigatórias e são seguidas por uma ou mais verificações. A sintaxe atual aceita a verificação de permissões de proprietário, grupo e arquivo.

É possível usar globs em FILE_CHECK (por exemplo: **`/var/log/*`**). No entanto, observe que globs serão expandidos apenas para arquivos, não para diretórios. Se for especificado um glob e um ou mais arquivos com correspondência forem ignorados pela pesquisa, use a palavra-chave “**`ignore`**” para especificar os arquivos a serem ignorados.

As palavras-chave permitidas são:

```
uid: Numeric User ID (e.g., 0)
gid: Numeric Group ID (e.g., 500)
check_uneveness: YES
system: Tipo de sistema (por exemplo, Linux)
description: Descrição de texto da verificação de arquivo
file: Caminho completo e arquivo para verificação
(por exemplo, /etc/sysconfig/sendmail)
owner: Proprietário do arquivo (por exemplo, root)
group: Proprietário de grupo do arquivo (por exemplo, bin)
mode: Modo de permissão (por exemplo, 644)
mask: Máscara de arquivo (por exemplo, 133)
md5: O hash MD5 de um arquivo (por exemplo, 88d3dbe3760775a00b900a850b170fcd)
ignore: Um arquivo para ignorar (por exemplo, /var/log/secure)
attr: Um atributo de arquivo (por exemplo,  ----i--------)
```

As permissões de arquivo serão consideradas desiguais se “grupo” ou “outros” tiverem permissões diferentes de “proprietário” ou se “outros” tiver permissões diferentes de “grupo”.

Veja alguns exemplos a seguir:

```
<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

A auditoria “FILE_CHECK_NOT” consiste em três ou mais palavras-chave. As palavras-chave `type`, **`description`** e **`file`** são obrigatórias e são seguidas por uma ou mais verificações. A sintaxe atual aceita a verificação de permissões de proprietário, grupo e arquivo. Da mesma forma que a auditoria FILE_CHECK, a palavra-chave “**`ignore`**” pode ser usada para ignorar um ou mais arquivos se for especificado um glob de arquivo.

Esta função é o oposto de FILE_CHECK. Uma política é reprovada se não existir um arquivo ou se o seu modo for o mesmo que o definido na verificação propriamente dita.

É possível usar globs em FILE_CHECK_NOT (por exemplo: **`/var/log/*`**). No entanto, observe que globs serão expandidos apenas para arquivos, não para diretórios.

Veja alguns exemplos a seguir:

```
<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

Da mesma maneira que os testes e configurações de arquivo, o conteúdo de arquivos de texto também pode ser analisado. Podem ser usadas expressões regulares para a busca de um ou mais locais de conteúdo existente. Use a palavra-chave "**ignore**" para ignorar um ou mais arquivos do(s) local(is) de busca especificado(s).

O campo **string_required** pode ser definido para especificar se a sequência auditada que está sendo procurada deve estar presente ou não. Se esta opção não estiver definida, presume-se que seja necessária. O campo **file_required** pode ser definido para especificar se o arquivo auditado deve estar presente ou não. Se esta opção não estiver definida, presume-se que seja necessária.

Veja alguns exemplos a seguir:

```
<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>
```

```
<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"
 file_required: NO
 string_required: NO
```

```
</custom_item>
```

Ao adicionar “~” a um parâmetro, é possível fazer com que FILE_CONTENT_CHECK verifique o conteúdo não conforme dos diretórios locais do usuário.

```
<custom_item>
 system: "Linux"
 type: FILE_CONTENT_CHECK
 description: "Check all user home directories"
 file: "~/.rhosts"
 ignore: "/.foo"
 regex: "\\+"
 expect: "\\+"
</custom_item>
```

FILE_CONTENT_CHECK_NOT

Esta auditoria examina o conteúdo de um arquivo para corresponder à descrição de regex no campo **`regex`**. Esta função anula FILE_CONTENT_CHECK. Ou seja, uma política será reprovada se o regex **tiver** uma correspondência no arquivo. Use a palavra-chave “**`ignore`**” para ignorar um ou mais arquivos do(s) local(is) de busca especificado(s).

Este item da política verifica se o arquivo contém a expressão regular regex e se a expressão coincide com o campo expect.

O tipo permitido é:

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

Tanto regex quanto expect devem ser especificados nesta verificação.

Observe o exemplo a seguir:

```
<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: "^\+::"
 file_required: NO
 string_required: NO
</custom_item>
```

GRAMMAR_CHECK

A verificação da auditoria “GRAMMAR_CHECK” examina o conteúdo de um arquivo e corresponde a uma gramática definida de forma imprecisa (formada por uma ou várias declarações regex). Se **uma** linha no arquivo de destino não corresponder a qualquer declaração de regex, o teste será reprovado.

Exemplo:

```
<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

A auditoria “PKG_CHECK” executa um **pkgchk** em um sistema SunOS. A palavra-chave **pkg** é usada para especificar o pacote a ser pesquisado e a palavra-chave **operator** especifica a condição para que a verificação seja aprovada ou reprovada com base na versão do pacote instalado.

Exemplos:

```
<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

Do mesmo modo que as verificações de arquivo, é possível verificar os processos em execução de uma plataforma Unix auditada. A implementação executa o comando “**chkconfig –list**” para obter uma lista de processos de execução.

Exemplos:

```
<custom_item>
 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

A verificação da auditoria “RPM_CHECK” é usada para examinar os números de versão dos pacotes RPM instalados no sistema remoto Esta verificação consiste em quatro palavras-chave obrigatórias (**type**, **description**, **rpm**, **operator**) e na palavra-chave **required** opcional. A palavra-chave **rpm** é usada para especificar o pacote a ser pesquisado e a palavra-chave **operator** especifica a condição para que a verificação seja aprovada ou reprovada com base na versão do pacote RPM instalado.

O uso das verificações RPM não é portável em todas as distribuições do Linux. Portanto, o uso de RPM_CHECK não é recomendável neste caso.

Os exemplos apresentados a seguir pressupõem que o **iproute-2.4.7-10** esteja instalado:

```
<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>
```

```
<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

A verificação da auditoria “SVC_PROP” permite que o usuário interaja com a ferramenta “**svcprop -p**” em um sistema Solaris 10. Pode ser usada para a consulta das propriedades associadas a um serviço específico. A palavra-chave **service** é usada para a especificação do serviço que estiver sendo auditado. A palavra-chave **property** especifica o nome da propriedade a ser consultada. A palavra-chave **value** é o valor esperado da propriedade. O valor esperado pode ser também uma regex.

O campo **svcprop_option** pode ser definido para especificar se a sequência auditada que está sendo procurada deve estar presente ou não. O acesso a este campo possui CAN_BE_NULL ou CANNOT_BE_NULL como argumentos.

