如何在 CentOS 7 上使用雙因素身份驗證來保護 SSH

介紹

認證因素是證明你有權登入到系統的一條訊息，SSH 預設使用密碼認證，這只是一個因素，如果您的密碼被擷取或被破壞，那麼沒有任何阻止壞角色擁有你的系統。這就是我們所說的“**單點失敗**”。在本教程中，我們將使用 Google-Authenticator 移動應用程式設定二次驗證機制，每次登錄系統時都會提供一次性密碼（OTP）。一般可應用於跳板機或較嚴謹的服務主機上。

安裝 Google 身份驗證器

Google-Authenticator 應用程序可在所有手機上使用，您可以從 Google Play 下載 Android 應用程序，從 App Store 下載 IOS 應用程序（iPhone 用戶）。

安裝 Google 的 PAM

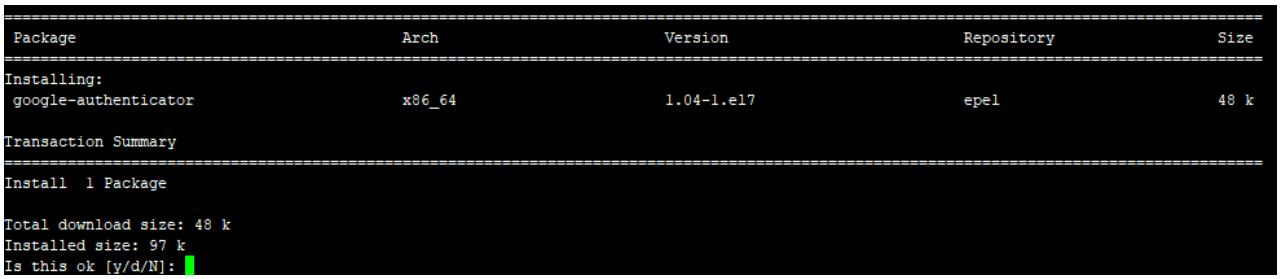
PAM（Pluggable Authentication Module，可插拔認證模塊）是基於 Linux 系統認證用戶的認證基礎設施。以一般 user 登入系統，這裡以 darwin43 身份；記得將 darwin43 帳號加入 sudo 群組

首先使用以下命令安裝 EPEL Repo 儲存庫：

```
sudo yum install epel-release
```

現在安裝 Google 的 PAM：

```
sudo yum install google-authenticator
```



配置 Google 的 PAM

安裝過程完成後，你可以執行腳本幫助要增加第二個因子的用戶產生的密鑰，該密鑰在基於用戶的系統上而不是系統範圍內生成，這意味著每個用戶想要使用 OTP 身份驗證將需要登錄並執行生成器腳本來獲得自己的密鑰。

執行初始化腳本：

```
google-authenticator
```

執行該命令後，系統會詢問你幾個問題。第一個問是否認證 Token 應該是基於時間的。建議用“Y”來回答。之後，產生一個超大的 QR Code 將出現在你的終端上，你必須使用手機進行掃描，以便個人資料自動增加到你的 Google-Authenticator 應用程序。而且還要記下“**密鑰**”，“**驗證碼**”和“**緊急防盜碼**”。如果你丟失手機或意外從中刪除應用程序，你將能夠登入到你的服務器。

Do you want authentication tokens to be time-based (y/n) **y**

Warning: pasting the following URL into your browser exposes the OTP secret to Google:

https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/root@elasticstack%3Fsecret%3DK5LSCYAFN3VIIWPQAR53FOYQQI%26issuer%3Delasticstack

Your new secret key is: YBJRADV63BVA2JMAHIUWR6SVXU

Your verification code is 894765

Your emergency scratch codes are:

39113650

64113553

33066213

44445047

98704222

Do you want me to update your "/root/.google_authenticator" file? (y/n) **y**

Do you want to disallow multiple uses of the same authentication

token? This restricts you to one login about every 30s, but it increases

your chances to notice or even prevent man-in-the-middle attacks (y/n) **y**

By default, a new token is generated every 30 seconds by the mobile app.

In order to compensate for possible time-skew between the client and the server,

we allow an extra token before and after the current time. This allows for a

time skew of up to 30 seconds between authentication server and client. If you

experience problems with poor time synchronization, you can increase the window

from its default size of 3 permitted codes (one previous code, the current

code, the next code) to 17 permitted codes (the 8 previous codes, the current

code, and the 8 next codes). This will permit for a time skew of up to 4 minutes

between client and server.

Do you want to do so? (y/n) **y**

If the computer that you are logging into isn't hardened against brute-force

login attempts, you can enable rate-limiting for the authentication module.

