Ssh Sifreleme Sistemi

Eger Freebsd 4.0 veya daha yüksek bir versiyonunu kullanıyorsanız sisteminiz OpenSSH ile beraber hazır halde gelmektedir.Adından da anlaşılacağı gibi SSH şifreleme sisteminin açık kod olarak kullanılışıdır.Daha fazla bilgi için OpenSSH websitesi`ne göz atabilirsiniz.

Uzun suredir kullanılan telnet programı ile sisteme girildiğinde sanki terminalin karsısında oturuyormuş gibi bütün istediğini yapabiliyordu.Yani her klavye kullanılışı klavye girdileri ağ üzerinden dolaşıp geliyor olca bile sisteme bağlı klavyeden geliyormuş gibi alınıyordu.Tabiki telnet`in en büyük handikaplarından biri klavye girdilerini ve buna gelen cevapların text olarak gönderilmesi ve bir sniffer tarafından izlenmesi sonucunda bu bilgilerin alınabileceği idi.

SSH`de telnet ile ayni mantıkta çalışarak sanki kullanıcı sistemde fiziksel olarak oturuyormuş gibi çalışmaktadır.Sisteme bağlanılması durumunda daha giriş komut satiri belirmeden her iki bilgisayar arasında iletişim sırasında bilgileri şifrelemek için kullanılacak bir anahtar oluşturulur.

SSH karşıdaki sisteme giriş için kullanılacağından, karşı bilgisayarda bir kullanıcı hesabi bulunmalı ve Tabiki SSH servisi çalışıyor olmalı.Normalde ssh istemcisi TCP 22 port üzerinden servise bağlanacaktır.

Freebsd sisteminiz SSH`yi olduğu gibi kullanmanıza izin verir.Yapmanız gereken tek şey `/etc/rc.conf` dosyasında `ssh_enable="YES"` parametresini bulundurmanız.

Daha iyi anlatmak için iki bilgisayar kullanacağım.10.0.0.1 SSH sunucusu ve 10.0.0.2 SSH istemcisi olacak.Her iki sistemde de “quantum” adında birer kullanıcı oluşturdum.

İlk önce sunucu sistemde host anahtarlarının oluşturulup oluşturulmadığını bulmamız gerekiyor.10.0.0.1`de su komutu deniyorum:

```
ls /etc/ssh
```

Eğer sonuç olarak:

```
moduli                  ssh_host_dsa_key.pub   ssh_host_rsa_key
ssh_config              ssh_host_key           ssh_host_rsa_key.pub
ssh_host_dsa_key        ssh_host_key.pub       sshd_config
```

alırsam gerekli anahtarların oluşturulduğunu anlıyorum.Fakat çıktı olarak:

```
moduli          ssh_config      sshd_config
```

alırsam herhangi bir anahtar yaratılmadığını anlıyorum.SSH sunucusunu başlatabilmem için bunları yaratmama gerekecek.Eger yaratmadan SSH sunucusunu başlatırsam alacağım mesaj su olacaktır:

```
sshd
Could not load host key: /etc/ssh/ssh_host_key
Could not load host key: /etc/ssh/ssh_host_dsa_key
Disabling protocol version 1. Could not load host key
Disabling protocol version 2. Could not load host key
sshd: no hostkeys available -- exiting.
```

Anahtar yaratmak için herhangi bir komut girmenize gerek yok.Daha önce bahsettiğim gibi `/etc/rc.conf` dosyası içinde :

```
sshd_enable="YES"
```

Parametresini ekleyerek bunların otomatik yaratılmasını sağlamış oluyoruz.

Rc.conf dosyasına gerekli parametreyi girip sakladıktan sonra

```
shutdown -r now
```

Komutu ile sistemi yeniden başlatıyoruz.