Exemplos:

```
<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>
```

```
<custom_item>
 type: SVC_PROP
 description: "Check if ipfilter is enabled - can be missing or not found"
 service: "network/ipfilter:default"
 property: "general/enabled"
 value: "true"
svcprop_option: CAN_BE_NULL
</custom_item>
```

XINETD_SVC

A verificação da auditoria “XINETD_SVC” é usada para a auditoria do status da inicialização dos serviços xinetd. Esta verificação consiste nas quatro palavras-chave obrigatórias: **type**, **description**, **service** e **status**.

Esta verificação funciona apenas em sistemas Red Hat ou um sistema Red Hat derivado, como o Fedora.

Exemplo:

```
<custom_item>
 type: XINETD_SVC
 description: "Make sure that telnet is disabled"
 service: "telnet"
 status: OFF
```

```
</custom_item>
```

Verificações incorporadas

As verificações que não forem abrangidas pelas verificações acima devem ser gravadas como nomes personalizados em NASL. Todas as verificações enquadram-se na categoria “incorporadas”. Cada verificação começa com um tag “<**item**>” e termina com “</**item**>”. As tags contêm listas de uma ou mais palavras-chave que são interpretadas pelo analisador de verificação de conformidade para executar as verificações. O exemplo a seguir apresenta uma lista de verificações disponíveis.

A palavra-chave “**system**” não está disponível para as verificações incorporadas e resultará em erro de sintaxe se for usada.

Gerenciamento de senhas

Nos exemplos abaixo, são usados <min> e <max> para representar um valor inteiro e não um string para usar nos dados do valor da auditoria.

Nos casos em que o valor mínimo ou máximo exato não for conhecido, substitua as strings “Min” ou “Max” pelo valor inteiro.

min_password_length

Uso

```
<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>
```

Esta verificação incorporada assegura que o comprimento mínimo da senha aplicado ao sistema remoto fique dentro no intervalo “**<min>..<max>**”. O comprimento mínimo de senha força os usuários a escolher senhas mais complexas.

Sistema operacional	Execução
Linux	O comprimento mínimo da senha é definido como PASS_MIN_LEN em **/etc/login.defs**.
Solaris	O comprimento mínimo da senha é definido como PASSLENGTH em **/etc/default/passwd**. Observe que isto também controla o comprimento máximo da senha.
HP-UX	O comprimento mínimo da senha é definido como MIN_PASSWORD_LENGTH em **/etc/default/security**.

Mac OS X	O comprimento mínimo da senha é definido como “minChar” na política local, definido com o uso do comando **`pwpolicy`**.

Exemplo:

```
 <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

Uso

```
<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>
```

Esta função incorporada assegura que a idade máxima da senha (por exemplo: o momento em que os usuários são forçados a mudar suas senhas) fique no intervalo definido.

A validade máxima da senha evita que os usuários mantenham a mesma senha por vários anos. A alteração de senhas evita que um hacker obtenha uma senha e use-a indefinidamente.

Sistema operacional	Execução
Linux	A variável PASS_MAX_DAYS é definida em **`/etc/login.defs`**.
Solaris	A variável MAXWEEKS em **`/etc/default/passwd`** define o número máximo de semanas que uma senha pode ser usada.
HP-UX	Este valor é controlado pela variável PASSWORD_MAXDAYS em **`/etc/default/security`**.
Mac OS X	A opção “maxMinutesUntilChangePassword” da política de senhas (definida pela ferramenta **`pwpolicy`**) pode ser usada para a definição deste valor.

Exemplo:

```
 <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

Uso
<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>

Esta função incorporada assegura que a validade mínima da senha (por exemplo: o momento necessário antes que os usuários tenham permissão de alterar suas senhas) fique no intervalo definido.

A validade mínima da senha evita que os usuários alterem senhas com frequência na tentativa de neutralizar o histórico de senha máximo. Alguns usuários executam o ciclo para obter a sua senha original, evitando as exigências de mudança de senha.

Sistema operacional	Execução
Linux	A variável PASS_MAX_DAYS é definida em **/etc/login.defs**.
Solaris	A variável MINWEEKS em **/etc/default/passwd** define o número máximo de semanas que uma senha pode ser usada.
HP-UX	Este valor é controlado pela variável PASSWORD_MINDAYS em **/etc/default/security**.
Mac OS X	Esta opção não é aceita.

Exemplo:

```
<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>
```

Root Access

root_login_from_console

Uso
<item> name: "root_login_from_console" description: "This check makes sure that root can only log in from the system console (not remotely)." </item>

Esta função incorporada assegura que o usuário “raiz” só possa efetuar login diretamente no sistema remoto por meio do console físico.

O fundamento lógico por trás desta verificação é que boas práticas administrativas não recomendam o uso direto da conta raiz para investigação do acesso de uma pessoa específica. Em vez disso, use uma conta de usuário genérico (membro do grupo wheel em sistemas BSD) e use “**su**” (ou **sudo**) para elevar os privilégios de execução de tarefas administrativas.

Sistema operacional	Execução
Linux and HP-UX	Certifique-se de que **/etc/securetty** exista e que contenha somente “console”.
Solaris	Certifique-se de que **CONSOLE=/dev/console** contenha a linha “**/etc/default/login**”.
Mac OS X	Esta opção não é aceita.

Permissions Management

accounts_bad_home_permissions

Uso

`<item>`
` name: "accounts_bad_home_permissions"`
` description: "This check reports user accounts that have home directories with incorrect user or group ownerships."`
`</item>`

Esta função incorporada assegura que o diretório local de cada usuário sem privilégios pertença ao usuário e que usuários de terceiros não possam alterá-lo (mesmo que pertençam ao mesmo grupo ou “todos”). Geralmente, recomenda-se que os diretórios locais do usuário sejam definidos no modo 0755 ou mais rigoroso (por exemplo: 0700). Este teste será bem-sucedido se cada diretório local for configurado corretamente, caso contrário, será reprovado. Qualquer uma das palavras-chave **mode** ou **mask** pode ser usada para especificar os níveis de permissão desejados para diretórios locais. A palavra-chave **mode** aceitará diretórios locais que correspondam exatamente a um nível especificado e a palavra-chave **mask** aceitará diretórios locais que estejam no nível especificado ou mais seguro.

Se usuários externos conseguirem alterar o diretório local de um usuário, poderão forçar o usuário a executar comandos arbitrários pela adulteração dos arquivos **~/.profile**, **~/.cshrc**, **~/.bashrc**.

Se for necessário compartilhar arquivos entre usuários do mesmo grupo, normalmente será recomendável o uso de um diretório dedicado que possa receber gravações para o grupo, mas não o diretório local de um usuário.

