Güvenli kanallardan iletişim (SSH) 

SSH Nedir ?

SSH(Secure Shell/Güvenli Kabuk) ağ üzerinden başka bilgisayarlara erişim sağlamak, uzak bir bilgisayarda komutlar çalıştırmak ve bir bilgisayardan diğerine dosya transferi amaçlı geliştirilmiş bir protokoldür. Güvensiz kanallar(internet vs) üzerinden güvenli haberleşme olanağı sağlar. Bir iletişimde SSH aşağıda belirtilen temel unsurları sağlar.

- authentication /Kimlik denetimi
- encryption /Şifreleme
- Integrity /Bütünlük.

SSH ile ilgili temel tanımlar

SSH1, Tatu Ylönen tarafından geliştirilen ilk orjinal SSH ürünü. SSH-1 protokolü temel alınarak geliştirilmiştir.

SSH2, Tatu Ylönen tarafından geliştirilen SSH-2 ürünü. *www.ssh.com*

SSH-1, SSH protocol 1.

SSH-2, SSH protocol 2 . Günümüzde yaygın kullanımda olan ve kullanımı tavsiye edilen ssh sürümü. IETF SECSH çalışma grubu tarafından standartları belirlenmiştir.

SSH Tarihçesi ve OpenSSH

SSH-1 protokolu ve SSH1, ilk olarak 1995 yılında Helsinki teknoloji universitesinde araştırma görevlisi olan "Tatu Ylönen" tarafından geliştirilmiştir. Aynı yılın Haziran ayında SSH1 kaynak kodları ile birlikte özgür olarak dağıtılmaya başlamıştır.

SSH protokolünün özgür bir uyarlaması olan OpenSSH *BSD Unix'lerin asi çocuğu olarak adlandırılan OpenBSD projesi çerçevesinde yürütülen SSH1 ve SSH2 protokollerini içeren yazılım takımıdır.

OpenSSH son özgür SSH versiyonu olan **ssh1.2.12** den türetilmistir. **Markus Friedl (**Aaron Campbell, Bob Beck, Niels Provos, Theo de Raadt, Dug Song**)** önderliğinde geliştirilen OpenSSH projesi dünya üzerinde birçok yazılımcının

katılımı ile iyi bir yol katetmiştir. Göreceli olarak özgür yazılım projeleri arasında en fazla kullanılan yazılımlardan biridir.

OpenBSD ile birlikte dağıtılan OpenSSH sürümü hariç diğer tüm sürümler OpenBSD için geliştirilen sürümün gerekli sisteme uyarlanmış halleridir(port edilmis).

OpenSSH'ın birçok platforma uyarlanmış sürümlerini bulabilirsiniz ve platformlar arası kullanımı çok az farklılıklar gösterir. Aşağıda OpenSSH'ın kullanılabileceği bazı platformlara örnek verilmiştir. Detaylı bilgi ve liste için **http://www.openssh.org/portable.html** adresini ziyaret edilebilir.

AIX, HP-UX , Irix , Linux , NeXT , SCO , SNI/Reliant Unix , Solaris , Digital , Unix/Tru64/OSF, Mac OS X

OpenSSH, bu liste haricinde Windows ortamında da çalışmaktadır. OpenSSH'ı windows üzerinde kullanmak için (http://sshwindows.sourceforge.net/) adresindeki cygwin+openssh windows portunun yüklenmesi gerekir.

Internette kullanılan SSH sunucuların büyük bir çoğunluğu (~%90)nu OpenSSH oluşturmaktadır. Bu sonuçlar scanSSH(**http://www.monkey.org/~provos/scanssh/**) adlı program ile yapılan değişik tarama sonuçlarından çıkarılmıştır. Detaylı bilgi için **http://www.openssh.com/usage/index.html**

OpenSSH BSD lisansı ile dağıtılmaktadır. OpenSSH lisansı ile ilgili detaylı bilgi için: **http://www.openbsd.org/cgi-bin/cvsweb/src/usr.bin/ssh/LICENCE?rev=HEAD**

Kullanım Alanları

SSH güvenli iletişimin gerektiği her ortamda kullanılabilir. Sadece karşı sisteme bağlanıp komut çalıştırmak ya da dosya aktarımı yapmak için değil, doğasında güvensiz(şifrelenmemiş trafik) olarak çalışan protokoller SSH üzerinden güvenli bir şekilde kullanabilir.

