Hardering routers and routing protocols

ROUTE Module 8

Agenda

- Securing control plane
- Hardening routing protocols
  - EIGRP
  - OSPF
  - RIP/RIPng
  - BGP

Elements of a router security policy

- Passwords
  - Complexity, encryption, How often should be changed?
- Authentication
  - Local database or AAA server? Banner?
- Access
  - What protocols are allowed for remote connection? SSH, HTTPS, SNMP?
- Services
  - What services run on the router?
- Filtering
  - Bogus filters on the edge (e.g., private addresses)
  - Antispoofing filters – e.g. ACL or uRPF

Elements of a router security policy

- Routing protocols
  - Should routing protocol authentication be used?
- Updates
  - What procedure is in place to update the version of Cisco IOS running on the router?

Other important topics to consider:

- Backups
- Documentation
- Monitoring
- Redundancy

Access Control Lists

ACL

- Standard ACL
  - Source or destination IP addresses
- Extended ACL
  - Source, destination IP addresses, ports or variety other criteria
- Time-Based ACL
  - If you want to allow specific protocols to come into your network during business
- Infrastructure ACL
  - Applied to routers sitting at the edge of an enterprise network

Time-Based ACL

- Allow HTTP traffic to a host during a working hours on weekdays

```
! Define time period
R1(config)# time-range WEEKDAYS
R1(config-time-range)# periodic weekdays 8:00 to 17:00

! Create ACL
R1(config)# access-list 100 permit tcp any host
192.168.1.10 eq 80 time-range WEEKDAYS

! Apply to an interface
R1(config)# interface serial 1/0
R1(config-if)# ip access-group 100 in
```

Infrastructure ACLs

- Extended ACL applied to routers residing on the outer edges of an enterprise network
- The primary purpose of infrastructure ACL is to prevent malicious traffic from entering the enterprise
- Examples:
  - Permit BGP traffic
  - Deny fragmented traffic
  - Allow only management protocols from management stations

Infrastucture ACL

```
ip access-list extended INFRASTRUCTURE

! Block packet fragments
deny tcp any any fragments
deny udp any any fragments
deny icmp any any fragments
deny ip any any fragments

! Allow routing protocol and network management traffic
permit tcp host <external-bgp-peer> host <internal-bgp-peer> eq bgp
permit tcp host <external-bgp-peer> eq bgp host <internal-bgp-peer>
permit tcp <address-of-management-stations> any eq 22
permit tcp <address-of-management-stations> any eq 161
permit icmp <address-of-management-stations> any echo

! BLOCK ALL OTHER TRAFFIC DESTINED FOR INTERNAL NETWORK
deny ip any <address-space-of-internal-network>

! PERMIT OFF-NET TO OFF-NET TRAFFIC
permit ip any any

! APPLY ACL TO AN INTERFACE CONNECTING TO AN EXTERNAL NETWORK
interface Serial1/0
ip access-group INFRASTRUCTURE in
```

Securing management plane

Telnet Vulnerabilities

- With **Telnet**, all usernames, passwords, and data sent over the public network in clear text are vulnerable.
- A user with an account on the system could gain elevated privileges.

SSH – Secure Shell

- SSH provides strong authentication and secure communications over insecure channels
- It is a replacement for rlogin, rsh, rcp, and rdist in addition to Telnet
- Entire login session, including transmission of password, is encrypted; therefore, it is almost impossible for an outsider to collect passwords
- *Although SSH is secure, vendors' implementations of SSH might contain vulnerabilities!*
- SHS version 1 implementations are vulnerable to various security compromises; whenever possible, use SSH version 2 instead of SSH version 1

Configuring SSH

1) Configure a user with a password.
2) Configure a hostname and a domain name.
3) Generate RSA keys.
4) Allow SSH transport on the vty lines.

```
switch(config)# username xyz privilege 15 secret abc123
switch(config)# ip domain-name xyz.com
switch(config)# crypto key generate rsa size
switch(config)# ip ssh version 2
switch(config)# line vty 0 15
switch(config-line)# login local
switch(config-line)# transport input ssh
```

Notes about password encryption

- Locally saved password can be compromised
  - Security recommendation is to encrypt all passwords
- enable secret password
  - SHA256 hash
- Service password encryption
  - Clear text password is encrypted
  - Weak, can be decrypted easily
- Local user database
  - Stores the password for the user as SHA256 hash

Unicast Reverse Path Forwarding (uRPF)

uRPF

- uRPF can help block packets having a spoofed IP address

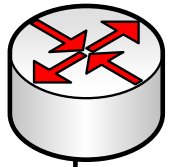
1. Checks the source IP address of a packet arriving on an interface
2. Determine whether that IP address is reachable, based on the router's FIB, (CEF must be enabled).