No caso de diretórios locais mal configurados, execute “**chmod 0755 <user directory>**” e modifique a propriedade de maneira correspondente.

accounts_bad_home_group_permissions

Uso

`<item>`
` name: "accounts_bad_home_group_permissions"`
` description: "This check makes sure user home directories are group owned by the user's primary group."`

```
</item>
```

A operação desta função incorporada é semelhante a **accounts_bad_home_permissions**, mas garante que os diretórios locais do usuário sejam do grupo que pertence ao grupo principal do usuário.

accounts_without_home_dir

Uso

```
<item>
 name: "accounts_without_home_dir"
 description: "This check reports user accounts that do not have home directories."
</item>
```

Esta função incorporada assegura que todo e qualquer usuário possua um diretório local. Será aprovada se um diretório válido for atribuído a cada usuário e será reprovada se ocorrer o contrário. Observe que esta verificação não verifique a propriedade ou as permissões do diretório local.

Geralmente, recomenda-se que cada usuário em um sistema possua um diretório local definido, pois algumas ferramentas podem precisar lê-lo ou alterá-lo (por exemplo: **sendmail** verifica um arquivo **~/.forward**). Se um usuário não efetuar o login, um shell inexistente deverá ser definido (por exemplo: **/bin/false**). Em muitos sistemas, um usuário sem diretório local algum terá privilégios de login concedidos, mas seu diretório local efetivo será /.

invalid_login_shells

Uso

```
<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>
```

Esta função incorporada assegura que cada usuário possua um shell válido, conforme definido em **/etc/shells**.

O arquivo **/etc/shells** é usado por aplicativos como Sendmail e servidores FTP para determinar se um shell é válido no sistema. Embora não seja usado pelo programa de login, os administradores poderão usar o arquivo para definir quais são os shells válidos no sistema. A verificação de **invalid_login_shells** determina se todos os usuários no arquivo **/etc/passwd** estão configurados com shells válidos, conforme definido no arquivo **/etc/shells**.

Isto evita práticas não aprovadas, como o uso de **/sbin/passwd** como um shell para permitir que usuários modifiquem suas senhas. Se não desejar que um usuário efetue o login, crie um shell inválido em **/etc/shells** (por exemplo: “**/nonexistent**”) e defina-o para os usuários desejados.

Se houver usuários sem um shell válido, defina um shell válido para eles.

login_shells_with_suid

Uso

```
<item>
 name: "login_shells_with_suid"
 description: "This check reports user accounts with login shells that have setuid or
setgid privileges."
</item>
```

Esta função incorporada assegura que nenhum shell possua recursos “set-uid”.

Um shell “setuid” significa que sempre que o shell for iniciado, o próprio processo terá os privilégios definidos para suas permissões (por exemplo: um shell “root” setuid concede privilégios de superusuário para qualquer pessoa).

O shell “setuid” anula a finalidade da existência de UIDs e GIDs e torna muito mais complexo o controle de acesso.

Remova o bit SUID de cada shell que seja “setuid”.

login_shells_writeable

Uso

```
<item>
 name: "login_shells_writeable"
 description: "This check reports user accounts with login shells that have group or
world write permissions."
</item>
```

Esta função incorporada assegura que nenhum shell seja alterável por qualquer usuário mundial/grupo.

Se um shell for alterável por qualquer pessoa do mundo (ou do grupo), os usuários sem privilégios poderão substituí-lo por qualquer programa. Isto permite que um usuário mal-intencionado force outros usuários do shell a executar comandos arbitrários ao efetuarem o login.

Certifique-se de que cada shell seja definido da forma correta.

login_shells_bad_owner

Uso

```
<item>
 name: "login_shells_bad_owner"
 description: "This check reports user accounts with login shells that are not owned by
root or bin."
</item>
```

Esta função incorporada garante que cada shell pertença aos usuários “root” ou “bin”.

Da mesma maneira que os shells com permissões inválidas, se um usuário possuir um shell usado por outros usuários, poderão modificá-lo para forçar usuários de terceiros a executar comandos arbitrários ao efetuarem o login.

Somente usuários “root” e/ou “bin” podem modificar binários no âmbito do sistema.

Gerenciamento de arquivos de senha

passwd_file_consistency

Uso

```
<item>
 name: "passwd_file_consistency"
 description: "This check makes sure /etc/passwd is valid."
</item>
```

Esta função incorporada garante que cada linha em **`/etc/passwd`** tenha um formato válido (por exemplo: sete campos separados por dois pontos). Se a linha estiver malformada, será relatada e a verificação será reprovada.

Um arquivo **`/etc/passwd`** malformado poderá interromper várias ferramentas de gerenciamento de usuários. Também pode indicar uma invasão ou uma falha (bug) em um aplicativo personalizado de gerenciamento de usuários. Além disso, pode indicar que alguém tentou adicionar um usuário com um nome inválido (antigamente, era comum criar um usuário denominado “toor:0:0” para obter privilégios root).

Se o teste for considerado não conforme, o administrador deverá remover ou corrigir as linhas transgressoras de **`/etc/passwd`**.

passwd_zero_uid

Uso

```
<item>
 name: "passwd_zero_uid"
 description: "This check makes sure that only ONE account has a uid of 0."
</item>
```

Esta função incorporada garante que somente uma conta possua um UID “0” em **`/etc/passwd`**. A função foi concebida e reservada para a conta “root”, mas é possível acrescentar contas adicionais com UID 0, que teriam o mesmo acesso com privilégios. Este teste será bem-sucedido se apenas uma conta possuir um UID zero, do contrário, resultará em reprovação.

Um UID “0” concede privilégios root no sistema. Um usuário raiz pode realizar qualquer ação no sistema, o que normalmente inclui investigação da memória de outros processos (ou do kernel), leitura e gravação em qualquer arquivo existente no sistema e assim por diante. Por ser tão poderosa, o uso da conta deve ser restrito quase ao mínimo e ser bem protegida.

As boas práticas administrativas determinam que cada UID seja exclusivo (por isso o “U” em UID). A posse de duas (ou mais) contas com privilégios “root” anula a responsabilidade do administrador de sistemas com relação ao sistema. Além disso, uma vez que muitos sistemas restringem o login direto do root ao console somente, o uso administrativo pode ser rastreado. Normalmente, os administradores de sistema devem efetuar login em sua própria conta e usar o comando `su` para terem privilégios de root. Uma conta adicional UID 0 adicional evita esta restrição.

Se o acesso “root” precisar ser compartilhado entre usuários, use uma ferramenta como **`sudo`** ou **`calife`** (ou RBAC no Solaris). Deverá existir apenas uma conta com UID “0”.

passwd_duplicate_uid

Uso

```
<item>
 name: "passwd_duplicate_uid"
 description: "This check makes sure that every UID in /etc/passwd is unique."
</item>
```

Esta função incorporada garante que toda e qualquer conta listada em **/etc/passwd** tenha um UID exclusivo. Este teste será bem-sucedido se cada UID for exclusivo, caso contrário, será reprovado.