Mesela POP3 servisi ağ üzerinden tüm iletişimini şifrelenmemiş şekilde gerçekleştirir, biz pop3 servisini SSH üzerinden aktararak şifrelenmiş ve güvenli hale getirebiliriz.

OpenSSH Kurulumu

OpenSSH birçok Unix/Linux dağıtımı ile öntanımlı olarak gelmektedir. OpenSSH kurulumu Kullandığınız Linux/UNIX dağıtımına gore değişiklik gösterebilir. Sistemde kurulu degilse Kullanılan paket yönetim sistemi kullanılarak son sürüm(bu yazı hazırlanırken 4.2 idi.)OpenSSH sisteme kurulur. Aşağıdaki komutla hangi SSH versiyonunun kullanıldığı öğrenilebilir.

$ssh -V

OpenSSH_4.2, OpenSSL 0.9.7g 11 Apr 2005

Temel SSH Kullanımı

Herhangi bir SSH sunucuya ilk bağlanıldığında SSH istemcisi bir uyarı verir. Bu
uyarıda bağlandığı sunucuya daha once baglanmadığını belirtir ve sunucu kimlik bilgisi yerine geçen ait rsa anahtarini ekrana basar, yes dedikten sonra da bunu bir dosyaya(~/.ssh/known_hosts) kaydeder ve bir sonraki bağlantıda sormaz. Eğer sunucu Ip adresi ya da SSH sunucusunda kimlik değişimi gibi bir değişiklik olursa bu uyarı farklı bir şekilde tekrar görünecektir.

$ ssh ssh_sunucu
The authenticity of host 'ssh_sunucu (14.2.7.x)' can't be established.
RSA key fingerprint is f3:ce:14:99:d7:19:44:ca:ff:5e:83:b6:79:52:4e:45.
Are you sure you want to continue connecting (yes/no)?yes
Warning: Permanently added 'ssh_sunucu,14.2.7.x' (RSA) to the list of known hosts.
huzeyfe@ssh_sunucu's password:

Sizden karşı sisteme erişmek için gerekli parolayı girmenizi bekler. Burada dikkatinizi bir noktaya çekmek istiyorum, ssh ile herhangi bir sunucuya bağlanmak istediğimizde SSH varsayılan olarak karşı sisteme şuanki yerel kullanıcı adınızla gireceğinizi düşünür. Bunu karşı sistemde bu kullanıcı adı var mı yok mu karşılaştırmadan işletir. Benim kullandığım makinedeki yerel kullancı adım huzeyfe,bunu

$ echo $LOGNAME
huzeyfe

komutu ile öğrenebilirsiniz. SSH_sunucu makinesine ssh ile bağlanmaya çalıştığımda yukarıda bahsettiğim durum gerçekleşti ve bana hangi kullanıcı adımla bağlanacağımı sormadan şu anki bağlı bulunduğum kullanıcı adı ile bağlantı sağladı. Bunu değiştirmek için OpenSSH bize -l parametresi ile kullanılan kullanıcı adı belirtme seçeneği sunar. Kullanımı şu şekildedir;

$ ssh -l rapsodi ssh_sunucu

-l parametresinden sonra karşı sisteme bağlanmak istediğimiz kullanıcı adı girilir ya da -l parametresi ile aynı işlevi sağlayan aşağıdaki yöntem de kullanılabilir.

$ ssh huzeyfe@enderunix.org

huzeyfe 'SSH sistemine bağlanmak istediğimiz kullanıcı adı
@ 'birleştirici karekter
enderunix.org 'bağlanmak istediğimiz SSH sunucunun adı ya da IP adresi

Farklı Porttan Çalışan SSH sunucularına Bağlantı

Buraya kadar olan örneklerimizde SSH sunucunun SSH sunucu yazılımını varsayılan olarak 22. porttan hizmet verdirdiğini hesaba katarak işlem yaptık. SSH sunucu yazılımı başka bir porttan hizmet veriyorsa bu durumda -p parametresi ile hangi porttan bağlanmak istenildiği belirtilebilir.

