

LLM for Cybersecurity

from
Threat Intelligence Analysis
to
Penetration Testing

Assistant Professor MA Yunshan
SCIS, Singapore Management University

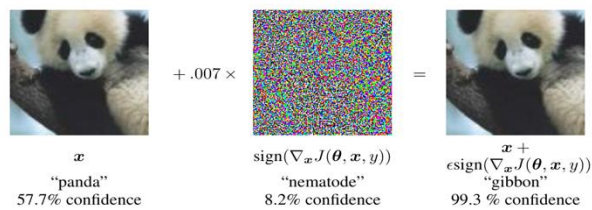
2017–2021, PhD student in NExT++
2022–2024, Postdoc in NExT++
2025–now, A/P in SCIS SMU



AI Security: protecting AI systems

AI systems face familiar security risks.

Adversarial inputs



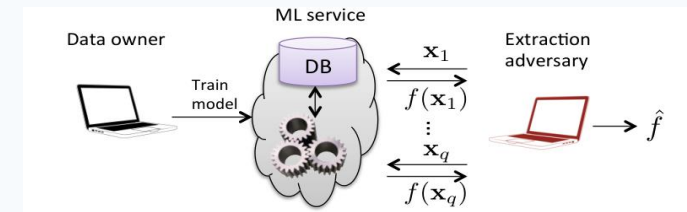
Small perturbation, changed prediction

Poisoning / backdoors



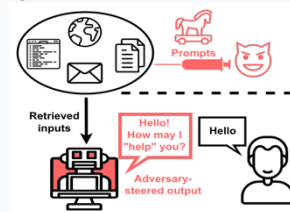
A trigger changes learned behavior

Privacy / model extraction



Queries can expose data or capability

Prompt injection



Untrusted content becomes instructions

Jailbreaks



An input bypasses safety constraints

Agent / multi-agent failures



Coordination



Verification



Unsafe actions or unchecked outcomes

Here, the AI system itself needs protection. Today we turn to conventional cybersecurity.

What is cybersecurity: a longstanding challenge

Major cybersecurity events in the past decades

1988

Morris worm

莫里斯蠕虫

An early Internet-wide worm outbreak.



2000

ILOVEYOU

爱虫病毒

An email attachment spreads malware.



2007

Panda Burning Incense

熊猫烧香

Malware spreads across personal computers.



2010

Stuxnet

震网

Malware targets industrial control systems.



2017

WannaCry / NotPetya

想哭 / NotPetya 攻击

Services and operations are disrupted.

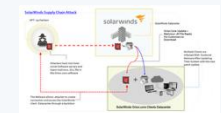


2020

SolarWinds

太阳风供应链入侵

Trusted updates spread the compromise.



What are the objectives to protect (attack) in cybersecurity

Protect digital and physical systems, their software, and the identities that connect them.

Desktop operating systems



Windows



Linux



macOS

Mobile operating systems



Android



iOS

Web and data services



Web

NGINX

Nginx



Database

Networks and identity



TCP/IP



DNS / TLS



Credentials

Software and cloud



Code / build



Docker



Kubernetes

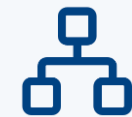
Industrial control (ICS)



PLC



SCADA



OT networks

Some popular cyber threat types

Familiar threats exploit software, human trust, and shared infrastructure.



Malware

Worms and viruses spread malicious code.



Phishing

Deceptive messages obtain access or credentials.



Ransomware

Encryption and extortion disrupt operations.



Exploitation

Software flaws provide an entry point.