Todos os usuários em um sistema Unix são identificados por seu identificador de usuário (UID), ou seja, um número entre 0 e 65.535. Se dois usuários compartilharem o mesmo UID, além de terem os mesmos privilégios concedidos, o sistema irá tratá-los como se fossem a mesma pessoa. Isto anula qualquer nível de responsabilidade, pois é impossível dizer quais ações foram executadas por qual usuário (normalmente, o sistema fará uma pesquisa reversa do UID e usará o primeiro nome das contas que compartilham o UID ao exibir logs).

As normas de segurança, como referenciais CIS, impedem o compartilhamento de um UID entre usuários. Se os usuários precisarem compartilhar arquivos, deverão usar grupos.

Conceda a cada usuário do sistema um ID exclusivo.

passwd_duplicate_gid

Uso

```
<item>
 name: "passwd_duplicate_gid"
 description: "This check makes sure that every GID in /etc/passwd is unique."
</item>
```

Esta função incorporada garante que o ID de grupo (GID) principal de cada usuário seja exclusivo. O teste será aprovado se cada usuário tiver um GID exclusivo, do contrário, será reprovado.

As normas de segurança recomendam a criação de um grupo por usuário (normalmente com o mesmo nome de usuário). Com esta configuração, os arquivos criados pelo usuário normalmente são “originalmente seguros”, pois pertencem a seu grupo principal e, portanto, só podem ser modificados pelo próprio usuário. Se o usuário desejar que o arquivo seja de propriedade dos outros membros de um grupo, deverá usar o comando **chgrp** de forma explícita para alterar a propriedade.

Outra vantagem deste método é que ele unifica o gerenciamento de participação no grupo em um único arquivo (**/etc/group**), em vez de combinar **/etc/passwd** e **/etc/group**.

Para cada usuário, crie um grupo com o mesmo nome. Gerencie a posse do grupo somente por meio de **/etc/group**.

passwd_duplicate_username

Uso

```
<item>
 name: "passwd_duplicate_username"
 description: "This check makes sure that every username in /etc/passwd is unique."
```

```
</item>
```

Esta função incorporada assegura que cada nome de usuário em **/etc/passwd** seja exclusivo. Será aprovada se for este o caso, do contrário, será reprovada.

Os nomes de usuário duplicados em **/etc/passwd** geram problemas, pois os privilégios da conta usados não são definidos.

O comando **adduser** impedirá a criação de um nome de usuário duplicado. Essa configuração geralmente significa que o sistema foi comprometido, que as ferramentas para gerenciamento de usuários estão com falha ou que o arquivo **/etc/passwd** foi editado manualmente.

Exclua os nomes de usuário duplicados ou modifique-os para ficarem diferentes.

passwd_duplicate_home

Uso

```
<item>
 name: "passwd_duplicate_home"
 description: "(arbitrary user comment)"
</item>
```

Esta função incorporada garante que cada usuário não pertencente ao sistema (cujo UID seja superior a 100) em **/etc/passwd** possua um diretório local exclusivo.

Cada nome de usuário em **/etc/passwd** deve ter um diretório local exclusivo. Se os usuários compartilharem o mesmo diretório local, um poderá forçar o outro a executar comandos arbitrários pela modificação dos arquivos de inicialização (**.profile** etc.) ou pela colocação de binários mal-intencionados no próprio diretório local. Além disso, um diretório local compartilhado anula a responsabilização do usuário.

Os requisitos de conformidade obrigam que cada usuário possua um diretório local exclusivo.

passwd_shadowed

Uso

```
<item>
 name: "passwd_shadowed"
 description: "(arbitrary user comment)"
</item>
```

Esta verificação incorporada garante que toda e qualquer senha em **/etc/passwd** esteja “sombreada” (ou seja, reside em outro arquivo).

Uma vez que /**etc/passwd** pode ser lida por qualquer pessoa em qualquer lugar, o armazenamento de hashes de senhas dos usuários permite que qualquer pessoa com acesso à função execute programas para violação de senhas. As tentativas de identificação da senha de um usuário por meio de ataque por força bruta (tentativas de login repetidas com o uso sequencial de senhas distintas) são geralmente detectadas em arquivos de log do sistema. Se o arquivo **/etc/passwd** contiver os hashes de senha, o arquivo pode ser copiado offline e usado como entrada de dados em um programa para violação de senhas. Isto permite que um invasor obtenha senhas de usuário sem detecção.

A maior parte dos sistemas Unix possui arquivos de senha sombreados. Consulte a documentação do sistema para saber como habilitar senhas sombreadas em seu sistema.

passwd_invalid_gid

Uso

```
<item>
 name: "passwd_invalid_gid"
 description: "This check makes sure that every GID defined in /etc/passwd exists in
/etc/group."
</item>
```

Esta função incorporada garante que cada ID de grupo (GID) listado em **/etc/passwd** exista em **/etc/group**. Será aprovada se cada GID for definido corretamente, caso contrário, será reprovada.

Cada vez que um ID de grupo for definido em **/etc/passwd**, deverá ser imediatamente listado em **/etc/group**. Do contrário, o sistema ficará em um estado inconsistente e poderão surgir problemas.

Considere a seguinte situação: o usuário (“bob”) possui um UID 1000 e GID 4000. O GID não está definido em **/etc/group** e, portanto, o grupo principal do usuário não concede a ele nenhum privilégio atualmente. Alguns meses depois, o administrador do sistema edita **/etc/group** e adiciona o grupo “admin”, selecionando o GID “unused” 4000 para identificá-lo. Dessa data em diante o usuário “bob” pertence ao grupo “admin”, mesmo que tenha não tenha sido intencional.

Edite **/etc/group** para adicionar os GIDs que faltam.

Gerenciamento de arquivos de grupo

group_file_consistency

Uso

```
<item>
 name: "group_file_consistency"
 description: "This check makes sure /etc/group is valid."
</item>
```

Esta função incorporada garante que cada linha em **/etc/group** tenha um formato válido (por exemplo: três campos separados por dois pontos e uma lista de usuários). Se a linha estiver malformada, será relatada e a verificação será reprovada.

Um arquivo **/etc/group** malformado poderá interromper várias ferramentas de gerenciamento de usuários. Também pode indicar uma invasão ou uma falha (bug) em um aplicativo personalizado de gerenciamento de usuários. Isto também demonstra que alguém tentou adicionar um usuário com um nome de grupo válido.

Edite o arquivo **/etc/group file** to para corrigir as linhas malformadas.

group_zero_gid

Uso

```
<item>
 name: "group_zero_gid"
```

```
 description: "This check makes sure that only ONE group has a gid of 0."
</item>
```

Esta função incorporada garante que somente um grupo tenha um ID de grupo (GID) 0. A função será aprovada se apenas um grupo tiver um GID 0, do contrário, será reprovada.

