Bu rehberde apache2 nin Ubuntu işletim sistemi üzerine yüklenilmesini ve sıkılaştırılmasını anlatacağız.

1. Yükleme ve Konfigrasyon

ubuntu# sudo apt-get update

ubuntu# sudo apt-get install apache2 -y

Apache2 yüklendikten sonra Ubuntu Firewall üzerinde izin vermemiz gerekmektedir. Bunun için;

ubuntu# sudo ufw allow 'Apache'

Kontrol edelim;

ubuntu# sudo ufw status

Ubuntu çalışır durumunu kontrol edelim;

ubuntu# sudo service apache2 status

Bu durumunda bir browser üzerinden apache2 statik html sayfasının gelip gelmediğini kontrol edebiliriz.

http://server_ip

A. VirtualHost Oluşturma

Virtualhost oluşturmak Apache server üzerinde sanallaştırma yapmayı sağlar. Tek bir server üzerinde birden fazla web uygulamasını host etmenizi sağlar.

Bunları yaparken yapılması gereken temel 3 adım vardır, bunlar;

1- Virtualhost un sunulacağı dizin adresinin oluşturulması

2- Apache konfigrasyonları içerisinde host conf dosyalarının ayarlanıp bunların enable edilmesi

3- Conf dosyaları içerisine adreslerin girilmesi ve aşağıdaki sıkılaştırma adımlarında anlatılan ayarların eklenmesi

Öncelikle virtualhost için dizinimizi oluşturalım;

ubuntu# sudo mkdir -R /var/www/ornek-site.com/html

ubuntu# sudo adduser ornek-user

Burada kullanıcı parolası verilir. Grup otomatik olarak oluşturulur.

ubuntu# sudo chown -R ornek-user:ornek-user /var/www/ornek-site.com/html

ubuntu# sudo chmod -R 755 /var/www/ornek-site.com/

Test index sayfası oluşturarak virtualhost un çalıştığını teyit edelim.

ubuntu# sudo nano /var/www/ornek-site.com/html/index.html

```
<html>
<head>
<title>Welcome to Ornek-Site.com!</title>
</head>
<body>
<h1>Success! The ornek-site.com server block is working!</h1>
</body>
</html>
```

Ctrl — X yaptıktan sonra Y tuşuna basarak kaydedip çıkalım.

Sıra geldi apache site konfigrasyon dizini içerisindeki ayarları yapmaya...

ubuntu# sudo nano /etc/apache2/sites-available/ornek-site.com.conf

Dosya içerisine;

ayarları eklenir. Dosya Ctrl — X yapılıp Y tuşuna basılarak kaydedilip çıkılır.

Daha sonra konfigrasyon dosyası enable edilmesi gerekmektedir.

ubuntu# sudo a2ensite ornek-site.com.cof

Daha sonra ön tanımlı 000 konfigrasyonunu disable edelim,

ubuntu# sudo a2dissite 000-default.conf

Ayarlar eklendikten sonra apache restart edilmelidir.

ubuntu# sudo service apache2 restart

Kendi host dosyamıza ornek-site.com eklendikten sonra browser üzerinden;

http://ornek-site.com

a gidiliğinde yukarıda yazdığımız success yazısını görmemiz gerekmektedir.

Bu şekilde birden fazla host servis edilebilir.

A. Mod Yükleme ve Ayarlama

I. Mod_rewrite yükleme

Bu modun özelliği htaccess ile url rewrite işlemlerimi yapabilmeyi sağlamaktadır. Mod default olarak apache içerisinde gelmektedir. Modu enable etmek için aşağıdaki komutu girmemiz yeterli olacaktır;

ubuntu# sudo a2enmod rewrite

ubuntu# sudo service apache2 restart

Örnek htaccess dosyası içeriği;

.htaccess adında base host yoluna bir dosya açılır ve içerisine aşağıdakiler eklenir;