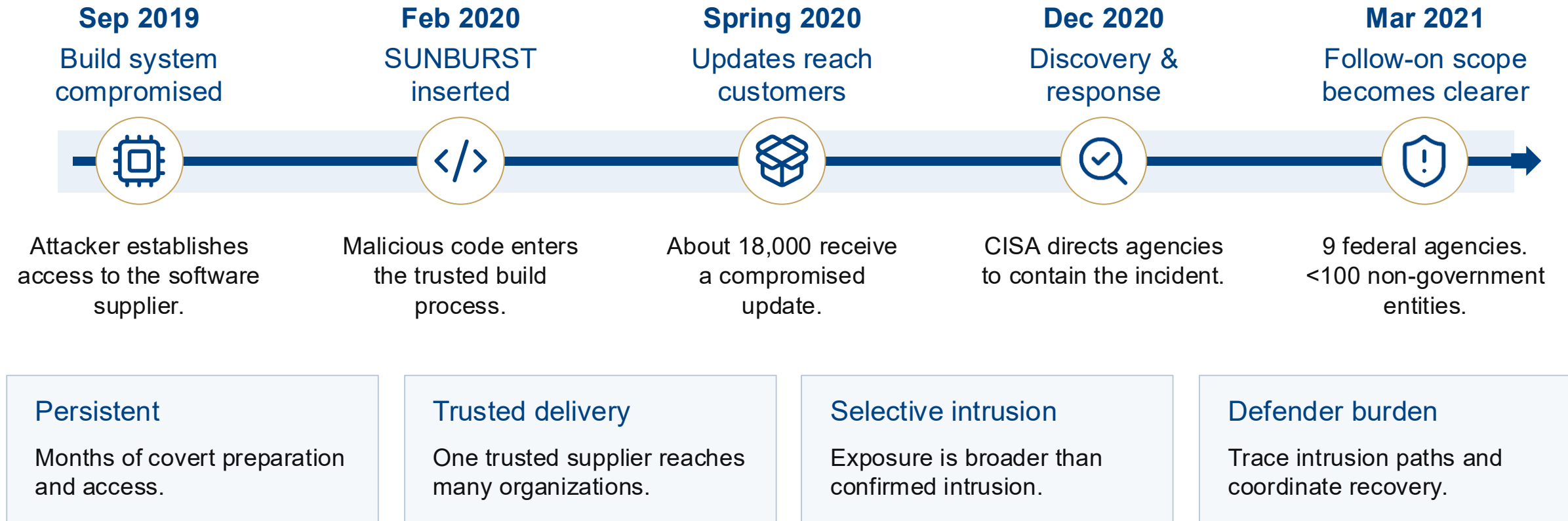
Supply-chain attacks

Trusted products or updates carry the compromise.

Individual attack methods keep advancing. More importantly, APTs combine multiple methods into sustained, coordinated campaigns.

APT: Advanced Persistent Threat

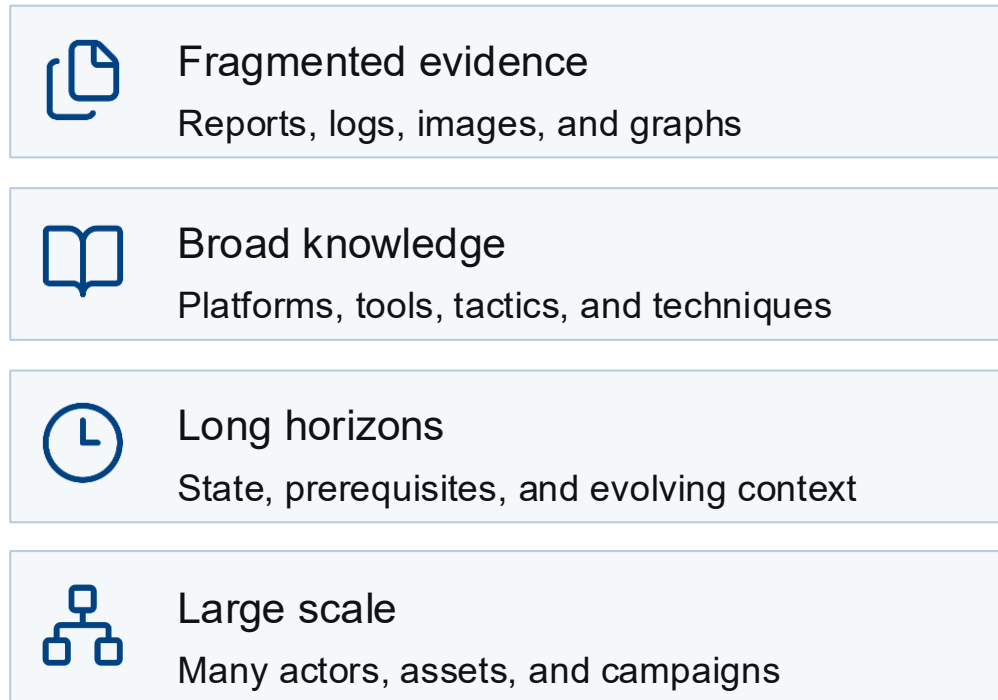
One example APT: SolarWinds/SUNBURST (太阳风供应链入侵), 2019–2021.