O GID “0” significa que os usuários membros do grupo também são membros do grupo raiz principal. Isto concede a eles privilégios root em quaisquer arquivos com permissões de grupo raiz.

Para definir um grupo de administradores, crie um grupo “admin”.

group_duplicate_name

Uso

```
<item>
 name: "group_duplicate_name"
 description: "This check makes sure that every group name in /etc/group is unique."
</item>
```

Esta verificação incorporada assegura que cada nome de grupo seja exclusivo. Será aprovada se for este o caso, do contrário, será reprovada.

Os nomes de grupo de usuários duplicados em **`/etc/group`** geram problemas, pois é impossível saber quais privilégios do grupo são usados. Isto significa que um nome de grupo duplicado pode conter membros ou privilégios que não deveria conter inicialmente.

Exclua ou renomeie nomes de grupo duplicados.

group_duplicate_gid

Uso

```
<item>
 name: "group_duplicate_gid"
 description: "(arbitrary user comment)"
</item>
```

Cada grupo em um sistema Unix é identificado por seu identificador de grupo (GID), ou seja, um número entre 0 e 65535. Se dois grupos compartilharem o mesmo GID, serão concedidos a eles mesmos privilégios e o sistema irá tratá-los como se fossem o mesmo grupo. Isto anula a finalidade de usar grupos para a segregação de privilégios do usuário.

As normas de segurança proíbem o compartilhamento de um GID entre grupos. Se dois grupos precisarem ter os mesmos privilégios, deverão ter os mesmos usuários.

Exclua os grupos duplicados ou atribua a uma das duplicatas um novo GID exclusivo.

group_duplicate_members

Uso

```
<item>
 name: "group_duplicate_members"
```

```
 description: "This check makes sure that every member of a group is listed once."
</item>
```

Esta função incorporada garante que cada membro de um grupo seja listado apenas uma vez. A função será aprovada se cada membro for exclusivo e reprovada se ocorrer o inverso.

Cada membro do grupo deve ser listado uma só vez. Embora a listagem repetida não cause problemas ao sistema operacional, isto pode dificultar a tarefa do administrador de sistemas, pois a revogação de privilégios torna-se mais complexa. Por exemplo: se o grupo “admin” tiver os membros “alice,bob,charles,daniel,bob”, “bob” deverá ser removido duas vezes se os privilégios forem revogados.

Certifique-se de que cada membro seja listado somente uma vez.

group_nonexistant_users

Uso

```
<item>
 name: "group_nonexistant_users"
 description: "This check makes sure that every member of a group actually exists."
</item>
```

Esta verificação garante que cada membro de um grupo realmente exista em **`/etc/passwd`**.

A inexistência de usuários em **`/etc/group`** indica que as práticas de administração não estão concluídas. Neste caso, o usuário não existe porque seu nome foi inserido incorretamente ou porque o usuário não foi removido do grupo ao ser removido do sistema.

Não é recomendável manter usuários “ocultos” em **`/etc/group`**. Se um usuário com o mesmo nome for adicionado posteriormente, poderá ter privilégios de grupo que não deveriam ser concedidos.

Remova os usuários não existentes de **`/etc/group`**.

Ambiente Root

dot_in_root_path_variable

Uso

```
<item>
 name: "dot_in_root_path_variable"
 description: "This check makes sure that root's $PATH variable does not contain any
relative path."
</item>
```

Esta verificação garante que o diretório de trabalho atual (“.”) não esteja incluído no caminho executável do usuário root. Isto previne que um usuário mal-intencionado eleve os privilégios para superusuário ao fazer com que um administrador conectado como "root" execute um cavalo de Tróia que pode ser instalado no diretório de trabalho atual.

writeable_dirs_in_root_path_variable

Uso

```
<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>
```

Esta verificação relata todos os diretórios alteráveis por qualquer pessoa em qualquer lugar ou grupo na variável PATH de usuários root. Todos os diretórios que retornam após esta verificação devem ser examinados cuidadosamente e as permissões alteráveis nos diretórios por local/grupo desnecessárias devem ser removidas conforme o exemplo a seguir:

```
 # chmod go-w path/to/directory
```

Permissões de arquivos

find_orphan_files

Uso

```
<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>
```

Esta verificação relata todos os arquivos que não têm proprietário no sistema.

Normalmente, a pesquisa é feita de maneira recorrente no diretório “/”. Isto torna a verificação extremamente lenta de executar, dependendo do número de arquivos presentes no sistema remoto. No entanto, se necessário, o diretório básico padrão poderá ser alterado com o uso da palavra-chave opcional **basedir**. Também é possível ignorar a busca de determinados arquivos dentro de um diretório base com o uso de outra palavra-chave opcional **ignore**. Ao buscar sistemas de arquivos, todos os diretórios NFS instalados, a menos que sejam especificados com a palavra-chave **dir** opcional, serão ignorados.

Devido à natureza da verificação, é normal mantê-la em execução por algumas horas, dependendo do tipo de sistema avaliado. Um valor de limite de tempo padrão, que é o tempo após o qual o Nessus deve parar de processar resultados da verificação, foi definido em cinco horas e não pode ser alterado.

Exemplo:

```
 <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

Uso

```
<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>
```

Esta verificação relata todos os arquivos alteráveis no sistema remoto por qualquer pessoa. Não dever haver nenhum arquivo alterável por qualquer pessoa no sistema remoto como, por exemplo, o resultado da verificação não deve exibir nenhum conteúdo. No entanto, em alguns casos, dependendo das necessidades organizacionais, os arquivos podem ser alterados por qualquer pessoa em qualquer lugar. Todos os itens que retornam desta verificação devem ser auditados cuidadosamente e os arquivos que não requerem atributos alteráveis por qualquer pessoa deverão ser removidos conforme o exemplo a seguir:

```
 # chmod o-w world_writeable_file
```

Normalmente, a pesquisa é feita de maneira recorrente no diretório “/”. Isto torna a verificação extremamente lenta de executar, dependendo do número de arquivos presentes no sistema remoto. No entanto, se necessário, o diretório básico padrão poderá ser alterado com o uso da palavra-chave opcional **basedir**. Também é possível ignorar a busca de determinados arquivos dentro de um diretório base com o uso de outra palavra-chave opcional **ignore**. Ao buscar sistemas de arquivos, todos os diretórios NFS instalados, a menos que sejam especificados com a palavra-chave **dir** opcional, serão ignorados.

Devido à natureza da verificação, é normal mantê-la em execução por algumas horas, dependendo do tipo de sistema avaliado. Um valor de limite de tempo padrão, que é o tempo após o qual o Nessus deve parar de processar os resultados da verificação, foi definido em cinco horas e o valor não pode ser alterado.

Exemplo:

```
 <item>
  name: "find_world_writeable_files"
  description: "Search for world-writable files"
  # Globs allowed (? and *)
  basedir: "/tmp"
  ignore: "/tmp/foo"
  ignore: "/tmp/bar"
 </item>
```

find_world_writeable_directories

Uso

