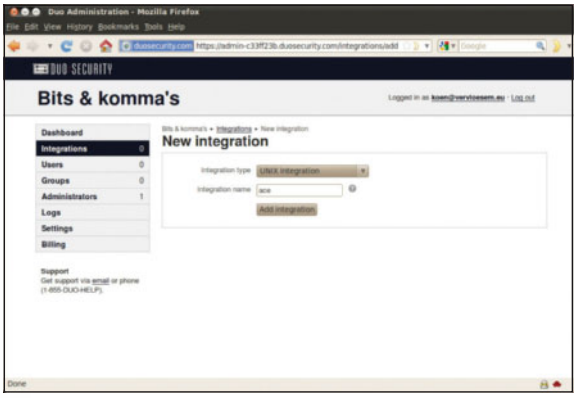
Duo Security:

If the only thing protecting your server is a password, you're taking a big risk. **Koen Vervloesem** shows you how to add a second layer of protection.



To be able to use Duo's two-factor authentication, you have to add a so-called Unix Integration to your account.

Our expert

Koen Vervloesem has been running free software since 2000 and he likes the countless ways to secure it.

Passwords are all well and good, but they're often all too easy to crack, and if there's a keylogger on your computer, an attacker can log in to your server.

Two-factor authentication combines 'something you have' with 'something you know' (the password). Then, even if an attacker has your password, he has to take a second hurdle before the server lets him in. This second layer of defence (something you have) can be a specialised hardware token or something like your smartphone.

One interesting two-factor solution is Duo Security (**http://www.duosecurity.com/**), which supports authentication for VPNs (Virtual Private Networks), websites, Unix accounts (SSH), and so on. It's an easy way to protect your systems with strong authentication, using a password as one component and a phone call, SMS code or smartphone application as the second.

If someone's stolen your password, they can't log into your system because they also need the second component. Duo Security is a free service for up to 10 users and in this tutorial we'll show you how to secure SSH access to your server with Duo Security's two-factor authentication.

Sign up for an account

Duo Security is a hosted solution: you install a client on the server you want to secure, and this client communicates with Duo Security's authentication server for the out-of-band authentication. So you first have to create an account on Duo's website. There's a Personal plan for up to 10 users, which is free and offers most of the functionality of the product, but no support. Just click on Sign Up and enter your name, email address, organisation and expected number of users. Then click on the activation link in the email you get.

To activate your administration account, you just have to fill in your name again, and then your phone number and a password. After submitting the form you'll get an automated phone call to confirm your number. Answer your phone and press #. After activating and logging in, click Integrations in the web interface to set up your first integration.

Click New integration > UNIX Integration for the integration type, fill in a descriptive name for the integration (eg your server's hostname) and click Add integration. The page about this integration now shows you an integration key and secret key, as well as an API hostname. You'll need all this information later. Don't write down the secret key or share it with anyone else, as it's like the password to use this service.

Configure the client

Now's the time to install the *duo_unix* client software on your server. You can download the source from **https://github.com/duosecurity/duo_unix/downloads**, unpack it and do the **configure && make && sudo make install** incantation. There are also pre-built packages for Debian Sid and other distros, and a PPA (Personal Package Archive) for Ubuntu. By the way, we're talking about Linux here, but – as the name makes clear – Duo's client software works on a lot of other Unix-like operating systems too, such as FreeBSD, NetBSD, OpenBSD, Mac OS X, Solaris/Illumos, HP-UX and AIX.

Once you have *duo_unix* installed, edit **/etc/duo/login_duo.conf** as root to add the integration and secret key and API hostname that you can find in the integration page of your account on the Duo Security website:

```
[duo]
; Duo integration key
ikey = INTEGRATION_KEY
; Duo secret key
skey = SECRET_KEY
```

