Configurazione del push SCP dei log di posta su ESA

Sommario

Introduzione

In questo documento viene descritto come configurare un Secure Copy Push (SCP) di log di posta (o altri tipi di log) da Cisco Email Security Appliance (ESA) a un server syslog esterno.

Premesse

Èpossibile che un amministratore riceva notifiche di errore in cui viene indicato che non è possibile eseguire il push dei registri tramite SCP oppure che nei registri di errore sia indicata una mancata corrispondenza delle chiavi.

Prerequisiti

Sul server syslog in cui l'ESA invia i file di registro SCP:

1. Assicurarsi che la directory da utilizzare sia disponibile.
2. Esaminare '/etc/ssh/sshd_config' per le impostazioni di AuthorizedKeysFile. In questo modo SSH accetta authorized_keys e cerca la stringa nome_chiave scritta nel file .ssh/authorized_keys nella home directory dell'utente:
   ```
   AuthorizedKeysFile      %h/.ssh/authorized_keys
   ```
3. Verificare le autorizzazioni della directory da utilizzare. Potrebbe essere necessario apportare modifiche alle autorizzazioni: Le autorizzazioni in '$HOME' sono impostate su 755.Le autorizzazioni per '$HOME/.ssh' sono impostate su 755.Le autorizzazioni per '$HOME/.ssh/authorized_keys' sono impostate su 600.

Limitazioni e autorizzazioni a livello di file su UNIX/Linux

Esistono tre tipi di restrizioni di accesso:

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

Esistono inoltre tre tipi di restrizioni per gli utenti:

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

Autorizzazioni cartella/directory:

```
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
```

Notazione numerica:

Un altro metodo per rappresentare le autorizzazioni Linux è la notazione ottale, come mostrato nella `stat -c %a`. Questa notazione è costituita da almeno tre cifre. Ognuna delle tre cifre a destra rappresenta un componente diverso delle autorizzazioni: proprietario, gruppo e altri.

Ciascuna di queste cifre è la somma dei bit che la compongono nel sistema numerico binario:

```
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
```

Per il passo 3, si consiglia di impostare la directory $HOME su 755 come indicato di seguito: 7_{rwx} 5_{r-x} 5_{r-x}

Ciò significa che la directory dispone delle autorizzazioni predefinite `-rwxr-xr-x` (rappresentato in notazione ottale come 0755).

Configurazione del push SCP dei log di posta su ESA

1. Eseguire il comando **logconfig** di CLI.
2. Selezionare l'opzione **new**.
3. Scegliere il tipo di file di log per questa sottoscrizione. Il valore sarà "1" per i log di IronPort Text Mail o qualsiasi altro tipo di file di log desiderato.
4. Immettere il nome del file di log.
5. Selezionare il livello di log appropriato. In genere, è necessario selezionare "3" per Informativo o qualsiasi altro livello di log desiderato.
6. Quando richiesto, 'Choose the method to retrieve the logs' (Scegli il metodo per recuperare i log), selezionare "3" per **SCP Push**.
7. Immettere l'indirizzo IP o il nome host DNS a cui recapitare i registri.
8. Immettere la porta a cui connettersi sull'host remoto.
9. Immettere la directory sull'host remoto in cui inserire i log.
10. Immettere un nome file da utilizzare per i file di log.
11. Se necessario, configurare gli identificatori univoci basati sul sistema, ad esempio *$hostname*, *$serialnumber* da aggiungere al nome file di log.

12. Impostare Dimensioni massime file prima del trasferimento.
13. Configurare il rollover basato sul tempo dei file di log, se applicabile.
14. Quando viene chiesto se si desidera attivare il controllo dei tasti host, immettere "Y".
15. Viene quindi visualizzato il messaggio "Inserire le seguenti chiavi SSH nel file authorized_keys per consentire il caricamento dei file di log."
16. Copiare la chiave, in quanto sarà necessario inserire la chiave SSH nel file 'authorized_keys' sul server Syslog. Incollare la chiave fornita da logconfig nel file $HOME/.ssh/authorized_keys sul server Syslog.
17. Dall'ESA, eseguire il comando **commit** della CLI per salvare ed eseguire il commit delle modifiche della configurazione.

La configurazione del registro può essere effettuata anche dalla GUI: **Amministrazione sistema > Registra sottoscrizioni**

> **Nota:** Consultare il capitolo Logging della Guida dell'utente ESA per i dettagli completi e ulteriori informazioni.

Conferma

Hostkeyconfig

Eseguire il comando **logconfig > hostkeyconfig**. una voce per il server syslog configurato dovrebbe essere elencata come "ssh-dss" con una chiave abbreviata simile alla chiave fornita durante la configurazione.

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

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

Log di sistema

I registri di sistema registrano quanto segue: informazioni di avvio, avvisi sulla scadenza delle licenze dei dispositivi virtuali, informazioni sullo stato DNS e commenti immessi dagli utenti tramite il comando commit. I registri di sistema sono utili per la risoluzione dei problemi relativi allo stato di base dell'accessorio.

L'esecuzione del comando **tail system_logs** dalla CLI consente di verificare in tempo reale lo stato del sistema.

Èpossibile anche scegliere il comando **rollover** della CLI e selezionare il numero associato al file di log. Il file di log SCP per il server syslog è visualizzato 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
```

Risoluzione dei problemi avanzata

Se i problemi di connettività al server syslog continuano, dall'host locale e utilizzando ssh, eseguire "ssh testuser@hostname -v" per verificare l'accesso utente in modalità dettagliata. Ciò può aiutare a risolvere il problema e a mostrare dove la connessione ssh non riesce.

```
$ 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
```