Verificações de conformidade Nessus

Configurações e conteúdo do sistema de auditoria

30 de agosto de 2012

(Revisão 61)

Índice

Introdução

Este documento descreve como o Nessus 5.x pode ser usado 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 conteúdo confidencial em vários sistemas.

As frases “Policy Compliance” (“Conformidade com a política”) e “Compliance Checks” (“Verificações da conformidade”) são usadas de forma intercambiável neste documento.

A auditoria do sistema Scada é possível com o Nessus. No entanto, este documento não abrande esta funcionalidade. Consulte a página de informações Nessus.org Scada aqui para obter mais informações.

A execução de uma auditoria de conformidade não é o mesmo que a execução de uma varredura da vulnerabilidade, embora existam alguns pontos em comum. Uma auditoria de conformidade determina se um sistema está configurado segundo uma política estabelecida. Uma varredura da vulnerabilidade determina se o sistema está aberto a vulnerabilidades conhecidas. Os leitores saberão quais tipos de parâmetros de configuração e dados confidenciais poderão ser auditados, como configurar o Nessus para a execução destas auditorias e como o SecurityCenter da Tenable poderá ser usado para o gerenciamento e a automatização deste processo.

Pré-requisitos

Este documento requer algum nível de conhecimento sobre o scanner da vulnerabilidade do Nessus. 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/.

Clientes do Nessus ProfessionalFeed e SecurityCenter

Os usuários devem assinar o Nessus ProfessionalFeed ou usar o SecurityCenter para a realização das verificações de conformidade descritas neste documento. Ambos estão disponíveis na Tenable Network Security (http://www.tenable.com/). Uma lista mais detalhada dos requisitos técnicos para a execução das verificações de auditoria é discutida nos próximos capítulos.

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.

Normas de conformidade

Há muitos tipos diferentes de requisitos governamentais e financeiros. É importante compreender que estes requisitos de conformidade são parâmetros mínimos que podem ser interpretados de maneira diferente, dependendo das metas de negócios da organização. Os requisitos de conformidade devem ser associados à metas de negócios para garantir que os riscos sejam corretamente identificados e abrandados Para obter mais informações sobre o desenvolvimento deste processo, consulte o documento da Tenable “Maximizing ROI on Vulnerability Management” em http://www.tenable.com/whitepapers/.

Uma empresa, por exemplo, pode ter uma política que exija que todos os servidores com informações pessoalmente identificáveis (PII) do cliente neles tenham registro habilitado e senha com, no mínimo, 10 caracteres. Esta política pode ajudar nos esforços de uma organização para manter conformidade com qualquer quantidade de diferentes regulamentos.

As normas e orientações frequentes de conformidade incluem, entre outras:

- BASEL II
- Center for Internet Security Benchmarks (CIS)
- Control Objectives for Information and related Technology (COBIT)
- Defense Information Systems Agency (DISA) STIGs
- Federal Information Security Management Act (FISMA)
- Federal Desktop Core Configuration (FDCC)
- Gramm-Leach-Bliley Act (GLBA)
- Health Insurance Portability and Accountability Act (HIPAA)
- Normas de segurança ISO 27002/17799
- Information Technology Information Library (ITIL)
- Diretrizes de configuração do National Institute of Standards (NIST)
- Diretrizes de configuração da National Security Agency (NSA)
- Payment Card Industry Data Security Standards (PCI DSS)
- Sarbanes-Oxley (SOX)
- Site Data Protection (SDP)
- United States Government Configuration Baseline (USGCB)
- Várias leis estaduais (por exemplo:Security Breach Notification Act - SB 1386 da Califórnia)

As verificações de conformidade lidam também com monitoramento em tempo real, como a execução de detecção de invasão de controle de acesso. Para obter mais detalhes sobre como as soluções de configuração, vulnerabilidade, gerenciamento, vazamento de dados, análise de logs e monitoramento de rede da Tenable podem ajudá-lo a realizar os regulamentos de conformidade, envie um e-mail ao endereço sales@tenable.com para solicitar uma cópia do documento “Real-Time Compliance Monitoring” (Monitoramento de Conformidade em Tempo Real).

Configurações de auditorias, vazamento de dados e conformidade

O que é uma auditoria?

O Nessus pode ser usado para efetuar login em servidores Unix e Windows, dispositivos Cisco, sistemas, servidores IBM iSeries, e bancos de dados SCADA para determinar se eles foram configurados segundo a política de segurança local da instalação. O Nessus pode também pesquisar o disco rígido inteiro dos sistemas Windows e Unix quanto a conteúdo não autorizado.

É importante que as organizações estabeleçam uma política de segurança da instalação antes da execução de uma auditoria para garantir que o patrimônio esteja protegido de maneira apropriada. Uma avaliação da vulnerabilidade determinará se os sistemas estão vulneráveis a exploits conhecidos, mas não determinará, por exemplo, se registros pessoais estão sendo armazenados em um servidor público.

Não há norma absoluta sobre segurança, pois se trata de uma questão de gerenciar riscos e isto varia entre as organizações.

Por exemplo: leve em consideração os requisitos de senha, como validades mínima/máxima de senhas e políticas de bloqueio de contas. Pode haver motivos reais para a alteração frequente ou não de senhas. Pode haver também motivos para o bloqueio de uma conta em caso de mais de cinco falhas de login, mas se este for um sistema de missão crítica, a definição de valores mais altos ou mesmo a desativação total de bloqueios seria uma decisão apropriada.

Estas opções de configuração têm muito a ver com o gerenciamento de sistemas e política de segurança, mas não especificamente vulnerabilidades do sistema ou falta de patches. O Nessus pode efetuar verificações de conformidade para servidores Unix e Windows. As políticas podem ser bastante simples ou complexas, dependendo dos requisitos de cada varredura de conformidade individual.

Auditoria x varredura da vulnerabilidade

O Nessus pode executar varreduras da vulnerabilidade de serviços da rede além de efetuar login em servidores para detectar a ausência de patches No entanto, a falta de vulnerabilidades não significa que os servidores estejam corretamente configurados ou “em conformidade” com uma norma específica.

A vantagem de se usar o Nessus para a execução de varreduras de vulnerabilidade e de auditorias de conformidade é que todos esses dados podem ser obtidos de uma só vez. Conhecer o modo como um servidor está configurado, de que forma recebeu patches e quais vulnerabilidades estão presentes pode ajudar a determinar as medidas para abrandar os riscos.

Em um nível mais alto, se estas informações forem agregadas a uma rede inteira ou a uma classe de ativo (como o SecurityCenter da Tenable), a segurança e os riscos poderão ser analisados de maneira global. Isto permite que auditores e gerentes de rede identifiquem tendências nos sistemas em não conformidade e ajustem controles para corrigi-los em uma escala maior.

Exemplo de itens de auditoria

As seções abaixo abordam as auditorias de configuração em sistemas Windows, Unix, de bancos de dados, IBM iSeries e Cisco.

O mecanismo de busca Nessus 5 regex baseia-se em um dialeto Perl com “Extended POSIX” devido a sua flexibilidade e velocidade.

Windows

O Nessus pode verificar qualquer definição configurada como uma "política" sob a estrutura do Microsoft Windows. Existem centenas de definições de registro que podem ser auditadas, cujas permissões de arquivos, diretórios e objetos podem ser analisadas. Uma lista parcial de exemplos de auditoria inclui o teste das definições a seguir:

- Duração do bloqueio de contas
- Retenção do log de segurança
- Permissão de logon local
- Aplicação do histórico de senhas

O exemplo de um item de "auditoria" para servidores Windows é exemplificado abaixo:

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

Esta auditoria específica procura a definição “comprimento mínimo de senha” em um servidor Windows e irá gerar um alerta se o valor for inferior a sete caracteres.

Além disso, o Nessus pode pesquisar dados confidenciais em computadores com Windows. O exemplo a seguir exibe números de cartão de crédito Visa em uma variedade de formatos de arquivo:

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

Esta verificação verifica os padrões de arquivos Excel, Adobe e de texto que indiquem se há um ou mais números válidos de cartão de crédito Visa presentes.

Unix
O Nessus pode ser usado para verificar permissões de arquivos, o conteúdo de um arquivo, a execução de processos e o controle de acesso de usuário em uma variedade de sistemas baseados em Unix. Atualmente, existem verificações disponíveis para a auditoria de derivativos Unix Solaris, Red Hat, AIX, HP-UX, SuSE, Gentoo e FreeBSD.

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

Esta auditoria verifica se o comprimento mínimo de senha em um sistema Unix é de 14 caracteres.

Cisco
O Nessus pode verificar a configuração de execução de sistemas com sistemas operacionais Cisco IOS e confirmar se está em conformidade com as normas de política de segurança. As verificações podem ser executadas por meio de um login sem privilégios ou que utilize a senha “enable” com privilégios.

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

IBM iSeries
Usando as credenciais fornecidas, a Nessus pode testar a configuração de sistemas com o IBM iSeries e confirmar se está de acordo com os padrões da política de segurança.

```
<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>
```

Bancos de dados
O Nessus pode ser configurado para se conectar aos seguintes tipos de banco de dados e para determinar a conformidade com a política de segurança local:

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

Em geral, a Tenable recomenda executar uma verificação de conformidade de banco de dados com um usuário com privilégios SYSDBA para Oracle, "**sa**" ou uma conta com função de servidor sysadmin para MS-SQL e conta de usuário da instância do DB2 para garantir a integridade do relatório, pois algumas tabelas e parâmetros do sistema ou ocultas só podem ser acessados por uma conta com privilégios. Observe que, para o Oracle, na maioria dos casos, um usuário com a função DBA atribuída executará a maioria das verificações nas auditorias da Tenable, mas algumas verificações resultarão em erros devido a privilégios de acesso insuficientes. Esse mesmo argumento também pode ser aplicável a outros bancos de dados. Uma conta de menor privilégio pode ser usada para auditoria de banco de dados, mas a desvantagem é que o relatório completo não pode ser garantido.

As auditorias de banco de dados normalmente englobam instruções de seleção que recuperam detalhes relativos à segurança do banco de dados do usuário, como a existência ou status de procedimentos armazenados sem segurança. O exemplo a seguir determina se o procedimento potencialmente perigoso "`xp_cmdshell`" armazenado está ativado:

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

A capacidade de gravação de arquivos de auditoria de cada organização e a pesquisa de dados confidenciais é bastante útil. Este documento descreve como criar políticas personalizadas para o exame de vários tipos de dados.

