

# Libraries for exploits



Yusuf Ismail, Kai Johnson

## ABSTRACT

Some user has **powerful permissions** over one file. Poor planning means we can **run commands on the server**. Combining these, we can hack our way into the user, and use their permissions to **escalate to root**.

## KEY CONCEPTS

### File Permissions

The most powerful user on a system can decide which sub-users are allowed to see, edit, and run which files. This powerful user is called “root.”

## REFERENCES

GitHub repo



LD\_PRELOAD



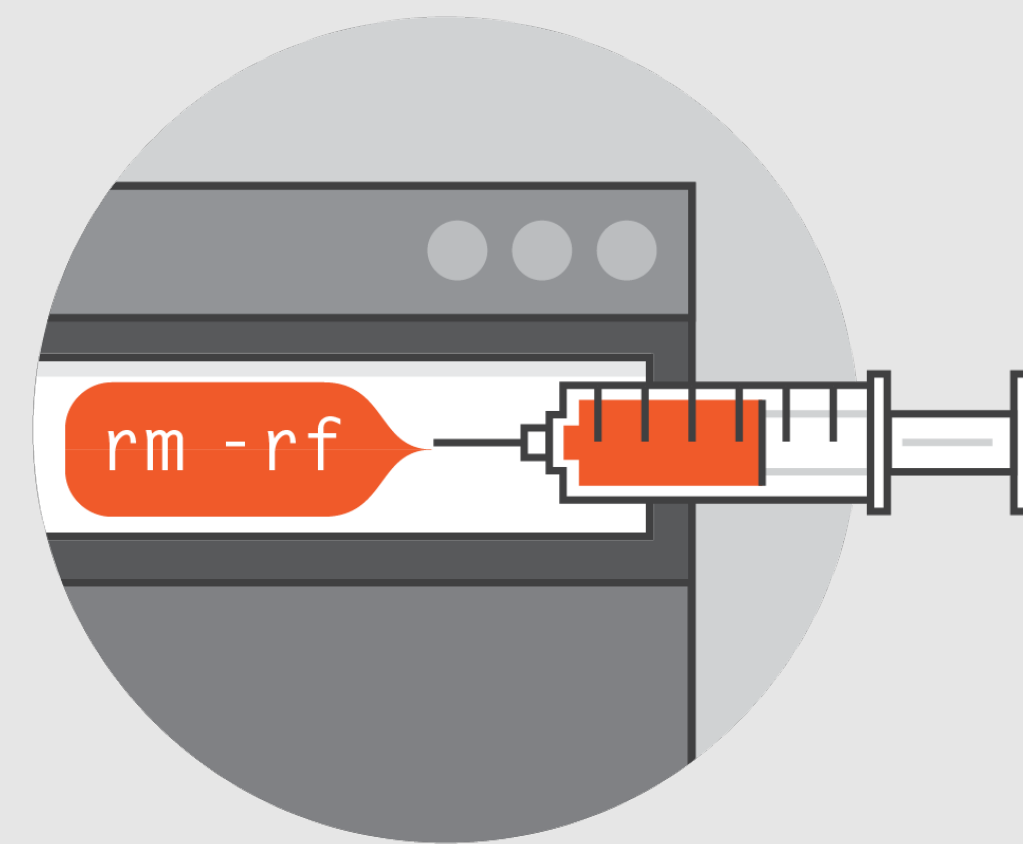
Password spraying



0

## COMMAND INJECTION

The Risk: Unfiltered user input allows attackers to run commands on your server. Even with limited "www-data" permissions, attackers can cause serious damage, make horizontal movements, or escalate privilege.



