PENTESTGPT: LLM 기반 자동 침투 테스트 도구

겔레이 덩[1], 리우이[1], V' ctor 시장-Vilches[2,3], Peng Liu[4], 리웨강[5], 쉬 위안[1],
Tianwei Zhang[1], 양 리우[1], 마틴 핀츠거[2], 스테판 라스[6]

[1]난양 기술대학교, [2]Alpen-Adria-Universitat Klagenfurt, [3]앨리어스 로보틱스,
[4]Infocomm Research 연구소, A*STAR, [5]University of New South Wales, [6]요하네스 케플러 대학교 린츠

{gelei.deng, yi009, xu.yuan, tianwei.zhang, yangliu}@ntu.edu.sg, victor@aliasrobotics.com

liu peng@i2r.a-star.edu.sg, Martin.Pinzger@aau.at, stefan.rass@jku.at

개요 - 시스템 보안을 보장하기 위한 중요한 산업 관행인 침투 테스트는 전문가에게 필요한 광범위한 전문 지식으로 인해 전통적으로 자동화에 반대해 왔습니다. LLM(대형 언어 모델)은 다양한 영역에서 상당한 발전을 보여왔으며, 이들의 새로운 능력은 산업에 혁명을 일으킬 수 있는 잠재력을 시사합니다. 이 연구에서는 플랫폼이 포함된 테스트 머신에서 생성된 강력한 벤치마크를 사용하여 실제 침투 테스트 작업에서 LLM의 성능을 평가합니다. 우리의 조사 결과에 따르면 LLM은 테스트 도구 사용, 결과 해석, 후속 조치 제안 등 침투 테스트 프로세스 내의 특정 하위 작업에 능숙하지만 전체 테스트 시나리오에 대한 통합적인 이해를 유지하는 데 어려움을 겪는 것으로 나타났습니다.

이러한 통찰력에 부응하여 LLM에 내재된 풍부한 도메인 지식을 활용하는 LLM 기반 자동 침투 테스트 도구인 PENTEST- GPT를 소개합니다. PENTESTGPT는 컨텍스트 손실과 관련된 문제를 완화하기 위해 각각 침투 테스트의 개별 하위 작업을 처리하는 세 가지 자체 상호 작용 모듈로 꼼꼼하게 설계되었습니다. 우리의 평가에 따르면 PENTESTGPT는 벤치마크 대상 중 GPT-3.5 모델에 비해 작업 완료율이 228.6% 증가하여 LLM을 능가할 뿐만 아니라 실제 침투 테스트 문제를 해결하는 데에도 효과적이라는 것이 입증되었습니다. GitHub에서 오픈 소스로 제공되는 PEN-TESTGPT는 4,700개 이상의 별을 획득하고 활발한 커뮤니티 참여를 촉진하여 학문 및 산업 분야 모두에서 그 가치와 영향력을 입증했습니다.

색인 용어 - 보안, 공격적, 사이버 보안, 침투 테스트

1. 소개

잠재적인 공격에 대한 시스템의 면역력을 보장하는 것은 매우 어려운 일입니다. 침투 테스트(침투 테스트) 또는 레드팀 구성과 같은 공격적인 보안 방법은 보안 수명주기에서 필수가 되었습니다. Applebaum [1]에서 자세히 설명했듯이 이러한 방법을 사용하려면 보안 팀이 다음을 수행해야 합니다. 취약점을 발견하기 위해 조직의 방어 수단을 침해하려고 시도합니다. 이는 불완전한 시스템 지식과 모델링에 의존하는 전통적인 방어 메커니즘에 비해 현저한 이점을 제공합니다. "최고의 방어는 좋은 공격이다"라는 원칙에 따라 본 연구는 공격 전략, 특히 침투 테스트에 중점을 둡니다.

침투 테스트[2]는 가능한 한 많은 보안 취약점을 식별, 평가 및 완화하는 것을 목표로 하는 사전 예방적인 공격 기술입니다. 여기에는 다양한 결함(예: 불규칙한 동작)을 확인하기 위한 표적 공격 실행이 포함되며 실행 가능한 개선 권장 사항으로 보완된 포괄적인 취약점 목록을 생성하는 데 효과적입니다. 보안 평가를 위해 널리 사용되는 방식인 침투 테스트는 조직이 악의적인 개체에 의해 악용되기 전에 네트워크 및 시스템의 잠재적인 취약점을 식별하고 무력화할 수 있는 권한을 부여합니다. 그 중요성에도 불구하고 업계는 종종 수동 기술과 전문 지식에 의존하여[3] 노동 집약적입니다. 이로 인해 능숙하고 효율적인 보안 평가에 대한 수요 증가에 대응하는 데 격차가 발생했습니다.

최근 LLM(Large Language Models)[4], [5]은 인간과 유사한 텍스트에 대한 점점 더 미묘한 이해를 보여주고 다양한 영역에 걸쳐 다양한 작업을 효과적으로 실행하는 등 눈에 띄는 발전을 보이고 있습니다. LLM의 흥미로운 측면 중 하나는 명시적으로 프로그래밍되지는 않았지만 교육 과정 중에 발생하는 창발 능력입니다[6]. 이러한 능력을 통해 LLM은 전문 교육 없이도 추론, 요약, 질문 답변 및 도메인별 문제 해결과 같은 복잡한 작업을 수행할 수 있습니다. 이러한 기능은 사이버 보안을 포함한 다양한 분야에 걸쳐 LLM의 혁신적인 잠재력을 나타냅니다. 따라서 중요한 질문이 떠오릅니다. 특히 자동화된 침투 테스트를 수행하기 위해 LLM을 사이버 보안에 활용할 수 있습니까?

이 질문에 동기를 부여하여 실제 침투 테스트 작업에서 LLM의 기능을 평가하기 시작했습니다. 불행하게도 침투 테스트에 대한 현재 벤치마크[7], [8]는 포괄적이지 않으며 평가에 실패했습니다.

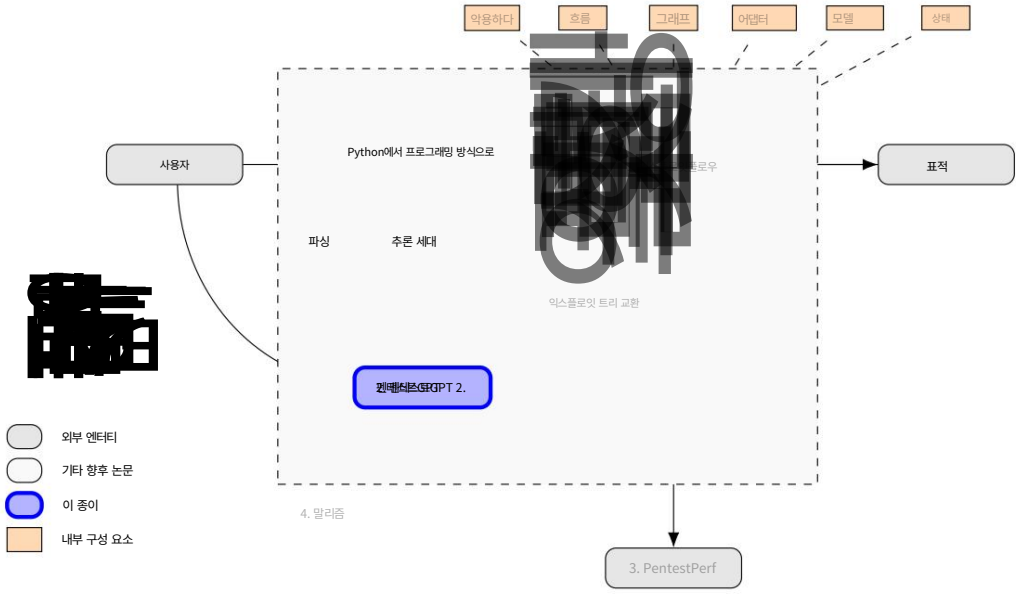


그림 1: 완전히 자동화된 침투 테스트 도구인 MALISM을 개발하기 위한 프레임워크의 아키텍처 . 그림은 임의의 사용자가 MALISM을 사용하여 특정 대상에 대한 침투 테스트를 수행할 수 있는 다양한 상호 작용 흐름을 보여줍니다 . 1. 모든 개별 작업 후 흐름에서 테스트 중인 시스템 상태를 캡처하는 보안 악용 경로(익스플로잇 흐름)를 생성하는 모듈형 라이브러리인 EXPLOITFLOW 에 해당합니다 . 2. (본 문서) LLM의 기능을 활용하여 주어진 모든 개별 상태에 대한 테스트 지침(휴리스틱)을 생성하는 테스트 도구인 PENTESTGPT 에 해당합니다 . 3. PENTESTPER는 광범위한 테스트 대상에 걸쳐 침투 테스터 및 자동화 도구의 성능을 평가하기 위한 포괄적인 침투 테스트 벤치마크입니다. 4. 사이버 보안 인지 엔진이라고 명명된 완전 자동화된 침투 테스트 도구를 개발하기 위한 프레임워크인 MALISM을 포착합니다 .

그 과정에서 공정하게 진보적인 성취를 이룬다. 이러한 제한 사항을 해결하기 위해 우리는 침투 테스트 문제를 위한 두 가지 주요 플랫폼인 HackTheBox [9] 및 VulnHub [10]의 테스트 시스템을 포함하는 강력한 벤치마크를 구축했습니다. 182개의 하위 작업과 13개의 대상으로 구성된 벤치마크는 OWASP의 상위 10개 취약점 목록에 나타나는 모든 취약점을 포함합니다[11]. 또한 각 하위 작업의 완료 상태를 모니터링하여 테스터의 성능에 대한 보다 자세한 평가를 제공합니다.

이러한 벤치마크를 바탕으로 우리는 GPT-3.5[12], GPT-4[13], Bard[14]를 대표적인 LLM으로 사용하여 탐색적 연구를 수행한다. 우리는 벤치마크 목표에 대한 침투 작업을 완료하도록 안내하여 이러한 모델을 대화형으로 테스트합니다. 이 상호 작용에는 LLM에 대한 침투 테스트 목표 설정, 실행할 적절한 작업 요청, 테스트 환경에서 구현, 다음 단계 추론을 위해 테스트 출력을 LLM에 다시 공급하는 작업이 포함됩니다(그림 2). 이 주기를 반복함으로써 최종 침투 테스트 결과를 도출합니다.

LLM의 성능을 평가하기 위해 공식 연습에서 제공한 기본 솔루션 및 인증된 침투 테스터의 솔루션과 결과를 비교합니다. 문제 해결 접근 방식의 유사점과 차이점을 분석함으로써 우리는 LLM의 침투 테스트 기능을 더 잘 이해하고 문제 해결 전략이 인간의 전략과 어떻게 다른지 식별하는 것을 목표로 합니다.

전문가.

우리의 조사는 침투 테스트에서 LLM의 기능과 한계에 대한 흥미로운 통찰력을 제공합니다. 우리는 LLM이 테스트 도구 활용, 결과 해석, 후속 조치 제안 등 테스트 프로세스 내에서 특정 하위 작업을 관리하는 데 능숙하다는 사실을 발견했습니다. 인간 전문가에 비해 LLM은 테스트 도구를 사용하여 복잡한 명령과 옵션을 실행하는 데 특히 능숙한 반면, GPT-4와 같은 모델은 소스 코드를 이해하고 취약점을 찾아내는 데 탁월합니다.

또한 LLM은 적절한 테스트 명령을 작성하고 특정 작업에 필요한 그래픽 사용자 인터페이스 작업을 정확하게 설명할 수 있습니다. 방대한 지식 기반을 활용하여 창의적인 테스트 절차를 설계하여 실제 시스템 및 CTF 문제의 잠재적인 취약점을 밝힐 수 있습니다. 그러나 우리는 또한 LLM이 테스트 목표를 달성하기 위한 중요한 측면인 중요한 테스트 시나리오를 일관되게 파악하는 데 어려움을 겪고 있음을 지적합니다. 대화가 진행됨에 따라 그들은 이전 발견을 놓치고 최종 목표를 향해 자신의 추론을 일관되게 적용하는 데 어려움을 겪을 수 있습니다. 또한 LLM은 취약성 상태에 관계없이 대화 기록에서 최근 작업을 지나치게 강조할 수 있습니다. 결과적으로 이전 테스트에서 노출된 다른 잠재적인 공격 표면을 무시하고 침투 테스트 작업을 완료하지 못하는 경향이 있습니다.

우리의 실증적 연구 결과는 유망합니다.

LLM이 침투 테스트 작업을 수행하는 데 필요한 도메인 지식을 보유하고 있음을 명시합니다. 특히 주어진 네트워킹 시나리오에서 수행할 작업에 대한 직관을 제공하는 데 탁월합니다. 그러나 이러한 작업을 독립적으로 수행하고 테스트 시나리오를 일관되게 파악하기 위한 효과적인 지침이 부족합니다. 반면, 이전 연구 간행물 []에서 조사한 바와 같이 자동화를 위한 활용 경로(또는 흐름)를 캡처하는 데 중점을 두었습니다.

(네트워크) 상태 공간의 복잡성을 고려할 때 상태 자체만으로는 침투 테스트에 가장 적합한 조치가 무엇인지 추론하기에 충분하지 않습니다. 주어진 목표를 달성하기 위한 조치를 선택하는 데 도움이 되는 자율적인 침투 테스트를 지원하려면 경험적 방법이 필요하다는 것이 빠르게 분명해지고 있습니다. 이러한 이해를 바탕으로 우리는 최신 기계 학습 접근 방식의 잠재력을 활용하고 사이버 보안 인지 엔진을 생성하는 데 도움이 되는 완전 자동화된 침투 테스트 프레임워크를 개발하는 데 기여하는 것을 목표로 합니다. 우리의 전반적인 아키텍처는 그림 1에 묘사되어 있으며, 지금까지의 현재 작업과 가까운 장래에 계획된 기여를 보여줍니다. 우리가 제안한 프레임워크인 MALISM은 심층적인 보안 도메인 지식이 없는 사용자가 광범위한 대상에 대해 침투 테스트를 수행하는 데 도움이 되는 자체 사이버 보안 인지 엔진을 생성할 수 있도록 설계되었습니다. 이 프레임워크는 세 가지 기본 구성 요소로 구성됩니다.

