تكوين دفع SCP لسجلات البريد على ESA

المحتويات

المقدمة
معلومات أساسية
_
المتطلبات الأساسية
القيود والأذونات على مستوى الملفات على UNIX/Linux
تكوين دفع SCP لسجلات البريد على ESA
تأكيد
Hostkeyconfig
سجلات النظام
أستكشاف الأخطاء وإصلاحها المتقدم

المقدمة

يصف هذا وثيقة كيف أن setup وشكلت يأمن نسخة دفع (SCP) من بريد سجل مقياس سرعة (أو آخر سجل نوع من cisco بريد إلكتروني أمن تطبيق (ESA) إلى خارجي syslog نادل.

معلومات أساسية

قد يتلقى المسؤول إعلامات الخطأ التي تشير إلى أنه لا يمكن دفع السجلات باستخدام SCP، أو قد يكون هناك سجلات خطأ تذكر المفتاح (المفاتيح) غير متطابق.

المتطلبات الأساسية

على خادم syslog الذي سيقوم ESA بتسجيل ملفات SCP إليه:

1. تأكد من توفر الدليل المطلوب إستخدامه.
2. مراجعة '/etc/ssh/sshd_config' لإعدادات AuthorizedKeysFile. وهذا يوجب على SSH قبول authorized_keys والبحث في الدليل الرئيسي للمستخدم عن علامة إختيار المفتاح_name المكتوبة في ملف ssh/authorized_keys:
```
AuthorizedKeysFile      %h/.ssh/authorized_keys
```
3. تحقق من أذونات الدليل المراد إستخدامه. قد تحتاج إلى إجراء تغييرات على الأذونات: تم تعيين الأذونات على '$HOME' إلى 755.تم تعيين الأذونات على '$HOME/.ssh' إلى 755.تم تعيين الأذونات على '$HOME/.ssh/authorized_keys' إلى 600.

القيود والأذونات على مستوى الملفات على UNIX/Linux

هناك ثلاثة أنواع من قيود الوصول:

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

هناك أيضا ثلاثة أنواع من قيود المستخدم:

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

أذونات المجلد/الدليل:

```
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
```

تدوين رقمي:

هناك طريقة أخرى لتمثيل أذون لينوكس وهي التدوين الثماني كما هو موضح ب `stat -c %a`. يتكون هذا التدوين من ثلاثة أرقام على الأقل. تمثل كل من الأرقام الثلاثة الموجودة في أقصى اليمين مكونا مختلفا للأذونات: المالك، والمجموعة، وغيرهم.

كل من هذه الأرقام هي مجموع وحدات بت المكونة لها في نظام الأرقام الثنائية:

```
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
```

بالنسبة للخطوة رقم 3، ستكون التوصية بتعيين دليل المنزل بقيمة 755 دولار: `rwx` 5=`r-x` 5=`r-x`=7

هذا يعني أن الدليل لديه الأذونات الافتراضية `rwxr-xr-x-` (ممثلة في التدوين الثماني على هيئة 0755).

تكوين دفع SCP لسجلات البريد على ESA

1. قم بتشغيل أمر **logconfig** CLI.
2. حدد الخيار **جديد**.
3. أختر نوع ملف السجل لهذا الاشتراك، سيكون هذا "1" لسجلات بريد IronPort النصي، أو أي نوع ملف سجل آخر من إختيارك.
4. أدخل اسم ملف التدوين.
5. حدد مستوى السجل المناسب. ستحتاج في العادة إلى تحديد "3" للحصول على المعلومات أو أي مستوى آخر من السجل تختاره.
6. عند مطالبتك باختيار "طريقة إسترداد السجلات"، حدد "3" **للدفعة عبر بروتوكول SCP**.
7. أدخل في عنوان IP أو اسم مضيف DNS لتسليم السجلات إلى.
8. أدخل المنفذ الذي تريد الاتصال به على المضيف البعيد.
9. أدخل الدليل على المضيف البعيد لوضع السجلات.
10. أدخل اسم ملف لاستخدامه لملفات السجل.
11. قم بتكوين المعرفات الفريدة المستندة إلى النظام، إذا لزم الأمر، مثل *$serialnumber$*، *hostname* لإلحاق اسم ملف السجل.
12. تعيين الحد الأقصى للتصفية قبل النقل.
13. قم بتكوين التمرير القائم على الوقت لملفات السجل، إذا كان ذلك ممكنا.
14. عند السؤال "هل تريد تمكين التحقق من مفتاح المضيف؟ "، أدخل "Y".
15. ثم يتم عرض الأمر "يرجى وضع مفتاح (مفاتيح) SSH التالية في ملف authorized_keys الخاص بك حتى يمكن تحميل ملفات السجل."

16. انسخ هذا المفتاح، حيث ستحتاج إلى وضع مفتاح SSH في ملف 'authorized_keys' على خادم syslog.
الصق المفتاح المعطى من logConfig إلى $HOME/.ssh/authorized_keys مبرد على خادم syslog.
17. من ال ESA، ركضت ال CLI أمر **يلتزم** أن يحفظ وينفذ تشكيل تغير.
يمكن أيضا إنجاز تكوين السجل من واجهة المستخدم الرسومية: **إدارة النظام > اشتراكات السجل**

ملاحظة: يرجى مراجعة الفصل الخاص بالتسجيل في دليل إستخدام الإيسا للحصول على تفاصيل كاملة ومعلومات إضافية.

تأكيد

Hostkeyconfig

قم بتشغيل الأمر **logconfig > hostkeyconfig**. يجب أن ترى إدخالا لخادم syslog الذي تم تكوينه مسرودا ك "ssh-dss" بمفتاح مختصر مشابه للمفتاح الذي تم توفيره أثناء التكوين.

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

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

سجلات النظام

تقوم سجلات النظام بتسجيل ما يلي: معلومات التمهيد وتنبيهات انتهاء صلاحية ترخيص الجهاز الظاهري ومعلومات حالة DNS والتعليقات التي تم كتابتها باستخدام أمر الالتزام. تعد سجلات النظام مفيدة لاستكشاف أخطاء الحالة الأساسية للجهاز وإصلاحها.

سيؤدي تشغيل الأمر **tail system_log** من واجهة سطر الأوامر إلى توفير نظرة مباشرة على حالة النظام.

يمكنك أيضا إختيار أمر **rollovernow** CLI وتحديد الرقم المرتبط بملف السجل. سترى هذا ال log file SCP إلى ك syslog نادل في system_log:

```
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
```

أستكشاف الأخطاء وإصلاحها المتقدم

إذا إستمرت المشاكل مع الاتصال بخادم syslog، من المضيف المحلي واستخدام SSH، فقم بتشغيل "ssh testuser@hostname -v" لاختبار وصول المستخدم في وضع السرعة. قد يساعد هذا في أستكشاف الأخطاء وإصلاحها لإظهار مكان عدم نجاح اتصال SSH.

```
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
```

حول هذه الترجمة
ترجمت Cisco هذا المستند باستخدام مجموعة من التقنيات الآلية والبشرية لتقديم محتوى دعم للمستخدمين في جميع أنحاء العالم بلغتهم الخاصة. يُرجى ملاحظة أن أفضل ترجمة آلية لن تكون دقيقة كما هو الحال مع الترجمة الاحترافية التي يقدمها مترجم محترف. تخلي Cisco Systems مسئوليتها عن دقة هذه الترجمات وتُوصي بالرجوع دائمًا إلى المستند الإنجليزي الأصلي (الرابط متوفر).