Eğer tüm sitelerinizin SSL üzerinde çalışmasını istiyorsanız. 000-default.conf u tekrar enable edip şu ayarları ekleyebilirsiniz:

Detaylı bilgi için;

https://httpd.apache.org/docs/trunk/rewrite/intro.html

II. Mod_ssl yükleme

Bu mod hostların güvenli client ve server iletişimini sağlamaktadır. SSL de bir hostu çalıştırmak için bu mod defaultta gelmektedir ve enable edilmesi gerekmektedir.

ubuntu# sudo a2enmod ssl

ubuntu# sudo service apache2 restart

Daha sonra */etc/apache2/sites-available/* içerisindeki enabled olan veya enabled yapacağınız host conflarının içerisinde ssl ayarları girilmelidir. Aşağıdaki konfigrasyonlarda önemli olan birkaç ayar bulunmaktadır. Bunlar;

VirtualHost *:443 : Burada * ile belirtilen alanın amacı o serverda birden fazla network interface i var ise bunların hangisine request gelirse gelsin 443 leri bu şekilde karşıla anlamına gelmektedir. Eğer spesifik bir adrese istek gelmesi durumunda yönlendirilmesi isteniyorsa * yerine ilgili IP adresi yazılır ve devamı aynı şekilde tamamlanmalıdır.

SSLEngine on : Burada on yapılmaması halinde 443 e de istek gelse 80 e istek gelmiş gibi non-ssl de çalışacaktır.

SSLCertificateFile : Burada ilgili hostun sertifikası eklenir. Self-signed ve örnek olduğu için basit bir örnek yazıldı.

SSLCertificateKeyFile : Burada sertifika istemi yapılırken oluşturulan key belirtilmelidir.

SSLCACertificateFile : Burada sertifikayı imzalayan yani sizin satın aldığınız kuruluşun CA sertifikası belirtilmelidir.

SSL sıkılaştırma ayarları sıkılaştırma adımında detaylı verilecektir.

III. Mod-Security2 Yükleme

Bu makalemizde mod security nin detaylı olarka konfigrasyonunu anlatmayacağız. İlgili işlemlere detaylı olarak aşağıdaki link üzerinden ulaşabilirsiniz:

Referans: https://www.linode.com/docs/web-servers/apache-tips-and-tricks/configure-modsecurity-on-apache/

B. Sıkılaştırma

I. Genel Sıkılaştırma

Server imzalarını ve tokenlarını gizleme:

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

> ServerTokens Prod
>
> ServerSignature Off

ubuntu# sudo service apache2 restart

Directory Browsing kapama:

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

> <Directory /var/www/ornek-site.com/test>
>
> Options None
>
> </Directory>

Burada None yerine -Indexes de eklenebilir ancak bu tamamen yöneticilerin ne yapmak istediğine bağlı olarak değişmektedir.

ubuntu# sudo service apache2 restart

None yapılması durumunda aşağıdaki gibi bir sonuç almamız gerekmektedir.

ETag opsiyonunu kapama:

Uzak saldırganların inode numarası, multipart MIME sınırı ve Etag başlığı üzerinden alt işlem gibi hassas bilgileri almasına izin verir.

PCI uyumluluğu için bu gerekli bir adımdır. Bunun için;

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

FileETag None

ubuntu# sudo service apache2 restart

I. Yetkilendirme

Apache yi düşük yetkili kullanıcı ile çalıştırma:

Apache olarak hem grup hem de kullanıcı oluşturarak bunların ayarlarını yapalım;

ubuntu# groupadd apache

ubuntu# useradd -G apache apache

Dizin yetkilerini değiştirelim;

ubuntu# sudo chown -R apache:apache /var/www/ornek-site.com/

Apache konfigrasyon dosyası içerisinde bu kullanıcı ve grubu tanımlayalım;

ubuntu# sudo nano /etc/apache2/apache2.conf

User apache

Group apache

ubuntu# sudo service apache2 restart

Diğer kullanıcıların conf ve bin dizinlerini görmesini engelleme;

ubuntu# sudo chmod -R 750 /etc/apache2/bin/ /etc/apache2/conf/

ubuntu# sudo service apache2 restart

Sistem Ayarlarının Korunması:

