OpenSSH & DNS

Emanuele Mazzoni

emanuelemazzoni93@gmail.com

16 Marzo 2017

POLITECNICO OPEN
unix LABS
Come hack with us.

Cosa è SSH? Cosa è OpenSSH?

- Secure SHell (SSH) è un protocollo di rete crittografato che permette di effettuare un login remoto su un altro dispositivo.
- OpenSSH è una suite open-source di programmi che implementano il protocollo SSH, tra cui le più importanti:
  - "ssh" per fare login e avere una shell remota
  - "scp" ed "sftp" per copiare file da e verso un server
  - "ssh-keygen" per generare chiavi crittografiche

Perché dovrei usare OpenSSH?

- Strumento fondamentale sia a livello professionale...
  - Gestione di un Server da remoto
  - Creare tunnel sicuri da utilizzare in applicazioni che non utilizzano connessioni cifrate
- ...sia a livello hobbistico!
  - Controllare un altro computer o dispositivo (ad esempio Raspberry Pi)
  - Condividere file tra vari dispositivi

Utilizzo base

- `ssh <flags> -p <porta> <username>@<host>`
- La porta di default è la 22
- L'username di default è quello dell'utente corrente (whoami)

Installazione sul Server

- Tramite package manager
  - Debian (e derivate come Ubuntu):

    ```
    $ sudo apt install openssh-server
    ```

  - Red Hat (e derivate come Fedora):

    ```
    # yum -y install openssh-server
    ```

  - Arch Linux:

    ```
    # pacman -S openssh
    ```

  - Oppure compilando da sorgenti (`https://github.com/openssh/openssh-portable`)
- Dopo l'installazione, si può avviare, riavviare e terminare come un normale demone Linux
  - $ sudo service ssh start / restart /stop
  - # systemctl start / restart /stop sshd
  - # systemctl enable sshd per far partire automaticamente il demone all'avvio

Setup nel Server

- Il file di configurazioni del server OpenSSH risiede in /etc/ssh/sshd_config
- Tra le impostazioni più spesso modificate:
  - Port: Specifica la porta in cui il demone OpenSSH ascolta
  - PermitRootLogin no per impedire il login tramite root (fortemente consigliato!)
  - AllowUsers: Permette di specificare per quali utenti è possibile fare il login via ssh (whitelist)
- Per la lista completa delle impostazioni:
  - `man sshd_config`