```
<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>
```

Esta verificação relata todos os diretórios que forem alteráveis por qualquer pessoa do mundo e cujo sticky bit não esteja definido no sistema remoto. A verificação da definição do sticky bit em todos os diretórios alteráveis por qualquer pessoa garante que somente o proprietário de arquivo dentro de um diretório possa excluir o arquivo. Isto evita que qualquer outro usuário exclua o arquivo de maneira acidental ou intencional.

Normalmente, a pesquisa é feita de maneira recorrente no diretório "/". Isto torna a verificação extremamente lenta de executar, dependendo do número de arquivos presentes no sistema remoto. No entanto, se necessário, o diretório básico padrão poderá ser alterado com o uso da palavra-chave opcional **basedir**. Também é possível ignorar a busca de determinados arquivos dentro de um diretório base com o uso de outra palavra-chave opcional **ignore**. Ao buscar sistemas de arquivos, todos os diretórios NFS instalados, a menos que sejam especificados com a palavra-chave **dir** opcional, serão ignorados.

Devido à natureza da verificação, é normal mantê-la em execução por algumas horas, dependendo do tipo de sistema avaliado. Um valor de limite de tempo padrão, que é o tempo após o qual o Nessus deve parar de processar os resultados da verificação, foi definido em cinco horas e o valor não pode ser alterado.

Exemplo:

```
  <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>
```

find_world_readable_files

Uso

```
<item>
 name: "find_world_readable"
 description: "This check finds all the files in a directory with world readable
permissions."
 # Globs allowed (? and *)
 (optional) basedir: "<directory>"
 (optional) ignore: "<directory>"
```

```
 (optional) dir: "<directory>"
</item>
```

Esta verificação informa todos os arquivos que podem ser lidos. A verificação dos arquivos legíveis, por exemplo, em diretórios locais do usuário, impede que os arquivos confidenciais sejam acessados por outros usuários (por exemplo: chaves privadas SSH).

Normalmente, a pesquisa é feita de maneira recorrente no diretório "/". Isso torna a verificação extremamente lenta de executar, dependendo do número de arquivos presentes no sistema remoto. No entanto, se necessário, o diretório básico padrão poderá ser alterado com o uso da palavra-chave opcional **basedir**. Também é possível ignorar a busca de determinados arquivos dentro de um diretório base com o uso de outra palavra-chave opcional **ignore**. Ao buscar sistemas de arquivos, todos os diretórios NFS instalados (a menos que sejam especificados com a palavra-chave **dir** opcional) serão ignorados.

Devido à natureza da verificação, é normal mantê-la em execução por algumas horas, dependendo do tipo de sistema avaliado. Um valor de limite de tempo padrão, que é o tempo após o qual o Nessus deve parar de processar os resultados da verificação, foi definido em cinco horas e o valor não pode ser alterado.

Exemplo:

```
 <item>
  name: "find_world_readable_files"
  description: "This check finds all the files in a directory with world readable
        permissions."
  basedir: "/home"
  ignore: "/home/tmp"
  dir: "/home/extended"
 </item>
```

find_suid_sgid_files

Uso

```
<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>
```

Esta verificação relata todos os arquivos com o conjunto de bit SUID/SGID. Todos os arquivos relatados por esta verificação devem ser auditados cuidadosamente, principalmente os scripts de shell e executáveis locais/internos como, por exemplo, executáveis que não acompanham o sistema. Os arquivos SUID/SGID apresentam o risco de elevação de privilégios de um usuário normal para privilégios de proprietário ou grupo do arquivo. Se os arquivos/scripts forem necessários, deverão ser examinados de forma especial para que se verifique se permitem criar arquivo com privilégios elevados.

Normalmente, a pesquisa é feita de maneira recorrente no diretório "/". Isto torna a verificação extremamente lenta de executar, dependendo do número de arquivos presentes no sistema remoto. No entanto, se necessário, o diretório básico padrão poderá ser alterado com o uso da palavra-chave opcional **basedir**. Também é possível ignorar a busca de determinados arquivos dentro de um diretório base com o uso de outra palavra-chave opcional `ignore`. Ao buscar sistemas de arquivos, todos os diretórios NFS instalados, a menos que sejam especificados com a palavra-chave **dir** opcional, serão ignorados.

Devido à natureza da verificação, é normal mantê-la em execução por algumas horas, dependendo do tipo de sistema avaliado. Um valor de limite de tempo padrão, que é o tempo após o qual o Nessus deve parar de processar os resultados da verificação, foi definido em cinco horas e o valor não pode ser alterado.

Exemplo:

```
<item>
 name: "find_suid_sgid_files"
 description: "Search for SUID/SGID files"
 # Globs allowed (? and *)
 basedir: "/"
 ignore: "/usr/sbin/ping"
</item>
```

home_dir_localization_files_user_check
Esta opção incorporada verifica se um arquivo de localização dentro do diretório local do usuário é de propriedade do usuário ou da raiz.

Um ou mais arquivos podem ser listados com o token “file”. No entanto, se o token “file” estiver ausente, a verificação pesquisará os seguintes arquivos, por padrão:

- **.login**
- **.cschrc**
- **.logout**
- **.profile**
- **.bash_profile**
- **.bashrc**
- **.bash_logout**
- **.env**
- **.dtprofile**
- **.dispatch**
- **.emacs**
- **.exrc**

Exemplo:

```
<item>
 name: "home_dir_localization_files_user_check"
 description: "Check file .foo/.foo2"
 file: ".foo"
 file: ".foo2"
 file: ".foo3"
</item>
```

home_dir_localization_files_group_check
Esta opção incorporada verifica se um arquivo de localização dentro do diretório local do usuário é um grupo de propriedade do grupo principal do usuário ou da raiz.

Um ou mais arquivos podem ser listados com o token “file”. No entanto, se o token “file” estiver ausente, a verificação pesquisará os seguintes arquivos, por padrão:

- **.login**
- **.cschrc**
- **.logout**
- **.profile**
- **.bash_profile**
- **.bashrc**
- **.bash_logout**
- **.env**
- **.dtprofile**
- **.dispatch**
- **.emacs**
- **.exrc**

Exemplo:

```
 <item>
  name: "home_dir_localization_files_group_check"
  description: "Check file .foo/.foo2"
  file: ".foo"
  file: ".foo2"
  file: ".foo3"
 </item>
```

Conteúdo de arquivos suspeitos

admin_accounts_in_ftpusers

Uso

```
<item>
 name: "admin_accounts_in_ftpusers"
 description: "This check makes sure every account whose UID is below 500 is present in
/etc/ftpusers."
</item>
```

Esta verificação audita se todas as contas administrativas e usuários com UID inferior a 500 estão presentes em **/etc/ftpusers**, **/etc/ftpd/ftpusers** ou **/etc/vsftpd.ftpusers**.

Arquivos desnecessários
find_pre-CIS_files

Uso

