UNIVERSIDADE ESTADUAL DE MARINGÁ
CENTRO DE TECNOLOGIA
DEPARTAMENTO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

Paulo Roberto de Oliveira

Apoio computacional para teste de invasão em sistemas de comunicação de VANTs

Maringá
2015

Paulo Roberto de Oliveira

Apoio computacional para teste de invasão em sistemas de comunicação de VANTs

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito para obtenção do título de Mestre em Ciência da Computação.

Orientadora: Prof[a]. Dr[a]. Valéria Delisandra Feltrim

Maringá
2015

Dados Internacionais de Catalogação na Publicação (CIP)
(Biblioteca Central - UEM, Maringá, PR, Brasil)

O48a

Oliveira, Paulo Roberto de
Apoio computacional para teste de invasão em sistemas de comunicação de VANTs / Paulo Roberto de Oliveira. – - Maringá, 2015.
85 f. : il. tabs., figs.

Orientadora: Profa. Dra. Valéria Delisandra Feltrim.
Dissertação (mestrado) – Universidade Estadual de Maringá, Centro de Tecnologia, Departamento de Informática,Programa de Pós-Graduação em Ciência da Computação, 2015.

1. Segurança de dados – Comunicação de VANTs. 2. Sistemas de Comunicação – Teste de invasão. 3. Segurança da Informação – Software. I. Feltrim, Valéria Delisandra, orient. II. Universidade Estadual de Maringá. Centro Tecnologia. Departamento de Informática. Programa de Pós-Graduação em Ciência da Computação. III. Título.

CDD 22.ed.005.8

MGC - 0018342

FOLHA DE APROVAÇÃO

PAULO ROBERTO DE OLIVEIRA

Apoio computacional para teste de invasão em sistemas de comunicação de VANTs

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação do Departamento de Informática, Centro de Tecnologia da Universidade Estadual de Maringá, como requisito parcial para obtenção do título de Mestre em Ciência da Computação pela Banca Examinadora composta pelos membros:

BANCA EXAMINADORA

Profa. Dra. Valéria Delisandra Feltrim
Universidade Estadual de Maringá – DIN/UEM

Profa. Dra. Luciana Andréia Fondazzi Martimiano
Universidade Estadual de Maringá – DIN/UEM

Prof. Dr. Nardênio Almeida Martins
Universidade Estadual de Maringá – DIN/UEM

Aprovada em: 31 de julho de 2015.
Local da defesa: Sala 101, Bloco C56, *campus* da Universidade Estadual de Maringá.

Apoio computacional para teste de invasão em sistemas de comunicação de VANTs

RESUMO

Nos últimos anos os sistemas embarcados vêm ganhando muito espaço no cenário tecnológico, sendo intrinsecamente ligados aos produtos do dia a dia, tais como: o sistema de freios de um carro, o controle de uma máquina de lavar roupas, dentre outros. Existe ainda uma classe mais específica de sistemas embarcados, denominada sistemas embarcados críticos. Tal referência é atribuída por causa das funções de alto risco que eles desempenham. Os VANTs (Veículos Aéreos Não Tripulados) se enquadram nessa classe e sua segurança se torna um fator de grande importância. Os estudos já realizados evidenciam que os VANTs possuem falhas em seus sistemas de comunicação, tornando-os vulneráveis a diversos tipos de ataques. Nota-se então, que os mecanismos de segurança estão sendo mal utilizados ou são inexistentes nessas aeronaves. Sendo assim, este trabalho realizou um amplo levantamento de ameaças e vulnerabilidades em VANTs, desenvolveu um modelo para testes de invasão em VANTs, juntamente com um *software* de testes de invasão aplicado a ambientes simulados, os quais se apresentam como um passo adiante na evolução da segurança em ambientes de VANTs.

Palavras-chave: VANT, Testes de invasão, Segurança, Comunicação Segura.

Computational support for penetration testing in VANTs communication systems

ABSTRACT

In recent years, embedded systems have gained much space in the technological scenario, being intrinsically linked to products of daily life, such as the brake system of a car, the control of a washing machine, among others. There is also a more specific class called critical embedded systems. Such denomination is assigned because the high risk functions they perform. UAVs (Unmanned Aerial Vehicles) fall into this category and its security becomes an important aspect. Existing studies show that UAVs have communication system failures, which make them vulnerable to various types of attacks. Note that the security mechanisms are being used in a wrong way or absent in these aircrafts. Thus, this work conducted a broad survey of threats and vulnerabilities in UAVs, developed a general model for application of penetration testing on UAVs, along with a penetration testing software applied to simulated environments, which are presented as a step forward in the evolution of security UAVs environments.
Keywords: UAV, Penetration Testing, Security, Secure Communication.

LISTA DE FIGURAS

LISTA DE TABELAS

LISTA DE SIGLAS E ABREVIATURAS

GPS: *Global Positioning System*
ICMP: *Internet Control Message Protocol*
IP: *Internet Protocol*
JNI: *Java Native Interface*
MAC: *Media Access Control*
MANET: *Mobile Ad-hoc Network*
SITL: *Software In The Loop*
SYN: *SYNchronize*
TCP: *Transmission Control Protocol*
UAV: *Unmanned Aerial Vehicle*
UAVNet: *Unmanned Aerial Vehicle Network*
UDP: *User Datagram Protocol*
USB: *Universal Serial Bus*
VANT: Veículo Aéreo Não Tripulado
WSN: *Wireless Sensor Network*

SUMÁRIO

1

Introdução

Mediante a globalização e a concorrência, o setor de inovação tecnológica de uma empresa necessita de uma melhora gradativa de seus produtos para se manter no mercado com produtos superiores aos fabricados pelos concorrentes. Em função de tal competição nesse nicho de mercado, muitas pessoas podem ter em suas mãos aparelhos considerados inteligentes.

Tais aparelhos são assim caracterizados por possuírem funcionalidades computacionais embutidas. Por exemplo, uma geladeira que avisa quando o gelo armazenado está acabando, um aparelho que desliga automaticamente tendo uma pré-configuração, um aparelho de telefone celular que pode realizar chamadas por um simples comando de voz, dentre outros. Todos esses sistemas possuem módulos computacionais (incluindo *hardware* e *software*) que são denominados sistemas embarcados.

Um sistema embarcado é um sistema computadorizado composto por *hardware* e *software* altamente acoplados, sendo que dependendo de seu tipo podem existir recursos adicionais, sejam mecânicos, de *software* ou elétricos, para determinado contexto do sistema (Li e Yao, 2003).

A crescente necessidade por melhor desempenho das aplicações aliada à limitação no consumo de potência dos sistemas embarcados impactam no esforço de se utilizar os melhores algoritmos possíveis, ou seja, aqueles com menor consumo de energia e melhor desempenho.

Existem também os sistemas embarcados baseados em plataformas, em que uma plataforma de *hardware* é escolhida para que o *software* seja integrado a ela. Como exemplo deste tipo de plataforma tem-se a *Beagleboard* (Coley, 2009).

Uma subcategoria ainda deve ser destacada, a dos sistemas embarcados críticos. Estes são classificados de tal maneira, pois em uma eventual falha ou mau funcionamento podem colocar vidas em risco, causar danos em equipamentos de alto custo, danos ambientais ou grandes perdas financeiras. Um exemplo interessante é o de um sistema de controle de avião, que caso o tempo de resposta não seja atendido, este pode cair provocando uma catástrofe (Berger, 2001). Outros exemplos de sistemas embarcados críticos são: usinas nucleares, sistemas médicos, VANTs (Veículos Aéreos Não Tripulados), dentre outros.

Muitos desses sistemas podem ser conectados em rede e se comunicar com outros dispositivos. Diante dessa característica, existem indivíduos interessados em capturar ilicitamente os dados que trafegam na rede, para que posteriormente possam tirar proveito. Para garantir um alto nível de segurança e dificultar o trabalho desses indivíduos, medidas devem ser tomadas.

A segurança é uma preocupação evidente, pois esta deve garantir que pessoas mal intencionadas não possuam acesso ou não estejam aptas a modificar mensagens secretas enviadas (Tanenbaum, 2003).

O objetivo dos invasores no mundo digital não se difere do objetivo dos fraudadores no mundo real, pois em ambos os casos eles tentam tirar proveito de recursos destinados a outros. Todos os tipos de comércio já inventados foram alvo de fraudes e no comércio eletrônico não é diferente (Rezende, 1998).

No mundo real é possível se passar por outra pessoa, porém esta tarefa não é simples. Já no digital tal tarefa possui um nível de dificuldade menor. Observa-se então que com o passar do tempo, a segurança vai ganhando mais importância e se tornando um aspecto primordial dentro de uma aplicação.

Após a apresentação dessas características, constata-se que os sistemas embarcados críticos merecem cuidados especiais, desde o seu planejamento até a sua manutenção, tendo sempre como objetivo principal a garantia de um alto nível de segurança em todas e quaisquer circunstâncias, para que não ocorram danos que possam prejudicar seriamente pessoas, empresas, instituições governamentais e outros.

Como os VANTs são sistemas críticos e estão sendo amplamente empregados nos mais diversos tipos de atividades, a segurança dessas aeronaves é um fator a ser tratado cuidadosamente. Os testes de invasão apresentam uma abordagem prática de detecção e correção de possíveis vulnerabilidades, no entanto, sua aplicação em sistemas embarcados tais como os VANTs é pouco explorada.

Sendo assim, as motivações para este trabalho foram: a crescente utilização de VANTs e seu mercado; os recentes ataques que ocorreram contra essa aeronaves, mostrando a

fragilidade que elas possuem; a pequena quantidade de estudos referentes à segurança dos VANTs; e o grande campo a ser explorado nessa área.

Dentro deste contexto, o principal objetivo deste trabalho foi desenvolver um *software* capaz de realizar testes de invasão em sistemas de comunicação de VANTs em ambiente simulado, denominado *SITL PenTest (Software In The Loop - Penetration Testing)*, o qual possibilita que sistemas mais seguros possam ser desenvolvidos. Como objetivos específicos têm-se:

1. **Analisar as principais vulnerabilidades em VANTs**;
2. **Analisar os principais tipos de ataques que ocorrem contra VANTs**;
3. **Estudar o comportamento de ferramentas de teste de invasão**;
4. **Desenvolver um modelo para aplicação de testes de invasão em VANTs**;
5. **Projetar e desenvolver o *software* de apoio computacional**;
6. **Integrar o *software* desenvolvido a um simulador de VANT**;
7. **Avaliar a ferramenta realizando testes de invasão em sistemas de comunicação de VANTs em ambiente simulado**.

O texto está organizado como segue: O Capítulo 2 apresenta uma revisão de literatura sobre segurança em VANTs, colocando em evidência as vulnerabilidades já encontradas. A fundamentação teórica sobre testes de invasão também é abordada nesse capítulo. No Capítulo 3 é apresentado o estado da arte dos trabalhos desenvolvidos até o presente momento, que têm o intuito de maximizar o nível de segurança encontrado atualmente nos VANTs. Já o Capítulo 4 apresenta o modelo para aplicação de testes de invasão no contexto dos VANTs. Já o Capítulo 5 apresenta as ferramentas utilizadas no desenvolvimento do *software SITL PenTest*, suas interfaces com o usuário, como utilizá-lo e os resultados de sua execução. Por fim, o Capítulo 6 apresenta as conclusões do trabalho, as contribuições, e os possíveis trabalhos futuros.

2

Fundamentação teórica: VANTs e Teste de Invasão

Neste capítulo são apresentados os conceitos relacionados aos VANTs, suas principais vulnerabilidades em relação à segurança, alguns trabalhos que discutem tais vulnerabilidades e a técnica de teste de invasão, a qual é a base para o desenvolvimento deste trabalho.

2.1 Veículos Aéreos Não Tripulados

Os VANTs (ou Drones) são todas as aeronaves controladas remotamente ou até mesmo de forma autônoma, sem a necessidade de um piloto para guiá-las (Staff, 2001). Tal característica os tornam altamente dependentes do bom funcionamento de seus subsistemas e dos canais de comunicação que os conectam com entidades externas. Os VANTs foram projetados com o intuito de serem utilizados em um ambiente militar, operando em locais de difícil acesso e em locais de alto risco, como por exemplo: fronteira entre países, áreas de mata fechada, regiões de guerra, dentre outros.

Ao se definir o que é um VANT, deve-se observar a diferença entre dois conceitos fundamentais: **automatismo** e **autonomia**.

Um sistema chamado de **automático** é aquele que executa uma determinada ação selecionada dentro de uma faixa de valores previamente estabelecidos. Um exemplo é o piloto automático de um avião, carro ou outro veículo, que é capaz de manter o mesmo sob seu controle e em uma rota previamente determinada, porém, não está apto a realizar o processo de navegação.

Já o sistema **autônomo** é aquele capaz de "tomar decisões", ou seja, ele possui a habilidade de avaliar quais as melhores ações a serem executadas para alcançar seu objetivo. Um exemplo de autonomia é um piloto humano de uma aeronave, que recebe um objetivo de voo, traça a rota e então navega até o seu destino. Tal processo é chamado de navegação.

Diante dos conceitos de automatismo e autonomia, nota-se que atualmente a maioria das aeronaves projetadas possuem mecanismos automáticos, porém, a autonomia está atrelada ao piloto (um ser humano). Já os VANTs, além de possuírem o sistema automático, possuem também um sistema autônomo. Dessa maneira, um VANT pode ser caracterizado como automático ou automático e autônomo.

Como os VANTs não possuem tripulação, seu tamanho e peso são reduzidos em relação aos aviões tripulados. Essa característica é muito importante em VANTs militares, pois eles se tornam mais difíceis de serem notados por radares de forças inimigas. Além disso, as missões executadas por essas aeronaves podem ser mais longas, pois elas não possuem limitações físicas, diferentemente dos seres humanos.

Pesquisas realizadas evidenciam que entre os anos de 2004 e 2008, o número de VANTs empregados em missões subiu consideravelmente, saindo da faixa de 1000 para alcançar a faixa de 5000 em operação. Tal trabalho ainda afirma que o rápido crescimento na área é em grande parte devido aos esforços dos Estados Unidos da América, pois o país investe fortemente na área, sendo tal investimento maior que o de qualquer outro país no mundo (Frost e Sullivan, 2007).

Com o aumento do número de pesquisas realizadas abordando VANTs, a diminuição do preço de seus componentes e a criação de projetos de VANTs de *hardware* e *software* abertos, tais aeronaves se tornaram populares de tal maneira que atualmente são amplamente empregadas nos mais diversos tipos de aplicações civis, tais como monitoramento de plantio, monitoramento de obras, avaliação de desastres, mapeamento de trânsito, dentre outras.

Atualmente, existem vários tipos de VANTs, desde aqueles de pequeno porte utilizados para diversão até os de grande porte geralmente utilizados em missões militares. Tal diferença pode ser notada na Figura 2.1. Podem existir entre eles grande variação de características, muitas vezes decorrente do tamanho, por exemplo: o *payload* que cada um pode transportar, a altura que cada um pode voar, a capacidade da bateria, dentre outras. Uma classificação de alguns tipos de VANTs existentes é apresentada na Tabela 2.1, que leva em consideração os aspectos de peso, kilometragem, altitude de voo e o tempo que cada tipo de aeronave pode voar sem a necessidade de reabastecimento, seja recarregamento de bateria ou outra fonte de energia.

Figura 2.1: Dois dos diversos tipos de VANTs existentes atualmente e sua diferença de tamanho.

Os VANTs não são compostos por um único sistema, mas um conjunto deles é responsável por desempenhar todas as funções da aeronave. Observando os VANTs de uma maneira simplista, os principais subsistemas (compostos por *hardware*, *software* ou ambos) existentes são: sistema base, sistema aviônico, sistema de comunicação e sistema de sensores.

O sistema base é responsável por interconectar os outros componentes existentes e controlá-los, agindo de uma maneira parecida com um sistema operacional do VANT.

O sistema aviônico realiza o gerenciamento da parte física da aeronave, transformando os comandos recebidos em comandos analógicos, como comandos de motor, estabilizadores, leme, dentre outros.

O sistema de comunicação é responsável por receber e enviar dados para entidades externas ao VANT, realizando tal trabalho sempre por meios de comunicação sem fio.

Por fim, o sistema de sensores consiste de todo o equipamento sensorial presente na aeronave, integrado com funcionalidades de pré processamento. Alguns dos sensores comumente utilizados são: radar, receptor GPS, sensor de altitude, sensor de velocidade, giroscópio, dentre outros. Dessa maneira, um VANT é composto basicamente de quatro subsistemas (que também podem ser subdivididos em uma análise mais precisa). Todos eles são apresentados na Figura 2.2.

Tabela 2.1: Tabela de classificação de VANTs. Adaptado de (Samad et al., 2013)

Nome da categoria	Massa (Kg)	Faixa (Km)	Altitude de voo	Resistência (Horas)
Micro	<5	<10	250	1
Mini	<25	<10	<300	<2
Baixo alcance	25 - 150	10 - 30	3000	2 - 4
Médio alcance	50 - 250	30 - 70	3000	3 - 6
Longo alcance	>250	>70	>3000	>6

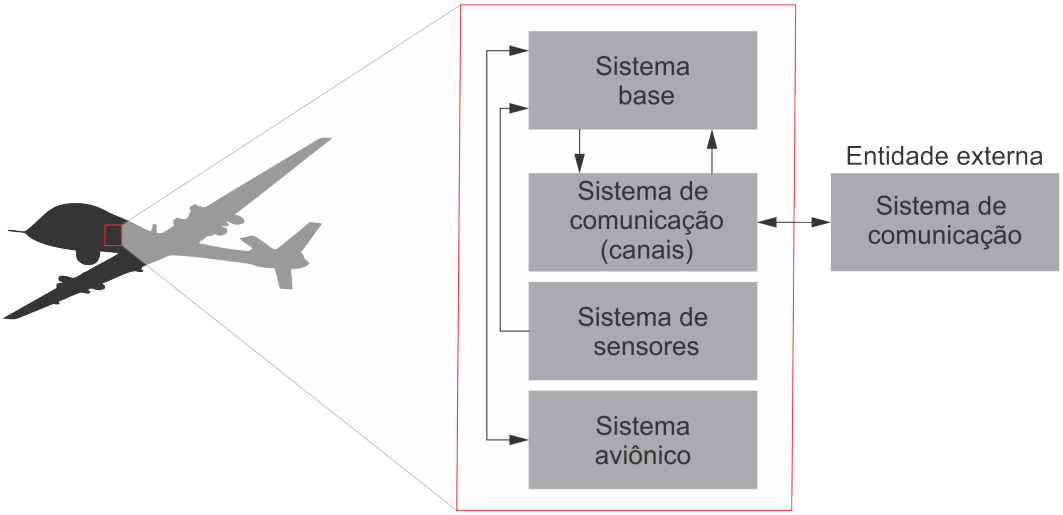

Figura 2.2: Os subsistemas essenciais de um VANT. Adaptado de (Hartmann e Steup, 2013)

Além dos subsistemas que compõem o núcleo de um VANT, este pode possuir subsistemas adicionais de acordo com o seu objetivo, por exemplo, em um VANT militar é de grande interesse que exista um subsistema de armamentos, pois um de seus propósitos é realizar ataques por meio de equipamento bélico. Outro subsistema que atualmente está presente em diversos VANTs é o de voo autônomo, que realiza tal processo de forma que não seja necessária a intervenção de um humano sequer para pouso. Outros subsistemas adicionais ainda podem ser encontrados, no entanto, os tipos de missões que a aeronave executa é de fundamental importância para a inclusão ou não desses subsistemas.

Até aqui os VANTs foram tratados como entidades únicas e independentes, no entanto, grande parte dessas aeronaves utilizam canais de comunicação para trafegar dados com entidades externas. Esses canais de comunicação existentes entre um VANT e entidades externas são dos seguintes tipos:

- **VANT - Estação Base Local**
- **VANT - Satélite**
- **VANT - Estação Base Portátil**

- **VANT - VANT**

Além desses canais de comunicação, os VANTs também coletam dados do ambiente em que sobrevoam por meio de seu sistema de sensoreamento.

Diante de todos os canais de comunicação e de coleta de dados que estão presentes em um sistema de comunicação de um VANT, observa-se que grande quantidade de dados pode estar em tráfego em um determinado momento, sendo que muitos deles são confidenciais ou até mesmo essenciais para o controle da aeronave. Dessa forma, é importante ressaltar que os VANTs se enquandram na categoria de sistemas embarcados críticos, uma vez que em uma eventual falha ou mal funcionamento de seu sistema, uma catástrofe pode ocorrer. Sendo assim, a segurança dos dados que trafegam no sistema de comunicação é um fator de substancial importância, pois um eventual vazamento dos dados, uma falsificação ou a indisponibilidade de enviar ou receber novos dados por parte de um dispositivo qualquer pode ocasionar problemas em grande escala, como por exemplo, uma catástrofe aérea.

A subseção seguinte apresenta alguns dos ataques recentes que ocorreram contra VANTs e que foram divulgados.

2.1.1 Ataques recentes a VANTs

Os VANTs foram idealizados e desenvolvidos em um primeiro momento focando sua utilização em um ambiente militar. Sua incorporação em missões de guerra, de vigilância e em outros ambientes hostis, veio acompanhada de vários incidentes de segurança. No entanto, até o ano de 2007 não foram registrados casos de ataques a VANTs, devido à pouca popularidade que eles possuíam até então.

O primeiro incidente de segurança ocorrido com um VANT, e publicamente conhecido, foi no ano de 2009. Militantes iraquianos utilizaram um *software* avaliado em $ 26,00 para interceptar vídeos ao vivo de canais de comunicação insegura utilizados pelo VANT. O *software* é conhecido como *SkyGrabber*, sendo utilizado juntamente com uma antena de satélite para realizar a captura de vídeos (Gorman et al., 2009).

Já em outubro de 2011, um *malware* foi encontrado nas estações base utilizadas para controlar os VANTs *Predator* e *Reaper*. O problema foi descoberto depois de uma conexão com um disco rígido portátil. O vírus se espalhou por vários computadores e felizmente não interrompeu as missões que os VANTs executavam, nem vazou informações confidenciais a pessoas não autorizadas (Mansfield et al., 2013).

Outro ataque ainda ocorreu no ano de 2011, no mês de dezembro. Um VANT americano do modelo RQ-170 *Sentinel* recebeu um sinal de rádio enviado à meia milha

de distância, enganando-o. Para o VANT parecia ser um comando para voltar até a base militar situada no Afeganistão, porém de fato, as coordenadas que ele seguia eram falsas, fazendo-o pousar em outro lugar. Após o pouso, foram identificados pequenos danos à aeronave, sendo esta capturada por Iranianos (Bhatti et al., 2012).

Mais recentemente em julho de 2012, foi realizada uma pesquisa pela Universidade do Texas em parceria com o Departamento de Segurança Nacional dos Estados Unidos, com o intuito de demonstrar a habilidade de sequestrar um VANT militar. Cerca de $ 1000,00 foram gastos com equipamentos para realizar um ataque conhecido como *spoofing*. Tal ataque consiste em emitir falsos sinais ao GPS da aeronave. Os pesquisadores foram capazes de assumir o controle da aeronave, evidenciando o quão perigosa é a utilização de um VANT que possui vulnerabilidades (Humphreys, 2012).