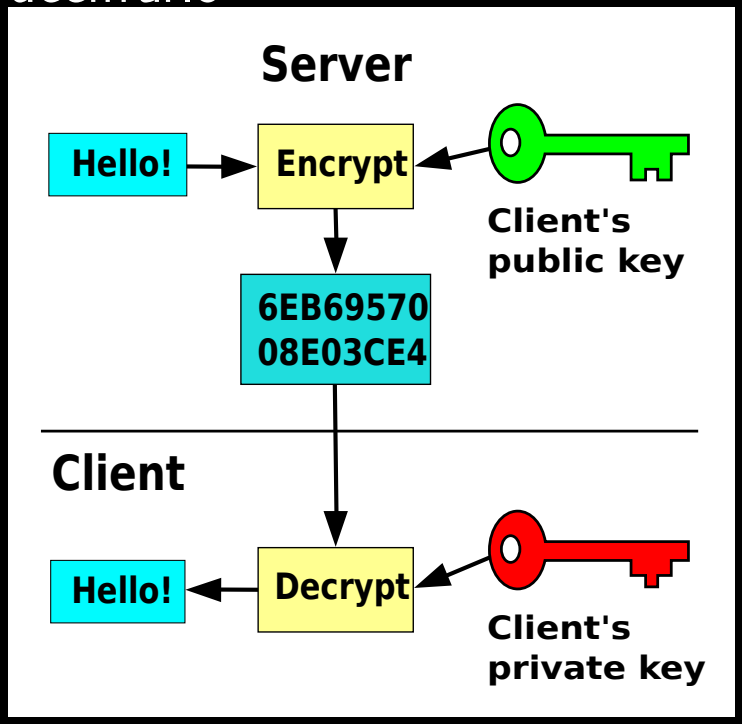
Login via Chiave Pubblica e Privata

- Il login tramite password risulta essere poco sicuro
  - La mente umana non è capace di generare password veramente casuali!
  - Soggetto ad attacchi bruteforce
  - Bisognerebbe creare password molto lunghe per ovviare a questo problema, ma andrebbero scritte manualmente ogni login e si rischia di dimenticarle più facilmente
- Possibile soluzione?
  - Impostare il login tramite chiavi crittografiche asimmetriche

Funzionamento del login a chiave Pubblica e Privata

- Il protocollo SSH utilizza Diffie–Hellman per iniziare la connessione in modo sicuro, ma non per autenticare il client che si connette!
  - Fino ad adesso riconosceva il client secondo un segreto condiviso (Password)
- La crittografia a chiave pubblica e privata funziona generando prima una coppia di chiavi crittografiche, una privata che dobbiamo tenere segreta, e una pubblica che condividiamo con tutti

Funzionamento del login a chiave Pubblica e Privata (cont.)

- Per comunicare in modo cifrato, si utilizza la chiave pubblica per cifrare il messaggio, e la propria chiave privata per decifrarlo
- 

- Generando una coppia di chiavi crittografiche, invieremo la nostra chiave pubblica al server e terremo al sicuro la chiave privata

Creazione chiavi e invio al Server

- Sul sistema da cui dobbiamo connetterci, generiamo la coppia di chiavi con il seguente comando
  - `ssh-keygen -b <nbit> -C <commento>`
  - Di default il valore di nbit è 2048, si può aumentare a 4096 se si è paranoici
  - Non serve specificare il tipo di chiave tramite il parametro -t, ssh-keygen sceglie in automatico il tipo di cifratura migliore
  - Possiamo scegliere se impostare una passphrase, servirà a proteggere la chiave salvata sul disco tramite crittografia simmetrica
- Le chiavi vengono salvate nella cartella ~/.ssh/
  - È possibile scegliere il nome della chiave tramite il parametro -i <nomechiave>
    - Di default viene chiamata id_<tipochiave> (ad esempio id_rsa)
    - La chiave privata viene salvata in quel file, mentre la chiave pubblica viene salvata come <nome>.pub (ad esempio id_rsa.pub)

Creazione chiavi e invio al Server (cont.)

- Per inviare la propria chiave pubblica al server, si può utilizzare il comodo comando della suite «ssh-copy-id»
- `ssh-copy-id -i .ssh/id_rsa.pub user@hostname`
  - Equivalente a lanciare il comando
    - `cat ~/.ssh/id_rsa.pub | ssh user@hostname 'cat >> .ssh/authorized_keys'`
- Ora connettendoci al server tramite ssh non ci chiederà la password! (ci chiederà la passphrase prima di connetterci solo se ne abbiamo impostata una)
- Dopo aver testato che la connessione tramite chiavi funziona, possiamo rimuovere il login tramite password effettuando le seguenti modifiche al file di configurazione sul server (e riavviando il demone)
  - `ChallengeResponseAuthentication no`
    `PasswordAuthentication no`
    `UsePAM no`

Secure Copy Protocol

- L'utility «scp» della suite OpenSSH permette di copiare file e cartelle con un server OpenSSH remoto
  - Funziona in modo molto simile all'utility Unix «cp»
- Per trasferire un file verso il server OpenSSH
  - scp «file» «host»:«cartella»
  - Ad esempio: scp Document.pdf emanuele@example.com:~ Copiera il mio documento pdf nella home dell'utente emanuele sul server remoto
- Per ricevere un file dal server OpenSSH
  - scp «host»:«percorso_al_file» «cartella_in_cui_salvarlo»
  - Ad esempio: scp emanuele@example.com:~/Document.pdf . Copiera il documento pdf presente nella home del server remoto nella cartella corrente
- Per copiare cartelle, aggiungere il flag -r (recursive) come per il comando «cp»
- In alternativa ad utilizzare «scp» da riga di comando, si può utilizzare «sftp» da un file browser grafico