- **Strict mode**
  - Source IP address is checked if it is reachable
  - Packet must also arrive on the same interface the router would use to send traffic back to that IP address
- **Loose mode**
  - Only verifies that the source IP address of a packet is reachable
- **VRF mode**
  - Similar to loose mode, source IP addresses are checked against the FIB for a specific VRF.

uRPF example

```
R1# show ip cef
Prefix Next Hop Interface
0.0.0.0/0 172.16.0.2 Serial2/0
0.0.0.0/8 drop
0.0.0.0/32 receive
10.0.0.0/24 attached FastEthernet0/0
10.0.0.0/32 receive FastEthernet0/0
10.0.0.1/32 receive FastEthernet0/0
192.168.0.0/24 receive FastEthernet1/0
192.168.0.1/32 receive FastEthernet1/0
…
…
```

```
interface FastEthernet1/0
  ip address 192.168.0.1 255.255.255.0
  ip verify unicast source reachable-via rx
```



192.168.0.0/24

src ip: 10.0.0.1

uRPF configuration

ip verify unicast source reachable-via {**rx** | **any**} [**allow-default**] [**allow-selfping**] [*acl*]

- `rx` – enable strict mode
- `any` – enable loose mode

- Example of configuration:

```
interface FastEthernet1/0
ip address 192.168.1.1 255.255.255.0
ip verify unicast source reachable-via rx
```

uRPF

- Strict mode could cause traffic to be dropped if an asynchronous routing situation exists
- By default, a router with uRPF configured would drop a packet whose source IP address was only reachable by a default route
  - Can be overridden by using **allow-default**
- uRPF is recommended by Best Current Practice (BCP38)
- Protect mainly “other” networks
  - uRPF deployments means less DDoS attacks
- There can be performance impact of enabling the feature

AAA – Authentication, Authorization, Accounting

AAA

- **Authentication**
  - User identification,
  - Login and password dialog
  - Challenge and response
- **Authorization**
  - Authorization service determines what the user is allowed to do
- **Accounting**
  - Start and stop times
  - Executed commands
  - Number of packets, bytes

AAA Advantages

- Flexibility
  - Offers additional authorization flexibility on a per-command or per-interface level
- Scalability
  - AAA provides a very scalable solution that is required when managing large networks.
- Standardized authentication method
  - RADIUS protocol is open standard to ensure interoperability with other vendor devices

Differences between RADIUS and TACACS protocols

Characteristics	TACACS+	RADIUS
Transport protocol	TCP	UDP
Modularity	Separate services for AAA	Combines authentication and authorization
Encryption	Encrypts entire packet	Encrypts password
Accounting func.	Basic accounting	Robust accounting features
Standards-based	No – proprietary	Yes

Basic AAA configuration

- RADIUS authentication + local database fallback

```
R1(config)# username ADMIN secret cisco
R1(config)# aaa new-model
R1(config)# radius server RADIUS-1
R1(config-radius-server)# address ipv4 192.168.1.101
R1(config-radius-server)# key 1234pass
R1(config)# aaa group server radius RADIUS-GROUP
R1(config-sg-radius)# server name RADIUS-1
R1(config)# aaa authentication login default group
RADIUS-GROUP local
R1(config)# aaa authentication login TELNET-LOGIN
group RADIUS-GROUP local-case
R1(config)# line vty 0 4
R1(config-line)# login authentication TELNET-LOGIN
```

SNMP Security

SNMP

- SNMP is widely used for gathering statistics and for network management
- NMS (Network Management Server)
  - Runs a network management application (gathers statistics, push configuration)
- SNMP Agent
  - A software that runs on a managed device (e.g., a router, or switch)
- Management Information Base
  - Information about a managed device’s resources and activity is defined by a series of objects.