Todos os ataques apresentados reforçam o fato de que a segurança dos VANTs deve deixar de ser um fator secundário e, se tornar parte do processo de desenvolvimento dessas aeronaves. Métodos seguros de se trafegar dados nos canais de comunicação existentes em um sistema de VANT devem ser discutidos e implementados.

2.1.2 Uma análise de segurança dos VANTs

A comunicação sem fio é uma característica que está presente em grande parte dos sistemas embarcados atuais. O tráfego de dados aumenta a necessidade de privacidade das informações para que somente as entidades autorizadas tenham acesso. Sendo assim, a solução para questões sobre segurança em relação à comunicação é imprescindível para apoiar o próprio caráter crítico desses sistemas.

Os sistemas embarcados necessitam lidar com aspectos de segurança constantemente (Venugopalan et al., 2003), (Wollinger et al., 2004) e (Ravi et al., 2004).Os principais aspectos são: a identificação e a autenticação das entidades envolvidas no ambiente, os mecanismos de acesso, a disponibilidade dos serviços, o armazenamento das informações e a comunicação entre as entidades. Adiante são discutidas questões relacionadas à segurança e à comunicação dos VANTs.

a) Comunicação em VANTs

Uma rede de comunicação utilizada em VANTs não é uma rede comum, e nem similar a uma rede de computadores tradicional (Javaid et al., 2012). Os autores relatam que muitos pesquisadores tratam esse tipo de rede como se fossem WSNs (*Wireless Sensor Network*) e também MANETs (*Mobile Ad-hoc Networks*), porém, ela possui características que as difere de todas as outras. Alguns exemplos são a quantidade de informações trafegadas

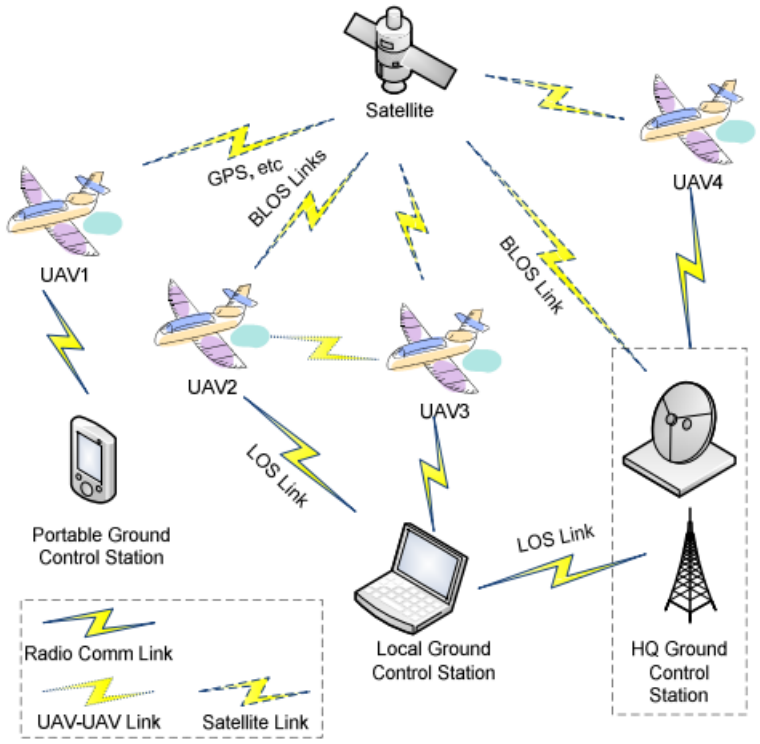

Figura 2.3: canais de comunicação existentes em uma rede de VANTs (Javaid et al., 2012)

em uma UAVNet (assim chamada uma rede utilizada em VANTs) sendo maior do que outros tipos de rede, e o número de nós, que em uma WSN é muito menor do que em uma UAVNet.

Na Figura 2.3 é possível visualizar os diferentes componentes e canais de comunicação que fazem parte de uma UAVNet. Em cada canal de comunicação pode trafegar um tipo diferente de dados. Geralmente três tipos de comunicação são encontrados nas UAVNets: comunicação de rádio, comunicação entre dois VANTs distintos (chamado neste trabalho de *UAV-UAV link*), e um canal de comunicação via satélite.

Nos canais de comunicação via rádio trafegam dados de telemetria, vídeo, áudio, controle e outros. Já nos canais de comunicação via satélite trafegam dados de GPS, que indicam posição, velocidade e condições meteorológicas. No canal de comunicação entre dois VANTs *(UAV-UAV link)* trafegam dados de posição, rotas, missão e outros.

Um trabalho interessante foi desenvolvido com foco na comunicação entre as entidades VANT e estação base (Chapman et al., 2007), criando-se um ambiente simulado que permite o tráfego de dados entre as duas entidades citadas. Tal trabalho pode servir de base para a realização de testes sobre uma plataforma de comunicação como essa.

b) Importância da segurança em canais de comunicação

Os canais de comunicação fornecem um grande poder para os mais variados tipos de sistemas, sejam eles embarcados ou não. Para os VANTs eles são fundamentais, pois todas as informações necessárias de posição de vôo, condições meteorológicas, velocidade, entre outras são fornecidas por meio deles.

Sendo assim, as questões relacionadas à segurança da comunicação em aeronaves não tripuladas devem ser tratadas com cautela, pois como essas fazem parte do grupo de sistemas embarcados críticos, a infiltração de *crackers* pode causar graves consequências.

Um exemplo que apresenta a importância dos canais de comunicação em aplicações de VANTs são aqueles utilizados no monitoramento de mísseis balísticos por meio de redes. Os veículos aéreos empregados neste tipo de missão têm o objetivo de patrulhar e vigiar determinadas áreas estratégicas de um país que possa ser alvo de mísseis. O maior problema de realizar tal tarefa é que os mísseis podem atingir velocidades muito altas, desta maneira o trabalho de detecção, rastreamento e eliminação devem ser realizados em um período de tempo bem curto.

Para a realização dessa tarefa, os projetistas de redes de defesa possuem o objetivo de interceptar um míssil em sua fase de impulsão (*boost*), que demora cerca de 2 a 5 minutos (Javaid et al., 2012). Caso não seja detectado dentro do tempo hábil, o míssil pode se movimentar em alta velocidade podendo então ir de encontro com o alvo. No entanto, os sensores de detecção de mísseis balísticos trabalham com uma rede de sensores sem fio híbrida, que por sua vez deve ter alta disponibilidade e atender a requisitos de segurança (Katopodis et al., 2007).

Um outro exemplo notável foi apresentado anteriormente, o ataque a um VANT americano em 2011, que foi capturado por Iranianos. Esse fato ocorreu devido à uma vulnerabilidade existente em aparelhos receptores de GPS, fazendo uso da técnica de invasão conhecida por *spoofing*.

Nota-se então que no caso do monitoramento de mísseis, se o canal de comunicação utilizado para tal propósito falhar ou ficar indisponível por alguns minutos, podem ocorrer sérios prejuízos para um país inteiro. Já o que aconteceu em terras iranianas foi a exploração de uma vulnerabilidade, sendo que tal fato pode também causar catástrofes.

Sendo assim, os canais de comunicação para aplicações de alta criticidade como os dois exemplos apresentados devem fazer o melhor uso possível de mecanismos de segurança, dificultando invasões.

2.1.3 Ameaças à segurança de VANTs

Os VANTs se comunicam com vários tipos de dispositivos por meio dos canais de comunicação como já citado. É inevitável se deparar com ameaças como: perda de comunicação de rádio, intervenção das informações transmitidas, ou ainda controlar eventuais intervenções externas (Rudinskas et al., 2009). Os autores ainda afirmam que as ameaças podem ser de dois tipos: àquelas dependentes de humanos e as independentes.

Alguns exemplos de ameaças dependentes de humanos são: escutas aos canais de comunicação, alteração, perda ou destruição de informações e bloqueio do canal. Um tipo de ameaça independente de humanos é um fenômeno natural que pode ter efeito na qualidade do sinal e não na segurança, como por exemplo, uma chuva de raios.

Independentemente de qual propósito um VANT possua, medidas em relação à segurança das informações trafegadas devem ser tomadas (Rudinskas et al., 2009). Portanto, a prevenção é o melhor caminho a se seguir, para que acessos não autorizados a informações privilegiadas possam servir de ferramentas para futuros ataques ao próprio VANT, ou aos sistemas embarcados.

Todas e quaisquer ameaças de segurança que possam se tornar verdadeiros problemas seja em um sistema Web, em um sistema embarcado e mais especificamente falando, em um VANT, possuem uma característica em comum, elas exploram vulnerabilidades existentes no sistema, sendo que todas podem ser enquadradas em um dos três tipos: ameaças à confidencialidade, integridade e disponibilidade. Todas elas são sensíveis às missões que um VANT executa, pois caso a confidencialidade, a integridade ou a disponibilidade do sistema seja afetada, a missão pode ser comprometida (Javaid et al., 2012). Nas subseções seguintes são apresentadas as ameaças comumente encontradas nos VANTs.

a) *Flooding*

O propósito de um ataque de *flooding* é inundar o seu alvo com um elevado número de pacotes, de maneira que esse fique extremamente lento afetando o seu desempenho, ou até mesmo indisponível (Bidgoli, 2004). O sistema alvo em questão pode ser desde uma aplicação ou um serviço específico, até uma rede inteira.

Um dos mais conhecidos tipos de *flooding* é o chamado *SYN flooding*, apresentado na Figura 2.4, oriundo da palavra *Synchronyze*. Tal ataque é realizado da seguinte maneira: o invasor envia uma série de requisições para o servidor, sendo que todas elas não possuem a última mensagem do *Three-Way Handshake* (processo inicial de conexão do protocolo TCP *Transmission Control Protocol*). Nesse caso, o servidor aguarda um tempo pelas

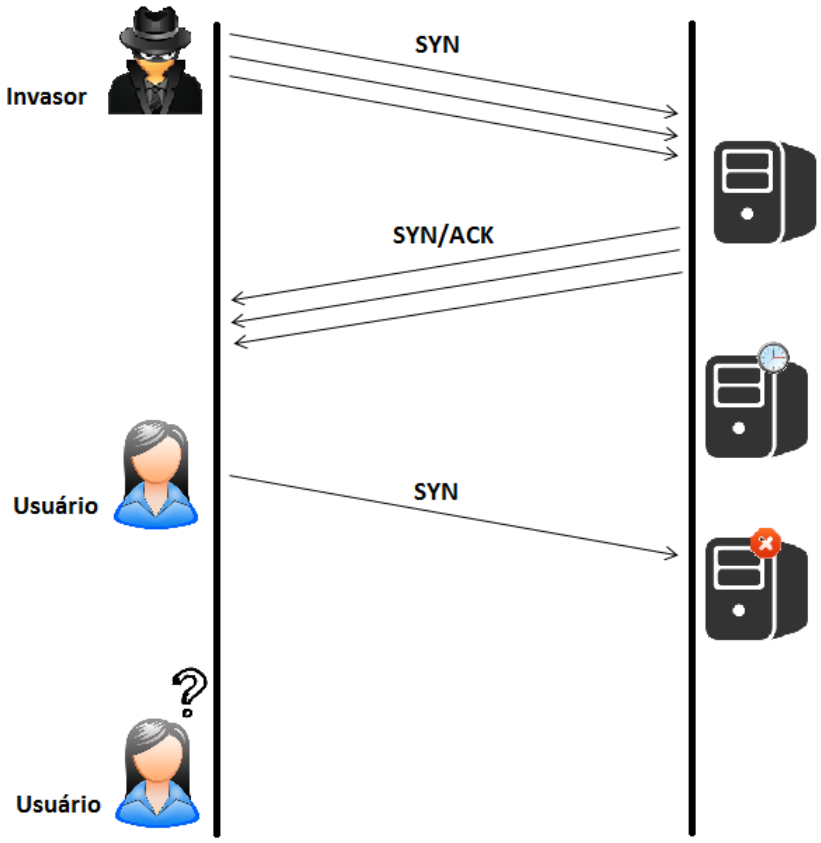

Figura 2.4: *SYN flooding*: Ataque de *flooding* muito comum

mensagens faltantes. Como elas não chegam e o servidor continua recebendo requisições de outros clientes, ele fica sobrecarregado não conseguindo atender toda a demanda, indisponibilizando o sistema para clientes legítimos (Schuba et al., 1997).

As ferramentas utilizadas para desferir ataques de *flooding* geralmente utilizam segmentos SYN (*SYNchronize*), segmentos UDP (*User Datagram Protocol*), pacotes ICMP (*Internet Control Message Protocol*) e ping.

O fato de que ataques de *flooding* interferem diretamente na disponibilidade do sistema, serviço ou rede alvo, faz com que eles sejam caracterizados como um tipo de ataque de negação de serviço.

b) *Jamming*

O ataque conhecido como *jamming* tem por objetivo romper uma comunicação por meio de colisões entre pacotes antes da recepção, ou a inserção de ruídos no canal de

Figura 2.5: Tipos de *jamming* e sua suas subdivisões

comunicação. Aquele que executa um ataque de *jamming* é denominado *Jammer*, que por sua vez interfere no tráfego de dados físicos em uma comunicação sem fio. Ao se realizar tal ataque, o *jammer* utiliza mecanismos que consomem a capacidade do canal de comunicação, total ou parcialmente. Dessa maneira, são desencadeadas falhas na comunicação, podendo ser de forma alternada ou até mesmo permanente.

O *jamming* é um tipo de ataque que pode ser realizado de maneira simples. Para o *jammer* é interessante, pois ele necessita de poucos recursos, porém para a rede em que o ataque é lançado um alto impacto é gerado (Barros, 2011). Além disso, o ataque de *jamming* pode ser aplicado tanto à camada de enlace quanto à camada física. Na primeira, o *jammer* causa interferências na rede reagindo conforme os quadros que são recebidos do meio físico, já na camada física são utilizadas ondas eletromagnéticas que corrompem os enlaces de comunicação.

Existem diferentes tipos de estratégia que um *jammer* pode utilizar para realizar a interferência de um canal de comunicação sem fio. Como consequência das diferentes filosofias existentes em ataques *jamming*, eles possuem diferentes níveis de efetividade, necessitando ainda de diferentes técnicas de detecção (Barros, 2011).

As técnicas de ataques de *jamming* podem ser subdivididas em ativos e intermitentes. Na categoria de *jamming* ativo, existem ainda o *jamming* constante e o deceptivo. Já nos ataques de *jamming* intermitente, existem o aleatório e o reativo, conforme apresentado na Figura 2.5.

O *jamming* constante emite sinais de rádio continuamente, não aguardando o canal de comunicação ficar inativo antes de iniciar o envio. Os bits enviados são aleatórios, não

seguindo as regras da camada de enlace. Sendo assim, o canal de comunicação estará sempre ocupado quando outros nós necessitarem enviar dados.

No *jamming* deceptivo, ao invés de enviar bits aleatórios, são enviados constantemente pacotes válidos, ou seja, pacotes que seguem as regras da camada de rede. Dessa maneira, os outros nós acreditam que existe um tráfego legítimo na rede, mantendo o rádio em modo de recepção. Em sistemas com capacidade de bateria limitada, tal ataque se torna um grande problema.

O *jamming* aleatório não realiza o envio dos pacotes ao canal de maneira constante, o envio é alternado entre o tempo de execução do *jamming* e tempo de espera. Dessa maneira, durante um período de tempo previamente determinado é realizado o envio de pacotes, logo após o *jammer* desliga, também por um determinado tempo.

O objetivo do *jamming* reativo não é deixar o canal de comunicação bloqueado todo o tempo. Nesse tipo de ataque utiliza-se a visão de que quando não existe tráfego na rede, não é necessário lançar o ataque. Sendo assim, o dispositivo utilizado para o ataque fica inativo juntamente com o canal de comunicação, porém, escutando-o. Quando o dispositivo detecta tráfego na rede, então o ataque de *jamming* é lançado. Essa técnica é interessante, pois é mais difícil de ser detectada.

c) *Ataque de escuta (Eavesdropping)*

O ataque de escuta também conhecido como *eavesdropping attack* se refere ao monitoramento não autorizado de dados trafegados em uma eventual comunicação. A ferramenta utilizada na realização do ataque de escuta é chamado de *sniffer*. Na Figura 2.6 apresenta-se um exemplo do ataque de escuta a um canal de comunicação *Telnet*, em que um *cracker* examina o tráfego de dados entre o computador de um usuário e um servidor de arquivos. No momento em que o *cracker* percebe que aquela é uma conexão *Telnet* e os dados de autenticação (*login* e senha) estão em tráfego, ele pode então acessar o servidor com o *login* e senha coletados, pois em uma conexão *Telnet* os dados trafegados não possuem nenhum tipo de criptografia, trafegando de maneira clara no canal. Ao acessar o servidor de arquivos com os dados do usuário, o *cracker* está realizando um outro tipo de ataque, o *spoofing* de identidade, pois se passa por outra pessoa.

O ataque de escuta pode ser aplicado em vários canais de comunicação, como por exemplo, sistemas de telefone, correio eletrônico, sistemas de mensagem em tempo real e vários outros tipos de serviços existentes, principalmente aqueles dependentes da Internet.

Tal ataque é muito difícil de ser percebido, pois ao aplicá-lo a operação da rede não é afetada e continua enviando e recebendo dados normalmente. Dessa maneira, tanto o

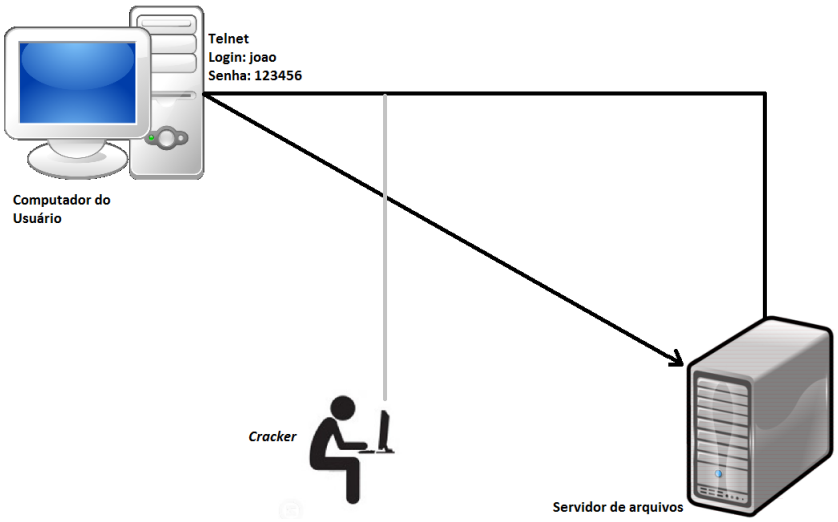

Figura 2.6: Exemplo de ataque de escuta a um canal de comunicação *Telnet*

remetente quanto o receptor dificilmente irão notar que os dados que estão sendo trocados podem estar sendo vistos por um terceiro.

Como os sistemas de comunicação sem fio e a Internet tornam-se cada vez mais populares, existe uma grande tendência das pessoas utilizarem cada vez mais os serviços *online*. Dessa maneira, o risco de uma pessoa estar sendo escutada enquanto utiliza tais serviços também aumenta.

Como os VANTs se comunicam com várias entidades por meio de um ambiente sem fio, uma grande quantidade de dados pode estar trafegando em um determinado momento. Dessa maneira, tais dados podem estar sendo capturados e analisados por meio de ferramentas *sniffer* em ataques de escuta.

d) *Cross Layer Attack*

O ataque conhecido como *cross layer* se refere à falta de interação entre as camadas de enlace e de rede. Tal ataque se propaga da camada de enlace, em que se manifesta como um ataque de negação de serviço para a camada de rede, consequentemente causando uma séria perda de desempenho na rede como um todo (Radosavac et al., 2004).

Uma pessoa mal intencionada pode congestionar o canal de comunicação utilizando-se da falta de interação entre as camadas de rede, indisponibilizando assim a comunicação entre alguns ou todos os dispositivos conectados à rede.

e) *Multi protocol Attack*

Uma rede de computadores, independentemente do seu tipo, é composta por vários conjuntos de regras, conhecidos como protocolos. Dessa maneira, os protocolos definem como os dispositivos presentes na rede devem se comportar, desde o início até o fim da conexão.

Em um determinado momento, várias instâncias de protocolos podem estar sendo executadas concomitantemente em uma rede compartilhada, tornando a verificação de segurança do canal amplamente complexa. Mesmo se tais instâncias que estão em uso em uma mesma rede estejam corretas, porém, quando executadas de maneira isolada, novos ataques podem ser inseridos na rede (Alves-Foss, 1998).

Aqueles ataques em que essencialmente mais de um protocolo deve estar envolvido, são chamados de ataques multiprotocolo. Tais ataques estão atrelados ao fato de que, se um dado protocolo de segurança é correto, então é possível construir outro protocolo de segurança também correto, porém, adaptado. A partir do momento que instâncias de ambos estejam sendo executadas em uma mesma rede, um intruso pode enviar mensagens de um protocolo realizando ataque a outro (Alves-Foss, 1998).

A título de exemplo tem-se um protocolo A, logo um protocolo B é criado e adaptado por um invasor. Dessa maneira, quando instâncias de ambos os protocolos estiverem sendo executados simultaneamente, o protocolo B pode enviar mensagens de ataque ao protocolo A.

f) *Message Modification Attack*

O ataque de modificação de mensagens se refere à alteração de cabeçalhos ou o próprio conteúdo dos pacotes que trafegam pela rede, afetando assim a integridade e até mesmo a autenticidade dos dados (Razak et al., 2004).

A alteração do cabeçalho de uma mensagem pode ocasionar o envio de dados a um destinatário diferente daquele contido no pacote original, ou enganar o destinatário com um endereço do dispositivo de origem falso. O último pode ser caracterizado como um ataque de *spoofing*, após a realização do ataque de modificação de mensagem. O ataque de *spoofing* é apresentado mais adiante.

Quando o cabeçalho de uma mensagem é alterada, existe a possibilidade de mensagens serem enviadas para o dispositivo errado, ou seja, dados secretos que eventualmente estariam trafegando na rede, poderão ser vistos por entidades não autorizadas.

Já se um invasor modifica o conteúdo de uma mensagem, esta perderá a sua integridade, podendo causar divergências entre as duas entidades que estariam se comunicando.

Em um ambiente em que VANTs trocam dados com vários dispositivos, nota-se que um ataque de modificação de mensagens seria capaz de comprometer o objetivo de uma missão, ou até mesmo derrubar a aeronave por meio de falsos comandos.

g) *Spoofing*

O ataque de *spoofing* é uma maneira de uma entidade (podendo ser um programa computacional ou uma pessoa) falsificar a sua identidade, ou seja, ela se mascara e se faz passar por outra. Existem vários tipos de *spoofing*, porém, o que é mais comumente aplicado em VANTs é o *spoofing* de GPS (o mesmo tipo de ataque aplicado no VANT americano em terras iranianas). O objetivo do *spoofing* é fornecer um falso sinal ao receptor, enganando-o (Wen et al., 2005). Sendo assim, o mesmo produzirá soluções de posicionamento de maneira errônea, deixando brechas de segurança em seu sistema.