Server safety

```
; Duo API hostname
host = API_HOSTNAME
```

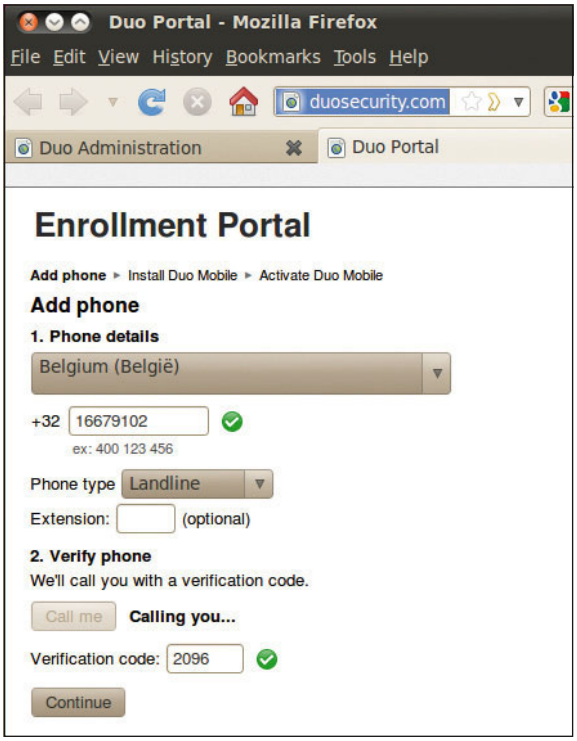
Now that the *login_duo* configuration is set up, try it out as a regular user:

```
$ /usr/local/sbin/login_duo
```

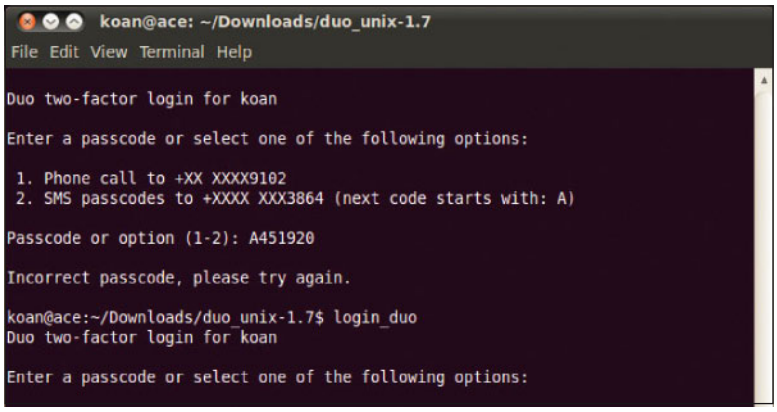
Because this is the first time that you run it, *login_duo* shows you a URL that you should open in your web browser to enroll your phone. On this page, you enter your phone number, phone type (mobile phone or landline) and the way you want to verify your phone. Duo Security can call or text you (the latter, of course, only if you use a mobile phone) with a verification code, after which you have to enter this code in the web form to prove that you are the owner.

The next time you run *login_duo*, it asks you to authenticate with your phone. Just as when you enrolled, you can choose a phone call or text message. When the automated system calls your phone, just answer it and press any key to log in.

If you choose the text message option, Duo Security sends some passcodes to your mobile phone, and every time you log in you have to enter the next passcode on your list. Try it out, and if this works, you're ready to enable *login_duo* to authenticate your users. (If you've installed the *Duo Mobile* smartphone app, there's also a third option, Duo Push, but we'll talk about that later.)



› Add your phone and you'll be called by a computer. Enter the verification code if you are able to understand it.



› Before you configure two-factor authentication for real, you better try it out with the *login_duo* program.

There are a couple of ways to use Duo Security to authenticate your users. For instance, you can force anyone wanting to log into your server via SSH to use Duo Security for it. Just add the following line to **/etc/ssh/sshd_config**:

```
ForceCommand /usr/local/sbin/login_duo
```

You can also limit the use of the Duo login to a subset of users, for instance by specifying the group in **/etc/duo/login_duo.conf**:

```
group = wheel
```

First request *sshd* to reload its configuration file with **sudo killall -HUP sshd**, and when you log into the server via SSH now, it first asks you for your password (if you have configured password authentication) or your passphrase for your private key (if you have configured public key authentication). Once the password or public key authentication succeeds, the login prompt shows you the Duo authentication choices.

If you use Duo for SSH authentication, it is strongly recommended to disable **PermitTunnel** and **AllowTcpForwarding** in your **sshd_config**, because *OpenSSH* sets up these features before it tries Duo's two-factor challenge. So in principle, an attacker who knows and enters your primary password may be able to access your internal network thanks to port forwarding before the secondary authentication method is completed. The following lines in **sshd_config** prevent this:

```
PermitTunnel no
AllowTcpForwarding no
```

Duo Mobile

By default, every user will be prompted to enroll their phone after completing primary authentication for the first time. They will have to input their phone number, which is then verified with the phone call or SMS message. After this, they can use Duo two-factor authentication. However, it's advisable that every user also installs *Duo Mobile*, a smartphone app that can generate passcodes. The login form shows this step after your phone is enrolled, but if you're happy with phone callback or SMS passcodes, you can skip it. »

Quick tip

Some VoIP providers intercept the # sign. If you're in this position, you cannot use your SIP phone to activate your admin account, so you'll have to use a mobile phone for this purpose. Once you're set up though, you can change the # sign to something else for other users.