```
<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>
```

Esta verificação foi adaptada para um Centro de Segurança na Internet (CIS) específico para a aprovação da certificação do referencial Red Hat CIS. Esta verificação é útil para usuários com sistemas Red Hat configurados/otimizados com base no referencial Red Hat CIS. A ferramenta de referencial CIS fornece um script de backup para de todos os arquivos do sistema modificados durante o processo de otimização e terão como sufixo a palavra-chave "–**preCIS**". Os arquivos devem ser removidos assim que todas as recomendações do referencial forem aplicadas e o sistema restaurado à sua condição de funcionamento. Esta verificação assegura que nenhum arquivo "**preCIS**" exista no sistema remoto.

Normalmente, a pesquisa é feita de maneira recorrente no diretório "/". Isto torna a verificação extremamente lenta de executar, dependendo do número de arquivos presentes no sistema remoto. No entanto, se necessário, o diretório básico padrão poderá ser alterado com o uso da palavra-chave opcional **basedir**. Também é possível ignorar a busca de determinados arquivos dentro de um diretório base com o uso de outra palavra-chave opcional **ignore**.

Devido à natureza da verificação, é normal mantê-la em execução por algumas horas, dependendo do tipo de sistema avaliado. Um valor de limite de tempo padrão, que é o tempo após o qual o Nessus deve parar de processar os resultados da verificação, foi definido em cinco horas e o valor não pode ser alterado.

Condições

É possível definir a lógica **if/then/else** na política do Unix. Isto permite ao usuário final usar um único arquivo que seja capaz de tratar diversas configurações. Por exemplo: o mesmo arquivo de política pode verificar as configurações de Postfix and Sendmail com o uso da sintaxe **if/then/else** adequada.

A sintaxe para executar as condições é a seguinte:

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

Exemplo:

```
 <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"
  regex: "^root"
  expect: "^root"
  </custom_item>
  </else>
 </if>
```

Independentemente de a condição ser reprovada ou aprovada, não aparecerá no relatório porque é uma verificação “silent” (silenciosa).

As condições podem ser do tipo “`and`” ou “`or`”.

Referência de arquivo de conformidade de auditoria para configuração do IBM iSeries

Esta seção descreve o formato e as funções das verificações de conformidade do IBM iSeries e o fundamento lógico por trás de cada configuração.

Uso de aspas:

As aspas simples e aspas duplas são intercambiáveis quando estiverem delimitando campos de auditoria, exceto nos dois casos abaixo:

1. Nas verificações de conformidade do Windows em que campos especiais como CRLF etc. devam ser interpretados literalmente, use aspas simples. Todos os campos incorporados interpretados como strings devem ser protegidos.

Por exemplo:

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

2. É necessário o uso de aspas duplas ao usar “include_paths” e exclude_paths" do WindowsFiles.

Se os strings forem usados em qualquer tipo de campo (descrição, value_data, regex, etc.) que contém

aspas simples ou aspas duplas, proceda da seguinte maneira:

a. Use o tipo oposto de aspas para as aspas delimitadoras mais afastadas.

Por exemplo:

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

b. Isole as aspas incorporadas com uma barra invertida (somente aspas duplas).

Por exemplo:

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

Privilégios do usuário

Para executar uma verificação de conformidade em um sistema iSeries, os usuários autenticados devem ter privilégios conforme definido abaixo:

1. Usuários com autoridade (*ALLOBJ) ou audit (*AUDIT) podem auditar todos os valores do sistema. Esse usuário normalmente pertence à classe (*SECOFR).

2. Os usuários de classe (*USER) ou (*SYSOPR) podem auditar a maioria dos valores, com exceção de QAUDCTL, QAUDENDACN, QAUDFRCLVL, QAUDLVL, QAUDLVL2 e QCRTOBJAUD.

Se um usuário não tiver privilégios para acessar um valor, o valor retornado será *NOTAVL.

Tipo de verificação

Todas as verificações de conformidade do conteúdo do IBM iSeries devem ser delimitadas por colchetes com o encapsulamento **`check_type`** e a designação “AS/400”. Isso é necessário para diferenciar arquivos **`.audit`** destinados a sistemas que executem o sistema operacional IBM iSeries a partir de outros tipos de auditorias de conformidade.

Exemplo:

```
<check_type:"AS/400">
```

Ao contrário de outros tipos de auditoria de conformidade, nenhuma palavra-chave adicional de tipo ou versão está disponível.

Palavras-chave

A tabela a seguir indica como poderá ser usada cada palavra-chave nas verificações de conformidade do IBM iSeries:

Palavra-chave	Exemplo de uso e configurações aceitas
`type`	`AUDIT_SYSTEMVAL` `SHOW_SYSTEMVAL`
`systemvalue`	Esta palavra-chave é usada para especificar um valor específico a ser verificado no sistema IBM iSeries. Exemplo: `systemvalue: "QALWUSRDMN"`
`description`	Esta palavra-chave permite adicionar uma breve descrição da verificação que estiver sendo realizada. Recomenda-se que o campo **`description`** seja exclusivo e que

	nenhuma verificação distinta tenha o mesmo campo de descrição. O SecurityCenter da Tenable usa este campo para a geração automática de um número exclusivo de identificação do plugin com base no campo **description**. Exemplo: `description: "Allow User Domain Objects (QALWUSRDMN) - '*all'"`

value_type	Esta palavra-chave é usada para definir o tipo de valor (“POLICY_DWORD” ou “POLICY_TEXT”) verificado no sistema IBM iSeries.
	
	Exemplo:
	`value_type: "POLICY_DWORD"`
	
	Exemplo:
	`value_type: "POLICY_TEXT"`

value_data	Esta palavra-chave define o valor de dados esperado para um valor do sistema. Exemplo: `value_type: "^([6-9]|[1-9][0-9]+)$"`

check_type	Esta palavra-chave define o tipo de verificação usada para um valor de dados.
	
	Exemplos:
	`check_type: "CHECK_EQUAL"`
	`check_type: "CHECK_NOT_EQUAL"`
	`check_type: "CHECK_GREATER_THAN"`
	`check_type: "CHECK_GREATER_THAN_OR_EQUAL"`
	`check_type: "CHECK_LESS_THAN"`
	`check_type: "CHECK_LESS_THAN_OR_EQUAL"`
	`check_type: "CHECK_REGEX"`
	
	Exemplo:
	`<custom_item>`
	` type: AUDIT_SYSTEMVAL`
	` systemvalue: "QUSEADPAUT"`
	` description: "Use Adopted Authority (QUSEADPAUT) - '!= *none'"`
	` value_type: POLICY_TEXT`
	` value_data: "*none"`
	` check_type: CHECK_NOT_EQUAL`
	`</custom_item>`

info	Esta palavra-chave é usada para adicionar uma descrição detalhada à verificação que está sendo executada, como regulamentação, URL, política corporativa ou outro motivo pelo qual a definição é necessária. Podem ser adicionados vários campos **info** em linhas separadas para a formatação do texto como um parágrafo. Não há limite predefinido para o número de campos **info** que podem ser usados. Exemplo: `info: "\nref : http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/books/sc415302.pdf pg. 21"`

Itens personalizados

Um item personalizado é uma verificação completa definida com base nas palavras-chave definidas anteriormente. A lista a seguir relaciona os tipos de itens personalizados disponíveis. Cada verificação começa com uma tag “**<custom_item>**” e termina com “**</custom_item>**”. As tags contêm listas de uma ou mais palavras-chave que são interpretadas pelo analisador de verificação de conformidade para executar as verificações.

As verificações de auditoria personalizadas podem usar “**</custom_item>**” e “**</item>**” de forma intercambiável para a tag de encerramento.

AUDIT_SYSTEMVAL

“AUDIT_SYSTEMVALUE” audita o valor da definição de configuração identificada pela palavra-chave “**systemvalue**”. O tipo de comparação com o valor auditado é especificado pela palavra-chave “**check_type**”.

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

SHOW_SYSTEMVAL

“SHOW_SYSTEMVAL” informa apenas o valor da definição de configuração identificada pela palavra-chave “**systemvalue**”.