Um aparelho chamado de *spoofer* foi criado com o intuito de se realizar testes em um VANT real, verificando assim se este seria vulnerável a um ataque de *spoofing* (Bhatti et al., 2012). Para a construção do *spoofer* não foram necessários grandes quantidades de materiais, porém, os autores do trabalho estimam que pouco mais de 100 pesquisadores ao redor do mundo poderiam construir um aparelho similar com aproximadamente um ano de esforço. O VANT testado no trabalho é apresentado na Figura 2.7, o qual possui grande comercialização e utilização. O resultado obtido pelos pesquisadores, evidencia que o modelo de receptor de GPS utilizado nesta aeronave (que é o mesmo de muitas outras) é vulnerável ao ataque de *spoofing*.

Ainda não foi encontrada uma solução simples, rápida e de baixo custo que trate este tipo de problema (Bhatti et al., 2012). Nota-se então que o ataque de *spoofing* é uma grande ameaça às aeronaves não tripuladas, o qual gera grande preocupação e a necessidade da intensificação das pesquisas referentes à essa área.

Figura 2.7: VANT utilizado para testes de segurança com o *spoofer* criado

h) *Códigos maliciosos (Malwares)*

Os códigos maliciosos são *softwares* de propósito específico desenvolvidos com o intuito de realizar atividades que causem danos, ou outros tipos de ações maliciosas que possam de alguma maneira, prejudicar quem faz uso do dispositivo infectado (cert.br, 2014).

A partir do momento que um computador passe a estar infectado por um código malicioso, o invasor pode ter acesso a dados sigilosos, enviá-los a outras pessoas, realizar tarefas ilícitas em nome de um usuário da máquina infectada, capturar senhas, dentre outras.

Os principais motivos que levam um invasor a criar e disseminar códigos maliciosos são: obter vantagem financeira por meio de dados obtidos do usuário infectado; se autopromover, muitas vezes em redes sociais; praticar chantagem contra o usuário do computador infectado; vandalismo; ou simplesmente pelo fato de sentir prazer ao burlar regras.

Alguns dos principais tipos de códigos maliciosos existentes são:

- **Vírus**: O vírus possui a característica de se propagar automaticamente e incorporar-se a outros *softwares* e arquivos existentes, de maneira semelhante à forma como os vírus biológicos se reproduzem (Tanenbaum, 2009). Para que o vírus se torne ativo, é necessário que o arquivo ou o *software* em que ele está inserido, seja executado. Os vírus podem ser adquiridos por meio de mídias removíveis, e-mail, comunicadores instantâneos, páginas HTML infectadas, dentre outros. Os vírus podem ainda ser classificados em vários tipos, como por exemplo: vírus de programas executáveis, vírus residentes na memória (RAM), vírus de setor de inicialização (discos rígidos), vírus de *drivers* de dispositivos (presente nos *drivers* do sistema), vírus de macro (residindo em macros tais como as do *Microsoft Office*), vírus de código-fonte (presente em linhas de código de um *software*).

- ***Worm***: Os *worms* são capazes de se propagar automaticamente pela rede criando cópias de si, porém, contrariamente aos vírus, eles não se incorporam a outros programas ou arquivos para se propagar, tal processo é realizado por meio da execução direta ou a exploração de vulnerabilidades existentes em computadores. Além disso, os *worms* são responsáveis por utilizar muitos recursos, afetando o desempenho da rede.

- ***Bot e botnet***: Os *bots* são códigos maliciosos que se propagam de maneira semelhante aos *worms*, ou seja, criando cópias de si automaticamente e explorando vulnerabilidades existentes. No entanto, a finalidade de um *bot* é se comunicar

remotamente com o invasor, que por sua vez pode enviar comandos ao *bot* para executar vários tipos de ações maliciosas, tais como, roubar dados, instalar outros códigos maliciosos, dentre outros. Um computador contaminado com um *bot* é chamado de zumbi, pois pode ser controlado remotamente por um invasor. A partir do momento que um invasor tem acesso a uma rede de computadores zumbi, esta é chamada de *botnet*.

- ***Spyware***: Tal código malicioso tem por finalidade coletar dados sobre o usuário e enviar para uma entidade externa, sem o seu consentimento ou ciência. Muitos *spywares* existentes são desenvolvidos por empresas, que procuram conhecer indevidamente os diversos hábitos de usuários e traçar um perfil de compra, podendo então tirar proveito disso e oferecer produtos específicos. No entanto, *spywares* também são desenvolvidos e utilizados por invasores que almejam coletar dados sigilosos dos usuários, tais como senhas bancárias. Um exemplo é o *spyware* chamado *keylogger*, que captura todas as teclas digitadas pelos usuários do computador infectado e enviam ao invasor. Eventualmente, senhas ou outros dados sigilosos podem ser coletados.

- ***Backdoor***: O código malicioso conhecido como *backdoor* é uma ferramenta complementar para aqueles invasores que desejam manter um canal de acesso remoto à uma máquina, após a realização de um ataque. A inserção de um *backdoor* é comumente realizada por meio de outros códigos maliciosos que exploram vulnerabilidades existentes no sistema. Estas permitem a inclusão de novos serviços ou a modificação de serviços existentes, que por sua vez fornecem o acesso remoto ao invasor.

- **Cavalo de Troia (*Trojan*)**: O cavalo de Troia é assim denominado em alusão à guerra de Troia, na qual um cavalo de madeira com seu interior repleto de soldados foi utilizado para enganar os inimigos e dessa maneira vencê-los. No ambiente computacional não é diferente, esse código malicioso aparentemente executa as funções para qual foi projetado, porém, além delas, outras funções maliciosas estão inclusas, ou seja, a vítima é enganada, assim como na guerra de Troia. Após a máquina da vítima ter sido infectada pelo código malicioso, este fica alojado de maneira oculta, permitindo que o invasor envie instruções para posteriormente executá-las. Dentre as instruções que um cavalo de Troia executa estão: instalar vários outros códigos maliciosos; destruir os dados da máquina afetada; formatar o disco rígido; instalar *softwares* de *spyware* para coletar dados da vítima; dentre outras.

- ***Rootkit***: Um *rootkit* é um conjunto de ferramentas computacionais utilizadas para ocultar programas infectados, dificultando ou evitando que outros tipos de códigos maliciosos, tais como os *backdoors*, sejam detectados por programas de segurança, como, anti-vírus e *firewalls*. Além disso, os *rootkits* podem capturar informações da rede em que a máquina afetada está situada, instalar outros códigos maliciosos, remover rastros em arquivos de logs, dentre outros (cert.br, 2014).

i) *Engenharia social*

A engenharia social pode ser definida como um grupo de atividades que invasores utilizam com o intuito de adquirir informações confidenciais, sejam elas de pessoas, empresas ou outras organizações.

Dentre as ferramentas que podem ser utilizadas por um invasor, as mais comuns para a realização de um ataque de engenharia social são aquelas relacionadas à interação pessoal, tais como a persuasão, a manipulação e a influência de pessoas. Além disso, instrumentos tecnológicos podem ser empregados em um eventual ataque.

Os invasores que praticam ataques de engenharia social muitas vezes se passam por outras pessoas, por exemplo, fingem ser técnicos de uma determinada empresa e que existe a necessidade de se realizar algum tipo de reparo na rede. No entanto, ao invés de executarem o reparo, os invasores obtêm informações privilegiadas em que posteriormente podem utilizar para benefício próprio.

Diante de tais características, nota-se que o trabalho de um invasor se torna menos árduo ao aplicar as técnicas de engenharia social, pois poupa-se o tempo e o trabalho em que ele teria para ultrapassar mecanismos de segurança em um ambiente virtual, tais como, quebrar chaves criptográficas, desbloquear *firewalls*, descobrir senhas e outros.

Essa técnica tem por finalidade tirar proveito de falhas existentes nas pessoas que não estão aptamente treinadas e adaptadas ao convívio com a prática da engenharia social, de maneira oposta a realização de ataques aos sistemas computacionais por meio de ferramentas virtuais. Sendo assim, a engenharia social se apresenta como uma abordagem de grande impacto em relação à segurança de organizações e sistemas computacionais de um modo geral.

j) Negação de serviço *(DoS - Denial of Service)*

O ataque de negação de serviço possui como ideia principal congestionar o canal de comunicação até que o sistema, o serviço ou até mesmo a rede se torne indisponível para os usuários legítimos (Barbeau, 2005). Sendo assim, os ataques de negação de serviço não

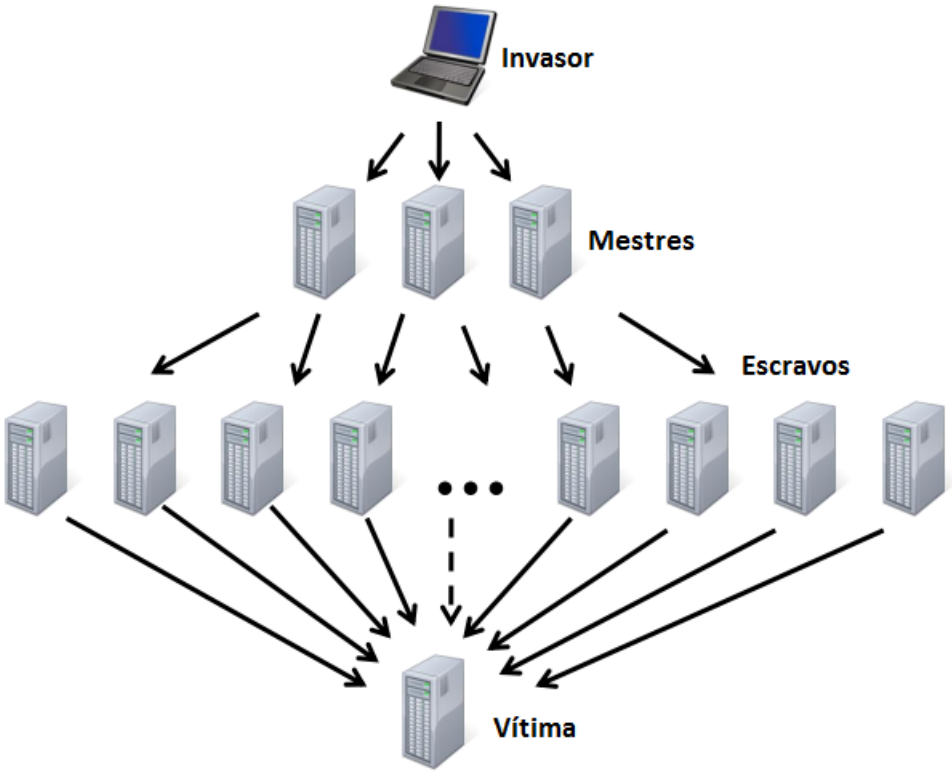

Figura 2.8: Ataque de negação de serviço distribuído *(DDoS)*

são tratados como uma invasão de sistema, pois esses visam sobrecarregar o seu alvo com um grande número de requisições.

Durante a execução do ataque, o alvo estará extremamente ocupado em responder às falsas requisições que estão surgindo a todo instante, que no momento em que chegarem requisições reais, ele não terá condições de respondê-las.

Uma evolução dos ataques de negação de serviço são os realizados de forma distribuída e são conhecidos como DDoS (*Distributed Denial of Service*). Nesse caso, muitas máquinas são infectadas e se tornam escravas. Dessa maneira, a máquina mestre lança o ataque de maneira sincronizada para um ou mais alvos, aumentando a dimensão de muitos para um ao ataque de negação de serviço (Rocha, 1990). Um exemplo de ataque de negação de serviço distribuído é apresentado na Figura 2.8.

Os ataques de negação de serviço ficaram amplamente conhecidos no ano de 2000, quando *websites* de grande porte como *eBay*, *CNN*, *Yahoo*, dentre outros foram atacados, e ficaram indisponíveis por um período de tempo (Kurose et al., 2010).

Existem diferentes tipos de ataques de negação de serviço, dentre eles os principais são: diminuição de largura de banda e esgotamento de recursos. O primeiro pode ainda

ser subdividido em ataques de inundação e amplificação e o último se refere a ataques que fazem o mal uso de protocolos de comunicação, enviando pacotes malformados (Rocha, 1990).

O ataque por inundação é caracterizado por enviar um grande número de pacotes ao sistema alvo, congestionando o seu canal de comunicação. Os reflexos disso no sistema alvo são: lentidão, indisponibilidade e sobrecarga de banda.

Já o ataque por amplificação é designado por enviar várias requisições que necessitam de resposta, para um grande número de máquinas ou para um endereço de *broadcast*. No cabeçalho das requisições enviadas, o *ip* de origem é alterado para o endereço da vítima, fazendo com que as respostas cheguem todas de uma única vez, ocasionando a indisponibilidade da vítima, lentidão e sobrecarga de banda.

No ataque por exploração de protocolos, falhas ou características específicas presentes nos protocolos utilizados na máquina vítima são exploradas. Um dos principais ataques que implementam tal ideia é o *SYN flooding*, apresentado anteriormente. Este ataque explora a característica de estabelecimento de conexão do protocolo TCP.

Os ataques de negação de serviço são amplamente utilizados e de baixa complexidade de implementação. Já para se proteger desses ataques não é uma tarefa simples.

k) Transbordamento de buffer *(Buffer overflow)*

O ataque de transbordamento de *buffer*, ou comumente chamado de *buffer overflow*, consiste em extravasar a quantidade de memória que está alocada para o programa em execução, fazendo com que a memória adjacente seja sobrescrita, podendo causar enormes danos em relação à segurança do sistema, da máquina e consequentemente do usuário.

O *buffer overflow* possui grande incidência sobre os sistemas que são escritos em linguagem C ou sua derivada C++. Tais linguagens são amplamente utilizadas por possuírem tempo de compilação e execução eficientes (Tanenbaum, 2009). No entanto, os compiladores da linguagem C não fazem verificação de limites de vetores, se tornando uma característica extremamente favorável para o ataque. Alguns exemplos disso são as funções padrões da linguagem C: *gets*; e *strcpy*; a primeira é responsável por obter uma *string* de um dispositivo de entrada, tal como o teclado e a segunda função realiza a cópia de *strings*. No entanto, nenhuma avaliação é realizada em relação ao tamanho máximo das *strings*.

Como os VANTs são sistemas embarcados críticos, seus recursos são limitados e devem ser utilizados de maneira extremamente consciente. Sendo assim, o tempo de resposta dos subsistemas que compõem o VANT é um fator de grande importância, pois quanto

maior a velocidade de execução de uma determinada função, mais rapidamente os recursos serão liberados para a execução de outra função. Isso faz com que a escolha da linguagem a ser utilizada na construção dos sistemas de um VANT se torne um ponto primordial. Diante desses aspectos, nota-se que a linguagem C se faz presente constantemente na implementação de sistemas de VANTs, tornando-os predispostos a ataques de *buffer overflow*.

Dentre os impactos que um ataque de *buffer overflow* pode causar estão: erros de acesso à memória, abertura de lacunas no sistema de segurança e até a interrupção do funcionamento do sistema por completo (Cowan et al., 2000). No entanto, a preocupação com ataques de *buffer overflow* só se manifesta geralmente a partir do momento em que um ataque bem sucedido já aconteceu (Tanenbaum, 2009).

2.1.4 Análise de ameaças mais comuns em uma arquitetura de comunicação de um VANT

A Figura 2.9 apresenta uma arquitetura de comunicação que inclui todos os canais relevantes em relação à segurança (Javaid et al., 2012). Pode-se notar que os diferentes componentes existentes nesse modelo dependem de comunicação sem fio.

À primeira vista, os canais de comunicação parecem ser similares, porém, em relação às questões de segurança, cada um possui as suas particularidades. Os ataques de confidencialidade, integridade e disponibilidade são citados com base nos canais de comunicação da Figura 2.9. Eles foram agrupados desta maneira, pois foi realizada uma análise mais geral das ameaças e não de cada canal de comunicação em específico (Javaid et al., 2012).

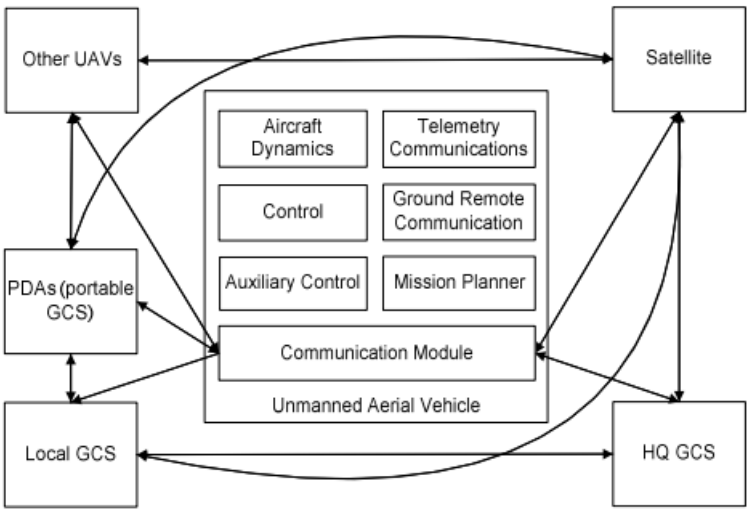

Figura 2.9: Arquitetura do modelo de comunicação para análise de ameaças (Javaid et al., 2012)

a) Ataques de confidencialidade

Estes tipos de ataque são relacionados ao comprometimento de informações de um sistema, interceptando-as e manipulando-as conforme o objetivo do invasor. Essas ações são realizadas por meio de acesso não autorizado às informações. Os quatro componentes de um modelo de VANT que são mais vulneráveis a tais tipos de ataque são o próprio VANT, todos os tipos de estação base, os canais de comunicação e ainda os seres humanos envolvidos com o projeto (Javaid et al., 2012).

Os autores ainda afirmam que as ameaças que são relacionadas aos VANTs em si geralmente são baseadas em ataques de *crackers*. Já as ameaças relacionadas às estações base são originadas em sua maioria por *crackers* por meio de *software*, tais como vírus, *keyloggers*, *trojans* e outros. Um outro ponto importante é que as ameaças de *software*, quando realmente se tornam falhas, podem afetar outras partes de um VANT.

Ao se tratar das ameaças referentes aos canais de comunicação, nota-se que os ataques mais comuns para esta classe são escutas do canal, *spoofing* de identidade, ataques de *cross-layer* (Wang et al., 2010) e ainda ataques de multi-protocolo (Alves-Foss, 1998). Existindo ainda o fator humano, são encontradas ameaças como: engenharia social, falsas competições *online*, exploração comportamental, dentre outras.

Todos os ataques citados não são aplicáveis para todos os canais de comunicação, ou seja, determinado ataque pode ser uma ameaça para um certo canal não sendo para outro, como por exemplo, a engenharia social não pode ser aplicada à um canal de comunicação.

b) Ataques de integridade

Os ataques de integridade são referentes à modificação de uma informação existente ou à criação de novas informações (Javaid et al., 2012). A primeira tem por objetivo obter dados que estão sendo trafegados em uma rede ou sendo armazenados e alterar o seu conteúdo, sendo que alguns fenômenos cotidianos podem causar a perda de integridade, tais como: a troca de polos magnéticos, relâmpagos, entre outros. Estes fenômenos são raros e os protocolos de comunicação já possuem cuidados especiais em uma eventual ocorrência.

Pensando ainda em modificação da informação é possível dividir as ameaças relacionadas aos VANTs em três categorias: *jamming*, comprometimento da integridade do sinal e captura do sinal. Entre elas a última é a mais difícil de se prevenir, pois é necessário muito conhecimento técnico em transmissões de sinal, frequência, faixa e outros (Javaid et al., 2012).

A criação de novas informações envolve a utilização de códigos maliciosos e subrotinas do sistema. As subrotinas procuram por vulnerabilidades que possam existir no sistema, e quando encontradas o sistema estará seriamente exposto.

c) Ataques de disponibilidade

Em parte, os ataques referentes à disponibilidade do sistema são iguais aos de integridade, são eles o *jamming*, falsificação de sinal e negação de serviço (Barbeau, 2005). Como já discutido, para a prevenção desses ataques é necessário muito conhecimento técnico, tornando-os, desta maneira, a maior ameaça à disponibilidade de um sistema. Falsos sinais, assim como aqueles aplicados com a técnica de *spoofing*, podem fazer um VANT pousar ou até mesmo atacar outros aviões, pessoas, bases do governo, entre outros.

2.1.5 Uma análise de probabilidade e impacto de ameaças

Uma análise de probabilidade, impacto e risco de ameaças foi realizada juntamente com suas respectivas classificações (Javaid et al., 2012). As ameaças foram classificadas por um *ranking* de risco podendo ir de 1 a 9, sendo 1 para menor risco e 9 para maior risco. Os autores afirmam que assim é possível analisar de uma maneira global o risco de cada ameaça realmente acontecer, quais os impactos causados e também qual o nível de dano causado por elas.

Na Tabela 2.2 é apresentada a maneira como os autores analisam o risco de cada ameaça. O parâmetro *Likelihood* é a possibilidade de ataques acontecerem, podendo ser

Tabela 2.2: Análise de riscos e ameaças (Javaid et al., 2012)

		Rationale		
Criteria	**Cases**	**Difficulty**	**Motivation**	**Ranks**
Likelihood	Unlikely	Strong	Low	1
	Possible	Solvable	Reasonable	2
	Likely	None	High	3
		User	***System***	
Impact	Low	Annoyance	Very Limited Outages	1
	Medium	Loss of Service (LoS)	Limited Outages	2
	High	Long time LoS	Long time Outages	3
Risk	Minor	No need for countermeasures		1,2
	Major	Threat need to be handled		3,4
	Critical	High priority		6,9

improvável, possível ou provável. Já o *Impact* é relacionado ao nível de dano ao sistema após um determinado ataque, podendo ser baixo, médio ou alto. Já o parâmetro *Risk* é um cálculo realizado por meio da multiplicação entre *Likelihood* e *Impact*, assim, risco = probabilidade * impacto.

A análise de riscos foi então realizada com base na arquitetura apresentada na Figura 2.9. O comportamento da rede em relação à probabilidade dos ataques ocorrerem e a gravidade com que o sistema pode ser afetado foram verificados.

Os resultados dessa pesquisa são apresentados na Tabela 2.3. Nota-se que os ataques com maior risco (igual a 9) são: *spoofing*, modificação de mensagens e negação de serviço. Integridade do sinal e escutas também são outros tipos de ataque que possuem alto risco (igual a 6). Deve ser salientado que em alguns tipos de ataques, mecanismos de segurança foram utilizados, diminuindo a probabilidade de ocorrência, conforme a Tabela 2.3.