Relatórios de auditoria

Quando uma auditoria é realizada, o Nessus tenta determinar se o host está em conformidade ou não ou se os resultados são inconclusivos.

Os resultados de conformidade no Nessus são registrados com o nível de severidade “Note” (Observação), os resultados de não conformidade são registrados com “Hole” (Falha) e os resultados inconclusivos (por exemplo: verificação de permissões para um arquivo não encontrado no sistema) são relatados como “Warning” (Advertência). O SecurityCenter da Tenable usa uma classificação de severidade baixa (“low”), média (“medium”) e alta (“high”), as verificações conformes são classificadas como baixas, as não conformes como altas e as inconclusivas como médias.

Ao contrário de uma verificação de vulnerabilidade que indica apenas se a vulnerabilidade está presente, uma verificação de conformidade sempre gera uma informação. Desta maneira, os dados podem ser usados como base de um relatório de auditoria para informar que um host foi aprovado ou não em um teste específico ou se não foi testado corretamente.

Tecnologia exigida

Plugins Nessus .nbin para conformidade de configuração do Unix e do Windows

A Tenable projetou dois plugins Nessus (IDs 21156 e 21157), que implementam as APIs usadas para a realização de auditorias dos sistemas Unix e Windows. Os plugins foram pré-compilados com o formato “**.nbin**” do Nessus.

Os plugins e as respectivas políticas de auditoria estão disponíveis para clientes do ProfessionalFeed e usuários do SecurityCenter. Este documento discute também duas ferramentas do Windows para ajudar na criação de arquivos **.audit** personalizados do Windows e uma ferramenta do Unix para criar arquivos **.audit** do Unix.

Para auditorias de conformidade com Unix, apenas a autenticação SSH é compatível. Os protocolos legados, como Telnet, não são permitidos por razões de segurança.

Plugin Nessus .nbin para conformidade de conteúdo do Windows

A Tenable projetou um plugin Nessus (ID 24760) denominado “Windows File Contents Check”, que implementa as APIs usadas para auditar sistemas Windows para conteúdo não conforme, como PII (Informações Pessoais Identificáveis) ou PHI (Informações de Integridade Protegidas). Os plugins foram pré-compilados com o formato “**.nbin**” do Nessus. Estes plugins e as respectivas políticas de auditoria estão disponíveis para clientes do ProfessionalFeed e usuários do SecurityCenter.

Observe que os sistemas Unix não são verificados pelo plugin 24760.

Plugin Nessus .nbin para conformidade do banco de dados

A Tenable projetou um plugin Nessus (ID 33814) denominado “Database Compliance Checks”, que implementa as APIs usadas para auditar vários sistemas de banco de dados. Este plugin é pré-compilado com o formato “**.nbin**” do Nessus. O plugin e as respectivas políticas de auditoria estão disponíveis para usuários do ProfessionalFeed e usuários do SecurityCenter.

As verificações de conformidade do banco de dados não estão disponíveis para uso com a versão 3.4.4 e anteriores do Security Center.

Plugin Nessus .nbin para conformidade do IBM iSeries

A Tenable projetou um plugin Nessus (ID 57860) denominado “IBM iSeries Compliance Checks”, que implementa as APIs usadas para verificar sistemas com o IBM iSeries. Esse plugin é pré-compilado com o formato “**.nbin**” do Nessus. O plugin e as respectivas políticas de auditoria estão disponíveis para clientes do ProfessionalFeed.

Plugin Nessus .nbin para conformidade do sistema Cisco

A Tenable projetou um plugin Nessus (ID 46689) denominado “Cisco IOS Compliance Checks”, que implementa as APIs usadas para auditar sistemas que executam o sistema operacional Cisco IOS. Este plugin é pré-compilado com o formato “**.nbin**” do Nessus. O plugin e as respectivas políticas de auditoria estão disponíveis para clientes do ProfessionalFeed. Esta verificação de conformidade pode ser executada no modo configuração Saved, Running ou Startup.

Políticas de auditoria

A Tenable desenvolveu diversas políticas de auditoria para plataformas Unix, Windows e Cisco. As políticas estão disponíveis como arquivos de texto **.audit** para assinantes do ProfessionalFeed e podem ser descarregadas do Tenable Support Portal no endereço https://support.tenable.com/. Para obter as notícias mais recentes a respeito da funcionalidade de auditoria da Tenable e as versões mais recentes do arquivo **.audit**, consulte os fóruns de discussão: https://discussions.nessus.org/.

Vários aspectos de auditorias de conformidade comuns foram considerados, como os requisitos de SOX, Fisma e PCI DSS durante a elaboração destas políticas de auditoria, embora não estejam representadas como arquivos de auditoria oficiais para estes critérios. Os usuários devem examinar estas políticas **.audit** e personalizar as verificações para seu ambiente local. Os usuários podem renomear os arquivos **.audit** para adaptá-los às descrições locais. Outras políticas **.audit** provêm diretamente de definições de configuração recomendadas pela Cert, CIS, NSA and Nist.

A Tenable espera criar vários tipos diferentes de arquivos **.audit** com base no feedback do cliente e nas "práticas recomendadas" emergentes. Clientes de várias organizações de consultoria e da Tenable também passaram a implementar suas próprias políticas **.audit** e pretendem compartilhá-las com outros usuários do ProfessionalFeed Nessus. Uma maneira conveniente de compartilhar políticas **.audit** ou apenas interagir com a comunidade Nessus é por meio dos fóruns de discussão do Tenable Network Security em https://discussions.nessus.org/.

Utilitários

A Tenable desenvolveu uma ferramenta para converter arquivos **.inf** em arquivos de auditoria **.audit** do Windows. Esta ferramenta é denominada **i2a** e será abordada em outras seções deste documento.

Existem duas ferramentas Unix que podem ser usadas para a criação de arquivos **.audit** Unix. A primeira ferramenta, denominada **c2a** ("auditoria de configuração"), pode ser usada para a criação de arquivos **.audit** diretamente de arquivos de configuração existentes. Por exemplo: caso o arquivo de configuração Sendmail esteja configurado corretamente segundo a política de instalação, a ferramenta **c2a** poderá criar uma política de auditoria com base na soma de verificação MD5 do arquivo ou com base nos pares valor específico e argumento do arquivo **sendmail.cf**. A segunda ferramenta, denominada **p2a** ("pacote de auditoria"), pode ser usada para a criação de arquivos **.audit** Unix a partir do pacote básico em um sistema Unix (Linux compatível com RPM ou Solaris 10) ou a partir de um arquivo de texto simples com uma lista de nomes de pacote.

Scanners Nessus Unix ou Windows

Uma variedade de plataformas pode ser usada para a execução de verificações de conformidade e, independentemente do sistema operacional no qual o Nessus reside. É possível executar auditorias de conformidade de um servidor Windows 2003 em um laptop OS X, além de ser possível auditar um servidor Solaris em um laptop Windows.

Credenciais de dispositivos a serem auditados

Em todos os casos, são exigidas credenciais de Unix SSH, domínio Windows, IBM iSeries, Cisco IOS ou de banco de dados para que o Nessus efetue login nos servidores de destino. Na maioria dos casos, o usuário precisa ser um "superusuário" ou um usuário comum sem capacidade de alteração de privilégio (por exemplo: **sudo**, **su** ou **su+sudo**). Se o usuário que realizar a auditoria não possuir privilégios de "superusuário", muitos dos comandos remotos do sistema não serão executados ou gerarão resultados incorretos.

A conta Windows usada para o início da sessão de credenciais deve ter permissão para leitura da política do computador local. Se um host de destino não for parte de um domínio Windows, a conta deverá ser membro do grupo de administradores do host. Se o host for parte de um domínio, o grupo de administradores do host e a conta terão acesso à política do computador local se for membro do grupo de administradores do domínio.

Para executar verificações de conformidade do conteúdo Windows, além de efetuar login no sistema com privilégios de domínio, o acesso à ferramenta Windows Management Instrumentation (WMI) também deverá ser permitido. Se este acesso não estiver disponível, o Nessus informará que o acesso ao WMI não estava disponível para varredura.

As verificações de conformidade do banco de dados requerem apenas as credenciais deste para a execução de uma auditoria completa de conformidade do banco de dados. Isto ocorre porque o banco de dados, e não o host do sistema operacional, está sendo examinado quanto à conformidade.

As verificações de conformidade do Cisco IOS normalmente requerem a senha "enable" para a execução de uma auditoria completa de conformidade da configuração do sistema. Isto ocorre porque o Nessus está auditando a saída do comando "**`show config`**", disponível apenas a usuários com privilégios. Se o usuário do Nessus usado para a auditoria já tiver privilégios "enable", não será necessária a senha "enable".

Para obter mais informações sobre como configurar o Nessus ou o SecurityCenter para a execução de verificações de vulnerabilidade credenciadas locais, consulte o documento "Nessus Credentials Checks for Unix and Windows" disponível em http://www.tenable.com/products/nessus/documentation.

Uso de "su", "sudo" e "su+sudo" para auditorias

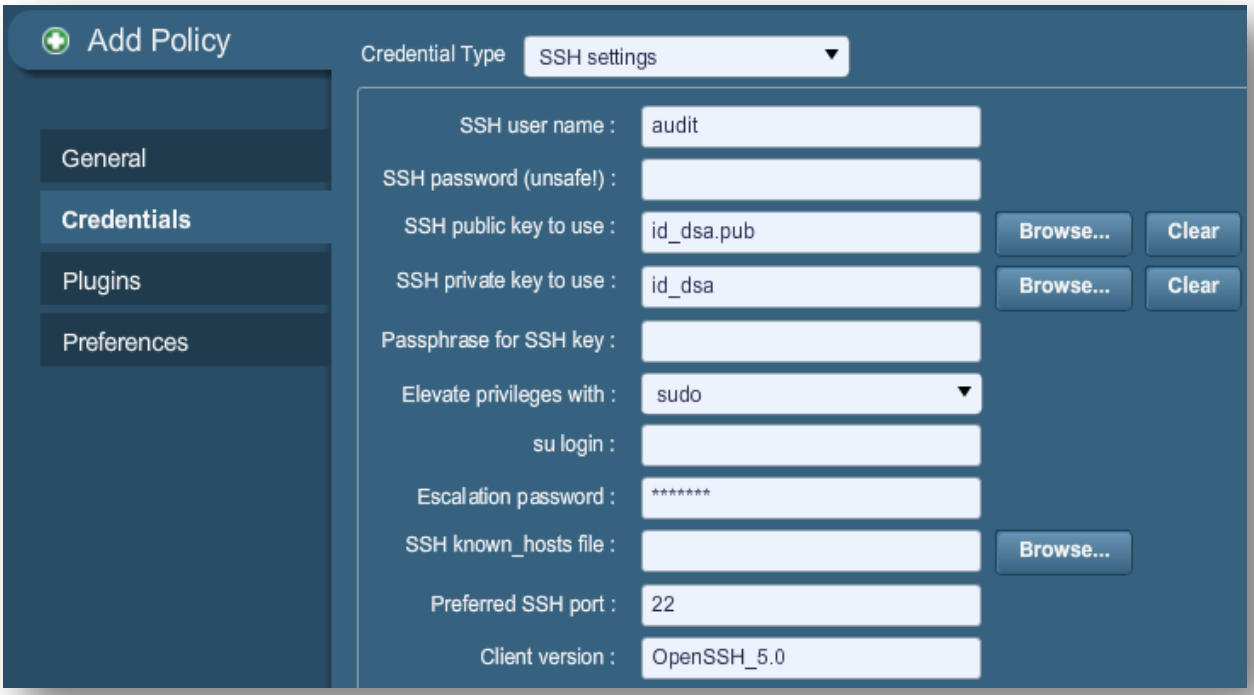
Use "`su+sudo`" nos casos em que a política da empresa impede o Nessus de efetuar login em um host remoto com o usuário raiz ou um usuário com privilégios "`sudo`". Em login remoto, o usuário do Nessus sem privilégios pode "`su`" (alternar usuário) para um usuário com privilégios `sudo`.

