MAVProxy Attack Module 说明文档

本文档详细说明了 attack.py 模块可以实现的各种攻击方式及其实现原理。

模块加载方式

可以通过以下几种方式加载攻击模块：

```
# QGC地面站
--master 127.0.0.1:14550 --out 127.0.0.1:14551 --cmd="module load attack"

# Gazebo仿真
--master=tcpin:127.0.0.1:4561 --out=tcp:127.0.0.1:4560 --cmd="module load attack"

# MAVROS
--master 127.0.0.1:24540 --out 127.0.0.1:24541 --cmd="module load attack"
```

MAVProxy 端口配置与数据流向

启动命令说明

```
python mavproxy.py --master=tcpin:127.0.0.1:4561 --out=tcp:127.0.0.1:4560 --
cmd='module load attack'
```

数据流向图

```
PX4 (SITL)      MAVProxy        Ground Control Station (QGC)
[Port 4561] <---> [MAVProxy] <---> [Port 4560]
```

端口配置详解

1. **输入端口 (--master)**：
    - 端口号：4561
    - 连接方式：tcpin
    - 目标：PX4 SITL 模拟器
    - 功能：接收来自飞控的 MAVLink 消息
2. **输出端口 (--out)**：
    - 端口号：4560
    - 连接方式：tcp
    - 目标：地面站（QGroundControl）

- 功能：转发处理后的消息到地面站

数据流向过程

1. **飞控到地面站**：

```
PX4 (4561) -> process_mavlink() -> 模块处理 -> 地面站 (4560)
```

   - 数据从 PX4 发出
   - 经过 process_mavlink 函数处理
   - 通过 mavlink_backward 函数进行模块处理
   - 最终发送到地面站

2. **地面站到飞控**：

```
地面站 (4560) -> process_master() -> 模块处理 -> PX4 (4561)
```

   - 数据从地面站发出
   - 经过 process_master 函数处理
   - 通过 mavlink_packet 函数进行模块处理
   - 最终发送到 PX4

Attack 模块介入点

1. **mavlink_backward 函数**：

   - 处理从飞控到地面站的数据
   - 可以修改传感器数据
   - 可以实现 GPS 欺骗
   - 可以修改状态信息

2. **mavlink_packet 函数**：

   - 处理从地面站到飞控的数据
   - 可以修改控制命令
   - 可以实现硬件后门攻击
   - 可以修改任务指令

配置特点

1. **本地环回连接**:
    - 使用 127.0.0.1 作为本地环回地址
    - 所有通信在本地完成
    - 便于调试和测试
2. **中间人位置**:
    - MAVProxy 作为中间代理
    - 可以监听所有通信数据
    - 可以修改任意数据包
    - 实现双向通信控制
3. **模块化设计**:
    - 通过 --cmd 加载攻击模块
    - 支持动态启用/禁用攻击功能
    - 便于扩展新的攻击方式

攻击类型及实现

---

1. 硬件后门攻击 (Hardware Backdoor)

- **命令**: `attack hardware_backdoor on/off`
- **目标**: 执行器控制信号
- **实现方式**:
    - 拦截 `HIL_ACTUATOR_CONTROLS` 消息
    - 修改 controls[0] 值（增加0.9）
    - 影响飞行器的执行器控制
- **影响**: 可能导致飞行器失控或执行异常动作

2. 消息修改攻击 (Message Modification)

- **命令**: `attack message_modification`
- **目标**: 任务航点信息

- **实现方式**:
  - 拦截 `MISSION_ITEM_INT` 消息
  - 修改航点坐标（x和y各增加3000）
- **影响**: 导致飞行器执行错误的任务路径

3. 传感器修改攻击 (Sensor Modification)

- **命令**: `attack sensor_modification on/off`
- **目标**: 传感器数据
- **实现方式**:
  - 拦截 `HIL_SENSOR` 消息
  - 修改传感器原始数据
- **影响**: 影响飞行器的状态估计和控制决策

4. 传感器注入攻击 (Sensor Injection)

- **命令**: `attack sensor_injection`
- **目标**: 传感器数据流
- **实现方式**:
  - 拦截 `HIL_SENSOR` 消息
  - 注入预设的虚假传感器数据

```
new_mav.hil_sensor_send(time_usec-1, 0.1, 0.1, 0.1, 0, 0, 0,
                        1, 1, 1, -3.94092e+29, 4.58141e-41,
                        -9.61733e-41, 4.58127e-36, 63, 0)
```

- **影响**: 使飞行器接收到完全错误的传感器信息

5. GPS欺骗攻击 (GPS Spoofing)

- **命令**: `attack gps on/off`
- **目标**: GPS定位信息
- **实现方式**:
  - 拦截 `HIL_GPS` 消息

    - 修改经纬度信息（各增加1000）
- **影响**: 导致飞行器定位错误，可能偏离预定航线

6. 地面站欺骗攻击 (GCS Spoofing)