Para valores menores que 3, nenhuma medida precisa ser tomada, de 3 a 4 as ameaças devem ser analisadas seriamente. Quando notam-se ameaças com números entre 6 e 9, estas devem ser tratadas com alta prioridade e são dadas como críticas.

Verifica-se que esse é um trabalho que possui conclusões subjetivas, pois não foram comprovadas estatisticamente ou matematicamente. Os autores afirmam então que as medidas utilizadas podem variar de acordo com a análise realizada.

De qualquer maneira, observa-se que esse trabalho é bastante abrangente, abordando vários tipos de ameaças e ataques, que evidenciam que um sistema embarcado crítico, como é um VANT, deve possuir mecanismos de segurança altamente confiáveis para realizar suas missões.

Tabela 2.3: Resultados obtidos a partir da análise de riscos (Javaid et al., 2012)

Threat	Algorithm(s)	Likelihood	Impact	Risk
Jamming		3	1	3
Scrambling/Distortion		2	1	2
Eavesdropping		3	2	6
Cross Layer Attacks		2	1	2
Multi-Protocol Attack		2	1	2
Social Engineering		2	2	4
Spoofing	Device List	3	3	9
	X.509 device Auth.	2	3	6
Command and Control Message Modification	No MAC	3	3	9
	SHA-1 MAC	2	3	6
	AES MAC	1	3	3
Data Traffic Modification	Without AES	3	1	3
	With AES	1	1	1
DoS on UAV/GCS	EAP/SHA-1/AES/MAC	3	3	9
Signal Integrity		3	2	6
Malicious Code, Subroutine Exploit		1	3	3
Virus, Malware, Trojans and Keyloggers		3	2	6

2.1.6 Análise de segurança de um sistema de comunicação de rádio de um VANT

Um trabalho semelhante ao apresentado na Subseção 2.1.5, porém menos abrangente, apresenta uma análise de segurança do sistema de rádio de um VANT (Rudinskas et al., 2009).

A referida análise reforça as ameaças que um VANT pode enfrentar. No entanto, várias maneiras de se mitigar os ataques aos canais de comunicação de rádio são discutidos.

- **Confiabilidade do canal de rádio:** A qualidade do sinal de rádio é um fator importante em veículos aéreos por vários fatores, tais como, condições de tempo, frequência de banda, entre outros. Os autores então recomendam utilizar transmissores duplicados com antenas perpendiculares, melhorando assim a qualidade do sinal.

- **Integridade:** Como já discutido, os canais de comunicação devem ser protegidos de alteração e criação de dados não autorizados. Sendo assim, recomenda-se que seja efetuada uma conferência entre os dados enviados e os recebidos.

- **Autenticidade:** Os VANTs devem também ser capazes de receber dados apenas de estações base e VANTs autorizados. Portanto, a aeronave deve checar a identidade das estações e de outros VANTs.

- **Confidencialidade:** Os dados trafegados entre um VANT e uma estação base devem ser confidenciais. O autor cita várias maneiras de se mitigar este problema, utilizando-se a criptografia como base.

- **Mitigação de *Jamming*:** Uma solução para este problema é a alteração das taxas de transmissão para conter a interferência do ataque (Li et al., 2007).

Essas são algumas maneiras de se prevenir eventuais ataques aos VANTs (Rudinskas et al., 2009). No entanto, outras abordagens são encontradas, dentre elas, uma se destaca na verificação do nível de segurança de um determinado sistema, tal técnica é conhecida como teste de invasão.

2.2 Teste de Invasão

Os testes de invasão, também conhecidos como *Penetration testing* ou *PenTest*, são utilizados para simular métodos que um invasor possa fazer uso em um eventual ataque com o intuito de ganhar acesso aos recursos de um determinado sistema (O'Gorman et al., 2011). Testes de invasão ainda são definidos como um processo a ser seguido para conduzir uma avaliação de segurança ou auditoria (Ali e Heriyanto, 2011).

Tal metodologia possui um conjunto de regras, práticas, procedimentos e métodos que devem ser seguidos durante todo o processo. Sendo assim, esse processo serve como um guia de conceitos práticos que devem ser aplicados cuidadosamente, garantindo que informações sigilosas não sejam extraviadas.

Os testes de invasão podem ser aplicados de maneira independente ou serem parte de processos de gerenciamento de riscos de segurança, sendo assim então introduzidos no escopo de um processo de desenvolvimento de *software*. É importante ressaltar que a segurança de um sistema não se limita apenas aos aspectos relacionados ao ambiente de Tecnologia da Informação (como linguagens utilizadas, bancos de dados, dentre outros), mas também às boas práticas de segurança. Dentre estas estão: implementação de recursos de segurança apropriados, modelagem de ameaças, revisões de código, realização de análise de risco e medição de segurança operacional.

Considerados como a última etapa de verificação da segurança de um sistema, os testes de invasão são classificados como a forma mais agressiva de avaliação de segurança, os quais devem ser conduzidos por profissionais qualificados, podendo estes possuírem algum ou nenhum conhecimento prévio sobre o sistema que será analisado (Ali e Heriyanto, 2011).

O processo de testes de invasão pode ser utilizado para avaliar diversos elementos da infraestrutura, tais como: o *software* em si, dispositivos que fazem parte da rede,

sistemas operacionais, segurança física, meios de comunicação e até psicologia humana (Ali e Heriyanto, 2011).

O resultado dos testes de invasão geralmente se resume a um relatório que contém todas as fragilidades encontradas no sistema, as medidas que podem ser tomadas para extinguí-las e recomendações para aumentar o nível de segurança dentro da organização ou de um sistema.

Existem diferentes tipos de testes de invasão, dentre eles, os mais conhecidos e amplamente utilizados na indústria são os de caixa preta e caixa branca. Ambos são discutidos nas subseções seguintes juntamente com o teste de caixa cinza.

2.2.1 Teste de caixa preta

A abordagem de teste de invasão de caixa preta pode ser classificada como um tipo de teste externo, pois o auditor (conhecido como *black-hat*) realiza os testes a partir de um computador remoto, acessando a infraestrutura de rede do alvo sem possuir qualquer conhecimento das tecnologias e ferramentas utilizadas dentro da organização. Essa técnica é a que mais se aproxima de um *cracker* real que não detém nenhuma informação do alvo que deseja atacar. Como o teste de caixa preta faz uso de técnicas *cracker* reais, podem ser descobertas tanto vulnerabilidades conhecidas como desconhecidas.

É imprescindível para o auditor que realiza os testes entender e classificar as vulnerabilidades encontradas de acordo com o nível de risco que cada uma delas proporciona, podendo ser: baixo, médio ou alto (Ali e Heriyanto, 2011). Geralmente os níveis de cada vulnerabilidade são medidos de acordo com a ameaça que é imposta e também a perda financeira que pode ocorrer caso um ataque aconteça.

Após a finalização do teste, um relatório deve ser gerado com todas as informações pertinentes para que a organização possa tomar as providências necessárias para minimizar ou até mesmo esgotar as chances de um futuro ataque.

2.2.2 Teste de caixa branca

Diferentemente do teste de caixa preta, a abordagem do teste de caixa branca pode ser classificada como um tipo de teste interno, pois o auditor (conhecido como *white-hat*) realiza os testes estando ciente de todas as ferramentas e tecnologias utilizadas dentro da organização. Partindo deste ponto, o auditor não enfrenta muitas dificuldades para avaliar quais são as lacunas de segurança existentes na organização. O teste de caixa branca também é conhecido como o de esforço mínimo possível.

Esse tipo de teste agrega maior valor à organização do que o de caixa preta, pois é possível eliminar grande parte dos problemas de segurança interna, dificultando muito o trabalho de um adversário externo que tente obter qualquer tipo de informação sigilosa (Ali e Heriyanto, 2011).

O teste de caixa branca necessita de menor quantidade de tempo e custo para encontrar as vulnerabilidades e propor uma solução em relação ao teste de caixa preta (Ali e Heriyanto, 2011).

2.2.3 Teste de caixa cinza

O teste de caixa cinza nada mais é do que uma combinação dos testes citados previamente, fornecendo um interessante ponto de vista da segurança interna e externa da organização. O auditor que aplica tal abordagem é conhecido como *grey-hat*.

Essa abordagem possui como benefício o conjunto de vantagens proposto pelas duas abordagens anteriores, porém, para a aplicação da mesma, o auditor deve possuir um conhecimento limitado do sistema interno da organização, para que assim ele possa escolher a melhor maneira de avaliar a segurança global.

Analisando o cenário de segurança externa, o teste de caixa cinza é bem parecido com o de caixa preta, porém ele proporciona melhores escolhas de teste, pois o auditor já está ciente das ferramentas e tecnologias que a organização utiliza.

2.2.4 As etapas da aplicação de Teste de Invasão

Como já mencionado, os testes de invasão são compostos por um conjunto de regras, práticas, procedimentos e métodos que devem ser seguidos para a obtenção da qualidade na utilização dessa metodologia. Os padrões de execução dos testes de invasão são divididos em sete diferentes etapas, sendo elas: planejamento do teste, aquisição de informação, modelagem de ameaças, análise de vulnerabilidades, exploração, pós-exploração e relatório (Weidman, 2014). Todas etapas são descritas a seguir.

1. **Planejamento do teste**: Antes do início dos testes são realizadas interações entre o auditor e cliente, garantindo dessa maneira que ambos compartilhem a ideia e analisem a viabilidade da execução dos testes de invasão, evitando que surpresas desagradáveis aconteçam para ambos. É nessa fase que são discutidos o escopo dos testes, os horários em que podem ser realizados, informações de contato, dentre outros.

2. **Aquisição de informação**: Tal etapa consiste na aquisição de informações que possam ser sensíveis na condução dos testes de invasão. Nesse momento, ferramentas já começam a ser utilizadas, como escaneadores de portas, os quais tornam possível descobrir quais serviços estão sendo utilizados e executados.

3. **Modelagem de ameaças**: Nessa etapa são utilizadas as informações adquiridas na etapa anterior. Nesse momento, é interessante que o auditor pense de maneira semelhante a um *cracker*, desenvolvendo planos para realizar o ataque (nesse caso o teste de invasão) de acordo com as informações obtidas.

4. **Análise de vulnerabilidades**: A etapa de análise de vulnerabilidades, tem como objetivo iniciar a descoberta ativa de quais vulnerabilidades o sistema alvo possui, podendo determinar, então, se as estratégias de exploração abordadas até o momento serão eficazes ou não. Frequentemente, essa etapa conta com a execução de *softwares* específicos de descoberta de vulnerabilidades, os quais utilizam um banco de dados de vulnerabilidades e realizam vários tipos de checagens no sistema alvo. Além disso, é de grande importância que uma análise de vulnerabilidades manual (sem a utilização de *softwares* de análise de vulnerabilidades) seja realizada pelo auditor, aumentando assim a confiabilidade nessa etapa.

5. **Exploração**: A etapa de exploração é a mais esperada pelos auditores, pois é nesse momento que as vulnerabilidades encontradas na etapa anterior são exploradas. Isso significa que todo o planejamento realizado até então é colocado à prova.

6. **Pós-exploração**: A etapa de pós-exploração é o momento de analisar as informações acerca dos sistemas explorados, observando se as vulnerabilidades encontradas e exploradas são relevantes dentro do cenário da organização ou não, ou seja, o que a exploração realizada pelo auditor significa para o cliente que contratou o teste? Essa questão resume qual é o objetivo da fase de pós-exploração. Por exemplo, se foram encontradas e exploradas vulnerabilidades em um sistema obsoleto, ou dados expostos não são de interesse de um possível invasor, isso não seria um problema para a organização. Já se um sistema em fase de desenvolvimento for exposto, isso se tornaria um ponto muito importante a ser corrigido pela organização.

7. **Relatório**: A última etapa de um teste de invasão é a geração do relatório. Este deve mencionar o que está sendo feito corretamente e o que deve ser mudado em relação à segurança da organização, quais os meios utilizados na exploração, o que foi encontrado pelo auditor, como resolver os problemas, dentre outros. Elaborar um

relatório de um teste de invasão não é uma tarefa simples, pois seu conteúdo deve ser claramente entendido por funcionários de tecnologia da informação responsáveis pelas correções e também pela alta gerência que as autorizam. O relatório ainda deve conter um sumário e um relatório técnico para melhor compreensão dos envolvidos.

3

Segurança em VANTs: Trabalhos relacionados

Este capítulo aborda o estado da arte dos trabalhos desenvolvidos até o presente momento, que trazem modelos, técnicas e aplicações de segurança, com o intuito de maximizar o nível de segurança encontrado atualmente nos VANTs. Esta área se tornou uma preocupação em nível mundial, devido à ocorrência de ataques lançados contra VANTs militares nos últimos anos.

No entanto, a segurança dos VANTs não é um assunto de interesse apenas para militares ou governos, pois essas aeronaves estão sendo amplamente utilizadas também em aplicações civis. Dessa maneira, existem muitos VANTs em operação que podem ser utilizados de maneira temerária por pessoas mal intencionadas.

Mesmo tendo se tornado uma área de grande interesse, a segurança em VANTs ainda é pouco explorada, visto que são encontrados poucos trabalhos que empregam esse tema como objetivo central.

Nas seções seguintes são apresentados alguns trabalhos que apresentam possíveis soluções para problemas de segurança enfrentados pelos VANTs atualmente.

3.1 Uma abordagem para análise de riscos em VANTs

Desenvolvido por pesquisadores da Universidade de *Otto-von-Guericke*, Alemanha, o trabalho de Hartmann e Steup (2013) possui grande relevância em relação à segurança de VANTs. Ele aborda:

i **O atual cenário dos VANTs**;

ii **As maiores preocupações e desafios no âmbito da segurança**;

iii **Um modelo de comunicação entre os componentes da aeronave**;

iv **O desenvolvimento de um modelo de análise de riscos para evitar a ocorrência de ataques cibernéticos**.

Desde o ano de 2008 até 2013, o número de incidentes reportados para as agências de notícias públicas aumentou, assim como o interesse em VANTs militares e civis. Reforçando tal afirmação, tem-se um aumento significativo também nos investimentos realizados em pesquisa e desenvolvimento dos VANTs.

Para se obter conhecimento acerca da segurança de um sistema, é de grande importância saber quais vulnerabilidades ele possui. Este é um requisito primordial para que uma análise de riscos pudesse ser realizada. Outro requisito essencial é a criação de um modelo de comunicação entre os componentes básicos que constituem o VANT e também a estação base. Dessa maneira, cada componente foi discutido no trabalho, juntamente com as principais vulnerabilidades de cada um deles.

Alguns dos ataques recentes que ocorreram com VANTs também foram abordados, evidenciando a importância de se investir na segurança de tais aeronaves.

Um esquema de análise de riscos foi então desenvolvido, o qual fornece informações da susceptibilidade dos componentes a ataques de confidencialidade, integridade e disponibilidade. Quando um ataque é considerado não suscetível, seu valor correspondente é 0, já no caso do ataque ser altamente suscetível, seu valor correspondente é 1.

Dessa forma, esse trabalho discute o quanto um ataque pode afetar a confidencialidade, integridade e disponibilidade do VANT em relação às ameaças as quais está suscetível, tais como: ameaças de ambiente, de canais de comunicação, sensores, dispositivos de armazenamento e mecanismos de gerenciamento de falhas. No entanto, os ataques mencionados especificamente nesse trabalho foram: *jamming* e *spoofing*.

Foi então realizada a análise de riscos de três VANTs, sendo eles o AR.DRONE, o MQ-9-REAPER e o RQ-170 SENTINEL (sendo esta uma análise parcial, pois este modelo foi atacado por tropas iranianas há pouco tempo, portanto pouco material está disponível para estudo). Foram apresentados alguns números referentes à confidencialidade, integridade e disponibilidade de tais aeronaves, porém, chegou-se à conclusão de que a análise de riscos é uma tarefa altamente complexa ao se tratar de VANTs, pois esta é altamente dependente da missão que os VANTs executam.

É importante destacar que o trabalho foi caracterizado como uma tentativa pioneira de padronizar e formalizar a análise de riscos em VANTs, porém ele apresenta um esquema que ainda é insuficiente, pois pouca informação pôde ser obtida dos incidentes conhecidos.

Tal trabalho propõe uma abordagem de análise de riscos inicialmente interessante, pois apresenta os componentes básicos dos VANTs que podem estar sujeitos a ataques, avaliando-os com base nas três características primordiais da segurança: confidencialidade, integridade e disponibilidade. Tal avaliação é realizada de maneira minuciosa, tratando de diferentes aspectos de ambiente, canais de comunicação, sensores, dispositivos de armazenamento e mecanismos de gerenciamento de falha. Dessa forma, tal trabalho desempenha o papel proposto, servindo como um ponto inicial para a formalização de um modelo de análise de riscos para VANTs.

3.2 Análise de vulnerabilidades do piloto automático de VANTs

Desenvolvido por pesquisadores da Universidade de *Purdue*, Estados Unidos, o trabalho de Kim et al. (2012) foca na análise de vulnerabilidades encontradas em sistemas de pilotos automáticos de VANTs e como eles se comportam em um ambiente pós-ataque. Tudo isso é realizado por meio de ferramentas de simulação.

Assume-se no trabalho que é possível corromper dados inseridos no sistema de piloto automático, por meio de métodos tais como o ataque de *buffer overflow* e outros. No entanto não se considera o método de ataque, mas por sua vez os efeitos que ele gera por meio dos dados corrompidos maliciosamente no VANT e os danos causados.

Como a estrutura dos sistemas de piloto automático foram introduzidos pelos aviões tripulados, a segurança em relação à ataques cibernéticos até pouco tempo não era tratada como um aspecto prioritário. Isso faz com que o sistema de piloto automático dos VANTs sejam vulneráveis a diversos tipos de ataques.

Um levantamento dos componentes utilizados em pilotos automáticos de VANTs foi realizado, os quais serviram como base para mapear as possíveis vulnerabilidades encontradas nesses sistemas, sendo elas classificadas em ataques de *hardware*, ataques de ambientes sem fio e *spoofing* de sensores. Esse trabalho identificou os seguintes ataques: *buffer overflow*; *spoofing*; códigos maliciosos; negação de serviço; e *jamming*.

No ambiente de simulação do trabalho, foram utilizadas as seguintes ferramentas:

- ***ScicosLab e Scicos***: Ferramentas similares ao MatLab e Simulink, porém, são opções de código aberto, projetadas para trabalhar com diagramas de blocos e utilizadas na análise e cálculos numéricos da dinâmica da aeronave.

- ***Arkscicos***: Biblioteca criada com o intuito de se integrar com as anteriores, fornecendo assim um ambiente de modelagem e simulação de voos não tripulados.

- ***JSBSim***: Ferramenta de análise numérica que simula a dinâmica de voo de uma aeronave. Em sua biblioteca existem diversos modelos de aviões disponíveis, sendo possível também utilizar modelos customizados.

- ***Flight Gear***: Simulador de voo de código aberto amplamente conhecido, utilizado para visualizar a simulação numérica, uma vez que ele é capaz de receber os dados oriundos do *JSBSim*, e apresentar as ações físicas do avião em um ambiente de simulação com grande precisão.

- ***Purdue HSL Analysis Tool***: É a junção de todas as ferramentas citadas anteriormente, sendo assim possível estudar cada aspecto de um VANT com grande riqueza de detalhes, tal como verificação de sensores, estabilidade do VANT, envio de comandos, dentre outros.

Nesse trabalho foi também desenvolvida uma arquitetura para a segurança no sistema de piloto automático, a qual possui um conceito de supervisor, em que seu papel consiste em detectar possíveis anormalidades ou atividades maliciosas. Isso faz com que o desempenho da aeronave diminua, porém o nível de segurança aumente.

Tal trabalho elenca os componentes básicos presentes no piloto automático dos VANTs. Com base neles, é realizada a identificação das vulnerabilidades e ameaças que podem conduzir a ataques de *hardware*, ataques em ambiente sem fio e de *spoofing* de sensores. Por meio das ferramentas citadas anteriormente foram realizadas análises decorrentes de ataques executados em ambiente simulado, os quais forneceram gráficos que salientam suas consequências.

A identificação de vulnerabilidades e a arquitetura desenvolvida com foco no supervisor de detecção e isolamento de ataques lidam especificamente com as adversidades de segurança presentes no sistema de piloto automático dos VANTs (Kim et al., 2012). Dessa forma, tal trabalho complementa o levantamento de vulnerabilidades e o modelo para testes de invasão descritos no Capítulo 4.

3.3 Análise de riscos de sequestro de VANTs e métodos de detecção

A integração dos VANTs no espaço aéreo dos Estados Unidos da América, já era um assunto em pauta no ano de 2013. Para a efetivação dessa integração, é imprescindível que a identificação e mitigação da ameaças que envolvem VANTs seja realizada. Dessa forma, o trabalho de Faughnan et al. (2013) apresenta um experimento que faz uso de um carro em movimento para simular o voo de um VANT e um alarme que é disparado ao detectar um sequestro. Esse experimento gera resultados que foram analisados estatisticamente. O trabalho foi dividido basicamente em duas etapas, a primeira aborda quais os riscos de um VANT ser sequestrado e a segunda discute como gerenciar os riscos.

Na primeira etapa, quatro questões fundamentais são tomadas como base, são elas:

- **O que pode acontecer de errado?**
- **Qual a probabilidade?**
- **Quais são as consequências?**
- **Em que período de tempo?**

Já na segunda etapa, outras importantes questões são utilizadas para avaliar cada cenário de risco dos VANTs. Algumas delas são:

- **O cenário é controlável ou reversível?**
- **Os efeitos são em cascata?**
- **Existem múltiplos caminhos até a falha?**

Cada questão utilizada em tal etapa, ajudou a detectar qual cenário possuía o maior risco, para que dessa maneira ele pudesse ser enfatizado.

No gerenciamento de riscos, o objetivo é mitigar as consequências de um possível sequestro. De uma maneira geral, os autores afirmam que um sistema de detecção de sequestro é interessante, pois informa o operador que a segurança do VANT está comprometida. No entanto, é discutido o custo benefício de se utilizar um sistema como esse, o qual inclui custos com implementação do sistema, treinamento do operador, falsa identificação de um VANT sequestrado, dentre outras.

Dessa maneira, chega-se à conclusão de que a implementação de tecnologias emergentes em relação à segurança de VANTs deve ser realizada de maneira efetiva e estável para que possam ser difundidas e assim aumentar o nível de segurança dos VANTs, seja no que diz respeito a sequestros ou outros aspectos.

O sistema de detecção de sequestro proposto não foi implementado, no entanto, possui grande relevância, pois pode auxiliar o operador de um VANT a tomar as melhores decisões em um cenário em que a aeronave tenha sido comprometida. Dessa forma, tal sistema tem potencial para ser utilizado juntamente com outras abordagens que possam aumentar o nível de segurança dos VANTs, tal como os testes de invasão. Estes podem ser utilizados para testar a segurança da aeronave antes mesmo dela realizar o seu primeiro voo, ou mesmo após ela já ter iniciado sua atividade em missões. Assim, os sistemas de detecção de sequestro e de testes de invasão podem ser aplicados concomitantemente, aumentando a segurança dos sistemas de VANTs e tornando-os mais confiáveis.