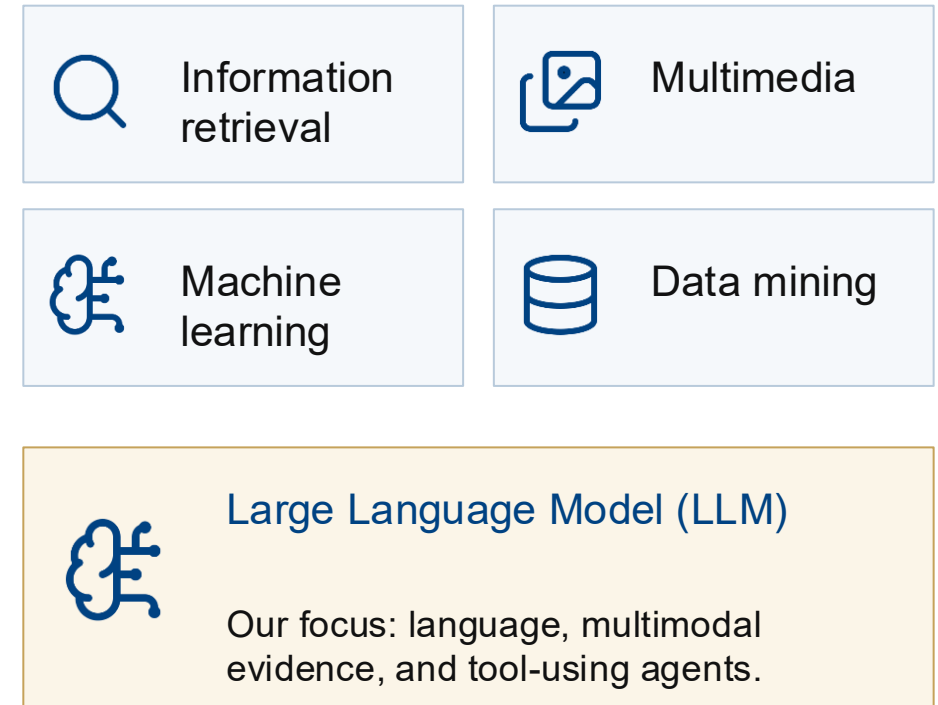
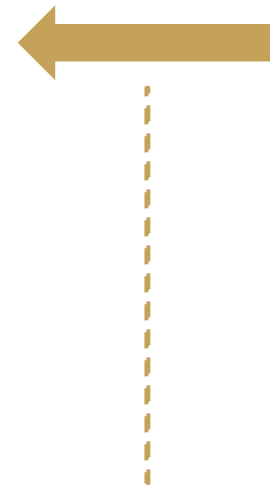
AI for Cybersecurity

As AI researchers, how can we use AI techniques to address these cybersecurity challenges?

