# Web Arayüzü

Cuckoo, bir Django uygulaması olarak tam teşekküllü bir web arayüzü sağlar. Bu arayüz, dosyaları göndermenize, raporları göz atmanıza ve tüm analiz sonuçları arasında arama yapmanıza olanak tanır.


- Konfigürasyon
- Web Arayüzünün Başlatılması

# Konfigürasyon

Web arayüzü, verileri bir Mongo veritabanından çeker, bu nedenle Web Arayüzünün çalışması için `reporting.conf` dosyasında Mongo raporlama modülünün etkin olması gereklidir. Eğer öyle değilse, Web Arayüzü başlatılamaz ve bunun yerine bir istisna oluşturulur.

`$CWD/web/local_settings.py` konfigürasyon dosyasında bazı ek konfigürasyon seçenekleri bulunmaktadır.

```
# Copyright (C) 2013 Claudio Guarnieri.
# Copyright (C) 2014-2017 Cuckoo Foundation.
# This file is part of Cuckoo Sandbox - http://www.cuckoosandbox.org
# See the file 'docs/LICENSE' for copying permission.

import web.errors

# Maximum upload size (10GB, so there's basically no limit).
MAX_UPLOAD_SIZE = 10 * 1024 * 1024 * 1024

# Override default secret key stored in $CWD/web/.secret_key
# Make this unique, and don't share it with anybody.
# SECRET_KEY = "YOUR_RANDOM_KEY"

# Language code for this installation. All choices can be found here:
# http://www.i18nguy.com/unicode/language-identifiers.html
LANGUAGE_CODE = "en-us"

ADMINS = (
    # ("Your Name", "your_email@example.com"),
)

MANAGERS = ADMINS

# Allow verbose debug error message in case of application fault.
# It's strongly suggested to set it to False if you are serving the
# web application from a web server front-end (i.e. Apache).
DEBUG = False
DEBUG404 = False

# A list of strings representing the host/domain names that this Django site
# can serve.
```

```
# Values in this list can be fully qualified names (e.g. 'www.example.com').
# When DEBUG is True or when running tests, host validation is disabled; any
# host will be accepted. Thus it's usually only necessary to set it in production.
ALLOWED_HOSTS = ["*"]

handler404 = web.errors.handler404
handler500 = web.errors.handler500

#A list of strings representing the subnets or ipaddresses that can download
#samples and dropped files
#Values in this list can be ipv4 or ipv6 separated by ","
#(e.g. '127.0.0.0/8,10.0.0.0/8,fd00::/8').
ALLOWED_FILEDOWNLOAD_SUBNETS = '127.0.0.0/8,10.0.0.0/8,fd00::/8'
```

Üretim ortamlarında DEBUG değişkenini False olarak tutmanız ve en az bir ADMIN girişi yapılandırmanız, e-posta ile hata bildirimini etkinleştirmeniz önerilir.

2.0.0 sürümünde değişiklik: Varsayılan maksimum yükleme boyutu 25 MB'den 10 GB'a yükseltilmiştir, bu nedenle neredeyse herhangi bir dosyanın kabul edilmesi gerekmektedir.

# Web Arayüzünün Başlatılması

Web arayüzünü başlatmak için web/ dizininden basitçe aşağıdaki komutu çalıştırabilirsiniz:

```
$ cuckoo web runserver
```

Web arayüzünü belirli bir portta herhangi bir IP'yi dinleyecek şekilde yapılandırmak istiyorsanız, aşağıdaki komutu kullanabilirsiniz (PORT'u istediğiniz port numarasıyla değiştirin):

```
$ cuckoo web runserver 0.0.0.0:PORT
```

Veya doğrudan aşağıdaki gibi runserver bölümü olmadan ve dinlenilecek host'u belirterek kullanabilirsiniz:

```
$ cuckoo web -H 0
```

## Web Dağıtımı

Web Arayüzü sunucuyu başlatmanın varsayılan yöntemi birçok durum için uygundur, ancak bazı kullanıcılar sunucuyu daha güvenilir bir şekilde dağıtmak isteyebilir. Web Arayüzünü bir WSGI uygulaması olarak bir web sunucusuna açmak bunu mümkün kılar. Bu bölüm, Web Arayüzünün uWSGI ve nginx üzerinden nasıl dağıtılacağını basit bir örnek gösterir. Bu talimatlar Ubuntu GNU/Linux göz önüne alınarak yazılmıştır, ancak diğer platformlara uyarlanabilir.