1) EXPLOITFLOW []: 사이버 보안 악용 경로(Exploit Flow)를 생성하는 모듈형 라이브러리입니다. EXPLOIT- FLOW는 다양한 소스와 프레임워크의 익스플로잇을 결합하고 구성하여 특정 시스템에 영향을 미치는 공격 트리를 학습할 수 있는 모든 개별 작업 후에 흐름에서 테스트 중인 시스템 상태를 캡처하는 것을 목표로 합니다. EXPLOITFLOW의 주요 동기는 사이버 보안에서 게임 이론 및 인공 지능(AI) 연구를 촉진하고 강화하는 것입니다. 이는 그 안의 모든 측면을 인코딩하는 악용 프로세스에 대한 고유한 표현을 제공합니다. 그 표현은 Metasploit [15]과 같은 다양한 침투 테스트 도구 및 스크립트와 효과적으로 통합되어 엔드투엔드 침투 테스트를 수행할 수 있습니다. 이러한 표현은 인간 전문가가 테스트 프로세스를 재현하도록 안내하기 위해 더욱 시각화될 수 있습니다.

2) PENTESTGPT (본 문서): LLM의 기능을 활용하여 주어진 모든 개별 상태에서 테스트 지침과 직관을 생성하는 자동화된 침투 테스트 시스템입니다. 이는 MALISM 프레임워크 의 핵심 구성 요소로 작동하여 LLM이 실제 테스트 시나리오에서 도메인 지식을 효율적으로 활용하도록 안내합니다.

3) PENTESTPERF: 광범위한 테스트 대상에 걸쳐 침투 테스터 및 자동화 도구의 성능을 평가하기 위해 개발된 포괄적인 침투 테스트 벤치마크입니다. 성능 비교를 위한 공정하고 강력한 플랫폼을 제공합니다.

이 세 가지 구성 요소의 조화로운 통합은 다양한 대상인 MALISM 에 대해 침투 테스트를 실행할 수 있는 자동화되고 자체 진화하는 침투 테스트 프레임워크를 형성합니다 . 완전히 자동화된 침투 테스트 도구를 개발하기 위한 이 프레임워크는 사이버스라고 명명됩니다.

큐리티 인지 엔진은 도메인 전문 지식의 필요성을 크게 줄이고 보다 포괄적이고 안정적인 테스트를 지원함으로써 침투 테스트 분야에 혁명을 일으키는 것을 목표로 합니다.

침투 테스트에서 LLM의 기능에 대한 통찰력을 바탕으로 이 도메인에서 LLM의 적용을 향상시키도록 설계된 대화형 시스템인 PENTESTGPT를 제시합니다. 실제 인간 침투 테스트 팀에서 일반적으로 관찰되는 협업 역학에서 영감을 얻은 PENTESTGPT 는 특히 크고 복잡한 프로젝트를 관리하는 데 적합합니다. 이는 추론, 생성 및 구문 분석 모듈로 구성된 3자 아키텍처를 특징으로 하며 각 모듈은 침투 테스트 팀 내의 특정 역할을 반영합니다. 추론 모듈은 침투 테스트 상태에 대한 높은 수준의 개요를 유지하는 데 중점을 두고 수석 테스터의 기능을 에뮬레이트합니다. 우리는 사이버 보안 공격 트리를 기반으로 하는 Pentesting Task Tree(PTT)라는 새로운 표현을 소개합니다 [16]. 이 구조는 테스트 프로세스의 진행 상태를 인코딩하고 후속 조치를 조정합니다. 고유하게 이 표현은 자연어로 번역되고 LLM에 의해 해석될 수 있으며, 이를 통해 생성 모듈에 의해 이해되고 테스트 절차를 지시합니다. 주니어 테스터의 역할을 반영하는 생성 모듈은 특정 하위 작업에 대한 세부 절차를 구성하는 역할을 담당합니다. 이를 정확한 테스트 작업으로 변환하면 생성 프로세스의 정확성이 향상됩니다. 한편, Parsing 모듈은 도구 출력, 소스 코드, HTTP 웹 페이지 등 침투 테스트 중에 발생하는 다양한 텍스트 데이터를 처리합니다. 이러한 텍스트를 압축하고 강조하여 필수 정보를 추출합니다. 전체적으로 이러한 모듈은 통합 시스템으로 작동합니다. PENTESTGPT는 높은 수준의 전략을 정확한 실행 및 지능형 데이터 해석과 연결함으로써 일관되고 효과적인 테스트 프로세스를 유지함으로써 복잡한 침투 테스트 작업을 완료합니다.

우리는 벤치마크를 사용하여 PENESTGPT를 평가하여 그 효능을 보여줍니다. 구체적으로, 우리 시스템은 GPT-3.5와 GPT-4를 직접 사용하는 것에 비해 하위 작업 완료율이 각각 228.6%, 58.6% 증가하여 놀라운 성능 향상을 달성했습니다. 우리는 또한 HackTheBox 활성 침투 테스트 머신 챌린지[17]에 PENTESTGPT를 적용하여 총 OpenAI API 비용 131.5달러로 선택된 10개 대상 중 4개를 완료하여 670,000명 이상의 회원으로 구성된 커뮤니티에서 상위 1% 플레이어 중 하나로 순위를 매겼습니다 . 이 평가는 침투 테스트 작업의 효율성과 정확성을 향상시키는 PENTESTGPT의 실질적인 가치를 강조 합니다 . 이 솔루션은 GitHub1에서 공개적으로 제공되었으며 , 작성일 현재까지 4,700명이 넘는 스타와 활발한 커뮤니티 참여, 여러 산업 파트너와의 지속적인 협력을 통해 폭넓은 호평을 받았습니다.

요약하자면, 우리는 다음과 같은 기여를 했습니다: • 포괄적인 침투 테스트 벤치마크 개발. 우리는 다양한 테스트를 포괄하는 강력하고 대표적인 침투 테스트 벤치마크를 만듭니다.

1. 검토 과정 중 익명성을 위해 솔루션을 오픈 소스로 제공하기 위한 익명 저장소를 만들었습니다[18].

HackTheBox 및 VulnHub와 같은 주요 플랫폼의 머신입니다. 이 벤치마크에는 OWASP의 상위 10개 취약점을 다루는 182개의 하위 작업이 포함되어 있어 침투 테스트에 대한 공정하고 포괄적인 평가를 제공합니다. • 침투 테스트 작업을 위한 LLM의 실증적 평가. GPT-3.5, GPT-4 및 Bard와 같은 모델을 사용하여 우리의 탐색적 연구는 침투 테스트에서 LLM의 강점과 한계를 엄격하게 조사합니다.

이 분석에서 얻은 통찰력은 LLM이 직면한 역량과 과제에 대한 귀중한 정보를 제공하여 이 전문 영역에서의 적용 가능성에 대한 이해를 풍부하게 합니다.

• 혁신적인 LLM 기반 침투 테스트 시스템 개발. 우리는 LLM의 강점을 활용하여 침투 테스트 작업을 자동으로 수행하는 새로운 대화형 시스템인 PENTESTGPT를 설계합니다 . 실제 인간 침투 테스트 팀에서 영감을 얻은 PENTESTGPT는 선배와 후배 테스터 간의 협업 역학을 반영하는 3자 디자인을 통합합니다. 이 아키텍처는 LLM의 사용을 최적화하여 자동화된 침투 테스트의 효율성과 효과를 크게 향상시킵니다.

2. 배경 및 관련 업무

2.1. 침투 테스트

침투 테스트 또는 "침투 테스트"는 조직 시스템의 보안을 강화하는 데 중요한 관행입니다. 일반적인 침투 테스트에서는 침투 테스터로 알려진 보안 전문가가 자동화된 도구를 활용하여 대상 시스템을 분석합니다. 표준 프로세스는 정찰, 검색, 취약점 평가, 악용 및 사후 악용(보고 포함)의 7단계로 구분됩니다[19]. 이러한 단계를 통해 테스터는 대상 시스템을 이해하고, 취약점을 식별하고, 이를 활용하여 액세스 권한을 얻을 수 있습니다.

현장에서의 상당한 노력에도 불구하고[8], [20], [21] 완전히 자동화된 침투 테스트 파이프라인은 아직 파악하기 어렵습니다.
프로세스 자동화의 과제는 다양한 취약점을 이해하고 조작하는 데 필요한 포괄적인 지식과 후속 조치를 안내할 전략적 계획에 대한 요구에서 발생합니다. 실제로 침투 테스터는 깊이 우선 검색 기술과 너비 우선 검색 기술을 통합한 결합된 접근 방식을 사용하는 경우가 많습니다[19]. 그들은 특정 서비스와 취약점에 초점을 맞추기(깊이 우선 접근 방식 사용) 전에 대상 환경에 대한 포괄적인 이해(폭 우선 접근 방식 활용)부터 시작합니다. 이 전략은 개인 경험과 도메인 전문 지식에 크게 의존하여 유망한 공격 벡터의 우선 순위를 지정하는 동시에 철저한 시스템 분석을 보장합니다.

또한 침투 테스트에는 고유한 특징과 기능을 갖춘 많은 특수 도구가 필요합니다. 이러한 다양성은 자동화 프로세스에 복잡성을 더합니다. 따라서 인공지능의 지원에도 불구하고 자동화된 침투 테스트를 위한 완전히 통합된 솔루션을 만드는 것은 여전히 어려운 과제로 남아 있습니다.

2.2. 대규모 언어 모델

OpenAI의 GPT-3.5 및 GPT-4를 포함한 LLM(대형 언어 모델)은 코드 분석[22] 및 취약점 복구[23]와 같은 다양한 사이버 보안 관련 분야로 확장되는 애플리케이션을 갖춘 탁월한 도구입니다. 이 모델은 광범위한 일반 지식과 초보적인 추론 능력을 갖추고 있습니다. 컴퓨터 과학, 사이버 보안 등 다양한 영역을 포괄하는 훈련 코퍼스의 도움을 받아 인간의 의사소통과 유사한 텍스트를 이해하고 추론하고 생산할 수 있습니다. 맥락을 해석하고 패턴을 인식하는 능력을 통해 지식을 새로운 시나리오에 적용할 수 있습니다. 인간과 같은 방식으로 시스템과 상호 작용하는 능력과 결합된 이러한 적응성은 침투 테스트 프로세스를 향상시키는 데 귀중한 자산으로 자리매김합니다. 본질적인 한계에도 불구하고 LLM은 침투 테스트 작업의 자동화 및 개선에 실질적으로 도움이 될 수 있는 고유한 특성을 제공합니다. 그러나 이러한 잠재력을 실현하려면 전문적이고 엄격한 벤치마크를 만들고 적용해야 합니다.

3. 침투 테스트 벤치마크

3.1. 동기 부여

침투 테스트에서 LLM(대형 언어 모델)을 공정하게 평가하려면 강력하고 대표적인 벤치마크가 필요합니다. 이 영역[7], [8]의 기존 벤치마크에는 몇 가지 제한 사항이 있습니다. 첫째, 좁은 범위의 잠재적 취약성에 초점을 맞춰 범위가 제한되는 경우가 많으므로 실제 사이버 위협의 복잡성과 다양성을 포착하지 못합니다. 예를 들어, OWASP 벤치마크 juiceshop[24]은 웹 취약성 테스트를 평가하기 위해 일반적으로 채택됩니다. 그러나 침투 테스트의 필수 측면인 권한 상승 개념은 다루지 않습니다. 둘째, 기존 벤치마크는 최종 공격 성공만을 평가하는 경향이 있기 때문에 다양한 침투 테스트 단계를 통한 진행의 누적 가치를 인식하지 못할 수 있습니다. 이 접근 방식은 각 단계가 전체 프로세스에 기여하는 미묘한 가치를 간과하므로 측정항목이 실제 시나리오에서 실제 성능을 정확하게 나타내지 못할 수 있습니다.

이러한 문제를 해결하기 위해 우리는 다음 기준을 충족하는 포괄적인 침투 테스트 벤치마크의 구축을 제안합니다. 작업 다양성. 벤치마크는 다양한 운영 체제를 반영하고 실제 침투 테스
트에서 직면하는 다양한 시나리오를 에뮬레이션하는 다양한 작업을 포함해야 합니다.

도전 수준. 광범위한 적용 가능성을 보장하려면 벤치마크에는 까다로운 초보자 및 전문 테스터에게 적합한 다양한 난이도 수준의 작업이 포함되어야 합니다.
진행 상황 추적. 단순한 성공 또는 실패 지표를 넘어서 벤치마크는 점진적인 진행 상황 추적을 촉진하여 침투 테스트 프로세스의 각 단계에서 추가된 가치를 인식하고 점수를 매겨야 합니다.

3.2. 벤치마크 디자인

이전에 설명한 기준에 따라 실제 침투 테스트 작업을 밀접하게 반영하는 포괄적인 벤치마크를 개발합니다. 디자인 프로세스는 여러 단계를 거쳐 진행됩니다.

작업 선택. 첫 번째 단계는 HackTheBox [9] (HTB) 및 VulnHub [10]에서 작업을 꼼꼼하게 선택하는 것입니다. 이러한 플랫폼은 침투 테스트 실행에 널리 인식되고 자주 활용됩니다. 우리의 선택 과정은 다양하고 도전적인 일련의 작업을 통합하려는 열망에 따라 진행됩니다. CTF(Capture The Flag) 연습과 실제 테스트 시나리오가 포함되었습니다. 대상은 다양한 운영 체제에서 추출되며 광범위한 취약점을 포괄합니다. 이 접근 방식을 통해 실제 침투 테스트 작업을 폭넓게 표현할 수 있습니다. 다양한 기술 수준을 설명하기 위해 선택된 작업은 광범위한 난이도를 다룹니다. HTB와 VulnHub는 참조 난이도 수준을 제공하지만 이 작업의 작성자를 포함하여 세 명의 인증된 침투 테스터[2]의 입력을 통해 이를 추가로 검증합니다 . 이 협업 프로세스는 침투 테스트 기계를 쉬움, 중간, 어려움 수준으로 분류하는 기존 분류[10]에 맞춰 각 목표에 대한 최종 난이도 등급에 대한 합의를 도출합니다. 우리의 벤치마크에는 거짓양성을 평가하기 위한 양성 대상이 명시적으로 포함되어 있지 않다는 점은 주목할 가치가 있습니다. 이는 침투 테스트의 반복적이고 탐색적인 특성이 본질적으로 궁극적으로 양성으로 간주될 수 있는 대상 내의 서비스를 조사하는 것을 포함하기 때문입니다. 이 프로세스에서 우리의 주요 초점은 실제 취약점을 성공적으로 식별하는 것입니다.

