Análise Comparativa: Abordagens AI-Driven vs. Human-Driven na Fase de Reconnaissance em Cibersegurança.

Pedro M. M. Hernandes[1], **Renan M. V. de Carvalho**[1]

[1]Pontifícia Universidade Católica de Minas Gerais
Caixa Postal 1.686 – 30.535-901 – Belo Horizonte – MG – Brazil

pedro.hernandes@sga.pucminas.br, renan.carvalho.1446987@sga.pucminas.br

1. Introdução

A segurança de sistemas computacionais tornou-se uma preocupação central para organizações públicas e privadas em todo o mundo. Com a proliferação de dispositivos conectados, infraestruturas distribuídas e sistemas ciberfísicos, a superfície de ataque disponível a agentes maliciosos cresce de forma contínua e acelerada [Heiding et al. 2023]. Nesse contexto, o teste de penetração (ou *pentesting*) estabelece-se como uma das principais práticas proativas para a avaliação da postura de segurança de sistemas, consistindo na simulação controlada de ataques com o objetivo de identificar vulnerabilidades antes que possam ser exploradas de forma maliciosa [Deng et al. 2024].

O mapeamento de infraestruturas, etapa inicial e crítica do processo de teste de penetração, compreende atividades de reconhecimento, enumeração de serviços e identificação de superfícies de ataque. Essa fase determina, em grande medida, a qualidade e a abrangência de todo o processo subsequente: falhas no reconhecimento implicam diretamente em lacunas na cobertura das vulnerabilidades avaliadas [Skandylas and Asplund 2025]. Tradicionalmente, essa etapa é conduzida por profissionais especializados que combinam ferramentas consolidadas, como Nmap e Metasploit, com conhecimento tácito acumulado ao longo de anos de experiência prática. Esse modelo, denominado aqui como paradigma *human-driven*, apresenta limitações estruturais relevantes: é intensivo em mão de obra qualificada, difícil de escalar, e altamente dependente do repertório individual do profissional envolvido [Heiding et al. 2023, Skandylas and Asplund 2025].

Diante dessas limitações, a comunidade científica tem investigado abordagens de automação para o processo de teste de penetração. Propostas baseadas em aprendizado por reforço, planejamento por inteligência artificial e grafos de ataque avançaram no sentido de reduzir a dependência humana, mas enfrentaram dificuldades crônicas de generalização e de aplicabilidade em ambientes reais heterogêneos [Skandylas and Asplund 2025]. Mais recentemente, os Modelos de Linguagem de Grande Escala (LLMs, do inglês *Large Language Models*) emergiram como uma alternativa promissora, dada sua capacidade de raciocínio contextual, geração de código e interpretação de saídas de ferramentas em linguagem natural [Deng et al. 2024, Wu et al. 2025b]. Sistemas como PentestGPT [Deng et al. 2024] e CurriculumPT [Wu et al. 2025b] demonstraram que agentes baseados em LLMs conseguem conduzir etapas significativas do processo de teste de penetração de forma autônoma, incluindo a fase de reconhecimento e mapeamento de infraestruturas.

Apesar dos avanços recentes, a literatura ainda carece de uma análise comparativa sistemática entre o paradigma tradicional de automação, apoiado em ferramentas determinísticas e roteiros fixos operados por humanos, e o paradigma emergente baseado em LLMs, no que diz respeito especificamente à etapa de mapeamento de infraestruturas em cibersegurança ofensiva. Trabalhos como o de Skandylas e Asplund [Skandylas and Asplund 2025] formalizam o problema do teste de penetração automatizado e propõem arquiteturas orientadas a decisões em tempo de execução, mas não exploram diretamente o papel dos LLMs nessa etapa. Por outro lado, estudos como o de Deng et al. [Deng et al. 2024] e Wu et al. [Wu et al. 2025b] concentram-se na avaliação de desempenho global de sistemas baseados em LLMs, sem isolar a contribuição de cada fase do processo nem compará-la com métricas equivalentes obtidas no paradigma tradicional.

Essa lacuna é relevante por razões práticas e científicas. Do ponto de vista prático, a escolha entre abordagens impacta diretamente o custo operacional, a cobertura do reconhecimento e o tempo necessário para a identificação de ativos e serviços expostos. Do ponto de vista científico, compreender em que condições os LLMs superam ou são superados pelas abordagens tradicionais fornece subsídios para o design de sistemas híbridos mais eficientes. Além disso, o crescente uso de infraestruturas complexas em domínios críticos, como veículos autônomos [Wu et al. 2025b] e dispositivos IoT residenciais [Heiding et al. 2023], reforça a urgência de se estabelecer metodologias de avaliação de segurança que sejam ao mesmo tempo escaláveis e adaptáveis. Nesse sentido, a presente pesquisa propõe-se a preencher essa lacuna por meio de uma investigação empírica comparativa, contribuindo para uma compreensão mais precisa do potencial e dos limites de cada paradigma no contexto específico do mapeamento de infraestruturas.

O objetivo geral desta pesquisa é investigar a eficiência situacional as fronteira operacionais entre a automação tradicional (ferramentas determinísticas guiadas por humanos) e os agentes autônomos baseados em Modelos de Linguagem de Grande Escala (LLMs), com foco exclusivo na fase de reconhecimento e mapeamento de infraestruturas. Para o alcance deste propósito, definem-se os seguintes objetivos específicos:

- Analisar quantitativamente o desempenho de ambas as abordagens, comparando métricas como taxa de conclusão de tarefas, tempo de execução e custo operacional (consumo de *tokens* versus uso de CPU).
- Investigar as configurações de infraestrutura e os cenários de rede nos quais a intuição humana aliada ao ferramental tradicional ainda se faz estritamente necessária.
- Descobrir os limites práticos, as barreiras de escalabilidade e as taxas de falha sistêmica inerentes aos agentes LLMs, tais como alucinações técnicas, degradação de performance por perda de contexto em conversas longas e incapacidade de interpretar saídas não estruturadas — ao processarem grandes volumes de dados de varredura provenientes de ambientes do mundo real.
- Analisar o grau de autonomia efetiva das ferramentas emergentes de IA, determinando em quais etapas precisas do fluxo de trabalho a intervenção de um especialista (abordagem *Human-in-the-loop*) torna-se indispensável para garantir a precisão e evitar pontos cegos no mapeamento da superfície de ataque.

A execução e o financiamento desta pesquisa justificam-se pela urgência em mitigar o alto risco financeiro e operacional associado à adoção precipitada de soluções de

Inteligência Artificial no setor de cibersegurança. Atualmente, o mercado corporativo investe volumes expressivos de capital em plataformas de segurança autônomas e agentes inteligentes, muitas vezes motivado pelo entusiasmo tecnológico, mas sem possuir diretrizes empíricas sobre o real Retorno sobre Investimento (ROI) e a eficácia prática dessas ferramentas quando comparadas às rotinas e equipes tradicionais de *pentest*.

Nesse sentido, faz-se estritamente necessário investir tempo, estudo e recursos nesta investigação empírica, pois ela entrega um diagnóstico vital para a resiliência das organizações. Descobrir as reais limitações da automação baseada em LLMs, especificamente na fase crítica de reconhecimento, que determina a qualidade de todas as etapas subsequentes de um ataque simulado — previne que empresas desenvolvam uma perigosa falsa sensação de segurança corporativa. Uma varredura automatizada que ignora o contexto de negócio ou apresenta alucinações pode atestar como segura uma infraestrutura vulnerável, resultando em brechas que custariam milhões em caso de exploração maliciosa e vazamento de dados. Ao mesmo tempo, a pesquisa evita o desperdício de capital humano, apontando cientificamente onde o especialista deve focar sua carga cognitiva e intuição, ao invés de atuar em tarefas repetitivas que a máquina já domina.

