Securing Remote Access to our infrastructure

tgndevs

Aleix Ramírez - @aleixrmbaena
December 2018

Introduction

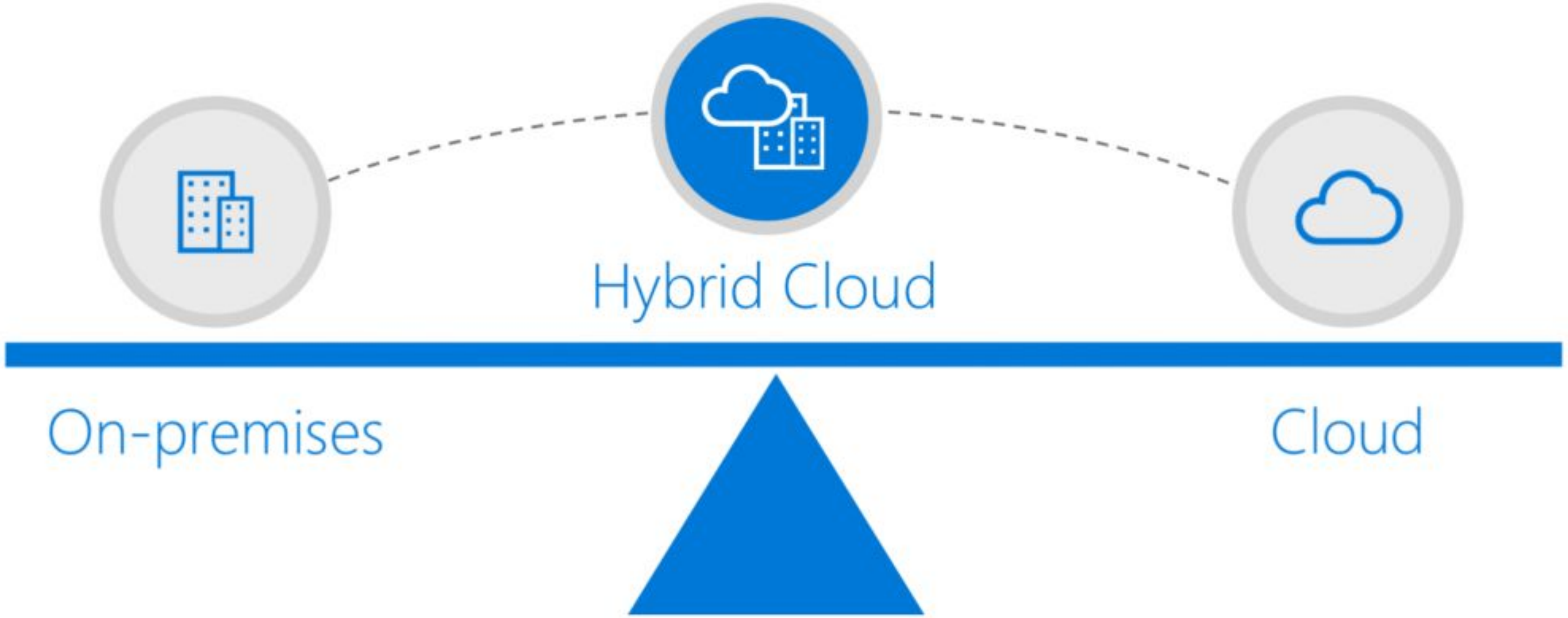
Why we need to allow remote access to our infrastructure?

- Ubiquity (e.g. work from home)
- Lack of host interactive console in some cloud providers
- Expose services (e.g. databases)

Why we need to secure the external access to our infrastructure?

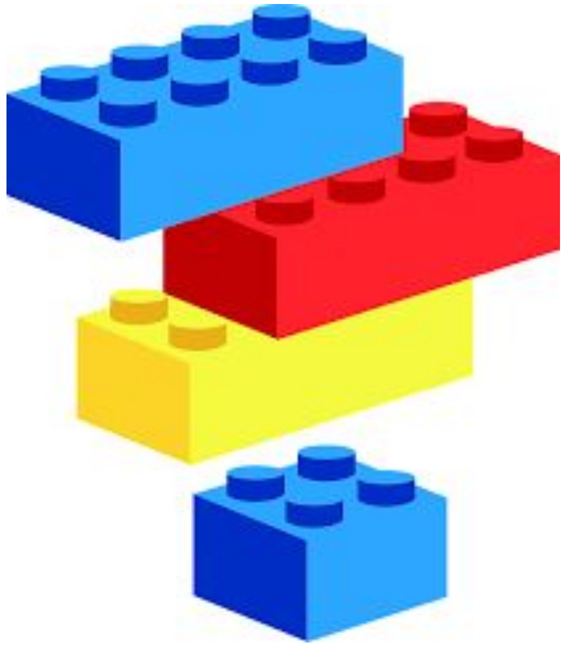
- Once some access point of your infrastructure is **exposed** to internet, it will **probably** be the objective of an **undetermined** number of **periodic** attacks

On-premises vs. Cloud



Infrastructure Security building blocks

- Firewall, Intrusion Prevention Systems (IPS)
- Network **Access Control Lists** (ACLs)
- **Authentication**
- **Authorization**
- **Encryption**
- Cloud provider reliability
- **User**
- … etc



Security Lemma

"Security is a chain; it's only as secure as ***the weakest link****."*

Authentication

- User repository (LDAP, Active Directory)
- Authentication Services / Protocols
  - Kerberos - GSSAPI
  - CAS
  - Active Directory Federation Services - SAML
- Authentication method
  - Password
  - **Public Key Authentication**
  - **One-time Password (OTP/HOTP)**
  - Physical tokens
  - Biometry
  - **Multi-factor authentication**

PAM - Pluggable Authentication Modules

- Linux flexible authentication framework
- When a program needs to authenticate a user, PAM provides a library containing the functions for the proper authentication scheme
- Changing authentication schemes can be done by simply editing a configuration file
- Enables the use of biometry, physical tokens, OTP, etc
- Compatible with login, ssh, among others

PAM + 2FA example

There are a lot of PAM modules available, e.g. this one from CERN that enables 2FA with yubikey physical tokens, OTP like google authenticator or via SMS

https://cern-cert.github.io/pam_2fa

Authorization

Access **policies** definition

- Which resources can access the authenticated user?
- Which actions over that resource can do the authenticated user?