```
<custom_item>
 type: SHOW_SYSTEMVAL
 systemvalue: "QAUDCTL"
 description: "show QAUDCTL value"
 severity: MEDIUM
</custom_item>
```

Condições

É possível definir a lógica **if/then/else** na política do IBM iSeries. Isso possibilita ao usuário final retornar uma mensagem de advertência em vez de aprovado/reprovado caso uma auditoria resulte em aprovação.

A sintaxe para executar as condições é a seguinte:

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

Exemplo:

```
<if>
 <condition type : "or">
  <custom_item>
   type: AUDIT_SYSTEMVAL
   systemvalue: "QDSPSGNINF"
   description: "Sign-on information is displayed (QDSPSGNINF)"
   info: "\nref :
       http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/books/sc415302.pdf
       pg. 23"
   value_type: POLICY_DWORD
   value_data: "1"
  </custom_item>
 </condition>

 <then>
  <custom_item>
   type: AUDIT_SYSTEMVAL
   systemvalue: "QDSPSGNINF"
   description: "Sign-on information is not displayed (QDSPSGNINF)"
   info: "\nref :
       http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/books/sc415302.pdf
       pg. 23"
   value_type: POLICY_DWORD
   value_data: "1"
  </custom_item>
 </then>

 <else>
  <report type: "WARNING">
   description: "Sign-on information is displayed (QDSPSGNINF)"
   info: ""\nref :
       http://publib.boulder.ibm.com/infocenter/iseries/v5r4/topic/books/sc415302.pdf
       pg. 23"
   info: "Check system policy to confirm requirements."
  </report>
 </else>
</if>
```

Independentemente de a condição ser reprovada ou aprovada, ela não aparecerá no relatório porque é uma verificação “silent” (silenciosa).

As condições podem ser do tipo “**and**” ou “**or**”.

Para obter mais informações

A Tenable produziu uma variedade documentos que detalham a instalação, implementação e configuração, operação do usuário e testes gerais do Nessus:

- **Guia de Instalação do Nessus** – instruções passo a passo da instalação.
- **Guia do Usuário Nessus** – como configurar e operar a interface de usuário do Nessus.
- **Verificações de Credenciais do Nessus para Unix e Windows** – informações sobre como realizar varreduras autenticadas de rede com o scanner de vulnerabilidades Nessus,

- **Verificações de Conformidade do Nessus** – guia geral para compreender e executar verificações de conformidade com o Nessus e o SecurityCenter.
- **Formato de arquivo Nessus v2** – descreve a estrutura do formato de arquivo `.nessus`, que foi introduzido com o Nessus 3.2 e NessusClient 3.2.
- **Especificação do protocolo Nessus XML-RPC** – descreve o protocolo e a interface XML-RPC do Nessus.
- **Monitoramento de Conformidade em Tempo Real** – descreve como as soluções da Tenable podem ser usadas para ajuda a cumprir muitos tipos diferentes de normas do governo e do setor financeiro.
- **Guia de administração SecurityCenter**

Outros recursos on-line são listados a seguir:

- Fórum de discussão do Nessus: https://discussions.nessus.org/
- Blog da Tenable: http://blog.tenable.com/
- Podcast da Tenable: http://blog.tenablesecurity.com/podcast/
- Vídeo de exemplos de uso: http://www.youtube.com/user/tenablesecurity
- Feed do Twitter da Tenable: http://twitter.com/tenablesecurity

Fique à vontade para entrar em contato com a Tenable pelo support@tenable.com, sales@tenable.com ou visite nosso site no endereço http://www.tenable.com/.

Apêndice A: Exemplo de arquivo de conformidade Unix

Nota: O arquivo **tenable_unix_compliance_template.audit** a seguir está disponível no Tenable Support Portal no endereço https://support.tenable.com/. Este arquivo relaciona os diferentes tipos de verificações de conformidade Unix que podem ser executados com o uso do módulo de conformidade Unix da Tenable. O arquivo contém algumas atualizações não descritas neste documento.

```
#
# (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"
      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
# 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'"
      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"
      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>

<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>
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>
```

Apêndice B: Exemplo de arquivo de conformidade Windows

Nota: O arquivo a seguir está disponível no Tenable Support Portal (Portal de suporte de Tenable) no endereço https://support.tenable.com/. O arquivo contém algumas atualizações não descritas neste documento. Este nome de script específico é denominado **financial_microsoft_windows_user_audit_guideline_v2.audit** e baseia-se em guias de otimização comuns para a administração de usuários. Esta política procura uma política de senhas e de bloqueio de contas apropriada e garante que os eventos de login estejam conectados ao log de eventos do 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]
      </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>
```

Sobre a Tenable Network Security

Tenable Network Security, líder em monitoramento unificado de segurança, é a criadora do scanner de vulnerabilidades Nessus e de soluções de primeira classe sem agente para o monitoramento contínuo de vulnerabilidades, pontos fracos de configuração, vazamento de dados, gerenciamento de logs e detecção de comprometimentos para ajudar a garantir a segurança da rede e o cumprimento das leis e normas FDCC, FISMA, SANS, CSIS e PCI. Os produtos premiados da Tenable são utilizados por muitas organizações da Global 2.000 e por órgãos públicos para tomar a iniciativa de reduzir os riscos nas redes. Para obter mais informações, visite http://www.tenable.com.

GLOBAL HEADQUARTERS

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