Ağ Kimlik Denetimi Çalışma Grubu

FREE-RADIUS - LDAP ile KİMLİK DENETİMİ

Hazırlayanlar:
(Alfabetik Sıra ile)

Gökhan AKIN
Hüseyin YÜCE
Hüsnü DEMİR

MAYIS 2008

İÇİNDEKİLER

1. GİRİŞ

Her ne kadar üniversiteler açısından tartışmalıda olsa 5651 sayılı kanun getirdiği sorumluluklar ve bu kanun çerçevesinde akademisyen, idari ve öğrenci kullanıcı grupları olan üniversite ağlarında kimlik denetimi hususu önem kazanmıştır.

Özellikle teknolojik gelişmeler ve kullanım kolaylığı nedeniyle kablosuz ağların yaygınlaşması ağ denetimini, kullanıcı kimlik denetimi zorlaştırmaktadır.

Teknolojik yaşam biçimi ve dünyadaki dijital değişim güncel yaşam sorunları ve suç şeklini de bu platforma taşımıştır. Özellikle denetimsiz ağlar bilişim suçları açısından birer suç kaynağı haline gelmesi kablolu ve kablosuz ağlarda kimlik denetiminin önemini bir kat daha artırmıştır.

Kimlik denetimi konusunda öne çıkan ve yani ağ cihazlarının çoğunun bir özellik olarak sunduğu IEEE 802.1x ağ erişim kontrol standardı kimlik denetimi konusunda bir yöntemdir.

802.1x kimlik doğrulama güvenliği, kablosuz istemciden erişim noktasına bir kimlik doğrulama talebini gönderir ve erişim noktası, istemciyi bir EAP uyumlu RADIUS sunucusunda doğrular. RADIUS sunucusu kullanıcıyı (parolalar ya da sertifikalar aracılığıyla) ya da sistemi (MAC adresi aracılığıyla) doğrular. Kuramsal olarak, kablosuz istemci işlemler tamamlanana kadar ağa katılamaz.

802.1x'in kimlik doğrulama algoritması için EAP-TLS, EAP-TTLS ve Korunmuş EAP (PEAP) kullanılabilir. Bunlar kablolu ve kablosuz istemcinin kendi kimliğini RADIUS sunucusunda doğrulama yöntemleridir. RADIUS kimlik doğrulamasında, kullanıcı kimlikleri bilgileri sunucu üzerinde var olan kullanıcılardan yapılabileceği gibi kullanım kolaylığı sağlayacak olan harici bir kullanıcı veri tabanından da yapılabilir. Bu veritabanı LDAP, xSQL(MSSQL, MySQL, PostgreSQL, Oracle) veya var olan bir e-posta sistemi olabilir.

RADIUS kimlik denetim (Authentication) amacı ile kullanılması yanı sıra, yetkilendirme (Authorization) ve kullanıcın takip edilesi (Accounting) içinde kullanılabilir. Bu işlemler kısaca AAA olarak da anılmaktadır.

Bu makalede özellikle ilgili kanunlar ile gelen bazı sorumlukların sonucunda kablolu ve kablosuz ağ kullanıcılarının kimlik denetimini sağlamak için yapılabilecek teknik yöntemlere ve bu yöntemlerin kavramlarına değinilmiştir.

2. TANIMLAR

2.1. IEEE 802.1x

IEEE 802.1x port tabanlı ağ erişim kontrol standardıdır. Kullanıcı bilgileri (kullanıcı adı-parola) yardımı ile ağa bağlanılmasına izin verilmesini sağlar (Bazı özel durumlarda MAC adreside kullanılmaktadır.). Kullanıcı doğrulama sırasında EAP (Extensible Authentication Protocol) yöntemi kullanılır. Bu şekilde ağ erişimi isteyen cihazdan doğrulama yapan mekanizmaya kadar kullanıcı bilgilerin sürekli şifreli gitmesini sağlanır.

802.1x sistemi kablolu ve kablosuz ağ bağlantılarında farklı olarak uygulanmaktadır. Birden fazla kullanıcı aynı kablosuz erişim cihazından (Access Point'ten) sanal portlar sayesinde ağ erişim izini alabilir. Ancak kablolu hizmet veren IEEE 802.1x destekleyen anahtar cihazlarında (switch) her bir fiziksel porttan sadece bir yetkilendirme yapılabilmektedir.

2.2. Extensible Authentication Protocol (EAP)

Genişletilebilir Kimlik Kanıtlama Protokolü (EAP - Extensible Authentication Protocol) [RFC 3748] kimlik kanıtlama için bir iletim protokolüdür, bir kimlik kanıtlama yöntemi değildir.

EAP kimlik kanıtlama sürecinde, kimlik kanıtlama sunucusu ile istemci arasında geçen ve tarafların hangi kimlik kanıtlama yöntemini kullanacaklarını belirler. EAP kimlik kanıtlama yöntemi olarak MD5, TLS, TTLS, PEAP, LEAP kullanır.

	EAP-MD5	**LEAP**	**EAP-TLS**	**EAP-TTLS**	**PEAP**
Sunucu Sertifikası	Hayır	Hayır	Evet	Evet	Evet
İstemci Sertifikası	Hayır	Hayır	Evet	Hayır	Hayır
WPA Anahtar Değişimi	Hayır	Evet	Evet	Evet	Evet
Güvenilirlik	Hayır	Hayır	Evet	Evet	Evet

2.3. Şifreleme Teknikleri

2.3.1 Simetrik Şifreleme

Aynı anahtar kelime ile verinin hem şifrelenmesi hem de geri çözülmesi şeklinde çalışan tekniktir.
Veri + Anahtar = Sifreli Veri
Şifreli Veri + Anahtar = Veri

Az sistem kaynağı tüketen bir şifreleme sistemidir ancak anahtarın karşılıklı taraflara güvenli ulaştırılması zordur.

2.3.2 Asimetrik Şifreleme

İki anahtardan oluşan bu sistemde anahtar1'in şifrelediğini anahtar2, anahtar2'nin şifrelediği ise anahtar1 açabilir.

Veri + Anahtar1 = Şifreli Veri
Şifreli Veri + Anahtar2 = Veri

Anahtarlardan bir tanesi istemcilere dağıtılır, bu sebepten de bu anahtara "public" anahtar ismi verilir. Diğer anahtar ise sadece sunucu tarafında kalır. Bu anahtara da "private" anahtar denir.

İstemcinin "public" anahtarı ile şifrelediği sadece sunucu "private" anahtarı ile açabilir. Ancak sunucun "private" anahtarı ile şifrelediğini herkes "public" anahtarı ile açabilir. Buradan da anlaşılacağı üzere bir public/private anahtar çifti ile çift yönlü güvenli bir haberleşme yapılamaz. Bu teknik ile çift yönlü güvenli haberleşme için bütün istemciler içinde ayrı public/private anahtar çiftinden oluşturulması gerekir. (EAP-TLS gibi)

2.4. Sertifika

Sertifika Public anahtarı ve bunu yanı sıra hizmet alınacak kurumun Adı, web adresi, mail adresi ...vs bilgileri barındıran bir dokümana verilen addır.