X Forwarding

- Lanciando ssh con il flag -X, si effettua l'X forwarding durante la connessione
- Ogni programma con interfaccia grafica (che necessita del server X.org) potrà essere aperto sul server, inviando l'interfaccia grafica al client che si connnette
- Bisogna abilitare prima l'opzione nel config del server
  - `AllowTcpForwarding yes`

    `X11Forwarding yes`

OpenSSH ”Aliases”

- Per semplificare l’operazione di login, si può salvare la propria configurazione di login secondo un alias e richiamarlo in seguito

```
emanuele ~ cat .ssh/config
Host pi
    HostName raspberrypi
    Port 22
    User pi
```

- Per la lista completa delle opzioni:

  `man ssh_config`

Port Forwarding

- Il Port Forwarding locale permette di effettuare un tunnel SSH per connetterci in modo sicuro a una porta su un server remoto, tramite una porta locale

  ```
  ssh -L
  portalocale:indirizzoremoto:portaremota <host>
  ```

  - Si possono aggiungere i flag -nNT per non aprire una shell remota nel caso in cui non serva
- Esempi di utilizzo:
  - `ssh -nNT -L 8080:www.example.com:80 <host>`

    Per connetterci a un sito web tramite host remoto (inserendo nel browser locale «localhost:8080»)
  - `ssh -nNT -L 6631:localhost:631 pi@raspberrypi`

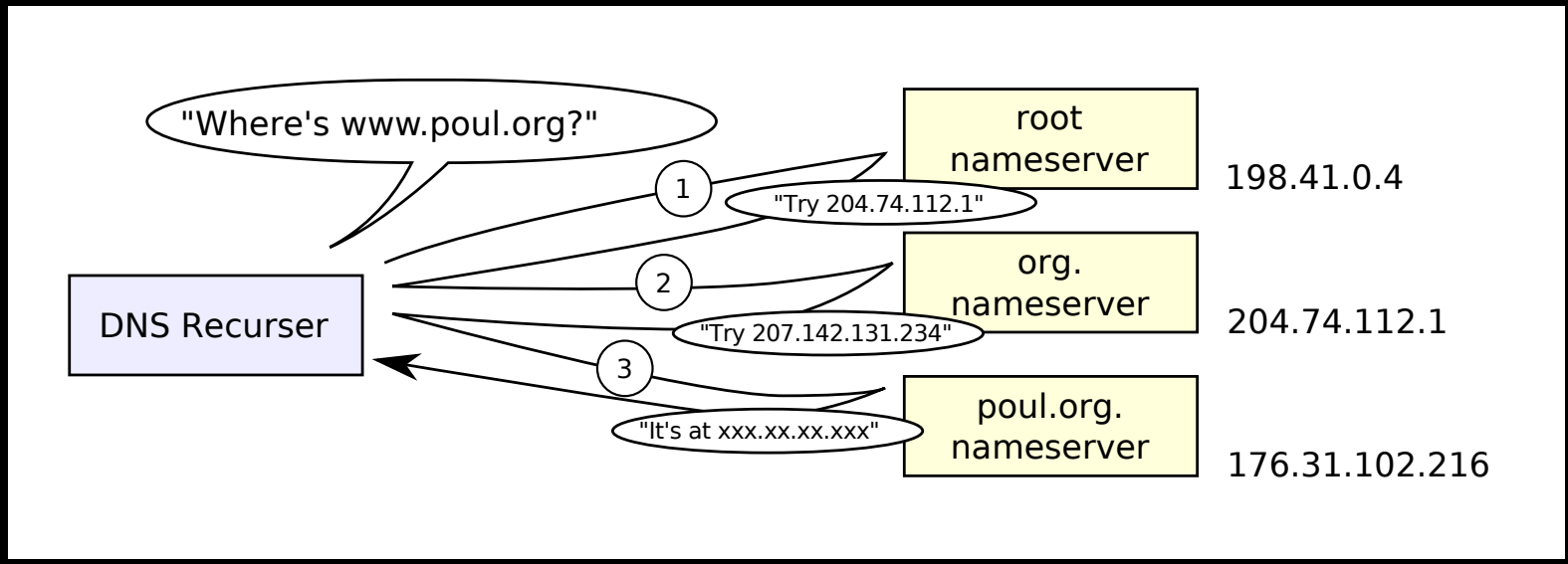
    Per aprire la schermata di CUPS per la gestione delle stampanti su un raspberrypi nella rete locale (CUPS non ascolta su connessioni esterne di default)

