Dinamik Yönlendirme Protokolleri

Routerlar kendisine gönderilen paketin hangi arayüzünden göndereceğine yönlendirme tablosuna bakarak kara verir. Yönlendirme tablolarında ise bu adresler statik olarak tanımlanabilirken çeşitli dinamik yönlendirme protokolleri kullanılarak da öğrenilebiliyor. Statik rota tanımları routera yetkisiz giriş söz konusu olmadığı sürece manipüle edilebilmeleri pek mümkün değildir. Aynı durum dinamik yönlendirme protokolleri için geçerli değildir.

Dinamik yönlendirme protokollerinde routerlar öğrendikleri network bilgilerini komşu routerlara da öğretirler ve öğrendikleri her bir network için en uygun rotayı tespit etmeye çalışırlar. Bu süreçte önlem alınmadığı taktirde bir tehdit aktörü bilgisayarından bağlı olduğu routera içeriği değiştirilmiş paketler (Örneğin OSPF Protokolü için LSU paketi) göndererek topolojideki routerların yönlendirme tablosunu manipüle edebilir ve trafiği istediği gibi yönlendirebilir. Bu sayede;

- Paketleri discart ettirebilir.
- Routing loop oluşturabilir.
- Trafiğin farklı bir rota üzerinde geçmesini sağlayarak trafiği dinleyebilir.

Yönlendirme protokollerine dışarıdan bir cihaz ile müdahale edilebilmesini önleyebilmek için routerlar arasında kimlik doğrulama mekanizmaları kullanılabiliyor. Bu sayede routerlara kaynağı belli olmayan cihazlardan yönlendirme protokolüne ait paketler gönderilemiyor.

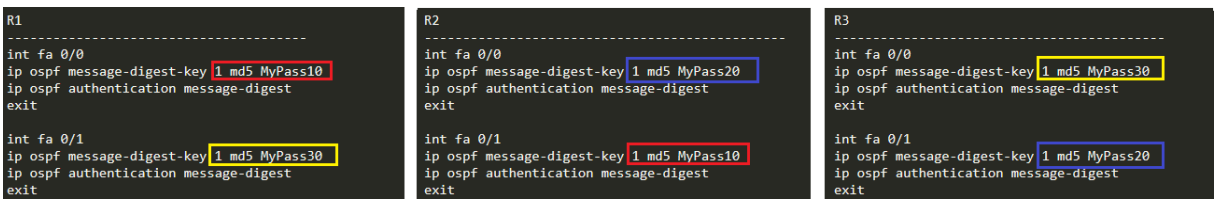
Günümüzde kurumsal ağlarda yaygın olarak kullanılan (şubeleri birbiriene bağlamak için vs.) dinamik yönlendime protokolü OSPF'dir. OSPF protokolünde MD5 veya SHA algoritmaları kullanılarak kimlik doğrulama mekanizmaları konfigüre edilebiliyor (**OSPF protokolünün işleyişi hakkında daha detaylı bilgi için "CCNA – 3.01 – 3.02 - OSPF" notlarını inceleyebilirsin**).

OSPF MD5 Routing Protokolü Kimlik Doğrulama Konfigürasyonu

OSPF protokolünde MD5 kimlik doğrulama mekanizması cihaz genelinde uygulanabilirken cihaz üzerindeki belirli portlara da uygulanabiliyor.

OSPF MD5 Kimlik Doğrulama Konfigürasyonu

- Belirli arayizlerde OSPF MD5 Kimlik Doğrulama konfigürasyonu yapabilmek için ilk olarak ilgili arayüze giriş yapılıyor ve arayüzde kimlik doğrulama mekanizmasında kullanılacak parola bilgisi için "**ip ospf message-digest-key <Key> md5 <Password**>" komutu kullanılıyor.
    - Burada Key olarak ne seçildiği önemli değildir. Sadece seçilen Key değeri birbirine bağlı arayüzlerde aynı seçilmelidir. Aynı şekilde Password bilgileri de birbirine bağlı arayüzler arasında aynı olmak zorundadır. Aksi taktirde cihazlar komşuluk kuramaz. (Uygulama yapabilmek için "Lab -> Calışma" adlı dizine bulabilirsin).
- Mekanizma üzerinde kullanılacak parola bilgisi tanımlandıktan sonra "**ip ospf authentication message-digest**" komutu kullanılarak limlik doğrulama mekanizması devreye alınıyor.



