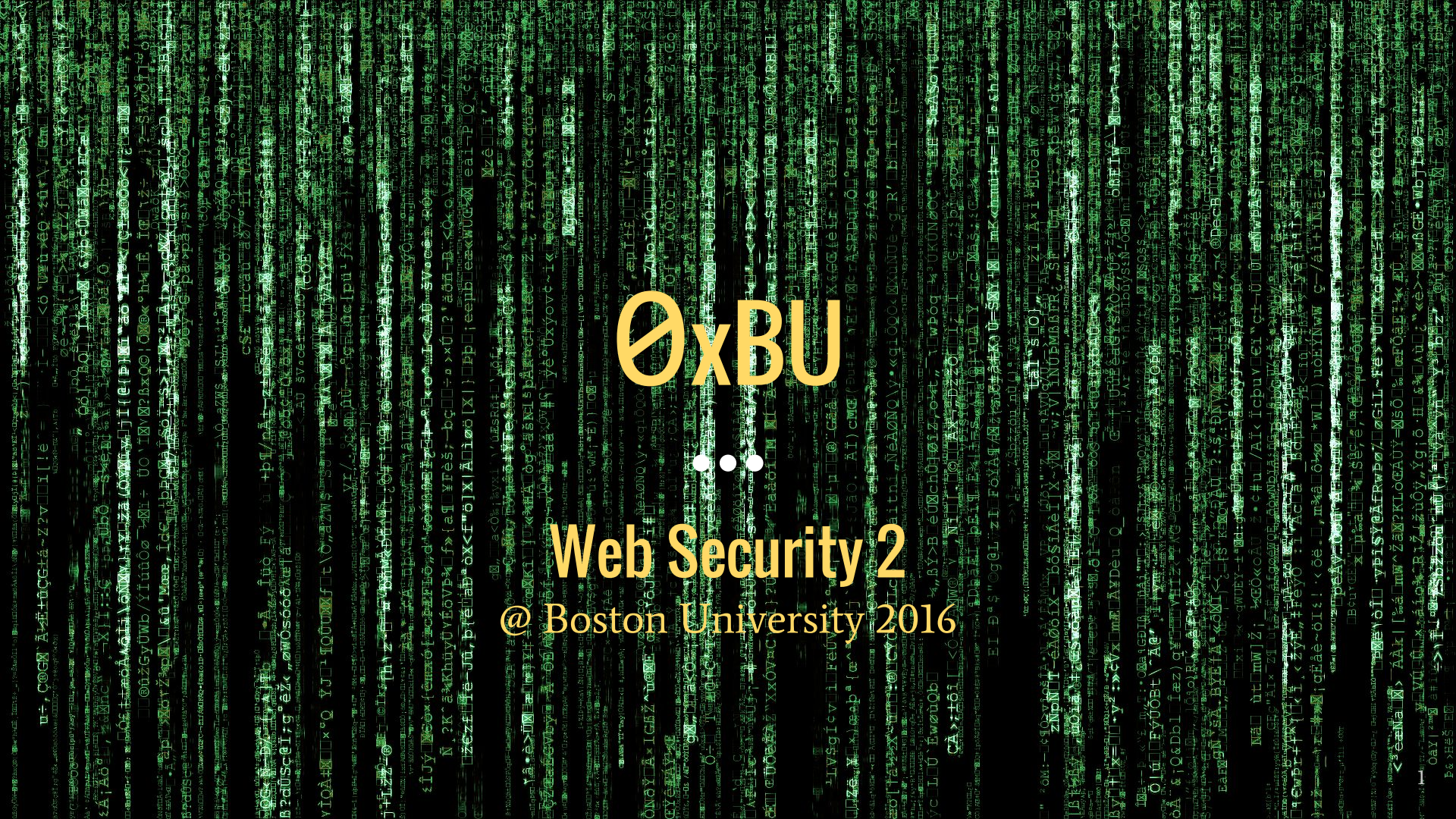

0xBU
Web Security 2
@ Boston University 2016

Landscape

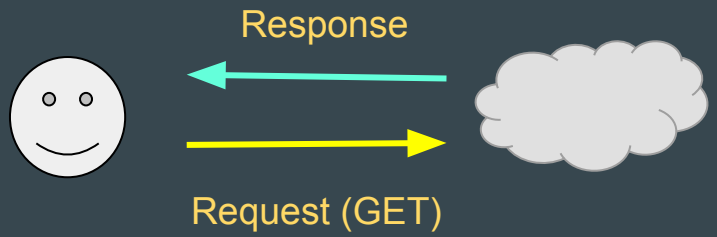
- Mainly focused on the application layer
- HTTP, is the language that the web speaks
- Firewalls must allow traffic to and from the port used for HTTP (80)
- Easiest way to compromise hosts, networks, and users
- Widely deployed, little logging, difficult to detect