Escape Keys

- Inserendo ~ e ? in una connessione ssh in corso, si apre il seguente menu di aiuto:
  - Supported escape sequences:
    - ~. - terminate connection (and any multiplexed sessions)
    - ~B - send a BREAK to the remote system
    - ~C - open a command line
    - ~R - request rekey
    - ~V/v - decrease/increase verbosity (LogLevel)
    - ~^Z - suspend ssh
    - ~# - list forwarded connections
    - ~& - background ssh (when waiting for connections to terminate)
    - ~? - this message
    - ~~ - send the escape character by typing it twice (Note that escapes are only recognized immediately after newline.)
- La command line aperta tramite ~C permette di gestire il port forwarding da una connessione già aperta
- Risulta comodo terminare connessioni in stallo (ad esempio per caduta della connessione ad internet) tramite ~. perché la shell invia ogni comando di interruzione standard (CTRL-C CTRL-D CTRL-Z) alla connessione in stallo

Che cosa è un DNS?

- Cosa succede quando scrivo sul browser «www.poul.org»?



- Il browser effettua prima la richiesta al DNS server root per ottenere l'indirizzo IP del DNS server «.org»
- Poi a sua volta dal nameserver «.org» ottiene l'indirizzo IP di «poul.org»
- Infine, il browser effettua la connessione vera e propria

Come ”ottenere” un dominio proprio

- Tutti i domini (ad eccezione dei .tk) hanno un costo, spesso non irrisorio
- C’è anche un problema aggiuntivo:
  - Non sempre abbiamo un indirizzo IP fisso!
- Questi problemi sono risolti utilizzando un servizio di Dynamic DNS (DDNS)

Dynamic DNS

- Esempi di servizi di Dynamic DNS:
  - `afraid.org` (FreeDNS)
  - `duckdns.org`
  - `dnsdynamic.org`
  - `freemyip.com`
- I fornitori di servizi DDNS offrono uno script che viene avviato periodicamente tramite CRON che aggiorna le informazioni sull'IP pubblico al fornitore del servizio

FreeDNS (afraid.org)

FreeDNS - Free DNS - Dynamic DNS - Static DNS subdomain and domain hosting

For Members:

- Main Menu
- Domains
- Subdomains
- Web Forward
- Dynamic DNS
- IPv6 Reverse
- Backup DNS
- Preferences
- Registry
- Logout

For Everybody:

- Home
- About Us
- FAQ
- News
- DNS Stats
- AUP/TOS
- Contact

- Router Setup Guide

Free DNS Hosting, Dynamic DNS Hosting, Static DNS Hosting, subdomain and domain hosting.

Why is it free? It's quite simple. We wanted a challenge... that's it.