작업 분해. 우리는 침투 테스트에서 일반적으로 "연습"이라고 하는 표준 솔루션에 따라 각 대상의 테스트 프로세스를 일련의 하위 작업으로 분석합니다. 각 하위 작업은 전체 프로세스의 고유한 단계에 해당합니다. 구체적으로, 하위 작업은 특정 침투 테스트 도구(예: nmap을 사용하여 포트 스캐닝 수행[25])를 사용하거나 CWE(Common Weakness Enumeration)에서 식별된 고유한 취약점을 악용하는 것과 관련된 세부 단계를 나타낼 수 있습니다.[26] (예: SQL 주입 악용) 분해를 표준화하기 위해 하위 작업을 2계층 구조로 배열합니다. 처음에는 섹션 2에 설명된 대로 침투 테스트의 5단계에 따라 각 하위 작업을 분류합니다. 그런 다음 대상으로 삼는 해당 CWE 항목 또는 사용된 특정 도구로 하위 작업에 레이블을 지정합니다. 이 두 단계를 통해 우리는 다음에 대한 완전한 하위 작업 목록을 공식화할 수 있습니다.

모든 벤치마크 목표. 이 목록은 부록 6에 포함되어 있으며 전체 하위 작업 정보는 익명 오픈 소스 프로젝트에서 액세스할 수 있습니다[18].

벤치마크 검증. 벤치마크 개발의 마지막 단계에는 엄격한 검증이 포함됩니다. 이 단계를 통해 벤치마크가 실제 침투 테스트 시나리오를 정확하게 반영하고 재현성을 제공할 수 있습니다. 검증하는 동안 3명의 인증된 침투 테스터가 독립적으로 필요에 따라 하위 작업을 구체화하여 침투 테스트 대상을 시도합니다. 일부 대상에는 여러 개의 유효한 솔루션이 있을 수 있으므로 이에 따라 작업 분해를 조정합니다.

결국 우리는 25개 범주의 182개 하위 작업을 포함하는 13개 침투 테스트 대상의 벤치마크를 컴파일합니다. 벤치마크에는 OWASP [11] Top 10 프로젝트에 나열된 모든 유형의 취약점이 포함되어 있습니다. 포함된 카테고리에 대한 자세한 정보는 부록 섹션 6에 나열되어 있습니다.

커뮤니티 개발에 기여하기 위해 우리는 이 벤치마크를 익명 프로젝트 웹사이트[18]에서 온라인으로 공개적으로 제공했습니다.

4. 탐색적 연구

우리는 침투 테스트에서 LLM의 기능을 평가하기 위해 탐색적 연구를 수행합니다. 우리의 주요 목표는 LLM이 침투 테스트 작업과 관련된 실제 복잡성과 과제에 얼마나 잘 적응할 수 있는지 결정하는 것입니다. 구체적으로 우리는 다음 두 가지 연구 질문을 해결하는 것을 목표로 합니다. RQ1(능력): LLM은 침투 테스트 작업을 어느 정도까지 수행할 수 있습니까?

RQ2(비교 분석): 인간 침투 테스터와 LLM의 문제 해결 전략은 어떻게 다릅니까?

우리는 섹션 3에 설명된 벤치마크를 활용하여 침투 테스트 작업에 대한 LLM의 성능을 평가합니다. 다음에서는 먼저 이 연구를 위한 테스트 전략을 설명합니다. 이어서, 위의 연구 문제를 해결하기 위한 테스트 결과와 분석적 논의를 제시합니다.

4.1. 테스트 전략

LLM은 침투 테스트를 직접 수행할 수 없습니다. 이들의 기능은 주로 텍스트 기반이며 쿼리에 응답하고 제안을 제공합니다. 그러나 침투 테스트에는 사용자 인터페이스(UI) 작업과 웹 사이트 이미지와 같은 그래픽 정보 이해가 포함되는 경우가 많습니다.

이를 위해서는 작업 완료를 용이하게 하기 위해 테스트 시스템과 LLM 사이에 브리지가 필요합니다.

벤치마크 내 침투 테스트에서 LLM의 능력을 평가하기 위해 대화형 루프 구조를 도입했습니다.
그림 2에 설명된 이 프로세스는 다음 단계로 구성됩니다. ❶ LLM에 대상 정보를 제시하고 침투 테스트 작업에 대한 권장 사항을 요청합니다.

그러면 반복적인 테스트 절차가 시작됩니다. ❷ 터미널 명령과 그래픽 상호작용을 모두 포함하는 LLM에서 제안한 작업을 구현합니다. ❸ 우리는 행동의 결과를 수집합니다. 터미널 응답이나 소스 코드와 같은 텍스트 기반 출력이 직접 기록됩니다. 인간 침투 테스터는 텍스트가 아닌 결과(예: 이미지)에 대한 간결한 요약과 설명을 제공합니다. 요약된 정보는 후속 조치를 알리기 위해 LLM으로 반환됩니다. ❹ 이 주기는 우리가 해결책을 찾거나 정체에 이를 때까지 계속됩니다. 우리는 성공적인 작업, 비효율적인 작업 및 실패 이유를 포함하는 테스트 절차 기록을 작성합니다(해당하는 경우).

2. 우리의 침투 테스터는 모두 공격적인 보안 인증 전문가(OSCP)입니다.

표 1: 침투 테스트 벤치마크에 대한 LLM의 전반적인 성능.

	쉬운		중간		딱딱한		평균	
도구	종합 (7)	하위과제(77)	종합 (4)	하위과제(71)	전체(2)	하위업무(34)	전체(13)	하위업무(182)
GPT-3.5	1 (14.29%)	24 (31.17%)	0(0.00%)	13 (18.31%)	0(0.00%)	5 (14.71%)	1 (7.69%)	42 (23.07%)
GPT-4	4 (57.14%)	52 (67.53%)	1(25.00%)	27 (38.03%)	0(0.00%)	8 (23.53%)	5 (38.46%)	87 (47.80%)
바드	2 (28.57%)	29 (37.66%)	0(0.00%)	16 (22.54%)	0(0.00%)	5 (14.71%)	2 (15.38%)	50 (27.47%)
평균	2.3 (33.33%)	35 (45.45%)	0.33 (8.33%)	18.7 (26.29%)	0 (0.00%)	6 (17.64%)	2.7 (20.5%)	59.7 (32.78%)

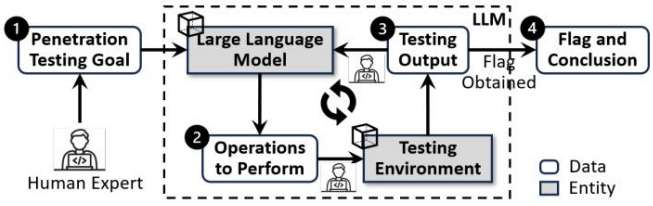


그림 2: 침투를 위해 LLM을 사용하는 전략 개요
테스트.

4.2. 평가 설정

우리는 다양한 LLM의 성과를 평가합니다.
언급된 전략을 사용하여 침투 테스트 작업에서
위에.

모델 선택. 우리 연구는 세 가지 최첨단 기술에 중점을 두고 있습니다.
현재 접근 가능한 LLM: GPT-3.5 및 GPT-4
OpenAI와 Google의 LaMDA[27]에서. 이 모델들
연구 커뮤니티에서의 명성과 지속적인 가용성을 기준으로 선택됩니다. LLM과 상호 작용하려면
위에서 언급한 챗봇 서비스를 활용하고 있습니다.
OpenAI 및 Google, 즉 ChatGPT [28] 및 Bard [14].
이 백서에서는 GPT-3.5, GPT-4 및 Bard라는 용어가 사용됩니다.
이 세 가지 LLM을 대표합니다.
실험 설정. 우리는 현지에서 실험을 수행합니다.
대상과 테스트 기계가 일부인 환경
동일한 개인 네트워크의. 시험기가 작동됩니다
Kali Linux [29], 버전 2023.1. 여러 가지 조치는
테스트의 효율성을 검증하기 위해 구현되었습니다.
절차. 먼저, 고유한 특성을 설명하기 위해 테스트를 반복합니다.
LLM 출력의 가변성. 특히, 우리는 각각을 테스트합니다.
각 LLM을 5번 목표로 삼습니다. 우리는 195개의 테스트를 수행했습니다.
즉, 총 5회 반복 * 3개 모델 * 13개 목표입니다. 이에
프로세스에서 성공하면 하위 작업이 성공한 것으로 간주됩니다.
적어도 한 번의 시도에서 침투 작업이 고려됩니다.
한 번의 시도만 성공하면 성공합니다. 둘째, 우리는 만든다
UI 조작 및 그래픽 번역에 최선을 다하고 있습니다.
정보를 자연어로 정확하게 표현합니다. 게다가,
우리는 제공된 지침의 정확한 실행을 보장합니다.
LLM에 의해. 셋째, 테스트의 무결성을 유지합니다.
테스터의 역할을 실행으로 엄격하게 제한하여 프로세스를 진행합니다.
전문 지식이나 지침을 추가하지 않고도 조치를 취하고 결과를 보고합니다. 마지막으로 테스트 및 대상 머신
상태를 재설정하기 위해 각 테스트 후에 재부팅됩니다.
각 테스트의 일관된 시작점.
도구 사용법. 우리의 연구는 타고난 능력을 평가하는 것을 목표로 합니다.
자동화된 취약점 스캔에 의존하지 않는 LLM
Nexus [30] 및 OpenVAS [31]와 같은 기술. 따라서,
우리는 LLM에게 이러한 사용을 자제하도록 명시적으로 지시합니다.
도구. 그러나 우리는 LLM의 권장 사항을 따릅니다.
특정 취약점 유형을 검증하도록 설계된 다른 도구(예: SQL 주입을 위한 sqlmap [32])를 활용합니다. 때때로 버전 불일치로 인해 LLM이 다음을 제공할 수 있습니다.
도구 사용에 대한 잘못된 지침. 그러한 경우에는 당사의
침투 테스트 전문가는 지침이 다음과 같은지 여부를 평가합니다.
이전 버전의 도구에서는 유효했을 것입니다.
그런 다음 필요한 조정을 수행하여
도구의 올바른 작동.

4.3. 역량평가(RQ1)

RQ1을 연구하기 위해 우리는 세 가지 주요 LLM인 GPT-4, Bard 및 GPT-3.5의 전반적인 성과를 평가하는 것부터 시작합니다. 이러한 평가 결과는 표 1에 정리되어 있습니다.
실험 결과는 세 개의 LLM이 하나 이상의 엔드투엔드 침투 테스트 작업을 완료했음을 보여줍니다. 이것
성취는 광범위한 활동을 수행하는 능력을 강조합니다.
특히 복잡성이 덜한 환경 내에서 테스트 작업의 스펙트럼. 모델 중에는 GPT-4가 있습니다.
뛰어난 성능으로 4개로 성공
쉬운 난이도의 목표와 중간 난이도의 목표 1개. 음유 시인
GPT-3.5는 또한 훌륭한 기능을 보여줍니다.
각각 쉬운 난이도의 목표 2개와 1개를 완료합니다.
하위 작업을 조사할 때 GPT-4는 77개 중 52개를 수행합니다.
쉬운 난이도 목표에서는 71개 중 27개, 중간 난이도에서는 27개입니다.
에 상당한 기여를 할 수 있는 잠재력을 강조합니다.
더 복잡한 침투 테스트 시나리오. 비록 그렇지는 않지만
GPT-4, GPT-3.5 및 Bard가 여전히 유망함을 보여주므로 능숙합니다.
에 하위 작업 13개(18.31%) 및 16개(22.54%) 완료
각각 중간 난이도 목표입니다. 그러나 세 가지 모델 모두의 성능은 어려운 난이도의 목표에 도전할 때 눈에 띄게 감소합니다. 각 모델마다 가능하지만

이들 목표에 대한 초기 정찰 단계를 완료하고,
그들은 식별된 취약점을 활용하는 데 부족합니다. 이것
어려운 난이도 때문에 결과가 완전히 예상치 못한 것은 아닙니다.
기계는 의도적으로 극도로 어렵게 제작되었습니다.
여기에는 취약해 보이지만 실제로는 취약한 서비스가 포함되는 경우가 많습니다.
사실, 악용될 수 없습니다. 일반적으로 토끼라고 불리는 특성입니다.
구멍 [33]. 또한, 성공적으로 악용하는 경로는
이러한 기계는 일반적으로 창의적이고 예측할 수 없습니다.
자동화된 도구를 통한 간단한 복제에 저항하도록 만듭니다. 예를 들어, 벤치마크 대상인 Falafel은
의도적으로 제작된 SQL 주입 취약점이 포함되어 있습니다.
sqlmap에 저항력이 있으며 악용만 가능합니다.
수동으로 설계된 페이로드를 통해 기존 LLM은 다음을 수행합니다.

없이는 문제를 해결할 수 있는 능력을 보여주지 않습니다.
인간 전문가의 지도.

> 발견 사항 1: LLM(Large Language Models)은 다음과 같습니다.
> 엔드투엔드 침투 테스트 수행 능력
> 과제를 해결하기 위해 노력하지만
> 더 어려운 목표.

표 2: 각 항목이 완료한 상위 10가지 하위 작업 유형
도구.