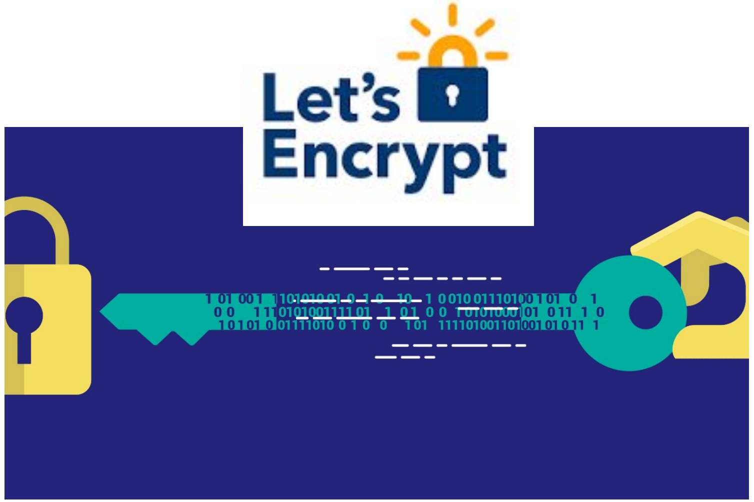
Principle of least privilege

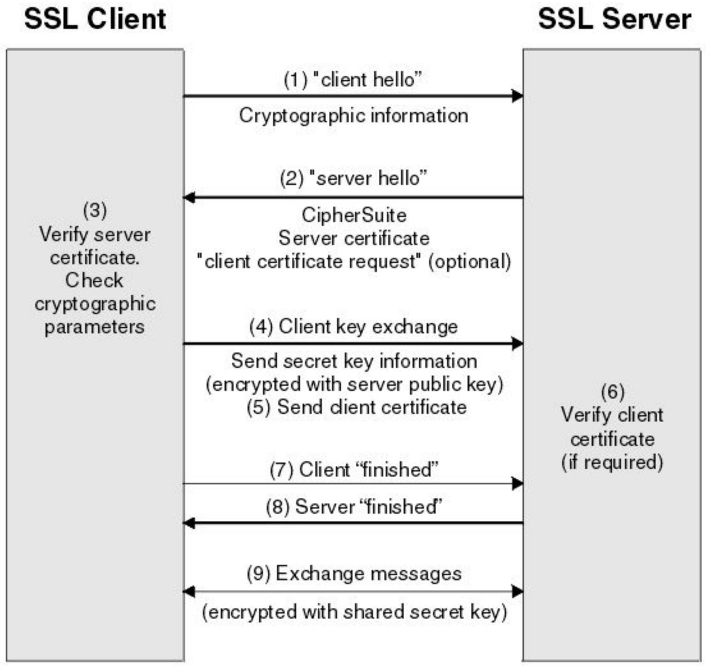
- A user must be able to access only to the resources that are necessary for its legitimate purpose

Data Encryption

Data in transit -> channel encryption

- Secure Sockets Layer (SSL/TLS)
- IPSec




SSL Client
SSL Server
(1) "client hello"
Cryptographic information
(2) "server hello"
CipherSuite
Server certificate
"client certificate request" (optional)
(3) Verify server certificate. Check cryptographic parameters
(4) Client key exchange
Send secret key information (encrypted with server public key)
(5) Send client certificate
(6) Verify client certificate (if required)
(7) Client "finished"
(8) Server "finished"
(9) Exchange messages
(encrypted with shared secret key)

Remote access methods

Remote access to networks

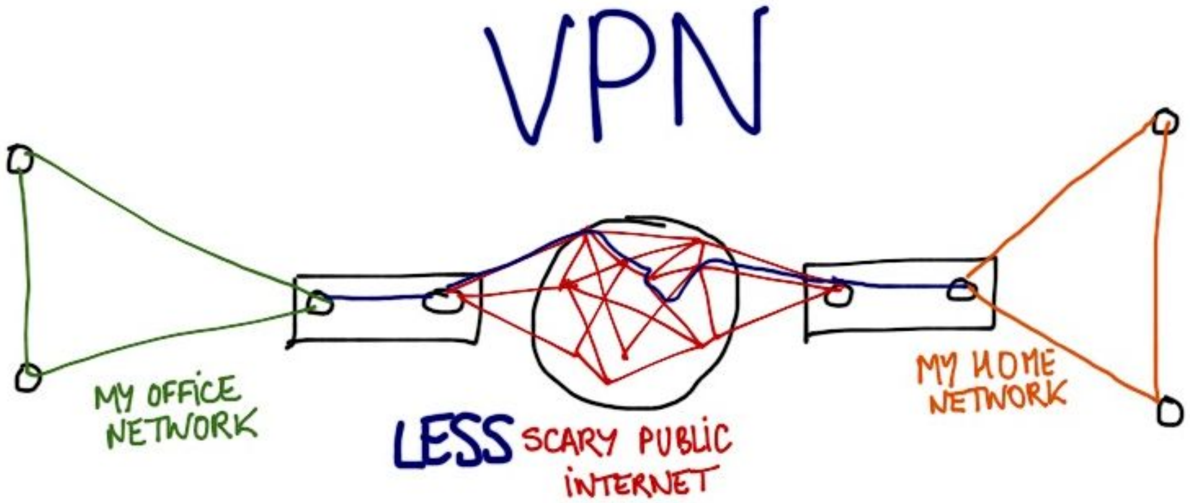
- Virtual private networks (VPN)

Remote access to hosts

- Terminal emulation: SSH, telnet
- Remote Desktop software: RDP
- Desktop sharing: VNC, TeamViewer
- Virtual Desktop Infrastructure (VDI)
- Application virtualization (Citrix)

VPN

- **IPSec**
- **SSL**
- PPTP*
- MPLS



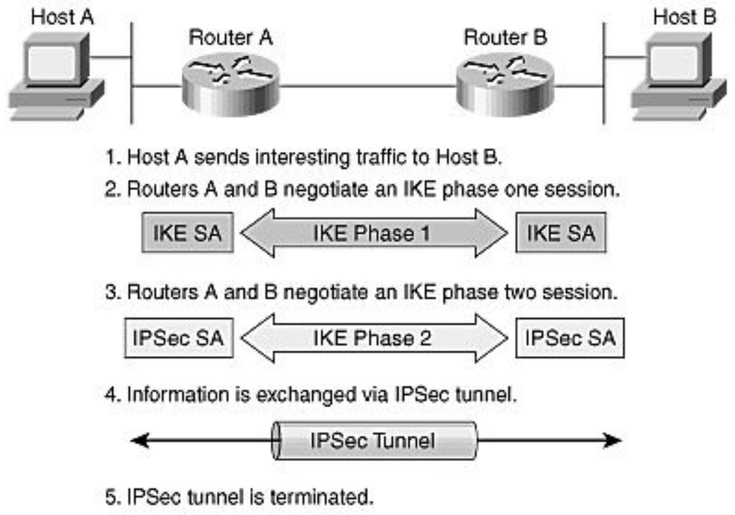
VPN - IPSec vs SSL

IPSec

- Network layer
- Full access to resources
- Less packet overhead (lower latency)

SSL

- Application - Transport layer
- Web application proxy - web browser
- Simple configuration
- Only specific port opened (easier to secure)


Host A
Router A
Router B
Host B
1. Host A sends interesting traffic to Host B.
2. Routers A and B negotiate an IKE phase one session.
IKE SA
IKE Phase 1
IKE SA
3. Routers A and B negotiate an IKE phase two session.
IPSec SA
IKE Phase 2
IPSec SA
4. Information is exchanged via IPSec tunnel.
IPSec Tunnel
5. IPSec tunnel is terminated.

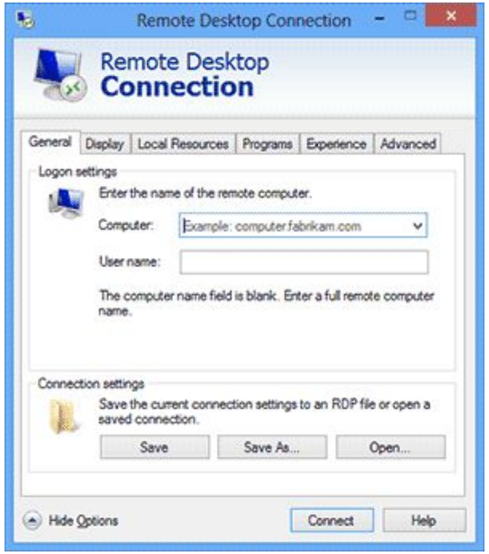
OpenVPN

- Full-featured SSL VPN
- Implements OSI layer 2 or 3 secure network extension
- Open Source (community edition)
- Multiplexes SSL tunnels on a single TCP/UDP port
- Good alternative to IPSec
- Integration with router firmwares (pfSense, DD-WRT…)
- Allows Access Control Policies configuration

https://openvpn.net/community-resources/how-to/

Remote Desktop Protocol - RDP

- Microsoft proprietary protocol
- Graphical interface to connect Windows Machines remotely
  - Also exists RDP server implementation for Linux & Mac
- SSL support
  - Prevents Man-in-the-middle attacks
  - Avoid self-signed certificates
- Connection can be tunneled by SSH or IPSec



Logon settings

Enter the name of the remote computer.