As varreduras credenciadas mais eficazes para Unix são aquelas em que as credenciais fornecidas têm privilégios "root". Uma vez que muitas instalações não permitem login remoto como raiz, os usuários do Nessus podem acessar "**`su`**", "**`sudo`**" ou "**`su+sudo`**" com uma senha distinta para uma conta configurada para ter os privilégios apropriados.

Além disso, se um arquivo SSH `known_hosts` estiver disponível e for fornecido com base na política de varredura, o Nessus tentará efetuar login apenas nos hosts existentes desse arquivo. Isto garante que o mesmo nome de usuário e senha usados para auditar os servidores SSH conhecidos não sejam usados para efetuar login em um sistema que não está sob seu controle.

Exemplo de sudo

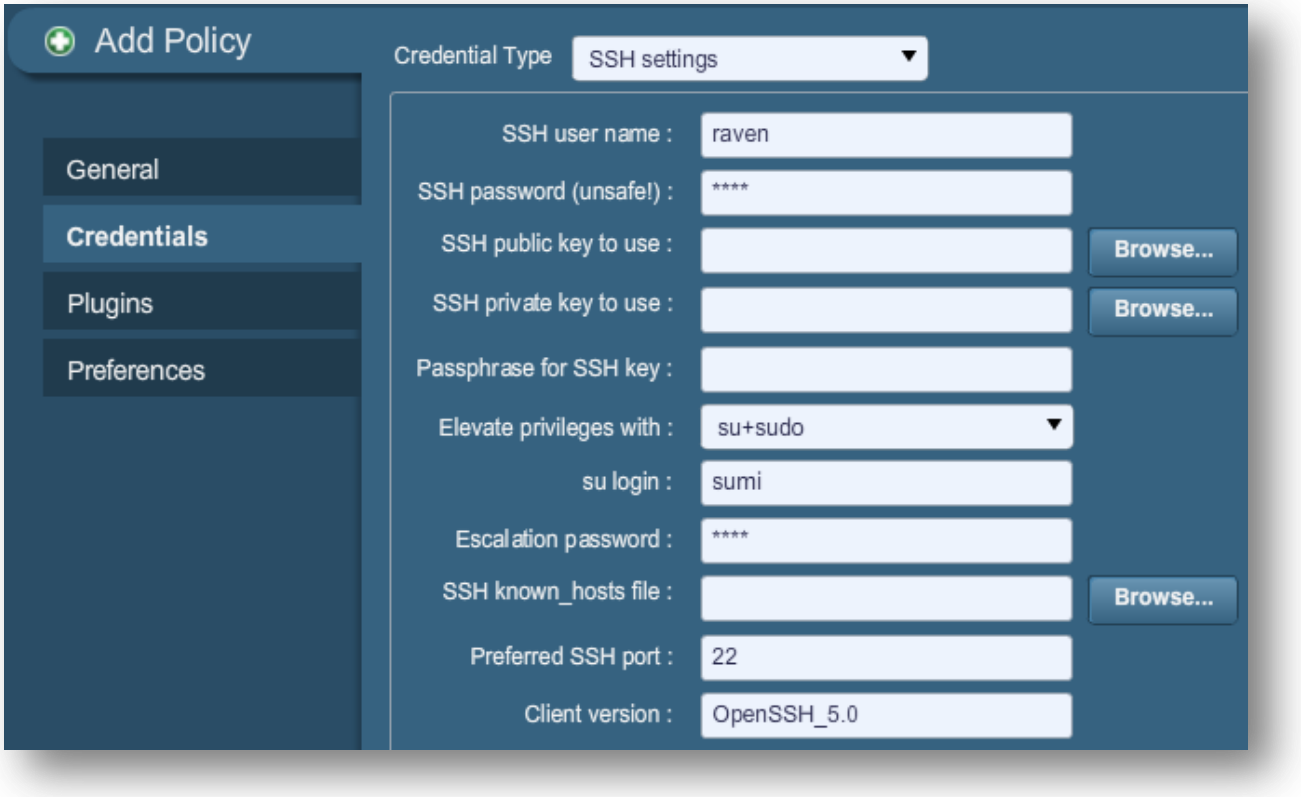
Veja a seguir um exemplo de imagem de tela "`sudo`" juntamente com chaves SSH. Neste exemplo, a conta de usuário é "audit", que foi adicionada ao arquivo `/etc/sudoers` no sistema a ser verificado. A senha fornecida é a senha para a conta "audit" e não a senha raiz. As chaves SSH correspondem às chaves geradas para a conta "audit":



Exemplo de su+sudo
Com o lançamento do Nessus 4.2.2, foi incluído um novo método de elevação de credencial para hosts com Unix que tenham **sudo** instalado "**su+sudo**." Este método permite fornecer credenciais para uma conta que não possui permissões **sudo su** a um usuário que as possui e emite o comando **sudo**.

Esta configuração oferece maior segurança para as credenciais durante a varredura e satisfaz os requisitos de conformidade de muitas organizações.

Para ativar este recurso, selecione "su+sudo" na seção "Elevate privileges with" (Elevar privilégios com) em configurações de credenciais/SSH, conforme mostrado na imagem de tela a seguir:



Nos campos "SSH user name" (nome de usuário) e "SSH password" (senha SSH), insira as credenciais que não possuem privilégios **sudo**. No exemplo acima, a conta do usuário é "raven." No menu suspenso "Elevate privileges with", selecione "su+sudo". Nos campos "su login" e "Escalation password" (Senha de elevação), insira o nome de usuário e a senha que *possuem* credenciais privilegiadas (neste exemplo, "sumi"). Nenhuma outra modificação de política é necessária.

Observação importante a respeito de sudo
Ao auditar sistemas Unix por meio de **su**, **sudo** ou **su+sudo**, considere os seguintes itens:

- Caso o sistema Unix tenha sido otimizado para limitar os comandos que podem ser executados por **sudo** ou os arquivos acessados por usuários remotos, isto poderá afetar a auditoria. Compare as auditorias não feitas na raiz a uma auditoria feita na raiz se suspeitar que está sendo limitada por medidas de segurança.
- comando **sudo** não se origina do Solaris e deve ser descarregado e instalado caso o sistema de destino seja o Solaris. Certifique-se de que o binário **sudo** esteja acessível como "**/usr/bin/sudo**".

- Ao verificar com **known_hosts**, a varredura do Nessus deve especificar também um host a ser verificado. Por exemplo: se o usuário verificou uma classe C, mas carregou um arquivo **known_hosts** que continha apenas 20 hosts individuais na classe C, o Nessus verificaria apenas estes hosts no arquivo.

- Algumas configurações baseadas no Unix requerem que os comandos iniciados por sudo sejam executados a partir de sessões **tty**. As varreduras de vulnerabilidade do Nessus executadas com a opção “su+sudo” não correspondem a este requisito. Se a opção su+sudo for usada, será preciso criar uma exceção no sistema de destino. Para determinar se isto é necessário para a distribuição do Unix, insira o seguinte comando como raiz no sistema que está sendo examinado:

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

  Se a linha “**requiretty**” estiver presente no arquivo de configuração **sudoers**, será preciso criar uma exceção para o arquivo **/etc/sudoers**, conforme o exemplo a seguir:

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

  Observe que {userid} é o nome de usuário que será usado para a execução do comando “sudo” (a página “su login” na seção credenciais/SSH de sua política). Além disso, certifique-se de ter a seguinte linha em seu arquivo **sudoers**:

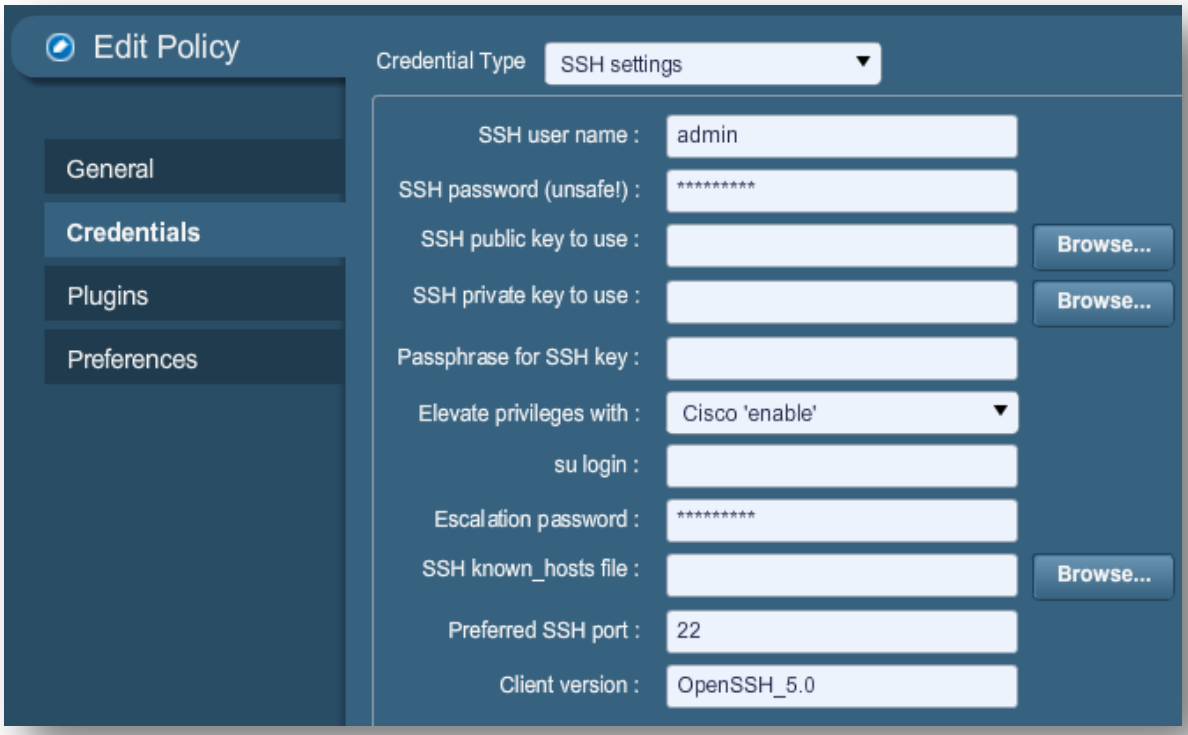
  ```
  {userid} ALL=(ALL) ALL
  ```

  Além disso, {userid} é o nome de usuário que será usado para a execução do comando “sudo” (o “su login” na seção credenciais/SSH de sua política).