SNMPv1/v2c security

- A malicious users can collect information or change the configuration of a managed device
- SNMPv1 and SNMPv2c use community strings to gain read-only or read-write access to a managed device

1. Change community string to non-default value
2. Protect SNMP access via ACL

```
R1(config)# snmp-server community $3cr3T ro 10
R1(config)# snmp-server community c1$c0 rw 10
R1(config)# access-list 10 permit host 10.1.1.1
```

SNMPv3

- SNMPv3 increases the security of network-management traffic and offers three primary security enhancements:
  - Integrity: SNMPv3 ensures that an SNMP message was not modified in transit.
  - Authentication: SNMPv3 can validate the source of an SNMP message.
  - Encryption: DES, 3DES, or AES encryption algorithms can be used

Routing Protocol Authentication

Routing protocol neighborships

- A malicious user can run a rogue router and inject false routing information into a network
  - Data traffic can flow through the rogue router, thus allowing traffic interception
- EIGRP and OSPF form neighborships with adjacent routers dynamically
  - Can be a concern from a security perspective – it is easy to create a neighborship
- BGP requires statically configured neighborship
  - Less prone to the previous attack

Authentication Methods

- Two authentication methods exist
  - Plain text authentication
  - Hashing authentication
- **Plain text authentication**
  - Supported for RIPv2, OSPFv2, IS-IS
  - Password is sent as a clear text together with routing update
- **Hashing authentication**
  - Key + routing update enter to a hash function
  - Hash digest is appended to the routing update
  - MD5 – supported by RIPv2, EIGRP, OSPFv2, OSPFv3, IS-IS, BGP
  - SHA – supported vary on IOS version – generally EIGRP, OSPF

RIP authentication

RIPv2 Authentication ①

- *Without authentication of sender RIP blindly trust every packet it accepts!*
- Configuration guide:
  1. Creation of “keychain” – list of keys
  2. Activation of authentication form on interface
  3. Activation of keychain on interface

RIPv2 Authentication ②

1. Creation of keychain:

```
Router(config)# key chain NAME
Router(config-keychain)# key NUMBER
Router(config-keychain-key)# key-string PASSWORD
```

2. Activation of authentication form:

```
Router(config-if)# ip rip authentication mode {md5|text}
```

3. Activation of keychain:

```
Router(config-if)# ip rip authentication key-chain NAME
```

EIGRP Authentication

Authentication

- EIGRP supports only MD5 authentication
- Configuration guideline is similar to RIPv2 authentication setup

1. Creation of key chain:

```
Router(config)# key chain NAME
Router(config-keychain)# key NUMBER
Router(config-keychain-key)# key-string PASSWORD
```

2. Activation of authentication on interface:

```
Router(config-if)# ip authentication mode eigrp AS md5
```

3. Activation of key chain on interface:

```
Router(config-if)# ip authenticat key-chain eigrp AS NAME
```

Authentication

- If Named Mode is used:

```
router eigrp TEST
 !
 address-family ipv6 unicast autonomous-system 1
  !
  af-interface GigabitEthernet0/1
   authentication mode md5
   authentication key-chain TEST
  exit-af-interface
```

Key Chain Lifetimes ①

```
Router(config-keychain-key)#
  accept-lifetime start-time {infinite | end-time |
  duration seconds}
```