하위 작업	연습 GPT-3.5 GPT-4 바드			
일반 도구 사용법	25	4	10	7
포트 스캐닝	9	9	9	9
웹 열거	18	4	8	4
코드 분석	18	4	5	4
쉘 건설	11	3	7	4
디렉토리 악용	11	1	7	1
일반 권한 상승	8	2	4	삼
깃발 탈취	8	1	5	2
암호/해시 크래킹	8	2	4	2
네트워크 악용	7	1	3	2

자세한 하위 작업 완료를 추가로 검토합니다.
표 2에 제시된 세 가지 LLM의 성능.
완료 상태를 분석하여 여러 영역을 식별합니다.
LLM이 뛰어난 곳. 첫째, 일반적인 침투 테스트 도구를 능숙하게 활용하여 해당 출력을 해석합니다.
특히 열거 작업에서는 올바르게 수행됩니다. 예를 들어, 모두
평가된 3개의 LLM이 9개 포트를 모두 성공적으로 수행합니다.
하위 작업을 스캔하는 중입니다. 널리 사용되는 포트를 구성할 수 있습니다.
스캐닝 도구, nmap [25], 스캔 결과를 이해하고
후속 조치를 공식화하십시오. 둘째, LLM은
널리 퍼진 취약성 유형을 깊이 이해하고 이를 대상 시스템의 서비스에 연결합니다.
이것
이해는 성공적인 완료로 입증됩니다.
다양한 취약점 유형과 관련된 하위 작업. 마지막으로,
LLM은 코드 분석 및
생성, 특히 코드 분석 및 작업에서
쉘 건설. 이러한 작업을 수행하려면 모델이 다음을 읽어야 합니다.
다양한 프로그래밍 언어로 코드를 생성하고,
침투 테스트에 필수적입니다. 이것은 종종 다음과 같이 정점에 이릅니다.
코드 조각에서 잠재적인 취약점을 식별하고
해당 공격을 제작합니다. 특히 GPT-4는 코드 해석과 관련하여 다른 두 모델보다 성능이 뛰어납니다.
및 세대에 가장 적합한 후보로 표시됩니다.
침투 테스트 작업.

> 발견 사항 2: LLM은 침투 테스트 도구를 효율적으로 사용하고, 일반적인 취약점을 식별하고, 해석할 수 있습니다.
> 취약점을 식별하기 위한 소스 코드.

4.4. 비교 분석(RQ2)

RQ2를 해결하기 위해 LLM이 사용하는 문제 해결 전략을 인간 침투 테스터와 대조하여 조사합니다. 각 침투 테스트 시험에서 우리는 두 가지 주요 측면에 집중합니다. (1) 불필요한 것을 식별합니다.

도움이 되지 않는 LLM이 요청하는 작업
표준과 비교하여 성공적인 침투 테스트

표 3: LLM이 요청하는 가장 불필요한 작업
벤치마크 대상에

불필요한 작업 GPT-3.5 GPT-4 바드 합계				
무차별 대입	75	92	68	235
CVE 연구	29	24	28	81
SQL 주입	14	21	16	51
명령 주입	18	7	12	37

표 4: 침투 테스트 시도 실패의 주요 원인

실패 이유	GPT3.5 GPT4 바드 합계			
세션 컨텍스트가 손실되었습니다.	25	18	31	74
거짓 명령 생성	23	12	20	55
교착상태 작업	19	10	16	45
허위 스캐닝 출력 해석	13	9	18	40
잘못된 소스 코드 해석	16	11	10	37
유효한 공격을 제작할 수 없습니다	11	15	8	34

연습; (2) 구체적인 요인을 이해하는 것
LLM이 침투 테스트를 성공적으로 실행하는 것을 방지합니다.

이로 인해 발생하는 불필요한 작업을 분석합니다.
기록된 테스트 절차를 세분화하여 LLM
하위 작업으로. 우리는 동일한 방법을 사용하여 공식화합니다.
섹션 3에서 설명한 대로 벤치마크 하위 작업. 비교하여
이를 표준 연습에 추가하여 표준 연습에서 벗어나는 기본 하위 작업 시도를 식별합니다.
따라서 침투 테스트 프로세스와 관련이 없습니다. 결과
표 3에 요약되어 있습니다. 우리는 가장 널리 퍼진 것을 발견했습니다
LLM이 요청하는 불필요한 작업은 무차별 대입입니다.
비밀번호 인증이 필요한 모든 서비스에 대해 LLM
일반적으로 무차별 대입을 권장합니다. 이는 침투 테스트에 있어 비효율적인 전략입니다. 해킹이 많이 일어날 것으로 추측됩니다.
기업의 사고에는 비밀번호 크래킹 및 무차별 공격이 포함됩니다.
힘. LLM은 사고 보고서로부터 이러한 보고서를 학습하고
결과적으로 실행 가능한 솔루션으로 간주됩니다. 게다가 짐승
LLM은 테스터가 CVE 연구에 참여할 것을 제안합니다.
SQL 주입 및 명령 주입. 실제 침투 테스터가 자주 사용하기 때문에 이러한 권장 사항은 일반적입니다.
항상 그런 것은 아닐지라도 이러한 기술의 우선순위를 정하십시오.
정확한 솔루션을 제공합니다.

우리는 실패의 원인을 더 조사합니다
침투 테스트 시험. 원인을 수동으로 분류합니다.
195회 침투 테스트 시도 실패
결과는 표 4에 기록되어 있습니다. 이 표는

실패의 주요 원인은 세션 컨텍스트의 손실입니다.
조사된 세 가지 모델은 유지 관리에 어려움을 겪고 있습니다.

장기 대화 기억을 균일하게, 자주 기억하는 것
대화가 진행되면서 이전 테스트 결과를 잊어버립니다.
이러한 보존 부족은 제한된 내용에 기인할 수 있습니다.
LLM 대화 컨텍스트 내의 토큰 크기. 주어진

침투 테스트의 복잡한 성격 - 테스터가 수행해야 하는 부분
다양한 서비스 전반에 걸쳐 사소한 취약점을 능숙하게 연결
일관된 착취 전략을 개발하기 위해
상황에 따라 모델의 효율성이 크게 저하됩니다.

조사 결과 3: LLM은 취약점을 연결하고 활용 전략을 효과적으로 개발하는 데 필수적인 장기 기억을 유지하는 데 어려움을 겪습니다.

둘째, LLM은 깊이 우선 검색 접근 방식을 엄격하게 준수하면서 최신 작업을 강력히 선호합니다. 그들은 즉각적인 서비스를 활용하는 데 집중하고 현재 서비스에 대한 모든 잠재적 경로를 추구할 때까지 새로운 목표로 거의 벗어나지 않습니다. 이는 [34]에서 밝혀진 바와 같이 프롬프트의 시작과 끝 부분에 더 초점을 맞춘 LLM의 관심 때문일 수 있습니다. 숙련된 침투 테스터는 일반적으로 더 넓은 관점에서 시스템을 평가하고 가장 실질적인 결과를 제공할 수 있는 후속 단계를 전략화합니다. 앞서 언급한 메모리 손실 문제와 결합하면 이러한 경향으로 인해 LLM이 특정 서비스에 지나치게 집착하게 됩니다. 테스트가 진행됨에 따라 모델은 이전 결과를 완전히 잊어버리고 교착 상태에 도달합니다.

조사 결과 4: LLM은 최근 작업과 깊이 우선 검색 접근 방식을 강력히 선호하므로 종종 한 서비스에 지나치게 집중하고 이전 조사 결과를 잊어버리게 됩니다.

마지막으로 LLM에는 [35]에 언급된 바와 같이 부정확한 결과 생성 및 환각 문제가 있습니다. 이 현상은 두 번째로 빈번한 장애 원인으로 꼽히며 잘못된 명령이 생성되는 것이 특징입니다. 우리의 연구에서 우리는 LLM이 작업에 적합한 도구를 자주 식별하지만 올바른 설정으로 도구를 구성하는 데 실수가 있음을 관찰했습니다. 어떤 경우에는 존재하지 않는 테스트 도구나 도구 모듈을 만들어내기도 합니다.

조사 결과 5: LLM은 종종 본질적인 부정확성과 환각으로 인해 부정확한 작업이나 명령을 생성할 수 있습니다.

침투 테스트 내에서 세 가지 LLM에 대한 탐색적 연구를 통해 엔드투엔드 작업을 실행할 수 있는 잠재력이 드러났습니다. 그럼에도 불구하고 장기 기억을 유지하고, 깊이 우선 접근 방식을 넘어서는 테스트 전략을 고안하고, 정확한 작업을 생성하는 데 어려움이 발생합니다. 다음 섹션에서는 이러한 문제를 해결하는 방법을 설명하고 LLM 기반 침투 테스트 도구를 설계하기 위한 전략의 개요를 설명합니다.

5. 방법론

5.1. 개요

이전 섹션에서 확인된 과제를 고려하여 3개의 LLM 기반 모듈의 시너지 상호 작용을 활용하는 제안된 솔루션 PENTESTGPT를 제시합니다. 그림 3에서 볼 수 있듯이 PENTESTGPT는 추론 모듈, 생성 모듈 및 구문 분석 모듈의 세 가지 핵심 모듈을 통합합니다 . 각 모듈은 대화 및 컨텍스트가 포함된 하나의 LLM 세션을 예약합니다. 사용자는 서로 다른 모듈이 다양한 유형의 메시지를 처리하는 PENESTGPT 와 원활하게 상호 작용합니다 .

이 상호 작용은 사용자가 수행해야 하는 침투 테스트 프로세스의 후속 단계를 제안하는 최종 결정으로 마무리됩니다. 다음 섹션에서는 설계 추론을 설명하고 PENTESTGPT 뒤에 있는 엔지니어링 프로세스에 대한 자세한 분석을 제공합니다.

5.2. 설계 이론적 근거

우리의 핵심 설계 고려 사항은 이전 탐색 연구(섹션 4)에서 관찰된 세 가지 과제에서 나타났습니다. 첫 번째 과제(발견 3)는 메모리 보존으로 인한 침투 테스트 컨텍스트 손실 문제와 관련이 있습니다. 원래 형태의 LLM은 토큰 크기 제한으로 인해 장기 메모리를 유지하는 데 어려움을 겪습니다. 두 번째 장애물(조사 결과 4)은 LLM 챗봇이 최근 대화 내용을 강조하는 경향에서 발생합니다. 침투 테스트 작업에서는 즉각적인 작업을 최적화하는 데 중점을 둡니다.

이 접근 방식은 침투 테스트의 복잡하고 상호 연결된 작업 환경에서는 부족합니다. 세 번째 장애물(조사 결과 5)은 LLM의 부정확한 결과 생성과 관련이 있습니다. 침투 테스트 단계에 대한 특정 작업을 직접 생성하는 작업을 수행하는 경우 출력이 부정확한 경우가 많으며 때로는 PENTESTGPT가 이러한 문제를 해결하도록 설계되어 침투 테스트 작업에 더 적합하도록 설계되었습니다. 우리는 감독이 전반적인 절차를 계획하고 이를 개

별 테스터를 위한 하위 작업으로 나누는 실제 침투 테스트 팀에서 사용하는 방법론에서 영감을 얻었습니다. 각 테스터는 독립적으로 작업을 수행하고 더 넓은 맥락을 철저하게 이해하지 않고도 결과를 보고합니다. 그런 다음 감독은 작업을 재정의하는 등 다음 단계를 결정하고 후속 테스트 라운드를 시작합니다. 기본적으로 감독은 테스트의 세세한 부분에 얽매이지 않고 전반적인 전략을 관리합니다. 이 접근 방식은 PENESTGPT의 기능에 반영되어 침투 테스트 수행 시 효율성과 적응성을 향상시킵니다. 우리의 전략은 침투 테스트를 다음 작업 식별과 작업 완료를 위한 구체적인 작업 생성이라는 두 가지 프로세스로 나눕니다. 각 프로세스는 하나의 LLM 세션으로 구동됩니다. 이 설정에서 작업 식별을 담당하는 LLM 세션은 진행 중인 침투 테스트 상태의 전체 컨텍스트를 유지합니다. 동시에 세부 작업 생성 및 정보 구문 분석은 다른 세션에서 관리됩니다. 이러한 책임 분할은 전반적인 맥락을 유지하면서 효과적인 작업 실행을 촉진합니다.

LLM이 침투 테스트 작업을 효과적으로 수행할 수 있도록 지원하기 위해 우리는 사용자 입력에 맞는 일련의 프롬프트를 설계합니다. 우리는 이 과정에서 CoT(Chain-of-Thought) [36] 방법론을 활용합니다. CoT에서 알 수 있듯이 LLM의 성능과 추론 능력은 입력, 사고 사슬, 출력 프롬프트 형식을 사용하여 크게 향상될 수 있습니다. 여기서 생각의 사슬은 결과로 이어지는 일련의 중간 자연어 추론 단계를 나타냅니다. 침투 테스트 작업을 마이크로 단계로 분석하고 침투 테스트 정보 처리를 통해 LLM을 안내하는 예제를 통해 프롬프트를 설계합니다.

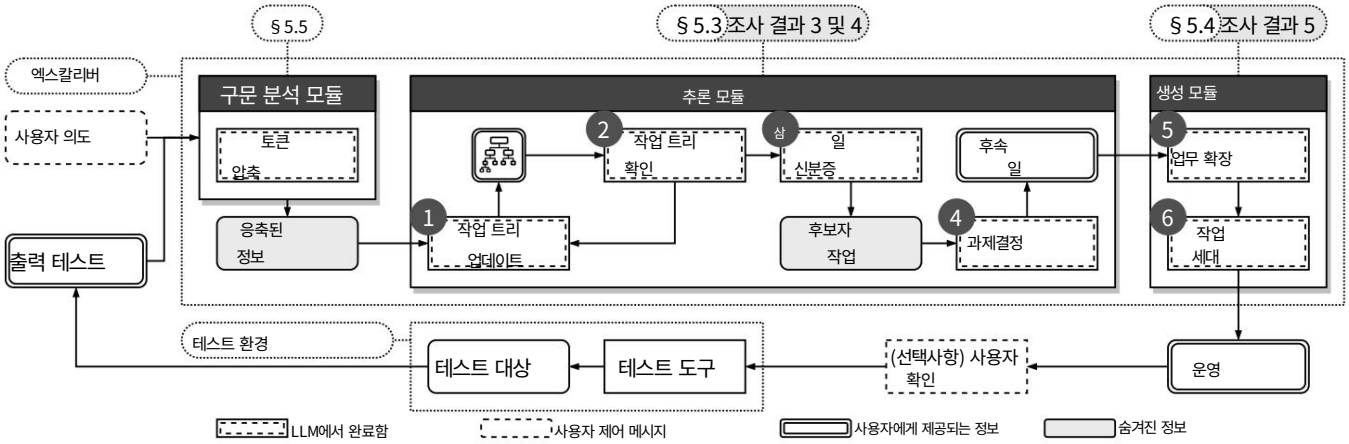


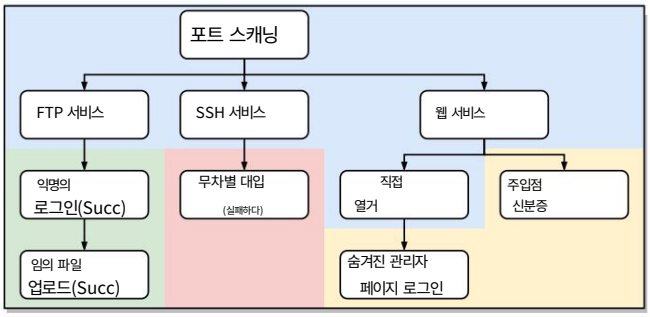
그림 3: PENESTGPT 개요 .