3.4 UAVSim: Uma plataforma de simulação para análise de segurança em redes de VANTs

Os VANTs estão sendo amplamente utilizados nas mais diversas áreas, diminuindo de maneira considerável o risco de se perder vidas humanas em tarefas de alto risco. No entanto, essas aeronaves apresentam ameaças caso a segurança não seja tratada de maneira adequada. Checar o impacto das tentativas de ataque, é uma abordagem interessante. Dessa maneira, o trabalho de Javaid et al. (2013) apresenta uma plataforma de simulação para analisar a segurança de redes de VANTs sobre determinados tipos de ataque.

Um ponto importante desse trabalho é a simulação de uma rede de VANTs, ou seja, não se trata de apenas um VANT, porém de um ambiente que pode conter vários deles. Além disso, alguns dos principais requisitos presentes na plataforma são:

- **Permitir o uso de vários modelos de VANT**;
- **Permitir o teste de medidas de segurança e checar o impacto**;
- **Ter uma interface fácil e intuitiva**;
- **Permitir aos usuários diminuir a velocidade da simulação**;
- **Tratar o VANT como uma rede de componentes**;
- **Ser compatível com modelos de VANT desenvolvidos em outros *softwares* e poder importá-los**.

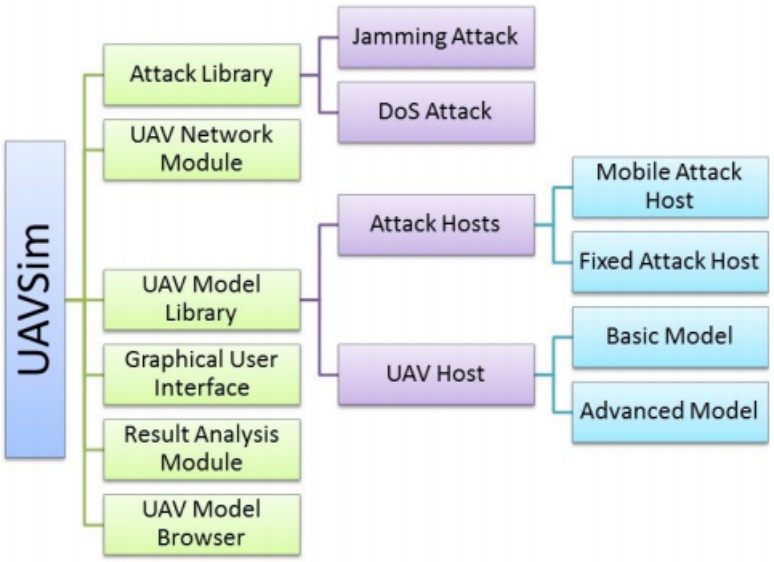

Figura 3.1: Arquitetura da plataforma de simulação UAVSim (Javaid et al., 2013)

O simulador desenvolvido como parte desse trabalho, chamado de UAVSim, é composto por seis módulos principais, são eles:

i **Biblioteca de ataques**;

ii **Módulo de rede de VANT**;

iii **Biblioteca de modelos de VANTs**;

iv **Interface gráfica**;

v **Módulo de análise de resultados**;

vi **Navegador de modelos de VANT**.

A arquitetura do UAVSim e a especialização de cada módulo são apresentadas na Figura 3.1. Nela é possível notar que foram cobertos na plataforma de simulação apenas os ataques de *jamming* e outros de negação de serviço.

A análise dos resultados da simulação é realizada com o auxílio de gráficos com diferentes informações, as quais dependem do tipo de ataque, dos parâmetros utilizados no ataque, dentre outros.

O UAVSim fornece então um ambiente de simulação de uma rede de VANTs, com a opção de escolha do modelo de aeronave, o tipo de ataque a ser aplicado na rede e a geração de resultados que são disponibilizados por meio de gráficos. Por se tratar de um

simulador, o UAVSim se apresenta como uma maneira de se trabalhar com segurança de VANTs sem a necessidade de altos custos com *hardware* e *software* específicos para tal propósito.

O trabalho possui grande contribuição para àqueles que necessitam de uma plataforma de simulação de rede de VANTs, com o intuito de testar, analisar e melhorar o atual nível de segurança encontrado nos componentes que compõem a rede, pois o ambiente necessário já está integrado e fornece vários recursos, tais como: simular vários VANTs em uma única rede, implementação existente de dois tipos de ataques pré-definidos, vários tipos de VANTs disponíveis, módulo de análise de resultados por meio de gráficos, dentre outros.

Um ponto de melhoria no UAVSim seria o aumento do número de ataques cobertos pela ferramenta, tornando o UAVSim uma ferramenta mais robusta.

3.5 Modelo de riscos e ameaças de canais de comunicação e estações base móveis

Desenvolvido por pesquisadores da Universidade George Washington, Estados Unidos, o trabalho de Mansfield et al. (2013) apresenta quais problemas estão sendo enfrentados pelo Departamento de Defesa do país, após a inserção de estações base móveis no campo de batalha. Dessa forma, tal trabalho apresenta uma análise de vulnerabilidades nos canais de comunicação, *tablets*, *smartphones* e aplicativos de *software* utilizados. Consequentemente é criado um modelo de riscos e ameaças que pode auxiliar na criação de uma rede de comunicação segura.

Como os VANTs e as estações base fixas vêm sofrendo diversos tipos de ataques ao longo dos últimos anos, as estações base móveis, da mesma forma, estão altamente vulneráveis, e as consequências de um possível ataque podem ser severas.

O Departamento de Defesa dos Estados Unidos da América realizou um estudo e constatou que 91,1% dos *smartphones* produzidos, utilizam ou o sistema operacional *Android* do *Google* ou o *iOS* da *Apple*. Com o desenvolvimento de um aparelho destinado especificamente para uso militar, várias técnicas de segurança poderiam ser utilizadas. No entanto, o tempo de desenvolvimento seria elevado, juntamente com os custos do aparelho e o treinamento para os militares.

Dessa maneira, o Departamento de Defesa dos Estados Unidos optou por utilizar os aparelhos com tecnologia *Android* e da *Apple* em tarefas de múltiplos propósitos, inclusive em estações base móveis para a comunicação com VANTs. Porém, essa é uma área em que

muitos milhões de dólares estão sendo investidos, desde a comunicação segura entre tais dispositivos, até aplicativos instalados e servidores de *download* de dados pelos soldados.

Constatada a grande importância dos dispositivos móveis em aplicações que demandam alto grau de segurança para uma nação, tais como os sistemas de VANTs, é imprescindível que os canais de comunicação em que os dados trafegam sejam seguros, assim como os aplicativos presentes no dispositivo, e a rede de um modo geral.

Nesse trabalho foi realizado então um levantamento das ameaças existentes em estações base móveis, ou seja, *smartphones* e *tablets* utilizados como estações base. As ameaças foram subdivididas em vulnerabilidades de *software*, *hardware* e de redes de comunicação, como apresentado na Tabela 3.1.

Tabela 3.1: Ameaças presentes em estações base móveis (Mansfield et al., 2013)

Vulnerabilities	**Threat**	**Security Objectives**		
		Confidentiality	*Integrity*	*Availability*
Hardware	*Battery Exhaustion*			X
	Flooding		X	X
	Surveillance	X	X	
	USB	X	X	
Software	*Malware*	X	X	X
	Phising		X	X
	Data Leakage	X		
Communication Network	*Eavesdropping*	X		
	Spoofing	X	X	
	Denial of Service			X
	Jamming			X

São discutidas tanto as vulnerabilidades de *hardware*, quanto de *software* e de redes de comunicação, assim como as técnicas de mitigação de cada uma delas, as quais levam em consideração diferentes aspectos do aparelho, tais como: o sistema operacional, os recursos disponíveis (câmera, GPS, microfone, dentre outros), os serviços e processos que são executados, dentre outros.

Por fim, chega-se à conclusão de que as ameaças de *hardware* são introduzidas principalmente por conexões físicas com dispositivos adulterados. Já as ameaças de *software* são inseridas em grande parte dos casos por meio de *malwares* (códigos maliciosos) existentes na rede ou em outras aplicações de *software*. Os autores afirmam ainda que as aplicações de *software* devem ser testadas e atualizadas regularmente para a garantia da segurança. Nota-se ainda, que a maior fonte de vulnerabilidades é a rede de comunicação, a qual

pode levar a outras ameaças que posteriormente serão classificadas como de *hardware* ou *software*.

Esse trabalho se difere de todos outros já apresentados, pois trata-se da comunicação de VANTs com estações base móveis, que estão sendo empregadas nos campos de batalha dos Estados Unidos da América. A análise realizada mostra vários tipos de vulnerabilidades. Parte delas estão igualmente presentes em estações base fixas, já outros tipos de vulnerabilidades são encontrados exclusivamente ou em estações base móveis ou em estações base fixas.

O modelo de riscos e ameaças em canais de comunicação com estações base móveis proposto pode ser caracterizado como uma análise complementar a todos os trabalhos relacionados à segurança, desenvolvidos com o foco em estações base fixas, pois estas apresentam diferentes características, aspectos e preocupações em relação às estações base móveis. Dessa maneira, o modelo complementa também este trabalho, que aborda a utilização de uma estação base fixa para controlar o voo do VANT.

3.6 Framework universal para testes de invasão em VANTs

Pesquisadores do Instituto Politécnico e Universidade Estadual da Virgínia nos Estados Unidos propuseram um *framework* para testes de invasão em VANTs, o qual tem o objetivo de analisar riscos de segurança de maneira mais rápida e consistente, garantindo então que a nova geração de VANTs seja desenvolvida de modo que a segurança se torne um aspecto de essencial importância. Dessa maneira, tais aeronaves poderiam fornecer maior confiança e efetividade no desempenho das futuras missões (Kobezak et al., 2013).

Atualmente, muitos dos sistemas de VANTs são projetados, testados e implantados rapidamente, sendo que em grande parte das vezes não são realizados testes de segurança adequados. Para contribuir com a redução de tempo no desenvolvimento de tais sistemas, dispositivos vendidos comercialmente são adquiridos e acoplados aos VANTs, no entanto, muitos deles podem introduzir vulnerabilidades no sistema, principalmente se os projetistas não compreendem totalmente a sua funcionalidade e limitações, além de que, os componentes não foram projetados com o foco de sua utilização em sistemas embarcados críticos.

São discutidos módulos existentes em sistemas de VANTs que podem possuir determinados tipos de vulnerabilidade, tais como: dispositivos de rede, canais de comunicação internos, programas de controle, programas lógicos, sensores e canais de comunicação externos.

O *framework* proposto nesse trabalho segue a abordagem tradicional dos testes de invasão, o qual é composto por etapas bem definidas a serem seguidas, como mencionado na Seção 2.2.4. No entanto, todas essas etapas são aplicadas em três estágios de testes, sendo elas: teste dos componentes de maneira independente, testes da rede ou subsistemas e o veículo totalmente integrado.

Outra parte do *framework* se resume em identificar e organizar os componentes eletrônicos presentes no VANT, os quais são relacionados com as possíveis vulnerabilidades as quais estão expostos. Posteriormente, as vulnerabilidades são classificadas com um nível de risco, o qual é juntamente relacionado ao impacto que cada vulnerabilidade pode causar. As vulnerabilidades encontradas nesse trabalho são apresentadas na Tabela 3.2, no entanto, não são discutidos os ataques que podem tirar proveito dessas vulnerabilidades.

Tabela 3.2: Tabela de exemplos de vulnerabilidades. Adaptado de Kobezak et al. (2013)

#	**Vulnerabilidade**	**Nível de risco**
1	Leitura de dados por entidades não autorizadas	3
2	Autenticação pode ser ignorada	1
3	Rede ou canal de comunicação vulnerável ao ataque de *spoofing*	1
4	Dispositivos não autorizados podem transmitir na rede ou canal de comunicação	2
5	Mensagens na rede ou canal de comunicação podem ser inibidas	1
6	Dados de posicionamento podem ser falsificados	1
7	Imagens podem ser falsificadas	2

Por fim, os autores chegam a conclusão de que os testes de invasão deveriam ser uma etapa essencial de sistemas não tripulados (sejam aéreos, terrestres ou outros). No entanto, os sistemas não tripulados atuais são altamente complexos, o que dificulta a tarefa dos auditores que aplicam os testes de invasão, pois eles devem instruir-se com novos conceitos, como por exemplo: novos protocolos, formatos de dados, arquiteturas de sistema, dentre outros. Os autores afirmam ainda que os testes de invasão muitas vezes são considerados improvisação, porém, seguindo-se uma metodologia repetitiva, podem ser alcançados resultados precisos e satisfatórios.

O *framework* desenvolvido por Kobezak et al. (2013) é aquele que mais se aproxima deste trabalho, pois em ambos são realizados o levantamento das vulnerabilidades existentes nos VANTs e a análise da utilização de testes de invasão em sistemas de VANTs. No entanto, este trabalho conta com um levantamento de vulnerabilidades mais completo (com maior número de vulnerabilidades), a criação e desenvolvimento de um modelo para testes de invasão, além da criação de uma ferramenta para auxiliar os pesquisadores da área de segurança de VANTs com testes de invasão em ambiente simulado.

4

Modelo para Teste de Invasão em VANTs

Diante dos trabalhos relacionados, nota-se que a segurança de veículos não tripulados, sejam eles terrestres ou aéreos, é uma área carente e pouco explorada até hoje, visto que trabalhos que propõem soluções para os problemas de segurança enfrentados por veículos não tripulados são poucos.

Como forma de aumentar o nível de segurança dos VANTs, este trabalho propõe o uso dos testes de invasão. Dessa maneira, foi desenvolvido um modelo para esse propósito composto por:

- **Esquema para testes de invasão em VANTs**;
- **Lista de vulnerabilidades comuns em VANTs**;
- **Lista de ameaças que podem comprometer o objetivo de um VANT**.

4.1 Esquema para Teste de Invasão em VANTs

O esquema apresentado na Figura 4.1 expressa de maneira geral como as ameaças e vulnerabilidades existentes podem afetar as missões de um VANT e quais os benefícios de utilizar testes de invasão. Independentemente da fase de projeto da aeronave, os testes de invasão podem ser inseridos, desde o planejamento (por meio de simulações) até a fase final do projeto, quando o VANT já atua em missões reais.

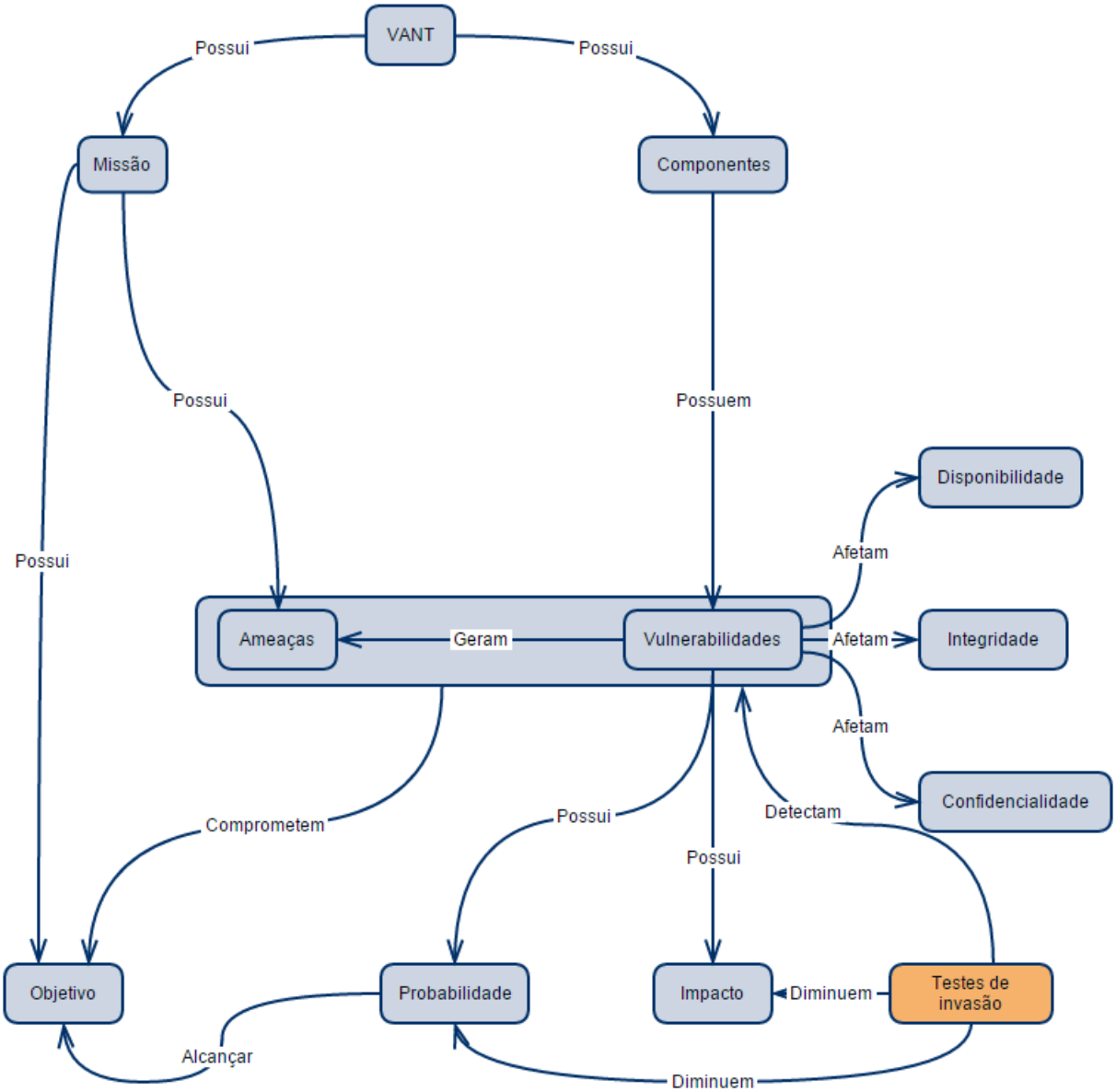

Figura 4.1: Esquema para testes de invasão em VANTs

Como os VANTs são sensíveis às missões que executam, essas se apresentam como um fator de grande importância dentro do esquema desenvolvido. Dessa maneira, um **VANT** pode possuir um ou mais tipos de **missões**, as quais são altamente dependentes do propósito do VANT. Por exemplo, se um VANT foi desenvolvido com o objetivo de trabalhar em missões militares, ele não será uma aeronave passível de ser utilizada em outros tipos de missões civis, pois um VANT que possui equipamento bélico não pode executar qualquer outro tipo de tarefa.

O esquema da Figura 4.1 apresenta também a relação entre **VANT** e **componentes**, isso indica que os **VANTs** são compostos por um grande número de **componentes** como: sistema de telemetria; módulo de comunicação *online*; módulo de comunicação *offline*; diversos tipos de sensores; dentre outros.

Atualmente, a indústria de VANTs cresce, no entanto, muitos dos projetos desenvolvidos não levam a sério as questões relacionadas à segurança, não são realizados esforços para que os **componentes** que compõem os VANTs sejam criados de maneira específica e segura, pois isso demanda tempo e esforço. Sendo assim, muitos dos **componentes** presentes nessas aeronaves já existem e são vendidos comercialmente, fazendo com que eventuais **vulnerabilidades** que existam em um **componente** específico sejam inseridas nos VANTs.

Ao se tratar das **vulnerabilidades** que podem ser encontradas nos **componentes** de um **VANT**, essas podem afetar os pilares da segurança: **confidencialidade**; **integridade**; e **disponibilidade**.

Os ataques de quebra de **confidencialidade** são relacionados ao comprometimento de informações de um sistema, os quais um invasor as intercepta e as manipula conforme seu objetivo. Tais tarefas são realizadas por meio de acesso não autorizado às informações.

Já os ataques de quebra de **integridade** são referentes à modificação de uma informação de maneira indevida ou à criação de novas informações não legítimas. Tais ataques violam a propriedade de manter a informação com todas as suas características originais.

Os ataques de quebra de **disponibilidade** têm o intuito de indisponibilizar o acesso ao sistema alvo. Esses ataques violam a propriedade de garantir que as informações estejam sempre disponíveis para a utilização de usuários legítimos, ou seja, aqueles que contam com a autorização cedida pelo proprietário da informação.

Ao se tratar das **missões** que o **VANT** pode executar, essas geralmente possuem um ou mais **objetivos** bem definidos. No entanto, as **ameaças** existentes podem não ser notadas, podendo ocasionar vários problemas, tais como: impossibilitar tarefas presentes na **missão**, indisponibilizar informações coletadas durante a **missão**, comprometer

completamente a **missão** não alcançando o **objetivo**, ocorrer a perda da aeronave e até mesmo desencadear tragédias decorrentes das **ameaças**.

As **vulnerabilidades** encontradas nos **componentes** de um **VANT** e as **ameaças** que estão envolvidas em uma determinada **missão** formam um conjunto de aspectos que contam com a **probabilidade** de ocorrência, juntamente com o **impacto** que tal ocorrência pode causar. Por exemplo, em uma missão cujo objetivo é gravar imagens em uma fronteira, um componente vulnerável a um ataque de *flooding* pode se tornar uma ameaça ao canal de comunicação, uma vez que o **VANT** pode ficar indisponível. Nesse caso, a **probabilidade** é uma estimativa de ocorrência do ataque de *flooding*, e o **impacto** é uma estimativa do dano que o ataque pode causar.

O conjunto de aspectos formado pelas ameaças à **missão**, juntamente com os pilares da segurança afetados por **vulnerabilidades** encontradas nos **componentes**, é um ponto que deve receber muita atenção, pois ele pode comprometer todo um planejamento e execução de uma determinada **missão**. Por conta disso, os **testes de invasão** se apresentam como uma solução para diminuir a **probabilidade** e o **impacto** de futuros ataques que possam ocorrer com um **VANT**.

Os **testes de invasão** se apresentam como uma forma de prevenção de ataques, os quais podem ser utilizados em várias fases da vida útil de um VANT, ou seja, desde o seu planejamento (fazendo uso de simulações) até o momento que a aeronave já estiver disponível para atuar nas missões (com a utilização de ferramentas específicas). Dessa forma, os **testes de invasão** são capazes de determinar se a aeronave possui vulnerabilidades e como elas podem ser mitigadas.

4.2 Listas de vulnerabilidades e ameaças encontradas em VANTs

Para o desenvolvimento deste trabalho, foram analisadas várias pesquisas que abordam a segurança no ambiente dos VANTs. Dessa maneira, constatou-se que parte delas realizam um levantamento de vulnerabilidades frequentemente encontradas nos VANTs, no entanto, não foram encontrados trabalhos que abordassem o levantamento de vulnerabilidades de maneira completa, ou seja, cada trabalho possui um subconjunto das vulnerabilidades.

Sendo assim, este trabalho apresenta uma ampla listagem das vulnerabilidades e ameaças encontradas no ambiente dos VANTs.