2.5. Hash

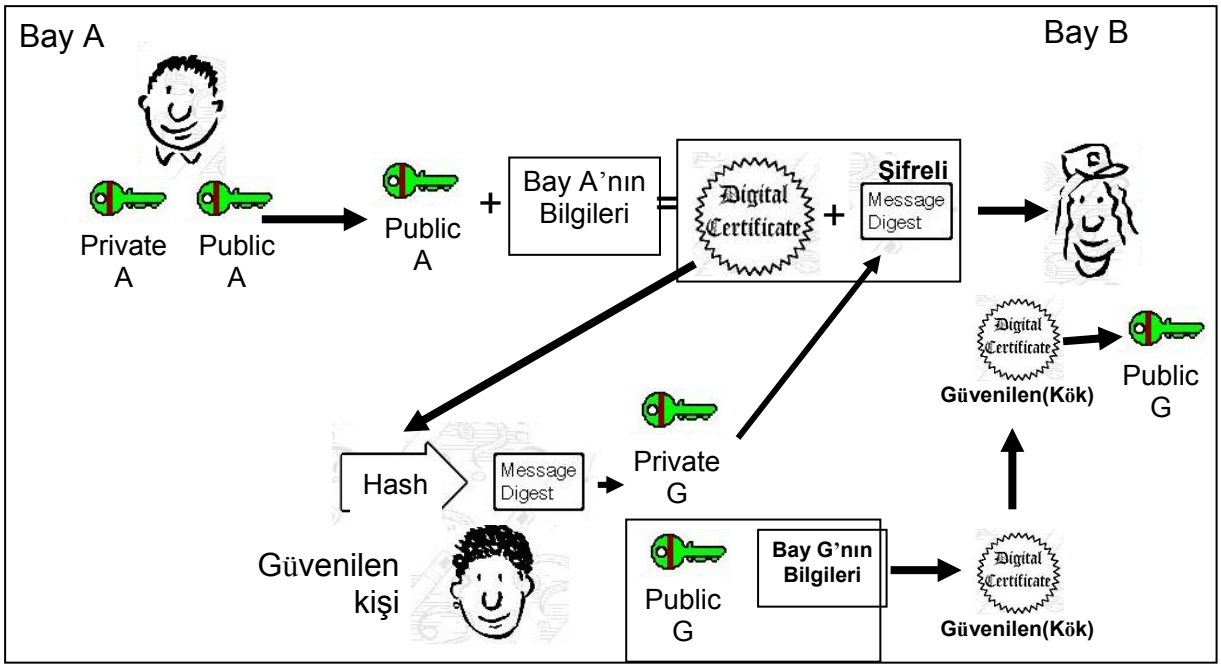
Hash belirli bir matematik fonksiyonu ile verinin tek yönlü (yani veri geri elde edilemeyecek şekilde) bir kontrol numarası elde etme tekniğidir. Hash kaynağının doğrulanması ve veri bütünlüğünü test etmek için kullanılır.

MD5 ve SHA1 günümüzde kullanılan popüler bir hash algoritmalarıdır. Kimlik doğrulama için EAP tüneli yöntemlerinde ve sertifika imzala amacı ile kullanılmaktadırlar.

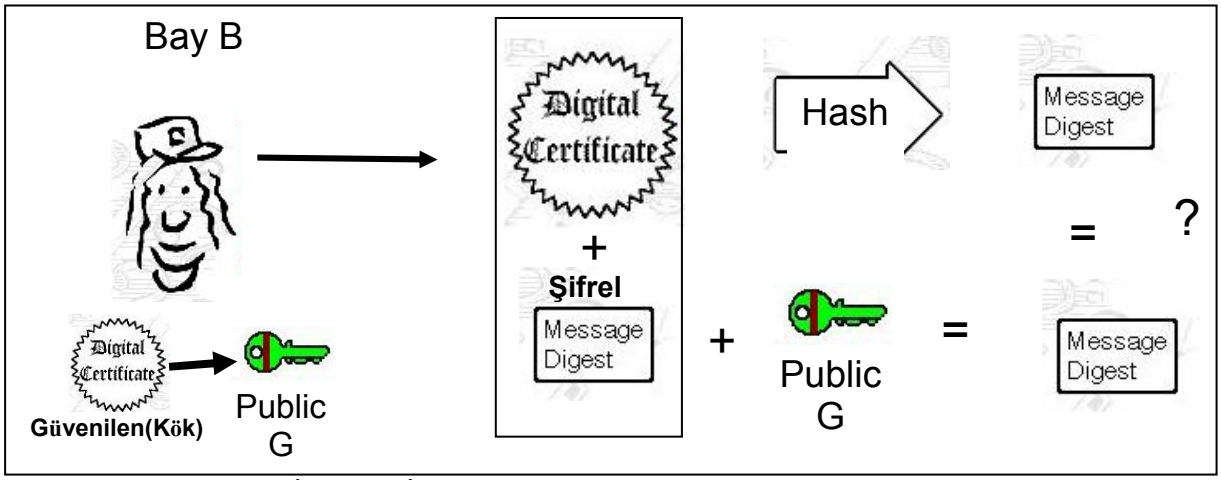
2.6. Kök Sertifika ve İmzalama

İmzalama elinizde bulunan Public anahtarın hizmet almak istediğiniz yetkili kişileremi ait olup olmadığını anlam için kullanılan bir tekniktir. Bunun anlayabilmek için güvenilen üçüncü bir kaynak yardımı ile bu sertifikanın doğru kaynaktan gelip gelmediği sorgulanır.

Bunun için kök sertifikalar kullanılır ve kök sertifikalar genellikle işletim sistemi ile beraber gelir. Verisign, Thawte gibi kurumlar geçerli olup olmağını sorgulayarak geçerli ise sertifikalarını imzalarlar. Bunun için önce alt sertifika'dan hash yardımı ile bir kontrol numarası elde edilir. Bu kontrol numarası da kök sertifikanın private anahtarı ile şifrelenip, sertifikanın altına eklenir. Sertifikayı alan istemci kendisinde bulunan kök sertifikanın public anahtarı ile bu kontrol numarasını çözer ve doğru olup olmadığını kontrol eder. İmzalama işlemi ve daha sonra istemcinin imzayı kontrol aşamaları aşağıdaki şekillerde gösterilmiştir.



Sertifikanın İmzalanması



İmzanın İstemci Tarafından Kontrol Edilmesi

2.7. EAP- TLS (Transport Layer Security)

TLS (İletim Katmanı Güvenliği), Secure Sockets Layer (SSL) in atası olan bir kriptografi protokolüdür.

Bu protokolün amacı haberleşen iki uygulama arasında veri güvenliği ve bütünlüğünün sağlanmasıdır. Protokol iki katmandan oluşur. İlk katman TLS kayıt protokolü, diğeri ise TLS el sıkışma (handshake) protokolüdür. TLS kayıt protokolü ile veriler simetrik şifreleme anahtarları ile şifrelenir. Her bağlantı için farklı bir simetrik şifreleme anahtarı kullanılır. Bu anahtar TLS el sıkışma protokolü kullanılarak alıcı ve verici tarafından paylaşılır. TLS el sıkışma protokolü ile haberleşecek tarafların birbirlerini yetkilendirmeleri, şifreleme algoritması ve anahtarların karşılıklı değişimi sağlanır. Bu çift yönlü doğrulama özelliği ile EAP-TLS en güvenilir EAP yöntemlerinden biri olarak bilinir. Bununla birlikte, her istemciye özgün sertifika üretilerek, güvenli bir şekilde

dağıtılmasını gerektiren bu yöntem, uygulamada getirdiği bu zorluk nedeni ile, çok sık uygulanan bir yöntem değildir.

2.8. Tunneled Transport Layer Security (TTLS) ve Protected EAP (PEAP)

EAP-TLS'e alternatif olan TTLS ve PEAP kimlik denetim sistemlerinde kimlik doğrulama bilgisinin güvenli bir şekilde iletilebilmesi için istemci ile sunucu arasında TLS ile şifreli bir oturum oluşturulur. Sorgulama ve yanıt paketleri, herkese açık olmayan TLS şifreli bir kanal üzerinden gönderilir. Bunun için sucunu için tek bir anahtar çifti oluşturulur. İstemci raslansal oluşturduğu anahtarı kendi public anahtarı ile şifreleyip sunucuya yollar. Bundan sonraki haberleşme bu yeni ortak anahtar ile simetrik şifreleme ile devam eder. Simetrik şifreleme Asimetrik şifreleme tekniklerine göre daha sistem kaynağı tüketir.

Kimlik doğrulama için TTLS yönteminde PAP, CHAP, MS-CHAP ve MS-CHAPv2 gibi kimlik doğrulama yöntemleri kullanılabilir PEAP ile ise sadece MS-CHAPv2 kullanılır.. RADIUS sunucusu, istemcinin kimliğini kendisine gönderdiği kimlik doğrulama verisi ile doğrularken, sunucu da sunucu sertifikası aracılığı ile doğrulama yapmaya yetkili olduğunu kanıtlar. Böylece, tek bir sunucu sertifikası ve kullanıcı adı/parola verisi kullanılarak, çift yönlü doğrulama gerçekleşmiş olur.

