Analiz

- Submission Utility
- Python Fonksiyonları

Submission Utility

Bir analiz göndermenin en kolay yolu, cuckoo submit yardımcı programını kullanmaktır. Şu anda şu seçeneklere sahiptir:

```
$ cuckoo submit --help
Usage: cuckoo submit [OPTIONS] [TARGET]...

  Submit one or more files or URLs to Cuckoo.

Options:
  -u, --url           Submitting URLs instead of samples
  -o, --options TEXT  Options for these tasks
  --package TEXT      Analysis package to use
  --custom TEXT       Custom information to pass along this task
  --owner TEXT        Owner of this task
  --timeout INTEGER   Analysis time in seconds
  --priority INTEGER  Priority of this task
  --machine TEXT      Machine to analyze these tasks on
  --platform TEXT     Analysis platform
  --memory            Enable memory dumping
  --enforce-timeout   Don't terminate the analysis early
  --clock TEXT        Set the system clock
  --tags TEXT         Analysis tags
  --baseline          Create baseline task
  --remote TEXT       Submit to a remote Cuckoo instance
  --shuffle           Shuffle the submitted tasks
  --pattern TEXT      Provide a glob-pattern when submitting a
                      directory
  --max INTEGER       Submit up to X tasks at once
  --unique            Only submit samples that have not been
                      analyzed before
  -d, --debug         Enable verbose logging
  -q, --quiet         Only log warnings and critical messages
  --help              Show this message and exit.
```

Bir seferde birden çok dosya veya dizini belirtebilirsiniz. cuckoo submit, dizinin tüm dosyalarını numaralandırır ve bunları birer birer gönderir.

Analiz paketleri kavramı, bu belgelerin ilerleyen kısımlarında ele alınacaktır (Analiz Paketleri bölümünde). İşte bazı kullanım örnekleri:

local binary gönderme:

```
$ cuckoo submit /path/to/binary
```

URL gönderme:

```
$ cuckoo submit --url http://www.example.com
```

Yerel bir binary dosya gönderme ve daha yüksek bir öncelik belirtme:

```
$ cuckoo submit --priority 5 /path/to/binary
```

Yerel bir binary dosya gönderme ve özel bir analiz süresi aşımını 60 saniye olarak belirtme:

```
$ cuckoo submit --timeout 60 /path/to/binary
```

Yerel bir binary dosya gönderme ve özel bir analiz paketi belirtme:

```
$ cuckoo submit --package <name of package> /path/to/binary
```

Yerel bir binary dosya gönderme ve özel bir yönlendirme belirtme:

```
$ cuckoo submit -o route=tor /path/to/binary
```

Yerel bir binary dosya gönderme ve özel bir analiz paketi ve bazı seçenekler belirtme (bu durumda kötü amaçlı yazılım için bir komut satırı argümanı):

```
$ cuckoo submit --package exe --options arguments=--dosomething
/path/to/binary.exe
```

Yerel bir binary dosya gönderme ve cuckoo1 adlı sanal makinede çalıştırılacak şekilde belirtme:

```
$ cuckoo submit --machine cuckoo1 /path/to/binary
```

Yerel bir binary dosya gönderme ve bir Windows makinesinde çalıştırılacak şekilde belirtme:

```
$ cuckoo submit --platform windows /path/to/binary
```

Yerel bir binary dosya gönderme ve analiz makinesinin tam bellek dökümünü almak için:

```
$ cuckoo submit --memory /path/to/binary
```

Yerel bir binary dosya gönderme ve analizin tam zaman aşımı süresince (Cuckoo'nun analizi sonlandırmaya ne zaman karar vereceği iç mekanizmasını göz ardı ederek) çalışmasını zorlama:

```
$ cuckoo submit --enforce-timeout /path/to/binary
```

Yerel bir binary dosya gönderme ve sanal makine saatinin ayarlanması. Biçimi %m-%d-%Y %H:%M:%S. Belirtilmezse, mevcut zaman kullanılır. Örneğin, örneği 24 Ocak 2001 tarihinde 14:41:20'de çalıştırmak istiyorsak:

```
$ cuckoo submit --clock "01-24-2001 14:41:20" /path/to/binary
```

Volatility analizi için bir örnek gönderme (cuckoo kanca kullanımının yan etkilerini azaltmak için seçenekleri free=True olarak kapatma):

```
$ cuckoo submit --memory --options free=yes /path/to/binary
```

Python Fonksiyonları

Gönderimleri, örnekleri ve genel yürütme durumunu takip etmek için, Cuckoo SQLite, MySQL veya MariaDB, PostgreSQL ve birçok diğer SQL veritabanı sistemini kullanmanıza izin veren popüler bir Python ORM olan SQLAlchemy kullanır.