4.2.1 Vulnerabilidades

As pesquisas relacionadas à Tecnologia da Informação avançam, entretanto, juntamente com esse rápido desenvolvimento, as vulnerabilidades dos sistemas também são encontradas em grande escala. Em redes de VANTs não é diferente, as vulnerabilidades estão presentes e devem ser analisadas minuciosamente.

Na Tabela 4.1 são apresentados os principais componentes dos VANTs e as vulnerabilidades as quais eles estão expostos.

Tabela 4.1: Tabela de componentes e vulnerabilidades

Componente geral	**Vulnerabilidade**	**Momento da exploração**
Estação base	Leitura de dados por entidades não autorizadas	*Online/Offline*
Estação base	Não utilização de mecanismos de autenticação obrigatórios	*Online/Offline*
VANT	Leitura de dados por entidades não autorizadas	*Online/Offline*
VANT	Não utilização de mecanismos de autenticação obrigatórios	*Online/Offline*
Canais de comunicação	Leitura de dados por entidades não autorizadas	*Online/Offline*
Canais de comunicação	Não utilização de mecanismos de autenticação obrigatórios	*Online/Offline*
Componente específico		
Receptor GPS	Falsificação de dados de posicionamento	*Online*
Câmera	Falsificação de imagens	*Online/Offline*
Microfone	Leitura de dados por entidades não autorizadas	*Online/Offline*
Bateria	Exaustão da bateria	*Offline*
Portas USB	Contaminação por *software malicioso*	*Offline*
Sensor sonar	Falsificação de dados de posicionamento	*Online*
Sensor de fluxo ótico	Falsificação de dados de posicionamento	*Online*
Sensor de alarme de bateria	Falsificação de dados de bateria	*Online*
Outros sensores	Falsificação de dados de entrada	*Online*

Na primeira coluna da Tabela 4.1 estão listados os componentes, os quais inicialmente são abordados de modo geral: estação base; VANT; canais de comunicação. É importante ressaltar que esses três módulos foram analisados no início da tabela de maneira ampla, no entanto, cada um deles pode ser tratado de maneira específica, fazendo sua subdivisão

e trabalhando com componentes menores. Na sequência são relacionados importantes componentes de maneira mais detalhada, tais como sensores, câmeras, dentre outros.

A segunda coluna apresenta as vulnerabilidades encontradas em cada um dos componentes previamente listados. A última coluna, rotulada como **momento da exploração**, corresponde ao momento em que a vulnerabilidade pode ser explorada, o qual pode ser classificado como:

- ***Online***: O momento em que o VANT está em operação;
- ***Offline***: O momento em que o VANT não está em operação;
- ***Online/Offline***: Ambos os momentos.

4.2.2 Ameaças

Muitas ameaças estão presentes em ambientes compostos por VANTs, estações base e canais de comunicação. Consequentemente, foi realizado o levantamento das principais ameaças que podem impedir um VANT de completar sua missão.

Na Tabela 4.2 são descritas as ameaças envolvidas no ambiente de segurança de VANTs, as quais são efetivadas por meio de ataques que se aproveitam de vulnerabilidades existentes nos componentes, estações base e canais de comunicação que os interligam.

A primeira coluna da tabela apresenta as ameaças que estão presentes no ambiente dos VANTs.

Na coluna **Motivação**, têm-se as possíveis motivações que podem levar um invasor a realizar um certo tipo de ataque contra um VANT, uma estação base ou um canal de comunicação. São listadas as principais motivações, no entanto, podem existir outras justificativas para a realização dos ataques e exploração de vulnerabilidades.

A coluna **Tipo de ataque** se refere ao meio utilizado para que uma determinada ameaça possa se tornar um ataque bem sucedido. Os tipos de ataque são classificados de quatro maneiras: ataques de ***Hardware***; ***Software***; ***Hardware/Software***; e ataque **físico**.

A última coluna, rotulada como **Característica de segurança afetada**, é referente às três principais características da segurança. Dessa maneira, as ameaças podem afetar: a **Confidencialidade**; **Integridade**; **Disponibilidade**; ou a combinação delas.

Com o mapeamento das vulnerabilidades e ameaças que podem impedir um ou mais VANTs de alcançarem seus objetivos, é possível analisar de maneira concreta quais as melhores abordagens a serem seguidas para a mitigação de tais problemas.

Neste trabalho foi adotada a abordagem de testes de invasão, a qual tem por objetivo diminuir a probabilidade e o impacto de um eventual ataque a um ou mais VANTs. Desse modo, haverá um aumento nas chances da missão proposta ser executada sem problemas relacionados à segurança.

Tabela 4.2: Tabela de ameaças comuns em VANTs

Ameaça	**Motivação**	**Tipo de ataque**	**Característica de segurança afetada**
Flooding	Indisponibilizar a comunicação entre as entidades	*Software*	Disponibilidade
Jamming	Romper a comunicação do VANT com a estação base, deixando a aeronave vulnerável e sem controle	*Hardware, Software*	Disponibilidade
Sniffing	Capturar informações para serem utilizadas em um próximo ataque	*Software*	Confidencialidade
Crosslayer	Indisponibilizar a comunicação entre as entidades comunicantes	*Software*	Disponibilidade
Multiprotocolo	Quebrar protocolos de segurança	*Software*	Confidencialidade, Integridade, Disponibilidade
Message modification	Modificar o destino da mensagem ou até mesmo seu conteúdo	*Software*	Autenticidade, Integridade
Spoofing	Gerar falsos dados de posicionamento ao VANT, podendo até ganhar total controle da aeronave	*Hardware, Software*	Confidencialidade, Integridade
Códigos maliciosos	Ganhar acesso à informações privilegiadas para um ataque posterior	*Software*	Confidencialidade, Integridade
Engenharia social	Ganhar acesso à informações privilegiadas para um posterior ataque via hardware ou software	Físico	Confidencialidade
Negação de serviço	Indisponibilizar a comunicação entre as entidades	*Software*	Disponibilidade
Buffer overflow	Causar falhas no sistema, podendo até torná-lo indisponível por um determinado período de tempo	*Software*	Confidencialidade, Integridade, Disponibilidade

De acordo com as informações apresentadas no modelo para testes de invasão em VANTs, acredita-se que é então possível desenvolver e adaptar abordagens, mecanismos e ferramentas para a mitigação dos problemas de segurança atualmente encontrados nos VANTs.

No Capítulo 5 são discutidas as ferramentas e métodos utilizados para a adaptação do ambiente de simulação de voo de VANTs, levando em consideração alguns tipos de

ataques que podem ser lançados contra essas aeronaves. Testes de invasão são utilizados em ambiente simulado por meio do *software* de apoio computacional desenvolvido, sendo possível analisar o comportamento de um VANT ao se defrontar com alguns tipos de ataque.

5

Ambiente simulado para aplicação de Teste de Invasão em VANTs

Este capítulo apresenta as ferramentas necessárias para a execução do simulador de voo de um VANT *Ardupilot Mega SITL* e o desenvolvimento do *software* de apoio computacional para testes de invasão denominado *SITL PenTest*. O simulador é uma ferramenta de código aberto disponível para os interessados em VANTs, já o *SITL PenTest* é parte dos resultados deste trabalho e foi desenvolvido com o propósito específico de aplicação de testes de invasão em ambientes simulados de VANTs.

Neste capítulo ainda são apresentadas as características do *software SITL PenTest* tais como: interfaces de comunicação com o usuário; como utilizar o *software*; e os resultados obtidos com sua execução.

O simulador *Ardupilot Mega SITL* conta com: a simulação do voo de um VANT durante a realização de uma missão; uma estação base que pode enviar diversos comandos para o VANT, tais como: alterar rotas, pousar a aeronave, acionar ou desativar o piloto automático, dentre outras funcionalidades; além disso, o ambiente possui também canais de comunicação para trafegar os dados que são trocados entre o VANT e a estação base por meio dos protocolos TCP e UDP.

5.1 Ambiente de simulação

O *Ardupilot Mega SITL* realiza simulação do sistema embarcado conhecido como *Ardupilot Mega*, amplamente utilizado em aplicações de VANTs civis. O *Ardupilot Mega* conta

com mecanismos de *hardware* existentes em VANTs, *software* para comunicação, piloto automático, dentre outros. Muitos dos recursos existentes no *Ardupilot Mega* estão também presentes no *Ardupilot Mega SITL*.

Já o *software SITL PenTest* desenvolvido neste trabalho aborda três tipos de ataques, sendo que dois deles fazem parte dos ataques de negação de serviço, são eles: ataque de *flooding* e ataque de *jamming*. O outro é um ataque do tipo passivo, pois o sistema em questão não é testado de maneira agressiva, apenas coleta dados que são trocados entre a estação base e o VANT. Esse é conhecido como ataque de escuta ou *sniffing*.

Por se tratar de uma versão inicial para utilização dos testes de invasão, o *software SITL PenTest* pode ser estendido posteriormente com a inclusão de novos tipos de ataques, os quais ameaçam a segurança dos VANTs, e consequentemente, a conclusão das missões que executam.

A seguir, são apresentados detalhadamente o ambiente utilizado na simulação e qual o papel de cada componente que o compõe.

5.1.1 Sistemas operacionais utilizados

O *Linux Ubuntu* foi o sistema operacional utilizado para a simulação do ambiente de voo de um VANT e aplicação dos testes de invasão. Tal escolha se deve ao fato de que a instalação e configuração dos componentes que constituem o ambiente podem ser realizadas de maneira rápida e fácil em relação a outros sistemas operacionais.

Já a estação base foi instalada em um computador com sistema operacional *Windows*, o qual foi escolhido por já possuir o sistema operacional em execução. No entanto, a estação base pode ser executada tanto em *Linux* quanto em *Windows*. Dessa maneira, todos os módulos relacionados ao VANT foram executados no *Linux Ubuntu*, já a estação base foi executada no *Windows*.

5.1.2 Instalação das ferramentas necessárias

Todas as ferramentas utilizadas e a documentação consultada para o desenvolvimento deste trabalho são descritas a seguir.

a) Instalação do simulador Ardupilot Mega SITL

Por opção, o simulador de voo foi dividido em duas partes de maneira que os módulos utilizados por VANT e estação base fossem executados separadamente, de modo seme-

lhante a um ambiente real. Sendo assim, o processo de instalação do simulador também foi dividido entre os dois sistemas operacionais.

No sistema operacional *Linux Ubuntu* foram instalados os seguintes *softwares*:

- **Compilador para a linguagem C++**: Vários compiladores para a linguagem C++ estão disponíveis para a plataforma Linux, no entanto, neste trabalho foi utilizado o compilador GNU C++, ou g++, o qual é um dos mais conhecidos da plataforma em questão.

- ***pymavlink***: Uma biblioteca para o gerenciamento do protocolo *MAVLink*, o qual é utilizado na comunicação de alguns VANTs de pequeno porte. Um exemplo de seu uso é a análise de arquivos de *log* que contêm dados de telemetria de uma determinada aeronave que faz uso do protocolo *MAVLink*.

- ***pyserial***: Essa biblioteca encapsula o acesso à porta serial. O simulador não faz uso de tal porta, no entanto, é necessária a biblioteca para a execução do simulador.

- ***JSBSim***: Modelo de dinâmica de voo amplamente utilizado, o qual simula os elementos físicos que estão presentes nas aeronaves, sendo que todo o seu código fonte foi escrito em linguagem C++. O JSBSim fornece suporte para vários sistemas operacionais e pode ser utilizado tanto para testes quanto para estudos.

A documentação com os passos necessários para se instalar o simulador em sistemas operacionais *Linux* pode ser encontrada em (Autopilot, 2015).

Já no sistema operacional *Windows* foi instalado apenas o seguinte *software*:

- ***MAVProxy***: Uma estação base para ser utilizada em conjunto com VANTs, sejam eles em ambientes simulados ou até mesmo reais. Tal estação base fornece uma linha de comando para o envio de mensagens aos VANTs; pode ser executada em rede por diversos terminais e dá suporte a vários sistemas operacionais.

A documentação para a instalação do *MAVProxy* em sistemas operacionais *Windows*, é simples e possui poucos passos, a qual pode ser encontrada em (Dade, 2015).

b) Instalação das ferramentas utilizadas no desenvolvimento do software SITL PenTest

O *software SITL PenTest* faz uso de duas linguagens de programação distintas, C e Java. Assim sendo, torna-se necessária a instalação dos compiladores para a execução do código fonte desenvolvido de ambas linguagens, bem como o ambiente de desenvolvimento das mesmas, os quais são listados a seguir:

1. **Compilador gcc**
2. **Compilador javac**
3. ***Gedit***
4. ***Netbeans***

5.1.3 Ardupilot mega SITL

O dispositivo conhecido como *Software In The Loop* utilizado neste trabalho pode simular voos tanto de um avião não tripulado (*ArduPlane*) como o de um quadricóptero (*ArduCopter*), sem a necessidade de qualquer tipo de *hardware*.

O código fonte presente no sistema embarcado *Ardupilot Mega* também é encontrado no simulador SITL, o qual necessita apenas emular o *hardware* inexistente. A emulação é realizada por meio de códigos em C++, os quais são fielmente representados quando se comparados com um VANT real (Drones, 2015).

Para a execução do simulador *Ardupilot Mega SITL*, são necessários os *softwares* descritos na Seção 5.1.2, item a.

A arquitetura do simulador *Ardupilot Mega SITL* pode ser visualizada na Figura 5.1, a qual exibe como são realizadas as conexões entre os componentes que compõem o simulador, quais as portas são utilizadas para o tráfego de dados entre estação base e VANT, os tipos de pacotes que são enviados pelos canais de comunicação e qual sistema operacional é utilizado em cada módulo.

Nem todos os módulos presentes na Figura 5.1 são necessários para a execução do simulador, como os módulos **FlightGear** e **Other GCS**. O primeiro se refere a um simulador de voo que representa visualmente as aeronaves com riqueza de detalhes. Já o segundo é referente a outras estações base que podem ser adicionadas para o controlar o VANT. Sendo assim, as características presentes nesses módulos se tornam irrelevantes para o objetivo do trabalho.

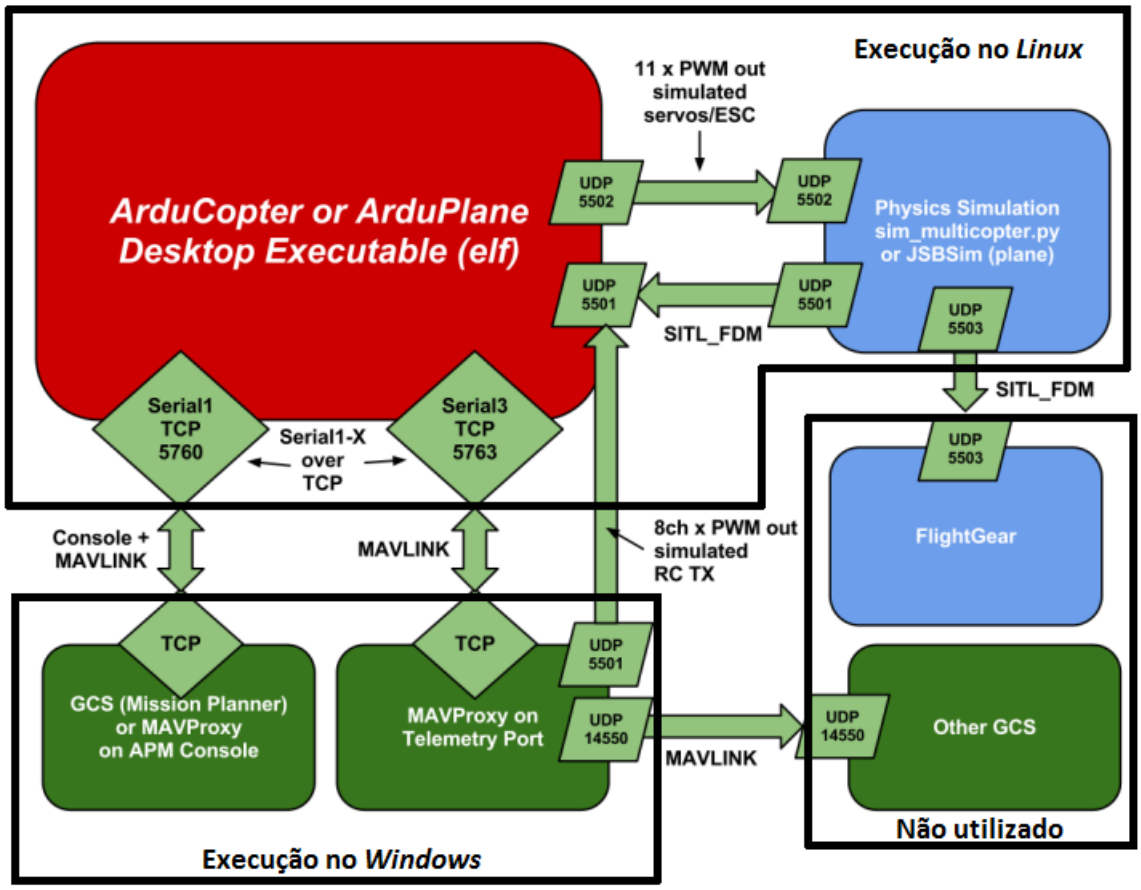

Figura 5.1: Arquitetura do *Ardupilot Mega SITL*. Adaptado de (Drones, 2015)

5.2 O software SITL PenTest

Esta Seção apresenta todos os componentes utilizados para o desenvolvimento do *software SITL PenTest*, o qual é responsável por realizar os testes de invasão no simulador SITL. Aqui são discutidas ferramentas, linguagens, dentre outros elementos necessários para o trabalho.

O nome escolhido para o *software* de apoio computacional se refere à simulação de VANTs por meio de *software (SITL - Software In The Loop)*, e o apelido para testes de invasão em língua inglesa *PenTest - Penetration Testing*.

a) *Flooding* com *Hping3*

A ferramenta *Hping3* fornece suporte para a geração de pacotes, os quais podem ser utilizados para diferentes propósitos, tais como: teste de *firewall*, teste de rede, mapeamento de rotas de pacotes, dentre outros. O *Hping3* foi desenvolvido para sistemas operacionais

baseados em *Linux* e sua interface de comunicação com o usuário se dá por meio de um terminal com linha de comando.

O *Hping3* foi incorporado ao *software SITL PenTest* com o intuito de enviar um grande número de pacotes para a máquina alvo, seja ela a estação base ou o próprio VANT, fazendo com que essa se torne indisponível, caracterizando assim um ataque de *flooding*, o qual é aplicado na camada de rede.

Ao se fazer uso de uma ferramenta amplamente conhecida e utilizada como o *Hping3*, o *software SITL PenTest* se torna mais seguro e confiável.

Além da utilização do *Hping3* para a geração de pacotes, o *software* para o ataque de *flooding* possui uma interface gráfica intuitiva e de fácil manuseio, voltada para a execução do ataque em um ambiente *SITL*.

Toda a implementação e integração com o *Hping3* foi realizada em linguagem Java.

b) Ataque de escuta ou *Sniffing*

O ataque de escuta é muitas vezes classificado como um ataque passivo, pois não visa gerar danos imediatos ao alvo. Seu objetivo é capturar dados que possam ser úteis em futuros ataques intrusivos.

Para que os dados capturados sejam úteis em um eventual ataque, o auditor realiza uma análise minuciosa de quais os tipos de dados que estão sendo trafegados, tipos de pacotes, dentre outras informações sensíveis que podem ajudar em um próximo ataque.

Para a implementação do ataque de escuta foram utilizadas a linguagem Java, juntamente com o a biblioteca *JNetPcap*. Essa biblioteca auxilia na captura e decodificação de pacotes capturados em tempo real.

O *software* de ataque de escuta desenvolvido possui as seguintes opções:

- **Escolha do número de porta a ser escutada**;
- **Escolha do número de pacotes a serem capturados**;
- **Gravação dos dados capturados em arquivo texto**.

c) Ataque de *Jamming*

Dentre os vários tipos de ataque de *jamming* existentes, optou-se pela escolha de apenas uma técnica, o *jamming* ativo constante, o qual envia quadros continuamente para o dispositivo que interconecta o VANT com a estação base, causando assim falhas na comunicação entre as entidades.

A implementação do ataque foi realizada em duas linguagens de programação, parte em C e parte em Java. Tal escolha se deu pela facilidade de se trabalhar com quadros na linguagem C, juntamente com seu desempenho.

Para a integração entre ambas linguagens, foi necessária a utilização da biblioteca *JNI (Java Native Interface)*, a qual permite que o código Java chame ou seja chamado por aplicações nativas ou bibliotecas desenvolvidas em outras linguagens, tais como: C, C++, dentre outras.

Dessa maneira, o núcleo da aplicação de *jamming* está escrita em linguagem C e outros recursos estão escritos em Java.

O ataque de *jamming* implementado trabalha na camada de enlace, o que faz com que ele só possa ser enviado em uma comunicação ponto a ponto. Dessa maneira, o *software* é capaz de trabalhar apenas com endereços *MAC (Media Access Control)*.

5.2.1 Uso do *software SITL PenTest*

O *software SITL PenTest* implementa três técnicas de ataque: *Jamming*; *Flooding*; e Escuta/*Sniffing*. Os dois primeiros são classificados como ataques de negação de serviço, enquanto que último é um ataque passivo. Os ataques de negação de serviço foram escolhidos, pois são grandes ameaças aos VANTs (Javaid et al., 2012). Já o ataque de *Sniffing* foi escolhido por estar presente na etapa de aquisição de informação, a qual é imprescindível para a realização dos testes de invasão.

Trabalhar com testes de invasão em ambiente simulado com o *software SITL Pentest* é interessante, pois as interfaces de comunicação com o usuário são intuitivas. Ao iniciá-lo, a tela principal é apresentada ao usuário, a qual é composta por um menu superior com as opções de ataque disponíveis e a opção de sair do sistema. A Figura 5.2, a Figura 5.3 e a Figura 5.4 mostram as telas iniciais do sistema.