Computer: Example: computer.fabrikam.com

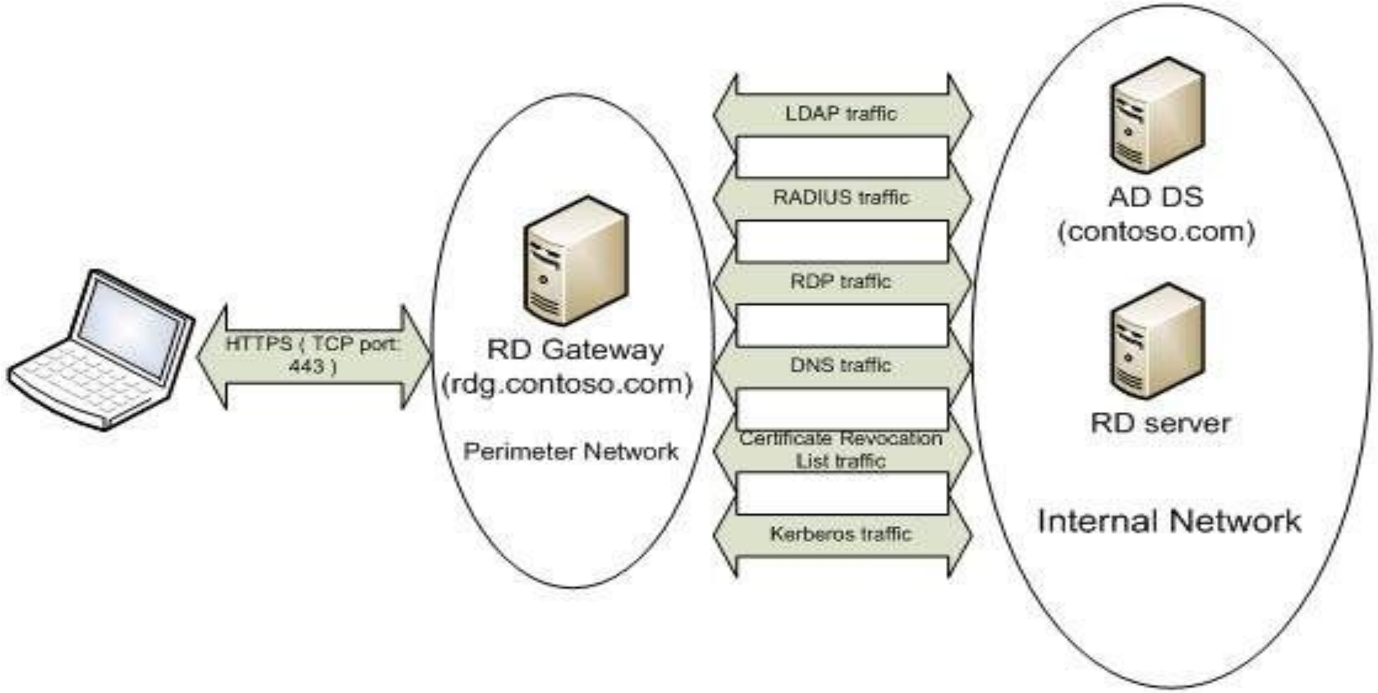
User name:

The computer name field is blank. Enter a full remote computer name.

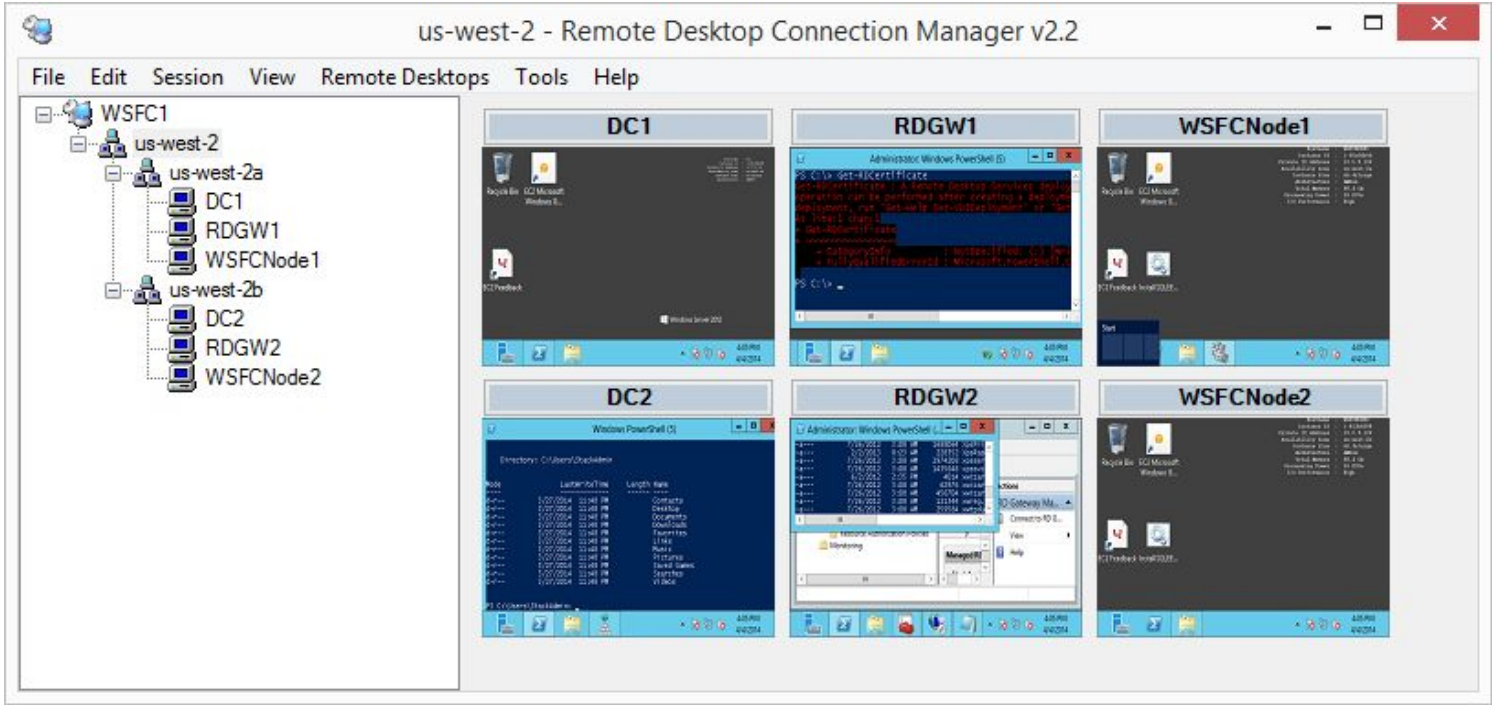
Connection settings

Save the current connection settings to an RDP file or open a saved connection.