2.9. LDAP

LDAP (Lightweight Directory Access Protocol : Hafif Dizin Erişim Protokolü) bir dizin servisi standardıdır. Genel olarak bu dizin ifadesi LDAP'in yapısı ve içerdiği bilgi itibari ile "veritabanı" olarak adlandırılmaktadır. Dizin ifadesi aslında dizinin içerdiği belirli bir sıradaki her bir nesne hakkında bilgi veren bir liste seklinde açıklanmaktadır. LDAP istemcisi ldap için yazılmış ve ldap istemcisinin içine gömülmüş API(Application Programming Interface) vasıtasıyla aradığı bilginin formatını oluşturarak TCP/IP vasıtasıyla dizin sunucusuna gönderir. Bu isteği alan dizin sunucusu bu isteği dizini içeren veriyi sorgulayarak gerekli bilgiyi yine ayni yolla istemciye gönderir.

2.10. Authentication, Authorization ve Accounting (AAA)

Bilgisayar ve ağ kaynaklarına erişimin denetlenmesini ve kullanıcı etkinliklerinin izlenmesini sağlayan sistemlerin tümüne verilen addır.

Authentication (Yetkilendirme) : Kullanıcı ya da kullanıcılara sisteme, programa veya ağa erişim hakkinin verilmesidir.
Authorization (Kimlik Doğrulama) : Sunucu, anahtar veya yönlendirici kullanımlarında cihaz ya da kullanıcının kimliğinin onaylanmasıdır.
Accounting (Hesap Yönetimi) : Herhangi bir kullanıcının ne yaptığı, kullanıcı hareketleri kullanıcı veri bağlantıları ve kullanıcı sistem kayıtlarının izlenebilmesi amacıyla yapılan işlemdir.

2.11. RADIUS

RADIUS (Remote Authentication Dial-In User Service: Uzak Kimlik- Doğrulama Çevirmeli Erişim Kullanıcı Hizmeti) kimlik kanıtlama için bir ağ protokolüdür.

Genelde bir RADIUS sunucusu, AAA görevleri gerçekleştirmek için kullanılır. AAA evreleri aşağıdaki gibidir:

Kimlik doğrulama evresi: Bir kullanıcı adı ve parolasını yerel bir veri tabanında doğrular. Kimlik bilgileri doğrulandıktan sonra yetkilendirme işlemi başlar.

Yetkilendirme evresi: Bir isteğin bir kaynağa erişmesine izin verilip verilmeyeceğini belirler. Çevirmeli istemci için bir IP adresi atanır.

Hesap yönetimi evresi: Kaynak kullanımına ilişkin bilgiler toplanarak eğilim analizi, denetim, oturum zamanı faturalama ya da masraf hesabı işlemlerinde kullanılır.

Bir RADIUS sunucu NAS'a üç cevaptan birini döndürür. "Nay" (Access Reject), "Challenge" (Access Challenge) veya "Yea" (Access Accept) dir.

3. FreeRADIUS

FreeRADIUS tamamen GPL altında lisanslanmış bir RADIUS sunucu uygulamasıdır. Kimlik kanıtlama mekanizmalarını geniş çapta desteklemektedir. Bu doküman kapsamında FreeRADIUS kurulumu Fedora Core 8 ile gerçekleştirilmiştir. Başka dağıtımlarda klasör yerleri değişik olabilir.

3.1. FreeRADIUS Version 2.0.3 ile PEAP – TTLS Kimlik Denetimi

3.1.1. FreeRADIUS 2.0.3'un Kurulumu

Sunucu kurulmadan önce kurulumun yapılacağı Linux'te opensll'in yüklü olduğu kontrol edilmelidir ve kurulu değilse önce OpenSSL kurulmalıdır. Ancak günümüzde bir çok dağıtım OpenSSL ile beraber gelmektedir. Bunun için "which openssl" komutu ile kurulu olduğu yer tespit edilebilir.

Daha sonra kurulum için aşağıdaki komutlar sırası ile girilmelidir.

```
#./configure
#make
#make install
```

3.1.2. Sertifika Zincirinin Oluşturulması

Freeradius kurulduktan sonra test amaçlı sertifiklar basar. Bu sertifikaların test dışında kullanılması tavsiye edilmez. Sertifikalar Freeradius klasörünün altındaki "certs" İsimli klasörde (/usr/local/etc/raddb/certs) durur. Bir karışıklıl olmasın diye bu klasorun altındaki test sertifikaları silinir. ("rm -f *.pem *.der *.csr *.crt *.key *.p12 serial* index.txt*") Certs klasöründe bulunan "ca.cnf" dosyasında kök sertifika için gereken tanımlamalar, "server.cnf" dosyasında ağ denetimi için kullanacağımız sertifika tanımlamaları bulunur. Burada ki değerler ihtiyaca göre değiştirilmelidir.

```
[ CA_default ]
default_days            = 365   (Kök sertifikanın geçerlilik süresi)
default_md              = md5  (Kullanılacak hash fonksyonu)

[ req ]
default_bits            = 2048           (Anahtar bit uzunlugu)
input_password          = whatever  (Anahtar şifresi)
output_password         = whatever (Anahtar şifresi)

[certificate_authority]
countryName             = TR       (Kalanlar Kurum ile ilgili bilgiler)
stateOrProvinceName     = KONYA
localityName            = Alaaddin Tepesi
organizationName        = ULAK-CSIRT
emailAddress            = akingok@itu.edu.tr
commonName              = "CSIRT Certificate Authority"
```

Gereken değişiklikler yapıldıktan sonra "bootstrap" script'i ile gereken sertifikaların basılması sağlanır.

3.1.3. Freeradius Konfigurasyon Dosyalarında Yapılacak Değişiklikler

Öncelikle sertifika oluşturulurken belirtilen parola eap.conf dosyasındaki tls bölümde belirtilir.

```
tls {
private_key_password = whatever
}
```

Freeradius 2 ile beraber "Virtual Servers" desteği gelmiştir. Bu sayede aynı Freeraidus sunucusu ile farklı IP ve Portlardan farklı profillerde sunucu hizmeti verilebilir. Bu aşamada burada böyle bir desteğe ihtiyaç olmadığından "eap.conf" klasöründe aşağıdaki şekilde bu özellik kapatılır.

```
peap {
#       virtual_server = "inner-tunnel"
}
ttls{
#       virtual_server = "inner-tunnel"
}
```

Bu durumda sunucu "sites-available" klasörünün altında bulunan "default" dosyasındaki konfigürasyon ile çalışır.

3.1.4. Kullanıcı Listesinin Radius üzerinde oluşturulması

Harici bir kaynakta değil de Radius sunucusu üzerinde kullanıcı verileri tutulacaksa "users" dosyasının içersine aşağıdaki gibi kullanıcılar eklenir.

```
kullanıcıadı    Cleartext-Password := "parola1234"
```

3.1.5. Sorguda bulunacak Cihazların Tanıtılması

"clients.conf" dosyasında sorguda bulunacak bütün cihazların (switch'lerin ve kablosuz erişim cihazlarının) IP adresleri ve karşılıklı belirlenmiş parola değerleri girilir.

```
client 192.168.1.60 {
secret = 1234
shortname =test_switch
}
```

3.1.6. FreeRADIUS Sunucusunun Çalıştırılıması

Sunucu başlangıçta test amaçlı ve detaylı log verecek şekilde "radiusd –X" komudu sunucu başlatılabilir. Daha sonra istenirse işletim sistemine bir servis olarak eklenebilir.

3.2. FreeRADIUS Version 1.1.7 ile PEAP – TTLS Kimlik Denetimi

Asılında yeni versiyon FreeRADIUS kullanılabilir. Son versiyonda bir çok şey baştan ayarlı ve otomatik hale geldiği için pek bir değişikliğe gerek kalmamıştır. Bu bölüm daha çok sertifika sisteminin kurulum detaylarının göstermek için incelenmiştir.

3.2.1. FreeRADIUS 1.1.7'un Kurulumu

Kurulumu FreeRADIUS 2.0.3 ile aynı şekildedir.