Cybersecurity challenges



AI techniques



The focus of this talk

LLM for Cybersecurity: the outline of this talk

1

Threat Intelligence Analysis

Information extraction
Knowledge graph construction
Event detection and forecasting
Temporal analysis
Report generation

2

Attack Surface Management

Representation learning
Contrastive learning
Graph reasoning
Graph RAG

3

Penetration Testing

Agentic systems
Agent post training
Dynamic environment construction

LLM for Cybersecurity: Threat Intelligence Analysis

1

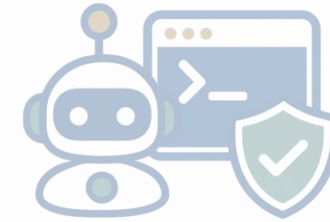
Threat Intelligence Analysis

Information extraction
Knowledge graph construction
Event detection and forecasting
Temporal analysis
Report generation

2

Attack Surface Management

Representation learning
Contrastive learning
Graph reasoning
Graph RAG

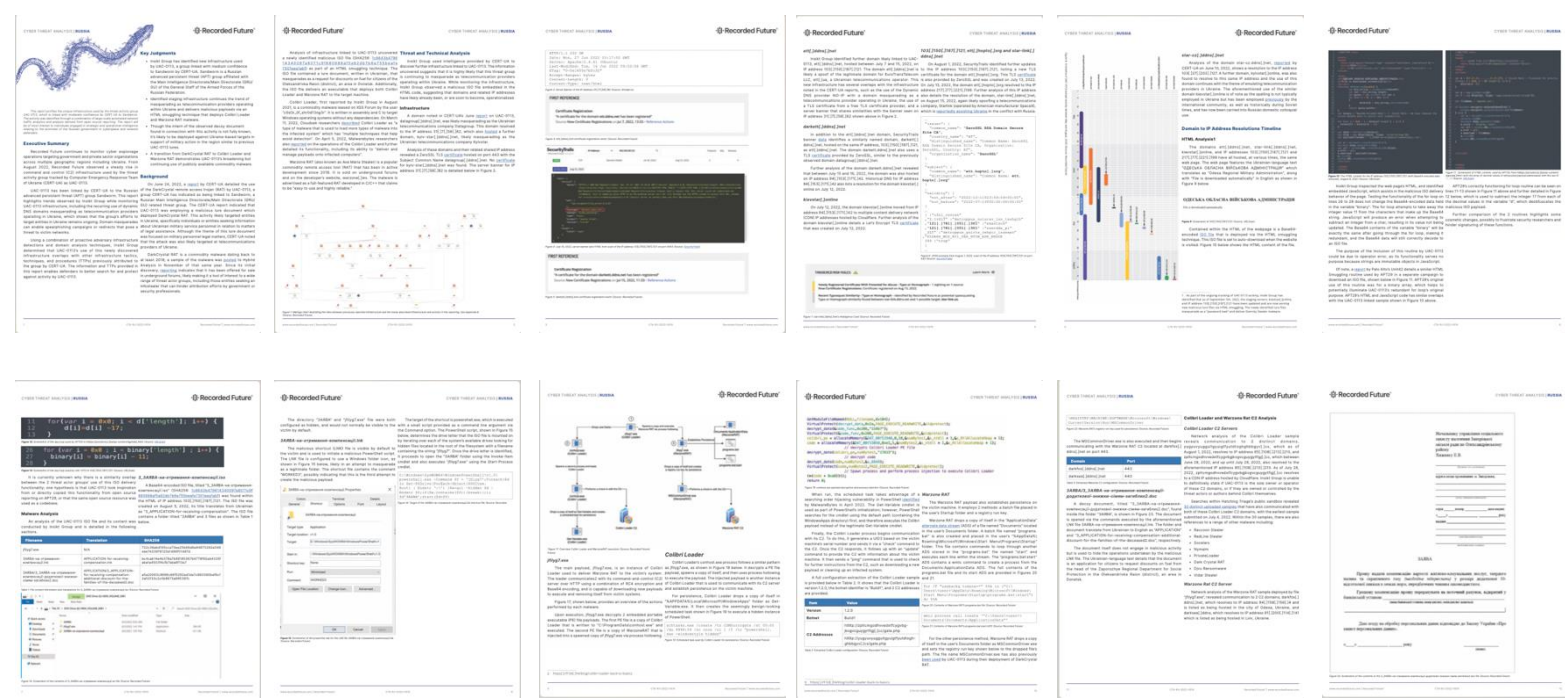
3

Penetration Testing

Agentic systems
Agent post training
Dynamic environment construction

Cyber threat intelligence (CTI):
Collect, analyze, and apply knowledge about threats,
adversaries, and attack methods to support security
decisions.

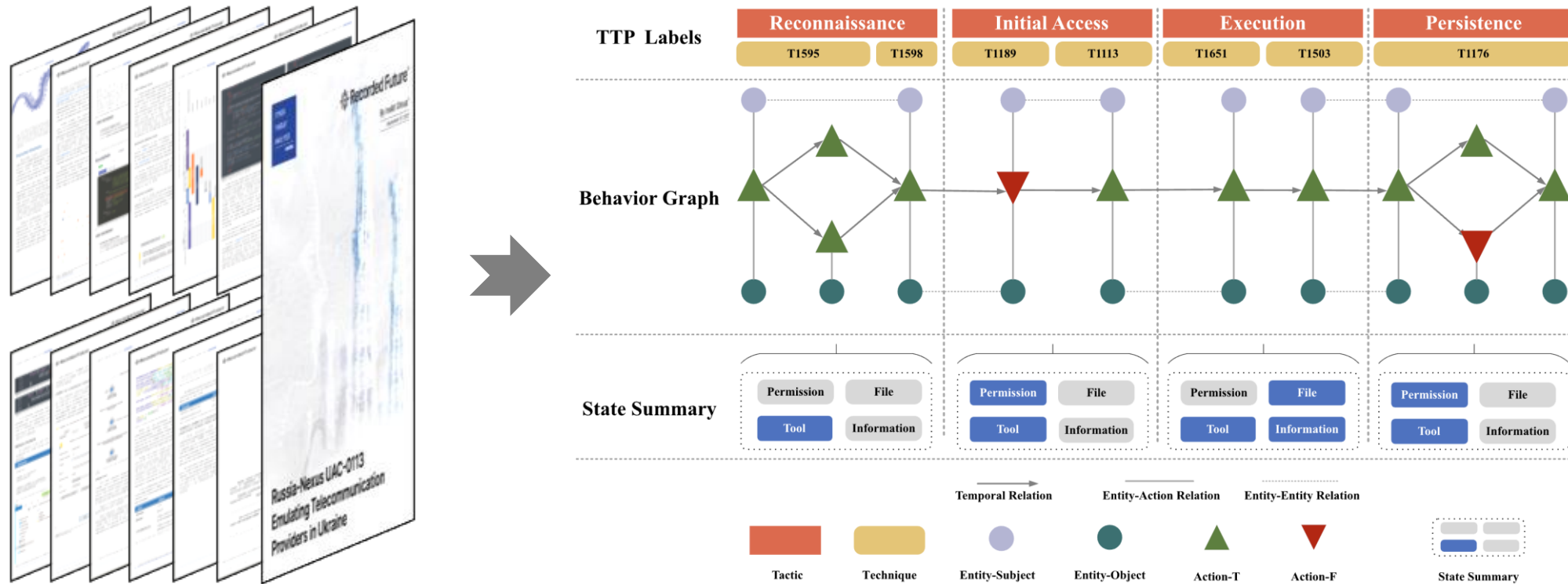
- CTI report:
- A document describing potential or observed threats, with evidence that helps organizations investigate and respond.



[2] Recorded Future / Insikt Group. Russia-Nexus UAC-0113 Emulating Telecommunication Providers in Ukraine. 2022.

Attack graph construction

- Task: turn an unstructured CTI report into an attack knowledge graph.