Sob uma perspectiva científica, o estudo se destaca por abordar uma lacuna metodológica relevante no estado da arte. Embora a literatura recente tenha se concentrado no desenvolvimento de modelos autônomos baseados em IA, ainda há escassez de análises comparativas rigorosas que estabeleçam uma linha de base clara entre esse paradigma emergente e abordagens tradicionais orientadas por humanos.

Ao investigar esses dois extremos, o trabalho oferece uma avaliação fundamentada em dados reprodutíveis, permitindo examinar em que medida a tecnologia de LLM atingiu maturidade estrutural, apresenta viabilidade em termos de custo-benefício e demonstra confiabilidade arquitetural. Dessa forma, o estudo contribui para distinguir o potencial teórico dos resultados efetivamente observados na prática, especialmente no contexto do mapeamento de infraestruturas críticas.

2. Referencial Teórico

Esta seção apresenta os conceitos fundamentais que embasam a investigação proposta. A estrutura esta organizada de modo a conduzir o leitor do processo geral de testes de penetração ate os mecanismos específicos de automação e as métricas capazes de diferenciar abordagens tradicionais das baseadas em grandes modelos de linguagem (LLMs).

2.1. O Processo de Penetration Testing e a Fase de Reconhecimento

O *penetration testing* (pentest) e uma pratica de segurança ofensiva na qual profissionais habilitados simulam ataques controlados contra sistemas com o objetivo de identificar vulnerabilidades antes que atores maliciosos o façam [Heiding et al. 2023]. O processo segue metodologias padronizadas, sendo a mais amplamente adotada a *Penetration Testing Execution Standard* (PTES), que organiza a atividade em cinco fases sequenciais: coleta de inteligencia (*intelligence gathering*), analise de vulnerabilidades, exploração, pós-exploração e geração de relatórios [Geng et al. 2025].

A fase inicial, denominada reconhecimento ou mapeamento de infraestrutura, consiste em reunir o maior volume possível de informações sobre o alvo antes de qualquer tentativa de exploração. Nessa etapa são identificados portas abertas, serviços

ativos, versões de *software*, topologia de rede e possíveis vetores de entrada. Ferramentas amplamente utilizadas nesse estagio incluem *Nmap* para varredura de portas e *Nikto* para enumeração de serviços web [Deng et al. 2024]. A qualidade das informações obtidas durante o reconhecimento impacta diretamente o sucesso das fases subsequentes, tornando essa etapa critica para qualquer estrategia de teste ofensivo [Skandylas and Asplund 2025].

Do ponto de vista operacional, a execução completa de um pentest demanda equipes de três a cinco profissionais especializados e pode se estender por períodos de duas a três semanas, dependendo da complexidade e do escopo do alvo avaliado [Geng et al. 2025]. Esse custo temporal e humano elevado motivou o crescente interesse em solucoes que ampliem o grau de automação do processo, mantendo ou superando a qualidade dos resultados produzidos por especialistas humanos.

Estudos de caso que envolvem ambientes heterogêneos, como residencias conectadas compostas por dispositivos de IoT, demonstraram que mesmo testadores com menor experiencia são capazes de identificar vulnerabilidades criticas quando amparados por metodologias sistemáticas [Heiding et al. 2023]. Esse achado reforça o argumento de que a estruturação formal do processo de pentest, mais do que o conhecimento tácito individual, e o fator determinante para a eficacia do mapeamento de infraestruturas.

2.2. Automação Tradicional Guiada por Humanos (Human-Driven Automation)

A automação tradicional em segurança ofensiva caracteriza-se pela combinação de ferramentas especializadas de linha de comando com decisões tomadas inteiramente por analistas humanos. Nesse paradigma, o especialista define o escopo, seleciona as ferramentas adequadas, interpreta os resultados parciais e determina os próximos passos com base em seu conhecimento prévio sobre as técnicas de ataque e o comportamento esperado dos sistemas avaliados.

Skandylas e Asplund [Skandylas and Asplund 2025] formalizaram o problema de pentest automatizado como um sistema de transição rotulada (*Labeled Transition System*, LTS), no qual estados representam o conhecimento acumulado do testador sobre o alvo e transições correspondem a varreduras ou ataques. Essa modelagem demonstra que, mesmo em ferramentas tradicionais, a seleção de próximas acoes depende de um modulo de raciocínio capaz de avaliar o estado corrente e priorizar opções de acordo com métricas de utilidade. A arquitetura proposta pelos autores, denominada ADAPT, implementa esse raciocínio por meio de um *loop* MAPE-K (*Monitor, Analyze, Plan, Execute*), no qual decisões em tempo de execução são tomadas sem necessidade de interação humana, mas ainda com base em repertórios de ataque e varredura predefinidos por especialistas.

O modelo MAPE-K separa conceitualmente a logica de decisão da execução dos instrumentos ofensivos, permitindo que varreduras e explorações ocorram em paralelo. Os experimentos conduzidos pelos autores sobre as maquinas virtuais Metasploitable2 e Metasploitable3, bem como sobre uma rede virtual utilizada em cursos de *ethical hacking*, demonstraram que o sistema e capaz de comprometer todos os hospedeiros exploráveis sem interação humana, alocando a maior parte do tempo de execução as operações das ferramentas do sistema gerenciado, enquanto o próprio loop de controle consome apenas dezenas de milissegundos por ciclo [Skandylas and Asplund 2025].

A abordagem tradicional apresenta, contudo, limitações estruturais que dificultam

sua generalização. O repertorio de ataques e varreduras precisa ser construído e mantido manualmente por especialistas, o que significa que técnicas de exploração inéditas ou especificas de um determinado alvo exigem atualização explicita do catalogo. Além disso, a seleção de acoes por funções de utilidade estáticas pode se mostrar subótima diante de alvos que exigem raciocínio contextual sobre cadeias de vulnerabilidades ou sobre informações obtidas em fases anteriores do teste [Skandylas and Asplund 2025]. Esses fatores posicionam a automação tradicional como eficiente para cenários bem mapeados, mas potencialmente insuficiente para ambientes complexos ou desconhecidos.

2.3. Large Language Models (LLMs) como Agentes Autonomos

A introdução de LLMs no contexto de segurança ofensiva representa uma mudança de paradigma: em vez de repertórios estáticos de acoes, o agente passa a contar com conhecimento de domínio vasto adquirido durante o pre-treinamento, capacidade de raciocínio contextual e habilidade de interpretar saídas não estruturadas de ferramentas de segurança [Deng et al. 2024].

O trabalho seminal de Deng et al. [Deng et al. 2024] estabeleceu um benchmark abrangente com 13 maquinas de pentest e 182 subtarefas, documentando sistematicamente as capacidades e limitações de modelos como GPT-3.5, GPT-4 e Bard. Os resultados indicaram que LLMs demonstram proficiência em subtarefas especificas, como configuração de ferramentas de varredura, interpretacao de resultados e proposição de próximas acoes. Entretanto, os modelos enfrentaram dificuldades para manter o contexto global do teste ao longo de conversas extensas, o que resultou em estrategias excessivamente focadas na tarefa mais recente e em perda de descobertas anteriores relevantes.