Sistem acılısında SSH host anahtarlarının yaratılısını görebiliriz:

```
Starting final network daemons: creating ssh1 RSA host key
Generating public/private rsa1 key pair.
Your identification has been saved in /etc/ssh/ssh_host_key.
Your public key has been saved in /etc/ ssh/ssh_host_key.pub.
The key fingerprint is:
12:d9:3d:f3:95:92:0e:e7:6b:54:09:80:77:a0:3e:cf root@hostname
creating ssh2 RSA host key
Generating public/private rsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_rsa_key.
Your public key has been saved in /etc/ssh/ssh_host_rsa_key.pub.
The key fingerprint is:
4b:cf:7e:af:f1:a8:01:08:64:1b:c0:79:e3:a2:58:78 root@hostname
creating ssh2 DSA host key
Generating public/private dsa key pair.
Your identification has been saved in /etc/ssh/ssh_host_dsa_key.
Your public key has been saved in /etc/ssh/ssh_host_dsa_key.pub.
The key fingerprint is:
22:69:d7:05:23:c6:db:d9:55:2a:20:a3:34:bd:f4:ef root@hostname
```

Şimdi bu çıktıya bir göz atalım.Fark ettiginiz gibi 3 ayri anahtar oluşturuldu.Rsa1, rsa ve dsa.Peki neden bu kadar çok anahtar oluşturuluyor?SSH protokolünün iki versiyonu bulunmakta ve OpenSSH bunların ikisini de desteklemekte.Rsa1 SSH v1 tarafından RSA ve DSA ise ssh2(v2) tarafından kullanılmakta.

Her anahtar ayri bir dosyada saklanmakta.Public(genel) anahtarları sonları `.pub` ile bittiği için kolayca tanımlayabilirsiniz.Buna ek olarak sadece o anahtara özel olan bir tanımlayıcı bulunmakta.

SSH sunucumuzun başlatılıp başlatılmadığından emin olmak için `sockstat` komutunu kullanabiliriz:

```
sockstat |grep ssh

root     sshd     69820    3 tcp4   *:22                  *:*
root     sshd     69820    3 tcp46  *:22                  *:*
```

10.0.0.1 22. port`u dinlediğinde göre 10.0.0.2`den “quantum” kullanıcısı ile bağlanmaya çalıştığımızda ne olduğunu görelim:

```
ssh 10.0.0.1
The authenticity of host '10.0.0.1 (10.0.0.1)' can't be established.
DSA key fingerprint is 22:69:d7:05:23:c6:db:d9:55:2a:20:a3:34:bd:f4:ef.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added '10.0.0.1' (DSA) to the list of known hosts.
Password:
```

Quantum kullanıcısı için şifreyi doğru olarak yazdığımda giriş mesajını alır ve daha sonra sistemden ayrılmak için “exit” komutunu kullanabiliriz

```
exit
logout
Connection to 10.0.0.1 closed.
```

Simdi ikinci kez denilelim:

```
ssh 10.0.0.1
Password:
```

İlk kez SSH kullandığımda karsıdaki sunucunun DSA anahtarını doğrulamam gerekmişti.Doğruladıktan 10.0.0.2`de bu anahtar kaydediliyor:

```
more ~genisis/.ssh/known_hosts
10.0.0.1 ssh-dss
AAAAB3NzaC1kc3MAAACBAPiOFgV4PpsbXfkaxGD+hCr02Cv9P3OJDKfaXge059
cSohLN/n/kd2Nz/E1mDvT4Y8nSAQL7M667iMeqJ0WTpcdI59ktuPtvOsYBc7SNoJ6aPqqoKo682mAf
C
NpUFZ3jirbWGFnaF3WpJsWFyeOY6vyD4hVT6CkunL2ovoYSJND7AAAAFQDjZ2TNBixZByXB+h00wxC
N
tHZ8zQAAAIEAl4lePs+e5v9f1H93l2GLXtxXhXyasr+X42HnKgKQTMR+iLgxhtD0Eb/ftTMK+n2ECn
9
3MwCNTgx5tdGX06dyBdK5xEfjV4tJnmnP42UweBwOHKpRkNLiMBN4onh7KKXjhXWmH0MpO5fhaHy6k
0
f+yTTLckCKd2IO/TgGJitjlo4AAACBAM+3JMr8M+MQoa6D7BU0pNJVGoTmGdxrLotMNLmUdqM0xIFK
r
3dBrgqY+gsDciQEG1CSqDDhusrkz3LRBmnuG68tE7WPPjzGZrT46ZYCmMeZume67xVN0dDd57BuxmH
K
B7iKvmlM0v+EkvJ0XTlNCwBTWuU3cdTdhkWT7swxGhvf
```