Why use LLMs for threat intelligence?

LLMs can generalize beyond rules and labeled examples.



Rules and patterns

Precise for known forms.
Brittle under wording changes.



Task-specific models

Learn entities and relations.
Require labeled examples.



LLM understanding

Use instructions and context.
Generalize, then verify.

Long narratives Implicit actions Evolving terminology Limited annotations

AttackKG+: from reports to attack graphs

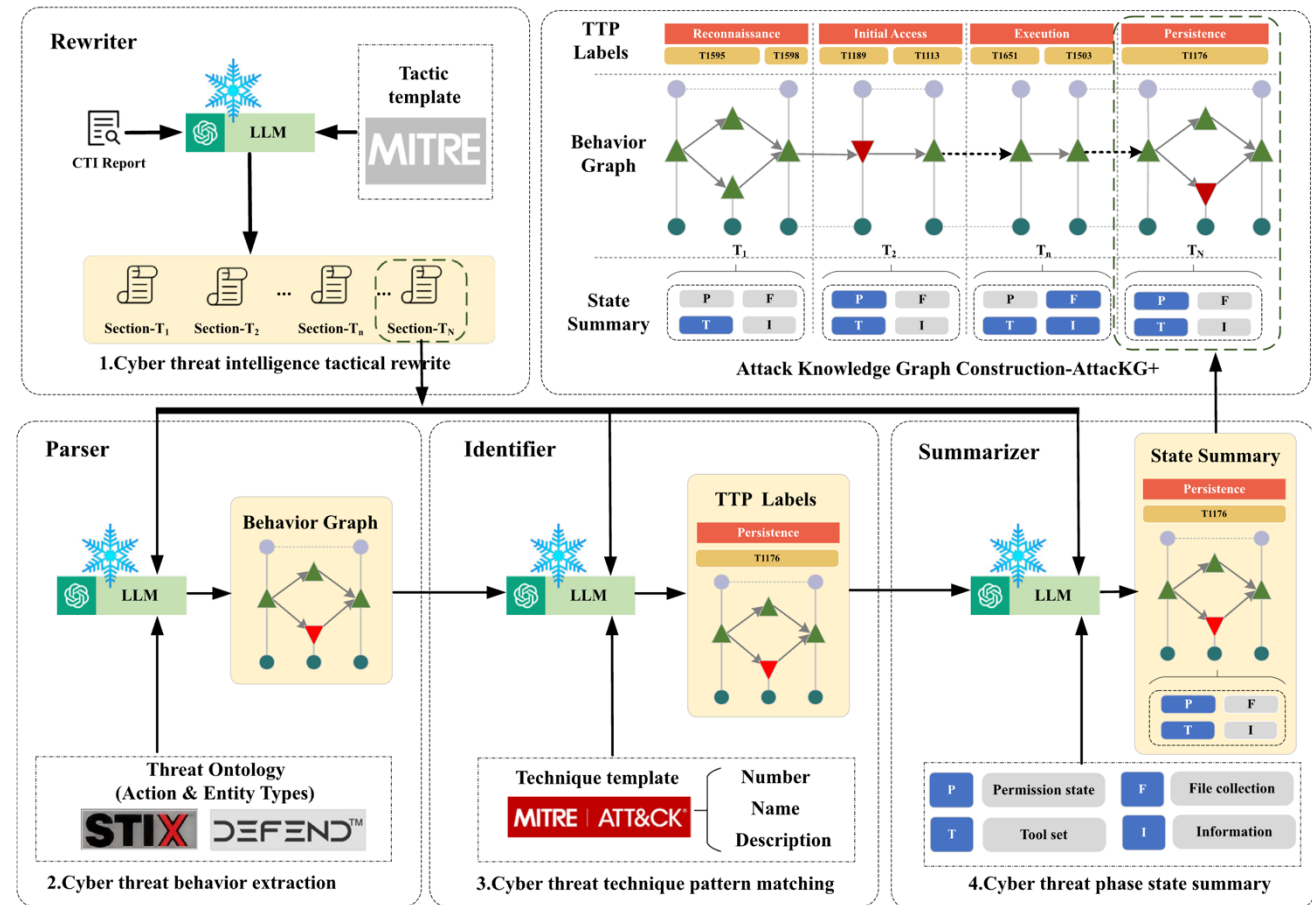
- Our proposal: use LLMs to construct attack graphs from reports.