Exemplo no Cisco IOS:

Somente a autenticação SSH é aceita. Os dispositivos IOS legados que requerem Telnet para autenticação não poderão ser examinados com verificações de conformidade Nessus Cisco.

As credenciais Cisco IOS são configuradas por meio da tela de credenciais “**SSH settings**” (Configurações de SSH) na interface de usuário do Nessus. Insira o nome de usuário e a senha do SSH exigidos para o login no roteador Cisco. Para especificar que os privilégios devem ser elevados com “Enable” (Ativar), selecione “**Cisco ‘enable’**” (‘Ativar‘ Cisco) ao lado da configuração “**Elevate privileges with**” (Elevar privilégios com) e insira a senha "enable" (ativar) ao lado de “**Escalation password**” (Senha de alteração).



Conversão de arquivos .inf do Windows em arquivos .audit com i2a

Se você ou sua organização de TI possuir arquivos de política do Windows (normalmente encontrados com a extensão "**.inf**"), poderão ser convertidos em arquivos **.audit** para uso com auditorias do Nessus em servidores Windows.

Obtenção e instalação da ferramenta

A ferramenta **i2a** está disponível como arquivo zip e pode ser obtida no Tenable Support Portal localizado em https://support.tenable.com/. Esta ferramenta não possui interface gráfica do usuário e é acionada na linha de comando.

Extraia o conteúdo do arquivo em um diretório de sua escolha e mova os arquivos Windows **.inf** para o mesmo diretório.

Converter .inf em .audit

Para executar a ferramenta de conversão no do prompt de comando, basta digitar:

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

Neste exemplo, **yourfile.inf** é o arquivo **.inf** de origem e **file.audit** é o arquivo de destino **.audit**.

Análise da conversão

A Tenable atingiu a taxa de aproximadamente 100% de conversão entre o conteúdo descrito em um arquivo **.inf f** e o conteúdo auditado em um arquivo **.audit**. No entanto, existem alguns itens de política que não podem ser testados com a atual tecnologia do Nessus 5.

Cria-se um log do processo de conversão para cada execução da ferramenta **i2a**. O log contém uma auditoria linha por linha de todo o processo de conversão. Se uma linha no arquivo **.inf** não puder ser convertida, será armazenada no arquivo de log.

Formato de configuração .inf correto

Para as verificações mostradas no arquivo de log e que não podem ser processadas, certifique-se de que estejam em conformidade com os formatos aceitáveis relacionados a seguir.

As configurações **System Access** (Acesso do sistema), **System Log** (Log do sistema), **Security Log** (Log de segurança), **Application Log** (Log do aplicativo) e **Event Audit** (Auditoria de eventos) compartilham o mesmo formato. Cada entrada é descrita por "**`Key`**" seguido por "**`value`**".

Sintaxe:

```
Key = value
```

No caso acima, **`Key`** é o item a ser auditado e **`value`** é o valor esperado dessa chave no sistema remoto.

Exemplo:

```
MinimumPasswordLength = 8
```

O formato das configurações de **Privilege Rights** (Direitos de Privilégio) é similar ao acima mencionado. No entanto, nesta configuração, o valor pode estar vazio.

Sintaxe:

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

Exemplo:

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

Ou:

```
SeTcbPrivilege =
```

A configuração de **Registry Key** (Chave de registro) consiste das quatro partes abaixo:

- Registry Key – a chave de registro que deve ser auditada.
- Inheritance Value – identifica se as permissões para a chave de registro são herdadas ou não. O valor pode ser [0-4].
- DACL – DACL é uma ACL controlada pelo proprietário de um objeto e que especifica o acesso que usuários ou grupos específicos podem ter ao objeto.
- SACL – SACL é uma ACL que controla a geração de mensagens de auditoria para tentativas de acesso a um objeto alcançável.

Sintaxe:

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

Os campos DACL e SACL podem estar vazios e, nesse caso, a verificação será ignorada.

Exemplo:

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

O formato da configuração de **File Security** (segurança de arquivos) é similar ao formato de chave de registro acima descrito.

Sintaxe:

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

Exemplo:

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

A configuração do **Service General** (valor geral do serviço) consiste nas quatro partes a seguir:

- Service Name (Nome do serviço) – o serviço que deve ser auditado.
- Service start type (Tipo de início do serviço) – manual, automático ou desativado. O valor pode ser [2-4].
- DACL – DACL é uma ACL controlada pelo proprietário de um objeto e que especifica o acesso que usuários ou grupos específicos podem ter ao objeto.
- SACL – SACL é uma ACL que controla a geração de mensagens de auditoria para tentativas de acesso a um objeto alcançável.

Sintaxe:

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

Exemplo:

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

Se as permissões da configuração de um serviço não forem verificadas (somente o tipo de inicialização), isto poderá ser feito conforme abaixo.

Sintaxe:

```
Service Name,Start type
```

Exemplo:

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

A configuração do **Registry Value** (valor de registro) consiste nas quatro partes abaixo:

- RegistryKey – A chave de registro que deve ser auditada.
- RegistryType – O tipo de registro: REG_DWORD, REG_SZ, etc.
- RegistryValue – O valor da chave de registro.

RegistryValue pode ser definido entre aspas duplas, aspas simples ou sem aspas.

Sintaxe:

```
RegistryKey,RegistryType,RegistryValue
```

Exemplo:

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

Caso queira comentar uma linha específica dentro do arquivo **.inf**, coloque um ponto e vírgula ";" na frente da linha pra que o script ignore esta linha.

Conversão de arquivos de configuração Unix em arquivos .audit com c2a

A ferramenta **c2a.pl** foi projetada para auxiliar os auditores na criação de arquivos **.audit** para a auditoria das configurações de aplicativos em uma determinada rede. Por exemplo: caso queira que todos os servidores Web em uma determinada rede sejam configurados exatamente como o host mestre X, executa esta ferramenta executado no host X, crie o arquivo **.audit** para **httpd** no sistema e, em seguida, insira o arquivo no daemon do Nessus e execute a varredura em todos os outros servidores Web para verificar a conformidade.

Como opção, esta ferramenta pode ser usada também para a criação de arquivos de auditoria MD5 de um host inteiro. Ela espera uma lista de arquivos/diretórios que precisam ser auditados em um arquivo de entrada que, em seguida, processa de maneira recorrente no caso de diretórios para criar um arquivo **.audit** para o sistema. Em seguida, este arquivo pode ser usado posteriormente para a varredura de modificações para arquivos e diretórios básicos.

Obtenção e instalação da ferramenta

A ferramenta **c2a** consiste em um arquivo **tar** compactado e pode ser obtida no Tenable Support Portal no endereço https://support.tenable.com/.

Extraia o conteúdo de **c2a-x.x.x.tar.gz** no computador local com o seguinte comando:

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

Isto criará um diretório "**c2a**" sob o diretório atual e extrairá os arquivos para ele. Caso queira extrair o conteúdo para um diretório de sua escolha, use o seguinte comando:

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

Após descompactar o arquivo, visualize os arquivos a seguir no diretório **~/c2a**:

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

Criar um arquivo de auditoria MD5

Execute a ferramenta de conversão com a opção "**-md5**" ao digitar:

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

A ferramenta requer um arquivo de entrada com uma lista de arquivos e diretórios que devem ser auditados para valores MD5, além de um nome de arquivo de saída para o arquivo de auditoria.

Ao adicionar arquivos ao arquivo de entrada, lembre-se de usar este formato:

/path/to/file

Use este formato quando adicionar diretórios:

/path/to/file/

Se este formato for usado e o arquivo for um arquivo real e não um diretório, a ferramenta `c2a` alertará que este arquivo não existe. A barra "/" é adequada para a adição de diretórios.

Se a entrada no arquivo de entrada for um arquivo MD5 normal, somente esse arquivo será calculado e gravado no formato **.audit**. No caso de diretório, o script investigará, de maneira recorrente, todo e qualquer arquivo do diretório. Se não um arquivo de saída for especificado, o resultado será gravado em **~/c2a/op.audit**.

Ao processar a lista de arquivos especificados pelo "inputfile", todos os links simbólicos encontrados serão ignorados. Será exibida uma mensagem de advertência indicando que o arquivo não existe ou é um link simbólico. A partir desta versão, o **c2a** não aceita links simbólicos.

Criar arquivo de auditoria com base em um ou mais arquivos de configuração

A ferramenta **c2a** é ideal para o processamento de arquivos de configuração que possuam conteúdo linha a linha exclusivos. Caso o arquivo de configuração tenha funcionalidade multilinha, como um arquivo de configuração XML, o **c2a** não será recomendável.

Execute a ferramenta de conversão com a opção "**-audit**" ao digitar:

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

A ferramenta requer um arquivo de entrada (**input.txt**) que contenha uma lista de arquivos de configuração a serem auditados, além de um nome de arquivo de saída para o arquivo de auditoria.