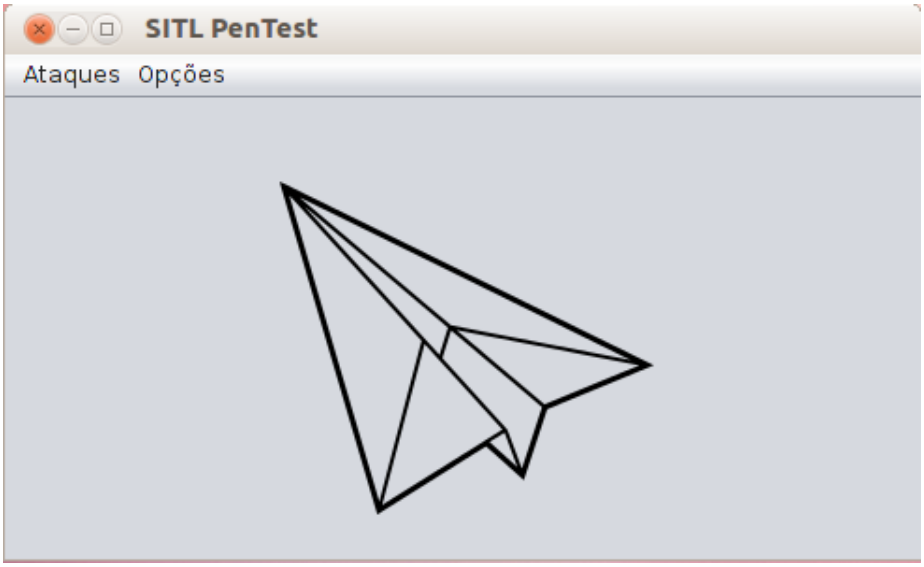

Figura 5.2: Tela inicial do sistema

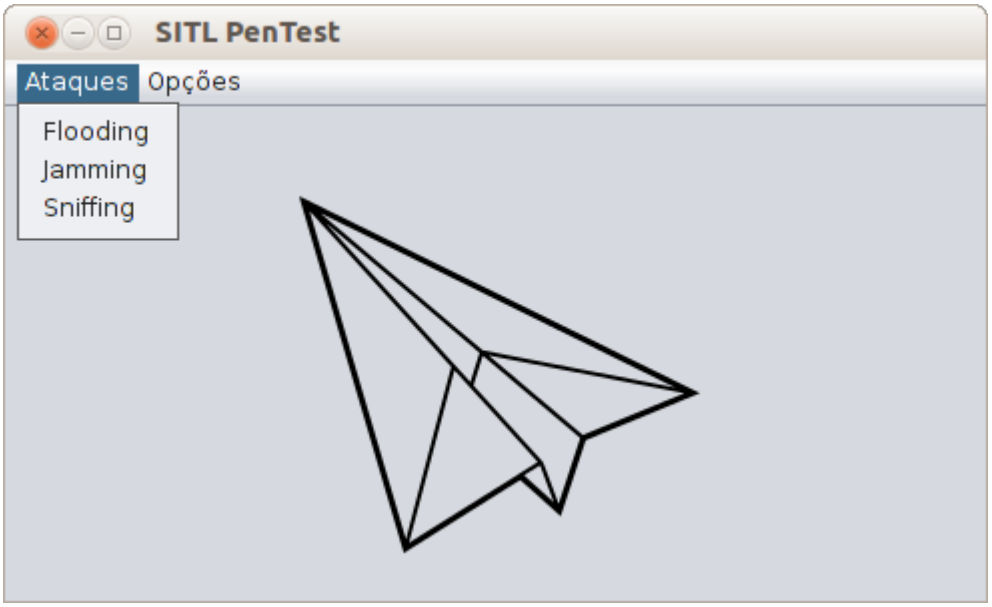

Figura 5.3: Tela inicial do sistema: opções de ataques

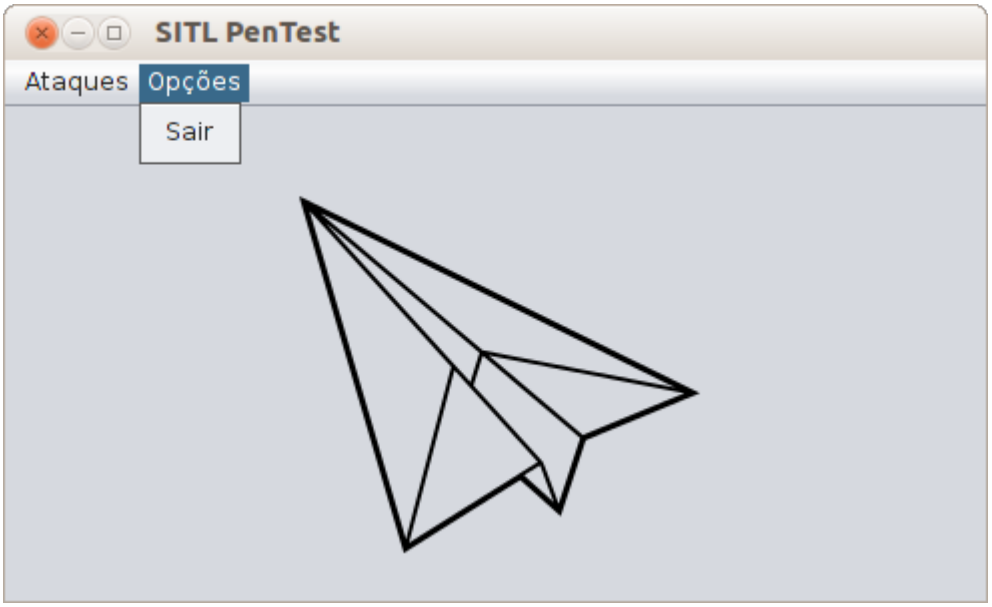

Figura 5.4: Tela inicial do sistema: opção de sair

Para a realização do ataque de *flooding*, é necessário apenas informar o endereço *IP (Internet Protocol)* da máquina alvo e a porta. Como o *software SITL PenTest* foi desenvolvido para atuar juntamente com o simulador *Ardupilot SITL*, o endereço IP da máquina local já vem preenchido (127.0.0.1), mas pode ser alterado. Existem dois botões disponíveis na interface, um para iniciar o ataque e outro para encerrá-lo. A Figura 5.5 apresenta as características descritas e disponíveis na interface gráfica.

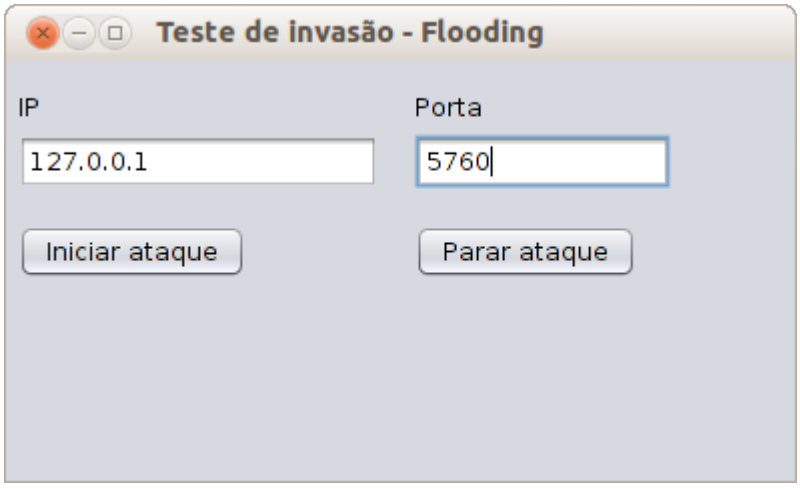

Figura 5.5: Tela para iniciar e finalizar o ataque de *flooding*

A maneira de se trabalhar com o ataque de *jamming* é semelhante ao anterior, no entanto, é necessário inserir apenas o endereço MAC da máquina alvo. Dessa maneira, o

ataque pode ser iniciado e encerrado em qualquer momento por meio de botões específicos. A Figura 5.6 apresenta as características descritas e disponíveis na interface gráfica.

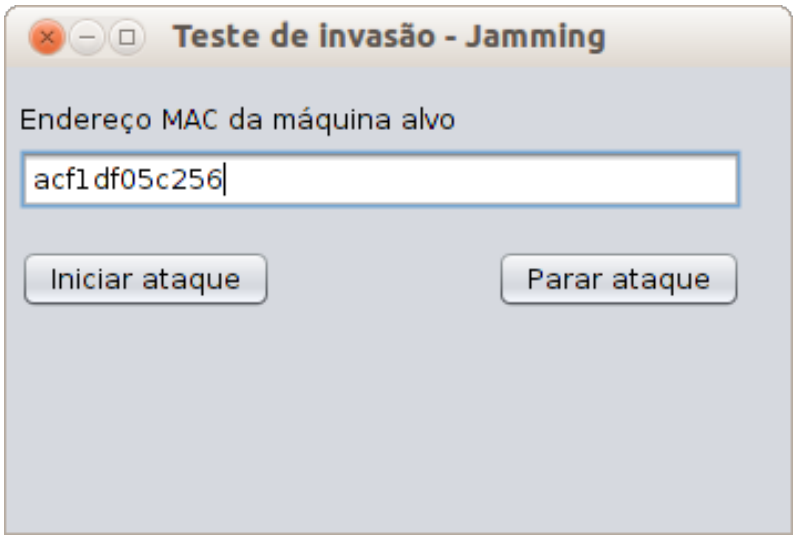

Figura 5.6: Tela para iniciar e finalizar o ataque de *jamming*

O último ataque presente no *software* desenvolvido é o de escuta/*sniffing*. Em sua interface gráfica existem três campos disponíveis ao usuário, endereço IP da máquina alvo, porta e o número de pacotes a serem capturados. Os dois últimos campos são opcionais e caso estejam em branco, não haverá filtro de que porta será monitorada e o número de pacotes capturados será de 10 (padrão).

Na interface do ataque de *sniffing* existem ainda dois botões, um para iniciar o ataque e outro para salvar os dados de pacotes capturados. Não existe o botão de parar o ataque, pois a execução é finalizada automaticamente quando o número de pacotes capturados alcança o valor digitado no campo, ou o valor padrão (caso o campo da interface esteja em branco). A interface gráfica do ataque de *sniffing* é apresentada na Figura 5.7.

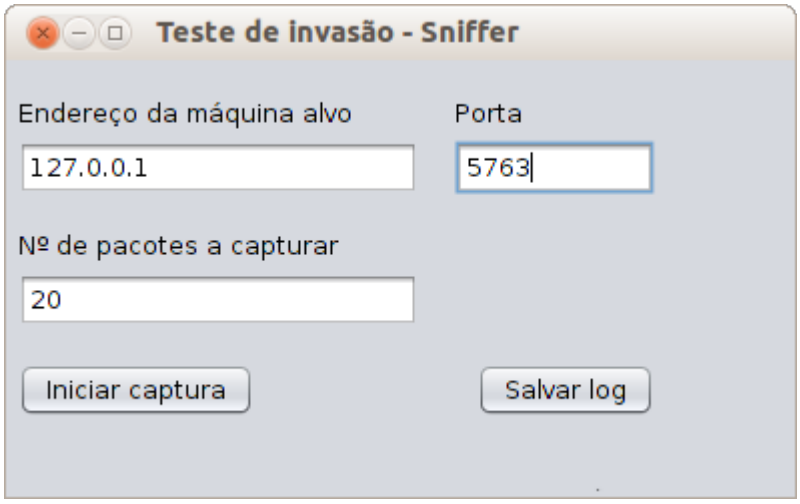

Figura 5.7: Tela para iniciar o ataque de *sniffing*

Ao selecionar a opção de salvar os dados por meio do botão *salvar log*, uma nova janela é aberta. Dessa maneira, o nome do arquivo pode ser inserido e gravado no local de preferência do usuário, conforme mostra a Figura 5.8.

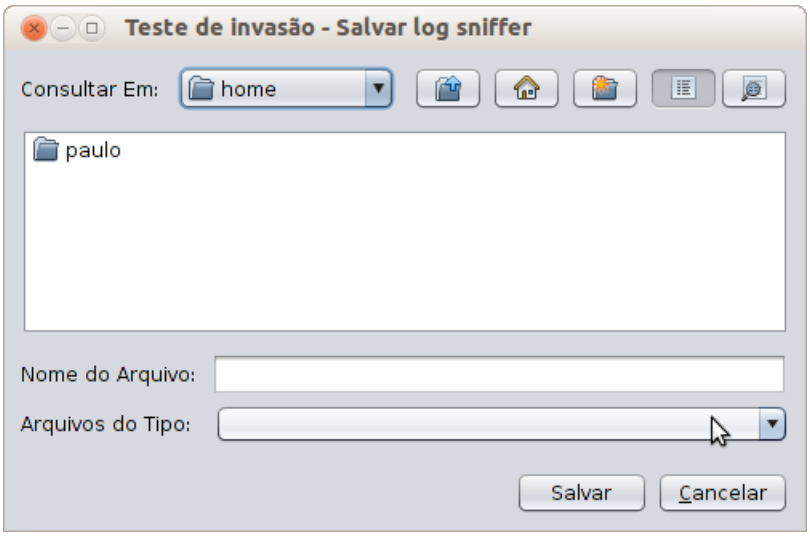

Figura 5.8: Tela para salvar os dados obtidos por meio do ataque de *sniffing*

5.2.2 Execução do *software SITL PenTest*

Esta Seção apresenta o modelo de comunicação entre VANT e estação base (ambos simulados), a integração do *software* desenvolvido com o simulador e os danos causados tanto ao VANT quanto à estação base quando aplicados os testes de invasão implementados.

É importante ressaltar que o VANT e a estação base foram simulados em diferentes computadores. Um deles responsável por todos componentes do VANT e outro contendo apenas a estação base, ambos interconectados por um canal de comunicação. A Figura 5.9 apresenta o modelo de comunicação utilizado na simulação.

A integração entre o simulador *Ardupilot Mega SITL* e o *software* de testes de invasão *SITL PenTest* foi realizada por meio das portas presentes no simulador destinadas à comunicação. Dessa maneira, o *software SITL PenTest* é capaz de se comunicar com o simulador.

Tendo em vista que o ambiente de simulação utilizado é o *Ardupilot Mega SITL*, tanto a estação base quanto o VANT podem ser visualizados de diferentes maneiras. A primeira é apresentada em linha de comando, a qual pode enviar comandos para o VANT e receber dados do voo e apresentá-los ao operador. Já o VANT pode ser visualizado em uma interface gráfica rica em detalhes, a qual possui o mapa em que a aeronave está situada (dentro de um círculo amarelo), a rota de voo (representada por linhas vermelhas), os pontos de referência da rota *(waypoints)*, a evolução no percurso da aeronave (o VANT em movimento rumo ao próximo ponto de referência), dentre outros. A Figura 5.10 e a Figura 5.11 apresentam a estação base e o VANT, respectivamente, ambos no momento em que o voo está ocorrendo sem qualquer tipo de problema.

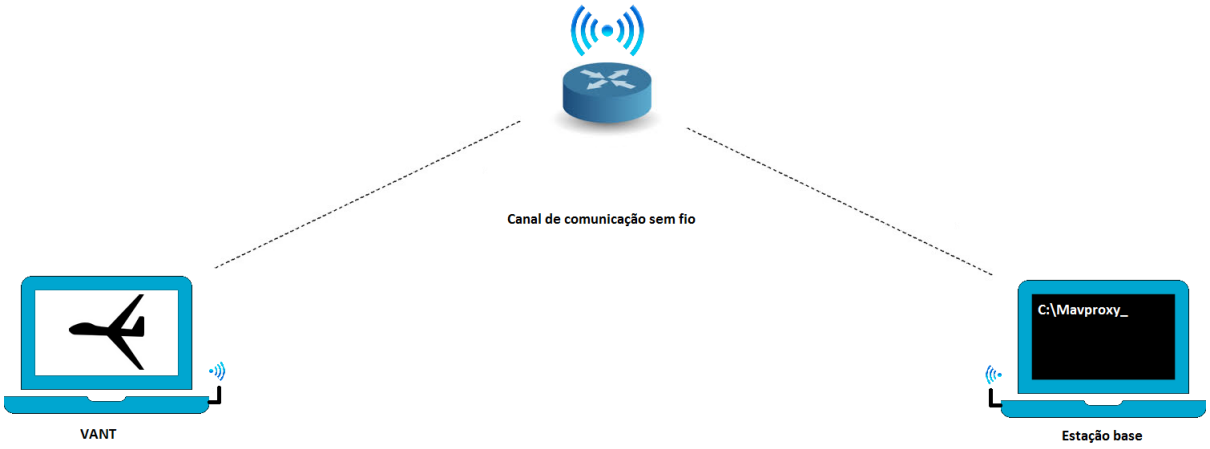

Figura 5.9: Modelo de comunicação utilizado na simulação

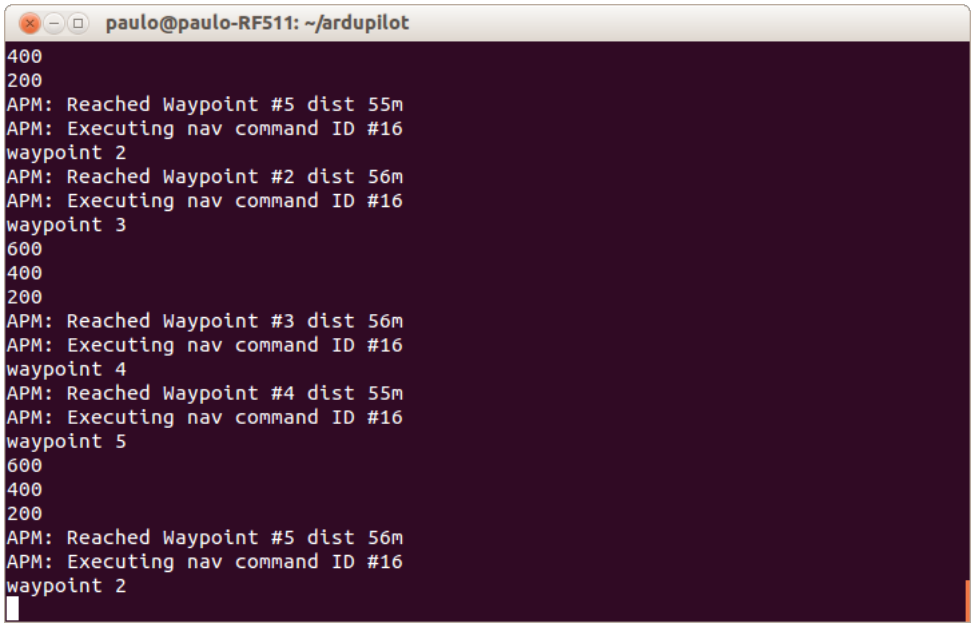

Figura 5.10: Estação base recebendo informações de um voo

Ao se aplicar testes de invasão no cenário apresentado, alguns aspectos são afetados na simulação. Os ataques de negação de serviço presentes no *software SITL PenTest* são os de *flooding* e de *jamming*, os quais provocam graves consequências na comunicação entre VANT e estação base. Já o ataque de *sniffing* não é caracterizado como um ataque que provoca grandes danos em um primeiro momento, pois sua função é coletar dados relevantes para futuros ataques.

O ataque de *flooding* aplicado ao ambiente simulado provoca a interrupção da comunicação entre a estação base e o VANT em voo. Com a perda da comunicação entre as entidades, comandos não podem ser enviados ao VANT e dados sobre o voo não são disponibilizados. Na simulação, o VANT para seu trajeto bruscamente e a estação base exibe mensagens de perda de conexão com o VANT. A Figura 5.12 apresenta a estação base com problemas de conexão após o ataque de *flooding*.

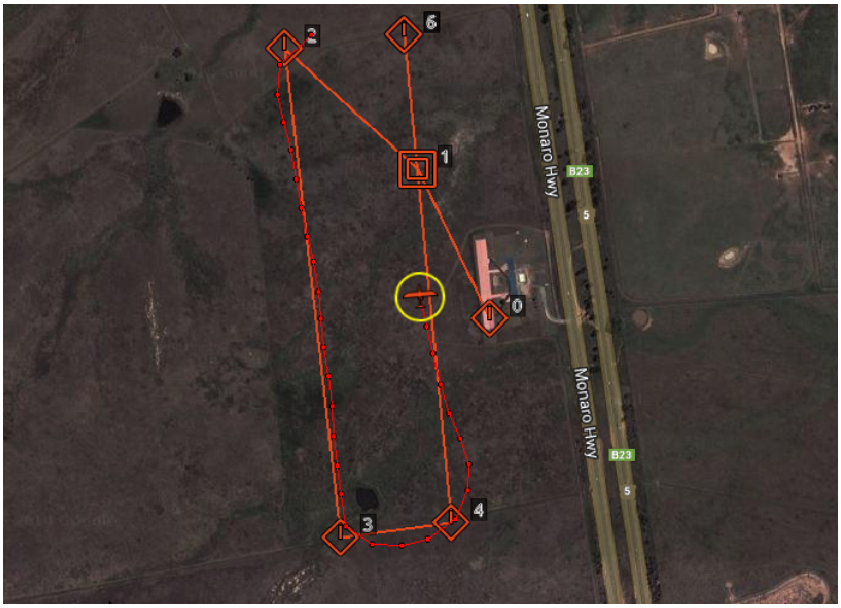

Figura 5.11: O mapa em que o VANT está situado, juntamente com sua rota de voo

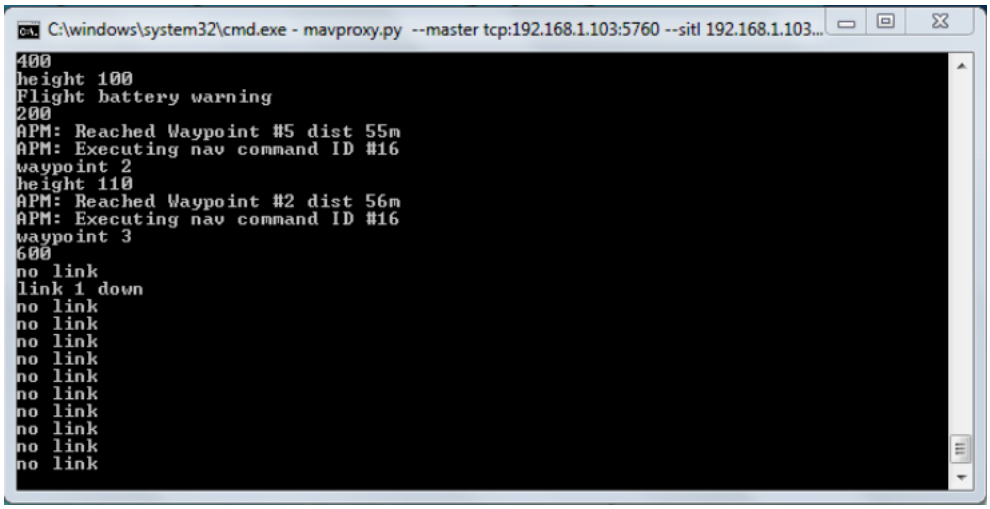

Figura 5.12: Estação base após ataque de *flooding* ao canal de comunicação

Assim como o ataque de *flooding*, o ataque de *jamming* tem como objetivo afetar ou interromper a comunicação entre a estação base e o VANT em voo. Consequentemente, tanto os problemas gerados quanto o comportamento do simulador são similares, ou seja, no ataque de *jamming* o VANT para bruscamente seu trajeto e a estação base apresenta mensagens de perda de comunicação. A Figura 5.13 apresenta a estação base com problemas de conexão após o ataque de *jamming*.

