Index

• Introduction
• UCCS VPC Objective
• Why VPC
• VPC Options
  – Slides 6-10
• Routing
• Security
  – Slides 13-20
• Summary
Introduction

• Amazon Web Services (AWS)
  – EC2, VPC, MapReduce, SimpleDB, CloudFront, Simple Storage Service (S3), CloudFormation….and more

• Amazon VPC
  – Cloud Isolation
  – Extension of existing infrastructure
    • Added Security
    • IP Assigning
UCCS VPC Objective

- **Learning**: Offer a safe environment to gain hands-on experience
- **Research**: Provide a flexible environment for a wide array of research projects.
- **Recognition**: Improve UCCS standing in cyber security competitions such as ICTF
Why VPC

Affordable
- Use a minimal amount of UCCS computer resources
- Require no additional equipment
- Practical operating cost

Flexible
- Ability to handle various user population sizes
- Ability to handle various project requirements

Safe
- Isolated from public environment
- Restricted to UCCS internal environment

Simple
- Documented
- Automated “baseline” setup/teardown
- Easy to expand outside of “baseline”
Create an Amazon Virtual Private Cloud

Select a VPC configuration below:

- **VPC with a Single Public Subnet Only**
  Your instances run in a private, isolated section of the AWS cloud with direct access to the Internet. Network access control lists and security groups can be used to provide strict control over inbound and outbound network traffic to your instances.

- **VPC with Public and Private Subnets**
  In addition to containing a public subnet, this configuration adds a private subnet whose instances are not addressable from the Internet. Instances in the private subnet can establish outbound connections to the Internet via the public subnet using Network Address Translation.

- **VPC with Public and Private Subnets and Hardware VPN Access**
  This configuration adds an IPsec Virtual Private Network (VPN) connection between your Amazon VPC and your datacenter - effectively extending your datacenter to the cloud while also providing direct access to the Internet for public subnet instances in your Amazon VPC.

- **VPC with a Private Subnet Only and Hardware VPN Access**
  Your instances run in a private, isolated section of the AWS cloud with a private subnet whose instances are not addressable from the Internet. You can connect this private subnet to your corporate datacenter via an IPsec Virtual Private Network (VPN) tunnel.

Create: a /16 network with two /24 subnets. Public subnet instances use Elastic IPs to access the Internet. Private subnet instances access the Internet via a Network Address Translation (NAT) instance in the public subnet. (Hourly charges for NAT instances apply.)
Single Subnet Only
Public, Private and VPN
Private Subnet Only

Internet

VPC 10.0.0.0/16

1

Amazon EC2 API endpoint

Amazon S3 API endpoint

VPN Gateway

Customer Network 172.16.0.0/8

Amazon Web Services

VPN-Only Subnet 10.0.1.0/24

R

10.0.1.5

10.0.1.6

10.0.1.7
### Amazon Elastic Compute Cloud

**US East (Northern Virginia) Region**

- **Amazon EC2 running Linux/UNIX**
  - $0.085 per Small Instance (m1.small) instance-hour (or partial hour)  
  - **315 Hrs**  
  - **26.78**

- **Amazon EC2 EBS**
  - $0.10 per GB-month of provisioned storage  
  - **6.263 GB-Mo**  
  - **0.63**

  - $0.10 per 1 million I/O requests  
  - **142,684 IOs**  
  - **0.01**

  - $0.15 per GB-Month of snapshot data stored  
  - **2.128 GB-Mo**  
  - **0.32**

  - $0.01 per 10,000 gets (when loading a snapshot)  
  - **5,936 Requests**  
  - **0.01**

- **Elastic IP Addresses**
  - $0.01 per non-attached Elastic IP address per complete hour  
  - **584 Hrs**  
  - **5.84**

### Amazon Simple Storage Service

**US Standard Region**

- $0.140 per GB - first 1 TB / month of storage used  
  - **1.999 GB-Mo**  
  - **0.28**

- $0.01 per 1,000 PUT, COPY, POST, or LIST requests  
  - **16 Requests**  
  - **0.01**

### Amazon Virtual Private Cloud

- $0.05 per VPN Connection-Hour  
  - **547 Hrs**  
  - **27.35**
Safe

- Use existing security infrastructure
  - Only available from within UCCS network
- Isolated
  - No outside connection from within VPC
  - Encrypted VPN connection
- Controlled operating time
  - Automated baseline setup
  - Automated complete teardown
Simple

