SecurusGlobal

Abrupt

Web App Pentest Framework

Thiébaud <tw@securusglobal.com>

Web App Pentest

- Existing tools:
  - General: Burp Suite, WebScarab
  - Specialized: sqlmap
- Limitations:
  - License
  - Inactive/outdated
  - Technologies (ajax, amf) & Evolution
  - Scenario of tests
  - WAF detected

Abrupt

- Python based (>=2.6)
- BSD licensed
- Scapy inspired
- Can be used as an interactive tool or a framework
- Include:
  - HTTP models
  - Proxy
  - Request generation (~ intruder)
  - Sessions

```
$ abrupt
Abrupt 0.1
>>>
```

```
$ abrupt
Abrupt 0.1
>>> proxy()
Running on port 8080
Ctrl-C to interrupt the proxy...
```

```
$ abrupt
Abrupt 0.1
>>> proxy()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /> ?
```

```
$ abrupt
Abrupt 0.1
>>> proxy()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /> ? help
(v)iew, (e)dit, (f)orward, (d)rop, (c)ontinue
[f]?
```

```
$ abrupt
Abrupt 0.1
>>> proxy()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /> ? help
(v)iew, (e)dit, (f)orward, (d)rop, (c)ontinue
[f]? f
<200 Gzip 5419>
<GET www.phrack.org /style.css> ?
```

```
$ abrupt
Abrupt 0.1
>>> proxy()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /> ? help
(v)iew, (e)dit, (f)orward, (d)rop, (c)ontinue
[f]? f
<200 Gzip 5419>
<GET www.phrack.org /style.css> ? c
…
```

```
$ abrupt
Abrupt 0.1
>>> proxy()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /> ? help
(v)iew, (e)dit, (f)orward, (d)rop, (c)ontinue
[f]? f
<200 Gzip 5419>
<GET www.phrack.org /style.css> ? c
…  ^-C
{200:2 | www.phrack.org}
>>>
```

```
$ abrupt
Abrupt 0.1
>>> proxy()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /> ? help
(v)iew, (e)dit, (f)orward, (d)rop, (c)ontinue
[f]? f
<200 Gzip 5419>
<GET www.phrack.org /style.css> ? c
…  ^-C
{200:2 | www.phrack.org}
>>> rs = _
>>> print rs
Method Path       Query Status Length
GET    /                200    5419
GET    /style.css       200    4406

>>>
```

```
>>> print rs
Method Path       Query Status Length
GET    /                200    5419
GET    /style.css       200    4406

>>> r = rs[0]
>>>
```

```
>>> print rs
Method Path       Query Status Length
GET    /                200    5419
GET    /style.css       200    4406

>>> r = rs[0]
>>> print r
GET / HTTP/1.1
Host: www.phrack.org
User-Agent: Mozilla/5.0 (X11; Linux i686;
rv:5.0) Gecko/20100101 Firefox/5.0
…
```

```
>>> r.hostname
’www.phrack.org’
>>>
```

