HACKTHEBOX

WHITEBOX ATTACKS CHEAT SHEET

Prototype Pollution

Exploitation

- Vulnerable NodeJS libraries: here
- Access prototype of an object via `__proto__` or `constructor.prototype` property
- Client-side prototype pollution vulnerabilities: here
- Safe Identification: here
  - Status Code: `__proto__.status`
  - Parameter Limit: `__proto__.parameterLimit`
  - Content-Type: `__proto__.content-type`

Prevention

- Check user-supplied properties against a whitelist
- Freeze prototype by calling `Object.freeze()`
- Create object without prototype with `Object.create(null)`

Timing Attacks & Race Conditions

Exploitation

Timing Attacks:

- Identify tasks that take a measurable amount of time
- Determine under which circumstances this task is executed
- Determine if you can infer information about the underlying system

Race Conditions:

- Identify tasks that assume single-threaded execution (file access, database access, ...)
- Identify if there is a possible timing window that could be exploited

Prevention

Timing Attacks:

- Make sure all computation paths take (nearly) the same amount of time

Race Conditions:

- Always assume multi-threaded execution
- Use file locks and database locks when performing critical operations

Type Juggling

Exploitation

- Identify loose comparisons using the `==` or `!=` operators
- Identify the PHP version used and look at the Type Comparison Table to determine inputs that cause an unexpected outcome

Prevention

Use strict comparison operators: `===` and `!==`

HTB ACADEMY CHEATSHEET