- Amazon Web Service (AWS) Management Console
  - Point-and-click web interface
  - Monitor services
  - Simplified setup
- AWS SDK for .NET
  - Automation using .NET framework
- Lots of documentation
Connecting to VPC

- Create a VPC
- Create a Customer Gateway
- Integrate
- Deploy AMIs
Gateway Requirements

IKE
- Establish IKE Security Association using Pre-Shared Keys

IPSec
- Establish IPSec Security Association in Tunnel mode
- AES 128-bit encryption
- Diffie-Helman Perfect Forward Secrecy ("Group 2" mode)
- IPSec Dead Peer Detection

Tunnel
- Bind tunnel to logical interface

BGP
- Establish Border Gateway Protocol (BGP)
CentOS Custom Gateway

• Install ipsec-tools
  – Racoon
  – Setkey

• Install quagga
  – Zebra
  – Bgp

• Bind tunnels to a logical interface

• Create point-to-point connection
CentOS Gateway Cont...

**IPSec Tunnel**

**IKE**: 72.21.209.225
Password

**IPSec**
- AES-128
- HMAC_SHA1
- DH_GROUP2
- SPD

**BGP**
- Peer: 169.254.255.2
- Remote ASN: 7224

**Tunnel 2**
- Remote: 72.21.209.193
- IP: 169.254.255.1

**Internet**

**Amazon Gateway**

**IPSec Tunnel**

**IKE**: 72.21.209.226
Password

**IPSec**
- AES-128
- HMAC_SHA1
- DH_GROUP2
- SPD

**BGP**
- Peer: 169.254.255.3
- Remote ASN: 7224
Summary

• Using Amazon’s VPC all three goals can be reached
  – Learning
    • Help solidify concepts through “hands on” experience
  – Research
    • Flexible environment with a vast support matrix to meet a wide array of research needs
  – Recognition
    • Through learning and research UCCS will be better equipped to compete on the world stage
IPSec

path include "/etc/racoon";
path pre_shared_key "/etc/racoon/psk.txt"

remote 72.21.209.225 {
    exchange_mode main;
    lifetime time 28800 seconds;
    dpd_delay 10;
    dpd_retry 3;
    proposal {
        encryption_algorithm aes128;
        hash_algorithm sha1;
        authentication_method pre_shared_key;
        dh_group 2;
    }
    generate_policy off;
}

remote 72.21.209.193 {
    exchange_mode main;
    lifetime time 28800 seconds;
    dpd_delay 10;
    dpd_retry 3;
    proposal {
        encryption_algorithm aes128;
        hash_algorithm sha1;
        authentication_method pre_shared_key;
        dh_group 2;
    }
    generate_policy off;
}

#!/sbin/setkey -f
flush;
spdflush;

spdadd 169.254.255.2/30 169.254.255.1/30 any -P out ipsec esp/tunnel/a.b.c.d-72.21.209.225/require;


spdadd 169.254.255.5/30 169.254.255.6/30 any -P out ipsec esp/tunnel/a.b.c.d-72.21.209.193/require;


spdadd 192.168.0.0/24 169.254.255.2/30 any -P out ipsec esp/tunnel/a.b.c.d-72.21.209.225/require;

spdadd 192.168.0.0/24 169.254.255.2/30 any -P in ipsec esp/tunnel/72.21.209.225-a.b.c.d/require;

spdadd 192.168.0.0/24 169.254.255.6/30 any -P out ipsec esp/tunnel/a.b.c.d-72.21.209.193/require;

spdadd 192.168.0.0/24 169.254.255.6/30 any -P in ipsec esp/tunnel/72.21.209.193-a.b.c.d/require;

spdadd 0.0.0.0/0 192.168.0.0/24 any -P out ipsec esp/tunnel/a.b.c.d-72.21.209.193/require;

spdadd 192.168.0.0/24 0.0.0.0/0 any -P in ipsec esp/tunnel/72.21.209.193-a.b.c.d/require;
hostname cgw-2493774d
password testPassword
enable password testPassword
!
log file /var/log/quagga/bgpd
!debug bgp events
!debug bgp zebra
debag bgp updates
!
router bgp 65000
bgp router-id a.b.c.d
network 169.254.255.2/30
network 169.254.255.6/30
network 0.0.0.0/0
!
! aws tunnel #1 neighbor
neighbor 169.254.255.1 remote-as 7224
!
! aws tunnel #2 neighbor
neighbor 169.254.255.5 remote-as 7224
!
line vty