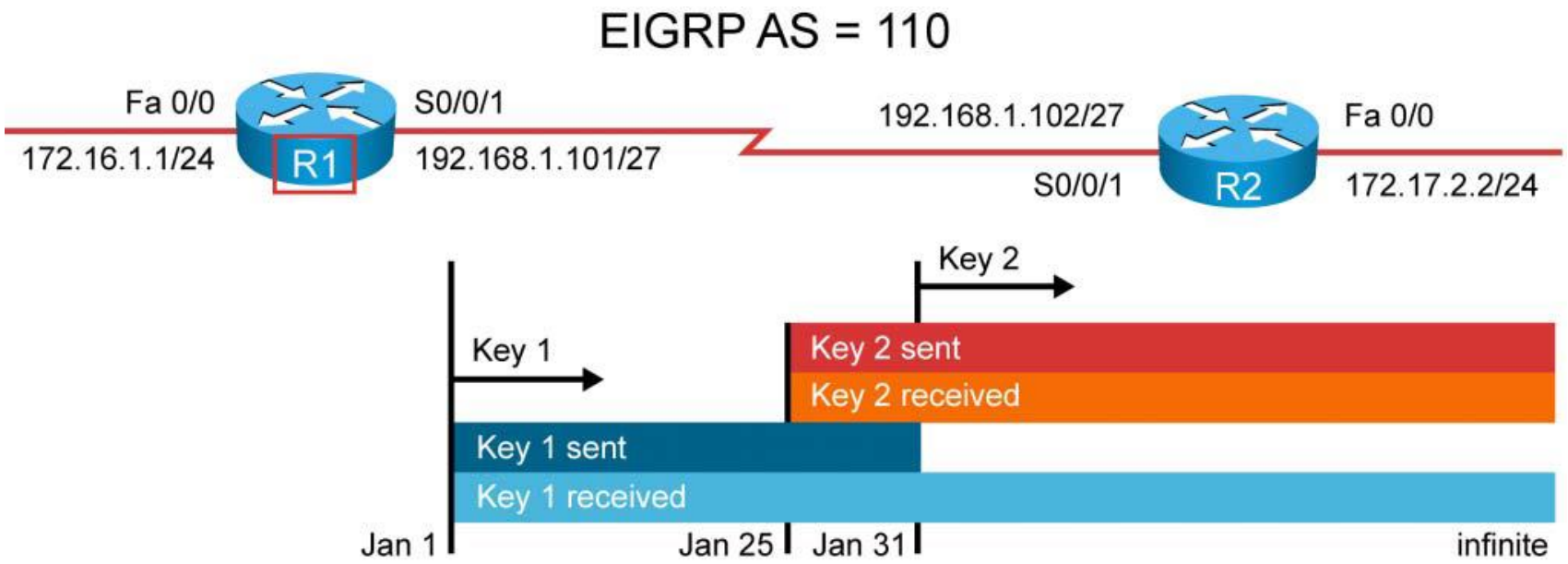
- Optional command: defining interval when router is checking received packets signed with this key

```
Router(config-keychain-key)#
  send-lifetime start-time {infinite | end-time |
  duration seconds}
```

- Optional command: defining interval when router sends packets signed with this key

Key Chain Lifetimes ②

```
R1(config)#
key chain routerR1chain
 key 1
  key-string firstkey
  accept-lifetime 04:00:00 Jan 1 2009 infinite
  send-lifetime 04:00:00 Jan 1 2009 04:00:00 Jan 31 2009
 key 2
  key-string secondkey
  accept-lifetime 04:00:00 Jan 25 2009 infinite
  send-lifetime 04:00:00 Jan 25 2009 infinite
```



OSPF Authentication

Differences between OSPFv2/v3

- Authentication is handled differently in OSPFv2/v3
- Three types of authentication are available for OSPFv2
  - None, clear text, MD5-based
- OSPFv3 does not provide authentication but relies on IPv6 IPSec
  - AH for authentication, ESP for encryption and authentication
  - IPSec parameters are usually handled by ISAKMP/IKE, however, for OSPFv3, they must be specified manually and must match on all routers
- Since IOS Release 15.4(2)T, there is a support for RFC 7166 - Authentication Trailer for OSPFv3
  - Similar authentication as in OSPFv2

Simple Password Authentication ①

```
Router(config-if)# ip ospf authentication-key password
```

- Configures plaintext password on interface

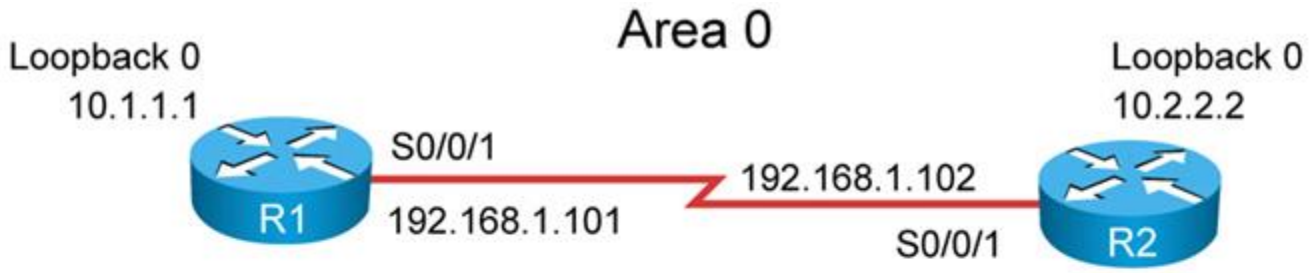
```
Router(config-router)# area area-id authentication
```