3.2.2. Sertifika Zincirinin Oluşturulması

3.2.2.1. Ön hazırlıklar

Ilk olarak openssl.cnf dosyasında sertifikaların oluşturulacak dizin yeri freeradius'un kurulduğu dizin olarak gösterilir.

```
[ CA_default ]
dir             = /usr/local/etc/raddb/certs    # Where everything is kept
certs           = $dir                  # Where the issued certs are kept
```

Not: /usr/local/etc/raddb/certs adresi dağıtımdan dağıtıma değişebilir.

Bu değişikliğin ardından FreeRADIUS'un ./certs klasöründe aşağıdaki değişiklikler yapılır.

```
cd /usr/local/etc/raddb/certs/
mkdir private     #  private anahtarların barınacağı klasör
mkdir newcerts    #Kök sertifika ile imzalanacak sertifikaların bulunacağı
klasör
touch index.txt         #Imzalan sertifikaların index dosyası oluşturulur.
```

Daha sonra serial diye bir dosya oluşturulup içine "00" değeri girilir, bu değer imzalan sertifikaların seri numarasını tutacaktır.

Windows XP işletim sisteminde ekstradan basılacak sertifikaya "object identifier" değerinin girilmesi gerekmektedir. Bu değerlerin kullanılabilmesi için xpextensions isminde bir dosya oluşturup aşağıdaki değerler girilir.

```
[ xpclient_ext]
extendedKeyUsage = 1.3.6.1.5.5.7.3.2

[ xpserver_ext ]
extendedKeyUsage = 1.3.6.1.5.5.7.3.1
```

3.2.2.2. Kök Sertifikanın Oluşturulması

Gereken eklemeler yapıldıktan sonra önce Kök Sertifika oluşturulur. Kök sertifika daha uzun geçerlilik süresi ile oluşturulmalıdır. Bu sertifika radius sunucusun kullanacağı sertifikayı imzalamak amacı ile oluşturulur. Bu sertifika PKI zincirinizi tepesi olacaktır ve kullanıcılarınızı bir web sayfası aracılığı ile dağıtılabilir. (Örnek pki.itu.edu.tr)

Bu komut ile 3650 gün geçerli bi sertifika oluşturulur. Buradaki "**cakey.pem**" kök private keyi, "cacert.pem" ise kök sertifikanın adıdır.

```
#openssl req -config /etc/pki/tls/openssl.cnf -new -x509 -keyout
private/cakey.pem -out cacert.pem -days 3650
```

Not: /etc/pki/tls/openssl.cnf adresi dağıtımdan dağıtıma değişebilir.

Komut girildikten sonra "PEM pass phrase" yani private anahtar ile imzalama yapılması gerektiğinde belirtilmesi gereken özel şifre belirtilir. Daha sonrada sırası ile kurum ile ilgili aşağıdaki değişkenler belirlenmelidir.

```
Country Name (2 letter code) [GB]:TR
State or Province Name (full name) [Berkshire]:Istanbul
Locality Name (eg, city) [Newbury]:Maslak
Organization Name (eg, company) [My Company Ltd]:ITU
Organizational Unit Name (eg, section) []:CSIRT
```

```
Common Name (eg, your name or your server's
hostname)[]:agdenetim.test.itu.edu.tr
Email Address []: radius@itu.edu.tr
```

Not: Kök private anahtar ile gereken imzalama yapıldıktan sonra güvenli başka bir yerde saklanması tavsiye edilir.

Sertifikanın bu halini Linux istemciler de kullanılabilir. Ancak Windows istemcilerde daha kolay kurulum yapılabilmesi için aşağıdaki komut ile sertifika formatını değiştirilmelidir. Yeni oluşan cacert.crt dosyası windows kullanıcılar tarafında kullanılacaktır.

```
#openssl x509 -in cacert.pem -out cacert.crt
```

3.2.2.3. Sunucu Sertifikasının oluşturulması

Sunucu sertifikası daha kısa süreler için oluşturulur (bu uygulamada 365 gün belirlenmiştir) ve belirlenen aralıklarda anahtar değiştirilir. İstemciye kök sertifika yüklendiği için bu işlem sonrası müdahaleye gerek kalmaz. Bu komut sonrası serverkey.pem ile private anahtar, serverreq.pem ile ise kök sertifika ile imzalanacak sertifika oluşturulur.

```
#openssl req -config /etc/pki/tls/openssl.cnf -new -keyout serverkey.pem -
out serverreq.pem -days 365
```

Komut girildikten sonra yine “PEM pass phrase” yani private anahtar şifresi ve kurum ile ilgili değişkenler belirlenir.

3.2.2.4. Sunucu Sertifikasının Kök Sertifika ile İmzalanması

Komut girildikten sonra kök sertifikanın “pass phase” değeri istenir. Komut sonrası server.pem adına sunucu sertifikası son halini almış olur.

```
#openssl ca -config /etc/pki/tls/openssl.cnf -policy policy_anything -out
server.pem -extensions xpserver_ext -extfile ./xpextensions
```

3.2.3. Freeradius Konfigurasyon Dosyalarında Yapılacak Değişiklikler

radiusd.conf dosyasında,

PEAP için “$INCLUDE ${confdir}/eap.conf” başlığı altındaki değerleri şu şekilde değiştirilmelidir.

```
mschap {
  authtype = MS-CHAP
  use_mppe = yes
  require_encryption = yes
  require_strong = yes
}
```

Authorize ve Authentice başlıkları altında aşağıdaki değerler aktif olmalıdır.

```
authorize {
    preprocess
    mschap
```

```
    suffix
    eap
    files
}
authenticate {
      Auth-Type MS-CHAP {
            mschap
      }
      eap
  }
```

TTLS için ise Authorize ve Authentice başlıkları altında aşağıdaki değerler aktif olmalıdır.

```
authorize {
    preprocess
    suffix
    eap
    files
}
authenticate {
      eap
  }
```

eap.conf dosyasında "default_eap_type" md5'dir. Bu değere değeri değiştirmek mecburi değildir. RADIUS sunucusu sırası ile kimlik denetimi yapılabilecek bütün teknikleri (eap-md5,tls,ttls,peap,pap,chap..vs) dener. Bu değeri kullanacağınız teknik ile değiştirmeniz sorguyu biraz hızlandıracaktır.
PEAP için:

```
default_eap_type = peap
```

TTLS için:

```
default_eap_type = ttls
```

Hem PEAP hemde TTLS için gereken sertifika ayarları TLS başlığı altında girilir.

```
tls {
# Sunucun Sertifikasi oluşturulurken girilen pass phrase
private_key_password = whatever
# Sunucu Private Key Dosyasının Adı
private_key_file = ${raddbdir}/certs/serverkey.pem
# Sunucunun Sertifika Dosyasının Adı
certificate_file = ${raddbdir}/certs/server.pem

#  Kök Sertifika Dosyasının Adı
CA_file = ${raddbdir}/certs/ cacert.pem

dh_file = ${raddbdir}/certs/dh
random_file = ${raddbdir}/certs/random

fragment_size = 1024
 }
```

PEAP için PEAP başlıkları altında aşağıdaki değerler aktif olmalıdır.

```
peap {
```

```
default_eap_type = mschapv2
copy_request_to_tunnel = no
use_tunneled_reply = no
}
```

