Konfigurieren des SCP-Push von E-Mail-Protokollen auf der ESA

Inhalt

Einführung

In diesem Dokument wird beschrieben, wie Sie Secure Copy Push (SCP) von Mail-Protokollen (oder anderen Protokolltypen) von einer Cisco E-Mail-Security-Appliance (ESA) zu einem externen Syslog-Server einrichten und konfigurieren.

Hintergrundinformationen

Ein Administrator erhält möglicherweise Fehlerbenachrichtigungen, in denen angegeben wird, dass Protokolle nicht über SCP gesendet werden können, oder es können Fehlerprotokolle mit einer oder mehreren nicht übereinstimmenden Schlüsseln vorliegen.

Voraussetzungen

Auf dem Syslog-Server, an den die ESA SCP-Protokolldateien angibt:

1. Stellen Sie sicher, dass das zu verwendende Verzeichnis verfügbar ist.
2. Überprüfen Sie '/etc/ssh/sshd_config' für die AuthorizedKeysFile-Einstellungen. Dadurch wird SSH angewiesen, authorized_keys zu akzeptieren und im Hauptverzeichnis des Benutzers nach Schlüsselnamen zu suchen, die in der Datei .ssh/authorized_keys geschrieben wurden:
   ```
   AuthorizedKeysFile      %h/.ssh/authorized_keys
   ```
3. Überprüfen Sie die Berechtigungen für das zu verwendende Verzeichnis. Möglicherweise müssen Sie Berechtigungen ändern: Die Berechtigungen für '$HOME' sind auf 755 festgelegt.Die Berechtigungen für '$HOME/.ssh' sind auf 755 festgelegt.Die Berechtigungen für '$HOME/.ssh/authorized_keys' sind auf 600 festgelegt.

Einschränkungen und Berechtigungen auf Dateiebene unter UNIX/Linux

Es gibt drei Arten von Zugriffsbeschränkungen:

```
Permission Action chmod option ==================================== read (view) r or 4 write (edit) w or 2 execute (execute) x or 1
```

Es gibt auch drei Arten von Benutzerbeschränkungen:

```
User ls output ================= owner -rwx------ group ----rwx--- other -------rwx
```

Ordner-/Verzeichnisberechtigungen:

```
Permission Action chmod option ============================================================ read (view contents: i.e., ls command) r or 4 write (create or remove files from dir) w or 2 execute (cd into directory) x or 1
```

Numerische Notation:

Eine weitere Methode zur Darstellung von Linux-Berechtigungen ist eine Oktalnotation, wie in dargestellt `stat -c %a`. Diese Notation besteht aus mindestens drei Ziffern. Jede der drei Ziffern rechts stellt eine andere Komponente der Berechtigungen dar: Eigentümer, Gruppe und andere.

Jede dieser Ziffern ist die Summe der zugehörigen Komponentenbits im binären numerischen System:

```
Symbolic Notation Octal Notation English ========== ---------- 0000 no permissions --- x--x--x 0111 execute --w--w--w- 0222 write --wx-wx-wx 0333 write & execute -r--r--r-- 0444 read -r-xr-xr-x 0555 read & execute -rw-rw-rw- 0666 read & write -rwxrwxrwx 0777 read. write & execute
```

In Schritt 3 wird empfohlen, das $HOME-Verzeichnis auf 755 festzulegen: 7=`rwx` 5 =`r-x` 5 =`r-x`

Das bedeutet, dass das Verzeichnis über Standardberechtigungen verfügt. `-rwxr-xr-x` (in Oktalnotation als 0755 dargestellt).

Konfigurieren des SCP-Push von E-Mail-Protokollen auf der ESA

1. Führen Sie den CLI-Befehl **logconfig aus**.
2. Wählen Sie die **neue** Option aus.
3. Wählen Sie den Protokolldatentyp für dieses Abonnement aus. Dies ist "1" für IronPort Text Mail Logs oder einen anderen von Ihnen gewählten Protokolldateityp.
4. Geben Sie den Namen für die Protokolldatei ein.
5. Wählen Sie die entsprechende Protokollstufe aus. In der Regel müssen Sie "3" als Informations- oder eine andere Protokollstufe Ihrer Wahl auswählen.
6. Wenn Sie gefragt werden 'Choose the method to obtain the logs' (Methode zum Abrufen der Protokolle auswählen), wählen Sie "3" für **SCP Push aus**.
7. Geben Sie die IP-Adresse oder den DNS-Hostnamen ein, an die die Protokolle übermittelt werden sollen.
8. Geben Sie den Port ein, mit dem eine Verbindung zum Remote-Host hergestellt werden soll.
9. Geben Sie das Verzeichnis auf dem Remotehost ein, um Protokolle zu erstellen.