- Cihaz genelinde OSPF md5 kimlik doğrulama mekanizmasını devreye alabilmek için öncelikle arayizlere giriş yaparak "**ip ospf message-digest-key <Key> md5 <Password**>" komutuyla arayüzlerde kullanılacak parolalar tanımlanıyor (Bu kısım bağlantı kurulan arayüzler arasında aynı olmak zorundaydı). Ardından Global konfigürasyon modunda "**router ospf <Proccess Number>**" komutuyla OSPF konfigürasyonlarının tanımlandığı prosese giriş yapılıyor. Ardıdan "**area <Area Id> authentication message-digest**" komutuyla md5 kimlik doğrulama mekanizması cihaz geneline devreye alınıyor.

```
R1
----------------------------
int range fa 0/0
ip ospf message-digest-key 1 md5 MyPass
exit
int fa 0/1
ip ospf message-digest-key 1 md5 MyPass
exit

router ospf 1
area 1 authentication message-digest
exit
```

```
R2
----------------------------
int range fa 0/0
ip ospf message-digest-key 1 md5 MyPass
exit
int fa 0/1
ip ospf message-digest-key 1 md5 MyPass
exit

router ospf 2
area 1 authentication message-digest
exit
```

```
R3
----------------------------
int range fa 0/0
ip ospf message-digest-key 1 md5 MyPass
exit
int fa 0/1
ip ospf message-digest-key 1 md5 MyPass
exit

router ospf 3
area 1 authentication message-digest
exit
```

OSPF SHA Routing Protokolü Kimlik Doğrulaması Konfigürasyonu

MD5 algoritması güvenli kabul edilmediği için yanlızca cihazlarda desteklenen daha güçlü bir algoritma olmadığı durumlarda kullanılmalıdır. OSPF protokolünde (cihazlar destekliyorsa) SHA algoritması kullanılarak da kimlik doğrulama işlemi yapılabilmektedir. Konfigürasyonu için;

- İlk adımda kullanılacak SHA algoritması için tanımlamalar yapılması gerekiyor.
    - "**key chain <Name>**" komutuyla bir anahtar zinciri oluşturuluyor.
    - "**key <Key ID>**" komutuyla bir anahtar belirleniyor. Bu anahtar birbirine bağlı her iki arayüzde de aynı seçilmek zorundadır.
        - "**key-string <String>**"komutuyla kimlik doğrulama işleminde kullanılacak parola tanımlanıyor.
        - "**cryptographic-algorithm (hmac-sha-1 | hmac-sha-256 | hmac-sha-384 | hmac-sha-512 | md5)**" komutuyla kullanılacak algoritma seçimi yapılıyor.
        - İsteğe bağlı olarak "**send-lifetime start-time {infinite |end-time | dıratin <seconds>}**" komutuyla tanımlanan anahtarın kullanım süüresinin ne zaman sonlanacağı belirtilebiliyor.
- SHA algoritması için tanımlamalar yapıldıktan sonra SHA kimlik doğrulama mekanizması kullanılmak istenen arayüzlere girilerek "**ip ospf authentication key-chain <Name>**" komutu uygulanıyor.

```
R1(config)# key chain SHA256
R1(config-keychain)# key 1
R1(config-keychain-key)# key-string ospfSHA256
R1(config-keychain-key)# cryptographic-algorithm hmac-sha-256
R1(config-keychain-key)# exit
R1(config-keychain)# exit
R1(config)# interface s0/0/0
R1(config-if)# ip ospf authentication key-chain SHA256
R1(config-if)#
```