# Nguyên lý cơ bản của Stack

- **Stack frame**: mỗi lần hàm được gọi → tạo stack frame mới.
- **EBP** (base pointer): đánh dấu gốc stack frame.
- **ESP** (stack pointer): trỏ đỉnh stack.
- **Return address**: lưu địa chỉ lệnh sẽ quay lại sau khi hàm kết thúc.
- **Local variables** (vd: `char password[64]`) cũng nằm trên stack.

Điểm yếu: **return address** và **local variables** gần nhau ⇒ nếu input tràn buffer, attacker có thể ghi đè return address.

# Từ Overflow → Arbitrary Code Execution

- Nếu nhập vào >64 bytes cho `password[64]`:
  - 64 byte đầu → lấp đầy buffer.
  - 4 byte tiếp theo → ghi đè saved EBP.
  - 4 byte sau nữa → ghi đè return address.
- Nếu return address bị ghi thành `0x41414141` (AAAA) ⇒ crash.
- Nếu return address bị ghi thành địa chỉ hàm hữu ích (vd: `go_shell()`) ⇒ **chạy code attacker mong muốn**.

# Shellcode

- Mục tiêu thật sự không chỉ gọi `go_shell()` mà **inject arbitrary code**.
- Shellcode được nhét trực tiếp vào buffer ⇒ sau đó return address được set trỏ về buffer ⇒ code chạy từ chính input.

## Tính chất shellcode:

1. **Nhỏ gọn** (fit vào buffer).
2. **Position-independent** (không phụ thuộc địa chỉ cố định).
3. **Không chứa null byte** (`\x00`) vì các hàm C coi đó là string terminator.
4. **Tự chứa** (không gọi data ngoài).

Ví dụ payload:

- Spawn shell (`/bin/sh`).
- Thêm user admin.
- Reverse shell.

# Kỹ thuật NOP Sled

- Thay vì để EIP nhảy chính xác vào đầu shellcode (khó đoán), attacker đặt nhiều **NOP instructions** trước shellcode.
- Nếu EIP rơi vào vùng NOP ⇒ CPU sẽ "trượt" đến khi gặp shellcode.

Payload dạng:
[NOP NOP NOP ...][Shellcode][AAAA][Addr trỏ vào buffer]

# Demo Payload

Ví dụ overflow `password[64]`:

<NOP x k lần><Shellcode><AAAA><Addr buffer>

- 64 bytes đầu = NOP + shellcode.
- Overwrite EBP = AAAA.
- Overwrite return address = địa chỉ buffer (nơi shellcode nằm).

Khi hàm return → EIP trỏ về buffer → shellcode chạy.

# Key Takeaways

- **Stack overflow** = ghi tràn local buffer ⇒ ghi đè control data (EBP, return address).
- **Goal**: chuyển EIP đến code attacker mong muốn (shellcode, hoặc function có sẵn như `system("/bin/sh")`).
- Cách tối ưu: **NOP sled + shellcode injection**.
- Stack overflow là kỹ thuật cơ bản nhất nhưng đặt nền tảng cho các khai thác phức tạp hơn (heap, format string, ROP…).