By default, this limits attackers to no more than 3 login attempts every 30s.

Do you want to enable rate-limiting? (y/n) **y**

[root@elasticstack ~]#

配置 SSH

在回答所有問題後，你的 Google PAM 已準備就緒。現在我們只需要為我們的 SSH 做一些配置。使用以下命令打開 SSH 配置檔案：

```
sudo  vi  /etc/pam.d/sshd
```

增加如下：

```
auth  required  pam_google_authenticator.so  nullok
```

```
[root@elasticstack ~]# sudo vi /etc/pam.d/sshd

#%PAM-1.0
auth       required     pam_sepermit.so
auth       substack     password-auth
auth       include      postlogin
auth       required     pam_google_authenticator.so nullok
# Used with polkit to reauthorize users in remote sessions
-auth      optional     pam_reauthorize.so prepare
account    required     pam_nologin.so
#%PAM-1.0
auth       required     pam_sepermit.so
auth       substack     password-auth
auth       include      postlogin
auth       required     pam_google_authenticator.so nullok
# Used with polkit to reauthorize users in remote sessions
-auth      optional     pam_reauthorize.so prepare
account    required     pam_nologin.so
account    include      password-auth
password   include      password-auth
# pam_selinux.so close should be the first session rule
session    required     pam_selinux.so close
session    required     pam_loginuid.so
# pam_selinux.so open should only be followed by sessions to be executed in the user context
session    required     pam_selinux.so open env_params
session    required     pam_namespace.so
session    optional     pam_keyinit.so force revoke
session    include      password-auth
session    include      postlogin

# Used with polkit to reauthorize users in remote sessions
-session   optional     pam_reauthorize.so prepare
```

存檔退出。

配置 SSH 來支持這種認證，使用下面的命令打開“sshd_config”檔案：

```
sudo  vi  /etc/ssh/sshd_config
```

找到 ChallengeResponseAuthentication 的行並將其值設置為“yes”與 AuthenticationMethods 設置為“keyboard-interactive”

```
ChallengeResponseAuthentication  yes

AuthenticationMethods  keyboard-interactive
```

```
# To disable tunneled clear text passwords, change to no here!
#PasswordAuthentication yes
#PermitEmptyPasswords no
PasswordAuthentication yes

# Change to no to disable s/key passwords
ChallengeResponseAuthentication yes
AuthenticationMethods keyboard-interactive
```

存檔退出。

重新啟動你的 SSH 服務

```
sudo systemctl restart sshd
```

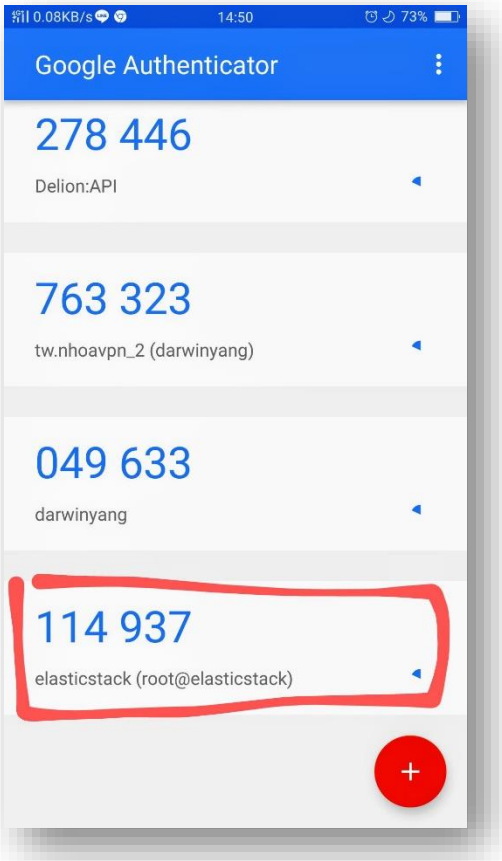
從現在開始，你將被要求提供一個“驗證碼”，你必須從你的 Google-Authenticator 應用程序中得到你的手機。 還記得紅色標示的那段 QR Code？

打開瀏覽器輸入

https://www.google.com/chart?chs=200x200&chld=M|0&cht=qr&chl=otpauth://totp/root@elasticstack%3Fsecret%3DK5LSCYAFN3VIIWPQAR53FOYQQI%26issuer%3Delasticstack