- **命令**: `attack gcs on/off`
- **目标**: 地面站显示的位置信息
- **实现方式**:
    - 拦截 `GLOBAL_POSITION_INT` 消息
    - 修改显示的经纬度坐标
- **影响**: 误导地面站操作员对飞行器位置的判断

7. 速度反转攻击 (Velocity Reversal)

- **命令**: `attack reverse_velocity`
- **目标**: 飞行器速度控制
- **实现方式**:
    - 拦截 `SET_POSITION_TARGET_LOCAL_NED` 消息
    - 将vx和vy速度值取反
- **影响**: 导致飞行器朝与指令相反的方向运动

MAVProxy 数据流处理机制

MAVProxy 通过两个核心函数处理 MAVLink 数据包，使得攻击模块能够在数据传输的任何位置进行干预。

数据流向图

```
飞控 (Autopilot) <---> MAVProxy <---> 地面站 (GCS)
     ↑                    ↑              ↑
process_mavlink() <--> 模块处理 <--> process_master()
```

1. process_master 函数

功能：处理来自地面站的数据包（地面站 → 飞控）

主要处理步骤：

1. 数据接收：接收最多 16KB 的数据
2. 空数据处理：避免 CPU 空转
3. 更新字节计数器：记录数据流量
4. 原始数据日志：记录原始数据
5. MAVLink 版本检测：自动适应版本
6. 数据包解析：解析 MAVLink 消息
7. 错误处理：处理异常数据

关键特点：

- 直接处理和转发数据
- 无特殊权限控制
- 处理 BAD_DATA 错误
- 支持调试模式
- 维护数据统计

2. process_mavlink 函数

功能：处理来自飞控的数据包（飞控 → 地面站）

主要处理步骤：

1. 数据接收：从飞控接收数据
2. 版本检测：自动检测 MAVLink 版本
3. 数据解析：解析 MAVLink 消息
4. 权限检查：检查转发权限
5. 模块处理：调用模块的处理函数
6. 转发控制：选择合适的转发链路
7. 日志记录：记录处理后的数据
8. 监视功能：支持消息监视

关键特点：

- 支持模块介入处理
- 有 mavfwd 权限控制
- 智能链路选择

- 支持消息监视
- 处理 MAVError 错误

两个函数的主要区别

特性	process_master	process_mavlink
数据流向	地面站 → 飞控	飞控 → 地面站
处理方式	直接处理和转发	经过模块处理后转发
权限控制	无特殊控制	有 mavfwd 权限控制
模块介入	不直接调用模块	调用模块的 mavlink_backward
错误处理	处理 BAD_DATA	处理 MAVError

在攻击模块中的应用

1. **硬件后门攻击**:
    - 通过 process_master 拦截和修改控制命令
    - 影响执行器的实际控制输出
2. **传感器攻击**:
    - 通过 process_mavlink 修改传感器数据
    - 影响飞控的状态估计
3. **GPS 欺骗**:
    - 通过 process_mavlink 修改位置信息
    - 干扰导航系统
4. **速度反转**:
    - 通过 process_mavlink 修改速度命令
    - 影响运动控制

安全影响

1. **数据完整性**：
    - 可以修改任何 MAVLink 消息
    - 影响系统的正常运行
2. **系统控制**：
    - 可以劫持控制命令
    - 可以注入虚假数据
3. **状态监控**：
    - 可以欺骗监控系统
    - 隐藏真实状态

防护建议

1. **通信加密**：
    - 使用加密通信链路
    - 实施消息签名机制
2. **权限控制**：
    - 严格控制转发权限
    - 实施访问控制
3. **数据验证**：
    - 验证数据的合理性
    - 检测异常模式
4. **监控告警**：
    - 实时监控数据流
    - 设置异常告警机制

注意事项

1. 这些攻击功能仅用于安全研究和测试目的
2. 在实际飞行器上测试这些攻击可能导致危险情况

3. 建议在仿真环境中进行测试
4. 使用这些功能时需要充分了解可能的风险

实现原理

模块通过两个主要的数据包处理函数实现攻击：

1. `mavlink_packet(self, m, mav)`: 处理来自主链路的MAVLink数据包
2. `mavlink_backward(self, m, mav)`: 处理来自从链路的MAVLink数据包

每种攻击都可以通过命令单独开启或关闭，实现了对无人机各个子系统的针对性攻击：

- 控制系统（硬件后门、速度反转）
- 导航系统（GPS欺骗）
- 任务规划（消息修改）
- 传感器系统（传感器修改和注入）
- 地面站显示（GCS欺骗）