You can let the form text you the installation link or show you the QR code for the app and, after installing it on your phone, it has to be activated to link the app to your Duo account. Activation is done by texting you an activation link or showing you a QR code, and when you visit this page on your mobile phone it generates a passcode, which you have to type into the form. And then eventually, the *Duo Mobile* app is activated and the login prompt is shown. Next time you log in, the whole activation procedure isn't needed anymore.

Authentication methods

A strong point of *Duo Security* is that it supports various authentication methods, so users have something at their disposal for every situation. Here's an overview:

1 Phone callback

If you choose the phone callback method, Duo calls your phone on login and asks you to press a key. By default, # is to authenticate and * to report fraud (for when you're suddenly called by the Duo service and you're sure you didn't start a login request yourself). You can change these or the setting so that you authenticate with the press of any key.

2 SMS passcodes

If you choose the SMS passcodes method, Duo sends you an SMS message with a batch of passcodes (10 by default). In the administrator's settings, you can change the batch size and the text shown before the passcodes. When you login, you enter the passcode starting with the shown letter. If you've used your batch, click on Send More.

3 Duo Mobile passcodes

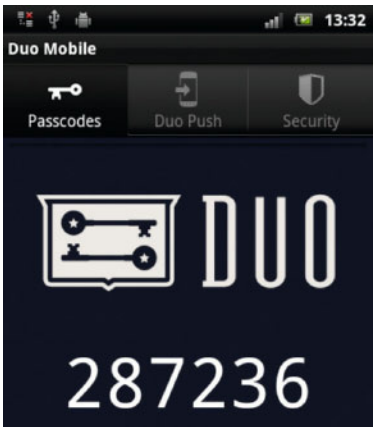
You can also generate authentication passcodes with the *Duo Mobile* app on your phone. This is interesting if you're in an area without cell or Wi-Fi coverage, because the app works completely offline. Just generate a passcode and enter it at login. Just as with the SMS passcodes, a passcode is only valid once: if someone else sees you entering it and enters the same one later, this 'replay attack' is detected and reported as a fraudulent login. The use of these passcodes requires you to install the *Duo Mobile* app on your phone.

4 Duo Push

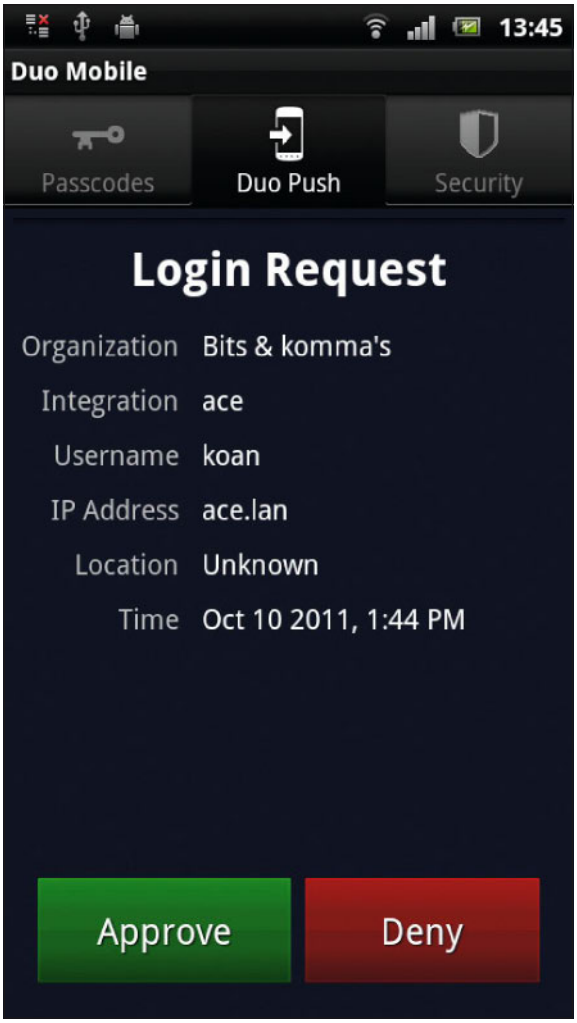
If you have the *Duo Mobile* app installed, there's another authentication option: Duo Push. It's only available for Android and iPhone because it uses push notifications. If you choose to use Duo Push and then log into your server, you'll get a push notification on your phone showing that there's a login request. If you click on the notification, the application shows you the full details of the login request. You can then approve or deny the request, after which authentication on your server succeeds or fails. If you deny the request, you can even specify whether the login request was a mistake or a fraudulent login attempt. In the latter case, the system administrator is notified of this possible malicious login attempt.