Motivation

- Use semantic understanding
 - Handle diverse report language
- Reduce task-specific annotation
 - Use zero-shot instructions and few-shot examples

Key modules

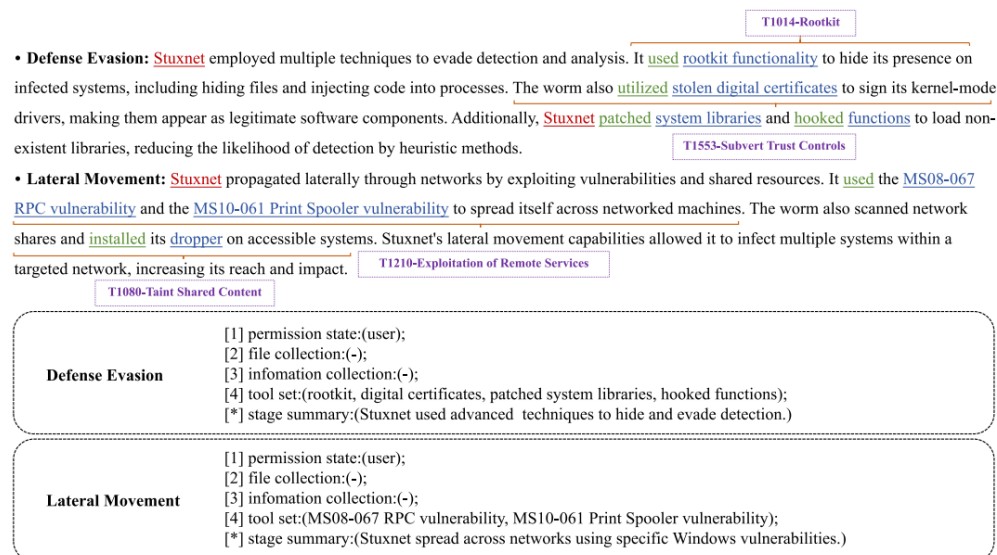
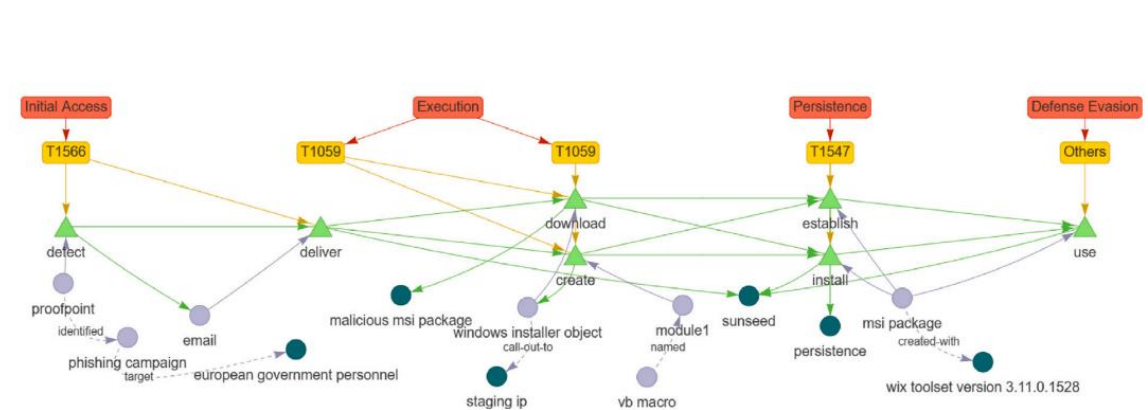
- Rewriter
- Parser
- Identifier
- Summarizer



AttackKG+: an intuitive attack graph example

Case study: recover a structured view of a complex attack.