Aşağıdaki örnekleri inceleyerek durumu daha iyi kavrayalım, ilk örnekte ssh sunucumuza -p parametresi belirlemeden bağlanmaya çalışıyoruz,fakat karşı sistemdeki ssh sunucusu 22.porttan hizmet vermiyor ve bize bağlantı redddedildi mesajı yolluyor.

$ ssh -l huzeyfe enderunix.org
ssh: connect to host enderunix.org port 22: Connection refused

2.örneğimizde ise karşı sistemin hangi porttan SSH hizmeti verdiğini belirterek bağlanmaya çalışıyoruz ve başarılı oluyoruz.

$ ssh –l huzeyfe enderunix.org -p 200

The authenticity of host 'enderunix.org (14.2.7.8)' can't be established.
RSA key fingerprint is 3f:98:e8:53:d7:62:1a:34:2e:57:39:47:f2:19:66:ea.
Are you sure you want to continue connecting (yes/no)?

Görüldüğü gibi karşı sistemdeki SSH sunucusu 200. porttan hizmet vermektedir

Uzak sistemlerde komut çalıştırmak

SSH'in uzak sistemlere bağlanmadan komut çalıştırıp çıktısını kendi ekranda alma imkanı da sunar.

Örnek;

$ssh huzeyfe@cc.kou.edu.tr ls /home/huzeyfe/

The authenticity of host 'cc.kou.edu.tr (1.2.7.8)' can't be established.

DSA key fingerprint is a6:d6:35:52:75:66:63:15:5d:f6:76:b4:52:56:b4:64.

Are you sure you want to continue connecting (yes/no)? yes

Warning: Permanently added 'cc.kou.edu.tr,1.2.7.8' (DSA) to the list of known hosts.

Password: XXX

6.0-BETA4-1-i386-disc2.iso

6.0-BETA4-i386-disc2.iso

6.0-BETA4-i386-disc2.iso.1

6.0-BETA4-i386-disc2.iso.2

Maildir

Manning,.Swt.JFace.in.Action.(2004).LiB(1).pdf

Manning.JUnit.Recipes.2005.pdf

Ev.

…

SSH ile Dosya Transferi

Günümüzde kullanılan en popülar dosya transfer aracı FTP'dir. SSH kullanarak hem ftp kolaylığında dosya transferi yapılabilir; hem de transfer edilen dosya şifrelenerek meraklı gözlerden korunmuş olur. SSH ile dosya transferinde temel iki seçenek var; biri SCP(secure Copy) diğeri de Sftp(Secure FTP). Temel bazı farklılıklar dışında her iki yöntem ile yapılabilecekler aynıdır.

SCP Kullanımı

$scp kullanici_adi@sunucu1:dosya kullanici2@sunucu2:dosya

Örnek;

a makinesinden test.exe dosyasını be makinesinin /usr/tmp dizinine kopyalamak için,

huzeyfe@a$ scp test.exe huzeyfe@b:/usr/tmp/

Sftp Kullanımı

$sftp huzeyfe@test.enderunix.org
Connecting to test.enderunix.org...
Password:
sftp> get ev
Fetching /usr/home/huzeyfe/ev to ev
Cannot download non-regular file: /usr/home/huzeyfe/ev
Sftp> help
...
...
Not: Sftp ile sadece binary modda transfer yapılabilir

SSH Sunucu Konfigürasyonu

SSH Sunucuyu farklı porttan Çalıştırmak

İstemci tarafında farklı porttan bağlanmayı gördük şimdide sunucu tarafında SSH sununucumuzun farklı porttan hizmet vermesi için gerekli olan işlemlere bakalım. OpenSSH'ın kullandığı yapılandırma dosyaları /etc/ssh/ dizininde bulunur(başka Linux/UNIX versiyonlarında başka dizinlerde bulunabilir)

 # cd /etc/ssh
#ls -l

-rw-r--r-- 1 root root 1167 Eyl 17 2003 ***ssh_config***
-rw------- 1 root root 2474 Eyl 17 2003 ***sshd_config***

….

sunucu için ayarlama yapacağımıza gore inceleyeceğimiz dosya sshd_config dır. Herhangi bir editörle bu dosyayı açarak #Port 22 satırını bulunuz ve bunun önündeki # karekterini kaldırarak 22 yerine de istediğiniz port numarasını yazınız. Bunları yaptıktan sonra da SSH sunucusunu yeniden başlatmayı unutmayınız Red Hat tabanlı sistemlerde bunu

#/etc/init.d/sshd restart

komutu ile yapabilirsiniz. sshd_config dosyası bize bundan daha fazlasını sunar. sshd_config dosyasındaki tüm seçenekleri ve yapılandırma ayarlarını görebilmek için

 # man sshd_config

komutunu vermeniz yeterlidir.