Possible Uses:

- Host your own site on your own connection from home/work/school/etc
- Access your computer with a name (like zeus.afraid.org or yourdomain.com) instead of a numeric IP address
- Run your own http server, ftp server, or anything you want to install on your computer/server
- Fetchable URL to update your IP instantly on our network if you have a dynamic address
- Hosts even work for your LAN. If you have a LAN connected to the internet you can point hosts to private IP addresses (even private IPv6 addresses) and they will work within your network
- Let your friends point theirname.yourdomain.com to their own connection
- Use web forwarding to transparently redirect a hostname to another URL. Let our servers handle the redirection
- afraid.org has been un-interrupted for hundreds of days at a time
- afraid.org is operated from multiple redundant high capacity well connected servers
- The FreeDNS router setup guide with DD-WRT is a guide that shows new users the most common/convenient configuration on a dynamic IP address, but is good for any new user to review to quickly understand the site flow.

Feature List:

- **Free DNS**, **Dynamic DNS**, **Static DNS** services
- Free subdomain hosting, free domain hosting, free backup dns, reverse IPv6 DNS hosting (forward/reverse)
- Free URL redirection [web forwarding]
- Paid services available for increased account capacity
- Unlimited number of domains per account (yes really)
- 5 free shared hostnames, use anywhere
- 20 free subdomains per domain, use on your own domains only
- **INSTANTLY** point yourname.afraid.org or yourname.com to any IP or URL
- Supports every TLD on the Internet
- Currently 78,190 other domains besides afraid.org in our shared domain pool
- Funding is supplied by the members who go premium. Funding goes directly to servers and high bandwidth connections they reside on
- Robust support for **CNAME**, **A**, **AAAA**, **MX**, **NS**, **TXT**, **LOC**, **RP**, **HINFO**, **SRV** records
- Are you a web developer? You can use nameserver branding to name our nameservers as your own! Let us worry about the maintenance/redundancy
- Round robin DNS supported (Multiple IP addresses for 1 hostname)
- IPv6 forward **AND** reverse (both .int and .arpa) supported
- **Dynamic DNS** supported, several clients for Win32 and UNIX available
- Forward your hosts to any existing URL on the internet (even to a different port if your ISP blocks 80) with the Web Forward system
- URL cloaking redirection supported, optionally hide real URL of your site in the address bar
- Allows you to change web hosting providers without messy DNS propagation delays
- Simple, fast, flexible and reliable interface, feedback is welcome
- Works with any existing web host you may already be using for both DNS and hosting
- If your web host goes down, visitors will see a "timeout" error instead of a "site does not exist" error, e-mail will also remain queued for 5 days
- If you put a domain in afraid.org, you can edit TTL, Minimum, Allow/Deny AXFR's, and approve/disapprove others from using hosts on your domain. You can also share your domain with the users of afraid.org, or your own web site visitors using our 'webclude' feature
- Support for vanity dns hosts (example: i.knew.you.were.afraid.org) currently 78,190 domains in the shared pool
- Fast and easy setup process. Setup an account in less than 5 minutes
- Extremely reliable, fast, and redundant hosting, and interface
- **All updates go live instantaneously.**

More Info, Q&A

Sign Up!

DNS Auth Trace

your.domain.com

Trace

Members:	2,839,189
Premium:	3,220
Records:	8,902,789
Zones:	990,216

+50 subdomains
+3 stealth flags
Wildcard DNS
Just $5 a month!
Go premium today!

Now accepting Bitcoin

Tip #1
Keep your email address current in the preferences area. If you forget your password, the only way you will be able to recover your account, is via the supplied email address.

Powered by... FreeBSD

Rendered in 0.002 seconds

FreeDNS (afraid.org) (cont.)

- Offre molte funzionalità avanzate (supporto ad IPv6, URL Redirection, ecc...)
- Possibilità di scelta tra quasi 80 mila domini diversi oltre ad afraid.org
- Numero illimitato di domini per singolo account

DuckDNS

- Privacy Minded
- Aggiornamento tramite semplice richiesta HTTP con token segreto
- Guida esaustiva sul setup

DuckDNS (cont.)