O script Perl **c2a.pl** depende de dois arquivos essenciais: **c2a.map** e **c2a_regex.map**. O script examina cada linha do arquivo de configuração auditado e verifica se a primeira palavra da linha coincide com o "tipo" no arquivo **c2a.map** (por exemplo: HTTP, SENDMAIL etc.) e o valor associado a ele. Por exemplo: as configurações HTTP forem auditadas, verifica se a palavra coincide com alguma das palavras-chave no arquivo **c2a.map**. Em caso positivo, aplica a expressão regex de **c2a_regex.map** à linha e extrai a configuração e o valor. Somente as configurações para as quais exista uma entrada em **c2a.map** serão auditadas.

Os arquivos de configuração que não requerem auditoria podem ser comentados com o uso do caractere "#".

Caso queira converter as configurações assinaladas como comentários no arquivo de configuração para o formato **.audit**, edite o **c2a.pl** e defina "`$ENFORCE_COMMENT = 1;`".

Como no caso anterior, se o arquivo de saída não for especificado, o resultado será gravado em **~/c2a/op.audit**.

Atualmente, a Tenable fornece configurações MAP para HTTP, Sendmail, Sysctl e Nessus. As configurações de aplicativos adicionais podem ser adicionadas com o uso de um script Perl **cmv.pl**. Consulte a seção seguinte para obter mais informações.

Criação de um arquivo MAP

A criação de um arquivo MAP para uma aplicação é simples. Basta executar o script **cmv.pl** conforme abaixo:

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

Onde:

- "**regex**" é o regex para extrair a definição da configuração e o par de valores. Normalmente, este é da forma "<name> = <value>". Mas em alguns casos, pode ser ligeiramente diferente, onde "=" pode ser substituído por um espaço, tabulação etc.
- "**tag**" é basicamente a palavra-chave que deseja marcar na aplicação que estiver sendo auditada. A palavra-chave **tag** vincula o **config_file** às palavras-chave em **c2a.map** e regex em **c2a_regex.map**, por isso, é importante que o tag em cada um destes arquivos seja o mesmo.
- "**config_file**" é o arquivo para o qual está sendo criado um arquivo MAP.

Por exemplo: para auditar definições de configuração para VSFTPD, execute os passos a seguir:

1. Primeiro, use **cmv.pl** conforme o exemplo abaixo:

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

   Isto criará o arquivo **tag.map** (por exemplo: **VSFTPD.map**). Por padrão, todas as linhas assinaladas como comentários serão ignoradas. Se todas as variáveis forem consideradas, altere o valor de **$ENFORCE_COMMENT** de "0" para "1" e, em seguida, execute o script novamente.

2. Inspecione e acrescente o arquivo MAP a **c2a.map**.

   Verifique o arquivo **VSFTPD.map** com relação a quaisquer valores indesejados que possam corresponder inadvertidamente à expressão regex. Após verificar se todas as palavras-chave estão corretas, acrescente-as a **c2a.map**.

3. Atualize **c2a_regex.map** com a mesma expressão usada por **cmv.pl** conforme abaixo:

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

Nota: é a mesma expressão regex usada pelo script Perl **cmv.pl**.

4. Atualize **input.txt** com o local do arquivo de configuração VSFTPD:

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

5. Execute o script **c2a.pl**:

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

6. Por último, verifique o arquivo de saída:

```
# vi op.audit
```

Outros usos para a ferramenta c2a

A Tenable incluiu várias entradas aos arquivos **c2a.map** e **c2a_regex.map** para habilitar a auditoria do Sendmail do Very Secure FTP Daemon (VSFTPD), Apache, arquivo Red Hat **/etc/sysctl.conf** e Nessus. Outros softwares serão acrescentados em breve. Caso queira enviar novos mapeamentos à Tenable para compartilhar com outros usuários do Nessus, envie-os ao endereço nessus-support@tenable.com.

Assim sendo, o script **c2a.pl** pode ser usado para ajudar a criar arquivos **.audit** do Nessus para várias aplicações ativas do Unix. Considere as seguintes ideias:

- Caso sua organização tenha muitos firewalls baseados em Unix, um arquivo **.audit** pode ser gerado para auditar as definições comuns e específicas de cada firewall. Por exemplo: se todos os firewalls precisarem ter filtros de endereços RFC 1918, as regras reais de firewall poderão ser analisadas.
- Se diversos aplicativos padrão forem executados fora do cron, os vários crontabs poderão ser auditados para se verificar se os aplicativos corretos estão sendo executados no momento correto.
- Para acesso centralizado, é possível verificar as configurações Syslog, Syslog-NG e Logrotate nos sistemas Unix remotos.

Ajuste fino dos arquivos .audit

Para finalizar, a saída do script **c2a.pl** pode ser também editada manualmente. Por exemplo: avalie a possibilidade de combinar as regras de soma de verificação MD5 às regras do arquivo File_Content_Check para transformá-las em uma só regra. A saída gerada pelo script **c2a.pl** também pressupõe que um arquivo de configuração esteja sempre em um só local. Avalie a possibilidade de modificar a palavra-chave "**file**" (arquivo) para especificar outros locais em que um arquivo de configuração possa estar localizado.

Se houver conteúdo indesejado nas configurações de arquivo remoto, pode-se acrescentar outras verificações com a palavra-chave File_Content_Check_Not. Isto permite executar auditorias para definições que devem estar presentes e outras que não devem.

Conversão de listas de pacotes Unix em arquivos .audit com p2a

A ferramenta **p2a.pl** foi projetada para auxiliar os auditores na criação de arquivos **.audit** para a instalação de configurações de pacotes em sistemas Linux e Solaris 10 baseados em RPM. Por exemplo: caso queira que todos os servidores Web Linux em uma determinada rede tenham a mesma base RPM que o host mestre X, é possível executar esta ferramenta no host X para criar um arquivo **.audit** contendo todos os pacotes de RPM nesse sistema. Em seguida, este arquivo **.audit** seria usado com o Nessus para executar uma varredura em outros servidores Web para verificação da conformidade.

Como opção, esta ferramenta pode ser usada para a criação de um arquivo de auditoria de uma listagem de texto de pacotes RPM ou Solaris 10. A ferramenta requer uma lista de pacotes, um por linha, em um arquivo de entrada e formata

um arquivo **.audit** para o sistema de destino. O arquivo **.audit** gerado pode ser usado posteriormente para a varredura de modificações a pacotes de instalação básicos.

Obtenção e instalação da ferramenta

A ferramenta **p2a** é um arquivo **tar** compactado de um único script Perl e um arquivo de ajuda **ReadMe.txt**. Pode ser obtido no Tenable Support Portal no endereço https://support.tenable.com/.

Extraia o conteúdo de **p2a-x.x.x.tar.gz** no computador local com o seguinte comando:

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

Isto criará um diretório "**p2a**" sob o diretório atual e extrairá os arquivos para ele.

Caso queira extrair o conteúdo para um diretório de sua escolha, use o seguinte comando:

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

Após descompactar o arquivo, visualize os arquivos a seguir no diretório **~/p2a**:

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

Torne executável o script executando:

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

Uso

Execute o script Perl da seguinte maneira:

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

"**-h**" é um argumento independente que exibe a ferramenta de ajuda.

Criar arquivos de saída com base em todos os pacotes instalados

Se o script for executado unicamente com a opção "**-o**", executará um comando de sistema para extrair todos os nomes de pacotes do sistema instalados no computador local e o arquivo **.audit** resultante será gravado em **/path/to/outputfile.audit**.

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

Os arquivos de saída devem incluir a extensão **.audit** para que o script seja executado. Do contrário, será gerado uma mensagem de erro indicando que a extensão do arquivo está incorreta.

Criar arquivo de saída com base na lista de pacotes e enviar à tela

Execute **p2a** para enviar toda saída resultante à janela do terminal com a seguinte sintaxe:

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

Esta opção requer um arquivo de entrada e gerará uma saída na janela do terminal (**stdout**) que poderá ser copiada e colada em seu arquivo **.audit**. O arquivo de entrada deve ser formatado com um pacote por linha e sem delimitadores adicionados.

Exemplo:

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

Uma vez que muitos sistemas baseados em Unix podem ter mais de mil pacotes instalados, a quantidade de saída poderá exceder o buffer de rolagem e dificultar toda a saída.

Criar arquivo de auditoria com base em um arquivo de entrada especificado

A execução de **p2a** com argumentos tanto de entrada como de saída resulta em uma listagem de pacotes formatada e gera um arquivo **.audit** no local especificado.

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

Os arquivos de entrada devem ser formatados com um pacote por linha e sem delimitadores adicionados.

Exemplo:

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

Os arquivos de saída devem incluir a extensão **.audit** para que o script seja executado. Do contrário, será gerado uma mensagem de erro indicando que a extensão do arquivo está incorreta.

Exemplo de uso da interface com o usuário do Nessus

Obtenção das verificações de conformidade

Os clientes do ProfessionalFeed já possui as verificações de conformidade de seu scanner Nessus e vários arquivos **.audit** estão disponíveis no Tenable Support Portal no endereço https://support.tenable.com/. Para confirmar isto, execute a interface com o usuário, autentique e gerencie ou edite uma política existente. Na guia “Plugins”, localize a opção “Policy Compliance” da família, clique no nome da família de plugins e verifique se os seguintes plugins são exibidos:

- Cisco IOS Compliance Checks
- Database Compliance Checks
- IBM iSeries Compliance Checks
- PCI DSS Compliance