Para mitigar essas limitações, o sistema PentestGPT introduziu uma arquitetura de três módulos interagentes: o modulo de raciocínio, responsável por manter uma arvore de tarefas de pentest (*Pentesting Task Tree*, PTT) em linguagem natural; o modulo de geração, que traduz subtarefas em comandos executáveis; e o modulo de analise, que condensa saídas verbosas de ferramentas antes de reintroduz-las no contexto do LLM [Deng et al. 2024]. Com essa estrutura, o PentestGPT-GPT-4 obteve aumento de 58,6% na taxa de conclusão de subtarefas em relação ao uso direto do GPT-4, e de 228,6% em relação ao GPT-3.5 sem o framework.

Arquiteturas de múltiplos agentes avançaram ainda mais nessa direção. Geng et al. [Geng et al. 2025] propuseram um framework com um *Controller* central que gerencia o fluxo de trabalho entre agentes especializados de reconhecimento, varredura e exploração. O componente controlador avalia o estado de execução de cada fase e decide se os resultados são suficientes para prosseguir ou se a fase deve ser reexecutada, evitando tanto a subexecução quanto o consumo desnecessário de *tokens*. Alem disso, o framework segmenta serviços e exploits em tarefas individuais, atribuindo um agente separado a cada alvo para reduzir a interferencia de informacoes entre diferentes serviços, problema que contribui para alucinacoes em modelos de linguagem. Os experimentos sobre o dataset AI-Pentest-Benchmark demonstraram que a abordagem superou ou igualou o estado da arte em taxa de conclusão de tarefas, consumindo entre 29,42% e 88,83% dos *tokens* utilizados pela baseline VulnBot.

Uma direção complementar e explorada por Wu et al. [Wu et al. 2025b] com o CurriculumPT, que incorpora aprendizado curricular ao pipeline de pentest com LLMs. O

sistema organiza tarefas baseadas em CVEs em níveis de dificuldade crescente (simples, médio e complexo), permitindo que agentes acumulem experiencia de forma progressiva antes de enfrentar exploirts encadeados de alta complexidade. Uma base de conhecimento de experiencias (*Experience Knowledge Base*, EKB) armazena estrategias bem-sucedidas, pares problema-solução e bibliotecas de comandos atômicos para reutilização em tarefas futuras. Os resultados mostram que o CurriculumPT superou as baselines AutoPT, VulnBot e PentestAgent em ate 18 pontos percentuais na taxa de sucesso de exploracao, ao mesmo tempo em que reduziu o tempo médio de execução em 20,6% e o consumo de *tokens* em 25,5%.

Em conjunto, esses trabalhos demonstram que LLMs trazem flexibilidade e capacidade de generalização que sistemas tradicionais de automação não possuem, mas ao custo de maior consumo computacional e de desafios como alucinação, perda de contexto em conversas longas e dificuldades com informações multimodais.

2.4. Métricas de Eficiência em Segurança Ofensiva

A avaliação comparativa entre paradigmas de automação em pentest requer métricas que capturem tanto a eficacia técnica quanto os custos operacionais envolvidos. A literatura recente converge para um conjunto de indicadores que permitem essa comparação de forma quantitativa e reproduzível.

A **taxa de conclusão de tarefas** ou *exploit success rate* (ESR) mede a proporção de objetivos de pentest efetivamente alcançados em relação ao total definido no escopo, seja em termos de subtarefas individuais ou de maquinas completamente comprometidas [Deng et al. 2024, Wu et al. 2025b]. Essa métrica e diretamente comparável entre sistemas distintos quando avaliados sobre os mesmos alvos e e considerada o principal indicador de eficacia em estudos da área.

O **consumo de *tokens*** funciona como proxy do custo computacional e financeiro em sistemas baseados em LLMs, capturando o overhead de raciocínio e de comunicação entre agentes [Geng et al. 2025]. Em sistemas tradicionais, a métrica equivalente e o tempo de CPU consumido pelas ferramentas de automação, que Skandylas e Asplund [Skandylas and Asplund 2025] reportam separadamente para os componentes de varredura, exploração e pós-exploração, mostrando que o loop MAPE-K em si consome apenas dezenas de milissegundos.

O **tempo médio de exploração** (*Average Time to Exploit*, ATE) quantifica a duração total desde o inicio do teste ate o alcance de um objetivo especifico, permitindo avaliar a velocidade operacional de cada abordagem [Wu et al. 2025b]. Essa métrica e particularmente relevante em cenários onde janelas de oportunidade são limitadas, como em exercícios de *red team* com prazo definido.

O **numero médio de passos por tarefa** (*Average Steps per Task*, AST) reflete a eficiência do planejamento do agente, indicando quantas iterações de raciocínio e execução foram necessárias para atingir cada objetivo [Wu et al. 2025b]. Um menor numero de passos sugere estrategias mais diretas e menor desperdício de tentativas infrutíferas.

Em contextos de IoT e ambientes domésticos conectados, Heiding et al. [Heiding et al. 2023] adicionam dimensões qualitativas a esse conjunto, como a **seve-**

ridade das vulnerabilidades descobertas segundo o sistema CVSS e a **cobertura de tipos de ataque**, verificando se o processo foi capaz de identificar classes diversas de falhas. Esses indicadores complementam as métricas quantitativas ao capturar a relevância pratica dos achados para a segurança dos sistemas avaliados.

Por fim, a **taxa de acerto da base de experiencias** (*Experience Knowledge Base Hit Rate*, EHR), introduzida pelo CurriculumPT [Wu et al. 2025b], mede com que frequência o conhecimento previamente armazenado e recuperado e aplicado com sucesso em novas tarefas. Essa métrica e especifica a arquiteturas com memoria persistente e captura a capacidade de transferência de conhecimento entre cenários, atributo ausente nos sistemas de automação tradicional.

A combinação dessas métricas oferece um quadro avaliativo multidimensional adequado para responder a pergunta central desta pesquisa, permitindo comparar as abordagens tradicional e baseada em LLMs ao longo de dimensoes complementares de eficacia, custo e generalidade.

2.5. Heurística e Pensamento Lateral na Segurança Ofensiva

O estudo de Sanders e Rand [Sanders and Rand 2024] investiga sistematicamente os processos cognitivos que diferenciam a análise humana da automação. O artigo demonstra que a segurança ofensiva em cenários complexos depende intrinsecamente da heurística e do pensamento lateral (divergente), permitindo que o profissional junte peças de eventos aparentemente isolados por meio da intuição e da experiência tácita acumulada ao longo da carreira.

A principal vantagem dessa abordagem focada no fator humano é a capacidade incomparável de identificar falhas de lógica de negócio e encadear vulnerabilidades que não possuem assinaturas digitais pré-catalogadas em ferramentas automatizadas. Por outro lado, a desvantagem estrutural desse modelo é a sua baixíssima escalabilidade e o alto custo operacional. O pensamento lateral é exaustivo, consome muito tempo e é altamente dependente da expertise e do estado físico do analista, tornando inviável a sua aplicação de forma manual e isolada em infraestruturas globais gigantescas.