When the login shell invocation sets a specific command to run, you can even let this command be shown in the details of the login request. For this to happen, you have to add the following line to **login_duo.conf**:

```
pushinfo=yes
```



With the *Duo Mobile* app, cellular or Wi-Fi access isn't even required: just generate a six-figure passcode.



Isn't this wonderful – an SSH key that calls you back.

5 Hardware tokens

There's even a fifth authentication method: hardware tokens. Of course you have to buy a separate device for it, but then you know this device is only used for the authentication codes, so in principle it's more secure than a smartphone with many other untrusted apps on it. You can add the serial number and type of device in the Users section of the administration interface.

Until now we have only shown the use of Duo two-factor authentication for SSH logins. However, the *duo_unix* software also comes with a PAM module to provide two-factor authentication for any PAM-enabled applications, such as **su**, **sudo** and even the *GDM* login screen. If you want to use Duo with PAM, first copy **login_duo.conf to pam_duo.conf** and then change your system's PAM configuration.

The specific line you have to add to your PAM configuration (and to which PAM configuration file) depends on your system, so we refer to Duo Security's documentation for this. We used Ubuntu 10.04 LTS, and there we had to change in **/etc/pam.d/common-auth** the following lines:

```
auth    [success=1 default=ignore]  pam_unix.so nullok_
secure
auth    requisite           pam_deny.so
```

to:

Quick tip

If you don't have root access on a server and you want to secure SSH access to your account with two-factor authentication, add **command="/path/to/login_duo -c /path/to/login_duo.conf"** to the beginning of your public key in your **~/.ssh/authorized_keys** file.

Quick tip

Duo two-factor authentication is also available for various VPN solutions, including *OpenVPN*. This is a great way to secure VPN connections to your company network.

```
auth    requisite           pam_unix.so nullok_secure
auth    [success=1 default=ignore]  pam_duo.so
auth    requisite           pam_deny.so
```

As a general rule, you have to place **pam_duo.so** on the line directly after **pam_unix.so**, change the control flag (what comes right after **auth**) of **pam_unix.so** to **requisite**, and set **pam_duo.so**'s control flag to the one that **pam_unix.so** previously had. Now, instead of the login configuration, it's possible to use the same PAM configuration for *OpenSSH*. Add the following lines to your **sshd_config**:

```
UsePAM                      yes
ChallengeResponseAuthentication     yes
```

Note that if you're using Duo Security for non-interactive SSH sessions such as *scp* in a script, there's a problem with this method: the process can't prompt the user to select the authentication method. To be able to still use two-factor authentication with *scp*, the authentication module just selects the first available method. This is Duo Push if you've installed the *Duo Mobile* app on your Android or iPhone, otherwise it will be phone callback.

Powers of administration

When your other users have also enrolled, they're added to the list in the management interface. As administrator, you can manually add users or edit existing ones. For example, you can disable a specific user to prevent him logging in, or you can change his status to Bypass, which skips the two-factor authentication. You can also add multiple phones to a user, for instance a work phone, home phone or mobile.

As administrator, you can also send SMS passcodes to a user so they don't have to initiate a login first. It's also possible to provision *Duo Mobile* from the user's page in the web interface (which sends an SMS with installation and activation instructions), or generate a temporarily valid bypass code which can be used to log in if someone has lost their phone.