10. Geben Sie einen Dateinamen ein, der für Protokolldateien verwendet werden soll.
11. Konfigurieren Sie ggf. systembasierte eindeutige Identifikatoren wie *$hostname*, *$serialnumber*, um an den Protokolldateinamen anzuhängen.
12. Legen Sie vor der Übertragung die maximale Dateigröße fest.
13. Konfigurieren Sie ggf. ein zeitbasiertes Rollover der Protokolldateien.
14. Wenn Sie gefragt werden "Möchten Sie die Hostschlüsselprüfung aktivieren?", geben Sie "J" ein.
15. Sie erhalten dann die Meldung "Legen Sie die folgenden SSH-Schlüssel in Ihre authorized_keys-Datei, damit die Protokolldateien hochgeladen werden können."
16. Kopieren Sie diesen Schlüssel, da Sie den SSH-Schlüssel in Ihrer Datei 'authorized_keys' auf dem Syslog-Server speichern müssen. Fügen Sie den von logconfig angegebenen Schlüssel in die Datei $HOME/.ssh/authorized_keys auf dem Syslog-Server ein.
17. Führen Sie in der ESA den CLI-Befehl **commit aus**, um Konfigurationsänderungen zu speichern und zu bestätigen.

Die Konfiguration des Protokolls kann auch über die GUI erfolgen: **Systemverwaltung > Protokollabonnements**

> **Hinweis**: Im Kapitel Protokollierung des ESA-Benutzerhandbuchs finden Sie ausführliche Informationen und weitere Informationen.

Bestätigung

Hostschlüsselkonfiguration

Führen Sie den Befehl **logconfig > hostkeyconfig aus**. Es sollte ein Eintrag für den als "ssh-dss" konfigurierten Syslog-Server mit einem abgekürzten Schlüssel angezeigt werden, der dem während der Konfiguration bereitgestellten Schlüssel ähnelt.

```
myesa.local > logconfig
...
[]> hostkeyconfig

Currently installed host keys:
1. 172.16.1.100 ssh-dss AAAAB3NzaC1kc3MAAACBAMUqUBGztO0T...OutUns+DY=
```

Systemprotokolle

Systemprotokolle zeichnen Folgendes auf: Bootinformationen, Alarmmeldungen zum Ablauf von Lizenzen für virtuelle Appliances, DNS-Statusinformationen und Kommentare, die Benutzer mit dem Commit-Befehl eingegeben haben. Systemprotokolle sind nützlich, um den grundlegenden Zustand der Appliance zu beheben.

Wenn Sie den Befehl **tail system_logs** über die Kommandozeile ausführen, wird der Systemstatus live angezeigt.

Sie können auch den CLI-Befehl **jetzt** auswählen und die Nummer auswählen, die der Protokolldatei zugeordnet ist. Sie sehen diese Protokolldatei SCP zu Ihrem Syslog-Server in system_logs:

```
myesa.local > tail system_logs

Press Ctrl-C to stop.
Thu Jan 5 11:26:02 2017 Info: Push success for subscription mail_logs: Log
mail_logs.myesa.local.@20170105T112502.s pushed via SCP to remote host 172.16.1.100:22
```