2.6. O Problema da Alucinação e Deriva de Contexto em Agentes LLM

A degradação de performance das Inteligências Artificiais em tarefas prolongadas de segurança é analisada em estudos recentes focados em testes de longa duração e alucinação de dados [Song and Lee 2025, Anonymous 2026]. Esses trabalhos mapeiam o fenômeno técnico conhecido como *Context Rot* (deriva ou degradação de contexto), provando empiricamente que, ao ultrapassar limiares entre 10.000 e 32.000 *tokens*, os agentes LLMs começam a esquecer informações críticas descobertas no início da varredura e passam a inventar ou distorcer parâmetros técnicos.

A vantagem de se utilizar agentes LLMs reside na sua força bruta inicial e na capacidade de processar volumes massivos de *logs* ou códigos em poucos segundos, acelerando exponencialmente o arranque da fase de reconhecimento. Entretanto, as desvantagens são severas quando operam de forma isolada: a alucinação de vulnerabilidades gera falsos positivos que custam tempo das equipes, enquanto a perda de contexto faz com que o agente ignore rotas de ataque óbvias (falsos negativos). Essa limitação arquitetural

reforça o perigo de se conceder autonomia cega à máquina, pois pode atestar como segura uma infraestrutura que ainda está vulnerável.

2.7. Arquiteturas *Human-in-the-loop* (HITL) em Cibersegurança

A integração estruturada entre homem e máquina é o foco das análises sobre o julgamento humano sob estresse e o uso de IA [He et al. 2025]. A literatura propõe e avalia as arquiteturas *Human-in-the-loop* (HITL), nas quais o fluxo de trabalho é desenhado para que a IA realize o mapeamento volumétrico de superfície e, em seguida, paralise a execução para exigir a validação, a correção ou o direcionamento cognitivo do analista humano antes de escalar um ataque simulado.

A grande vantagem da arquitetura HITL é a capacidade de unir o melhor dos dois mundos: a velocidade e a escala implacáveis do processamento do LLM com a precisão, heurística e o entendimento de contexto do analista, mitigando os falsos positivos. Como desvantagem, a implementação desse modelo reduz a velocidade operacional absoluta que uma automação 100% autônoma teria, criando gargalos temporais nas etapas que aguardam aprovação humana. Além disso, se os fluxos não forem bem desenhados, o humano pode desenvolver o "viés de automação", passando apenas a aprovar mecanicamente os *prompts* sem realizar a devida análise crítica.

2.8. A Fricção Cognitiva e a Fadiga de Alertas

O impacto psicológico e operacional da automação irrestrita de varreduras é explorado no framework de Ferrag et al. [Ferrag et al. 2024]. O estudo demonstra como o uso de automações probabilísticas que operam sem o filtro do contexto de negócio do alvo acaba inundando os painéis de segurança com alertas irrelevantes. Isso gera uma intensa fricção cognitiva e desencadeia a Fadiga de Alertas (*Alert Fatigue*), um estado de esgotamento onde o cérebro humano se dessensibiliza diante da avalanche constante de notificações triviais.

A vantagem original de ferramentas que geram esse volume gigantesco de alertas é a premissa da cobertura total: teoricamente, nenhuma porta aberta ou versão desatualizada passa despercebida pelos rastreadores algorítmicos. A desvantagem fatal, contudo, é que esse excesso de "ruído" sobrecarrega as equipes, inviabilizando a análise minuciosa. A fadiga leva os profissionais a ignorarem alertas que são genuinamente críticos no meio dos milhares de falsos positivos, transformando a ferramenta que deveria mapear a infraestrutura no próprio causador do ponto cego e do erro operacional.

3. Trabalhos Relacionados

Esta seção apresenta uma revisão de trabalhos relevantes para o escopo desta pesquisa, destacando suas estratégias metodológicas, vantagens, desvantagens e a forma como se associam ao problema motivador deste estudo: a necessidade de diretrizes de uso híbrido para a fase de reconhecimento em testes de penetração.

[Skandylas and Asplund 2025] propõem a arquitetura ADAPT, focada na automação tradicional sem o uso de modelos de linguagem. Como estratégia, os autores formalizaram o teste de penetração automatizado como um Sistema de Transição Rotulada (LTS) e implementaram um *loop* de controle autônomo baseado no modelo MAPE-K

(Monitor, Analyze, Plan, Execute) para tomar decisões em tempo de execução. A principal vantagem dessa abordagem é a sua execução extremamente rápida, com ciclos de decisão consumindo apenas dezenas de milissegundos, sendo capaz de comprometer hospedeiros vulneráveis conhecidos de forma totalmente autônoma. No entanto, o modelo apresenta a desvantagem de ser inerentemente inflexível, pois o repertório de ataques e varreduras precisa ser construído e updated manualmente por especialistas, falhando em contextos inéditos que exigem raciocínio dinâmico e adaptação. Em fluxo, este trabalho associa-se ao nosso problema ao justificar e demonstrar empiricamente por que a automação tradicional pura, baseada em regras estáticas, é insuficiente para lidar com a complexidade e a dinamicidade do reconhecimento de infraestruturas modernas, exigindo novas abordagens.

Em contrapartida, [Wu et al. 2025b]. exploram a orquestração avançada de Modelos de Linguagem de Grande Escala através do sistema CurriculumPT. A estratégia do *framework* consiste em incorporar o aprendizado curricular ao fluxo de *pentest*, organizando as tarefas de avaliação em níveis de dificuldade crescente e armazenando soluções bem-sucedidas em uma Base de Conhecimento de Experiências (EKB). Como vantagem, a abordagem superou as ferramentas de base tradicionais em até 18 pontos percentuais na taxa de sucesso de exploração, além de reduzir consideravelmente o tempo médio de execução e o consumo de *tokens*. Apesar da eficiência algorítmica, o modelo possui a desvantagem de sofrer com o viés da base de dados de treinamento, carecendo do ”pensamento lateral”humano para extrapolar e inferir cenários de ataque que fujam aos padrões engessados catalogados em CVEs. Esse cenário reforça a associação com o argumento central do nosso trabalho: mesmo arquiteturas avançadas de IA possuem limitações algorítmicas graves de contexto, evidenciando a necessidade de estabelecer diretrizes de colaboração híbrida em vez de delegar total autonomia à máquina.

Adicionalmente, [Deng et al. 2024] introduzem o *framework* PentestGPT, projetado para avaliar e mitigar as limitações de Modelos de Linguagem de Grande Escala na condução de testes de penetração automatizados. Como estratégia, a arquitetura implementa três módulos interagentes: o módulo de raciocínio, responsável por manter uma árvore de tarefas (*Pentesting Task Tree* - PTT) em linguagem natural; o módulo de geração de comandos executáveis; e o módulo de análise, encarregado de mitigar saídas verbosas de ferramentas antes de reintroduzi-las no contexto do modelo. A principal vantagem dessa abordagem é a sua capacidade de raciocínio adaptativo baseado em um *benchmark* quantitativo sistemático, o que resultou em um expressivo aumento de 58,6% na taxa de conclusão de subtarefas em comparação ao uso direto de modelos puros. Por outro lado, o sistema apresenta desvantagens significativas associadas ao alto custo operacional decorrente das constantes chamadas de APIs de modelos comerciais e à persistência de falhas críticas, como alucinações técnicas e perda de contexto em janelas longas de execução. Esse trabalho associa-se diretamente ao problema de pesquisa proposto ao demonstrar como o uso isolado de agentes autônomos, sem uma orquestração ou intervenção contextual com especialistas, ainda esbarra em gargalos que comprometem a integridade global do mapeamento de ativos.