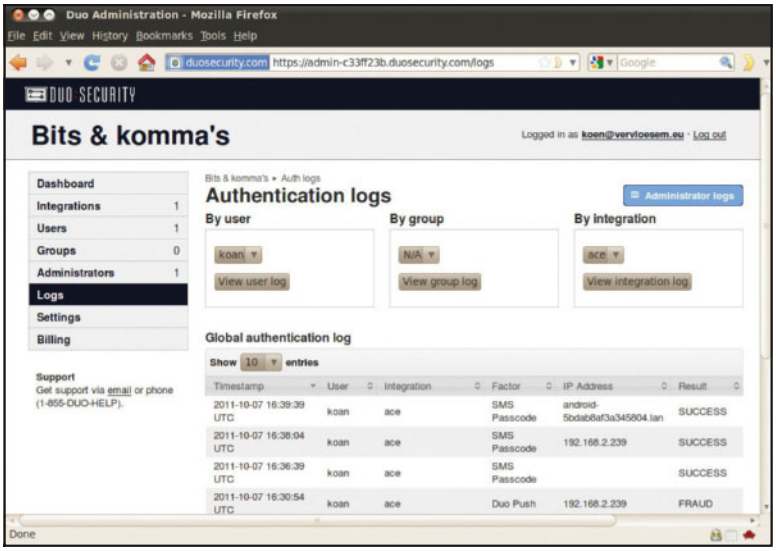
The logs show everything that has happened with your Duo Security authentication. The authentication log shows you which user has tried to log in, when, on which integration, from which IP address, with which authentication method, and with success or failure. The administrator logs show you the actions of the administrator(s), such as the addition of an integration or modifications to users.

There are also some interesting extra settings in the Settings tab in the management interface. You can change the description sent with every batch of SMS passcodes, as well as the batch size, but you can also expire SMS passcodes after a set time. For instance, you can send just one SMS passcode each time and make it only valid for five minutes.

This means that if someone who steals your user's phone and finds a batch of SMS passcodes they can't go through your two-factor authenticated login. Unfortunately, it also means that your users can't log in anymore if they have no cellular coverage. Duo Security can automatically send you a new batch of SMS passcodes when you've used the last one.

A user's account is locked when Duo Security sees 10 consecutive failed login attempts, but you can change this number. You can also enter an email address that is notified when a user reports a fraudulent authentication attempt with Duo Push, or when he is locked out due to a number of failed authentication attempts. And, as we've already said, you can also change the voice callback keys on this page.

Those of a paranoid disposition probably won't like Duo Security using its API server to authenticate you on your server; you can't have an authentication server in your own network, except in the Enterprise pricing plan. And, as there's no price listed on the website, we can probably assume it's quite expensive. So you have to trust Duo Security to handle your authentication credentials with great care.



› We'll have to look into this login attempt that has been specified as fraudulent by one of our users.

The reliance on Duo's infrastructure is minimal, though: your primary authentication method (something you know) is completely independent from Duo's (something you have), so neither its server nor its client software will ever see your SSH password or the passphrase for your private key.

If you log into your server, your own *OpenSSH* server checks your SSH credentials, and only after this has been checked does it start the Duo authentication step. So when Duo's server is compromised, the crackers don't get any access to your primary credentials.

Duo Push gives you another layer of defence. Even if someone accesses to all the secret codes in Duo's database, they would be unable to forge successful authentication responses for Duo Push: all communications between Duo's servers and a Duo Push-enabled smartphone are signed and verified by asymmetric cryptography.

The first time you set it up, an asymmetric keypair is generated: the private key is saved on your mobile and the public key on Duo's server. The private key is used to sign all authentication responses (such as your Approve or Deny responses in the *Duo Mobile* app when you receive a Duo Push login request), and the server uses the public key to verify the signature. So, even if the attacker steals the public keys, he cannot forge signatures for Duo Push authentication responses, because he needs the private key for that.

Many two-factor authentication systems have some issues that prevent them becoming popular. For instance, a lot of the two-factor solutions only support one method of authentication, such as SMS messages, one-time passwords or hardware tokens. Duo Security, in contrast, is very flexible, as it supports automated phone calls, SMS messages, smartphone apps *and* hardware tokens. This is not only about what you prefer, but also about flexibility to be able to use two-factor authentication in all circumstances: for instance you could use Duo Push when you have internet access, and you could use a passcode in a received SMS when you're in an area with poor cellular coverage. Moreover, Duo Security can be linked to websites, Unix logins, VPN logins and so on. If you're still using just a password to keep attackers out, Duo Security is a solution to seriously consider. LXF

Quick tip

If you've compiled *duo_unix* yourself, you have to run **./configure** with the **--with-pam** option to also compile the PAM module. Also install *libpam* first (**libpam-dev** on Debian/Ubuntu, **pam-devel** on Fedora/CentOS).

Quick tip

There's also a *Drupal* module and a *WordPress* plugin to integrate Duo two-factor authentication. This is an easy way to secure write access to a content management system.