Erweiterte Fehlerbehebung

Wenn weiterhin Probleme mit der Verbindung zum Syslog-Server auftreten, vom lokalen Host aus und mithilfe von ssh, führen Sie "ssh testuser@hostname -v" aus, um den Benutzerzugriff im ausführlichen Modus zu testen. Dies kann die Fehlerbehebung erleichtern, um anzuzeigen, wo die ssh-Verbindung nicht erfolgreich ist.

```
$ ssh testuser@172.16.1.100 -v
OpenSSH_7.3p1, LibreSSL 2.4.1
debug1: Reading configuration data /Users/testuser/.ssh/config
debug1: /Users/testuser/.ssh/config line 16: Applying options for *
debug1: Reading configuration data /etc/ssh/ssh_config
debug1: /etc/ssh/ssh_config line 20: Applying options for *
debug1: Connecting to 172.16.1.100 [172.16.1.100] port 22.
debug1: Connection established.
debug1: identity file /Users/testuser/.ssh/id_rsa type 1
debug1: key_load_public: No such file or directory
debug1: identity file /Users/testuser/.ssh/id_rsa-cert type -1
debug1: identity file /Users/testuser/.ssh/id_dsa type 2
debug1: key_load_public: No such file or directory
debug1: identity file /Users/testuser/.ssh/id_dsa-cert type -1
debug1: key_load_public: No such file or directory
debug1: identity file /Users/testuser/.ssh/id_ecdsa type -1
debug1: key_load_public: No such file or directory
debug1: identity file /Users/testuser/.ssh/id_ecdsa-cert type -1
debug1: key_load_public: No such file or directory
debug1: identity file /Users/testuser/.ssh/id_ed25519 type -1
debug1: key_load_public: No such file or directory
debug1: identity file /Users/testuser/.ssh/id_ed25519-cert type -1
debug1: Enabling compatibility mode for protocol 2.0
debug1: Local version string SSH-2.0-OpenSSH_7.3
debug1: Remote protocol version 2.0, remote software version OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8
debug1: match: OpenSSH_6.6.1p1 Ubuntu-2ubuntu2.8 pat OpenSSH_6.6.1* compat 0x04000000
debug1: Authenticating to 172.16.1.100:22 as 'testuser'
debug1: SSH2_MSG_KEXINIT sent
debug1: SSH2_MSG_KEXINIT received
debug1: kex: algorithm: curve25519-sha256@libssh.org
debug1: kex: host key algorithm: ssh-dss
debug1: kex: server->client cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression:
zlib@openssh.com
debug1: kex: client->server cipher: chacha20-poly1305@openssh.com MAC: <implicit> compression:
zlib@openssh.com
debug1: expecting SSH2_MSG_KEX_ECDH_REPLY
debug1: Server host key: ssh-dss SHA256:c+YpkZsQyUwi3tkIVJFXHAstwlkdewO1G0s7P2khV7U
debug1: Host '172.16.1.100' is known and matches the DSA host key.
debug1: Found key in /Users/testuser/.ssh/known_hosts:5
debug1: rekey after 134217728 blocks
debug1: SSH2_MSG_NEWKEYS sent
debug1: expecting SSH2_MSG_NEWKEYS
debug1: rekey after 134217728 blocks
debug1: SSH2_MSG_NEWKEYS received
```

```
debug1: Skipping ssh-dss key /Users/testuser/.ssh/id_dsa - not in PubkeyAcceptedKeyTypes
debug1: SSH2_MSG_SERVICE_ACCEPT received
debug1: Authentications that can continue: publickey,password
debug1: Next authentication method: publickey
debug1: Offering RSA public key: /Users/testuser/.ssh/id_rsa
debug1: Authentications that can continue: publickey,password
debug1: Trying private key: /Users/testuser/.ssh/id_ecdsa
debug1: Trying private key: /Users/testuser/.ssh/id_ed25519
debug1: Next authentication method: password
testuser@172.16.1.100's password: <<< ENTER USER PASSWORD TO LOG-IN >>>
debug1: Enabling compression at level 6.
debug1: Authentication succeeded (password).
Authenticated to 172.16.1.100 ([172.16.1.100]:22).
debug1: channel 0: new [client-session]
debug1: Requesting no-more-sessions@openssh.com
debug1: Entering interactive session.
debug1: pledge: exec
debug1: No xauth program.
Warning: untrusted X11 forwarding setup failed: xauth key data not generated
debug1: Requesting authentication agent forwarding.
debug1: Sending environment.
debug1: Sending env LANG = en_US.UTF-8
debug1: Sending env LC_CTYPE = en_US.UTF-8
```