Sob a perspectiva dos riscos comportamentais da automação, [Perry et al. 2023]. investigam o impacto do uso de assistentes de Inteligência Artificial na segurança de sistemas computacionais. A estratégia metodológica envolveu a condução de um estudo

empírico controlado com usuários seniores e estudantes para avaliar a qualidade técnica do código produzido sob o suporte de modelos generativos. Como vantagem, a pesquisa quantifica de forma precisa a influência psicológica da máquina sobre o operador, revelando um expressivo ganho de velocidade e produtividade inicial nas tarefas delegadas à tecnologia. Contudo, a desvantagem crítica identificada pelo estudo é que os usuários sob o auxílio da IA geraram entregas substancialmente menos seguras e desenvolveram um forte viés de superconfiança, acreditando de forma equivocada que os resultados gerados estavam livres de vulnerabilidades. Este trabalho associa-se perfeitamente à nossa motivação ao fornecer a evidência científica para o fenômeno da falsa sensação de segurança corporativa, demonstrando que a ausência de mecanismos metodológicos de dupla validação induz ao erro operacional, um reflexo direto do que ocorre em incidentes reais de mercado como o observado recentemente na Meta.

[Sanders and Rand 2024] realizam uma investigação focada exclusivamente no fator humano dentro de operações de segurança. A estratégia do estudo envolve mapear os processos cognitivos de analistas de segurança da informação, investigando sistematicamente como profissionais utilizam a intuição e o pensamento divergente (lateral) para investigar incidentes e vulnerabilidades. A vantagem evidenciada é a capacidade única e insubstituível do ser humano de identificar falhas de lógica de negócio e encadear vulnerabilidades que não possuem assinaturas digitais ou padrões óbvios mapeáveis por rastreadores automatizados. Contudo, a desvantagem desse modelo exclusivamente manual (*Human-Driven*) é a sua baixíssima escalabilidade, pois o uso contínuo da heurística para varreduras de superfície é exaustivo, consome tempo excessivo e acarreta um alto custo operacional (financeiro e cognitivo) para as organizações. A associação desse estudo ao nosso problema é direta, pois ele é a peça fundamental que embasa a solução *Human-in-the-loop* (HITL) proposta neste trabalho, provando cientificamente o valor da intuição humana no fluxo de varredura como mecanismo indispensável para evitar falsos positivos e a fadiga de alertas.

Iniciando uma abordagem distinta focada na robustez arquitetural, Wu et al. [Wu et al. 2025a] introduzem o sistema AutoPT, que enfrenta as limitações estruturais de memória e a tendência de travamento dos agentes baseados em LLMs convencionais [Wu et al. 2025a]. A estratégia do framework consiste em mapear a atividade de testes de penetração em uma Máquina de Estados de Penetração (PSM) baseada em autômatos de Mealy, decomponendo a tarefa global em estados dedicados de Agent e Rule [Wu et al. 2025a]. A principal vantagem desse modelo é a mitigação do estouro de contexto por meio do isolamento de dados entre as fases de varredura, reconhecimento e exploração, o que elimina a necessidade de manter todo o histórico da sessão de chat e duplica a taxa de sucesso em alvos simples [Wu et al. 2025a]. Como desvantagem estrutural, o AutoPT adota uma estratégia de execução puramente sequencial que sacrifica a flexibilidade do agente em cenários complexos, sendo incapaz de suportar o processamento paralelo ou avaliações de estado concorrentes em ambientes de rede de grande escala [Wu et al. 2025a]. Este trabalho associa-se diretamente ao nosso problema ao demonstrar como a imposição de restrições externas por máquinas de estados melhora a precisão técnica e reduz custos, servindo de linha de base para entender o comportamento de barreiras algorítmicas no mapeamento automatizado.

Por outro lado, expandindo os mecanismos de persistência de intenção e apren-

dizado adaptativo, Dai et al. [Dai et al. 2025] propõem o ecossistema RefPentester, um framework que integra o conhecimento especializado de cibersegurança e loops de reflexão interna [Dai et al. 2025]. A estratégia adotada baseia-se em modelar o processo ofensivo como uma Máquina de Estágios de sete estados acoplada a uma arquitetura de memórias de curto e longo prazo, alimentada por um pipeline RAG hierárquico construído a partir das matrizes MITRE ATT&CK e do OWASP Testing Guide [Dai et al. 2025]. A grande vantagem dessa metodologia reside na sua capacidade de auto-correção iterativa através de um componente Reflector, o qual atribui recompensas verbais e diagnósticos de falha que guiam o Generator a reestruturar comandos técnicos após tentativas infrutíferas [Dai et al. 2025]. No entanto, a abordagem apresenta a desvantagem de depender significativamente de interações com operadores humanos para a inserção de logs de feedback e execução de payloads, o que reduz a velocidade operacional absoluta e introduz potenciais gargalos cognitivos na validação de saídas [Dai et al. 2025]. Esse cenário reforça a associação com o argumento central do nosso estudo ao evidenciar que, embora a engenharia de prompts auto-reflexivos aumente a taxa de transição entre os estágios de reconhecimento e identificação, a autonomia completa permanece condicionada a infraestruturas híbridas e a validações pontuais.

4. Metodologia

Para conduzir a investigação empírica proposta e mitigar as lacunas identificadas na literatura, estabeleceu-se um fluxo metodológico rigoroso estruturado em três fases interdependentes, cujos processos e ramificações operacionais encontram-se sintetizados na Figura 2. O desenho experimental foi arquitetado para isolar as variáveis de desempenho, custo e precisão técnica entre os paradigmas de varredura, garantindo que todas as abordagens sejam submetidas ao exato mesmo escopo de superfície de ataque em ambiente controlado.

4.1. Processo de Levantamento e Filtragem Bibliográfica

Antes da definição do ambiente experimental, a primeira fase desta pesquisa consistiu em uma Revisão da Literatura para compor o referencial teórico e estruturar o estado da arte. Para garantir o rigor científico, o processo de busca, seleção e extração de dados foi estruturado metodologicamente.

O levantamento inicial utilizou *strings* de busca focadas em LLMs, cibersegurança e interação humana, consultando bases acadêmicas consolidadas (IEEE Xplore, ACM Digital Library, Google Scholar e arXiv). Em seguida, aplicaram-se filtros de exclusão por título e resumo, descartando artigos fora do escopo (aplicações em outras áreas) ou duplicados. Por fim, a leitura integral baseou-se em critérios de inclusão estritos, priorizando trabalhos com testes empíricos e métricas quantitativas. O fluxograma desse processo de triagem é apresentado na Figura 1.