- For backward compatibility with older IOSes also area authentication mode is supported

```
Router(config-if)# ip ospf authentication [null]
```

- In newer IOSes each interface could support different authentication method
- **`null`** optional argument deactivates authentication on target interface

Simple Password Authentication ②



R1#

```
<output omitted>
interface Loopback0
 ip address 10.1.1.1 255.255.255.0

<output omitted>
interface Serial0/0/1
 ip address 192.168.1.101 255.255.255.224
 ip ospf authentication
 ip ospf authentication-key plainkey

<output omitted>
router ospf 10
 log-adjacency-changes
 network 10.1.1.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
```

R2#

```
<output omitted>
interface Loopback0
 ip address 10.2.2.2 255.255.255.0

<output omitted>
interface Serial0/0/1
 ip address 192.168.1.102 255.255.255.224
 ip ospf authentication-key plainkey

<output omitted>
router ospf 10
 log-adjacency-changes
 network 10.2.2.2 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
 area 0 authentication
```

MD5 Authentication ①

```
Router(config-if)#ip ospf message-digest-key keyid md5 key
```

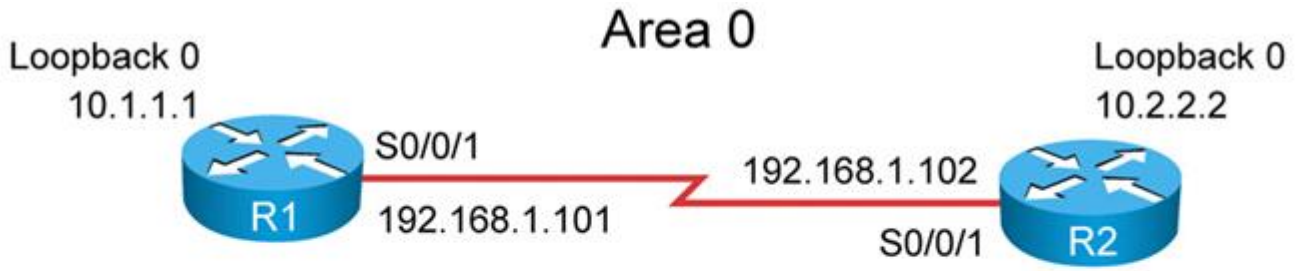
- Creates key and binds it with ID
  - Pair (KeyID)-Key must be same between neighbors
  - IF multiple keys are present on interface THEN the last added key is used for signing outgoing message
  - All of present keys are used when accepting message
- Older IOS whole area MD5 authentication

```
Router(config-router)#
  area area-id authentication message-digest
```

- It turns MD5 authentication on target interface. And as in previous case optional argument **null** deactivates it.

```
Router(config-if)#
  ip ospf authentication {message-digest | null}
```

MD5 Authentication ②



R1#

```
<output omitted>
interface Loopback0
 ip address 10.1.1.1 255.255.255.0

<output omitted>
interface Serial0/0/1
 ip address 192.168.1.101 255.255.255.224
 ip ospf authentication message-digest
 ip ospf message-digest-key 1 md5 mysecret

<output omitted>
router ospf 10
 log-adjacency-changes
 network 10.1.1.1 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
```

R2#

```
<output omitted>
interface Loopback0
 ip address 10.2.2.2 255.255.255.0

<output omitted>
interface Serial0/0/1
 ip address 192.168.1.102 255.255.255.224
 ip ospf message-digest-key 1 md5 mysecret

<output omitted>
router ospf 10
 log-adjacency-changes
 network 10.2.2.2 0.0.0.0 area 0
 network 192.168.1.0 0.0.0.255 area 0
 area 0 authentication message-digest
```