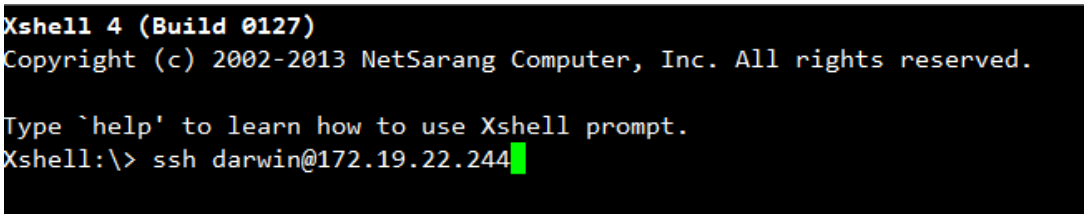
用手機將剛下載的 Google Authenticatort 掃瞄條碼一下即可加入私鑰

下圖為日後登入系統用的動態驗證碼

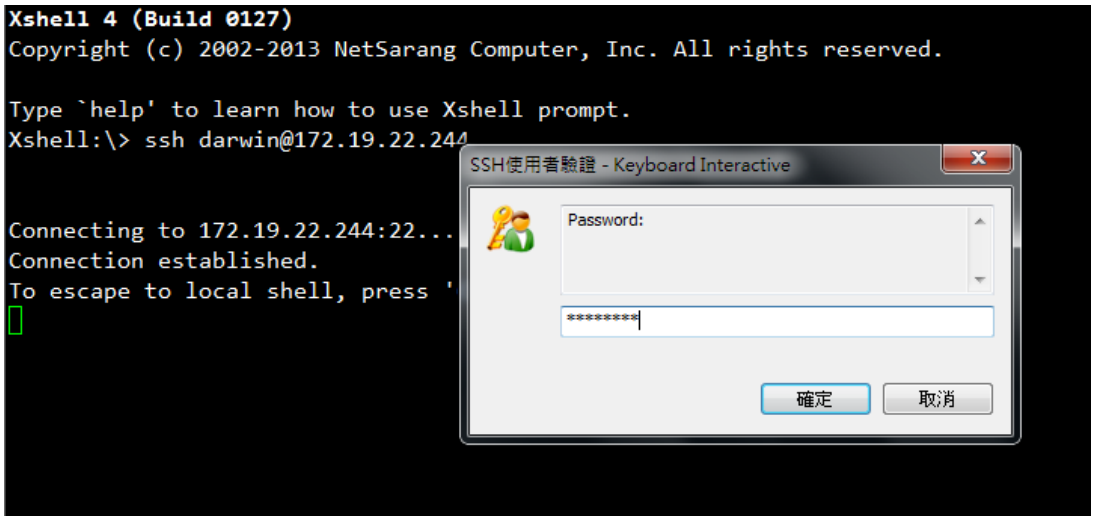


登入 SSH

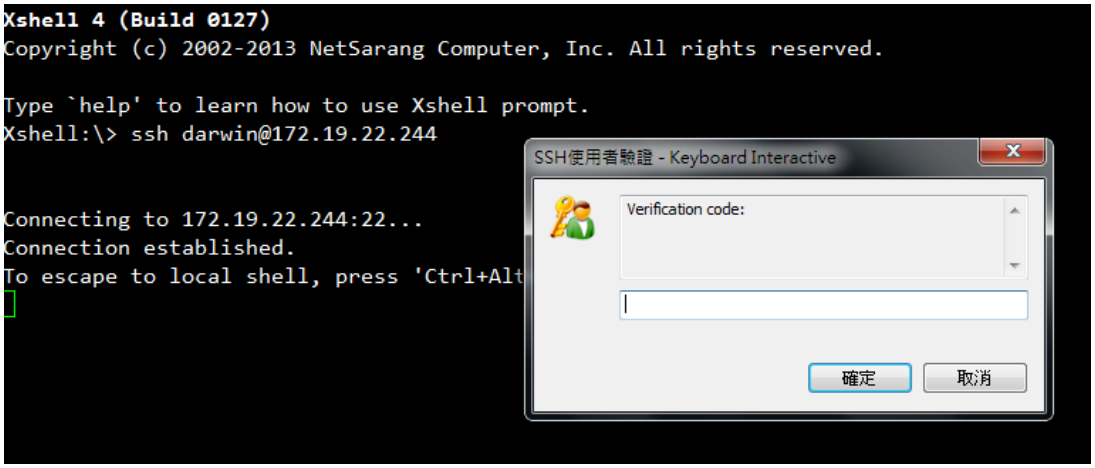
登入方式採用 SSH 帳密+F2A 方式, 如下所示



第一道關卡請輸入該 darwin43 帳號所屬的密碼



第二道關卡則配拿手機上的 F2A 驗證碼做登入, 每 30 秒變更一次



如果都正確則正常進入系統~ 大功告成!