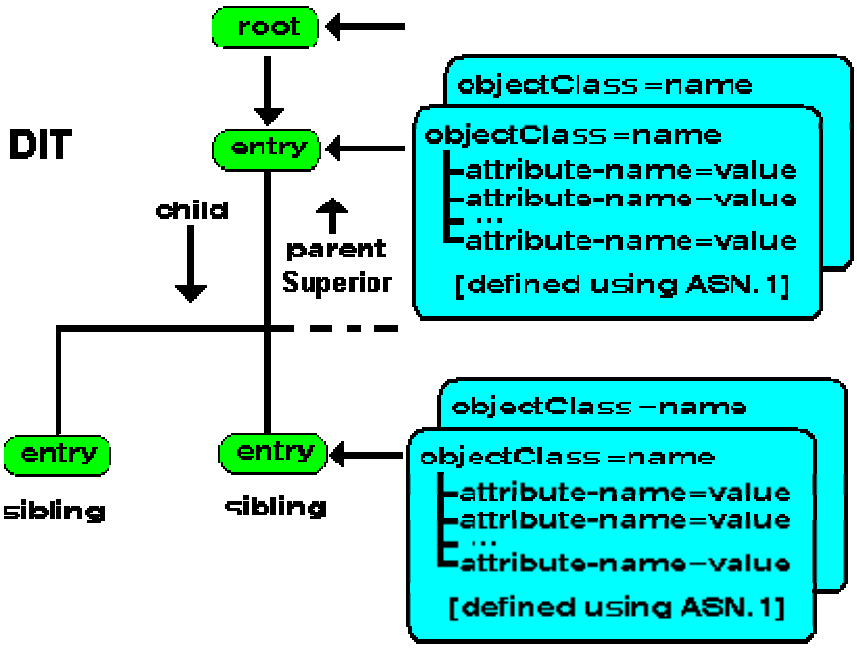
TTLS için ise TTLS başlıkları altında aşağıdaki değerler aktif olmalıdır.

```
ttls {
default_eap_type = md5
copy_request_to_tunnel = no
use_tunneled_reply = no
}
```

4. OpenLDAP

Yukarıdaki tanımlarda LDAP in barındırdığı veriler itibari ile bir veritabanı olduğu vurgulanmıştı. Bu aşamada dizin(directory) ve veritabanı(database) arasındaki farkı açıklamakta fayda vardır. Veritabanı tasarımında amaç hem HIZLI yazmak hemde mümkün olduğunca HIZLI okumaktır. LDAP da ki ana amaç aranan verinin mümkün olan en kısa sürede bulunmasıdır. Bu ana özelliğinden dolayı kimlik denetimi uygulamalarda verilerin saklanması için LDAP en çok kullanılan yöntemlerden biridir. Ücretsiz LDAP uygulamalarına OpenLDAP, Fedora Directory Server, OpenDS, ApacheDS örnek verilebilir. Bu çalışmada bir açık kaynak kodlu LDAP uygulaması olan OpenLDAP'a yer verilmiştir.

LDAP'da veriler hiyerarşik nesneler şeklindedir ve bu nesneler giriş (entry) olarak adlandırılır. Bu nedenle ağaç yapısı şeklinde olan bu yapıya Data Information Tree (DIT) denir. Her bir giriş ağaç da bir ana girişe ve bir yada daha fazla çocuk girişe(nesneye) sahiptir. Tüm bu veri bilgi ağacının tepesinde ise kök (root) vardır.



Data Information Tree'deki her bir entry entry'i açıklayan bir ve/veya daha fazla "attribute" icerir ve her bir attribute bir "type" ve "value" icermektedir.

Bir "objectclass" bir entry içinde bulunabilecek attribute 'ları tanımlar. LDAP dizinine bu entry'ler LDIF (LDAP Data Interchange Format) girdi dosyası ile eklenir. Objectclass'ların tanımları schema dosyalarında tanımlanmaktadır.

Şart olmamakla birlikte genellikle ağaç yapısının tepe noktası yani kök 'o' (organization)'dur. Daha altında genellikle 'ou' (organizational unit)'ler bulunur. Her organization'un altında çeşitli 'cn' (common name)'ler bulunur. Bir ou'nun altına başka bir ou konabilir.

OpenLDAP uygulaması öntanımlı olarak kurulduğunda yapılandırma dosyaları Linux sistemlerde "/etc/openldap", BSD sistemlerde "/usr/local/etc/openldap" klasöründe bulunur. Düzenlenecek olan ilk yapılandırma dosyası "slapd.conf" dir.

Örnek slapd.conf yapılandırma dosyası açıklamaları ile birlikte aşağıdaki şekildedir.

#slapd.conf

```
include         /usr/local/etc/openldap/schema/core.schema
include         /usr/local/etc/openldap/schema/cosine.schema
```

#bu iki direktif LDAP sunucumuzun şemasını belirecektir

```
pidfile         /var/run/openldap/slapd.pid
argsfile        /var/run/openldap/slapd.args
```

"pidfile" direktifi OpenLDAP (slapd)'ın PID'i nereye yazacağını tanımlar.
#"argsfile" direktifi, OpenLDAP'ın komut satırında hangi parametre ile çalışacağını belirler.
#içerigi
/usr/local/libexec/slapd -h ldapi://%2fvar%2frun%2fopenldap%2fldapi/ ldap://0.0.0.0/ -u
#ldap -g ldap
#şeklindedir.

```
modulepath      /usr/local/libexec/openldap
moduleload      back_bdb
```

#"modulepath" direktifi, OpenLDAP tarfından yüklenebilir modüllerin (overlays) yerini
#gösterir. "moduleload" direktifi ile verilen tanım bu dizinde olmalıdır. Burada "back_bdb"
#kullanılmıştır. Bu şekilde "Berkeley Database" kullanılabilecektir.

```
database bdb
suffix "dc=marmara,dc=edu,dc=tr"
rootdn "cn=root,dc=marmara,dc=edu,dc=tr"
rootpw {SSHA}K73K9RFa7ti+Dz+RpCyG9L6M0YXyb5SE
```

"database" direktifi Data Information Tree (DIT)'nin ne şekilde yapılandırılacağını belirler.
"suffix" direktifi veritabanında tutulacak Data Information Tree (DIT)'nin hiyerarşisini
#yada Distinguished Name'in en üst düğüm noktasını belirler.
"rootdn" en üst düğüm noktasını "rootpw" direktifinde verilen şifre ile erişebilecek süper
#kullanıcı tanımını belirler.
"rootpw" direktifi süper kullanıcının (burada root) veritabanına erişimi için kullanılır.
komut satırında "slappasswd -h {SSHA} -s konya" kullanılarak hash li bir şifre
#oluşturulabilir.

```
directory       /var/db/openldap-data
index           objectClass     eq
index           uid             eq
```

"directory" direktifi veritabanının hangi dizinde tutulacağını belirler
"index" direktifi ile hangi alanların indekslemesi yapılacağı belirlenir. Bu şekilde daha hızlı
sorgulama yapılabilir.

Bu tanımlamalar ile artık OpenLDAP veri girilebilir ve sorgulanabilir hale gelmiştir. İlk olarak ağaç yapısının oluşturulması gerekir. Aşağıdaki LDAP Data Interchange Files (LDIF) formatındaki girdi ile bu oluşturulabilir.