Duck DNS spec about why **install** faqs

Sign in with Persona | Sign in with Twitter | Login with Facebook | login with reddit | Sign in with Google

Duck DNS

free dynamic DNS hosted on Amazon VPC

support us: become a Patreon
new: DotNet Core Script & Hardware installs
update: DynDns now support V3 API

Operating Systems

linux cron | linux GUI | DotNet Core Script | windows-gui | windows-script | windows-powershell | osx | osx-homebrew | osx IP Monitor | osx-ios RealDNS | android | pi | raspbmc | ec2

Routers

openwrt | tomatoUSB | mikrotik | fritzbox | dd-wrt | allied telesis | technicolor tg582n | zte h108n | pfSense | freenas | hardware

Standards

GnuDIP.http | DynDns | inadyn | DNSOmatic

linux cron

If your linux install is running a crontab, then you can use a cron job to keep updated
we can see this with

```
ps -ef | grep cr[o]n
```

If this returns nothing - then go and read up how to install cron for your distribution of linux.
also confirm that you have **curl** installed, test this by attempting to run curl

```
curl
```

If this returns a command not found like error - then find out how to install curl for your distribution.
otherwise lets get started and make a directory to put your files in, move into it and make our main script

```
mkdir duckdns
cd duckdns
vi duck.sh
```

now copy this text and put it into the file (In vi you hit the **i** key to insert, **ESC** then **u** to undo) you **must** change your **token** and **domain** to be the one you want to update
you can pass a comma separated (no spaces) list of domains
you can if you need to hard code an IP (best not to - leave it blank and we detect your remote ip)
hit **ESC** then use use arrow keys to move the cursor **x** deletes, **i** puts you back into insert mode

```
echo url="https://www.duckdns.org/update?domains=exampledomain&token=a7c4d0ad-114e-40ef-bald-d217904a50f2
&ip=" | curl -k -o ~/duckdns/duck.log -K -
```

now save the file (In vi hit **ESC** then **:wq!** then **ENTER**)
this script will make a https request and log the output in the file duck.log
now make the duck.sh file executeable

```
chmod 700 duck.sh
```

next we will be using the cron process to make the script get run every 5 minutes

```
crontab -e
```

copy this text and paste it at the bottom of the crontab

```
*/5 * * * * ~/duckdns/duck.sh >/dev/null 2>&1
```

now save the file (**CTRL+o** then **CTRL+x**)
lets test the script

```
./duck.sh
```

this should simply return to a prompt
we can also see if the last attempt was successful (**OK** or bad **KO**)

```
cat duck.log
```

If it is KO check your Token and Domain are correct in the **duck.sh** script

what now?