Attack Surface

- The web is the wild west, but there is one common “stack”, LAMP
  - Linux: the operating system
  - Apache HTTPd: the HTTP server
  - MySQL: the database server
  - PHP: the programming language
- Another common stack, MEAN
  - MongoDB: the database server
  - Express.js: The HTTP server
  - Angular.js: The frontend programming
  - Node.js: The backend programming

HTTP Refresher

Messages have header, and optional body




```
GET /doc/test.html HTTP/1.1
Host: www.test101.com
Accept: image/gif, image/jpeg, */*
Accept-Language: en-us
Accept-Encoding: gzip, deflate
User-Agent: Mozilla/4.0
Content-Length: 35

bookId=12345&author=Tan+Ah+Teck
```

Request Line
Request Headers
Request Message Header
A blank line separates header & body
Request Message Body

```
HTTP/1.1 200 OK
Date: Sun, 08 Feb xxxx 01:11:12 GMT
Server: Apache/1.3.29 (Win32)
Last-Modified: Sat, 07 Feb xxxx
ETag: "0-23-4024c3a5"
Accept-Ranges: bytes
Content-Length: 35
Connection: close
Content-Type: text/html

<h1>My Home page</h1>
```

Status Line
Response Headers
Response Message Header
A blank line separates header & body
Response Message Body

HTTP Injection

Server receives requests - the request is completely controlled by a client/browser

- User can edit headers, and body to anything they want
- Solution: Server developers must know this, and only rely on data that cannot be edited, e.g. symmetrically encrypted.
- Stackoverflow pwned:
  - http://blog.ircmaxell.com/2012/11/anatomy-of-attack-how-i-hacked.html
- Demos!

Scripting Language Injection (1/5)

Basic web architecture:

1. User inputs, and sends data, e.g. “username and password”
2. Server receives “username and password”
3. Server runs some sort of program written by a developer to process the data
4. Server returns the output of that program back to the user

Scripting Language Injection (2/5)

- Using a URL for web functionality
  - e.g. http://example.com/news.php?page=25
  - eg. http://example.com/robot/refuel

Typical backend code:

```
@app.route('/robot/<path:task>', methods=['GET'])
def robot_task(task):
        print("python " + task + ".py")
        r = check_output("python " + task + ".py", shell=True)
        return r
```

See the bug?

Oh my!
http://example.com/robot/refuel%3Bcat%20%2Fetc%2Fpasswd%3Becho%20

Scripting Language Injection (3/3)

Different languages have their own problems

- PHP
  - http://example.com/forgot_password.php?reset_key=90210applesauce

```php
$reset_key = $_GET['reset_key'];
$correct_reset_key = get_correct_value_somehow();

if (strcmp($reset_key, $correct_reset_key) == 0) {
    // Go ahead and reset the password
} else {
    echo 'Sorry, incorrect key';
}
```

See the bug?

Scripting Language Injection (3/3)

Different languages have their own problems

- PHP
  - http://example.com/forgot_password.php?reset_key=90210applesauce

```php
$reset_key = $_GET['reset_key'];
$correct_reset_key = get_correct_value_somehow();

if (strcmp($reset_key, $correct_reset_key) == 0) {
    // Go ahead and reset the password
} else {
    echo 'Sorry, incorrect key';
}
```

http://example.com/forgot_password.php?reset_key[]=90210applesauce

$reset_key = [“90210applesauce”] # Note that it’s a list
strcmp(list, anything) = NULL in PHP
NULL == 0 in PHP # Not true for ‘===’, always use ===!
Oh my!

Scripting Language Injection (3/3)

Different languages have their own problems

- Perl
  - Live @ `0xBU`
  - `http://makeaface.picoctf.com/`

NoSQL Injection

- NoSQL databases promise simpler storage mechanisms
  - Basically toss in a JSON object into it and done
  - No need for schemas, or thinking of object relations
  - “Safer”, SQL is dangerous
  - Not necessarily better in any way than classic SQL databases
- MySQL is boring now, NoSQL is the new hotness
  - MongoDB, Redis, Cassandra, etc....


```
SELECT * FROM users WHERE username = '$username' AND password = '$password'
```

becomes

```
db.users.find({username: username, password: password});
```

NoSQL Injection

Typical node.js backend using MongoDB:

```
app.post('/', function (req, res) {
        db.users.find({username: req.body.username, password: req.body.password}, function (err, users) {
            // TODO: handle the rest
        });
});
```

See the bug?

NoSQL Injection

Typical node.js backend using MongoDB:

```
app.post('/', function (req, res) {
        db.users.find({username: req.body.username, password: req.body.password}, function (err, users) {
            // TODO: handle the rest
        });
});
```

See the bug?