Cuckoo, daha büyük çözümlere kolayca entegre edilebilen ve tamamen otomatik hale getirilebilen bir tasarıma sahiptir. Analiz gönderimini otomatikleştirmek için REST API arayüzünü kullanmanızı öneririz (bkz. REST API), ancak kendi Python gönderim betiğinizi yazmak istiyorsanız add_path() ve add_url() işlevlerini kullanabilirsiniz.

> **`add_path`*(file_path[, timeout=0[, package=None[, options=None[, priority=1[, custom=None[, owner=""[, machine=None[, platform=None[, tags=None[, memory=False[, enforce_timeout=False], clock=None[]]]]]]]]]]]])***

Bekleyen analiz görevleri listesine yerel bir dosya ekler. Yeni oluşturulan görevin ID'sini döndürür.

Parametreler:

- file_path (string): Gönderilecek dosyanın yolu
- timeout (integer): Analiz süresinin maksimum saniye cinsinden miktarı
- package (string veya None): Belirtilen dosya için kullanmak istediğiniz analiz paketi
- options (string veya None): Analiz paketine iletilmesi gereken seçeneklerin listesi (key=value, key=value formatında)
- priority (integer): Belirtilen dosyaya atanacak önceliğin sayısal temsili (1 düşük, 2 orta, 3 yüksek)
- custom (string veya None): İşleme veya raporlamada kullanılmak üzere iletilmesi gereken özel değer
- owner (string veya None): Görev sahibi
- machine (string veya None): Kullanmak istediğiniz sanal makinenin Cuckoo kimliği; belirtilmezse otomatik olarak bir tane seçilir
- platform (string veya None): Çalıştırmak istediğiniz analizin işletim sistemi platformu (şu anda yalnızca Windows)
- tags (string veya None): Makine seçimi için etiketler
- memory (True veya False): Analiz makinesinin tam bellek dökümünü oluşturmak için True olarak ayarlayın
- enforce_timeout (True veya False): Süreyi tam olarak uygulamak için True olarak ayarlayın
- clock (string veya None): Analiz makinesinde ayarlanacak özel bir saat zamanı

Return type: integer

Örnek kullanım:

```
>>> from cuckoo.core.database import Database
>>> db = Database()
>>> db.add_path("/tmp/malware.exe")
1
>>>
```

add_url(*url[, timeout=0[, package=None[, options=None[, priority=1[, custom=None[, owner=""[, machine=None[, platform=None[, tags=None[, memory=False[, enforce_timeout=False], clock=None[]]]]]]]]]]]])

Bekleyen analiz görevleri listesine yerel bir dosya ekler. Yeni oluşturulan görevin ID'sini döndürür.

Parametreler:

- url (string): Analiz edilecek URL
- timeout (integer): Analiz süresinin maksimum saniye cinsinden miktarı
- package (string veya None): Belirtilen URL için kullanmak istediğiniz analiz paketi
- options (string veya None): Analiz paketine iletilmesi gereken seçeneklerin listesi (key=value, key=value formatında)
- priority (integer): Belirtilen URL'ye atanacak önceliğin sayısal temsili (1 düşük, 2 orta, 3 yüksek)
- custom (string veya None): İşleme veya raporlamada kullanılmak üzere iletilmesi gereken özel değer
- owner (string veya None): Görev sahibi
- machine (string veya None): Kullanmak istediğiniz sanal makinenin Cuckoo kimliği; belirtilmezse otomatik olarak bir tane seçilir
- platform (string veya None): Çalıştırmak istediğiniz analizin işletim sistemi platformu (şu anda yalnızca Windows)
- tags (string veya None): Makine seçimi için etiketler
- memory (True veya False): Analiz makinesinin tam bellek dökümünü oluşturmak için True olarak ayarlayın
- enforce_timeout (True veya False): Süreyi tam olarak uygulamak için True olarak ayarlayın
- clock (string veya None): Analiz makinesinde ayarlanacak özel bir saat zamanı

Return type: integer

Örnek kullanım:

```
>>> from cuckoo.core.database import Database
>>> db = Database()
>>> db.connect()
>>> db.add_url("http://www.cuckoosandbox.org")
2
```

```
>>>
```