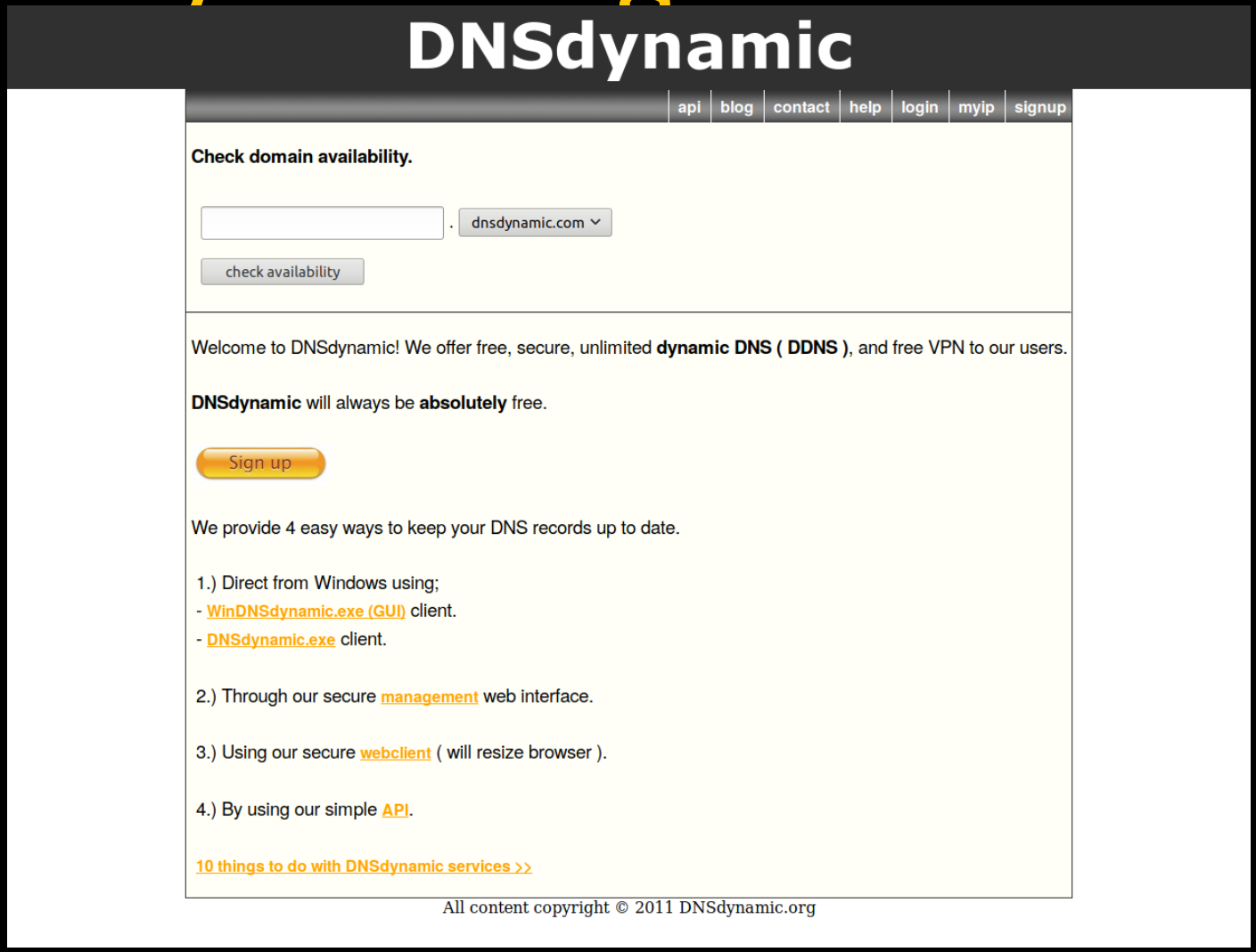
Well you probably want to setup port forwarding on your router to make use of your new DDNS name
we recommend portforward.com to learn all about this.