İkinci kez sunucuya bağlanmam sırasında bu anahtar karsıdan gönderilen anahtar ile tekrar karsılaştırıldı ve ayni oldukları için ikinci bir kez bunu doğrulamam istenmedi.

Arka planda neler oluyor onlara bir göz atalım:

1. Ssh istemcisi ssh sunucusu ile 22. port`tan iletişime geçiyor.
2. Ssh sunucusu ssh istemcisine kendi kimligini belirtmek icin public(genel) anahtarini yolluyor.
3. Ssh istemcisi sunucu genel anahtarı ve kendisinde bulunan kopyayı karsılaştırıyor.Eger kopyası yok ise kullanıcıya bunu doğrulama icin sunuyor.
4. Sunucu doğrulandığında sunucu ve istemci arasında iletişimde kullanılacak anahtar oluşturuluyor.
5. Bu sefer sunucu istemciden kimligini doğrulamasını istiyor.Kullanıcı şifreyi girdiğinde bu şifre şifreli olarak sunucuya iletilip doğrulanıyor.
6. Kullanıcı da doğrulandığında artık sisteme giriş yapılıyor ve iki yönlü olarak şifreli iletişime geceliyor.

OpenSSH diğer şifreleme yöntemlerini de desteklemekte fakat Freebsd sisteminizde kurulumla gelen konfigürasyonda kullanılan yöntemler bunlar.

Kullanıcının normalde şifre dışında herhangi bir anahtara ihtiyacı olmadığını gördük.Simdi bunu değiştirelim.İlk önce 10.0.0.2`de quantum kullanıcısı icin `ssh-keygen` komutu ile anahtar yaratalım:

```
ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/home/genisis/.ssh/id_rsa):
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/genisis/.ssh/id_rsa.
```

```
Your public key has been saved in /home/genisis/.ssh/id_rsa.pub.
The key fingerprint is:
fd:5a:cc:cf:a9:f0:ea:9c:93:ea:1a:04:48:b1:47:14 genisis@hostname
```

Ssh-keygen komutunu kullanırken t switch`iyle hangi tür anahtar yaratmak istediğimizi belirtiyoruz.Ssh v.2 v.1`den daha güvenli ve RSA, DSA`dan daha güvenli olduğu icin Ssh v.2 RSA anahtarı yaratıyorum.Bunu yaratırken daha sonra bu anahtarın sahibi olduğumu göstermeme yarayacak olan bir şifre soruluyor.Private (özel) anahtarımı kullanacağım zaman bu şifre bana sorulacağı icin hatırlanabilir bir şifre yazıyorum.Eğer bu şifreyi unutursam tekrar yeni bir anahtar oluşturmak zorunda kalırım.

Unutmayın ki public(genel) anahtar genel kullanım içindir fakat private(özel) anahtarınızın saklanması size aittir.Simdi bu anahtarlar üzerindeki kurulum ile gelen izinlere bir göz atalım.

```
ls ~genisis/.ssh
total 4
-rw-------  1 genisis  genisis  951 Nov  9 15:00 id_rsa
-rw-r--r--  1 genisis  genisis  247 Nov  9 15:00 id_rsa.pub
```