Root olarak SSH sunucusuna giriş

Root olarak sisteme giriş iznini değiştirebileceğiniz yapılandırma satırı

#PermitRootLogin yes dır, YES yaparak root kullanıcısının sisteme ssh üzrinden bağlanmasını, no yaparak da bağlanamamasını sağlayabilirsiniz.

SSH'a belirli kullanıcıların bağlanma izni

Bunun için kullanmamız gereken seçenek AllowUsers 'dır.

AllowUsers Huzeyfe, ismail, murat

Bu tanımlama ile bu ssh sunucusuna sadece yukarıda ismi yazılan kullanıcıların girebilmesini sağlamış oldu. Burada kullanıcı adı yerine *** ?** şeklinde joker karekterler de kullanabiliriz. Mesela sadece son üç harfi ray olan kullanıcıların bağlanmasını sağlamak için
AllowUsers *ray

Şeklinde bir tanımlama yapabiliriz. Bu durumda Giray, nuray, firay ray kullanıcıları SSH sunucuya bağlanabilecektir.

Belli kullanıcı ya da grubun bağlanmasını engelleme :

Grup yasaklaması için ;

DenyGroups root bin admin

Kullanıcı yasaklaması için ;

DenyUsers cin ali

Allowusers opsiyonunda geçerli olan "wildcard(* ?)"kullanımı burada da geçerlidir.

Konfigürasyon dosyası kontrolü

sshd –t
Komutu ile sshd_config dosyasındaki yanlış yazımlar control edilebilir.

Karşılama Mesajı

SSH sunucuya bağlanan kullanıcıya sistemle ilgili uyarı amaçlı bilgi mesajı gösterilebilir. Bunun için OpenSSH özel bir dosyada metni kullanıcıya gösterme olanağı sunar.

Banner /usr/local/etc/warning.txt

Ek olarak OpenSSH'da kullanıcı sisteme girdikten sonra *message of the day*(motd) ile bilgilendirme yapılır. Bunu kapamak Printmotd ile yapılabilir.

Printmotd no

SSH ile ileri seviye Uygulamalar

Anahtar ile Kimlik dogrulama

SSH, kullanici adi/parola ikilisi haricinde şifreli anahtarlar aracılığı ile de kimlik kontrolü yapabilir.

public key authentication

Anahtarlar sifreleme dünyasindaki kimliklerimizdir. Kimligimiz 2 anahtardan olusur; biri açik anahtarimiz -ki bunu herkesle paylasiriz-. Digeri de gizli anahtarimiz bunu sadece biz biliriz.

Anahtar ile kimlik doğrulama adımları;

1. ssh istemcisi sunucuya xxx kullanici adi ile baglanmak istedigini belirtti
2. sunucu istemciden gelen istegi alir ve istemcinin kendini kanitlamasi icin challenge mesaji gonderir
3. istemci challange olarak gizli anahtarini ve gelen challange verisini kullanarak sunucuya cevap döner

4. sunucu kendi tarafinda gelen sifreyi karsilastirarak kullaniciya giris hakki tanir ya da reddeder.

NOT: Burada dikkat edilmesi gerken nokta bu iletisimde arada ne açik anahtar ne de gizli anahtar geçmediğidir.

SSH istemcisinde anahtarlari olusturmak

ssh anahtar çiftini olusturmak için OpenSSH ile birlikte gelen ssh-keygen programi kullanilir.
RSA veya DSA tipinde seçim yapilmasi istenir, -t ile seçim yapilir.

$ssh-keygen -t rsa
Generating public/private rsa key pair.