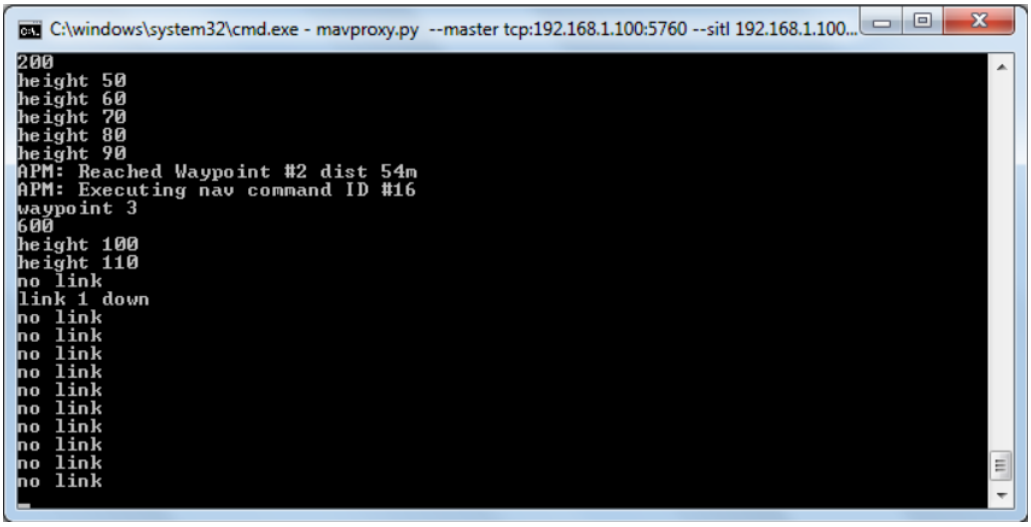

Figura 5.13: Estação base após ataque de *jamming* ao canal de comunicação

Diferentemente dos ataques anteriores, o ataque de *sniffing* não interrompe a comunicação entre VANT e estação base, pelo contrário, ele tira proveito da comunicação por meio da captura de pacotes, os quais podem ser gravados em arquivos de texto para uma análise posterior. Entre os dados capturados de um único pacote destacam-se como relevantes: origem do pacote (endereço IP e MAC); destino do pacote (endereço IP e MAC); portas de origem e destino; dados de *payload*; dados de protocolos; dentre outros. A Figura 5.14 apresenta os dados de um pacote capturado pelo ataque de *sniffing*.

Após a apresentação do *software SITL PenTest* e as consequências que os testes de invasão podem causar em um VANT em voo, nota-se que, tanto no ataque de *flooding* como de *jamming*, a comunicação é perdida entre a estação base e VANT. Tal fato evidencia que, caso um desses dois ataques sejam lançados com sucesso em um ambiente real, o operador da estação base não será capaz de obter informações sobre a missão em andamento e nem enviar comandos para a aeronave. Nesse contexto, o VANT estará operando de maneira totalmente autônoma, podendo se tornar uma ameaça a todos que estão ao seu redor.

Atualmente, alguns tipos de VANTs mais sofisticados possuem mecanismos que os permitem voar de modo autônomo e até pousar sem a necessidade de um operador. Para tais casos, os ataques de *jamming* e de *flooding* podem parecer pequenas ameaças, no entanto, se combinados com outros tipos de ataque, tais como o *spoofing*, até os mais sofisticados VANTs podem se tornar grandes ameaças.

```
DADOS DO FRAME
Frame:                    number = 0   NÚMERO DO PACOTE
Frame:                  timestamp = 2015-04-24 16:33:38.872
Frame:                 wire length = 55 bytes
Frame:              captured length = 55 bytes
Frame:  DADOS DO PROTOCOLO ETHERNET
Eth:  ******* Ethernet - "Ethernet" - offset=0 (0x0) length=14 protocol suite=LAN
Eth:
Eth:              destination = e0:06:e6:ff:9d:cb  MAC DESTINO
Eth:        .... ..0. .... .... = [2] LG bit
Eth:        .... ...0 .... .... = [2] IG bit
Eth:                 source = 90:a4:de:f5:46:95 MAC ORIGEM
Eth:        .... ..0. .... .... = [2] LG bit
Eth:        .... ...0 .... .... = [2] IG bit
Eth:                   type = 0x800 (2048) [ip version 4]
Eth:  DADOS DO PROTOCOLO IP
Ip:  ******* Ip4 - "ip version 4" - offset=14 (0xE) length=20 protocol suite=NETWORK
Ip:
Ip:                  version = 4
Ip:                    hlen = 5 [5 * 4 = 20 bytes, No Ip Options]
Ip:                 diffserv = 0x0 (0)
Ip:           0000 00.. = [0] code point: not set
Ip:           .... ..0. = [0] ECN bit: not set
Ip:           .... ...0 = [0] ECE bit: not set
Ip:                  length = 41
Ip:                    id = 0x4467 (17511)
Ip:                 flags = 0x2 (2)
Ip:           0.. = [0] reserved
Ip:           .1. = [1] DF: do not fragment: set
Ip:           ..0 = [0] MF: more fragments: not set
Ip:                offset = 0
Ip:                  ttl = 64 [time to live]
Ip:                  type = 6 [next: Transmission Control]
Ip:              checksum = 0x724E (29262) [correct]
Ip:                source = 192.168.1.100   IP ORIGEM
Ip:           destination = 192.168.1.101  IP DESTINO
Ip:  DADOS DO PROTOCOLO TCP
Tcp:  ******* Tcp offset=34 (0x22) length=20
Tcp:
Tcp:                  source = 5760  PORTA ORIGEM
Tcp:             destination = 50535 PORTA DESTINO
Tcp:                   seq = 0xA27B3681 (2725983873)
Tcp:                   ack = 0x37D0295C (936388956)
Tcp:                  hlen = 5
Tcp:              reserved = 0
Tcp:                 flags = 0x18 (24)
Tcp:           0... .... = [0] cwr: reduced (cwr)
Tcp:           .0.. .... = [0] ece: ECN echo flag
Tcp:           ..0. .... = [0] ack: urgent, out-of-band data
Tcp:           ...1 .... = [1] ack: acknowledgment
Tcp:           .... 1... = [1] ack: push current segment of data
Tcp:           .... .0.. = [0] ack: reset connection
Tcp:           .... ..0. = [0] ack: synchronize connection, startup
Tcp:           .... ...0 = [0] fin: closing down connection
Tcp:                 window = 123
Tcp:              checksum = 0x1725 (5925) [correct]
Tcp:                 urgent = 0
Tcp:
Data:  ******* Payload offset=54 (0x36) length=1
Data:
0036: fe                                                .
-------------------------------------------------------------------
                                                  DADOS DE PAYLOAD
0000: *e0 06 e6 ff  9d cb 90 a4  de f5 46 95  08 00*45 00   ..........F...E.
0010: 00 29 44 67  40 00 40 06  72 4e c0 a8  01 64 c0 a8   .)Dg@.@.rN...d..
0020: 01 65*16 80  c5 67 a2 7b  36 81 37 d0  29 5c 50 18   .e...g.{6.7.)\P.
0030: 00 7b 17 25  00 00*fe                        .{.%...
```

Figura 5.14: Dados de um pacote enviado do VANT para a estação base capturado pelo ataque de *sniffing*

Já para VANTs que não possuem a tecnologia de voo autônomo, os ataques de *flooding* e *spoofing* se apresentam como grandes ameaças, sendo capazes de causar a queda de tais tipos de aeronave, ocasionando grandes catástrofes.

Como previamente mencionado, o ataque de *sniffing* não é intrusivo, no entanto, em um ambiente real, ele é capaz de coletar vários tipos de informações presentes na comunicação entre o VANT e a estação base. Isso faz com que ataques de maior impacto possam ser realizados contra o VANT com maior chance de sucesso.

Dentro do contexto dos testes de invasão, o *software SITL PenTest* abrange duas etapas, as quais foram realizadas por meio do teste de caixa cinza e são descritas a seguir:

- **Aquisição de informação**: Consiste da aquisição de informações que possam ser relevantes na condução dos testes de invasão. É neste momento em que a utilização de ferramentas é iniciada, como escaneadores de portas, *softwares* específicos para captura de informações, dentre outros. Nessa etapa dos testes de invasão que o *sniffer* presente no *software SITL PenTest* pode ser utilizado pelo auditor.

- **Exploração**: A etapa de exploração é a mais esperada pelos auditores, pois nesse momento as vulnerabilidades encontradas em fases anteriores são exploradas, fazendo com que o planejamento dos testes realizado anteriormente seja colocado a prova. Nessa etapa dos testes de invasão podem ser empregadas as funções com ataques de *flooding* e *jamming* presentes no *software SITL PenTest*.

Dessa maneira, para se realizar um teste de invasão completo e seguindo todos os passos necessários, é indispensável a utilização de ferramentas adicionais responsáveis pelas outras etapas não cobertas pelo *software*.

Sendo assim, o *software SITL PenTest* pode executar as etapas de aquisição de informação e de exploração presentes nos testes de invasão, as quais cobrem três tipos de ataque em ambiente simulado. Tal *software* auxilia a visualização dos danos que tais tipos de ataque podem causar em um canal de comunicação entre VANT e estação base, apresenta o comportamento de ambas as entidades em um eventual ataque, e verifica se as entidades estão expostas a tais ataques. Outra função de grande importância, é servir como um *software* base para o incentivo na utilização de testes de invasão em ambientes de VANTs, o qual pode detectar vulnerabilidades, corrigí-las e assim prevenir as aeronaves futuramente.

6

Conclusão e trabalhos futuros

O número de VANTs empregados nas mais diversas tarefas vem aumentando rapidamente. Por serem classificados como sistemas embarcados críticos, eles necessitam de cuidados especiais em relação à sua segurança, pois caso um ataque seja efetuado com sucesso contra um VANT, uma grande catástrofe pode se desencadear e causar grandes perdas, como financeiras, humanas, ambientais ou outras.

Dentro deste contexto, o principal objetivo deste trabalho foi desenvolver um *software* capaz de realizar testes de invasão em sistemas de comunicação de VANTs em ambiente simulado, possibilitando que sistemas mais seguros possam ser desenvolvidos. Esse objetivo foi alcançado com o desenvolvimento do *software SITL PenTest*.

Os objetivos específicos alcançados neste trabalho foram:

- **Analisar as principais vulnerabilidades em VANTs**.
- **Analisar os principais tipos de ataques que ocorrem contra VANTs**: Um amplo levantamento de vulnerabilidades atualmente encontradas em VANTs foi realizado, juntamente com as ameaças que podem se tornar realidade e os componentes mais afetados em eventuais ataques ao ambiente dos VANTs.
- **Estudar o comportamento de ferramentas de teste de invasão**: Algumas ferramentas de testes de invasão foram estudadas para o desenvolvimento do *software SITL PenTest*, as quais são partes do sistema operacional específico para esse propósito, o *Kali Linux* (Pritchett e De Smet, 2013). Dentre as ferramentas estudadas estão: *Aircrack-NG*; *Spooftooph*; *Bluemaho*; dentre outras.

- **Desenvolver um modelo para aplicação de testes de invasão em VANTs**: O modelo foi desenvolvido, evidenciando a importância dos testes de invasão na segurança dos VANTs.
- **Projetar e desenvolver o *software* de apoio computacional**: O *software* foi desenvolvido e testado com sucesso.
- **Integrar o *software* desenvolvido a um simulador de VANT**: O *software SITL PenTest* é capaz de se comunicar com o simulador *Ardupilot Mega SITL*.
- **Avaliar a ferramenta realizando etapas dos testes de invasão em sistemas de comunicação de VANTs em ambiente simulado**: Os testes foram realizados com sucesso e os resultados apresentados no Capítulo 5

Em meio ao objetivo principal e aos objetivos específicos, destacam-se: a realização do levantamento mais completo das vulnerabilidades, a criação do modelo para testes de invasão e o desenvolvimento do *software SITL PenTest*.

O modelo desenvolvido neste trabalho expressa de modo geral como os testes de invasão podem ser aplicados e os benefícios de sua utilização.

Ao se realizar o levantamento e a análise de vulnerabilidades atualmente encontradas em ambientes de VANTs, nota-se que estes estão suscetíveis a diversos tipos de ataques, os quais se apresentam como grandes ameaças para o sucesso de missões aéreas executadas por essas aeronaves.

O *software* de apoio computacional desenvolvido *(SITL PenTest)* pode vir a se tornar relevante para a área de testes de invasão em sistemas de comunicação de VANTs, pois esta ainda é pouco explorada. Tal *software* testa se o ambiente do VANT está ou não vulnerável aos ataques de *flooding*, *jamming* e *sniffing*, evidenciando as falhas existentes, as quais podem ser corrigidas em um estágio precoce de desenvolvimento de um VANT.

O *software* foi desenvolvido em um primeiro momento apenas para ambientes simulados, no entanto, pode servir como base para o desenvolvimento de *softwares* mais complexos e para o avanço dos testes de invasão no campo de atuação dos VANTs. Além disso, o *software SITL PenTest* pode ser extendido para a cobertura de novos ataques e até para a aplicação em um ambiente real.

É importante ressaltar que a aplicação de testes de invasão em um ambiente real continua sendo um desafio, pois como discutido na Seção 3.6, os sistemas não tripulados atuais são altamente complexos, fator que dificulta o trabalho dos auditores responsáveis pela aplicação dos teste de invasão, os quais devem conhecer as particularidades pertencentes ao ambiente dos VANTs.

A partir de todos os aspectos citados, este trabalho gerou contribuição científica em uma área até então pouco explorada. Além disso, a realização de um amplo levantamento de vulnerabilidades em sistemas de comunicação de VANTs pode ser muito útil em trabalhos futuros, assim como o modelo e o *software SITL PenTest* desenvolvidos.

6.1 Trabalhos futuros

Como trabalhos futuros, têm-se:

- **Continuar o monitoramento de vulnerabilidades em ambientes de VANTs**: Necessário para a melhoria do trabalho e constante atualização;
- **Estender o *software SITL PenTest***: Novas funcionalidades, tais como: inserção de novos tipos de ataques, integração com outros simuladores de VANTs e o principal ponto de melhoria, que seria a portabilidade do *software* para trabalhar em um ambiente real, sendo assim possível a realização dos testes de invasão em projetos de VANTs utilizados atualmente.

REFERÊNCIAS

Ali, S.; Heriyanto, T. *Backtrack 4: Assuring security by penetration testing: Master the art of penetration testing with backtrack.* Packt Publishing Ltd, 2011.

Alves-Foss, J. Multiprotocol attacks and the public key infrastructure. In: *Proc. National Information System Security Conference*, 1998, p. 566–576.

Autopilot, A. M. Setting up sitl on linux. [*On-line*].
Disponível em <http://dev.ardupilot.com/wiki/setting-up-sitl-on-linux/>

Barbeau, M. Wimax/802.16 threat analysis. In: *Proceedings of the 1st ACM International Workshop on Quality of Service &Amp; Security in Wireless and Mobile Networks*, Q2SWinet '05, New York, NY, USA: ACM, 2005, p. 8–15 (*Q2SWinet '05*,).
Disponível em http://doi.acm.org/10.1145/1089761.1089764

Barros, U. S. d. *Um sistema baseado na teoria do perigo para detectar ataques jamming em manets.* Dissertação de Mestrado, Universidade Federal do Paraná, Curitiba, 2011.
Disponível em <http://hdl.handle.net/1884/32220>

Berger, A. S. *Embedded systems design: an introduction to processes, tools, and techniques.* CRC Press, 272 p., 2001.

Bhatti, J. A.; Humphreys, T. E.; Shepard, D. P. Drone hack: spoofing attack demonstration on a civilian unmanned aerial vehicle. *GPS World*, p. 30, 2012.

Bidgoli, H. The internet encyclopedia volume 1 a–e. 2004.

cert.br Cartilha de segurança para internet - parte iv: Códigos maliciosos (malware). [*On-line*].
Disponível em <http://cartilha.cert.br/malware/>

Chapman, R.; Hamilton, D.; Box, D.; Kuhr, M.; MacDonald, J.; Hamilton, S. Simulation of army unmanned aerial vehicle communications. In: *Simulation Conference, 2007 Winter*, IEEE, 2007, p. 1324–1327.

Coley, G. *Beagleboard system reference manual.* beagleboard.org, Texas, TX, USA, 2009.
Disponível em <http://beagleboard.org/static/BBSRM_latest.pdf>

Cowan, C.; Wagle, P.; Pu, C.; Beattie, S.; Walpole, J. Buffer overflows: Attacks and defenses for the vulnerability of the decade. In: *DARPA Information Survivability Conference and Exposition, 2000. DISCEX'00. Proceedings*, IEEE, 2000, p. 119–129.

Dade, S. Mavproxy: A uav ground station software package for mavlink based systems. [*On-line*].
Disponível em <http://tridge.github.io/MAVProxy/>

Drones, D. Ardupilot: Software in the loop. [*On-line*].
Disponível em https://code.google.com/p/ardupilot-mega/wiki/SITL

Faughnan, M. S.; Hourican, B. J.; MacDonald, G. C.; Srivastava, M.; Wright, J.; Haimes, Y.; Andrijcic, E.; Guo, Z.; White, J. Risk analysis of unmanned aerial vehicle hijacking and methods of its detection. In: *Systems and Information Engineering Design Symposium (SIEDS), 2013*, IEEE, 2013, p. 145–150.

Frost; Sullivan Study analysing the current activities in the field of uav. *European Commision Enterprise and Industry Directorate-General*, p. 96, 2007.

Gorman, S.; Dreazen, Y. J.; Cole, A. Insurgents hack us drones. *Wall Street Journal*, v. 17, 2009.

Hartmann, K.; Steup, C. The vulnerability of uavs to cyber attacks-an approach to the risk assessment. In: *Cyber Conflict (CyCon), 2013 5th International Conference on*, IEEE, 2013, p. 1–23.

Humphreys, T. How college students hijacked a government spydrone. [*On-line*].
Disponível em <http://www.foxnews.com/tech/2012/06/25/drones-vulnerable-to-terrorist-hijacking-researchers-say/>

Javaid, A. Y.; Sun, W.; Alam, M. Uavsim: A simulation testbed for unmanned aerial vehicle network cyber security analysis. In: *Globecom Workshops (GC Wkshps)*, IEEE, 2013, p. 1432–1436.

Javaid, A. Y.; Sun, W.; Devabhaktuni, V. K.; Alam, M. Cyber security threat analysis and modeling of an unmanned aerial vehicle system. In: *Homeland Security (HST)*, IEEE, 2012, p. 585–590.

Katopodis, P.; Katsis, G.; Walker, O.; Tummala, M.; Michael, J. A hybrid, large-scale wireless sensor network for missile defense. In: *System of Systems Engineering, 2007. SoSE '07. IEEE International Conference on*, 2007, p. 1–5.

Kim, A.; Wampler, B.; Goppert, J.; Hwang, I.; Aldridge, H. Cyber attack vulnerabilities analysis for unmanned aerial vehicles. *The American Institute of Aeronautics and Astronautics: Reston, VA, USA*, 2012.

Kobezak, P.; Abbot-McCune, S.; Tront, J.; Marchany, R.; Wicks, A. Universal framework for unmanned system penetration testing. In: *SPIE Defense, Security, and Sensing*, International Society for Optics and Photonics, 2013, p. 87411A–87411A.

Kurose, J. F.; Ross, K. W.; Marques, A. S.; Zucchi, W. L. *Redes de computadores ea internet: Uma abordagem top-down.* Pearson, 945 p., 2010.

Li, M.; Koutsopoulos, I.; Poovendran, R. Optimal jamming attacks and network defense policies in wireless sensor networks. In: *INFOCOM 2007. 26th IEEE International Conference on Computer Communications. IEEE*, 2007, p. 1307–1315.

Li, Q.; Yao, C. *Real-time concepts for embedded systems.* CRC Press, 2003.

Mansfield, K.; Eveleigh, T.; Holzer, T.; Sarkani, S. Unmanned aerial vehicle smart device ground control station cyber security threat model. In: *Technologies for Homeland Security (HST), 2013 IEEE International Conference on*, 2013, p. 722–728.

O'Gorman, J.; Kearns, D.; Aharoni, M. *Metasploit: the penetration tester's guide.* No Starch Press, 2011.

Pritchett, W. L.; De Smet, D. *Kali linux cookbook.* Packt Publishing Ltd, 2013.

Radosavac, S.; Benammar, N.; Baras, J. S. Cross-layer attacks in wireless ad hoc networks. In: *in Proceedings of CISS*, 2004.

Ravi, S.; Raghunathan, A.; Kocher, P.; Hattangady, S. Security in embedded systems: Design challenges. *ACM Trans. Embed. Comput. Syst.*, v. 3, n. 3, p. 461–491, 2004.
Disponível em <http://doi.acm.org/10.1145/1015047.1015049>

Razak, S. A.; Furnell, S.; Brooke, P. Attacks against mobile ad hoc networks routing protocols. In: *Proceedings of 5th Annual Postgraduate Symposium on the Convergence of Telecommunications, Networking & Broadcasting (PGNET'04)*, Citeseer, 2004.

Rezende, P. A. D. Criptografia e segurança na informática. *Apostila-Capítulos*, v. 1, n. 2, p. 3, 1998.

Rocha, R. C. d. O. *Detecção em tempo real de ataques de negação de serviço na rede de origem usando um classificador bayesiano simples.* Monografia, Pontifícia Universidade Católica de Minas Gerais, Belo Horizonte, 1990.

Rudinskas, D.; Goraj, Z.; Stankūnas, J. Security analysis of uav radio communication system. *Aviation*, v. 13, n. 4, p. 116–121, 2009.

Samad, A.; Kamarulzaman, N.; Hamdani, M.; Mastor, T.; Hashim, K. The potential of unmanned aerial vehicle (uav) for civilian and mapping application. In: *System Engineering and Technology (ICSET), 2013 IEEE 3rd International Conference on*, 2013, p. 313–318.

Schuba, C.; Krsul, I.; Kuhn, M.; Spafford, E.; Sundaram, A.; Zamboni, D. Analysis of a denial of service attack on tcp. In: *Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on*, 1997, p. 208–223.

Staff, J. Joint publication 1-02. *Department of Defense Dictionary of Military and Associated Terms*, v. 12, p. 297, 2001.

Tanenbaum, A. S. *Redes de computadores.* 4 ed. Elsevier Brasil, 945 p., 2003.

Tanenbaum, A. S. *Sistemas operacionais modernos.* 3 ed. São Paulo: Prentice Hall, 2009.

Venugopalan, R.; Ganesan, P.; Peddabachagari, P.; Dean, A.; Mueller, F.; Sichitiu, M. Encryption overhead in embedded systems and sensor network nodes: Modeling and analysis. In: *Proceedings of the 2003 international conference on Compilers, architecture and synthesis for embedded systems*, ACM, 2003, p. 188–197.

Wang, W.; Sun, Y.; Li, H.; Han, Z. Cross-layer attack and defense in cognitive radio networks. In: *Global Telecommunications Conference (GLOBECOM 2010), 2010 IEEE*, 2010, p. 1–6.

Weidman, G. *Penetration testing: A hands-on introduction to hacking.* No Starch Press, 2014.

Wen, H.; Huang, P. Y.-R.; Dyer, J.; Archinal, A.; Fagan, J. Countermeasures for gps signal spoofing. In: *ION GNSS*, 2005, p. 13–16.

Wollinger, T.; Guajardo, J.; Paar, C. Security on fpgas: State-of-the-art implementations and attacks. *ACM Transactions on Embedded Computing Systems (TECS)*, v. 3, n. 3, p. 534–574, 2004.