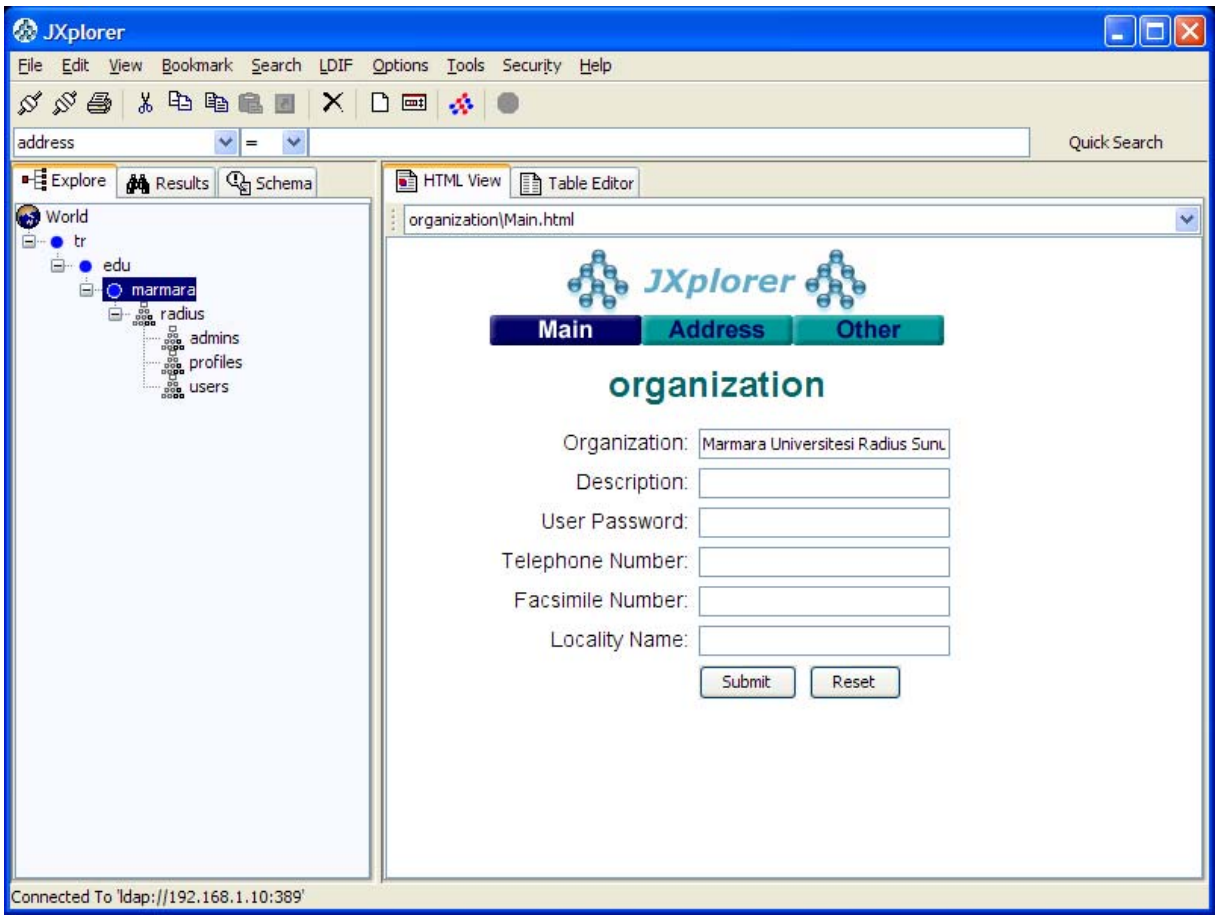
Base.ldif

```
dn: dc=marmara,dc=edu,dc=tr
objectclass: dcObject
objectclass: organization
o: Marmara Universitesi LDAP Sunucusu
dc: marmara
```

Aşağıdaki komut ile bu tanımlar LDAP sunucusuna eklenebilir.

```
#ldapadd -H ldap://127.0.0.1 -x -D "cn=root,dc=marmara,dc=edu,dc=tr" -f
base.ldif -W
```

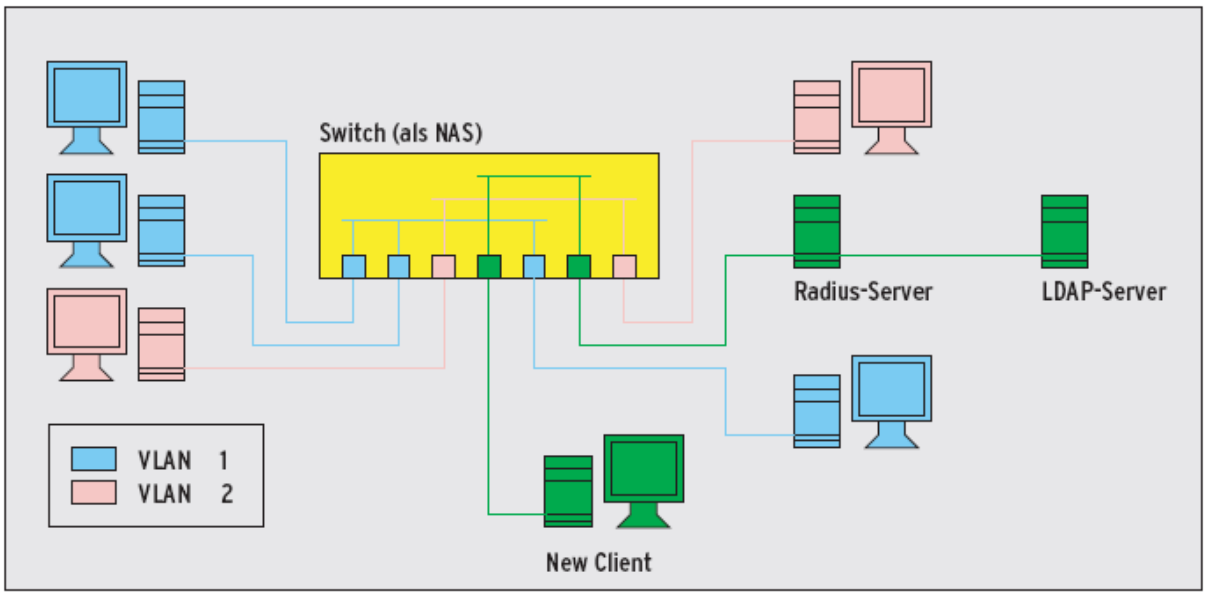
Son olarak artık bir LDAP Browser ile OpenLDAP sunucumuza bağlanabiliriz. Görüntü aşağıdaki gibi olacaktır.



Artık LDAP sunucusu kullanıma hazırdır. Aşağıdaki LDAP Data Interchange Files (LDIF) formatındaki girdiler ile örnek veriler oluşturulabilir.

5. OpenLDAP ↔ FreeRADIUS

FreeRADIUS kimlik doğrulamasında, kullanıcı kimlikleri bilgileri ve erişim tanımları sunucu üzerinde yapılabileceği gibi kullanım kolaylığı sağlayacak LDAP sunucusunda da yapılabilir. LDAP sunucunda daha önceden tanımlanmış kullanıcı tanımlarını da kullanmak mümkündür.



Daha önce kurulan FreeRADIUS'la birlikte gelen RADIUS LDAP şema dosyasını "/usr/local/share/doc/freeradius/ldap_howto.txt" dosyasından düzenleyerek RADIUS-LDAPv3.schema adında OpenLDAP schema dizinine kopyalanması gerekir.

Yeni bir LDAP sunucusu FreeRADIUS için kuruluyorsa bunun için ilk olarak "slapd.conf" dosyasına aşağıdaki girdinin girilmesi gerekir.

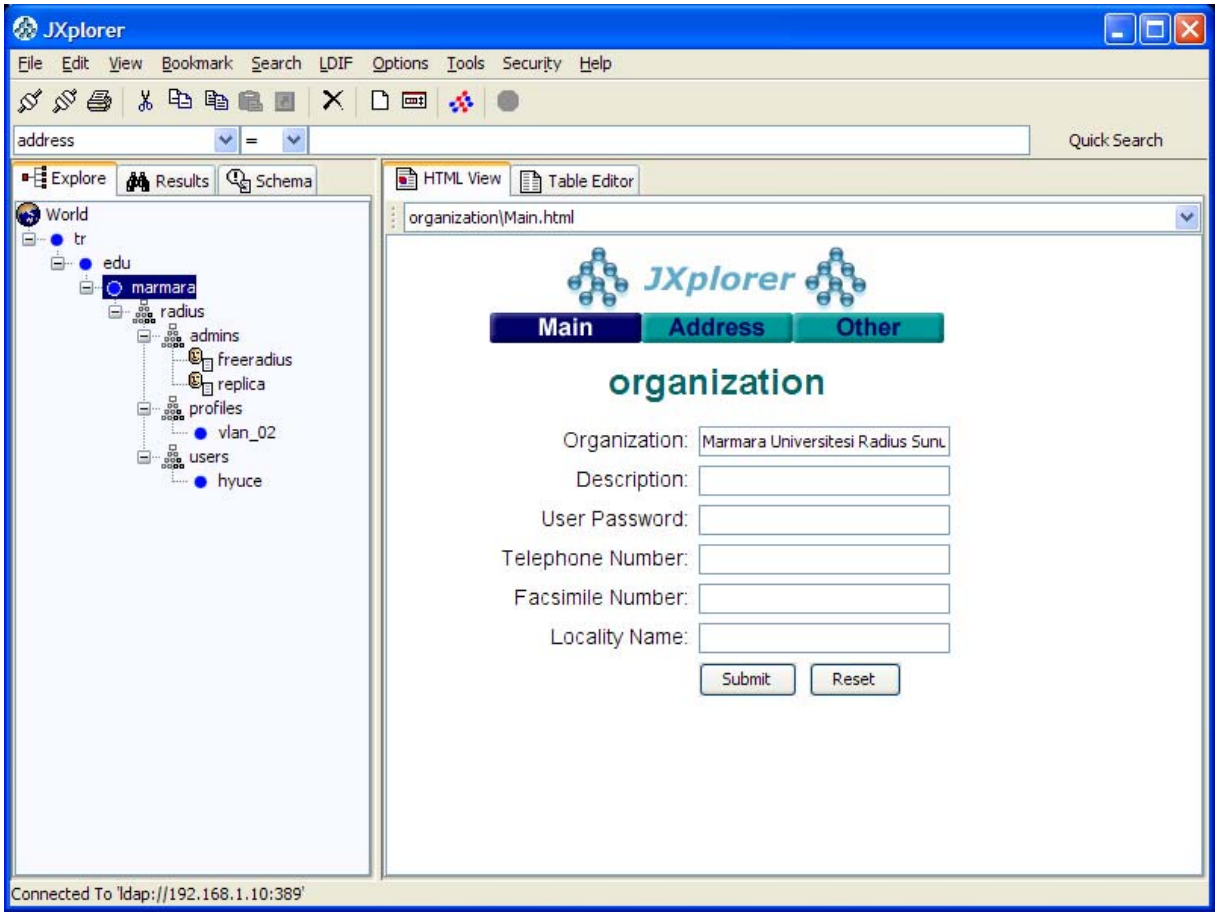
```
include         /usr/local/etc/openldap/schema/RADIUS-LDAPv3.schema
```