Enter file in which to save the key (/home/huzeyfe/.ssh/id_rsa):
Created directory '/home/huzeyfe/.ssh'.
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /home/huzeyfe/.ssh/id_rsa.
Your public key has been saved in /home/huzeyfe/.ssh/id_rsa.pub.
The key fingerprint is:
58:af:43:fd:b9:ba:26:d3:38:21:45:5d:dd:ac:d4:de huzeyfe@home-fw.my.domain

ssh-keygen sonucu asagidaki dosyalar olusur.
$ ls ~/.ssh
id_rsa   id_rsa.pub

Açik anahtari sunucuya aktarmak için güvenli bir seçim yapilmalidir bunun için **scp** kullanilabilir.

Anahtar kullanarak bağlanılmak istenen hostta **~/.ssh/authorized_keys** dosyası oluşturularak , istemcinin id_rsa.pub dosyasinin içerigi aktarilmalidir. Aktarımlar tamamlandıktan sonra bağlanılmak istenilen SSH sunucuya anahtarlar aracılığı ile bağlanılabilir

$ssh test.enderunix.org
..
huzeyfe@test.enderunix$

Parolasiz ssh erisimleri

$chmod 600 ~/.ssh/id_rsa

Burası önemli, bu dosya başkaları tarafından okunabilir olursa ssh bağlanmayı reddecektir.

Örnek olarak bu dosyanın haklarını tum kullanıcılar tarafındanokunabilir hale getirip bağlanmaya çalışalım;

$ chmod 755 ~/.ssh/id_rsa
$ ssh test.enderunix.org
@@@
@   **WARNING: UNPROTECTED PRIVATE KEY FILE!**   @
@@@
Permissions 0755 for '/home/huzeyfe/.ssh/id_rsa' are too open.
It is recommended that your private key files are NOT accessible by others.

This private key will be ignored.
bad permissions: ignore key: /home/huzeyfe/.ssh/id_rsa
Enter passphrase for key '/home/huzeyfe/.ssh/id_rsa':

NOT:SCP ve sftp kullanırkende parolası(anahtarlar ile)erisim sağlanır.