단계별로 진행하여 궁극적으로 원하는 결과를 얻습니다. 전체 프롬프트는 익명화된 오픈 소스 프로젝트[18]에서 확인할 수 있습니다.

5.3. 추론 모듈

추론 모듈은 거시적 관점에서 침투 테스트 작업을 감독하는 팀 리더와 유사하게 우리 시스템에서 중추적인 역할을 합니다. 사용자로부터 테스트 결과나 의도를 파악하고 다음 단계를 위한 테스트 전략을 준비합니다. 이 테스트 전략은 추가 계획을 위해 생성 모듈로 전달됩니다.

침투 테스트 프로세스를 효과적으로 감독하고 정확한 지침을 제공하려면 테스트 절차와 결과를 자연어 형식으로 변환하는 것이 중요합니다. 침투 테스트 절차를 개략적으로 설명하는 데 자주 사용되는 공격 트리[37]의 개념에서 영감을 얻어 침투 테스트 작업 트리(PTT)의 개념을 소개합니다. 상태 표현 테스트에 대한 이 새로운 접근 방식은 속성 트리 [38]: 정의 1(속성 트리) 개념에 뿌리를 두고 있습니다. 속성 트리는 모서리 레이블이 지정된 속성 폴리트리입니다. G = (V, E, λ, μ) 여기서 V 는 노드(또는 정점) 집합이고, E 는 방향이 있는 모서리 집합이며, λ : E → Σ 는 모서리입니다. labeling 함수는 알파벳 Σ 부터 각 edge와 μ : (V ∪ E) × K → S 의 label 을 edge와 node에 key(from K)-value(from S) 쌍의 속성을 할당하는 함수이다.



a) PTT 대표자

작업 트리:
1. 포트 스캐닝 수행(완료)
- 포트 21, 22, 80이 열려 있습니다.
- 서비스는 FTP, SSH, 웹 서비스입니다.
2. 테스트 수행 2.1 FTP 서비스 테스트
2.1.1 익명 로그인 테스트(성공)
2.1.1.1 익명 업로드 테스트(성공)
2.2 SSH 서비스 테스트 2.2.1 무차별 대입
(실패)
2.3 웹 서비스 테스트(진행중)
2.3.1 디렉토리 열거 2.3.1.1 숨은 관리자 찾기(할 일)
2.3.2 주입 식별(todo)

b) 자연 언어로 PTT 표현

그림 4: a) 시각화된 트리 형식 및 b) LLM으로 인코딩된 자연어 형식의 침투 테스트 작업 트리.

속성 트리의 정의가 주어지면 PTT는 다음과 같이 정의됩니다. 정의 2(침투 작업 트리). PTT T는 (N, A) 쌍입니다 . 여기서: (1) N 은 트리 구조로 구성된 노드 집합입니다. 각 노드에는 고유한 식별자가 있으며, 부모가 없는 루트라는 특수 노드가 있습니다. 루트를 제외한 각 노드에는 정확히 하나의 상위 노드와 0개 이상의 하위 노드가 있습니다. (2) A는 각 노드 n ∈ N 에 속성 집합 A(n) 을 할당하는 함수입니다 . 각 속성은 (a, v) 쌍입니다 . 여기서 a 는 속성 이름이고 v 는 속성 값입니다. 속성 세트는 각 노드마다 다를 수 있습니다.

그림 3에 설명된 대로 추론 모듈의 작동은 PTT를 통해 작동하는 네 가지 주요 단계로 진행됩니다. ❶ 모듈은 처음에 사용자의 의도를 흡수하여 자연어 형태로 초기 PTT를 구성합니다.

이는 세심하게 제작된 프롬프트를 사용하여 PPT의 예와 정의를 LLM에게 주의 깊게 지시함으로써 달성됩니다. LLM 출력을 구문 분석하여 트리 구조의 형식이 정확하게 지정되었는지 확인합니다. 트리 구조의 특성으로 인해 그림 4와 같이 계층화된 글머리 기호를 통해 자연어 형식으로 표현될 수 있습니다. 추론 모듈은 효과적으로

전체 침투 테스트 프로세스를 포괄하는 작업 트리를 유지하여 메모리 손실 문제를 극복합니다. ❷ 트리 정보를 업데이트한 후 새로 업데이트된 PTT에 대해 정확성을 확인하는 검증 단계를 진행합니다. 이 프로세스는 침투 테스트 프로세스의 원자적 작업이 가장 낮은 수준 하위 작업의 상태에만 영향을 미쳐야 한다는 원칙에 따라 PTT의 리프 노드만 수정되었는지 명시적으로 확인합니다. 이 단계는 추론 과정의 정확성을 확인하고 LLM의 환각으로 인해 전체 트리 구조가 변경될 수 있는 가능성을 방지합니다. 불일치가 발생하면 수정 및 재생성을 위해 정보가 LLM으로 되돌아갑니다. ❸ 업데이트된 PTT를 통해 추론 모듈은 현재 트리 상태를 평가하고 추가 테스트를 위한 후보 단계로 사용할 수 있는 실행 가능한 하위 작업을 찾아냅니다. ❹ 마지막으로 모듈은 이러한 하위 작업이 성공적인 침투 테스트 결과로 이어질 가능성을 평가합니다. 그런 다음 최상위 작업을 출력으로 추천합니다. 이 작업의 예상 결과는 이후 심층 분석을 위해 생성 모듈로 전달됩니다. 이는 탐색적 연구에서 입증된 바와 같이 LLM, 특히 GPT-4가 시스템 상태 정보와 함께 제공될 때 잠재적인 취약점을 식별할 수 있기 때문에 실현 가능합니다. 이러한 절차적 접근 방식을 통해 추론 모듈은 LLM의 고유한 한계 중 하나, 즉 가장 최근 작업에만 집중하는 경향을 정확하게 해결할 수 있습니다. 테스터가 올바른 작업이 잘못되었거나 선호하는 방식으로 완료되지 않았음을 식별하는 경우 섹션 5.6에서 자세히 설명하는 대화형 핸들을 통해 PTT를 수동으로 수정할 수도 있습니다.

우리는 각 단계가 완료될 때까지 추론 모듈을 순차적으로 안내하기 위해 4가지 세트의 프롬프트를 고안합니다.
결과의 재현성을 강화하기 위해 힌트 생성[39]이라는 기술을 사용하여 이러한 프롬프트를 더욱 최적화합니다. 실제 경험을 통해 우리는 LLM이 침투 테스트와 관련된 트리 구조 정보를 해석하는 데 능숙하고 테스트 결과에 따라 이를 정확하게 업데이트할 수 있다는 것을 확인했습니다.

5.4. 생성 모듈

생성 모듈은 추론 모듈의 특정 하위 작업을 구체적인 명령이나 지침으로 변환합니다. 새로운 하위 작업이 수신될 때마다 생성 모듈에서 새로운 세션이 시작됩니다. 이 전략은 실행 중인 즉각적인 작업에서 중요한 침투 작업의 컨텍스트를 효과적으로 분리하여 LLM이 특정 명령 생성에 전적으로 집중할 수 있도록 합니다.

수신된 하위 작업을 특정 작업으로 직접 변환하는 대신 우리 설계에서는 CoT 전략[36]을 사용하여 이 프로세스를 두 개의 순차적 단계로 분할합니다.
이 설계 결정은 모델의 추론 기능을 향상시켜 모델의 부정확성 및 환각과 관련된 문제를 직접적으로 해결합니다. 특히, ❺ 추론 모듈로부터 간결한 하위 작업을 수신하면 생성 모듈은 이를 일련의 세부 단계로 확장하는 것으로 시작합니다. 특히, 프롬프트

이 하위 작업과 관련하여 LLM은 테스트 환경 내에서 사용 가능한 도구 및 작업을 고려해야 합니다. ❻ 이어서 생성 모듈은 이러한 확장된 각 단계를 실행할 준비가 된 정확한 터미널 명령이나 수행할 특정 그래픽 사용자 인터페이스(GUI) 작업에 대한 자세한 설명으로 변환합니다. 이 단계별 번역을 통해 잠재적인 모호성을 제거하여 테스터가 지침을 직접 원활하게 따를 수 있습니다. 이 2단계 프로세스를 구현하면 LLM이 실제 시나리오에서 실현할 수 없는 작업을 생성하는 것을 효과적으로 방지하여 침투 테스트 절차의 성공률을 높일 수 있습니다.

추론 모듈에서 제공하는 전략적 통찰력과 침투 테스트 수행에 필요한 실행 가능한 단계 사이의 가교 역할을 함으로써 생성 모듈은 상위 수준 계획이 정확하고 실행 가능한 단계로 변환되도록 보장합니다. 이러한 변환 프로세스는 침투 테스트 절차의 전반적인 효율성을 크게 강화합니다.

예시적인 예. 우리는 실제 실행 예제를 활용하여 추론 모듈과 생성 모듈이 어떻게 협력하여 침투 테스트 작업을 완료하는지 설명합니다. 그림 5 는 중간 난이도 목표인 HackTheBox 머신 Car-rier[40]에서 작동하는 PENTESTGPT의 단일 반복을 보여줍니다 . a-1)에 설명된 대로 PTT는 자연어 형식으로 테스트 상태를 인코딩하여 대상 시스템의 열린 포트(21, 22,80)를 표시합니다.

이후 추론 모듈은 사용 가능한 작업을 식별하도록 지시받습니다. 빨간색으로 강조 표시된 것처럼 서비스 검색은 PTT의 리프 노드에서 사용 가능한 유일한 작업입니다. 따라서 이 작업은 선택되어 명령 생성을 위해 생성 모듈로 전달됩니다. 생성된 명령은 테스트 환경에서 실행되며, 실행 결과는 Reasoning Module로 전달되어 PTT를 업데이트합니다. a-2)에서 Reasoning Module은 이전 스캔 결과를 PTT에 통합하고 이를 이전 PTT와 상호 참조하여 리프 노드만 업데이트합니다. 그런 다음 실행할 수 있는 작업을 찾습니다. 이 경우 포트 80에서 웹 서비스를 검색하고 SSH 서비스에서 알려진 취약점을 확인하는 두 가지 작업이 나타납니다. LLM은 어떤 작업이 더 유망한지 평가하고 종종 더 취약하다고 여겨지는 웹 서비스를 조사하기로 선택합니다. 이 작업은 생성 모듈로 전달됩니다. 생성 모듈은 일반적으로 사용되는 웹 스캐닝 스크립트인 nikto [41]를 사용하여 이 일반적인 작업을 세부 프로세스로 전환합니다. 테스터가 침투 테스트 작업을 완료할 때까지 반복 프로세스가 계속됩니다.

5.5. 구문 분석 모듈

구문 분석 모듈은 지원 인터페이스로 작동하여 사용자와 다른 두 핵심 모듈 간에 교환되는 자연어 정보를 효과적으로 처리할 수 있습니다. 두 가지 요구 사항이 이 모듈의 존재를 정당화할 수 있습니다. 첫째, 보안 테스트 도구 출력은 일반적으로 장황하고 관련 없는 세부 정보로 가득 차 있어 계산 비용이 많이 들고 입력하는 데 불필요하게 중복됩니다.

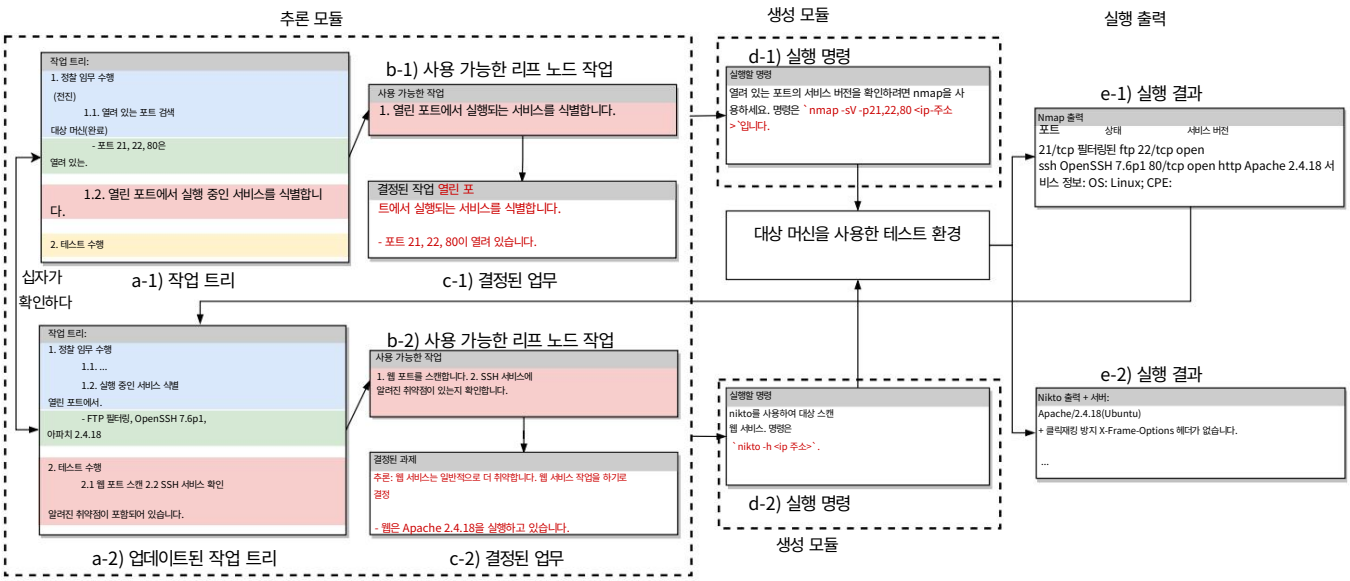


그림 5: 테스트 대상 HTB-Carrier의 작업 트리 업데이트 프로세스 시연

이러한 확장된 출력은 LLM으로 직접 전달됩니다. 둘째, 보안 도메인에 대한 전문 지식이 없는 사용자는 보안 테스트 결과에서 주요 통찰력을 추출하는 데 어려움을 겪을 수 있으며, 이는 중요한 테스트 정보를 요약하는 데 어려움을 겪을 수 있습니다. 결과적으로, 파싱 모듈은 이 정보를 합리화하고 압축하는 데 필수적입니다.