Remote Desktop Gateway



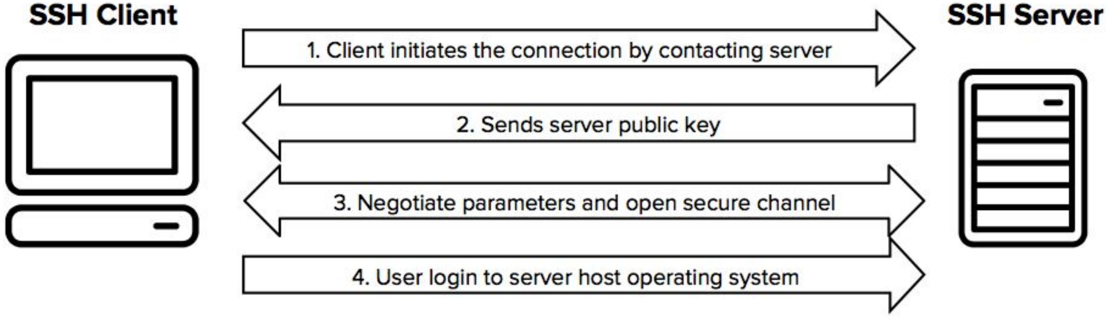
Remote Desktop Gateway



```
PS C:\> Get-RDCertificate
...
PS C:\>
```

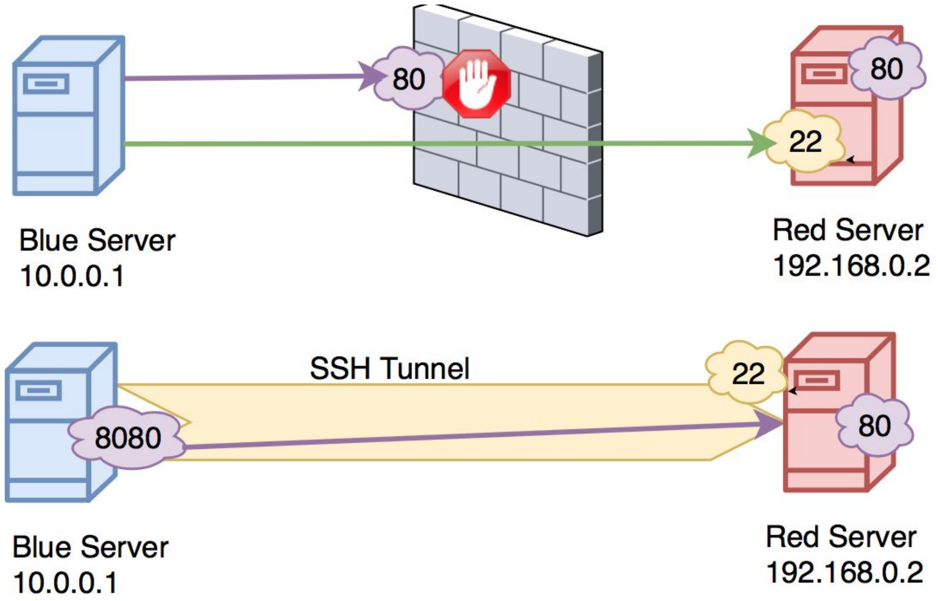
Secure SHell (SSH)

- Cryptographic network protocol
- Any network service can be protected with SSH - works at application level
- Supports public-key cryptography for authentication
- Supports tunneling, port-forwarding and X11 connections
- Client-server model
- autossh


SSH Client
SSH Server
1. Client initiates the connection by contacting server
2. Sends server public key
3. Negotiate parameters and open secure channel
4. User login to server host operating system

SSH Tunneling - Local Port forwarding

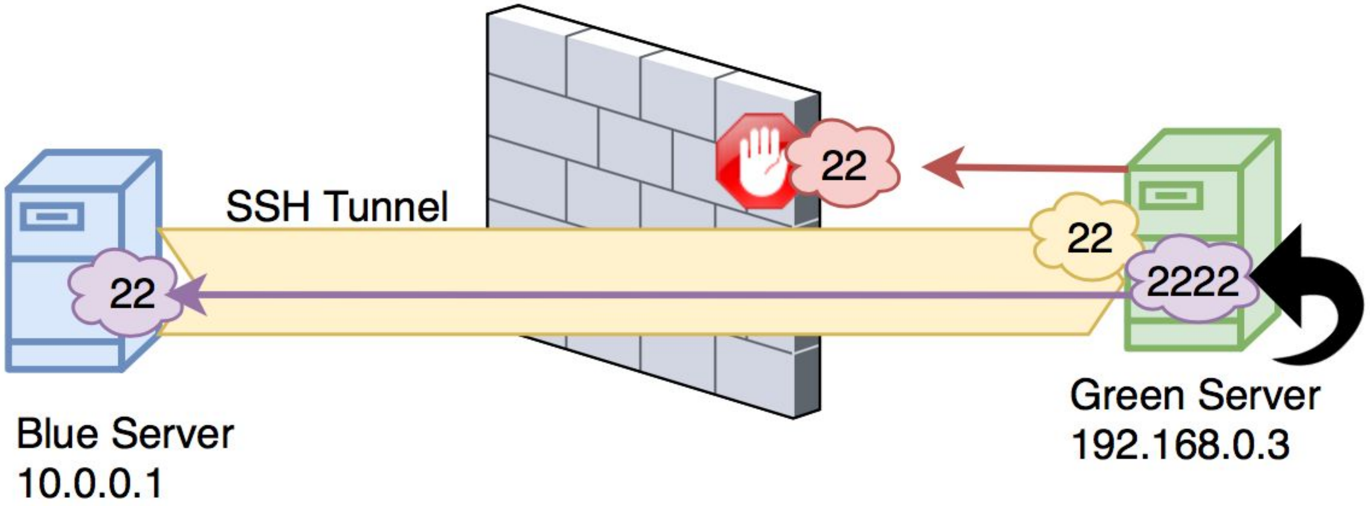
- Accessing a denied service on a host reachable by SSH
- blueuser@blue.server$ ssh **-L** 8080:red.server:80 reduser@red.server



https://www.tunnelsup.com/how-to-create-ssh-tunnels

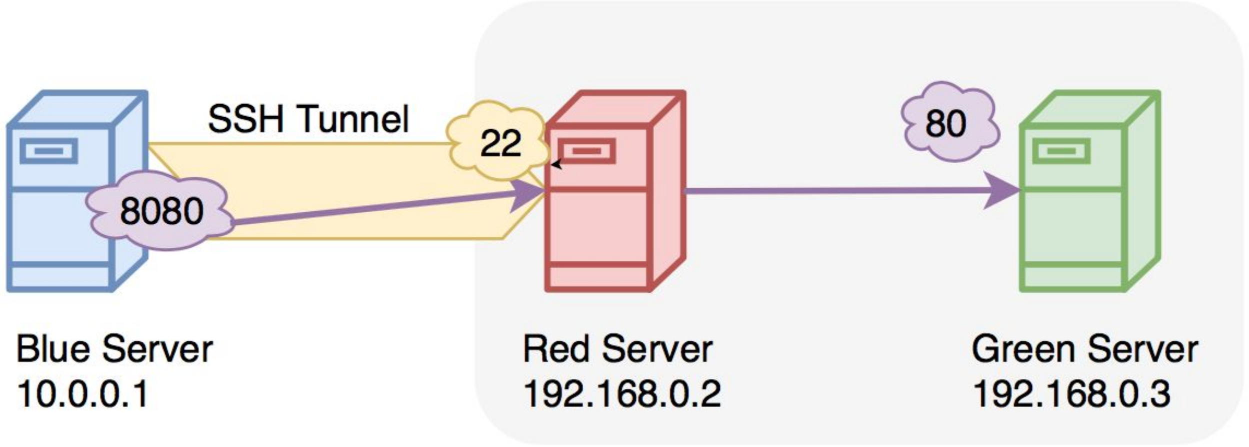
SSH Tunneling - Remote Port forwarding

- Reverse tunnel: the inaccessible host initiates the tunnel, then forwards our remote port to its local port
- blueuser@blue.server$ ssh **-R** 2222:localhost:22 greenuser@green.server