LDAP sunucumuzu tekrar başlattığımızda artık FreeRADIUS için kullanıcı tanımları yapılabilir. Bu tanımların bulunduğu örnek "freeradius.ldif" dosya içeriği aşağıda verilmiştir.

freeradius.ldif

```
dn: ou=radius,dc=marmara,dc=edu,dc=tr
objectclass: organizationalunit
ou: radius

dn: ou=profiles,ou=radius,dc=marmara,dc=edu,dc=tr
objectclass: organizationalunit
ou: profiles

dn: ou=users,ou=radius,dc=marmara,dc=edu,dc=tr
objectclass: organizationalunit
ou: users

dn: ou=admins,ou=radius,dc=marmara,dc=edu,dc=tr
objectclass: organizationalunit
ou: admins
```

```
dn: uid=vlan_02,ou=profiles,ou=radius,dc=marmara,dc=edu,dc=tr
uid: vlan_02
radiusTunnelMediumType: IEEE-802
radiusTunnelType: VLAN
radiusTunnelPrivateGroupId: 2
objectClass: radiusprofile

dn: uid=hyuce,ou=users,ou=radius,dc=marmara,dc=edu,dc=tr
objectclass: radiusprofile
uid: hyuce
userPassword: hyuce
radiusGroupName: vlan_02

dn: cn=freeradius,ou=admins,ou=radius,dc=marmara,dc=edu,dc=tr
objectclass: person
sn: freeradius
cn: freeradius
userPassword: freeradius

dn: cn=replica,ou=admins,ou=radius,dc=marmara,dc=edu,dc=tr
objectclass: person
sn: replica
cn: replica
userPassword: replica
```



Aşağıdaki komut ile bu tanımlar LDAP sunucusuna eklenebilir.

```
#ldapadd -H ldap://127.0.0.1 -x -D "cn=root,dc=marmara,dc=edu,dc=tr" -f
freeradius.ldif -W
```

Ldapsearch komutu ile girdiler kontrol edilebilir.

```
#ldapsearch -x -b "ou=radius,dc=marmara,dc=edu,dc=tr" "(uid=hyuce)"

# hyuce, users, radius, marmara.edu.tr
dn: uid=hyuce,ou=users,ou=radius,dc=marmara,dc=edu,dc=tr
objectClass: radiusprofile
uid: hyuce
radiusGroupName: vlan_02
userPassword:: aHl1Y2U=
```

OpenLDAP sunucu için bunlar yeterlidir. FreeRADIUS için LDAP desteğinin eklenmesi gerekmektedir. Çalışmanın bu kısmında FreeRADIUS 2.0.3 sürümü kullanılmıştır. Radius sunucu için kullanacağımız yapılandırma dosyaları "radiusd.conf" , "eap.conf", "users" , "clients.conf" ve raddb/certs dizindeki "ca.cnf" , "client.cnf" , "server.cnf" dosyalarıdır.

Sertifikaların oluşturulması için raddb/certs dinindeki sertifika bilgilerin isteğe göre düzenlenebilir. Bu yapılandırma dosyalarında ki "input_password" ve "output_password" girdileri daha sonra kullanılacağından değiştirilmesi uygun olacaktır. Bu değişikliklerden sonra sertifika oluşturmak için "make" komutunu kullanarak sertifikaların oluşturulması sağlanır.

…. Kesildi

```
Using configuration from ./server.cnf
Check that the request matches the signature
Signature ok
Certificate Details:
        Serial Number: 1 (0x1)
        Validity
            Not Before: Apr 29 18:17:49 2008 GMT
            Not After : Apr 29 18:17:49 2009 GMT
        Subject:
            countryName               = TR
            stateOrProvinceName       = Goztepe
            organizationName          = Marmara Universitesi
            commonName                = MARUN Server Certificate
            emailAddress              = huseyin@marmara.edu.tr
        X509v3 extensions:
            X509v3 Extended Key Usage:
                TLS Web Server Authentication
Certificate is to be certified until Apr 29 18:17:49 2009 GMT (365 days)
```

…. Kesildi

FreeRADIUS rlm_ldap modülü kullanarak LDAP kullanılabilir. Bunun için yapılandırma dosyalarında aşağıdaki değişikliklerin yapılması yeterli olacaktır.

radiusd.conf

…. Kırpıldı

```
modules {

$INCLUDE eap.conf
        # Lightweight Directory Access Protocol (LDAP)
        #
        ldap {
                server = "127.0.0.1"
    #identity = "cn=freeradius,ou=admins,ou=radius,dc=marmara,dc=edu,dc=tr"
        #       password = freeradius
                basedn = "ou=radius,dc=marmara,dc=edu,dc=tr"
                filter = "(uid=%{Stripped-User-Name:-%{User-Name}})"
        #       base_filter = "(objectclass=radiusprofile)"

                tls {
                        start_tls = no

                }
                dictionary_mapping = ${confdir}/ldap.attrmap
                password_attribute = userPassword

        }
```

…. Kırpıldı

```
authorize {
    eap
    suffix
    ldap
}

authenticate {
        eap
}
```

…. Kırpıldı

users

```
DEFAULT Auth-Type := LDAP
        Fall-Through = 1
```

eap.conf

```
        eap {
                default_eap_type = ttls
                timer_expire     = 60
                ignore_unknown_eap_types = no
                cisco_accounting_username_bug = no

                ## EAP-TLS
                tls {
                        certdir = ${confdir}/certs
                        cadir = ${confdir}/certs
                        private_key_password = marmara
                        private_key_file = ${certdir}/server.pem
                        certificate_file = ${certdir}/server.pem
                        CA_file = ${cadir}/ca.pem
                        dh_file = ${certdir}/dh
                        random_file = ${certdir}/random
                        make_cert_command = "${certdir}/bootstrap"
```

```
                }
                ttls {
                        default_eap_type = md5
                        copy_request_to_tunnel = no
                        use_tunneled_reply = yes
                }
                peap {
                        default_eap_type = mschapv2
                        copy_request_to_tunnel = no
                        use_tunneled_reply = no
                }
        }
```

clients.conf

```
client localhost {
        ipaddr = 127.0.0.1
        secret = testing123
        shortname = localhost
        require_message_authenticator = no
        nastype     = other     # localhost isn't usually a NAS...
}
```

Artik FreeRADIUS kullanım için hazırdır. Aşağıdaki komutlar kullanarak test edilebilir.

```
# radtest hyuce "hyuce" localhost 1 testing123
Sending Access-Request of id 241 to 127.0.0.1 port 1812
        User-Name = "hyuce"
        User-Password = "hyuce"
        NAS-IP-Address = 192.168.1.10
        NAS-Port = 1
rad_recv: Access-Accept packet from host 127.0.0.1 port 1812, id=241,
length=20

#radtest hyuce "hyucex" localhost 1 testing123
Sending Access-Request of id 217 to 127.0.0.1 port 1812
        User-Name = "hyuce"
        User-Password = "hyucex"
        NAS-IP-Address = 192.168.1.10
        NAS-Port = 1
rad_recv: Access-Reject packet from host 127.0.0.1 port 1812, id=217,
length=20
```

6. KABLOLU ve KABLOSUZ ÖRNEK AĞ CİHAZI TANIMLARI

Bu bölümde kablolu ağ cihazlarında yapılabilecek tanımlamalardan örnek verilecektir. Genel itibari ile bu tanımlamalar aynı kalmakla beraber komutlar üreticiden üreticiye değişecektir.