4.2. Classificação e Natureza da Pesquisa

Esta investigação caracteriza-se epistemologicamente como uma pesquisa aplicada e de natureza empírica, adotando uma abordagem quanti-qualitativa com objetivos eminentemente experimentais e comparativos. A classificação como pesquisa experimental justifica-se pela manipulação controlada de variáveis independentes — especificamente, o paradigma de orquestração técnica e o grau de autonomia algorítmica (humano isolado, agente puramente probabilístico e arquitetura híbrida cooperativa) —

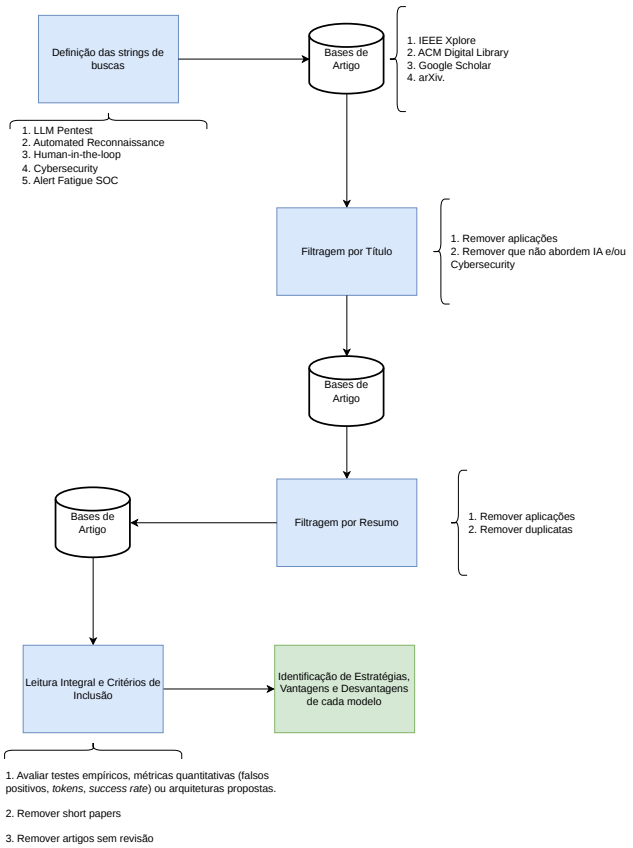

Figura 1. Fluxograma da metodologia de levantamento e filtragem de trabalhos relacionados.

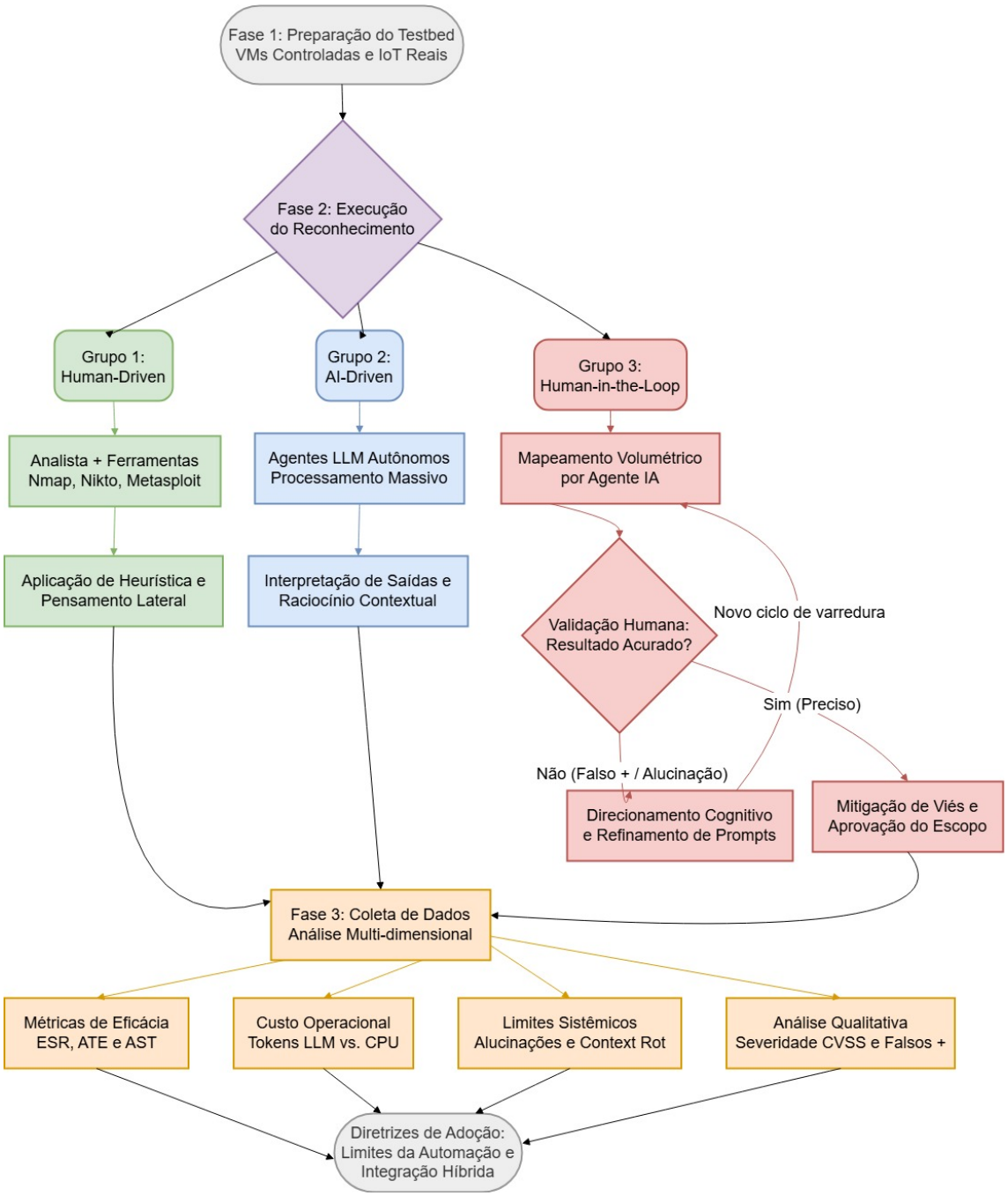

Figura 2. Fluxograma metodológico multi-paradigma: da preparação do *testbed* à consolidação das diretrizes híbridas.

para observar, mensurar e contrastar seus efeitos diretos sobre o conjunto de variáveis dependentes do sistema, representadas pelas métricas de eficácia e custo operacional [Skandylas and Asplund 2025, Wu et al. 2025b].

A vertente quantitativa viabiliza a coleta, a tabulação e o tratamento estatístico de dados de performance direta. Complementarmente, a abordagem qualitativa provê o aporte analítico indispensável para avaliar o nível de severidade das falhas e documentar fenômenos comportamentais e cognitivos gerados pela automação, tais como o surgimento de vieses de superconfiança e a dessensibilização humana provocada pela avalanche de notificações irrelevantes [Perry et al. 2023, Ferrag et al. 2024].

4.3. Fase 1: Preparação do Testbed e Especificação do Cenário

A etapa inaugural compreendeu o provisionamento de um ambiente de testes (*testbed*) heterogêneo, isolado de redes externas para evitar riscos de segurança, mas dotado de alta fidelidade arquitetural para replicar cenários reais do mercado corporativo e residencial. O ambiente foi subdividido em duas frentes analíticas estruturais.

A primeira vertente, focada em **Ambientes Virtuais Controlados (VMs)**, destina-se a avaliar infraestruturas tradicionais de tecnologia da informação. Foram provisionadas instâncias da máquina virtual de segurança ofensiva *Metasploitable 3* (nas variantes Linux e Windows), acopladas a servidores Windows Server configurados para emular uma topologia corporativa típica de *Active Directory* (AD). Este cenário inclui intencionalmente desconfigurações latentes, políticas de senha fracas, e serviços vulneráveis de rede (como versões obsoletas de SMB, RPC e servidores Web) para viabilizar o mapeamento de caminhos de ataque complexos [Chen et al. 2024].