PENTESTGPT 에서 구문 분석 모듈은 네 가지 고유한 유형의 정보를 처리하도록 고안되었습니다. (1) 다음 작업 과정을 지시하기 위해 사용자가 제공한 지시문인 사용자 의도, (2) 원시 출력을 나타내는 보안 테스트 도구 출력 다양한 보안 테스트 도구에 의해 생성된 것, (3) HTTP 웹 인터페이스에서 파생된 모든 원시 정보를 포함하는 원시 HTTP 웹 정보, (4) 침투 테스트 프로세스 중에 추출된 소스 코드입니다. 사용자는 자신이 제공하는 정보의 카테고리를 지정해야 하며, 각 카테고리는 세심하게 디자인된 프롬프트 세트와 짝을 이룹니다. 소스 코드 분석을 위해 GPT-4 코드 해석기[42]를 통합하여 작업을 실행합니다.

5.6. 적극적인 피드백

LLM은 통찰력 있는 결과를 생성할 수 있지만 결과에 수정이 필요한 경우도 있습니다. 이를 촉진하기 위해 우리는 사용자가 추론 모듈과 직접 상호 작용할 수 있도록 하는 활성 피드백이라고 알려진 대화형 핸들을 PENESTGPT 에 도입했습니다. 이 프로세스의 중요한 특징은 사용자가 일부 정보를 업데이트하기를 명시적으로 원하지 않는 한 추론 모듈 내의 컨텍스트를 변경하지 않는다는 것입니다. PTT를 포함한 추론 컨텍스트는 고정된 토큰 청크로 저장됩니다. 이 토큰 덩어리는

활성 피드백 상호 작용 중에 새로운 LLM 세션에 제공되며 사용자는 이에 관해 질문을 할 수 있습니다.
이렇게 하면 원래 세션이 영향을 받지 않고 유지됩니다.

사용자는 불필요한 변경 없이 항상 추론 컨텍스트를 쿼리할 수 있습니다. 사용자가 PTT를 업데이트하는 것이 필요하다고 생각하는 경우 모델에 그에 따라 추론 컨텍스트 기록을 업데이트하도록 명시적으로 지시할 수 있습니다.
이는 사용자가 의사결정 프로세스에 적극적으로 참여할 수 있는 강력하고 유연한 프레임워크를 제공합니다.

5.7. 논의

우리는 탐색 연구에서 확인된 과제를 해결하기 위해 PENTEST-GPT 에 대한 다양한 설계 대안을 탐색합니다 . 우리는 다양한 디자인을 실험했으며 여기서는 몇 가지 주요 결정에 대해 논의합니다.

토큰 크기로 컨텍스트 손실 해결: 컨텍스트 손실을 완화하기 위한 간단한 솔루션은 토큰 크기가 확장된 LLM 모델을 사용하는 것입니다. 예를 들어 GPT-4는 8k 및 32k 토큰 크기 제한이 있는 버전을 제공합니다.

그러나 이 접근 방식은 두 가지 실질적인 문제에 직면합니다. 첫째, dirbuster[43]와 같은 단일 테스트 도구의 출력이 수천 개의 토큰으로 구성될 수 있으므로 32k 토큰 크기도 침투 테스트 시나리오에 부적절할 수 있습니다. 결과적으로 32k 제한이 있는 GPT-4는 전체 테스트 컨텍스트를 유지할 수 없습니다. 둘째, 전체 대화 기록이 32k 토큰 경계 내에 맞는 경우에도 API는 여전히 최신 콘텐츠에 치우쳐 로컬 작업에 초점을 맞추고 더 넓은 컨텍스트를 간과할 수 있습니다. 이러한 문제는 추론 모듈과 구문 분석 모듈의 설계를 공식화하는 데 도움이 되었습니다.

컨텍스트 길이를 향상시키는 벡터 데이터베이스: LLM의 컨텍스트 길이를 향상시키는 또 다른 기술에는 벡터 데이터베이스가 포함됩니다[44], [45]. 데이터를 벡터 임베딩으로 변환함으로써 LLM은 정보를 효율적으로 저장하고 검색하여 실질적으로 장기 메모리를 생성할 수 있습니다. 이론적으로 침투 테스트 도구 출력은 보관될 수 있습니다.

벡터 데이터베이스에 있습니다. 하지만 실제로는 많은 결과가 미묘한 차이만 있을 뿐 매우 유사하고 다양하다는 것을 알 수 있습니다. 이러한 유사성은 종종 혼란스러운 정보 검색으로 이어집니다. 벡터 데이터베이스에만 의존하는 것은 침투 테스트 작업에서 컨텍스트 손실을 극복하는 데 실패합니다. 벡터 데이터베이스를 PENTESTGPT 설계에 통합하는 것은 향후 연구를 위한 길입니다.

정보 추출의 정확성: 정확한 정보 추출은 토큰 사용을 보존하고 LLM의 장황함을 방지하는 데 중요합니다. 다양한 정보를 추출하기 위해 규칙 기반 방법이 일반적으로 사용됩니다. 그러나 규칙 기반 기술은 자연어의 본질적인 복잡성과 침투 테스트의 정보 유형의 다양성을 고려할 때 엔지니어링 측면에서 비용이 많이 듭니다. 우리는 여러 가지 일반적인 입력 정보 유형을 관리하기 위해 구문 분석 모듈을 고안했으며, 이는 실현 가능하고 효율적인 전략입니다.

LLM의 한계: LLM은 모든 것을 포괄하는 솔루션이 아닙니다. 현재 LLM은 환각 [46] 및 오래된 지식을 포함한 결함을 나타냅니다. 환각을 방지하기 위해 작업 트리 검증을 구현하는 등의 완화 노력은 추론 모듈이 잘못된 결과를 생성하는 것을 완전히 방지하지 못할 수도 있습니다.

따라서 LLM을 효과적으로 운영하는 데 필요한 전문 지식과 지침의 입력을 촉진하는 인간 참여형 전략이 중요해졌습니다.

6. 평가

이 섹션에서는 PENTEST 의 성능을 평가합니다.
GPT는 다음 네 가지 연구 질문에 중점을 둡니다. RQ3(성능): PENTESTGPT 의 성능은 기본 LLM 모델 및 인간 전문가의 성능과 어떻게 비교됩니까?

RQ4(전략): PENTESTGPT는 LLM이나 인간 전문가가 사용하는 것과 비교하여 다른 문제 해결 전략을 사용 합니까 ?

RQ5(Ablation): PENTEST- GPT 내의 각 모듈은 전체 침투 테스트 성능에 어떻게 기여합니까?

RQ6(실용성): PENTESTGPT가 실제 침투 테스트 작업에 실용적이고 효과적 입니까 ?

6.1. 평가 설정

우리는 익명화된 프로젝트 웹사이트[18]에서 사용할 수 있는 1,700줄의 Python3 코드와 740개의 프롬프트로 PENESTGPT를 구현합니다. 섹션 3에서 구축한 벤치마크에 대한 성능을 평가합니다. 이 평가에서는 PENTESTGPT를 GPT-3.5 및 GPT-4와 통합하여 PENTESTGPT-GPT-3.5 및 PENTESTGPT-GPT-4라는 두 가지 작업 버전을 구성합니다. API 액세스가 부족하기 때문에 Bard와 같은 다른 LLM 모델을 선택하지 않습니다. 이전 실험과 마찬가지로 동일한 실험 환경 설정을 사용하고 PENTESTGPT에 자동화되지 않은 침투 테스트 도구만 사용하도록 지시합니다.

6.2. 성능평가(RQ3)

PENTESTGPT- GPT-3.5, PENTESTGPT-GPT-4 의 전반적인 작업 완료 상태 와 LLM의 순진한 사용이 그림 6a에 나와 있습니다. 그림에서 볼 수 있듯이 LLM을 기반으로 하는 당사의 솔루션은 단순한 LLM 적용에 비해 우수한 침투 테스트 기능을 보여줍니다. 특히, PENTESTGPT-GPT-4는 다른 세 가지 솔루션을 능가하여 쉬운 난이도 목표 7개 중 6개와 중간 난이도 목표 4개 중 2개를 성공적으로 해결했습니다.

이 성능은 PENESTGPT-GPT-4가 쉬운 수준부터 중간 수준까지의 침투 테스트 대상을 처리할 수 있음을 나타냅니다. 한편, PENTESTGPT-GPT-3.5는 쉬운 난이도의 두 가지 과제만 해결하는데, 이는 GPT-4에서 발견된 침투 테스트 관련 지식이 부족한 GPT-3.5에 기인할 수 있는 불일치입니다.

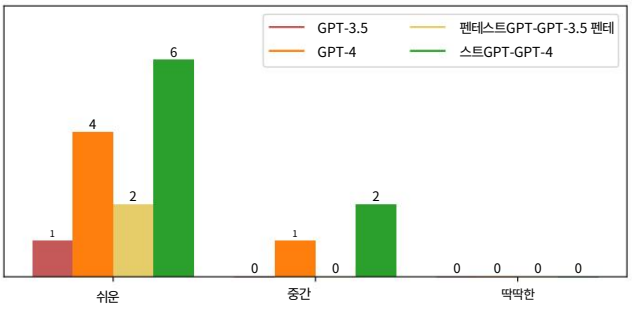
PENTESTGPT-GPT-3.5, PENTESTGPT-GPT-4 의 하위 작업 완료 상태 와 LLM의 순진한 사용은 그림 6b에 나와 있습니다. 그림에서 볼 수 있듯이 PENTESTGPT-GPT-3.5와 PENTESTGPT-GPT-4는 모두 LLM의 표준 활용도보다 더 나은 성능을 발휘합니다. PENTESTGPT-GPT-4는 순진한 GPT-4에 비해 중간 난이도 목표를 하나 더 해결할 뿐만 아니라 111% 더 많은 하위 작업(57 대 27)을 수행한다는 점은 주목할 만합니다 .

이는 우리의 설계가 컨텍스트 손실 문제를 효과적으로 해결하고 보다 유망한 테스트 결과로 이어진다는 점을 강조합니다.
그럼에도 불구하고 모든 솔루션은 어려운 테스트 목표로 인해 어려움을 겪고 있습니다. 섹션 4에서 자세히 설명했듯이 어려운 난이도 목표는 일반적으로 침투 테스터의 깊은 이해가 필요합니다. 테스트 목표를 달성하려면 기존 침투 테스트 도구나 스크립트를 수정해야 할 수도 있습니다. 우리의 설계는 LLM의 취약성에 대한 지식을 확장하지 않으므로 이러한 보다 복잡한 대상에 대한 성능을 눈에 띄게 향상시키지 않습니다.

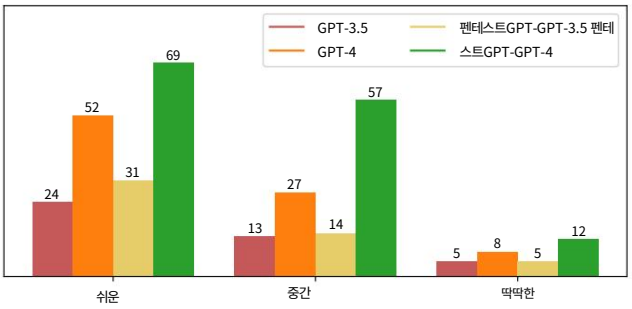
6.3. 전략평가(RQ4)

그런 다음 PENTESTGPT 가 사용하는 문제 해결 전략을 조사하여 LLM 및 인간 전문가의 전략과 대조합니다. PENTESTGPT 의 침투 테스트 프로세스를 수동으로 분석하여 문제 해결에 대한 기본 접근 방식을 종합합니다. 우리는 놀랍게도 PENTESTGPT가 인간 전문가와 유사한 방식으로 침투 테스트 작업을 분해하여 전반적인 목표를 성공적으로 달성한다는 것을 발견했습니다. PENTESTGPT는 가장 최근에 발견된 작업에만 초점을 맞추는 대신 성공적인 결과로 이어질 수 있는 잠재적인 하위 작업을 정확히 찾아낼 수 있습니다.

그림 7 은 VulnHub 시스템인 Hackable II[47]를 처리하는 동안 GPT-4와 PENTESTGPT 간의 전략적 차이점을 보여주는 예시를 제공합니다 . 이 표적은 임의의 파일 업로드를 허용하는 FTP 서비스와 FTP를 통해 파일을 볼 수 있는 웹 서비스라는 두 가지 취약한 서비스로 구성됩니다. 악용에 성공하려면 FTP 서비스를 통해 악성 PHP 셸을 업로드하고 웹 서비스를 통해 이를 트리거하여 두 서비스를 모두 악용해야 합니다. 그림과 같이 GPT-4는 FTP 서비스를 열거하는 것부터 시작하여 파일 업로드 취약점(❶-❸)을 성공적으로 식별합니다. 그러나 상관 관계에 실패합니다.



(a) 전체 완료 상태.



(b) 하위 작업 완료 상태.

그림 6: 전체 목표 완료 및 하위 작업 완료에 대한 GPT-3.5, GPT-4, PENTESTGPT-GPT-3.5 및 PENTESTGPT-GPT-4 의 성능 .