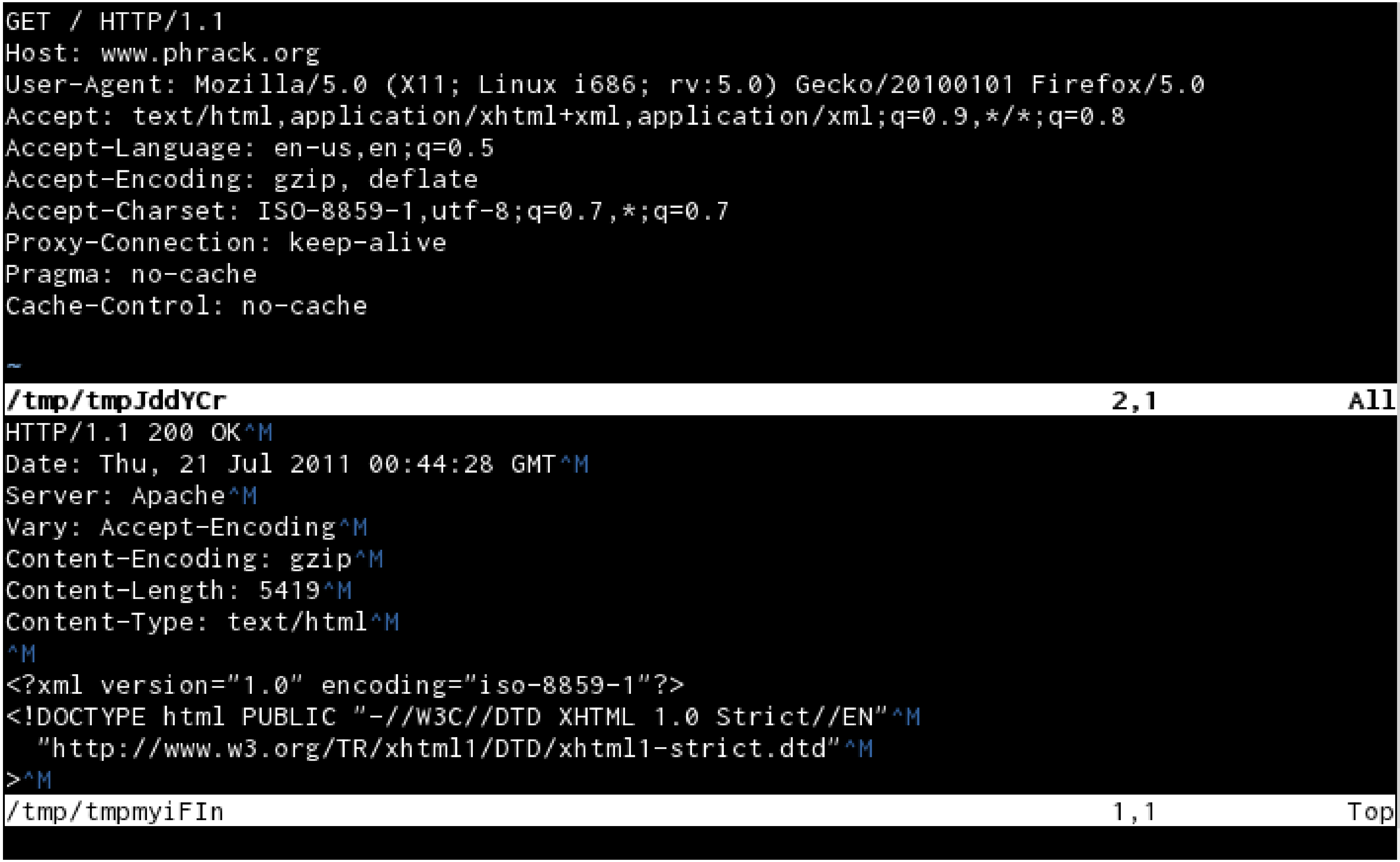
```
>>> r.hostname
'www.phrack.org'
>>> r.method
'GET'
>>>
```

```
>>> r.hostname
'www.phrack.org'
>>> r.method
'GET'
>>> r.response
<200 Gzip 5419>
>>>
```

```
>>> r.hostname
'www.phrack.org'
>>> r.method
'GET'
>>> r.response
<200 Gzip 5419>
>>> r.response.status
'200'
>>>
```

```
>>> r.hostname
'www.phrack.org'
>>> r.method
'GET'
>>> r.response
<200 Gzip 5419>
>>> r.response.status
'200'
>>> print r.response
HTTP/1.1 200 OK
Date: Thu, 21 Jul 2011 00:44:28 GMT
Server: Apache
…
```

```
>>> r.play()
>>> r.play()
```



```
GET / HTTP/1.1
Host: www.phrack.org
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:5.0) Gecko/20100101 Firefox/5.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip, deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Proxy-Connection: keep-alive
Pragma: no-cache
Cache-Control: no-cache

~
/tmp/tmpJddYCr                                                        2,1            All
HTTP/1.1 200 OK^M
Date: Thu, 21 Jul 2011 00:44:28 GMT^M
Server: Apache^M
Vary: Accept-Encoding^M
Content-Encoding: gzip^M
Content-Length: 5419^M
Content-Type: text/html^M
^M
<?xml version="1.0" encoding="iso-8859-1"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"^M
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd"^M
>^M
/tmp/tmpmyiFIn                                                        1,1            Top
```

```
>>> rs += p()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /issues.html> ?
```

```
>>> rs += p()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /issues.html> ? c
<200 Gzip 14496> ^C
3 requests intercepted
>>>
```

```
>>> rs += p()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /issues.html> ? c
<200 Gzip 14496> ^C
3 requests intercepted
>>> rs
{200:5 | www.phrack.org}
>>>
```

```
>>> rs += p()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /issues.html> ? c
<200 Gzip 14496> ^C
3 requests intercepted
>>> rs
{200:5 | www.phrack.org}
>>> r = rs[4]
>>>
```

```
>>> rs += p()
Running on port 8080
Ctrl-C to interrupt the proxy...
<GET www.phrack.org /issues.html> ? c
<200 Gzip 14496> ^C
3 requests intercepted
>>> rs
{200:5 | www.phrack.org}
>>> r = rs[4]
>>> print r
GET /issues.html?issue=67&id=1 HTTP/1.1
…
```

```
>>> irs = inject(r, issue=“sqli”)
>>> irs
{unknown:106 | www.phrack.org}
>>>
```

```
>>> irs = inject(r, issue=“sqli”)
>>> irs
{unknown:106 | www.phrack.org}
>>> irs()
Running 106 requests...done.
>>>
```

```
>>> irs = inject(r, issue=“sqli”)
>>> irs
{unknown:106 | www.phrack.org}
>>> irs()
Running 106 requests...done.
>>> irs
{200:106 | www.phrack.org}
>>>
```

```
>>> irs = inject(r, issue=“sqli”)
>>> irs
{unknown:106 | www.phrack.org}
>>> irs()
Running 106 requests...done.
>>> irs
{200:106 | www.phrack.org}
>>> print irs
Method Payload              Status Length
GET    issue=‘              200    2396
GET    issue=’ --           200    2396
GET    issue=a’ or 1=1 --   200    5287
…
```