Bu çözüm, uWSGI, uWSGI Python eklentisi ve nginx gerektirir. Tümü paketler olarak mevcuttur:

```
$ sudo apt-get install uwsgi uwsgi-plugin-python nginx
```

## uWSGI Kurulumu

İlk olarak, uWSGI'yi Web Arayüzü sunucusunu bir uygulama olarak çalıştırmak için kullanın.

Başlamak için, `cuckoo web --uwsgi` komutu tarafından rapor edilen gerçek konfigürasyonu içeren `/etc/uwsgi/apps-available/cuckoo-web.ini` adında bir uWSGI yapılandırma dosyası oluşturun, örneğin:

```
$ cuckoo web --uwsgi
[uwsgi]
plugins = python
virtualenv = /home/cuckoo/cuckoo
module = cuckoo.web.web.wsgi
uid = cuckoo
gid = cuckoo
static-map = /static=/home/..somepath..
# If you're getting errors about the PYTHON_EGG_CACHE, then
# uncomment the following line and add some path that is
# writable from the defined user.
# env = PYTHON_EGG_CACHE=
env = CUCKOO_APP=web
env = CUCKOO_CWD=/home/..somepath..
```

Bu yapılandırma, dağıtımın varsayılan uWSGI yapılandırmasından bir dizi ayarı devralır ve gerçek işi yapmak için Cuckoo paketinden cuckoo.web.web.wsgi'yi içe aktarır. Bu örnekte, Cuckoo'yu /home/cuckoo/cuckoo konumunda bir sanal ortamda kurduk. Cuckoo global olarak yüklendiyse sanal ortam seçeneği gerekli değildir (ve cuckoo web --uwsgi bunu bildirmez).

Uygulama yapılandırmasını etkinleştirin ve sunucuyu başlatın.

```
$ sudo ln -s /etc/uwsgi/apps-available/cuckoo-web.ini /etc/uwsgi/apps-enabled/
$ sudo service uwsgi start cuckoo-web     # or reload, if already running
```

> Uygulama için günlükler, dağıtım uygulama örnekleri için standart dizinde bulunabilir, yani /var/log/uwsgi/app/cuckoo-web.log. UNIX soketi de geleneksel bir konumda oluşturulur, yani /run/uwsgi/app/cuckoo-web/socket.

# nginx Kurulumu

Web Arayüzü sunucusu uWSGI'de çalışırken, nginx artık bir web sunucusu/ters proxy olarak ayarlanabilir ve HTTP isteklerini ona yönlendirebilir.

Başlamak için, cuckoo web --nginx komutu tarafından bildirilen gerçek konfigürasyonu içeren bir nginx konfigürasyon dosyası oluşturun:

```
```

```
$ cuckoo web --nginx
upstream _uwsgi_cuckoo_web {
    server unix:/run/uwsgi/app/cuckoo-web/socket;
}

server {
    listen localhost:8000;

    # Cuckoo Web Interface
    location / {
        client_max_body_size 1G;
        uwsgi_pass  _uwsgi_cuckoo_web;
        include     uwsgi_params;
    }
}
```

nginx'nin uWSGI soketine bağlanabilmesi için kullanıcısını cuckoo grubuna ekleyerek emin olun:

```
$ sudo adduser www-data cuckoo
```

Sunucu yapılandırmasını etkinleştirin ve sunucuyu başlatın:

```
$ sudo ln -s /etc/nginx/sites-available/cuckoo-web /etc/nginx/sites-enabled/
$ sudo service nginx start    # or reload, if already running
```

Bu noktada, Web Arayüzü sunucusunun sunucuda 8000 numaralı portta kullanılabilir olması gerekmelidir. Bu yapılandırmayı genişletmek için çeşitli konfigürasyonlar uygulanabilir, örneğin sunucu performansını ayarlamak, kimlik doğrulama eklemek veya HTTPS kullanarak iletişimi güvence altına almak gibi. Ancak, bunu kullanıcıya bir egzersiz bırakıyoruz.