

# 3分 Ruby ハッキング

テーマ: RubyでApacheを拡張しよう



かずひこ@ネットワーク応用通信研究所  
<http://www.fdiary.net/>

# はじめに



- /.J

U(´A` )U

- 2ch

祭だワシヨーイ!

U(´A` )U

U(´A` )U

祭だワシヨーイ!

U(´A` )U

U(´A` )U

祭だワシヨーイ!

U(´A` )U

# はじめに

- /.J

U(A`U)

- 2ch

祭だワショーイ!

U(A`U)

U(A`U)

祭だワショーイ!

U(A`U)

サーバ管理者の敵!!!

U(A`U)

祭だワショーイ!

U(A`U)

# 今日のメニュー



「アンチ祭」フィルタ

# 材料



- Apache
- Ruby
- mod\_ruby

新鮮なものを使いましょう

# mod\_ruby

- ApacheにRubyを組み込む
- CGIが呼ばれるたびにRubyを起動しなくて済む
- だからCGIを速く動かせる

```
<Files index.cgi>
```

```
SetHandler ruby-object
```

```
RubyRequire 'apache/ruby-run'
```

```
RubyHandler Apache::RubyRun.instance
```

```
</Files>
```

- それだけではありません

# mod\_ruby



- いろんなハンドラがある
- だからApacheにちょっかいを出せる

# mod\_ruby

- RubyAccessHandler
  - アクセスの可否をチェックするためのハンドラ
  - check\_accessメソッドが呼ばれる
- RubyLogHandler
  - ログを取るためのハンドラ (レスポンスを返した後に呼ばれる)
  - log\_transactionメソッドが呼ばれる
- 他にもいろいろ

# レシピ



- リファラごとにカウントする
- アクセスを受けたらカウンタを増やす
- 閾値を越えたらアクセスを拒否する  
(HTTP\_SERVICE\_UNAVAILABLE)
- 応答を返したらカウンタを減らす

# ソース



```
require 'singleton'  
require 'dbm'  
  
module Apache  
  class AntiFestival  
    include Singleton  
  
    LIMIT_COUNT = 3  
    DB_PATH = '/var/tmp/antifestival.db'
```

# ソース

```
def check_access(r)
  @referer = r.headers_in['Referer'].to_s
  if inc_counter >= LIMIT_COUNT
    dec_counter
    return Apache::HTTP_SERVICE_UNAVAILABLE
  end
  return Apache::OK
end

def log_transaction(r)
  dec_counter
  return Apache::OK
end
```

# ソース



```
private

def inc_counter
  counter = 0
  transaction do |db|
    counter = db[@referer].to_i + 1
    db[@referer] = counter.to_s
  end
  counter
end
```

# ソース

```
def dec_counter
  transaction do |db|
    counter = db[@referer].to_i - 1
    if counter <= 0
      db.delete( @referer )
    else
      db[@referer] = counter.to_s
    end
  end
end
```

# ソース



```
def transaction
  begin
    DBM.open( DB_PATH ) do |db|
      yield db
    end
  end
  rescue Errno::EAGAIN
    retry
  end
end
end
```

- .htaccessの設定

```
<Files index.cgi>
```

```
  RubyRequire 'antifestival'
```

```
  RubyAccessHandler AntiFestival.instance
```

```
  RubyLogHandler AntiFestival.instance
```

```
</Files>
```

# できあがり



- ブラウザで祭
- ab (Apache Bench) で祭
- ログを確認

# テキスト



- <http://wiki.modruby.net/ja/?ApacheEnhancement>
- <http://shugo.net/article/cmagazine/8th/>



Ruby  
ハッキング

終