メールの SCP プッシュを設定することは ESA を口グオンします

目次

概要

この資料に方法を記述されていますメールのセキュア コピー プッシュ（SCP）を外部 の syslog サーバに Cisco E メール セキュリティ アプライアンス（ESA）から（または他の口グ型）記録しますセットアップおよび設定するために。

背景説明

口グが SCP を使用して押すことができないまたはキー ミスマッチを示す工ラー口グがあるかもしれませんことを示している管理者は工ラー通知を受信するかもしれません。

前提条件

syslog サーバ ESA が SCP 口グファイルにこと:

1. 使用されるべきディレクトリが利用できること保証して下さい。
2. AuthorizedKeysFile 設定のための「/etc/ssh/sshd_config」を検討して下さい。 これは SSH を .ssh/authorized_keys ファイルに書かれる key_name 刺し傷のためのユーザのホーム ディレクトリの authorized_keys および外観を受け入れるように告げます:
   ```
   AuthorizedKeysFile      %h/.ssh/authorized_keys
   ```
3. 使用されるディレクトリの権限を確認して下さい。 権限変更を行なう必要がある場合もあります:「$HOME」の権限は 755 に設定 されます。「$HOME/.ssh」の権限は 755 に設定されます。「$HOME/.ssh/authorized_keys」の権限は 600 に設定 されます。

UNIX/Linux のファイル レベル制限および権限

アクセス制限には 3 つの型があります:

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

またユーザ制限には 3 つの型があります:

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

フォルダ/ディレクトリ許可:

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

数字表示法:

Linux 権限を表すためのもう一つの方式はによって示されているように 8 表示法`- c %a.`です この表示法は少なくとも 3 ディジットで構成されています。 3 つの右端のディジットのそれぞれは権限の異なるコンポーネントを表します: オーナー、グループ、および他。

これらのディジットのそれぞれは 2 進数システムのコンポーネント ビットの合計です:

```
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
```

ステップ #3 に関しては、755 に $HOME ディレクトリを設定 する 推奨事項は次のとおりです: 7=rwx `5=r-x 5=r-x`

これはディレクトリにデフォルト 許可が`- rwxr-xr-x` あることを意味します (0755) として 8 表示法で表される。

メールの SCP プッシュを設定することは ESA をログオンします

1. CLI コマンド **logconfig を**実行して下さい。
2. **新しい**オプションを選択して下さい。
3. このサブスクリプションに対するログファイル型を選択して下さい、これは IronPort テキスト メール ログのため選択の "1" の、または他のどのログファイル型もです。
4. ログファイルの名前を入力して下さい。
5. 水平な適切なログを選択して下さい。 通常選択の水平な Informational に "3"、または他のどのログも選択する必要があります。
6. ログ」を取得するために「プロンプト表示された場合方式を選択しなさい **SCP プッシュ**に "3" を選択して下さい。
7. ログをに渡すために IP アドレスか DNS ホスト名で入力して下さい。
8. リモートホストでに接続するためにポートを入力して下さい。
9. ログを置くためにリモートホストのディレクトリを入力して下さい。
10. ログファイルのために使用するためにファイル名で入力して下さい。
11. 、もし必要なら、*$ ホスト名*のようなシステムによって基づく固有の識別番号を、ログにファイル名を追加 する *$serialnumber* 設定して下さい。
12. 最大を filesize 転送する前に設定 して下さい。

13. ログファイルの時間ベース ロールオーバーを、該当する場合設定して下さい。
14. 「頼まれたときホストキー チェックを有効に したいと思うためにか。」、「Y を」入力して下さい。
15. ログファイルが」。アップロードされるように authorized_keys ファイルにそれから「置きます次の SSH キーを示されます
16. Syslog サーバに「authorized_keys」ファイルに SSH キーを置く必要があるので、そのキーをコピーして下さい。 logconfig から Syslog サーバの $HOME/.ssh/authorized_keys ファイルに与えられるキーを貼り付けて下さい。
17. ESA から、CLI コマンドをコンフィギュレーション変更を保存し、保存するために**託します**実行して下さい。

ログの設定はまた GUI から堪能である場合もあります: **システム 管理 > ログ サブスクリプション**

> **注:** 完全な詳細およびより詳しい 情報のための ESA ユーザガイドの口ギング章を検討して下さい。

確認

Hostkeyconfig

コマンド **logconfig > hostkeyconfig を**実行して下さい。 「ssh dss として」設定の間に提供されるキーへの短縮されたキー類似したでリストされている syslog サーバについては設定されるエントリを見るはずです。

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

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

システムログ

システムログは次を記録します: 情報を、仮想 な アプライアンス ライセンスの有効期限アラート、DNS ステータス情報起動し、ユーザ commit コマンドを使用してタイプされるコメントします。 システムログはアプライアンスの基本的な状態のトラブルシューティングを実行するために役立ちます。

CLI からコマンド**末尾 system_logs を**実行することはシステム状態にライブ外観を提供します。

また CLI コマンド **rollovernow を**選択し、ログファイルに関連付けられる番号を選択することができます。 system_logs の syslog サーバにこれがログファイル SCP 表示されます:

```
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
```

高度なトラブルシューティング

ローカル ホストからの syslog サーバへの接続においての継続的だった問題が、および使用 ssh あったら、冗長 モードのユーザアクセスをテストするために「ssh testuser@hostname - v」を実行して下さい。 これは補佐官トラブルシューティング ssh 接続がどこに成功していないか示すかもしれません。

```
$ 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
```