Varsayılan ayarlarda kullanıcılar htaccess dosyaları ile sistem ayarlarında istedikleri geçersiz kılmaları yapabilmektedir. Bunun önüne geçmek için root levelda üzerine yazmayı kapatmanız gerekmektedir;

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

```
<Directory />

Options -Indexes

AllowOverride None

</Directory>
```

ubuntu# sudo service apache2 restart

HTTP İstek Metodlarını Sınırlama:

Apache üzerinde GET, POST, HEAD metodları dışındaki metodlar sınırlandırılmalıdır. Bunun için;

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

```
<LimitExcept GET POST HEAD>

deny from all

</LimitExcept>
```

ubuntu# sudo service apache2 restart

II. Web Uygulama Güvenliği

a. Cookieler

Trace HTTP Request Disable Yapma:

Apache web sunucusunda varsayılan olarak Trace yöntemi gelmektedir. Bu özelliği etkinleştirmek, Cross Site Tracing saldırısına izin verebilir ve bir hackerın çerez bilgilerini çalması için bir seçenek sunabilir.

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

> TraceEnable off

ubuntu# sudo service apache2 restart

HTTPOnly ve Secure Flag lerini Aktif Etme:

XSS ataklarına karşı bu flaglar enable edilerek saldırılara karşı önlem alabilirsiniz:

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

> Header edit Set-Cookie ^(.*)$ $1;HttpOnly;Secure

ubuntu# sudo service apache2 restart

b. Clickjacking Atakları

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

> Header always append X-Frame-Options SAMEORIGIN

ubuntu# sudo service apache2 restart

c. X-XSS Protection

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

> Header set X-XSS-Protection “1; mode=block”

ubuntu# sudo service apache2 restart

d. Disable HTTP 1 Protocol

apache.conf içerisinde mod_rewrite ın yüklendiğinden emin olunuz yoksa rehberdeki ilgili bölümden ilgili ayarları apache.conf içerisine uygulayınız.

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

```
RewriteEngine On

RewriteCond %{THE_REQUEST} !HTTP/1.1$

RewriteRule .* — [F]
```

ubuntu# sudo service apache2 restart

e. Timeout Değeri Ekleme

Bu ayarın yapılmasının amacı Slowloris ve Diğer HTTP DOS ataklarının etkilerini düşürmektedir. Düşük timeout değeri ile bu saldırılar azaltılabilir.

ubuntu# sudo nano /etc/apache2/apache.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

```
Timeout 60
```

ubuntu# sudo service apache2 restart

f. SSL

SSL Cipherlerını Strong Seviyede Ayarlama:

Burada belirteceğimiz ayarlar Mod_ssl yükleme adımında belirttiğimiz gibi ornek-site.conf dosyasında 443 alanına sertifikalardan sonra eklenecektir.

ubuntu# sudo nano /etc/apache2/sites-enabled/ornek-site.com.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

```
SSLCipherSuite EECDH+AESGCM:EDH+AESGCM:AES256+EECDH:AES256+EDH
```

ubuntu# sudo service apache2 restart

SSLv2, SSLv3, TLSv1.0, TLSv1.1 disable Etme:

Burada belirteceğimiz ayarlar Mod_ssl yükleme adımında belirttiğimiz gibi ornek-site.conf dosyasında 443 alanına sertifikalardan sonra eklenecektir.

ubuntu# sudo nano /etc/apache2/sites-enabled/ornek-site.com.conf komutu ile apache konfigrasyon dosyası açılır ve aşağıdaki satırlar işlenir:

```
SSLProtocol –all +TLSv1.2
```

```
ubuntu# sudo service apache2 restart
```