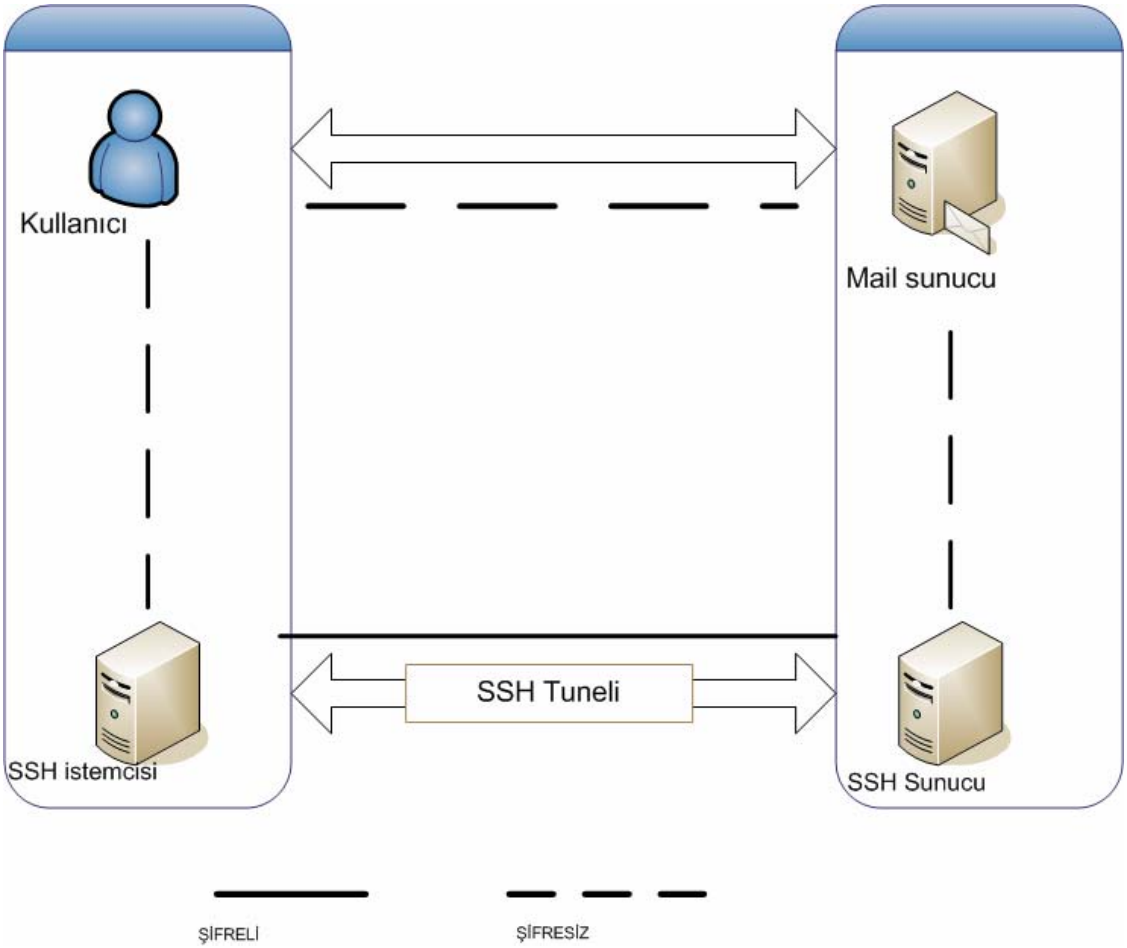
SSH Port Forwarding

Forwarding bir tür tünnelleme olarak düşünülebilir. Yapılan iş bir protokolu başka bir protokol aracılığı kullanmaktır. Genellikle yapılan, iletişimde sifreleme altyapısına sahip olmayan protokollerin(pop/imap/smtp vs) şifreleme kullanan bir protokol aracılığı ile güvenli bir şekilde kullanımıdır.

Yerel Yönlendirme(local forwarding)

Örnekleri POP3 protokolu üzerinden yürütecek olursak,

POP3 TCP/110 uzerinden çalışır. Istemci ile POP sunucu arasindaki iletişim clear text(sifrelenmemis) olarak gerceklesir. Uzaktaki POP sunucu ile istemci arasindaki trafigi SSH port forwarding kullanarak şifreleyebiliriz. Nasil mi? Yerel makinemizde 1024-65535 arası bir port seçelim, mesela 5000 olsun. POP istemcimizde kullandigimiz pop sunucuyu localhost ve portunu da 5000 olacak sekilde degistirelim. Sonra ssh ile asagıdaki yönlendirmeyi yapalım



$ssh –L5000:localhost:110 mail_sunucu(POP)

Bu komut ile yapılanlar;

Mail_sunucu makinesine gecerli bir hesap ile ssh baglantısı yapılmış olur ve localhost'un 5000. portu mail_sunucu makinesinin 110. portuna SSH üzerinden tunnelleme yapilmiş olur. Böylece POP istemcimizin ayarını localhost ve portunu 5000 olarak değiştirdiğimiz zaman mail sunucu ile aramızdaki trafik SSH araciligi ile şifrelenmiş olur.

Adım adım inceleyecek olursak ;

1. POP istemcisi yerel agdaki 5000. porta baglanıyor
2. Yerel agdaki ssh istemcisi 5000. porta gelen veriyi şifreliyor ve ssh aracılığı ile **mail_sunucu** makinesine gönderiyor.
3. **Mail_sunucu** makinesi ssh aracılığı ile gelen veriyi çözerek 110.porta iletiyor
4. **Mail_sunucu** makinesindeki POP sunucu isteği cevaplıyor ve paket aynı şekilde ters yönden istemciye ulaşıyor.

Burada dikkat edilmesi gereken yapılan forwarding işleminde sadece ssh istemcisinin çalıştığı makinenin tunneling yapabileceğidir. **Host:Localhost, port:5000 bind işlemi** olur. Yani forwarding işleminde ssh istemcisi localhost'u kullanır. Bu da demektir ki bu tunellemeyi sadece bu makine uzerindeki POP3 istemcisi kullanabilir.

OpenSSH bu kısıtlamayı kaldıracak bir seçenek içerir. -g ile dışarıdaki makinelerin de bu tunellemeyi
kullanabilmesine izin verir. Fakat bu secenek varsayılan kurulumda aktif olarak gelmez.

