## Meta Parameters (Global Script Settings)

| Parameter | Values | Description |
|---|---|---|
| `description` | String | Brief explanation of the script's purpose |
| `repeat` | Integer (default: 1) | Number of times to repeat the entire script |
| `enableLed` | `true` (default) / `false` | Enable/disable LED feedback during execution |
| `savePcap` | `true` (default) / `false` | Automatically save captured packets to PCAP files |

---

## Stage Commands (Executed Sequentially)

### 1. `scan`

Scan for WiFi devices.

json
Copy

```
"scan": {
  "type": "ap" | "station",  // Scan access points or client stations
  "timeout": [seconds],      // Duration of the scan
  "channel": [1-11]          // Optional: Limit to a specific channel
}
```

### 2. `select`

Filter or select targets from scan results.

json
Copy

```
"select": {
  "type": "ap" | "station" | "ssid",  // Target type
  "filter": "all" | "contains -f '{SSID fragment}' | equals '{SSID}'",  // Match
criteria
  "indexes": [0, 1, 2...]             // Specific indexes from scan results
}
```

### 3. `deauth`

Launch a deauthentication attack.

json
Copy

```
"deauth": {
  "timeout": [seconds]  // Attack duration
}
```

### 4. `probe`

Send probe requests to discover hidden networks.

json
Copy

```
"probe": {
  "timeout": [seconds]  // Duration of probing
}
```

### 5. Sniffing Commands

Capture specific packet types:

- **`sniffRaw`**: All WiFi traffic
- **`sniffBeacon`**: Beacon frames
- **`sniffDeauth`**: Deauthentication packets
- **`sniffEsp`**: ESP device packets
- **`sniffPwn`**: PWNagotchi packets
- **`sniffPmkid`**: PMKID handshakes (WPA/WPA2)

json
Copy

```
"[sniffType]": {
  "timeout": [seconds],       // Sniffing duration
  "forceDeauth": true|false,  // For PMKID: Force deauth to trigger handshake
  "channel": [1-11]          // Optional: Lock to a channel (PMKID only)
}
```

### 6. `beaconList`

Spam fake beacon frames with custom/random SSIDs.

json
Copy

```
"beaconList": {
  "ssids": ["SSID1", "SSID2"],  // Custom SSIDs to broadcast
  "generate": [number],         // Generate random SSIDs (alternative to `ssids`)
  "timeout": [seconds]          // Duration of beacon spam
}
```

### 7. `beaconAp`

Impersonate a specific access point.

json
Copy

```
"beaconAp": {
  "timeout": [seconds]  // Impersonation duration
}
```

### 8. `exec`

Run raw CLI commands (advanced).

json
Copy

```
"exec": {
  "command": "[CLI command]"  // e.g., "clearlist -a", "attack -t beacon"
}
```

### 9. `delay`

Pause between stages.

json
Copy

```
"delay": {
  "timeout": [seconds]  // Delay duration
}
```

---

## Example CLI Commands for `exec` Stage

| Command | Description |
|---|---|
| `listap -a` | List all access points |
| `liststa -a` | List all client stations |
| `clearlist -a` | Clear AP/station lists |
| `attack -t deauth` | Start deauthentication attack |
| `attack -t beacon` | Start beacon spam |
| `update -a` | Update target list from scan results |

---

## Key Notes

1. **Placeholders**: Use `{SSID}` or `{MAC}` in filters to reference scanned data.
2. **Order Matters**: Stages run sequentially (e.g., `scan` before `select`).
3. **Combining Stages**:

   json
   Copy

   ```
   "stages": [
     {"scan": {"type": "ap", "timeout": 10}},
     {"select": {"type": "ap", "filter": "contains -f 'Guest'"}},
     {"deauth": {"timeout": 20}}
   ]
   ```

4. **Logs/Outputs**: Saved to SD card in `apps_data/marauder/` (e.g., `pmkid.log`, `ap_list.log`).

Let me know if you want a script for a specific attack chain (e.g., PMKID capture + deauth)!