SSH Proxy to remote server

- Blue host cannot reach green server, but can reach red server by ssh
- Red server can reach green server
- blueuser@blue.server$ ssh **-L** 8080:green.server:80 reduser@red.server



SSH Authentication best practices

SSH public key login

- $ ssh-keygen
    - generates 2 keys -> 1 public (id_rsa.pub) & one private (id_rsa) -> $HOME/.ssh
    - **Never share the private key and store it in a safe place!**
    - Copy the public key in the .ssh/authorized_keys file of the destination host

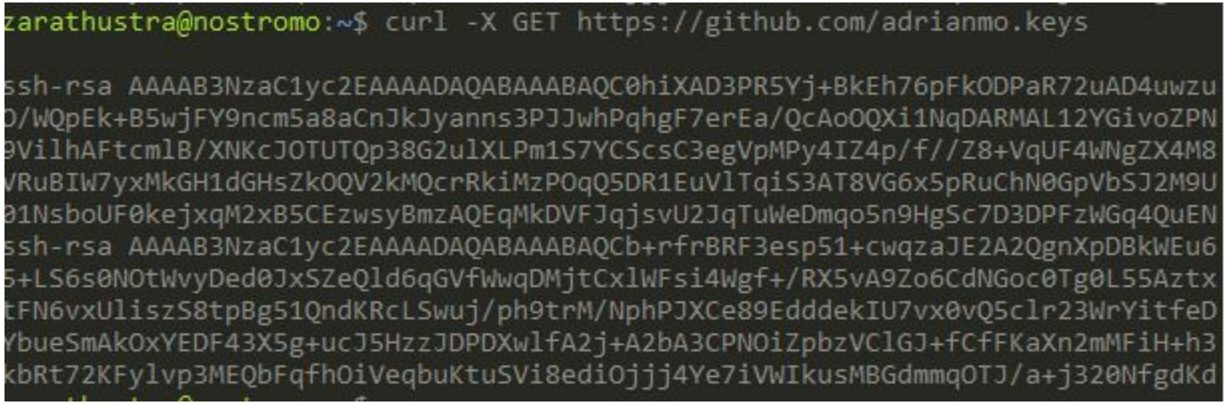
SSH Agent

- $ eval "$(ssh-agent -s)" && ssh-add ~/.ssh/id_rsa
    - Unlocks and stores keys in memory (ask for passphrase just once)
    - **Agent forwarding**: Allows the usage of local SSH keys through remote servers
        - $ ssh -A user@remoteHost

SSH Authentication best practices

- Public SSH Key distribution automation
  - e.g. with Ansible playbooks: authorized_key module
    - https://docs.ansible.com/ansible/latest/modules/authorized_key_module.html

- Public Key repository
  - GITHUB/GITLAB! :D -> curl -X GET https://github.com/<username>.keys



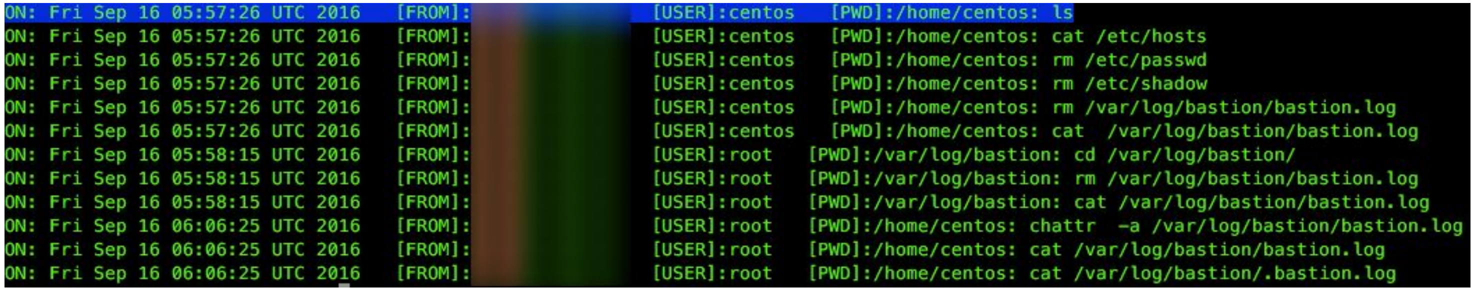
```
zarathustra@nostromo:~$ curl -X GET https://github.com/adrianmo.keys

ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQC0hiXAD3PR5Yj+BkEh76pFkODPaR72uAD4uwzu
O/WQpEk+B5wjFY9ncm5a8aCnJkJyanns3PJJwhPqhgF7erEa/QcAoOQXi1NqDARMAL12YGivoZPN
9VilhAFtcmlB/XNKcJOTUTQp38G2ulXLPm1S7YCScsC3egVpMPy4IZ4p/f//Z8+VqUF4WNgZX4M8
VRuBIW7yxMkGH1dGHsZkOQV2kMQcrRkiMzPOqQ5DR1EuVlTqiS3AT8VG6x5pRuChN0GpVbSJ2M9U
01NsboUF0kejxqM2xB5CEzwsyBmzAQEqMkDVFJqjsvU2JqTuWeDmqo5n9HgSc7D3DPFzWGq4QuEN
ssh-rsa AAAAB3NzaC1yc2EAAAADAQABAAABAQCb+rfrBRF3esp51+cwqzaJE2A2QgnXpDBkWEu6
5+LS6s0NOtWvyDed0JxSZeQld6qGVfWwqDMjtCxlWFsi4Wgf+/RX5vA9Zo6CdNGoc0Tg0L55Aztx
tFN6vxUliszS8tpBg51QndKRcLSwuj/ph9trM/NphPJXCe89EdddekIU7vx0vQ5clr23WrYitfeD
YbueSmAkOxYEDF43X5g+ucJ5HzzJDPDXwlfA2j+A2bA3CPNOiZpbzVClGJ+fCfFKaXn2mMFiH+h3
kbRt72KFylvp3MEQbFqfhOiVeqbuKtuSVi8ediOjjj4Ye7iVWIkusMBGdmmqOTJ/a+j320NfgdKd
```

Bastion host

- Hardened server designed to withstand attacks
- Provides access to a private network from an external network
- Must minimize chances of penetration (only one exposed port)
- Acts as a ‘jump’ server, allowing to access servers through RDP or SSH