Same thing as SQL Injection! Validation failure to ensure username and password are strings!

```
POST http://target/ HTTP/1.1
Content-Type: application/json

{
    "username": {"$gt": ""},
    "password": {"$gt": ""}
}
```

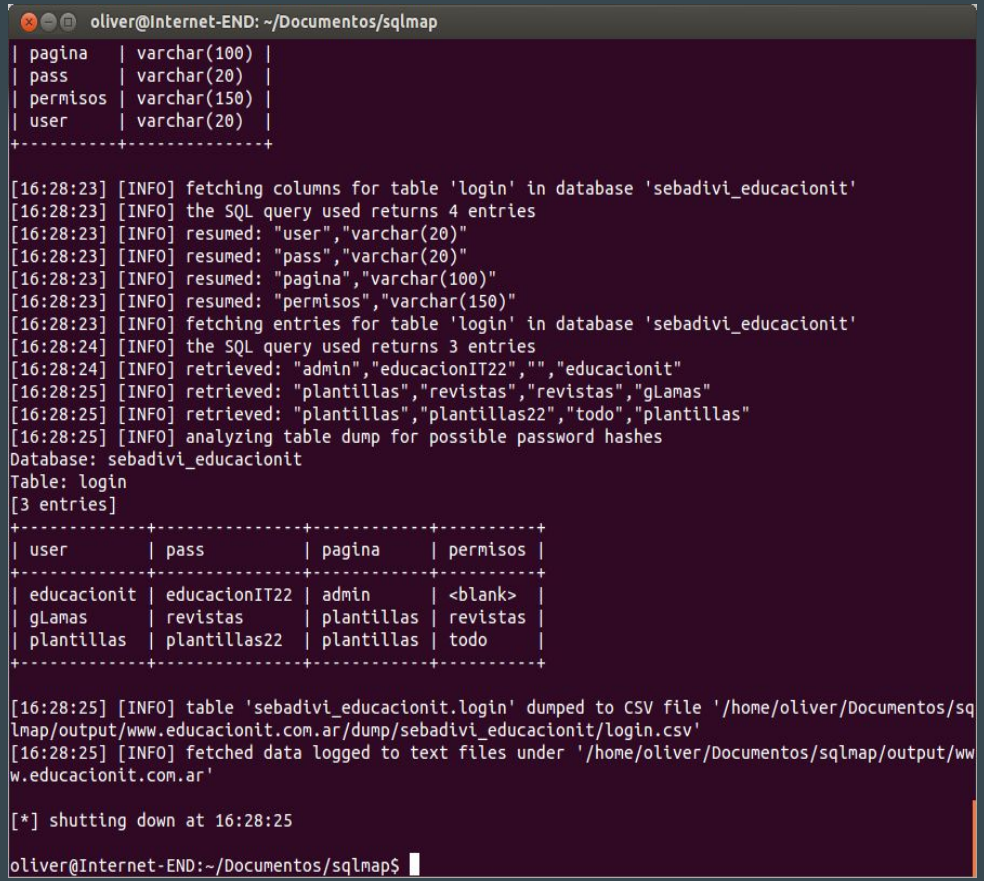
```
for all usernames in DB:
        if username > “” and
          password > “”:
        return True
```

Nearly always true!

H4CK3r T00LZ 2

Auto pwners

- Sqlmap
  - Attempts SQL injection on fields
  - Highly unlikely to work
    - But you can hope
  - CTFs ban it because of server strain!
- NoSqlmap
  - Same as bove, except for NoSQL
- dirs3arch / dirbuster
  - Bruteforce find hidden files/folders
  - Likely to work, but boring results
  - CTFs ban it because of server strain!