A segunda vertente baseia-se em **Dispositivos IoT Reais**, visando capturar a complexidade descrita por [Heiding et al. 2023]. Integrou-se à rede física um ecossistema ciberfísico composto por dispositivos de automação residencial autênticos. Foram instalados roteadores domésticos operando com *firmwares* legados desatualizados, câmeras de monitoramento IP com credenciais administrativas padronizadas de fábrica (*hardcoded*) e sensores inteligentes baseados em protocolos de comunicação específicos. Essa frente mista impõe aos agentes a necessidade de interpretar respostas não-padronizadas e lidar com limitações de hardware típicas da internet das coisas.

4.4. Fase 2: Execução de Reconhecimento e Protocolos dos Grupos

Uma vez homologado o *testbed*, a execução da fase de reconhecimento e mapeamento volumétrico de superfície foi iniciada de forma paralela e estritamente isolada entre três grupos de controle analítico.

4.4.1. Grupo 1: Abordagem Human-Driven (Análise Tradicional)

Este grupo representa a linha de base (*baseline*) histórica da segurança ofensiva. Profissionais de segurança conduziram rotinas manuais utilizando ferramentas determinísticas consolidadas de linha de comando: *Nmap* para varreduras de portas e identificação de sistemas operacionais, *Nikto* para enumeração detalhada de servidores web, e o console do *Metasploit* para inspeção auxiliar de vulnerabilidades. A condução baseou-se estritamente na aplicação de heurística própria e pensamento lateral divergente

[Sanders and Rand 2024], em que o especialista conecta eventos aparentemente desconexos por intuição prática. Conforme formalizado pela modelagem LTS (*Labeled Transition System*), as transições de estado dependem da avaliação cognitiva individual do analista sobre o estado atual do alvo [Skandylas and Asplund 2025].

4.4.2. Grupo 2: Abordagem AI-Driven (Automação Baseada em LLM)

Para testar os limites práticos da autonomia probabilística cega, este braço operou sem qualquer interação humana. Implementou-se um *framework* multi-agente avançado baseado nos modelos fundacionais de linguagem de grande escala *Claude Sonnet 4.6* e *ChatGPT 5.5*. A arquitetura operacional inspirou-se no conceito de *Pentesting Task Tree* (PTT) proposto pelo *PentestGPT* [Deng et al. 2024]. Neste grupo, os modelos receberam os dados brutos e foram incumbidos de realizar de forma massiva o processamento de logs, a tradução de saídas não estruturadas em linguagem natural, e a proposição autônoma das varreduras subsequentes. Em consonância com as restrições apontadas na literatura recente, o agente autônomo foi rigorosamente monitorado quanto à sua propensão a sofrer de *Context Rot* (degradação da memória em interações longas) e alucinações técnicas que distorcem argumentos de comandos [Song and Lee 2025, Anonymous 2026].

4.4.3. Grupo 3: Abordagem Human-in-the-Loop (Arquitetura Híbrida)

O terceiro grupo materializa a proposta de colaboração simbiótica. O fluxo foi desenhado para segmentar a carga cognitiva de acordo com as forças de cada paradigma. Inicialmente, o agente de IA assume o mapeamento volumétrico de força bruta, processando logs massivos de varredura em alta velocidade. No entanto, o sistema incorpora uma restrição de controle estrita baseada em uma máquina de estados síncrona [Wu et al. 2025a]. A execução entra em uma pausa determinística na qual o analista de segurança atua como o validador de acurácia, ramificando-se em duas ações possíveis a partir do nó de decisão visualizado na Figura 2.

Na primeira hipótese, focada em cenários de falsos positivos ou alucinações, caso o analista identifique que a IA gerou parâmetros inexistentes ou interpretou erroneamente um ruído de rede, ele intervém diretamente. O humano aplica direcionamento cognitivo e refinamento de *prompts* corretivos, forçando o sistema a ignorar o erro e injetando um novo ciclo de varredura limpo, atuando de maneira similar aos módulos autorreflexivos de frameworks de orquestração [Dai et al. 2025]. Na segunda hipótese, atestando a precisão do resultado, o analista homologa o escopo mapeado. Essa validação mútua funciona como um filtro psicológico, mitigando o viés de automação [Perry et al. 2023] e erradicando falsos positivos antes do avanço prático para a exploração de fato.

4.5. Fase 3: Coleta de Dados e Modelo de Análise Multidimensional

Os dados resultantes dos ensaios experimentais foram consolidados sob uma perspectiva multidimensional, estruturada para contrastar quantitativa e qualitativamente as fronteiras de eficiência de cada abordagem através de quatro eixos analíticos.

O eixo de **Métricas de Eficácia** contempla uma avaliação comparativa direta baseada nos indicadores propostos por [Wu et al. 2025b], compreendendo a Taxa de Conclusão de Tarefas (ESR - *Exploit Success Rate*), o Tempo Médio de Exploração (ATE - *Average Time to Exploit*) para cronometrar a velocidade de arranque, e o Número Médio de Passos por Tarefa (AST), que reflete a assertividade do planejamento tático.

Em paralelo, o eixo de **Custo Operacional** realiza o contraste direto entre o impacto econômico das chamadas de API dos LLMs comerciais, quantificado em milhões de *tokens* de entrada e saída consumidos, em detrimento dos custos de infraestrutura local e processamento computacional consumidos pelas ferramentas tradicionais do Grupo 1 [Skandylas and Asplund 2025, Geng et al. 2025].

O eixo de **Limites Sistêmicos** foca no mapeamento quantitativo das falhas arquiteturais inerentes aos modelos de linguagem de grande escala ao longo do tempo, registrando numericamente as taxas de alucinação de dados técnicos e os episódios de perda de descobertas por estouro da janela de contexto de atenção (*Context Rot*) [Song and Lee 2025, Anonymous 2026].

Por fim, a **Análise Qualitativa** propõe a avaliação da severidade intrínseca das vulnerabilidades descobertas contra a matriz de pontuação do *Common Vulnerability Scoring System* (CVSS) e a mensuração do impacto gerado por falsos positivos na incidência de fadiga de alertas nas equipes [Ferrag et al. 2024]. A intersecção crítica destes quatro eixos fornece a fundamentação empírica indispensável para sintetizar as Diretrizes de Adoção.

4.6. Cronograma Esperado e Planejamento Temporal

Para assegurar a viabilidade metodológica e o cumprimento rigoroso dos objetivos propostos dentro do horizonte do projeto, estabeleceu-se um cronograma esperado de atividades distribuído ao longo de doze meses (M1 a M12), conforme detalhado na Tabela 1. Para garantir o perfeito alinhamento estético e impedir o transbordo das margens do padrão SBC, a estrutura matricial foi normalizada proporcionalmente ao limite da coluna de texto.

Tabela 1. Cronograma esperado das macroetapas de desenvolvimento e testagem futura da pesquisa.

Etapas / Atividades Operacionais	M1	M2	M3	M4	M5	M6	M7	M8	M9	M10	M11	M12
1. Revisão Bibliográfica Avançada e Alinhamento Teórico	X	X	X	X								
2. Modelagem Detalhada do Framework PTT Baseado em Agentes			X	X								
3. Configuração, Instalação e Homologação Física do *Testbed*				X	X	X						
4. Execução dos Ensaios Experimentais (Grupos 1, 2 e 3)						X	X	X				
5. Triagem de Logs, Mineração e Análise Multidimensional								X	X	X		
6. Redação Científica do Artigo e Relatórios de Desempenho									X	X	X	X
7. Consolidação das Diretrizes Híbridas e Defesa Final												X

O encadeamento das atividades prevê intencionalmente uma margem de sobreposição planejada entre a finalização da montagem do *testbed* e o início dos ensaios experimentais (M6). Essa folga estratégica justifica-se pela necessidade de realizar testes de sanidade (*smoke tests*) nos scripts de captura automatizada de tráfego e calibrar a comunicação com as APIs dos modelos fundacionais, mitigando o risco de perda de dados e assegurando a estrita reprodutibilidade estatística do ecossistema.