Bastion host hardening

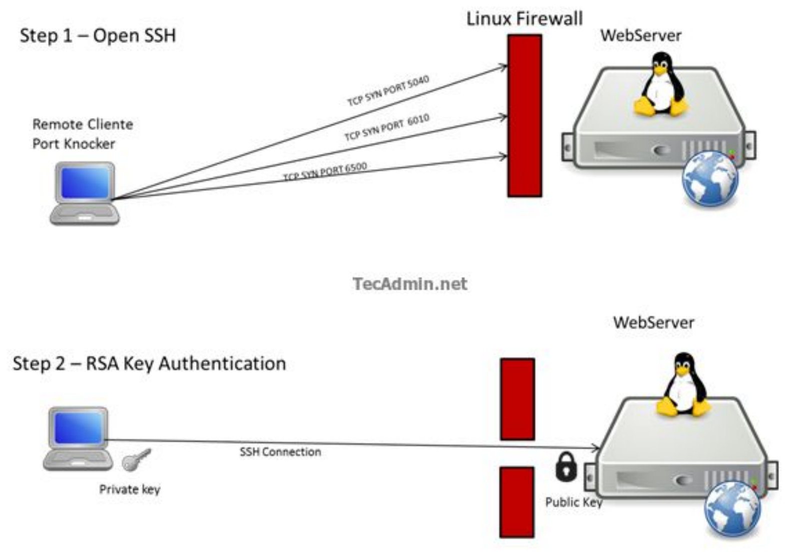
- Change default port for SSH or encapsulate RDP via SSL (RD gateway)
- Disable Root access and Password Authentication on SSH
  - Use public keys & 2FA, whitelisting allowed users
- Uninstall or disable unnecessary services
- Keep your OS updated! -> apt-get update && apt-get upgrade // cron-apt
- Log sessions & executed commands (OSSEC, cloudwatch,etc)

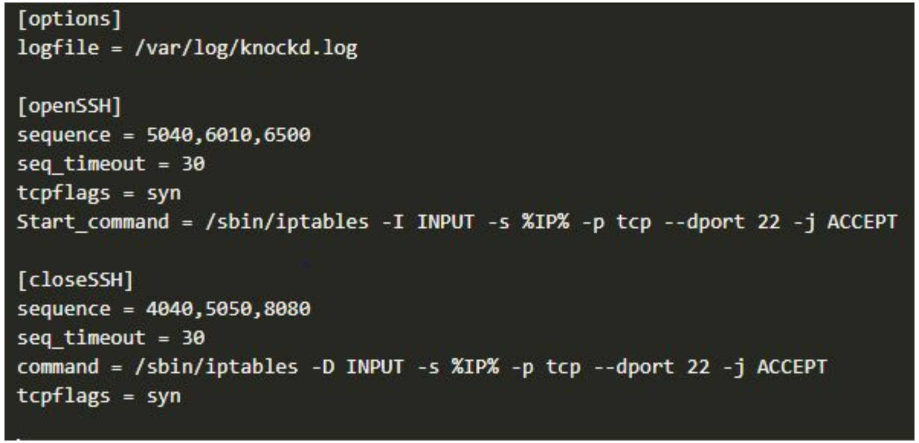


```
ON: Fri Sep 16 05:57:26 UTC 2016    [FROM]: [redacted]  [USER]:centos    [PWD]:/home/centos: ls
ON: Fri Sep 16 05:57:26 UTC 2016    [FROM]:             [USER]:centos    [PWD]:/home/centos: cat /etc/hosts
ON: Fri Sep 16 05:57:26 UTC 2016    [FROM]:             [USER]:centos    [PWD]:/home/centos: rm /etc/passwd
ON: Fri Sep 16 05:57:26 UTC 2016    [FROM]:             [USER]:centos    [PWD]:/home/centos: rm /etc/shadow
ON: Fri Sep 16 05:57:26 UTC 2016    [FROM]:             [USER]:centos    [PWD]:/home/centos: rm /var/log/bastion/bastion.log
ON: Fri Sep 16 05:57:26 UTC 2016    [FROM]:             [USER]:centos    [PWD]:/home/centos: cat  /var/log/bastion/bastion.log
ON: Fri Sep 16 05:58:15 UTC 2016    [FROM]:             [USER]:root    [PWD]:/var/log/bastion: cd /var/log/bastion/
ON: Fri Sep 16 05:58:15 UTC 2016    [FROM]:             [USER]:root    [PWD]:/var/log/bastion: rm /var/log/bastion/bastion.log
ON: Fri Sep 16 05:58:15 UTC 2016    [FROM]:             [USER]:root    [PWD]:/var/log/bastion: cat /var/log/bastion/bastion.log
ON: Fri Sep 16 06:06:25 UTC 2016    [FROM]:             [USER]:root    [PWD]:/home/centos: chattr  -a /var/log/bastion/bastion.log
ON: Fri Sep 16 06:06:25 UTC 2016    [FROM]:             [USER]:root    [PWD]:/home/centos: cat /var/log/bastion/bastion.log
ON: Fri Sep 16 06:06:25 UTC 2016    [FROM]:             [USER]:root    [PWD]:/home/centos: cat /var/log/bastion/.bastion.log
```

Bastion host hardening

- Port knocking technique -> Prevents scan attacks

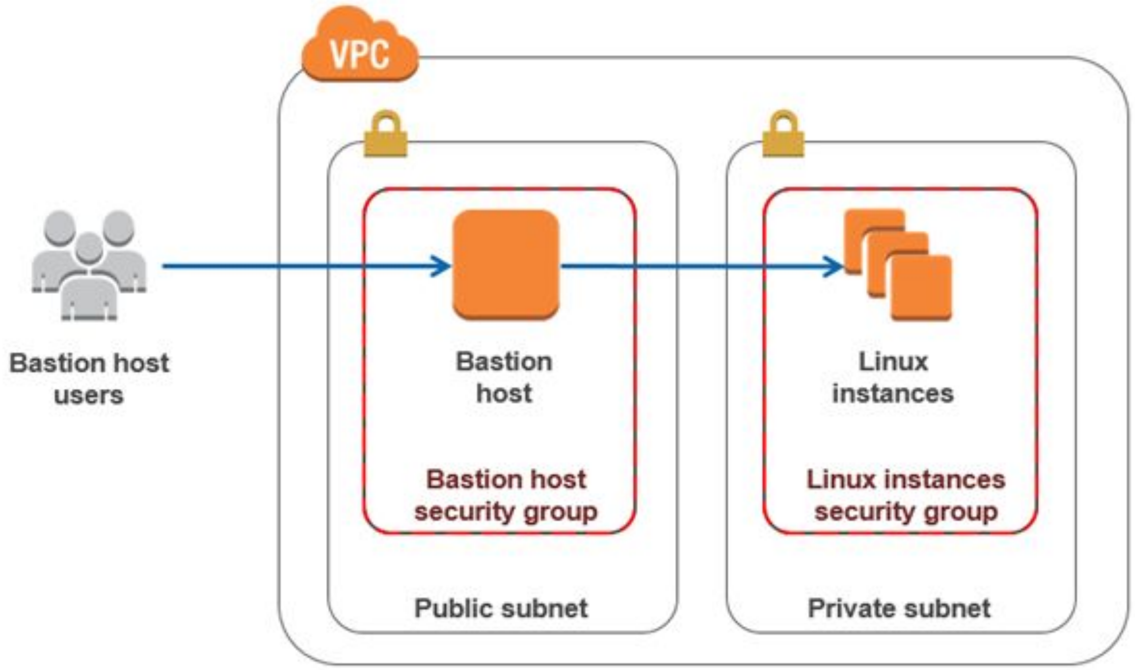




```
[options]
logfile = /var/log/knockd.log

[openSSH]
sequence = 5040,6010,6500
seq_timeout = 30
tcpflags = syn
Start_command = /sbin/iptables -I INPUT -s %IP% -p tcp --dport 22 -j ACCEPT

[closeSSH]
sequence = 4040,5050,8080
seq_timeout = 30
command = /sbin/iptables -D INPUT -s %IP% -p tcp --dport 22 -j ACCEPT
tcpflags = syn
```