$ssh –g -L2001:localhost:110 Server

Remote forwarding

Local yönlendirmeden farkı işlemlerin ters olmasıdır yani tunelleme işlemini sunucu tarafında yapılıyor.

$ssh –R5000:localhost:110 istemci_makine

Bu komutla istemci makinin 5000. portu ile sunucu makine arasında bir tunnel oluşturulmuş olur. Istemci makinede gecerli bir ssh hesabının olması gerekir.

Ek notlar:

- Genllikle ssh izni ile, firewalldan izin verilmeyen diger portlar/protokollerin kullanimi icin kullanılır.
- ftp protokolu icin sadece control baglantısını yanı kullanıcı adı ve parola bilgilerini koruma altına alabilir. Arada gidip gelen veriyi tünellemez…(OpenSSH'daşimdilik)
- ssh ile sadece tcp tabanlı protokoller tünnellenebilir. Udp tabanlı protokoller ya da ip tabanlı olmayan protokoller tünellenemez. Bu da ssh'in gercek VPN baglantılar karşısında bir dezavantajıdır.

Dynamic port forwarding

OpenSSH Dynamic Port forwarding desteği ile bir nevi socks proxy vazigesi görür. Socks RFC-1928 ile tanımlanmış basit ama güçlü bir TCP protokoldür. Socks 5 ile UDP desteği de eklenmiştir.

$ssh –D 8080 ssh.enderunix.org

Bundan sonra kullandığım browserin proxy ayarlarından 8080 olacak şekilde yapılandırırsam herhangi bir kısıtlama olmaksızın ssh.enderunix.org makinesi aracılığı ile özgürce gezebilirim.

Scponly Kullanımı

Scponly sisteme login izni olmayan kullanıcılara sistemden dosya alışverişine izin veren bir program. /etc/passwd dosyasında kullanıcının shell'i /bin/scponly olarak ayarlanir.

Ayrıca SCPonly detaylı loglama özelliklerine sahiptir. Hangi IP'den hangi zamanda, hangi kullanici adi ile bağlanıldı gibi bilgileri syslog'a gönderebilir. Kullanıcıları kendi ev dizinlerine hapsederek diğer kullanıcıların dizin ve dosyalarını görmesini engeller.(Chroot)

Kurulum
http://www.sublimation.org/scponly/ adresinden son sürüm scponly paketi indirilerek klasik unix kurulum prosedürü uygulanır.

FreeBSD için ;

#cd /usr/ports/shells/scponly

#make install

SSH sunucu güvenliğini artırıcı önlemler

Son zamanlarda SSH protokolüne karşı yapılan saldırılar artmıştır. Eğer bir SSH sunucu çalıştırıyorsanız sistem loglarında sunucunuzdaki SSH servisine yapılan atakları görebilirsiniz. Bu ataklar genellikle SSH sunucudaki zayıf parolalarla korunmuş sistem hesaplarını ele geçirmek için yapılır. Bu tip ataklara birçok farklı şekilde önlem alınabilir. Kompleks çözümlere kaçmadan yapılacak birkaç basit ayarlar bu tip saldırıların %90'na karşı doğal koruma sağlanmış olur. Bu doğal korumalar;

- Kullandığınız OpenSSH sürümünün güncel olmasına özen gösterin.
- Çok özel bir gereksinimiz yoksa SSH sunucunun portunu 22 den farklı bir porta alın.Mesela doğum tarihiniz.
- Sisteme erişim yetkisi vermek istediğiniz kullanıcıları yapılandırma dosyasında belirtin.
- Sisteme root olarak erişim izni vermeyin
- Mümkünse sisteme parola ile girişi yasaklayıp erişimleri anahtarlara aracılığı ile yapmaya çalışın.
- SSH erişimini tüm internete açmayın. Varsa sabit bağlantınız sadece belirli Iplere erişim açın. Herhangi bir firewall kullanarak ya da hosts.allow/hosts.deny dosyaları kullanarak yapılabilir

Kaynaklar:

SSH, the Secure Shell, 2nd Edition

www.openssh.com

http://www.linux.com/article.pl?sid=05/02/02/1254222

http://www.linuxjournal.com/article/8600

http://www.openssh.com/press.html

Huzeyfe ÖNAL
huzeyfe@enderunix.org

EnderUNIX Yazılım Geliştirme Ekibi