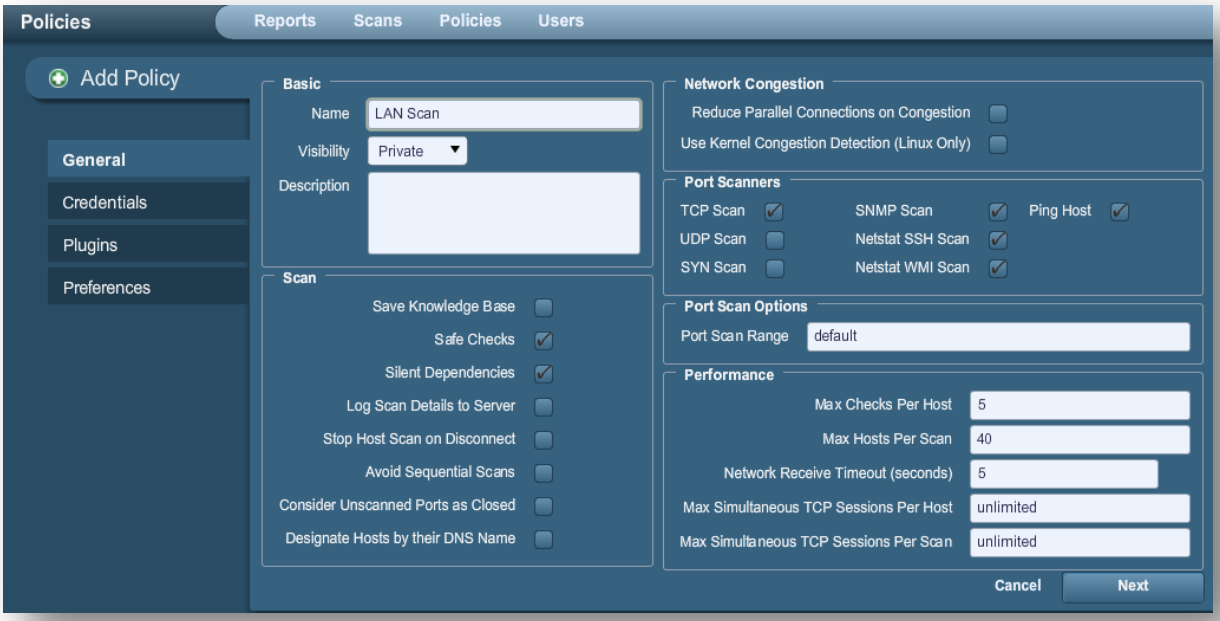
- PCI DSS Compliance: Database Reachable from the Internet
- PCI DSS Compliance: Handling False Positives
- PCI DSS Compliance: Insecure Communication Has Been Detected
- PCI DSS Compliance: Remote Access Software Has Been Detected
- PCI DSS Compliance: Passed
- PCI DSS Compliance: Tests Requirements
- Unix Compliance Checks
- Windows Compliance Checks
- Windows File Contents Compliance Checks

Configuração de uma política de varredura

Para possibilitar as verificações de conformidade no Nessus, uma política de varredura deverá ser criada com os seguintes atributos:

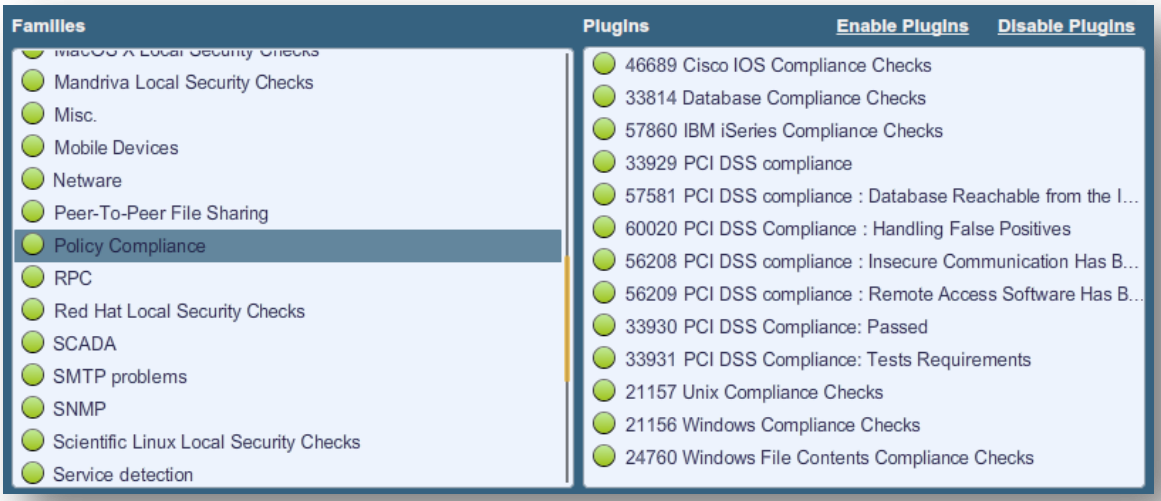
- Ativar os plugins de verificação de conformidade que estiverem na família de plugins “Policy Compliance”
- Especificar uma ou mais políticas de conformidade .audit como preferência
- Especificar as credenciais para acesso ao servidor de destino, incluindo as credenciais do banco de dados na guia “Preferences” (Preferências), se for o caso
- Habilitar as dependências de plugin

É importante compreender as verificações nos arquivos **`.audit`** selecionado, principalmente se forem criados arquivos personalizados. Ao usar dois arquivos **`.audit`** na mesma varredura, ambos serão combinados para que os resultados de cada arquivo sejam produzidos em uma só varredura. Se houver resultados conflitantes entre os arquivos, o usuário receberá um resultado aprovado e um reprovado para ambos. Certifique-se sempre de conferir as conclusões em seus relatórios.



Para criar uma política, acesse a interface de usuário do Nessus, autentique e selecione “Policies” (Políticas). Editar uma política existente ou criar uma nova. É possível especificar as credenciais para acessar o servidor de destino na guia “**Credentials**” (Credenciais) à esquerda.

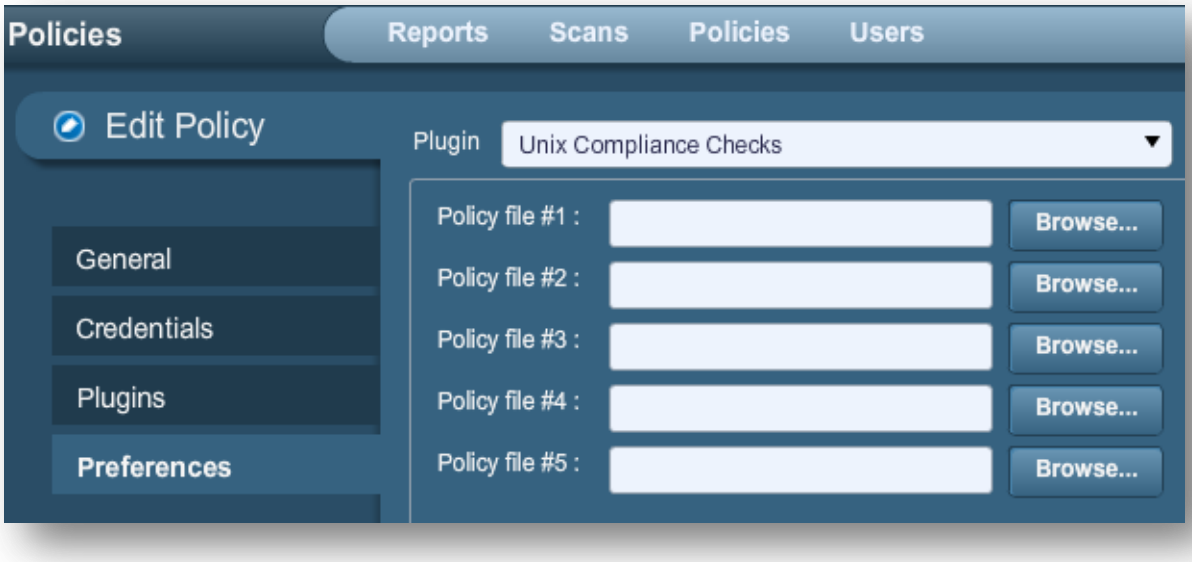
Na guia “**Plugins**”, ative “Policy Compliance” (Conformidade com a política) da família de plugins e certifique-se de que “**`auto_enable_dependencies`**” está definido como “yes” no arquivo **`nessusd.conf`** (configuração padrão):



Editar política de varredura para verificar se Policy Compliance está disponível

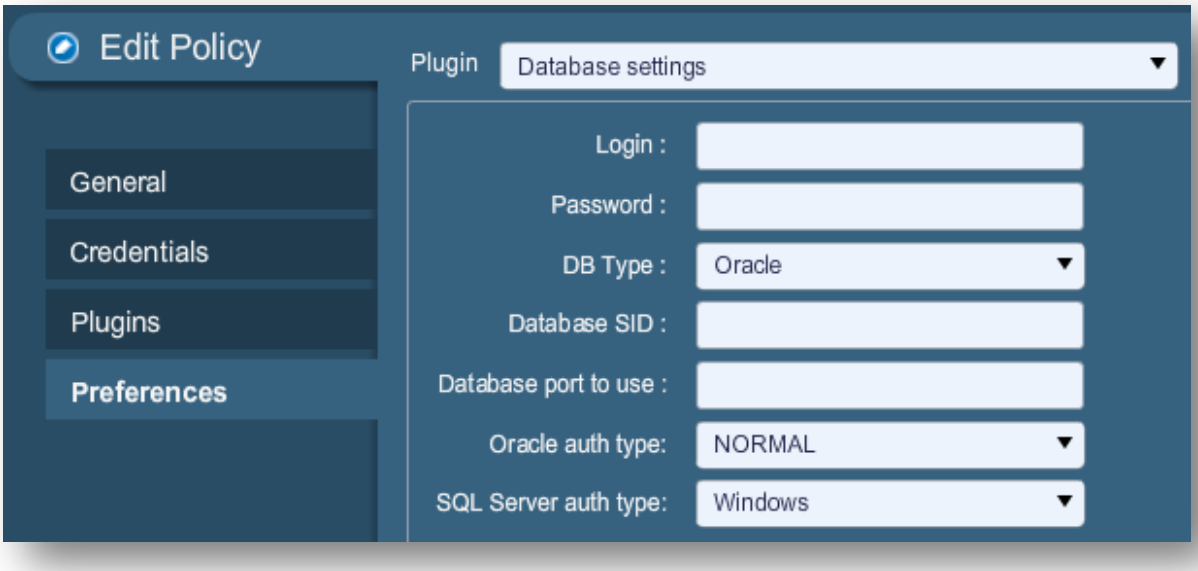
Para ativar o uso de um arquivo **`.audit`**, na guia “**Preferences**” (Preferências), selecione “Cisco IOS Compliance Checks” (Verificações de conformidade Cisco IOS), “Unix Compliance Checks”(Verificações de conformidade Unix), “Windows Compliance Checks”(Verificações de conformidade Windos), “Windows File Content Compliance

Checks”(Verificações de conteudo de arquivos Windows), “IBM iSeries Compliance Checks”(Verificações de conformidade IBM iSeries), ou “Database Compliance Checks”(Verificações de conformidade de banco de dados) no menu suspenso. Há cinco campos em cada seção que podem especificar arquivos `.audit` distintos. Os arquivos especificados devem ser pré-descarregados no sistema cliente local do Tenable Support Portal.