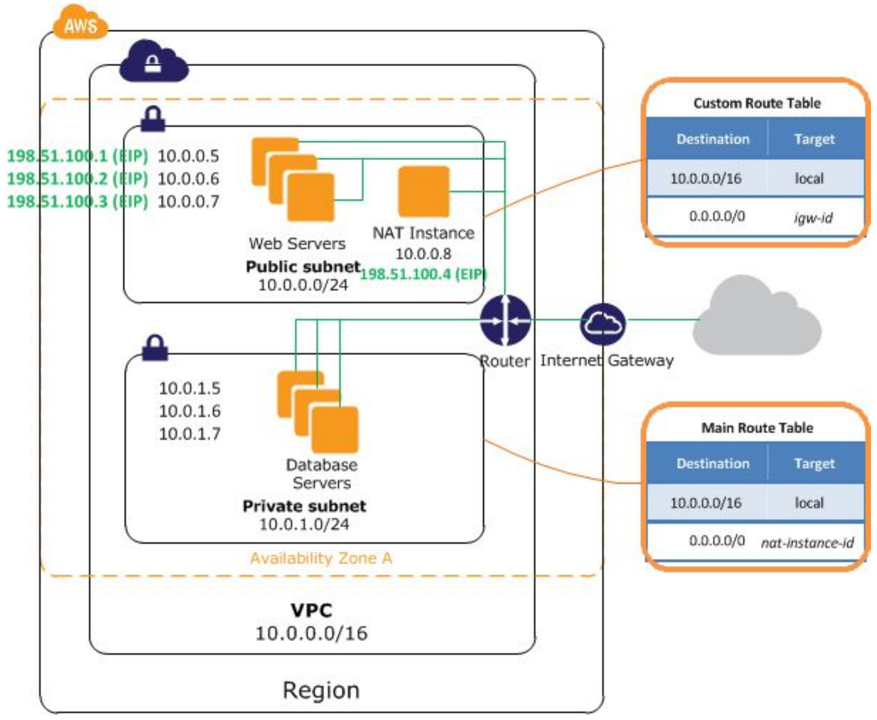
https://bosscownet.blogspot.com/2014/09/secure-ssh-menggunakan-port-knocking.html

AWS remote access architecture - Bastion host



https://docs.aws.amazon.com/quickstart/latest/linux-bastion

Outbound traffic forwarding through NAT Instance



Custom Route Table

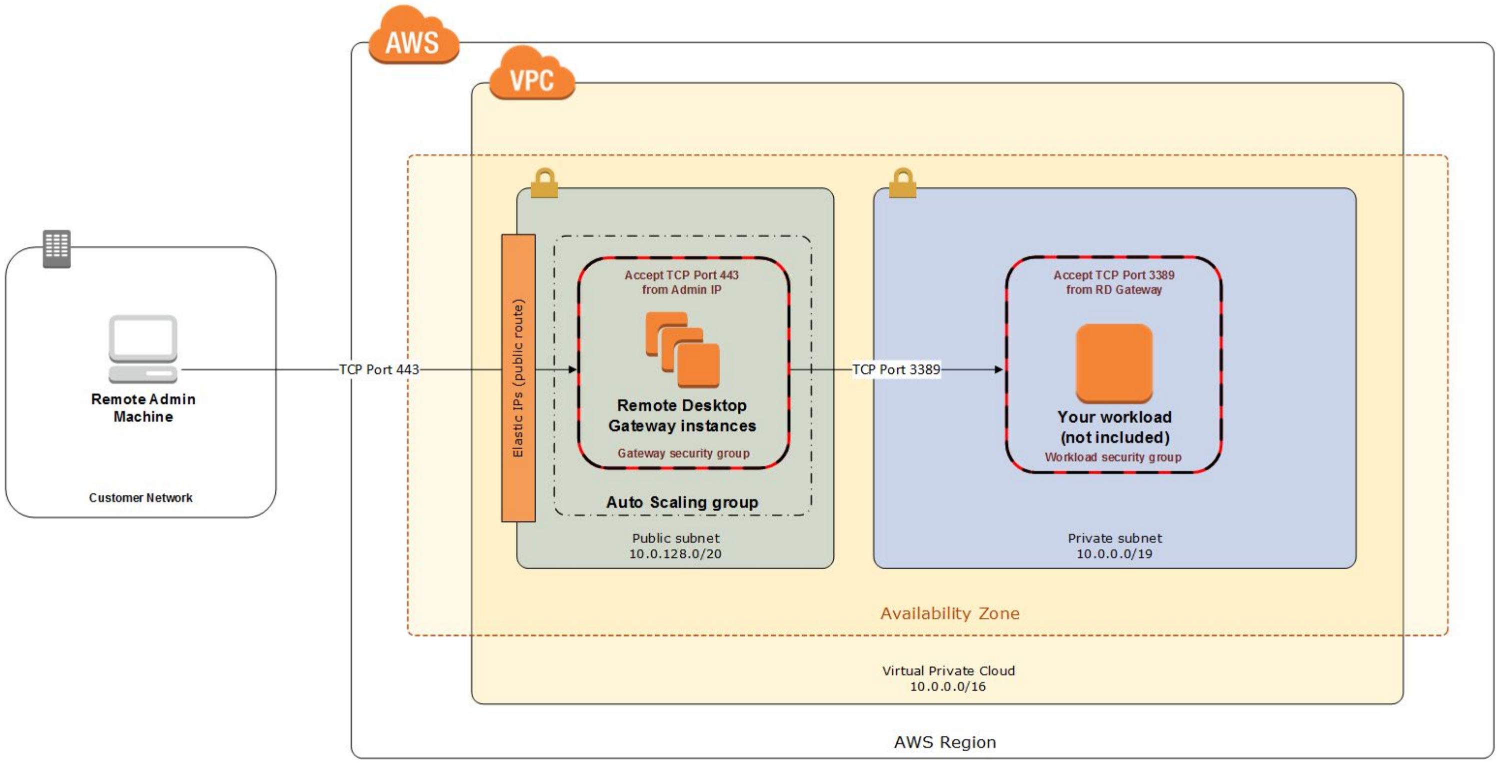
Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*igw-id*

Main Route Table

Destination	Target
10.0.0.0/16	local
0.0.0.0/0	*nat-instance-id*

https://docs.aws.amazon.com/vpc/latest/userguide/VPC_NAT_Instance.html

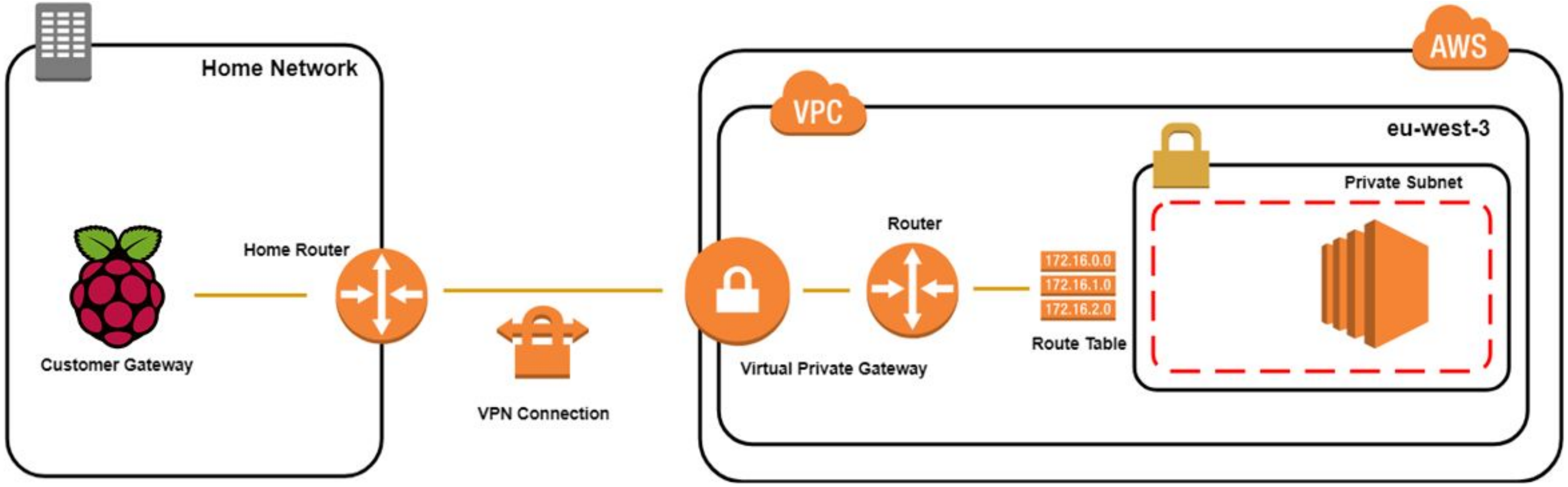
AWS remote access architecture - RD Gateway