이는 웹 서비스와 관련되어 다음 단계에서 불완전한 악용이 발생합니다. 반대로 PENTESTGPT는 FTP 서비스 열거와 웹 서비스 검색 사이를 전환하는 보다 전체적인 접근 방식을 따릅니다. 특히, PENTESTGPT는 먼저 ❶ FTP 서비스와 ❷ 웹 서비스를 열거하여 전반적인 상황을 파악합니다. 그런 다음 ❸ FTP 서비스의 우선순위를 지정하고 ❹ 결국 파일 업로드 취약점을 발견합니다. 더 중요한 것은 이 프로세스에서 PENESTGPT가 FTP에서 사용 가능한 파일이 웹 서비스의 파일과 동일하다는 것을 식별한다는 것입니다. 이러한 발견을 연결함으로써 PENESTGPT는 테스터에게 ❺ 셸 업로드를 수행하도록 안내하여 ❻ 성공적인 역방향 셸로 이어집니다. 이 전략은 연습 솔루션과 일치하며 침투 테스트 프로세스에 대한 PENTESTGPT의 포괄적인 이해와 다음에 추구할 최적의 하위 작업에 대한 효과적인 결정을 내리는 능력을 강조합니다. 이는 PENESTGPT의 전략적 사고와 테스트 프로세스의 다양한 측면을 통합하는 능력을 보여줍니다.

두 번째 관찰은 PENTESTGPT가 인간 전문가와 더 유사하게 행동하지만 여전히 인간이 적용하지 않을 일부 전략을 보여준다는 것입니다 . 예를 들어, PENTESTGPT는 여전히 취약점 검색 전에 무차별 대입 공격을 우선시합니다. 이는 PEN-TESTGPT가 항상 대상 시스템에서 SSH 서비스를 무차별 대입하려고 시도하는 경우에 분명합니다 .

그런 다음 실패한 침투 테스트 사례를 분석하여 PENESTGPT의 한계를 이해합니다. 일부 고급 침투 테스트 기술이 없다는 점 외에도 두 가지 주요 문제가 드러납니다. 첫째, PENESTGPT가 어려움을 겪습니다.

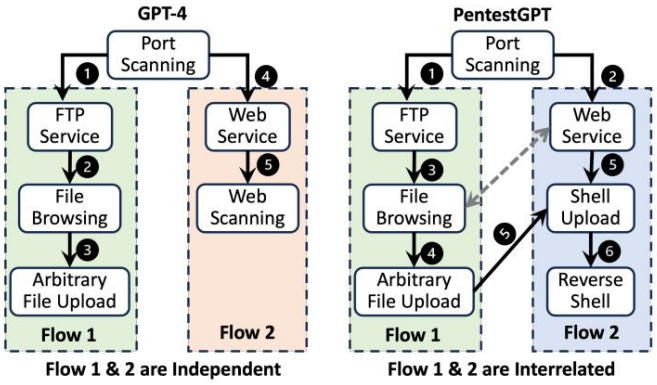


그림 7: VulnHub-Hackable II에서 GPT-3.5와 PENESTGPT 간의 침투 테스트 전략 비교 .

이미지를 해석하는 것입니다. LLM은 텍스트 이해에만 국한되어 이미지를 정확하게 처리할 수 없습니다. 이 문제는 텍스트와 시각적 데이터를 이해하기 위한 대규모 다중 모드 모델을 개발함으로써 해결될 수 있습니다. 둘째, 특정 사회공학적 속임수와 미묘한 단서를 파악하지 못합니다. 예를 들어, 실제 침투 테스터는 대상 서비스에서 수집한 정보를 사용하여 무차별 단어 목록을 만드는 경우가 많습니다. PENTESTGPT는 웹 서비스에서 이름 목록을 검색할 수 있지만 해당 이름에서 단어 목록을 생성하는 도구 사용을 지시하지 않습니다 . 이러한 제한은 인간의 통찰력과 복잡한 추론이 자동화된 솔루션보다 여전히 더 능숙한 영역에서 개선의 필요성을 강조합니다.

6.4. 절제 연구(RQ5)

우리는 Reasoning Module, Generation Module, Parsing Module의 세 가지 모듈이 PENTESTGPT 성능에 어떻게 기여하는지에 대한 Ablation 연구를 수행합니다 .
우리는 세 가지 변형을 구현합니다: 1)

PENTESTGPT-NO-PARSING: 구문 분석 모듈이 비활성화되어 모든 데이터가 시스템에 직접 공급됩니다.

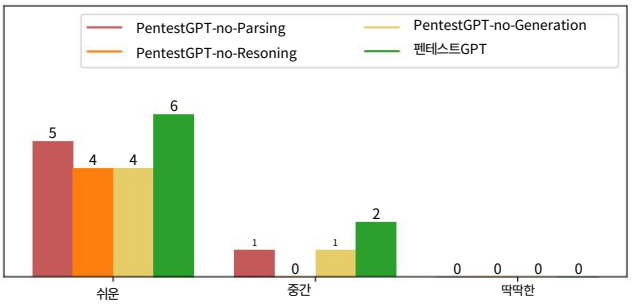
2) PENESTGPT-NO-GENERATION: 생성 모듈이 비활성화되어 추론 모듈 자체 내에서 작업 생성이 완료됩니다.

작업 생성에 대한 프롬프트는 일관되게 유지됩니다.

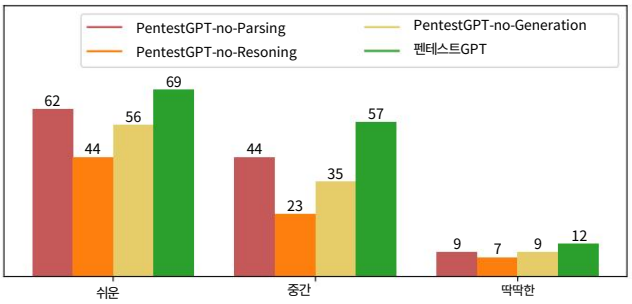
3) PENTESTGPT-NO-REASONING: 추론 모듈이 비활성화되었습니다. PTT 대신 이 변형은 탐색적 연구에 설명된 대로 침투 테스트를 위해 LLM에 사용되는 것과 동일한 방법론을 채택합니다.

모든 변형은 테스트를 위해 GPT-4 API와 통합되었습니다.

침투 테스트 벤치마크에서 테스트한 세 가지 변형의 결과는 그림 8에 나와 있습니다. 일반적으로 PENTESTGPT는 전체 목표 및 하위 작업 완료와 관련하여 세 가지 절제 기준보다 우월함을 보여줍니다. 주요 결과는 다음과 같습니다. (1) 구문 분석 모듈이 없는 경우 PENTESTGPT-NO-PARSING은 전체 구성에 비해 전체 작업 및 하위 작업 완료에서 약간 낮은 성능을 얻습니다. 침투 테스트에서는 정보를 구문 분석하는 것이 유리하지만,



(가) 전체 완료현황



(나) 하위과제 완료현황

그림 8: 정규화된 평균 코드 적용 범위 (µLOC) 와 버그 감지 모두에 대한 PENTESTGPT, PENTESTGPT -NO-ANNOTATION, PENTESTGPT-OPERATION- ONLY 및 PENTESTGPT-PARAMETER-ONLY 의 성능 .

32k 토큰 크기 제한은 다양한 출력에 충분한 경우가 많습니다.
추론 모듈의 고유한 설계를 고려하면
전체 테스트 컨텍스트, 구문 분석 모듈 부족
도구의 성능을 크게 저하시키지 않습니다. (2)
PENTESTGPT-NO-REASONING이 최악의 상황을 맞이하여 완료되었습니다.
전체 솔루션으로 달성한 하위 작업은 53.6%에 불과하며,
테스트에서 GPT-4를 순진하게 적용한 것보다 훨씬 열등한 결과입니다. 우리
는 이것을 생성 모듈에 기인합니다
LLM 컨텍스트에 보충 하위 작업을 추가합니다. 부터
프롬프트는 추론 모듈이 없는 시나리오에 맞춰져 있지 않으며 결과 출력은 시
나리오와 관련이 없습니다.
생성 모듈이 없는 순진한 LLM. 뿐만 아니라,
확장된 생성 출력은 원래 컨텍스트를 모호하게 합니다.
LLM이 업무에 집중하는 능력을 방해합니다.
따라서 테스트에 실패합니다. (3) 펜테스트GPT-NO-GENERATION
순진하게 사용된 GPT-4보다 약간 높은 성능을 구현합니다. 이는 세대가 없기
때문에 발생합니다.
모듈, 테스트 절차는 사용법과 매우 유사합니다.
LLM의 특히, 생성 모듈은 주로
테스터가 정확한 침투를 수행하도록 안내하기 위한 것입니다.
테스트 작업. 이 모듈이 없으면 테스터는 다음을 수행할 수 있습니다.
도구를 작동하기 위해 보충 정보에 의존
또는 테스트를 완료하는 데 필수적인 스크립트.

6.5. 실용성 연구(RQ6)

PENESTGPT가 실용성을 보여줍니다.
제작된 벤치마크를 넘어서는 실제 침투 테스트를 위한 것입니다. 이를 위해 우리
는 PENESTGPT 에 참여합니다.

표 5: 활성화된 Hack-TheBox 챌린지에 대한 PENESTGPT 성능.

기계	난이도	완료 완료	사용자 비용(USD)		
사우	쉬운	✓	4798	15.2	
순례	쉬운	✓	5474	12.6	
토폴로지	쉬운		4500	8.3	
PC	쉬운		6061	16.1	
모니터2	쉬운	✓	8684	9.2	
권한	중간	✓	1209	11.5	
샌드웜 미디엄			2106	10.2	
목성	중간		1494	6.6	
기민한	중간		4395	22.5	
OnlyForYou 미디엄			2296	19.3	
총	-	✓ 6	-	131.5	

HackTheBox 활성 머신 챌린지, 글로벌 테스터에게 공개된 일련의 침투 테스
트 목표입니다. 각 도전
두 가지 구성요소로 구성됩니다: 사용자 플래그, 검색 가능
초기 사용자 액세스 및 루트 플래그 획득 후 획득 가능
루트 액세스. 우리의 평가에는 5가지 쉬운 목표가 포함됩니다.
난이도는 5개이고 중간 난이도입니다. 이번 운동 중에는
PENTESTGPT는 GPT-4의 32k 토큰 API를 활용하여
각 대상에 대해 최대 5개의 테스트를 수행합니다. 성공은 오로지 다음으로 정의됩니다.
루트 플래그 캡처. 표 5에 성능이 자세히 나와 있습니다.
이러한 과제에 대한 PENTESTGPT 의 3 . 궁극적으로 PEN-TESTGPT는 3개
의 쉬운 과제와 5개의 중간 과제를 완료합니다.
이 연습에 대한 총 지출은 131.5입니다.
USD, 대상당 평균 21.92 USD. 이 비용은 눈에 띄게
인간 침투 테스터를 고용하는 것보다 낮아서 넘어집니다.
허용 가능한 범위 내에서. 따라서 우리의 평가는 실행 가능한 침투를 제공하는
PENESTGPT의 능력을 강조합니다.
효율적인 비용으로 실제 환경에서 결과를 테스트하고,
이를 통해 실용적인 도구로서의 잠재력을 강조합니다.
사이버보안 도메인.

7. 토론

우리는 침투 테스트 연습이
테스트 대상 교육 자료의 일부였을 수도 있습니다.
LLM은 잠재적으로 결과를 편향시킵니다. 이를 완화하기 위해 우리는
두 가지 조치를 취하십시오. 먼저 LLM이
대상 컴퓨터에 대한 사전 지식이 없습니다. 우리는하다
테스트된 기계가 다음 범위 내에 있는지 LLM에 메시지를 표시하여 이를 수행합니다.
그들의 지식 기반. 둘째, 2021년 이후에 출시되는 침투 테스트 대상 머신을 벤
치마크에 포함시켰습니다.
이는 OpenAI 모델의 훈련 데이터에 속하지 않습니다.
최신 HackTheBox에 대한 실용성 연구
과제는 또한 PENESTGPT가 해결할 수 있음 을 보여줍니다.
목표에 대한 사전 지식 없이 도전합니다.
빠르게 진화하는 LLM의 특성과 불일치
사용 가능한 API에서 PENESTGPT의 설계가 무효화될 수 있습니다.
프롬프트. 우리는 메시지를 일반적이고 적절하게 만들기 위해 노력합니다.
다양한 LLM을 위한 그러나 해킹 특성상
일부 LLM은 특정 침투 테스트 생성을 거부합니다.
구체적인 역방향 쉘 스크립트와 같은 콘텐츠. 우리의 프롬프트
LLM이 침투 테스트 관련 정보를 생성하도록 안내하는 탈옥 기술[48]을 포함합
니다. 생성 방법

3. 완료 사용자(Completed Users)는 전 세계적으로 누적된 사용자 수를 나타냅니다.
원고 제출 시점을 기준으로 목표를 완료했습니다. 참고하세요
HackTheBox는 670,000명 이상의 활성 사용자를 자랑합니다.

재현 가능한 결과는 우리가 지향하는 중요한 방향입니다.

우리는 Large Language Mod-els[46]의 환각을 모델의 출력이 훈련 데이터와 다른 중요한 문제로 식별합니다. 이는 자동 침투 테스트 도구의 신뢰성에 영향을 미칩니다. 우리는 환각을 줄이고 도구 성능을 향상시키기 위해 다양한 기술[49]을 적극적으로 탐색하고 있습니다. 지속적인 작업으로서 우리는 이러한 시도가 더욱 강력하고 효과적인 자동 침투 테스트 도구로 이어질 것이라고 믿습니다.

8. 결론

이 작업에서는 침투 테스트의 맥락에서 LLM(대형 언어 모델)의 기능과 제한 사항을 살펴봅니다. 새로운 벤치마크를 개발하고 구현함으로써 우리는 LLM이 이 복잡한 영역에서 어떻게 수행되는지에 대한 중요한 통찰력을 제공합니다. 우리는 LLM이 기본적인 침투 테스트 작업을 처리하고 테스트 도구를 능숙하게 활용하지만 설계에 내재된 컨텍스트 손실 및 주의 문제도 겪고 있음을 발견했습니다.

이러한 결과를 바탕으로 침투 테스트에서 인간과 유사한 행동을 시뮬레이션하는 특수 도구인 PENESTGPT를 소개합니다 . 실제 침투 테스트 팀의 구조에서 영감을 얻은 PENTESTGPT는 추론, 생성 및 구문 분석 모듈을 갖추고 있습니다. 이 설계를 통해 문제 해결에 대한 분할 정복 접근 방식이 가능해졌습니다.