Familiarity with Web Frameworks

Every language has common development platforms w/ their own quirks

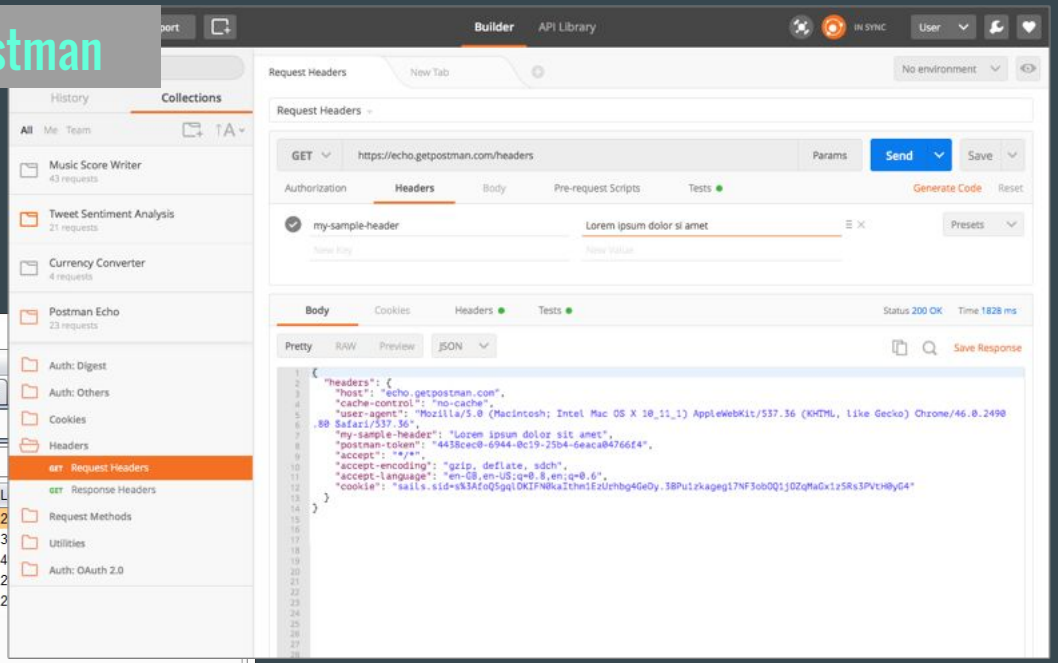
- PHP
  - CakePHP
- Python
  - Flask
  - Django
  - MongoDB
- Perl
  - CGI
- Ruby
  - Sinatra
  - Rails
- Javascript
  - node.js

Play with HTTP

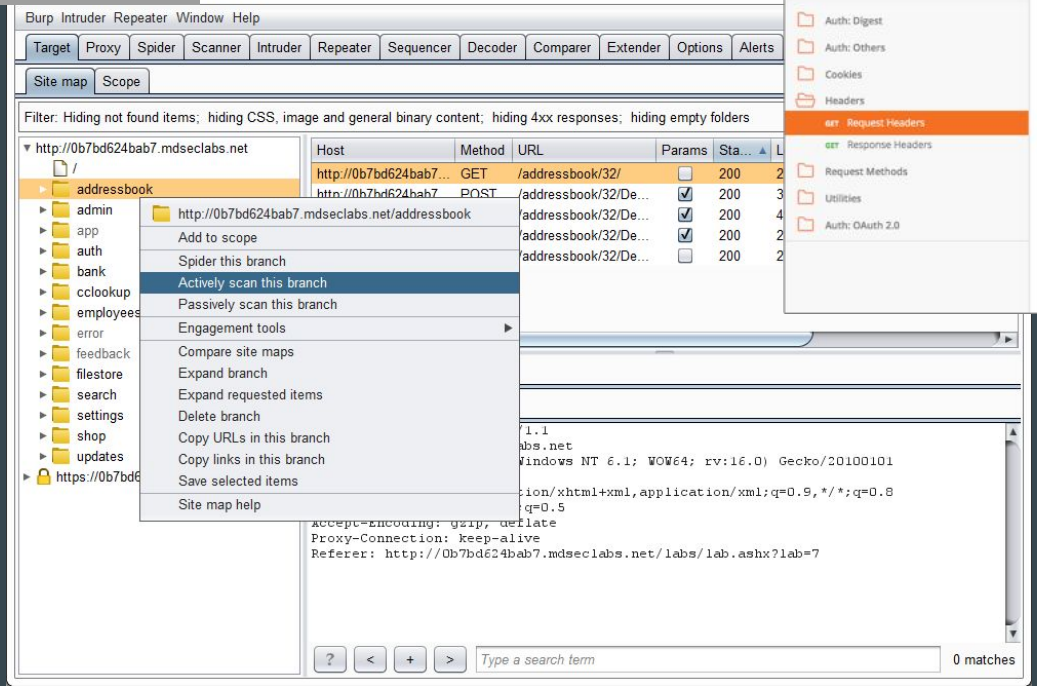
- Burpsuite
  - Intercept, modify, create, send, receive and more HTTP packets
- Postman
  - Create, send, and receive HTTP packets
    - More so for web development
- Firefox/Browser plugins
  - Modify packets
  - Chrome doesn't let you I believe?
- http://meyerweb.com/eric/tools/dencoder/
  - Encode and decode strings into passable URLs

Play with HTTP

Postman



Burp



Demos

Live @ 0xBU

Next week

Challenge:

Easy/Medium: http://web2014.picoctf.com/injection4/

Easy/Medium: https://2013.picoctf.com/problems/php4/

Hard: https://wildwildweb.fluxfingers.net:1424/

__libc_fini