And much more

- Save and load through sessions
- Transparent SSL support
- Different injection modes
- Spidering (Experimental)
- And lots coming …

Real world example

- CSRF protected form. One parameter is an upper case. Which letters are accepted by the app ?
  - Options?
    - By hand
    - Burp recently included ‘macro’ concept
    - Use Abrupt
  - Abrupt steps:
    - intercept a GET to retrieve a nonce
    - intercept a POST to test an upper case
    - Save both in a session
    - Write 11 lines of python

Enumeration CSRF protection: request

```
>>> print r_post
POST / HTTP/1.1
Host: 127.0.0.1:5000
User-Agent: Mozilla/5.0 (X11; Linux i686; rv:5.0)
Content-Type: application/x-www-form-urlencoded
Content-Length: 59

memberRole=A&my_app.nonce=361a3aa8e86352c8df4
```

Enumeration CSRF protection: code

```
from abrupt.all import *
import re, string

switch_session('my_session')
rs = RequestSet()

for u in list(string.uppercase):
  r_get()
  nonce = re.search(r"value=\'(.+?)\'",
          r_get.response.readable_content).groups()[0]
  r_post_nonced = i_at(r_post, "361a3aa8e86352c8df4", [nonce,])
  rs += i(r_post_nonced, memberRole=[u,])
  rs()
print rs
```

Enumeration CSRF protection: code

```
from abrupt.all import *
import re, string

switch_session(’my_session’)
rs = RequestSet()

for u in list(string.uppercase):
  r_get()
  nonce = re.search(r”value=\’(.+?)\’”,
          r_get.response.readable_content).groups()[0]
  r_post_nonced = i_at(r_post, ”361a3aa8e86352c8df4”, [nonce,])
  rs += i(r_post_nonced, memberRole=[u,])
  rs()
print rs
```

Enumeration CSRF protection: code

```
from abrupt.all import *
import re, string

switch_session('my_session')
rs = RequestSet()

for u in list(string.uppercase):
  r_get()
  nonce = re.search(r”value=\'(.+?)\'”,
                r_get.response.readable_content).groups()[0]
  r_post_nonced = i_at(r_post, ”361a3aa8e86352c8df4”, [nonce,])
  rs += i(r_post_nonced, memberRole=[u,])
  rs()
print rs
```

Enumeration CSRF protection: code

```
from abrupt.all import *
import re, string

switch_session('my_session')
rs = RequestSet()

for u in list(string.uppercase):
  r_get()
  nonce = re.search(r”value=\'(.+?)\'”,
          r_get.response.readable_content).groups()[0]
  r_post_nonced = i_at(r_post, ”361a3aa8e86352c8df4”, [nonce,])
  rs += i(r_post_nonced, memberRole=[u,])
  rs()
print rs
```

Enumeration CSRF protection: code

```
from abrupt.all import *
import re, string

switch_session('my_session')
rs = RequestSet()

for u in list(string.uppercase):
  r_get()
  nonce = re.search(r”value=\'(.+?)\'”,
          r_get.response.readable_content).groups()[0]
  r_post_nonced = i_at(r_post, ”361a3aa8e86352c8df4”, [nonce,])
  rs += i(r_post_nonced, memberRole=[u,])
  rs()
print rs
```

Enumeration CSRF protection: exec

```
$ python csrf_example.py
Loading 2011.07.21_13.42.p
Running 26 requests...done.

Method Path Payload      Query Status Length
POST   /    memberRole=A       302    221
POST   /    memberRole=B       500    -
POST   /    memberRole=C       500    -
POST   /    memberRole=D       500    -
POST   /    memberRole=E       302    221
POST   /    memberRole=F       500    -
…
```

Enumeration CSRF protection: extra

```
redirected = rs.filter(response__status=”302”)
print redirected.extract(”memberRole”)
```

```
[’A’, ’E’, ’I’, ’S’, ’T’, ’X’]
```

Clone / Fork me:

https://github.com/securusglobal/abrupt

Comments / Bugs: tw@securusglobal.com

Thanks: Securus crew ([_] & ace) and Ruxmon guys, awesome job!

Questions?