5. Resultados Esperados

A execução da pesquisa deve mostrar diferenças claras de desempenho entre os três grupos avaliados na fase de reconhecimento [Skandylas and Asplund 2025]. No Grupo 1, focado na análise humana tradicional, espera-se alta precisão na identificação de serviços e quase nenhum alerta falso. Isso ocorre porque o especialista usa sua intuição e seu pensamento lateral para entender conexões complexas [Sanders and Rand 2024]. Como exemplo sintético, em uma rede com um servidor *web* mal configurado, espera-se que o humano encontre falhas lógicas de autenticação que ferramentas automáticas ignorariam. No entanto, este grupo deve ser o mais lento e apresentar dificuldades para lidar com grandes volumes de dados. Se o ambiente simular uma rede residencial com dezenas de dispositivos IoT conectados simultaneamente [Heiding et al. 2023], o analista humano não terá tempo suficiente para inspecionar cada equipamento detalhadamente dentro do prazo estabelecido.

Por outro lado, o Grupo 2, formado apenas por agentes autônomos de Inteligência Artificial [Deng et al. 2024], deve ser muito mais rápido na etapa inicial. A IA consegue processar milhares de linhas de *logs* quase instantaneamente. Porém, espera-se que a precisão caia bastante em testes mais longos. Como exemplo sintético prático, ao analisar muitas informações de rede por várias horas, o agente de IA pode perder o contexto da conversa devido à sobrecarga de memória [Anonymous 2026]. Com isso, a máquina passaria a inventar dados, um fenômeno de alucinação técnica [Song and Lee 2025], indicando falsamente que um roteador seguro possui uma vulnerabilidade. Isso confirma que deixar a IA trabalhar totalmente sozinha gera excesso de alertas irrelevantes, cansa a equipe e reduz a confiança no teste [Ferrag et al. 2024].

No Grupo 3, que utiliza o modelo híbrido de colaboração (*Human-in-the-Loop*), espera-se unir o melhor dos dois cenários: a rapidez da máquina e a precisão humana. O sistema fará o processamento pesado, mas usará pausas automáticas baseadas em máquinas de estados [Wu et al. 2025a] para pedir a validação do especialista sempre que houver dúvida. Em um exemplo sintético, imagine que a IA encontre um tráfego de dados confuso em um servidor. Em vez de tentar adivinhar a falha e registrar um alerta falso, ela pausa e chama o operador humano. O analista percebe que é apenas um fluxo normal da rede e ajusta as instruções da máquina [Dai et al. 2025] para que ela ignore aquele padrão. Assim, o erro é evitado antes de se propagar, mitigando também o viés de automação [Perry et al. 2023]. Com esse modelo em equipe, espera-se atingir a maior taxa de sucesso geral, sendo uma abordagem mais rápida que a humana e mais confiável que a totalmente automatizada [Wu et al. 2025b].

6. Conclusão

O crescimento rápido das redes corporativas e o aumento de dispositivos conectados tornaram a tecnologia muito complexa [Chen et al. 2024]. Por isso, é necessário repensar como realizamos os testes de segurança. Este projeto compara três formas de mapear sistemas antes de um ciberataque simulado: o modelo tradicional feito por humanos, a automação total por Inteligência Artificial e o trabalho em equipe entre humano e máquina.

Ao criar um ambiente de testes que imita redes reais, o estudo pretende provar que depender apenas do esforço humano não funciona mais, pois a quantidade de dados

a serem analisados é grande demais. Ao mesmo tempo, a pesquisa busca mostrar que deixar a Inteligência Artificial fazer o mapeamento sozinha também é muito perigoso. A máquina ainda comete erros de interpretação e pode atestar que sistemas vulneráveis estão seguros, gerando uma falsa sensação de segurança [Perry et al. 2023]. Portanto, conclui-se que o futuro da área de segurança não está em substituir o profissional por ferramentas autônomas, mas sim em integrar as suas habilidades [He et al. 2025]. O caminho mais seguro e viável é juntar a enorme velocidade de leitura da máquina com a intuição e o julgamento crítico do especialista. Os resultados esperados por este trabalho pretendem oferecer um guia prático para o mercado, mostrando como investir em tecnologias de automação de forma inteligente para garantir a proteção real das organizações.

Referências

Anonymous (2026). Context relay for long-running penetration-testing agents. *Network and Distributed System Security Symposium (NDSS)*, 33:1–18.

Chen, Z., Kang, F., Xiong, X., and Shu, H. (2024). A survey on penetration path planning in automated penetration testing. *Applied Sciences*, 14(18):8355.

Dai, H., Li, Y., Zhang, Z., and Yan, J. (2025). Refpentester: A knowledge-informed self-reflective penetration testing framework based on large language models. *arXiv preprint arXiv:2505.07089v3*.

Deng, G., Liu, Y., Mayoral-Vilches, V., Liu, P., Li, Y., Xu, Y., Zhang, T., Liu, Y., Pinzger, M., and Rass, S. (2024). PentestGPT: Evaluating and harnessing large language models for automated penetration testing. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 847–864, Philadelphia, PA. USENIX Association.

Ferrag, M. A. et al. (2024). Policy-guided threat hunting: An llm enabled framework with splunk soc triage. *IEEE Access*, 12:45000–45015.

Geng, X., An, N., Xu, B., Yang, X., Jiang, B., Liu, B., and Liu, J. (2025). Controller makes pentesting better: An improved multi-agent automated penetration testing framework. *2025 IEEE 24th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*, page 845–852.

He, J., Treude, C., and Lo, D. (2025). Llm-based multi-agent systems for software engineering: Literature review, vision, and the road ahead. *ACM Transactions on Software Engineering and Methodology*, 34(5):1–30.

Heiding, F., Süren, E., Olegård, J., and Lagerström, R. (2023). Penetration testing of connected households. *Computers I&; Security*, 126:103067.

Perry, N., Srivastava, M., Kumar, D., and Boneh, D. (2023). Do users write more insecure code with ai assistants? *ACM Conference on Computer and Communications Security (CCS)*, 23:2–16.

Sanders, C. and Rand, S. (2024). Creative choices: Developing a theory of divergence, convergence, and intuition in security analysts. *Journal of Cybersecurity*, 10:1–15.

Skandylas, C. and Asplund, M. (2025). Automated penetration testing: Formalization and realization. *Comput. Secur.*, 155(104454):104454.

Song, S. and Lee, D. (2025). Measuring hallucination in large language models for cyber threat intelligence: An exploratory study. *arXiv preprint*, abs/2501.00001:1–12.

Wu, B., Chen, G., Chen, K., Shang, X., Han, J., He, Y., Zhang, W., and Yu, N. (2025a). Autopt: How far are we from the fully automated web penetration testing? *IEEE Transactions on Information Forensics and Security*, 20:9657–9672.

Wu, X., Tian, Y., Chen, Y., Ye, P., Cui, X., Jia, J., Li, S., Liu, J., and Niu, W. (2025b). Curriculumpt: Llm-based multi-agent autonomous penetration testing with curriculum-guided task scheduling. *Applied Sciences*, 15(16):9096.