Invia richiesta

Bitcoin 1CoHE96MHeDkygnqAFmy6Qn9NwP6wVF7bm

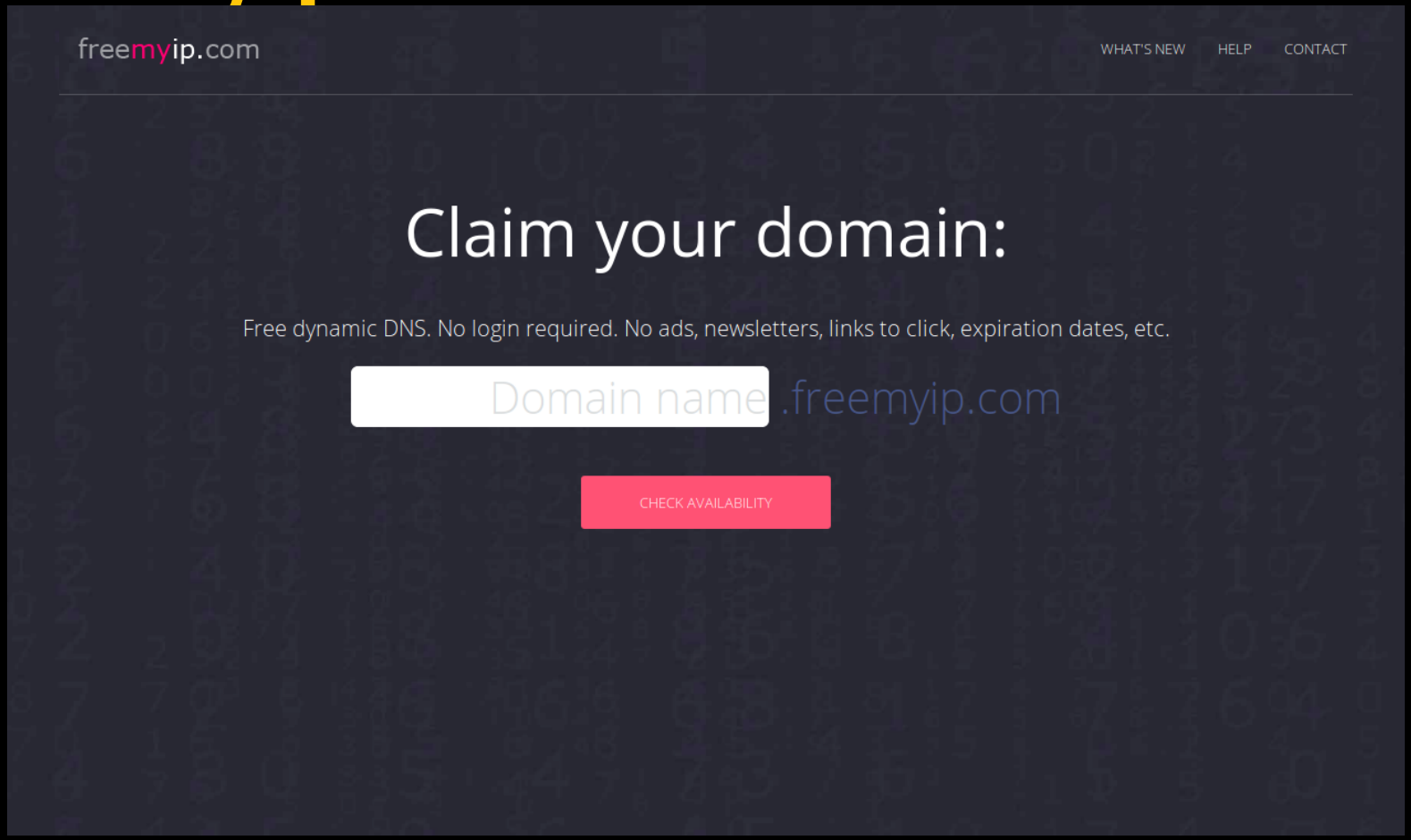
patreon

DnsDynamic.org



- Si può aggiornare l'IP anche tramite interfaccia web
- Possibilità di utilizzare le loro API per aggiornare l'IP

FreeMyIp.com



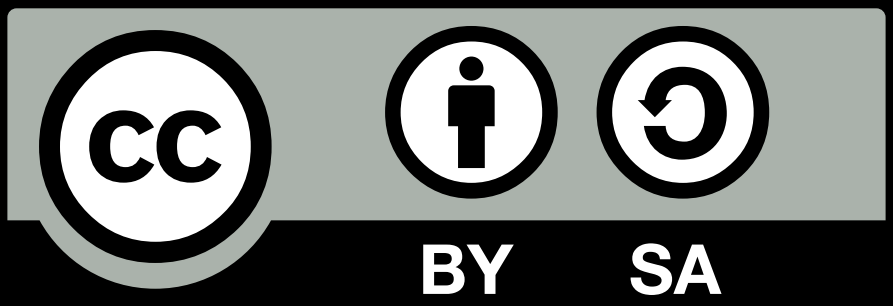
- Non richiede nessun tipo di registrazione o login
  - Il sottodominio è offerto alla prima persona che lo richiede
- Possiblità di impostare un indirizzo IP diverso da quello attuale
- Aggiornamento tramite semplice richiesta HTTP con token segreto

Fonti

- Man Pages per le informazioni
- Wikipedia per alcune immagini
- I vari siti di DDNS per le loro feature

Fine

Grazie per l'attenzione!



`http://www.poul.org`