AWS
VPC
Remote Admin Machine
Customer Network
TCP Port 443
Elastic IPs (public route)
Accept TCP Port 443 from Admin IP
Remote Desktop Gateway instances
Gateway security group
Auto Scaling group
Public subnet 10.0.128.0/20
TCP Port 3389
Accept TCP Port 3389 from RD Gateway
Your workload (not included)
Workload security group
Private subnet 10.0.0.0/19
Availability Zone
Virtual Private Cloud 10.0.0.0/16
AWS Region

https://docs.aws.amazon.com/quickstart/latest/rd-gateway

AWS remote access architecture - Home 2 AWS VPN

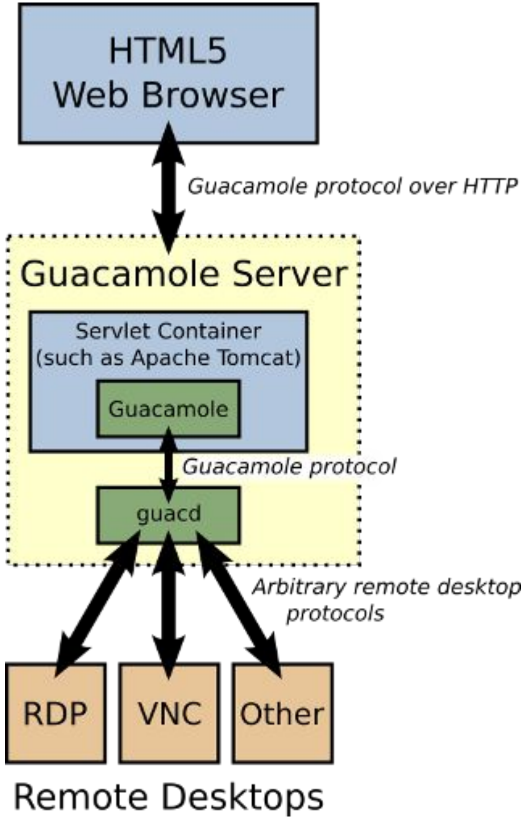
Raspberry PI 3 as an VPN customer gateway for AWS



https://www.blog.labouardy.com/vpn-connection-aws-resources-raspberry-pi-3

All-in-one: Apache guacamole

- Clientless remote desktop gateway
- Supports VNC, RDP and SSH
- Guacamole server installed in the bastion host
- Accessible by HTML5 compatible browsers

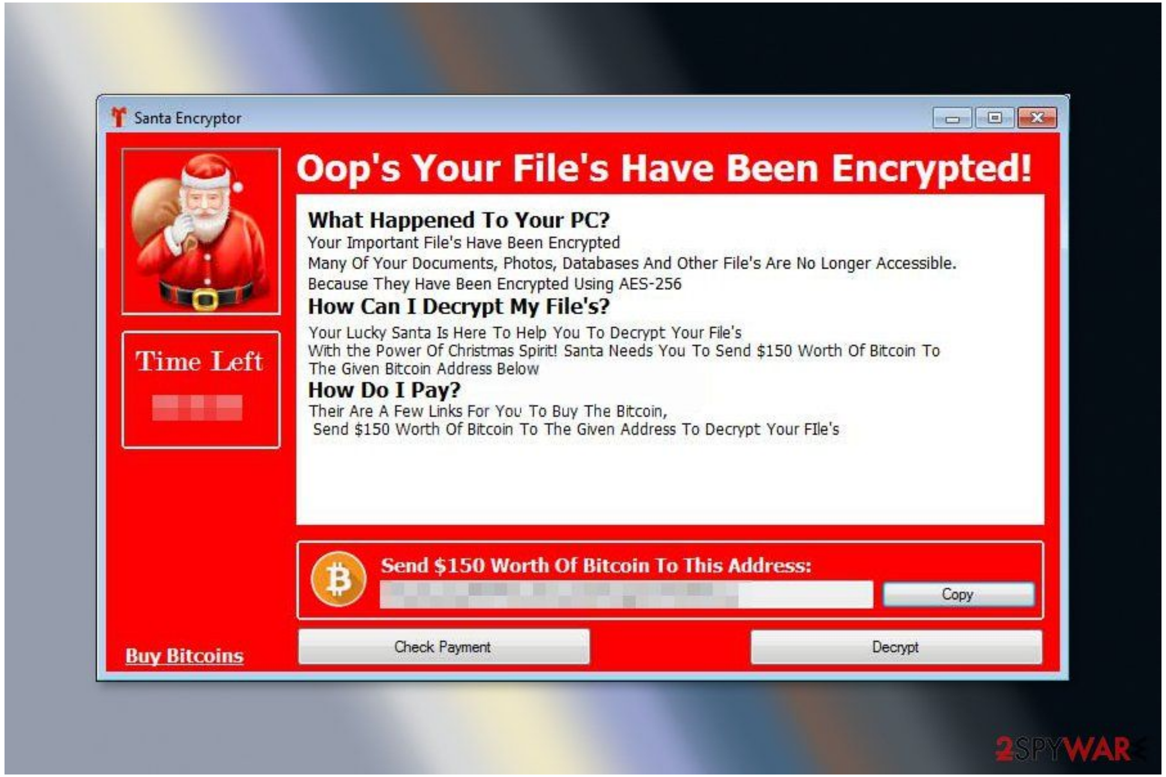


https://aws.amazon.com/marketplace/pp/B06Y67KPD9

Apache guacamole DEMO

https://ec2-63-33-68-143.eu-west-1.compute.amazonaws.com:8443/guacamole

Thank you for your attendance and Merry Xmas!



Oop's Your File's Have Been Encrypted!

What Happened To Your PC?
Your Important File's Have Been Encrypted
Many Of Your Documents, Photos, Databases And Other File's Are No Longer Accessible.
Because They Have Been Encrypted Using AES-256

How Can I Decrypt My File's?
Your Lucky Santa Is Here To Help You To Decrypt Your File's
With the Power Of Christmas Spirit! Santa Needs You To Send $150 Worth Of Bitcoin To
The Given Bitcoin Address Below

How Do I Pay?
Their Are A Few Links For You To Buy The Bitcoin,
Send $150 Worth Of Bitcoin To The Given Address To Decrypt Your FIle's

Send $150 Worth Of Bitcoin To This Address: