

Oscar David Quiñones Franco

Cod: 2215738

1. FTP Seguro protegido por firewall

Para esta prueba se usaron dos maquinas: servidor y firewall.

Configuración vagrantfile:

```

# vagrantfile
# -*- mode: ruby -*-
# vi: set ft=ruby :

# The name of the VM (hostname: ip)
config.vm.define :cliente do |cliente|
  cliente.vm.box = "centos/stream"
  cliente.vm.network :private_network, ip: "192.168.50.2"
  cliente.vm.hostname = "cliente"
end

# The name of the VM (hostname: ip)
config.vm.define :servidor do |servidor|
  servidor.vm.box = "centos/stream"
  servidor.vm.network :private_network, ip: "192.168.50.3"
  servidor.vm.network :forwarded_port, guest: 80, host: 5567
  servidor.vm.network :forwarded_port, guest: 443, host: 5568
  servidor.vm.hostname = "servidor"
end

# The name of the VM (hostname: ip)
config.vm.define :servidor2 do |servidor2|
  servidor2.vm.box = "centos/stream"
  servidor2.vm.network :private_network, ip: "192.168.50.4"
  servidor2.vm.hostname = "servidor2"
end

# The name of the VM (hostname: ip)
config.vm.define :clienteuf do |clienteuf|
  clienteuf.vm.box = "centos/stream"
  clienteuf.vm.network :private_network, ip: "209.191.100.2"
  clienteuf.vm.hostname = "clienteuf"
end

# The name of the VM (hostname: ip)
config.vm.define :firewall do |firewall|
  firewall.vm.box = "centos/stream"
  firewall.vm.network :private_network, ip: "209.191.100.3"
  firewall.vm.network :private_network, ip: "192.168.100.3"
  firewall.vm.network :forwarded_port, guest: 80, host: 5571
  firewall.vm.network :forwarded_port, guest: 443, host: 5572
  firewall.vm.hostname = "firewall"
end

# The name of the VM (hostname: ip)
config.vm.define :servidor3 do |servidor3|
  servidor3.vm.box = "centos/stream"
  servidor3.vm.network :private_network, ip: "192.168.100.4"
  servidor3.vm.network :forwarded_port, guest: 80, host: 5569
  servidor3.vm.network :forwarded_port, guest: 443, host: 5570
  servidor3.vm.hostname = "servidor3"
end
end

```

Configuración firewall lista dmz:

```
root@firewall:/usr/lib/firewall/services
[root@firewall services]# firewall-cmd --zone=dmz --list-all
dmz (active)
target: default
icmp-block-inversion: no
interfaces: eth0 eth1 eth2
sources:
services: ftp http https ssh
ports: 21/tcp 20/tcp 1024-50000/tcp
protocols:
masquerade: yes
forward-ports:
  port=21:proto=tcp:toport=21:toaddr=192.168.50.3
  port=1024-50000:proto=tcp:toport=1024-50000:toaddr=192.168.50.3
source-ports:
icmp-blocks: echo-request
rich rules:
[root@firewall services]#
```

Configuración vsftpd.conf firewall:

```
# root@firewall:/usr/lib/firewall/services
# predicted this attack and has always been safe, reporting the size of the raw file.
# ADULT mangling is a horrible feature of the protocol.
#rcil_enable_enable=YES
#rcil_download_enable=YES

# You may fully customize the login banner string:
ftpd_banner=bienvenido al servicio ftp de oscar quifones en el firewall..

# You may specify a file of disallowed anonymous e-mail addresses. Apparently
# useful for combatting certain DoS attacks.
#deny_email_enable=YES
#(default follows)
#deny_email_file=/etc/vsftpd/daemon.deny_mails

# You may specify an explicit list of local users to chroot() to their home
# directory. If chroot_local_user is YES, then this list becomes a list of
# users to NOT chroot().
#(warning) chroot'ing can be very dangerous. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# /chroot.
#chroot_local_user=YES
#chroot_list_enable=YES
# (default follows)
#chroot_list_file=/etc/vsftpd/chroot_list
#allow_writeable_chroot=YES

# You may activate the "-B" option to the builtin ls. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites, however some broken FTP clients such as "vsftp" and "lvsftp" assume
# the presence of the "-B" option, so there is a strong case for enabling it:
#ls_enable_enable=YES

# When "listen" directive is enabled, vsftpd runs in standalone mode and
# listens on IPv4 sockets. This directive cannot be used in conjunction
# with the "listen_ipv6" directive.
listen=YES

# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv4 "any" address (:) will accept connections from both IPv4
# and IPv6 clients. It is not necessary to listen on "both" IPv4 and IPv6
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of vsftpd with our configuration
# files.
#(warning) If you only have one instance of the listen options is commented ()
listen_ipv6=YES

#am_service_name=vsftpd
#userlist_enable=YES
#(etc/vsftpd/vsftpd.conf" 1291, 5149C
```