Verification of Authentication

```
R1#show ip ospf interface
Serial2/0 is up, line protocol is up
  Internet Address 192.168.1.101/27, Area 0
  Process ID 10, Router ID 10.1.1.1, Network Type POINT_TO_POINT, Cost: 64
  Transmit Delay is 1 sec, State POINT_TO_POINT
Neighbor Count is 1, Adjacent neighbor count is 1
    Adjacent with neighbor 10.2.2.2
  Suppress hello for 0 neighbor(s)
  Message digest authentication enabled
    Youngest key id is 1
Loopback0 is up, line protocol is up
  Internet Address 10.1.1.1/24, Area 0
  Process ID 10, Router ID 10.1.1.1, Network Type LOOPBACK, Cost: 1
  Loopback interface is treated as a stub Host

R1#ping 10.2.2.2
Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 10.2.2.2, timeout is 2 seconds:
!!!!!
Success rate is 100 percent (5/5), round-trip min/avg/max = 28/28/32 ms
```

```
R1#
*Feb 17 18:51:31.242: OSPF: Rcv pkt from 192.168.1.102, Serial0/0/1 :
Mismatch Authentication type. Input packet specified type 0, we use type 1

R2#
*Feb 17 18:50:43.046: OSPF: Rcv pkt from 192.168.1.101, Serial0/0/1 :
Mismatch Authentication type. Input packet specified type 1, we use type 0
```

```
R1#
*Feb 17 18:54:01.238: OSPF: Rcv pkt from 192.168.1.102, Serial0/0/1 :
Mismatch Authentication Key - Clear Text

R2#
*Feb 17 18:53:13.050: OSPF: Rcv pkt from 192.168.1.101, Serial0/0/1 :
Mismatch Authentication Key - Clear Text
```

Extended Crypto Authentication for OSPF

- Create a keychain

```
Router(config)# key chain OSPF
Router(config-keychain)# key 1
Router(config-keychain-key)#
  cryptographic-algorithm {hmac-sha-{1|256|384|512}|md5}
Router(config-keychain-key)# key-string ChciBytCCIE
```

- Apply authentication on OSPFv3 interface

```
Router(config-if)# ip ospf authentication key-chain OSPF
```

IPsec protection for OSPFv3

- The FastEthernet0/0 interface is configured with AH-based authentication

```
Router(config-if)# ipv6 ospf auth ipsec spi 1000
sha1 8E63C2FF7E2997D7D26FD80E047C43A7FEEA9833
```

- The Serial1/0 interface is configured with ESP-based encryption and authentication

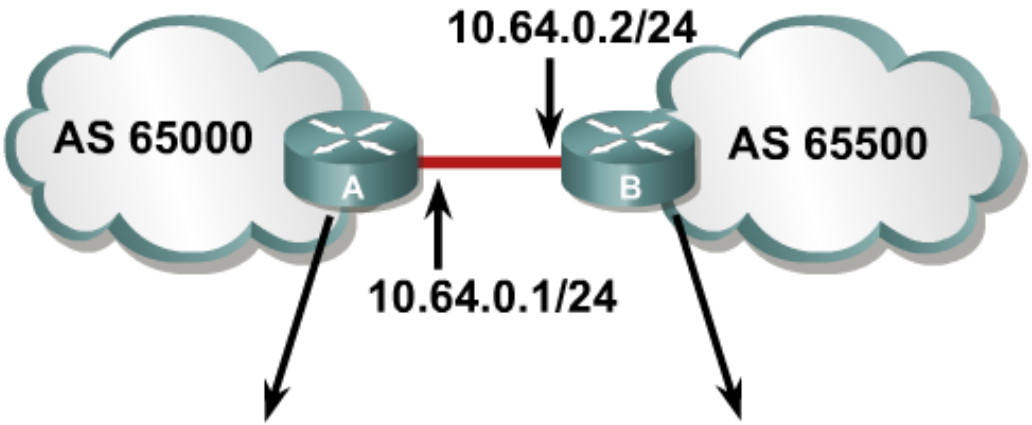
```
Router(config-if)#
  ipv6 ospf encryption ipsec spi 1001 esp
  aes-cbc 128 DE7EC1FDF5BDC3367DB071BF090FFA2A
  sha1 6D858314599428 7B6088A2D674E412A5F862DD5B
```

BGP Authentication

BGP Authentication

- BGP neighbors are statically configured
  - TCP session can be, however, hijacked
- BGP uses MD5 authentication to mitigate the attack

```
Router(config-router)#
neighbor {ip-address | peer-group-name} password string
```




```
router bgp 65500
 neighbor 10.64.0.2 remote-as 65500
 neighbor 10.64.0.2 password v61ne0qkel33&
```

```
router bgp 65500
 neighbor 10.64.0.1 remote-as 65000
 neighbor 10.64.0.1 password v61ne0qkel33&
```

Slides created by Matěj Grégr and Vladimír Veselý partially from official course materials, but the most of the credit goes to CCIE#23527 Ing. Peter Palúch, Ph.D.

Last update: 2016-05-02