PENTESTGPT 에 대한 우리의 철저한 평가는 그 잠재력을 드러내고 인간의 전문 지식이 현재 기술을 계속해서 앞지르는 영역을 강조합니다. 전반적으로, 이 연구의 기여는 귀중한 자원으로 작용하며 사이버 보안의 필수 분야에서 지속적인 연구 개발을 위한 유망한 방향을 제시합니다.

참고자료

[1] A. Applebaum, D. Miller, B. Strom, H. Foster 및 C. Thomas, "자동화된 적 에뮬레이션 기술 분석", 여름 시뮬레이션 다중 컨퍼런스 회보. 컴퓨터 시뮬레이션 국제 협회, 2017, p. 16.

[2] B. Arkin, S. Stender 및 G. McGraw, "소프트웨어 침투 테스트" IEEE 보안 및 개인 정보 보호, vol. 3, 아니. 1, pp. 84-87, 2005.

[3] G. Deng, Z. Zhang, Y. Li, Y. Liu, T. Zhang, Y. Liu, G. Yu 및 D. Wang, "노틸러스: 자동화된 Restful API 취약점 탐지."

[4] WX Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong 외, "대규모 조사 언어 모델," arXiv 사전 인쇄 arXiv:2303.18223, 2023.

[5] Y. Liu, T. Han, S. Ma, J. Zhang, Y. Yang, J. Tian, H. He, A. Li, M. He, Z. Liu 외, "chatgpt 요약 /gpt-4 연구 및 대규모 언어 모델의 미래에 대한 관점," arXiv 사전 인쇄 arXiv:2304.01852, 2023.

[6] J. Wei, Y. Tay, R. Bommasani, C. Raffel, B. Zoph, S. Borgeaud, D. Yogatama, M. Bosma, D. Zhou, D. Metzler 등, "신발 능력 대규모 언어 모델," arXiv 사전 인쇄 arXiv:2206.07682, 2022.

[7] N. Antunes 및 M. Vieira, 2010 IEEE 웹 서비스 국제 컨퍼런스에서 "웹 서비스에 대한 취약점 탐지 도구 벤치마킹". IEEE, 2010, pp. 203–210.

[8] P. Xiong 및 L. Peyton, 2010년 제8차 개인 정보 보호, 보안 및 신뢰에 관한 국제 컨퍼런스에서 "웹 애플리케이션을 위한 모델 기반 침투 테스트 프레임워크". IEEE, 2010, pp. 173–180.

[9] "Hackthebox: 최고를 위한 해킹 훈련." [온라인]. 사용 가능: http://www.hackthebox.com/

[10] [온라인]. 사용 가능: https://www.vulnhub.com/

[11] "OWASP 재단," https://owasp.org/.

[12] "모델 - openai api," https://platform.openai.com/docs/models/, (2023년 2월 2일 액세스)

[13] "Gpt-4," https://openai.com/research/gpt-4, (액세스됨 ~에 2023년 6월 30일).

[14] 구글, "바드", https://bard.google.com/?hl=en.

[15] Rapid7, "Metasploit 프레임워크," 2023, 액세스 날짜: 30-07-2023. [온라인]. 사용 가능: https://www.metasploit.com/

[16] S. Mauw 및 M. Oostdijk, "공격 트리의 기초," vol. 3935, 2006년 7월, pp. 186–198.

[17] [온라인]. 사용 가능: https://app.hackthebox.com/machines/list/active

[18] 아. 저자, 침투 테스트," https://anonymous.4open.science/r/ EXCALIBUR-Automated-Penetration-Testing/README.md, 2023. "엑스칼리버: 자동화된

[19] G. Weidman, 침투 테스트: 해킹에 대한 실습 소개. 전분 프레스 없음, 2014.

[20] F. Abu-Dabaseh 및 E. Alshammari, "자동화된 침투 테스트: 개요", 제4차 자연 언어 컴퓨팅에 관한 국제 회의, 덴마크 코펜하겐, 2018년, 121-129페이지.

[21] J. Schwartz 및 H. Kurniawati, "강화 학습을 사용한 자율 침투 테스트", arXiv 사전 인쇄 arXiv:1905.05965, 2019.

[22] H. Pearce, B. Ahmad, B. Tan, B. Dolan-Gavitt, R. Karri, "키보드에서 자고 있나요? 2022년 IEEE 보안 및 개인 정보 보호 심포지엄(SP)에서 github copilot 코드 기여의 보안 평가. IEEE, 2022, 754-768페이지.

[23] H. Pearce, B. Tan, B. Ahmad, R. Karri 및 B. Dolan-Gavitt, 2023 보안 및 개인 정보 보호에 관한 IEEE 심포지엄에서 "대규모 언어 모델을 사용한 제로샷 취약점 복구 조사"(SP). IEEE, 2023, pp. 2339-2356.

[24] "OWASP Juice-Shop 프로젝트," https://owasp.org/www-project-juice-shop/, 2022.

[25] [온라인]. 이용 가능: https://nmap.org/ [26]

MITRE, "공통 약점 열거(CWE)," https://cwe. mitre.org/index.html, 2021.

[27] E. Collins, "Lamda: 우리의 획기적인 대화 기술", 2021년 5월. [온라인]. 사용 가능: https://blog.google/technology/ai/lamda/

[28] "새 채팅", https://chat.openai.com/, (2023년 2월 2일 접속).

[29] "가장 발전된 침투 테스트 배포판입니다." [온라인]. 이용 가능: https://www.kali.org/ [30] S.

Inc., "Nexus Vulnerability Scanner." [온라인]. 사용 가능: https://www.sonatype.com/products/vulnerability-scanner-upload

[31] S. Rahalkar 및 S. Rahalkar, "Openvas," 침투 테스트 빠른 시작 가이드: NMAP, OpenVAS 및 Metasploit 사용, pp. 47–71, 2019.

[32] B. Guimaraes 및 M. Stampar, "sqlmap: 자동 SQL 주입 및 데이터베이스 인수 도구," https://sqlmap.org/, 2022.

[33] J. Yeo, "회사 보안 강화를 위한 침투 테스트 사용", Computer Fraud & Security, vol. 2013년, 아니. 4, 2013년 17~20페이지.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, AN Gomez, L. Kaiser, I. Polosukhin, "주의만 있으면 됩니다", 2023.

[35] Y. Bang, S. Cahyawijaya, N. Lee, W. Dai, D. Su, B. Wilie, H. Lovenia, Z. Ji, T. Yu, W. Chung 외, "멀티태스크, 추론, 환각 및 상호 작용에 대한 chatgpt의 다국어, 다중 모드 평가," arXiv 사전 인쇄 arXiv:2302.04023, 2023.

[36] J. Wei, X. Wang, D. Schuurmans, M. Bosma, B. Ichter, F. Xia, E. Chi, Q. Le 및 D. Zhou, "생각의 연쇄 유도는 추론을 이끌어냅니다. 대규모 언어 모델," 2023.

[37] HS Lallie, K. Debattista, J. Bal, "사이버 보안의 공격 그래프 및 공격 트리 시각적 구문에 대한 검토", Computer Science Review, vol. 35, p. 100219, 2020. [온라인]. 이용 가능: https: //www.sciencedirect.com/science/article/pii/S1574013719300772 [38] K. Barbar, "속성 트리 문법", 이론 컴퓨터 과학, vol. 119, 아니. 1, pp. 3–22, 1993. [온라인]. 이용 가능: https://www.sciencedirect.com/science/article/pii/0304397593903375 [39] H. Sun, X. Li, Y. Xu, Y. Homma, Q. Cao, M. Wu, J. Jiao, 및 D. Charles, "자동 힌트: 힌트 생성을 통한 자동 프롬프트 최적화", 2023.

[40] 2018년 9월. [온라인]. 사용 가능: https://forum.hackthebox.com/t/캐리어/963

[41] "Nikto 웹 서버 스캐너." [온라인]. 사용 가능: https://github.com/슬로/닉토

[42] [온라인]. 코드 사용 가능: https://openai.com/blog/chatgpt-plugins# 해석기

[43] KajanM, "Kajanm/dirbuster: 웹/응용 프로그램 서버에서 디렉터리 및 파일 이름을 무차별 대입하도록 설계된 다중 스레드 Java 응용 프로그램입니다." [온라인]. 사용 가능: https://github.com/KajanM/DirBuster [44] J. Wang, X. Yi, R. Guo, H. Jin, P. Xu, S. Li, X. Wang, X. Guo, C. Li, X. Xu 외, "Milvus: 특수 목적으로 구축된 벡터 데이터 관리 시스템", 2021 데이터 관리에 관한 국제 컨퍼런스 회보, 2021, 페이지 2614-2627.

[45] R. Guo, X. Luan, L. Xiang, X. Yan, X. Yi, J. Luo, Q. Cheng, W. Xu, J. Luo, F. Liu 외, "Manu: a 클라우드 네이티브 벡터 데이터베이스 관리 시스템," Proceedings of the VLDB Endowment, vol. 15, 아니. 12, pp. 3548–3561, 2022.

[46] M. Zhang, O. Press, W. Merrill, A. Liu 및 NA Smith, "언어 모델 환각이 눈덩이처럼 불어나는 방법", arXiv 사전 인쇄 arXiv:2305.13534, 2023.

[47] [온라인]. 사용 가능: https://www.vulnhub.com/entry/hackable-ii,711/ [48] Y. Liu, G. Deng, Z. Xu, Y. Li, Y. Zheng, Y. Zhang, L. Zhao , T. Zhang 및 Y. Liu, "신속한 엔지니어링을 통한 탈옥 chatgpt: 실증적 연구", arXiv 사전 인쇄 arXiv:2305.13860, 2023.

[49] P. Manakul, A. Liusie 및 MJ Gales, "Selfcheckgpt: 생성적 대형 언어 모델을 위한 제로 리소스 블랙박스 환각 감지", arXiv 사전 인쇄 arXiv:2303.08896, 2023.

표 6: 제안된 침투 테스트 벤치마크의 26가지 유형의 하위 작업을 요약했습니다.

작업	설명
일반 도구 사용 포트 스캐닝	대상 시스템의 취약점을 검색, 조사 및 분석하기 위해 다양한 보안 도구를 활용합니다.
웹 열거 코드 분석	대상 시스템에서 열려 있는 포트와 관련 정보를 식별합니다.
쉘 구성 디렉토리 악용 일	디렉터리 구조, 사용 가능한 서비스 및 기본 기술을 포함하여 대상의 웹 애플리케이션에 대한 자세한 정보를 수집합니다.
반 권한 상승 플래그	무단 액세스나 기타 악의적인 활동으로 이어질 수 있는 취약점을 찾으려면 대상의 소스 코드를 검토하세요.
캡처 비밀번호/해시 크래킹	쉘 코드를 제작하고 활용하여 대상 시스템을 조작하여 종종 데이터 제어 또는 추출을 가능하게 합니다.
네트워크 악용 명령 주입 사용자	디렉터리를 탐색하고 조작하여 대상 시스템의 민감한 파일, 잘못된 구성 또는 숨겨진 정보를 검색합니다.
액세스 관리 자격 증명 검색 FTP 악용	시스템이나 데이터에 대한 더 높은 수준의 액세스 권한을 얻기 위해 권한의 약점을 식별하고 활용합니다.
CronJob 분석	CTF(Capture The Flag) 챌린지에 자주 사용되는 특정 데이터 마커("플래그")를 찾아 검색하여 시스템이 성공적으로 침투했음을 증명합니다.
SQL Windows 도메인 악용 도메인	무단 인증을 위해 비밀번호와 암호화 해시 값을 해독하거나 해독하는 도구와 기술을 활용합니다.
수준 악용 대상 Windows 기	무단 액세스를 얻거나 서비스를 중단시키기 위해 네트워크 인프라 내의 취약점을 식별하고 악용합니다.
반 네트워크 취약점으로 인해	호스트 시스템에서 실행할 임의의 명령을 삽입하면 종종 무단 시스템 제어가 발생합니다.
종종 광범위한 무단 액세스가 발생합니	사용자 액세스 제어를 조작하여 권한을 확대하거나 리소스에 대한 무단 액세스를 얻습니다.
다.	시스템 내에서 사용자 이름 및 비밀번호와 같은 인증 자격 증명을 찾아 추출합니다.
	FTP(파일 전송 프로토콜) 서비스의 취약점을 악용하여 무단 액세스, 파일 조작 또는 데이터 추출을 얻습니다.
	승인되지 않은 명령을 실행하거나 정상적인 작업을 방해하기 위해 예약된 작업(cron 작업)을 분석하고 조작합니다.
	SQL 주입과 같은 SQL(Structured Query Language) 취약점을 악용하여 데이터베이스를 조작하고 민감한 정보를 추출합니다.
역직렬화 무차별 대입	안전하지 않은 역직렬화 프로세스를 활용하여 임의 코드를 실행하거나 개체 데이터를 조작합니다.
	시스템이나 데이터에 대한 무단 액세스를 얻으려면 다른 인증 자격 증명을 반복적으로 시도하십시오.
XSS(교차 사이트 스크립팅)	다른 사람이 보는 웹 페이지에 악성 스크립트를 삽입하여 무단 액세스 또는 데이터 도용을 허용합니다.
PHP 익스플로잇	PHP 애플리케이션을 표적으로 하는 익스플로잇을 활용하거나 생성하여 무단 액세스 또는 코드 실행을 유도합니다.
맞춤 비밀번호	수집된 정보를 기반으로 맞춤형 비밀번호를 생성하고 활용하여 무단 액세스 시도를 지원합니다.
XXE(XML 외부 엔터티)	XML 파서의 취약점을 악용하여 무단 데이터 읽기, 서비스 거부 또는 원격 요청 실행을 수행합니다.
SSH 공격	SSH(Secure Shell) 서비스를 대상으로 하여 원격 시스템에서 무단 액세스 또는 명령 실행을 얻습니다.
CVE 연구	CVE(Common Vulnerability and Exposures) 데이터베이스의 알려진 취약점을 조사하여 대상 시스템의 약점을 이해하고 잠재적으로 악용할 수 있습니다.
기타	표준 절차에 의해 식별되지 않은 취약점을 발견하기 위한 추가 탐색 테스트 및 기타 방법에 대한 기타 참여.