Özel anahtarınız, `id_rsa`, sadece quantum kullanıcısı tarafından okunup yazılabilir halde.Genel anahtar , `id_rsa.pub`, ise herkes tarafından okunabilir halde.Eger `file` komutunu kullanırsam, her iki dosyanında ASCII formatında olduğunu ve dolayısıyla bir pager vasıtası ile okunabileceğini görebilirim.

```
file ~genisis/.ssh/*
id_rsa:     ASCII text
id_rsa.pub: ASCII text

more id_rsa.pub
ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAIEA1gfc4NRnq9K17TLqhhKT3L6feKUttHTJvM054k+WhjI
vsdt4YoeNa3m6lplnOxwOh2w6o+xu+xuiHa/CQkvkAdxFU1ZGtnxtQWV06QJdodUEk55U/0y417TaD
F
H1aYjsgPPSpjulKCLQv263C9KOSpjDrjZ74ZLOlQHtsJINY2c= genisis@hostname
```

Anahtarı yaratmak yeterli değil.Hala SSH sunucusundaki home klasörüme genel anahtarımı kopyalamam gerekiyor.Sunucu benim genel anahtarımı aldıktan sonra her girişte SSH genel anahtarımı bana soracak, karsılaştıracak ve eğer ayni ise girişim onaylatılacak.

Kopyalama işlemi için `scp` veya güvenli kopyalama komutunu kullanacağız.Bu komut OpenSSH ile geliyor ve şifreli bir oturum üzerinden dosya transfer etmenize izin veriyor.Kullanımı bildiğimiz `cp` komutuyla ayni.Tek farkı karsı tarafa göndereceğimiz dosyanın isminin karsı sistemin IP numarası ile başlaması.Daha sonra bir boşluk verip karsı taraftaki dosya ismini yazıyoruz.

10.0.0.2`de genel anahtarımızı karsı sunucuya kopyalayalım:

```
scp ~/.ssh/id_rsa.pub 10.0.0.1:~/.ssh/authorized_keys
Password:
id_rsa.pub:    100% |**************************|   247      00:00
```

Burda da görüldüğü gibi özel anahtarımı değil, genel anahtarımı karsı tarafa kopyaladım.Bunun yanında karsı tarafta genel anahtarimi yükleyeceğim dosya ismini `~/.ssh/authorized_keys`. Olarak belirledim.

Simdi SSH`yi tekrar denilelim:

```
ssh 10.0.0.1
```

Bu sefer şifre sorulmadan girebildim.Karsı taraf benim genel anahtarımla kendi anahtarini karsılaştırdı ve aynı olduğu için girişe izin verdi.

`ssh`, kullanırken karsı tarafa bağlanacağınız kullanıcı ile aynı kullanıcı altında bağlı olmak zorunda değilsiniz.Örneğin istemci makinede “özgür” kullanıcısı olarak çalışırken karsı tarafa “quantum” olarak bağlanabiliriz.Bunun icin `l` switch`ini kullanıyoruz:

```
ssh -l genisis 10.0.0.1
```

Bunun yanında ayrıca:

```
ssh genisis@10.0.0.1
```

Komutunu da kullanabiliriz.

Özgür adlı kullanıcı home klasöründe quantum kullanıcısının genel anahtarının bir kopyası bulunmadığı icin şifre ile doğrulama yoluna gidilir.Sorulduğunda quantum kullanıcısı icin kullanılan şifreyi girip doğrulama yapıldıktan sonra karsı sisteme giriş yapabiliriz.

Simdi de superuser(root) olarak karsı tarafa bağlanmaya çalısalım:

```
ssh 10.0.0.1
Password:
Password:
Password:
root@10.0.0.1's password:
Permission denied. Please try again.
root@10.0.0.1's password:
^c
```

Gerçek root şifresini yazsam da system girişime izin vermeyecektir.Bu güvenlik acısından düşünülürse çok iyi.Karsı sisteme ilk once bir kullanıcı olarak bağlanmalı sonra root olmalısınız.Böylece root şifresini eline geçiren bir kişi bile, diğer bir kullanıcı ve şifresini bilmeden SSH üzerinden sisteme bağlanamayacaktır.

Özgür Özdemircili
http://www.enderunix.org
http://news.enderunix.org
http://haber.enderunix.org

Sorularınız için : dionypheles@gmx.net

Kaynaklar

Freebsd Handbook