Vsftpd.conf servidor:

```
root@kali:~# cat /etc/ssh/sshd_config

# You may wish to enable the login banner string.
# login_banner=BIENVENIDO AL SERVICIO FTP DE OSCA QUILMES.

# You may specify a file of disallowed usernames or e-mail addresses. Apparently
# useful for preventing certain host attacks.
# DenyUsers some_mail:me@xyz.com
# (default follows)
# DenyUsers mail:/etc/ssh/sshdknown_emails

# You may optionally use explicit list of local users to chroot() to their home
# directory. If DenyUsers is YES, then this list becomes a list of
# users to NOT chroot().
# ChrootUsers some_user
# (Starting) chroot log can be very noisy. If using chroot, make sure that
# the user does not have write access to the top level directory within the
# chroot.
# ChrootLocalUsers YES
# ChrootList enable=YES
# (default follows)
# ChrootList /etc/ssh/sshd/chroot_list
# Allow to libexec chroot YES
# Configuración del sitio seguro
# SshCerts /etc/ssh/pki/ssh/certs/cert
# Private key file /etc/pki/ssh/private/ssh_key
# Ssh enable=YES

# You may optionally use "X" option to the multiple it. This is disabled by
# default to avoid remote users being able to cause excessive I/O on large
# sites. However, some broken FTP clients such as "ncftp" and "nirx" assume
# the presence of the "X" option, so there is a strong case for enabling it.
# X yes/no/ask/YES

# Want "listen" directive is enabled, sshd will try to standalone mode and
# listens on IPad sockets. This directive cannot be used in conjunction
# with the listen,ipad directive.
listen:0

# This directive enables listening on IPv6 sockets. By default, listening
# on the IPv6 "any" address (:::) will accept connections from both IPv6
# and IPv4 clients. It is not necessary to listen on "listen" IPv6 and IPv4
# sockets. If you want that (perhaps because you want to listen on specific
# addresses) then you must run two copies of sshd with the configuration
# files.
# ListenIPv6 yes/no/ask/YES

# ListenIPv4 yes/no/ask/YES

# Listen service name=vsftpd
# ListenList enable=YES

root@kali:~# cat /etc/ssh/sshd.conf | grep listen | grep -v '#'
```

Certificado FTP seguro:

Detalles del certificado

Certificado

Vista previa

Huella digital (SHA-256): 61ee4b20b5f8ab5b5eeef3542c53c60410:
e5454fb9b05e5e5ed236c20b7a146038f3

Huella digital (SHA-1): d92b85d58ed18980c7f7033150e2496016179280

Período de validez: De 13/03/2024 10:17:23 p. m. a 12/03/2024 10:17:23 p. m.

Asunto

Nombre común: servidor

Organización: universidad autonoma de occidente

Unidad: automantica

País: CO

Estado o provincia: valle del cauca

Localidad: cali

Correo electrónico: oscar.quinones@uao.edu.co

Editor

Igual que el asunto, el certificado está autofirmado

Detalles

De serie: 7562e7e6a477ba00ed965f4331e538df49b418c8

Algoritmo de clave pública: RSA con 2048 bits

Algoritmo de firma: RSA-SHA256

Detalles de la sesión

Sitio: 209.191.100.321

Protocolo: TLS1.3

Cifrado: AES-256-GCM

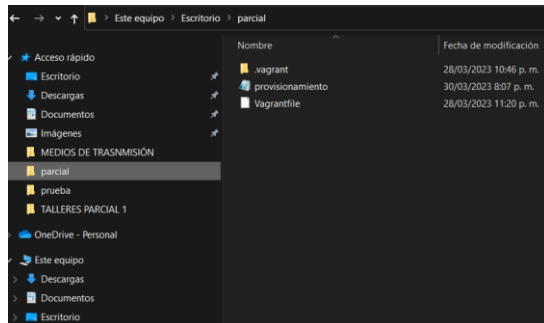
Intercambio de clave: ECDHE-SECP256R1-RSA-PSS-RSAE-SHA384 Mac: AEAD

Aceptar

2. DNS over TLS

Para esta prueba se uso una maquina llamada dnstest

Configuración vagrantfile:



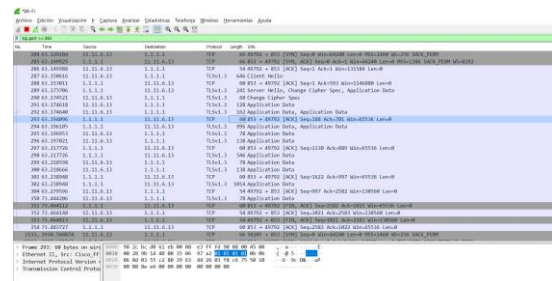
```
Vagrantfile: Bloc de notas
Archivo Edición Formato Ver Ayuda
# -*- mode: ruby -*-
# vi: set ft=ruby :
Vagrant.configure("2") do |config|
  if Vagrant.has_plugin? "vagrant-vbguest"
    config.vbguest.no_install = true
    config.vbguest.auto_update = false
    config.vbguest.no_remote = true
  end
  config.vm.define :dnstest do |dnstest|
    dnstest.vm.box = "bento/ubuntu-20.04"
    dnstest.vm.network :private_network, ip: "192.168.20.2"
    dnstest.vm.hostname = "dnstest"
    config.vm.provision :shell, path: "provisionamiento.sh", run: "always"
  end
end
```

Comando resolvf a Google.com

```
vagrant@dnstest: ~
vagrant@dnstest:~$ vagrant@dnstest:~$ resolvectl query google.com
google.com: 172.217.173.46 -- link: eth0

-- Information acquired via protocol DNS in 193.8ms.
-- Data is authenticated: no
vagrant@dnstest:~$
```

Trama wireshark:



Los apartados que se registran al realizar una trama por filtrado tcp.port == 853 utilizando un analizador de protocolos como Wireshark son los siguientes:

No.: Número de la trama en la secuencia de captura.

Time: Tiempo transcurrido desde el inicio de la captura hasta que se capturó esta trama.

Source: Dirección IP del origen de la trama.

Destination: Dirección IP del destino de la trama.

Protocol: Protocolo utilizado para la trama (en este caso, TCP).

Length: Longitud en bytes de la trama capturada.

Info: Información adicional sobre la trama, como el puerto de origen y destino, la secuencia y el número de reconocimiento, entre otros detalles específicos del protocolo TCP.