Exemplo de caixa de diálogo da interface com o usuário do Nessus para especificar arquivos Unix .audit

Se tiver sido selecionado “Database Compliance Checks”(Verificações de conformidade de banco de dados) no menu suspenso anterior, os parâmetros de login do banco de dados deverão ser inseridos em “**Preferences**” -> “**Database Setting**”:



O menu “Database Settings” possui diversas opções disponíveis, tais como:

Opção	Descrição
Login	O nome de usuário do banco de dados.
Password	A senha para o nome de usuário fornecido.
DB Type	Oracle, SQL Server, MySQL, DB2, Informix/DRDA e PostgreSQL são permitidos.
Database SID	ID do sistema de banco de dados para auditar. Aplicável somente a Oracle, DB2 e Informix.
Oracle auth type	Normal, Sysoper Sysdba são permitidos.
SQL Server auth type	Compatível com Windows ou SQL Server.

Consulte o administrador do banco de dados local para obter os valores corretos destes campos.

Neste ponto, clique em “Save” (Salvar) na parte inferior da janela para concluir a configuração. A nova política de varredura será adicionada à lista de políticas de varredura gerenciadas.

Realizar uma varredura

A execução de uma varredura com verificações de conformidade ativadas não difere da execução de outras varreduras de auditoria de correção local ou de varreduras de rede normais. Essas varreduras podem ser combinadas e executadas ao mesmo tempo, se necessário.

Exemplo de resultados

No Nessus 4, todos os resultados de conformidade são enviados com a identificação do plugin que estiver executando o teste. No exemplo abaixo, todos os dados retornados relativos a um servidor Windows verificado originam-se do plugin Windows Compliance `.nbin` identificado como plugin 21156.

Exemplo de resultados de conformidade durante a verificação de um servidor Windows

O relatório HTML, que pode ser descarregado da guia “Reports” na interface de usuário do Nessus 4, realça os testes de conformidade aprovados na cor azul com a mensagem “Passed” (Aprovado). Os testes reprovados são realçados em vermelho com a mensagem “Failed” (Reprovado) e todos os itens não auditados são realçados em amarelo com a mensagem “Error” (Erro).

No exemplo acima, somente quatro itens são mostrados. Cada item resulta de uma verificação de política de conformidade de controle de acesso quanto à presença de serviços e protocolos desnecessários e sem segurança. Alguns desses serviços não estavam em execução e satisfizeram as expectativas da política **`.audit`**, ao passo que outros (como o serviço de “registro remoto”) estavam em execução e foram relacionados como “Failed” (Reprovados). Recomenda-se que os itens relacionados como “Failed” sejam configurados para satisfazer a política de acordo com as normas de segurança do usuário.

Exemplo de uso da linha de comando do Nessus para Unix

Obtenção das verificações de conformidade

Caso o daemon do Nessus seja configurado para receber o ProfessionalFeed de plugins, haverá cinco arquivos **`.nbin`** de conformidade no diretório de plugins.

Obtenha os arquivos **`.audit`** necessários no Tenable Support Portal no endereço https://support.tenable.com/ e coloque-os no diretório de plugins do scanner. Na maioria das distribuições, a localização padrão é o seguinte diretório:

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

Os plugins estarão presentes entre os mais de 40.000 arquivos de plugin **`.nasl`** usados pelo Nessus para a verificação de vulnerabilidades. É possível pesquisá-los na extensão **`.nbin`** conforme o exemplo a seguir:

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

cisco_compliance_check.nbin                 database_compliance_check.nbin
```

```
compliance_check.nbin                          unix_compliance_check.nbin
compliance_check_windows_file_content.nbin
```

Pode haver outros arquivos **.nbin** entregues pela Tenable, como o Skype plugin, que nada tem a ver com a execução de verificações de conformidade.

Se não tiver acesso local ao daemon do Nessus, mas possui um nome de usuário e uma senha para efetuar o login no servidor, é possível solicitar uma lista de plugins com a opção "**-p**" do cliente da linha de comando **nessus**, conforme o exemplo a seguir:

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

A consulta pode demorar alguns minutos para ser executada. Se a consulta for feita com sucesso mas não retornar nenhuma informação, as verificações de conformidade não estão instaladas no scanner Nessus remoto.

Uso dos arquivos .nessus

O Nessus pode salvar políticas de varredura configuradas, alvos de rede e relatórios como arquivo **.nessus**. A seção "Exemplo de uso da interface de usuário do Nessus" descreve a criação de um arquivo **.nessus** que contém uma política de varredura para verificação de conformidade. Para obter de instruções sobre como executar uma varredura por linha de comando com o arquivo **.nessus**, consulte o "Guia do Usuário do Nessus" disponível em: http://www.tenable.com/products/nessus/documentation.

Uso dos arquivos .nessusrc

O cliente da linha de comando do Nessus possui também a capacidade de exportar políticas de varredura configuradas como arquivos **.nessusrc**. Isto pode ser conveniente para habilitar a varredura da linha de comando. A seção "Exemplo de uso da interface de usuário do Nessus" descreve os passos para a criação de uma política para verificações de conformidade no Nessus.

Para acessar uma linha de comando com o Nessus, é precisa especificar o seguinte:

- Plugins de verificação de conformidade do Unix, Windows ou banco de dados
- Credenciais do(s) host(s) de destino que está(ão) sendo verificado(s)
- Um ou mais arquivos **.audit** para executar os plugins de verificação de conformidade
- Quais dependências foram habilitadas

As entradas de dados importantes em um arquivo **.nessusrc** têm o seguinte formato (com uns conteúdos omitidos):

