Cyber Offense and Defense

UNIVERSITÀ DELLA CALABRIA
il Campus per eccellenza

Information disclosure + Directory traversal

Mario Alviano

Main References

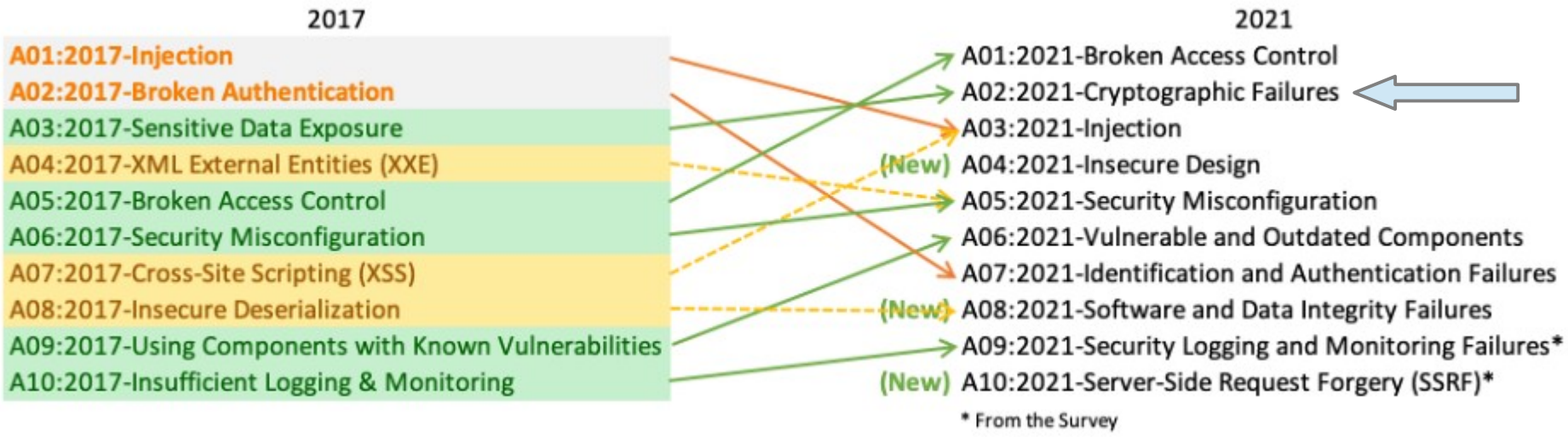
Bug Bounty Bootcamp – Chapter 21

https://portswigger.net/web-security/information-disclosure

https://portswigger.net/web-security/file-path-traversal

OWASP Top Ten

A broad consensus about the most critical security risks to web applications



Information Disclosure Vulnerabilities

A website unintentionally reveals sensitive information to its users (information leakage).

- Data about other users, such as usernames or financial information
- Sensitive commercial or business data
- Technical details about the website and its infrastructure

Examples

- Files for web crawlers
  - /robots.txt and /sitemap.xml
- Directory listings
  - Easier to discover unintended files
- Developer comments
  - Credentials or known bugs
- Error messages and debugging data
  - Stack trace and other internal data
- Backup files and version control history
  - Containing source codes or credentials

https://portswigger.net/web-security/information-disclosure/exploiting#common-sources-of-information-disclosure

Prevention

- Identify all sensitive information (every developer must be aware of them)
  - Audit code for potential information disclosure
  - Don’t hardcode credentials and sensitive information
- Use generic error messages
  - Implement a global exception handler
- Debugging and diagnostic features must be disabled
  - Test for them in the deployed system
- Don’t use configurations or third-party technologies if you don’t understand them

Directory Traversal (Path Traversal)

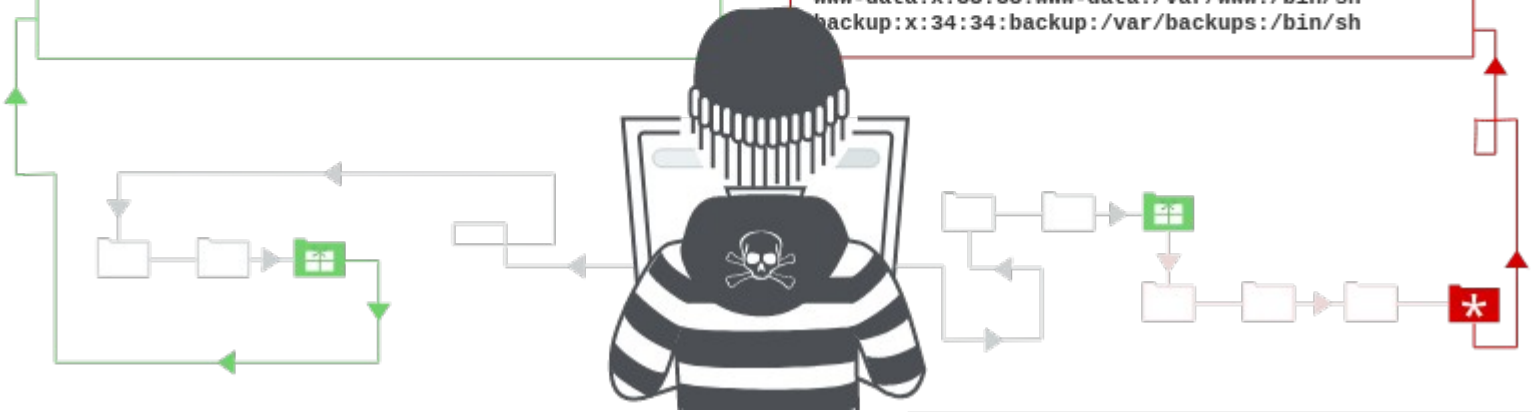
An endpoint to fetch images (given their names)… what a time saver!

/loadImage?filename=gift.png

/loadImage?filename=../../../etc/passwd

```
root:x:0:0:root:/root:/bin/bash
daemon:x:1:1:daemon:/usr/sbin:/bin/sh
bin:x:2:2:bin:/bin:/bin/sh
sys:x:3:3:sys:/dev:/bin/sh
sync:x:4:65534:sync:/bin:/bin/sync
games:x:5:60:games:/usr/games:/bin/sh
man:x:6:12:man:/var/cache/man:/bin/sh
lp:x:7:7:lp:/var/spool/lpd:/bin/sh
mail:x:8:8:mail:/var/mail:/bin/sh
news:x:9:9:news:/var/spool/news:/bin/sh
uucp:x:10:10:uucp:/var/spool/uucp:/bin/sh
proxy:x:13:13:proxy:/bin:/bin/sh
www-data:x:33:33:www-data:/var/www:/bin/sh
backup:x:34:34:backup:/var/backups:/bin/sh
```

I would like to fetch the **“image”** *../../../etc/passwd*



https://portswigger.net/web-security/file-path-traversal

Improper sanification and validation

What can go wrong?

- Reject strings with ../ => use absolute paths
- Remove any ../ (non recursively) => use ….//
- Sanification before URL decode => URL encode the URL encoded payload
- Only accept paths with a fixed prefix => use /prefix/path/../../etc/passwd
- Only accepts paths with a fixed suffix => add a null byte before the suffix

Prevention

Don’t use strings… use filesystem APIs,
get the canonical form of the path,
validate it against your business rules

Questions