Bu yapılandırmalar komut satırından (CLI) yapılabileceği gibi Web tabanlı sistemlerden de yapılabilmektedir. Çoğu sistemlerde Web tabanlı tanımlamaları yapmak kolaylık sağlayacaktır. Daha sonra eğer çoklu sistemlere uygulanması gerekiyorsa CLI tabanlı sistem devreye girebilir.

İlk önce bir adet Cisco anahtarın yapılandırmasına bakacağız. Daha sonra HP anahtar yapılandırmasını ele alacağız. Bu yapılandırmalar kablolu sistemlere örnek olacaktır. Son olarak da kablosuz yapılara örnek olarak 1 adet Cisco marka kablosuz erişim cihazını ele alacağız.

6.1. Cisco 2950 Ethernet Anahtarı

Radius sunucusunu anahtar cihaza tanımlayalım:

```
#radius-server host <radius sunucu ip adresi> auth-port 1812 acct-port 1813
timeout 3
#radius-server retransmit 3
#radius-server attribute nas-port format b
#radius-server key 7 <sunucuya özel anahtar kelime>
```

Burada Radius sunucusunun portlarının 1812 ve 1813 olarak tanımlandığını varsaydık. Eski sistemlerde bu rakamlar değişebilmektedir. Bu sistemler UDP port 1645 ve 1646 kullanmaktadırlar. Radius ile cihazlar arasındaki uyum muhakkak gözetilmelidir. FreeRADIUS UDP port 1812 ve 1813. portları kullanmaktadırlar.

Kullanıcı doğrulama yöntemini belirtelim:

```
#dot1x system-auth-control
#aaa authentication dot1x default group radius
#aaa accounting dot1x default start-stop group radius
#aaa accounting system default start-stop group radius
```

Portlara tanımın girilmesi:

```
#dot1x port-control auto
```

6.2. HP-Procurve 2650 Ethernet Anahtarı

Radius sunucusunu anahtar cihaza tanımlayalım:

```
#radius host <radius sunucu ip adresi> key <sunucuya özel anahtar kelime>
```

Kullanıcı doğrulama yöntemini belirtelim:

```
#aaa port-access authenticator active
#aaa authentication port-access eap-radius
```

Portlara tanımın girilmesi:

```
#aaa port-access authenticator <port listesi> control auto
```

6.3. Cisco 1130 Kablosuz Erişim Cihazı

Radius sunucusunu anahtar cihaza tanımlayalım:

```
#radius-server host <radius sunucu ip adresi> auth-port 1812 acct-port 1813
key 7 <sunucuya özel anahtar kelime>
```

Kullanıcı doğrulama yöntemini belirtelim:

```
#aaa group server radius rad_eap1
 server <radius sunucu ip adresi> auth-port 1812 acct-port 1813
#aaa group server radius rad_acct1
 server <radius sunucu ip adresi> auth-port 1812 acct-port 1813

#aaa authentication login eap_methods1 group rad_eap1
#aaa accounting network acct_methods1 start-stop group rad_acct1
```

Yayın tanımın girilmesi:

```
#dot11 ssid <yayın adı>
   vlan <vlan no>
   authentication open eap eap_methods1
   authentication network-eap eap_methods1
   authentication key-management wpa
   accounting acct_methods1
   guest-mode
   mbssid guest-mode
```

Diğer parametreler:

```
#dot11 aaa csid ietf

#interface Dot11Radio0

      no ip address
      no ip route-cache
      encryption vlan  <yayın no> mode ciphers tkip
      ssid eduroam
      speed basic-1.0 basic-2.0 basic-5.5 basic-6.0 basic-9.0 basic-11.0
      basic-12.0 basic-18.0 basic-24.0 basic-36.0 basic-48.0 basic-54.0
      channel 2412
      station-role root access-point
      world-mode dot11d country TR both
      bridge-group 1
      bridge-group 1 block-unknown-source
      no bridge-group 1 source-learning
      no bridge-group 1 unicast-flooding
      bridge-group 1 spanning-disabled

#interface Dot11Radio0.<yayın no>

       encapsulation dot1Q <yayın no>
       no ip route-cache
       bridge-group <yayın no>
       bridge-group <yayın no> subscriber-loop-control
       bridge-group <yayın no> block-unknown-source
```

```
       no bridge-group <yayın no> source-learning
       no bridge-group <yayın no> unicast-flooding
       bridge-group <yayın no> spanning-disabled

#interface FastEthernet0

       no ip address
       no ip route-cache

#interface FastEthernet0. <yayın no>

       encapsulation dot1Q <yayın no>
       no ip route-cache
       bridge-group <yayın no>
       no bridge-group <yayın no> source-learning
       bridge-group <yayın no> spanning-disabled
```

Dikkat ederseniz bu son yapılandırma VLAN yani sanal ağ ile yapılmıştır. Bu parametrelerin ayarlanması çok zor olmamakla beraber sadece hatırlatmak amaçlı olarak konulmuştur. Daha önce de belirtildiği gibi Web ara yüzünden yapılan yapılandırmalar daha sonra pek çok kolaylık sağlayacaktır. Toplu işlemlerde CLI kullanımına istenildiği zaman geçilebilir.

7. KAYNAKÇA

[1]. ULAK-CSIRT Blog Web Site, http://blog.csirt.ulakbim.gov.tr/?p=52#more-52, IEEE 802.1x ve Kurulumu

[2]. Enderunix Web Site, http://www.enderunix.org/docs/ldap_fundamentals, LDAP Nedir

[3]. Wikipedia Web Site, http://en.wikipedia.org/wiki/AAA_protocol, AAA protocol

[4]. Wikipedia Web Site, http://en.wikipedia.org/wiki/RADIUS, RADIUS

[5]. RFC-3580, http://tools.ietf.org/html/rfc3580, IEEE 802.1x Remote Authentication Dial In User Service (RADIUS) Usage Guidelines

[6]. Kampus Ağlarında Açık Kaynak Kodlu Yazılımlar İle 802.1x Uygulamaları, Doğu Teknik Üniversitesi, Bilgi İşlem Daire Başkanlığı, Akademik Bilişim 2007, Kütahya

[7]. OpenLDAP Web Site http://www.openldap.org/, *OPENLDAP Project*

[8]. Belgeler Web Site, http://www.belgeler.org/howto/p8021x-howto-intro.html, 802.1x Port Tabanlı Kimlik Kanıtlama

[9]. E-Dönüşüm Türkiye Projesi Birlikte Çalışabilirlik Esasları Rehberi, DPT Müsteşarlığı

[10]. Kampüs Ağı Yenileme Sürecindeki Çalışmalar ve Dinamik VLAN yapısına Geçiş, Aydın Mutlu, Akademik Bilişim 2007, Kütahya

[11]. Microsoft Help and Support WebSite, Certificate requirements when you use EAP-TLS or PEAP with EAP-TLS, http://support.microsoft.com/kb/814394/en-us

[12]. Linux juarnal Web Site, http://www.linuxjournal.com/article/8017

[13]. LDP Web Site, http://www.tldp.org/HOWTO/SSL-Certificates-HOWTO/index.html

[14]. Securing Network Access with 802.1X, Radius, and LDAP, Mıchael Schwartzkopff, Linux Magazine

[15]. http://www2.itu.edu.tr/~akingok