### Prevention:

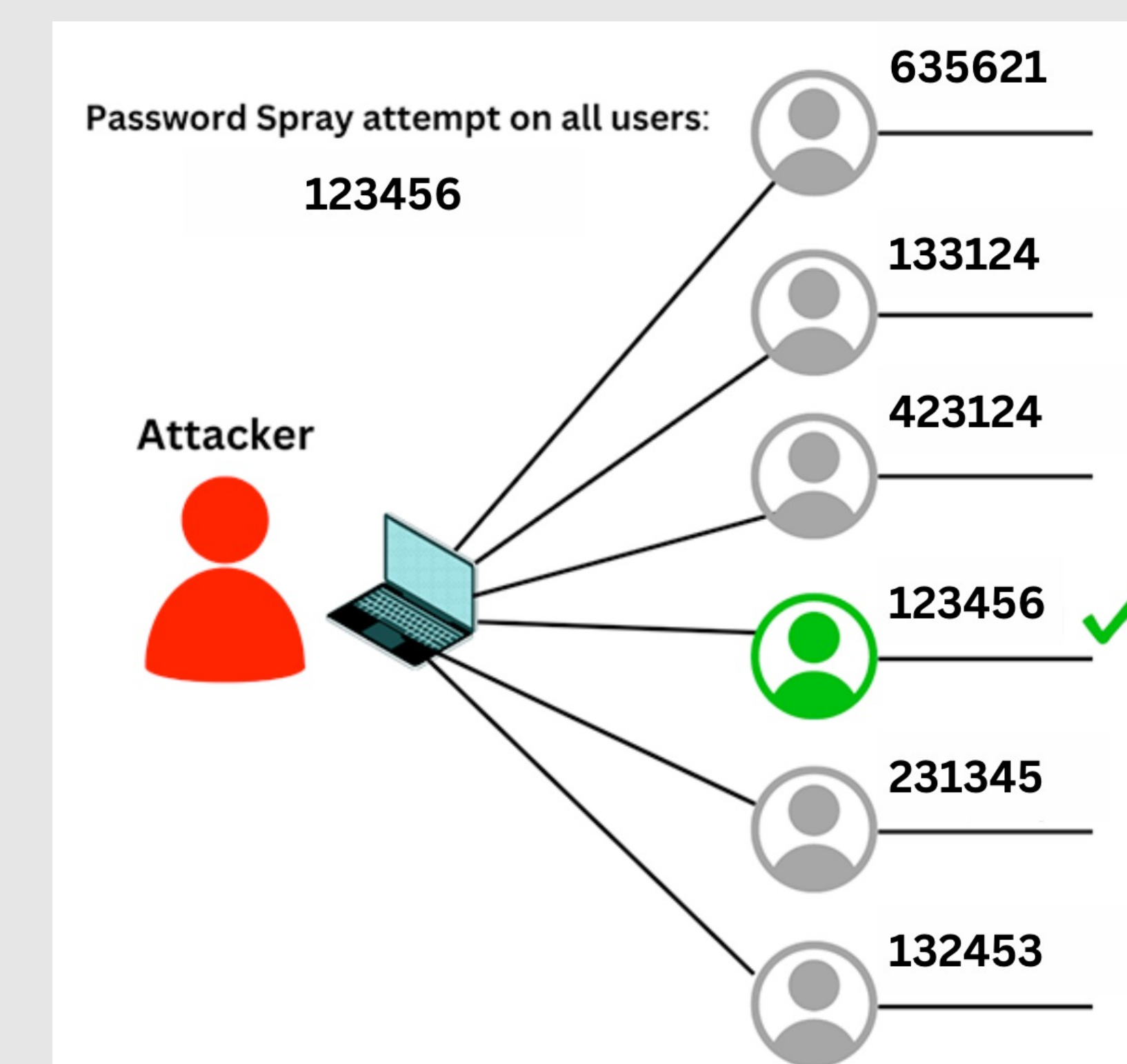
- **Validate:**  
Rigorously check all input against strict rules.
- **Sanitize:**  
Remove or convert dangerous characters.
- **Use Safer Alternatives:**  
Employ libraries or APIs designed for executing commands.

1

## PASSWORD SPRAYING

Our system uses pin codes: passwords are 6 digits.

$$6^{10} = 60,466,176$$



### Prevention:

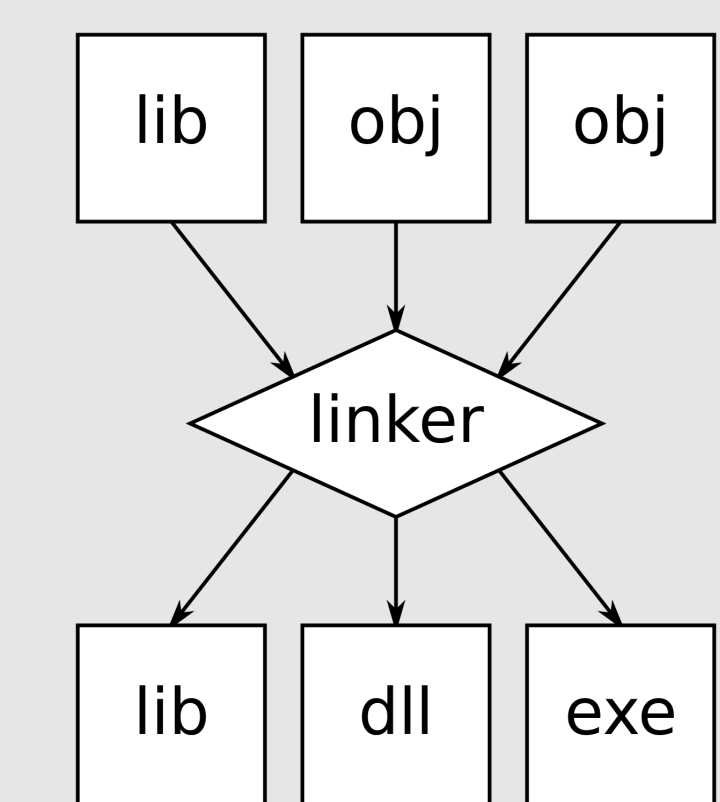
- **Block Brute Force:**
  - Limit failed logins per IP address to 10.
- **Stronger Passwords:**
  - More character variations
  - 6 characters from letters and numbers =  $6^{62}$  variations
  - Simply longer passwords

2

## LD\_PRELOAD

### Before executing a script:

- **Magic number & permissions**
  - Type of file
  - User and group permissions
- **Resolving links**
  - Which libraries to use
  - File dependencies



- **Loading shared libraries**
  - Malloc(), printf()
- **Trick:**
  - Load a shared library containing **void \_init()** using **LD\_PRELOAD**
  - Run a reverse shell from C with elevated privileges

### Prevention:

- Limit sudoers permissions as much as possible

## EXTENSIONS

**Cron Jobs:** Scheduled tasks for repeated attacks

**Systemd Jobs:** System services for stealthy execution

**.bash\_profile Autostart:** Launch attacks at login

**SSH Keys:** Authorize your ssh keys for keyless login as root

**Removing History:** remove bash execution history

