ESA에서 SCP 메일 로그 푸시 구성

목차

소개

이 문서에서는 Cisco ESA(Email Security Appliance)에서 외부 syslog 서버로 메일 로그(또는 기타 로그 유형)의 SCP(Secure Copy Push)를 설정하고 구성하는 방법에 대해 설명합니다.

배경 정보

관리자는 SCP를 사용하여 로그를 푸시할 수 없다는 오류 알림을 받거나 키 불일치를 나타내는 오류 로그가 있을 수 있습니다.

사전 요구 사항

ESA에서 SCP 로그 파일을 SCP로 로깅하는 syslog 서버에서 다음을 수행합니다.

1. 사용할 디렉토리를 사용할 수 있는지 확인합니다.
2. AuthorizedKeysFile 설정에 대한 '/etc/ssh/sshd_config'를 검토합니다. 이렇게 하면 SSH가 authorized_keys를 수락하고 .ssh/authorized_keys 파일에 기록된 key_name 문자열을 사용자의 홈 디렉토리에서 찾을 수 있습니다.
   ```
   AuthorizedKeysFile      %h/.ssh/authorized_keys
   ```
3. 사용할 디렉토리의 권한을 확인합니다. 사용 권한을 변경해야 할 수 있습니다. '$HOME'에 대한 사용 권한이 755로 설정되었습니다.'$HOME/.ssh'에 대한 권한이 755로 설정되어 있습니다 .'$HOME/.ssh/authorized_keys'에 대한 권한이 600으로 설정되어 있습니다.

UNIX/Linux의 파일 레벨 제한 사항 및 권한

액세스 제한 유형에는 세 가지가 있습니다.

```
Permission Action chmod option ==================================== read (view) r or 4 write (edit) w or 2 execute (execute) x or 1
```
또한 세 가지 유형의 사용자 제한이 있습니다.

```
User ls output ================= owner -rwx------ group ----rwx--- other -------rwx
```
폴더/디렉터리 권한:

```
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
```
숫자 표기법:

Linux 권한을 나타내는 다른 방법은 다음과 같이 8진수 표기법입니다. `stat -c %a`.이 표기법은 3자리 이상으로 구성됩니다.가장 오른쪽 세 자리 각각은 사용 권한의 다른 구성 요소를 나타냅니다.소유자, 그룹 및 기타

이러한 각 자릿수는 이진 숫자 시스템에서 해당 구성 요소 비트의 합계입니다.

```
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
```
#3단계의 경우 $HOME 디렉토리를 755로 설정하는 것이 좋습니다.7=`rwx` 5=`r-x` 5=`r-x`

이는 디렉터리에 기본 권한이 있음을 의미합니다. `-rwxr-xr-x` (8진수 표기법으로 0755로 표시됨)

ESA에서 SCP 메일 로그 푸시 구성

1. CLI 명령 logconfig를 **실행합니다**.
2. **new** 옵션을 **선택합니다**.
3. 이 구독의 로그 파일 유형을 선택합니다. 이 유형은 IronPort 텍스트 메일 로그에 대해 "1" 또는 선택한 다른 로그 파일 유형입니다.
4. 로그 파일의 이름을 입력합니다.
5. 적절한 로그 레벨을 선택합니다. 일반적으로 정보 또는 선택한 다른 로그 레벨에 대해 "3"을 선택해야 합니다.
6. '로그를 검색할 방법을 선택하십시오.'라는 메시지가 나타나면 **SCP Push**에 대해 "3"을 **선택합니다**.
7. 로그를 전달할 IP 주소 또는 DNS 호스트 이름을 입력합니다.
8. 원격 호스트에서 연결할 포트를 입력합니다.
9. 로그를 배치할 원격 호스트의 디렉토리를 입력합니다.
10. 로그 파일에 사용할 파일 이름을 입력합니다.
11. 필요한 경우 시스템 기반 고유 식별자(예: *$hostname*, *$serialnumber)를 구성하여* 로그 파일 이름에 추가합니다.
12. 전송하기 전에 최대 파일 크기를 설정합니다.
13. 해당되는 경우 로그 파일의 시간 기반 롤오버를 구성합니다.
14. "Do you want to enable host key checking(호스트 키 검사를 활성화하시겠습니까?)"이 표시되면 "Y"를 입력합니다.

15. 그런 다음 "다음 SSH 키를 authorized_keys 파일에 넣어 로그 파일을 업로드할 수 있습니다."라는 메시지가 표시됩니다.
16. Syslog 서버의 'authorized_keys' 파일에 SSH 키를 넣어야 하므로 해당 키를 복사합니다. logconfig에서 제공된 키를 Syslog 서버의 $HOME/.ssh/authorized_keys 파일에 붙여넣습니다.
17. ESA에서 CLI 명령 **커밋**을 실행하여 컨피그레이션 변경 사항을 저장하고 커밋합니다.

GUI에서 로그 컨피그레이션을 수행할 수도 있습니다.**시스템 관리 > 로그 서브스크립션**

참고:자세한 내용 및 자세한 내용은 ESA 사용 설명서의 로깅 장을 참조하십시오.

확인

Hostkeyconfig

logconfig > hostkeyconfig 명령**을 실행합니다**. 컨피그레이션 중에 제공된 키와 유사한 축약된 키를 사용하여 "ssh-dss"로 구성된 syslog 서버에 대한 항목을 확인해야 합니다.

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

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

시스템 로그

시스템 로그는 다음을 기록합니다.부트 정보, 가상 어플라이언스 라이센스 만료 알림, DNS 상태 정보, commit 명령을 사용하여 사용자가 입력한 주석시스템 로그는 어플라이언스의 기본 상태를 트러블슈팅하는 데 유용합니다.

CLI에서 **tail system_logs** 명령을 실행하면 시스템 상태를 확인할 수 있습니다.

CLI 명령 롤오버를 선택하고 로그 파일**과** 연결된 번호를 선택할 수도 있습니다. system_logs에서 syslog 서버에 대한 로그 파일 SCP가 표시됩니다.

```
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
```

고급 문제 해결

syslog 서버, 로컬 호스트에서 및 ssh를 사용하는 경우 계속 문제가 발생하면 "ssh testuser@hostname -v"를 실행하여 자세한 정보 표시 모드에서 사용자 액세스를 테스트합니다. 이 경우 ssh 연결이 성공하지 않는 위치를 표시하기 위해 트러블슈팅을 수행할 수 있습니다.

```
$ 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
```