```
begin(SERVER_PREFS)
…
auto_enable_dependencies = yes
…
end(SERVER_PREFS)
```

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

O exemplo anterior omitiu diversos fragmentos de dados que especificam o que uma varredura pode executar. O conteúdo omitido inclui a ativação do arquivo de política **.audit** específico em uso, a ativação de dependências e os próprios plugins de conformidade.

Realizar uma varredura

A execução de uma varredura com verificações de conformidade ativadas não difere da execução de outras varreduras de auditoria de correção local ou de varreduras de rede normais. As varreduras podem ser combinadas de modo que sejam executadas ao mesmo tempo, se necessário.

Exemplo de resultados

Da mesma maneira que os clientes da interface gráfica do usuário, todos os resultados conformes ou não detectados são relatados no seguinte formato:

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

Os dados se encontram no formato de relatório **.nsr** do Nessus. Todos são eventos não conformes.

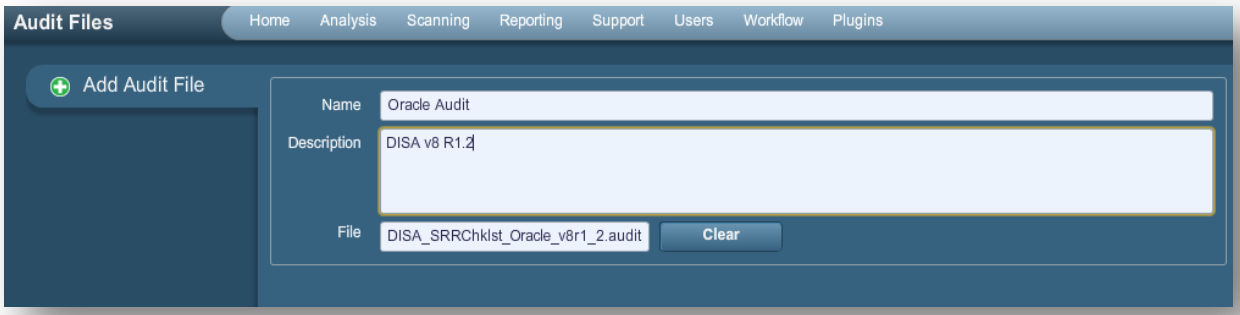
Uso do SecurityCenter

As informações a seguir baseiam-se na execução das varreduras de conformidade com o SecurityCenter 4 ou superior. Para usuários do Security Center 3.x, consulte “Security Center 3.4 Documentation”, disponível no Tenable Support Portal: https://support.tenable.com/.

Obtenção das verificações de conformidade

Todos os clientes do SecurityCenter têm acesso aos plugins do Nessus ProfessionalFeed. Isto inclui os plugins de verificação de conformidade do Cisco, Unix, Windows, Windows File Contents e bancos de dados. Os plugins permitem que o usuário carregue e execute verificações de conformidade com o uso de arquivos **.audit** fornecidos pela Tenable. Obtenha um dos arquivos **.audit** necessários no Tenable Support Portal, no endereço https://support.tenable.com/. Os

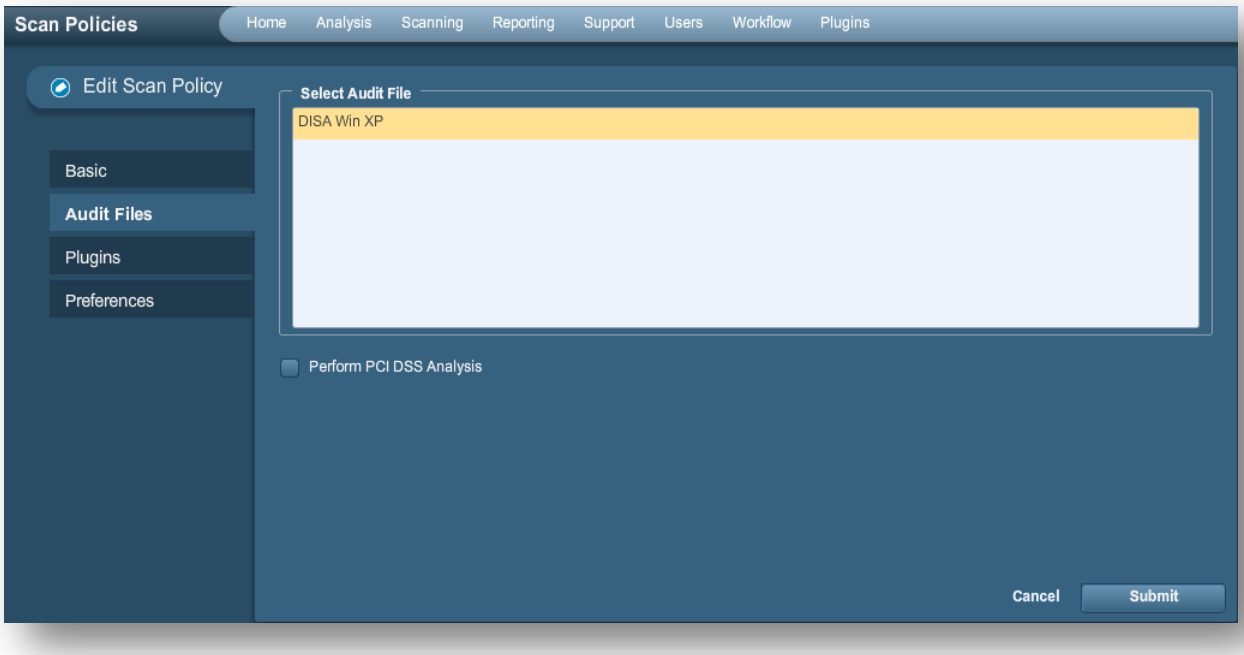
arquivos **.audit** podem ser carregados ao SecurityCenter por qualquer usuário com a permissão “Create Audit Files” (Criar arquivos de auditoria) com a ferramenta “Add Audit File” (Adicionar arquivo de auditoria) da guia “Support” (Suporte).



Todos os arquivos **.audit** carregados no SecurityCenter estarão disponíveis para qualquer usuário do SecurityCenter com a permissão “Create Policies” (Criar políticas). O SecurityCenter também processará a distribuição de arquivos **.audit** novos e atualizados para os scanners Nessus.

Configuração de política de varredura para execução de uma auditoria de conformidade

Para executar uma varredura de conformidade com o SecurityCenter, os usuários devem configurar uma política de varredura com as definições relacionadas à conformidade. Esta política especifica as opções de varredura, os arquivos de auditoria, os plugins ativados e as preferências avançadas. A segunda página da “Scan Policy” especifica os arquivos **.audit** a serem usados para a auditoria de conformidade.



Neste caso, um ou mais arquivos **.audit** podem ser selecionados ao realçar o arquivo **.audit** e clicar em “Submit” (Enviar). Para selecionar vários arquivos **.audit**, use a tecla “Ctrl” para realizar várias seleções. Se for necessária uma análise básica do PCI DSS, certifique-se de que a caixa de seleção “Perform PCI DSS Analysis” esteja selecionada antes do envio.

O padrão PCI DSS (Payment Card Industry Data Security Standard) é um conjunto completo de normas de segurança estabelecido pelos membros fundadores do PCI Security Standards Council, que inclui Visa, American Express, Discover Financial Services e MasterCard. O PCI DSS proporciona um método comum de proteção dos dados confidenciais do portador do cartão de todas as bandeiras e é usado por vários fornecedores de comércio eletrônico que aceitam e armazenam dados de cartões de crédito.

A Tenable fornece doze plugins para todos os usuários do SecurityCenter que possuem processos de auditoria de PCI DSS automatizados. Para obter a lista de plugins, consulte a tabela abaixo.

Esses plugins avaliam os resultados e a configuração da varredura para determinar se o servidor de destino satisfaz os requisitos de conformidade de PCI publicados. Os plugins não executam varredura real, mas analisam os resultados de outros plugins. Para ativar os plugins PCI DSS, marque a caixa de seleção “Perform PCI DSS Analysis” (Realizar análise PCI DSS) na tela “Compliance” (Conformidade).

Após selecionar o(s) arquivo(s) **`.audit`** e as definições de PCI DSS desejadas, clique na guia “Plugins” para confirmar as definições de plugins. Os itens da família de plugins “Policy Compliance” devem ser ativados na política para executar uma varredura de conformidade.

Quando o usuário seleciona um ou mais arquivos de auditoria na guia “Audit Files” da política de varredura, o plugin correto é ativado automaticamente na guia “Plugins”. O SecurityCenter analisa o(s) arquivo(s) **`.audit`** selecionado(s) e, com base no tipo especificado dentro do arquivo, o(s) plugin(s) correto(s) são ativados.

Na família “Policy Compliance”, há treze plugins disponíveis para a auditoria de conformidade. Os plugins são os seguintes:

ID do plugin	Nome do plugin	Descrição do plugin
21156	Windows Compliance Checks	Usado para auditar definições de configuração comuns do Windows.
21157	Unix Compliance Checks	Usado para auditar definições de configuração comuns do Unix.
24760	Windows File Contents Compliance Checks	Usado para auditar conteúdo confidencial do arquivo em servidores Windows.
33814	Database Compliance Checks	Usado para auditar definições de configuração comuns do banco de dados.
33929	PCI DSS compliance	Determina se o servidor web remoto está vulnerável a ataques com criação de scripts entre instalações (XSS, cross-site scripting), se a versão antiga da criptografia SSL2.0 está instalada, se está executando software obsoleto ou é afetado por vulnerabilidades perigosas (pontuação básica CVSS >= 4).
57581	PCI DSS Compliance: Database Reachable from the Internet	Detecta a presença de um banco de dados acessível na Internet, resultando em uma auditoria de configuração fracassada.
60020	PCI DSS Compliance: Handling False Positives	Comenta o tratamento adequado de falsos positivos em varreduras de PCI DSS.
56208	PCI DSS Compliance: Insecure Communication Has Been Detected	Determina se uma porta, um protocolo ou um serviço inseguro foi detectado, o que resultaria em falha de conformidade.
56209	PCI DSS Compliance: Remote Access Software Has Been Detected	Detecta a presença de software de acesso remoto, o que resultaria em falha de conformidade.

33930	PCI DSS Compliance: Passed	Com base nas informações disponíveis da varredura, o Nessus não detecta nenhuma falha impeditiva neste host.
33931	PCI DSS Compliance: Tests Requirements	Analisa se a varredura do Nessus satisfaz ou não os requisitos de teste de PCI. Este relatório pode ser insuficiente para certificar este servidor, mesmo que os testes técnicos sejam aprovados.
46689	Cisco IOS Compliance Checks	Usado para auditar definições de configuração comuns do dispositivo Cisco.
57860	IBM iSeries Compliance Checks	Usado para verificar definições de configuração comuns do IBM iSeries.

Gerenciamento de credenciais

Uma vantagem do SecurityCenter na execução de varreduras com base em dados credenciados é que pode ajudar a gerenciar as credenciais em uso. Para criar as credencials do SecurityCenter, selecione a guia “Support”, clique em “Credentials” e, em seguida, clique em “Add”.

As credenciais Unix, Windows e Cisco são armazenadas e gerenciadas em separado da política de varredura. As credenciais podem ser criadas com visibilidade “User” para o usuário atual ou com visibilidade “Organizational” onde possam ser usados por outros usuários do SecurityCenter. Isto permite que os usuários trabalhem com os resultados das varreduras e executem novas varreduras sem precisar conhecer as credenciais envolvidas na varredura.

São necessárias credenciais adicionais para a varredura de sistemas de banco de dados. As credenciais ficam armazenadas dentro da política de varredura e são configuradas por meio de “Database settings” (plugin 33815) nas preferências da política de varredura. Estas credenciais são configuradas separadamente das credenciais especificadas no parágrafo anterior.

Análise dos resultados

O SecurityCenter pode ser usado para analisar e relatar os dados de conformidade que retornam das varreduras do Nessus de várias maneiras. Os relatórios comuns incluem:

- Listagem de todas as vulnerabilidades conformes ou em não conformes por grupo de ativos
- Listagem de todas as vulnerabilidades conformes ou não conformes por host ou rede
- Resumo de todos os problemas de não conformidade
- Definições da auditoria de banco de dados quanto a erros de configuração
- Geração de relatório do status do usuário ou do software com base nas necessidades de TI

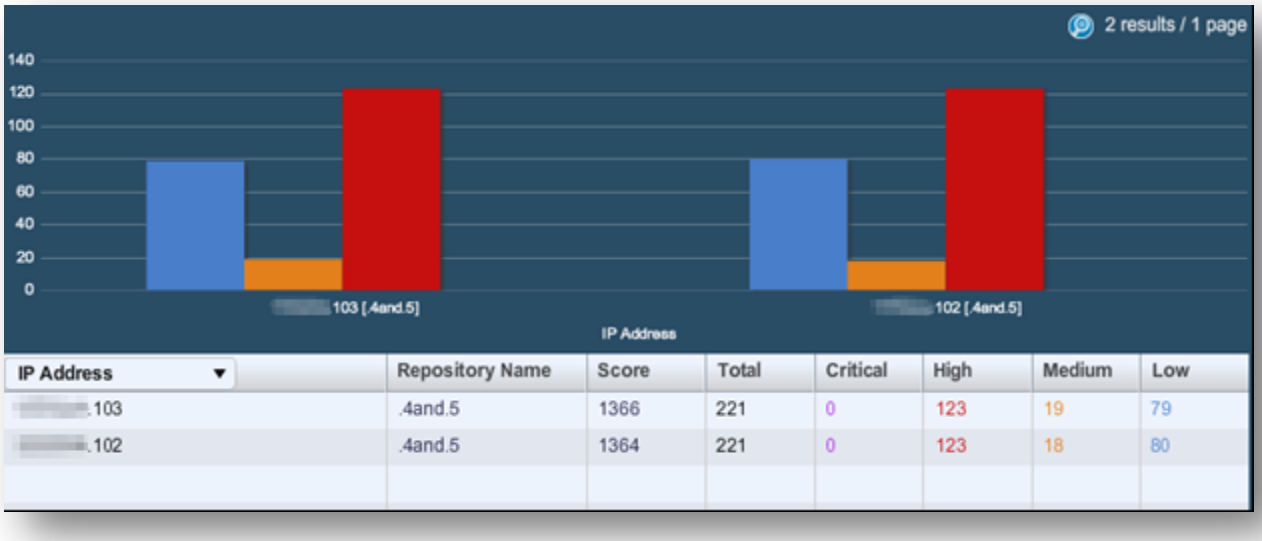
Uma vez localizados os dados de conformidade pelo SecurityCenter, as ferramentas de emissão de tickets, geração de relatórios e ferramentas analíticas podem ser usadas para determinar o procedimento adequado para a reconfiguração

dos dispositivos auditados. Estes dados podem ser analisados juntamente com outras informações de vulnerabilidade, patches de segurança ou detectadas de maneira passiva.

Alguns de exemplos do SecurityCenter usados para analisar as informações de conformidade sobre hosts verificados são mostrados a seguir:

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

Exemplo de listagem de Dados de Auditoria de Conformidade com o SecurityCenter



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

Exemplo de listagem de Dados de Auditoria de Conformidade por servidor com o SecurityCenter

Para obter mais informações sobre o uso do SecurityCenter, consulte a documentação do SecurityCenter disponível em https://support.tenable.com/.

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,
- **Referência de Verificações de Conformidade do Nessus** – guia completo da sintaxe das verificações de conformidade do Nessus.
- **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/.

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