- An attack on SK Communications illustrates how the framework connects actions and their objects.
- The multilevel graph makes the attack progression easier to inspect.



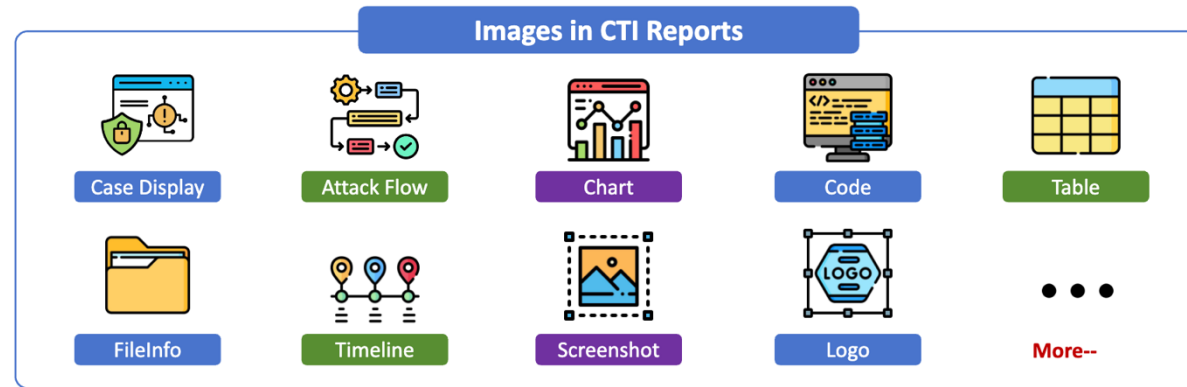
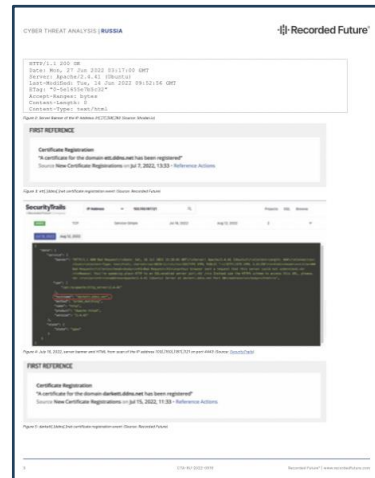
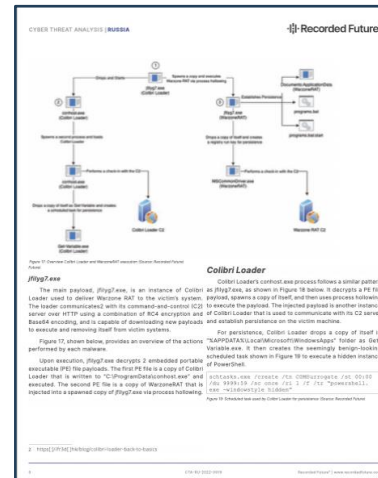
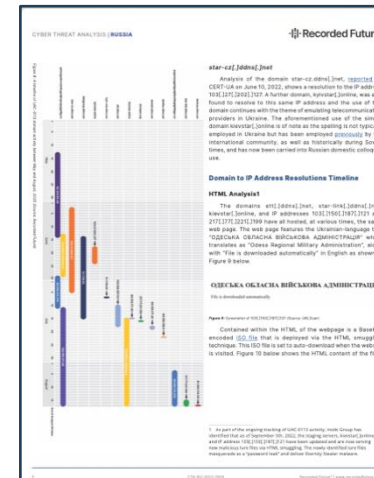
An attack case against SK Communications

Example of AttackKG+ extraction (Stuxnet)

MM-AttackKG: from text to multimodal evidence

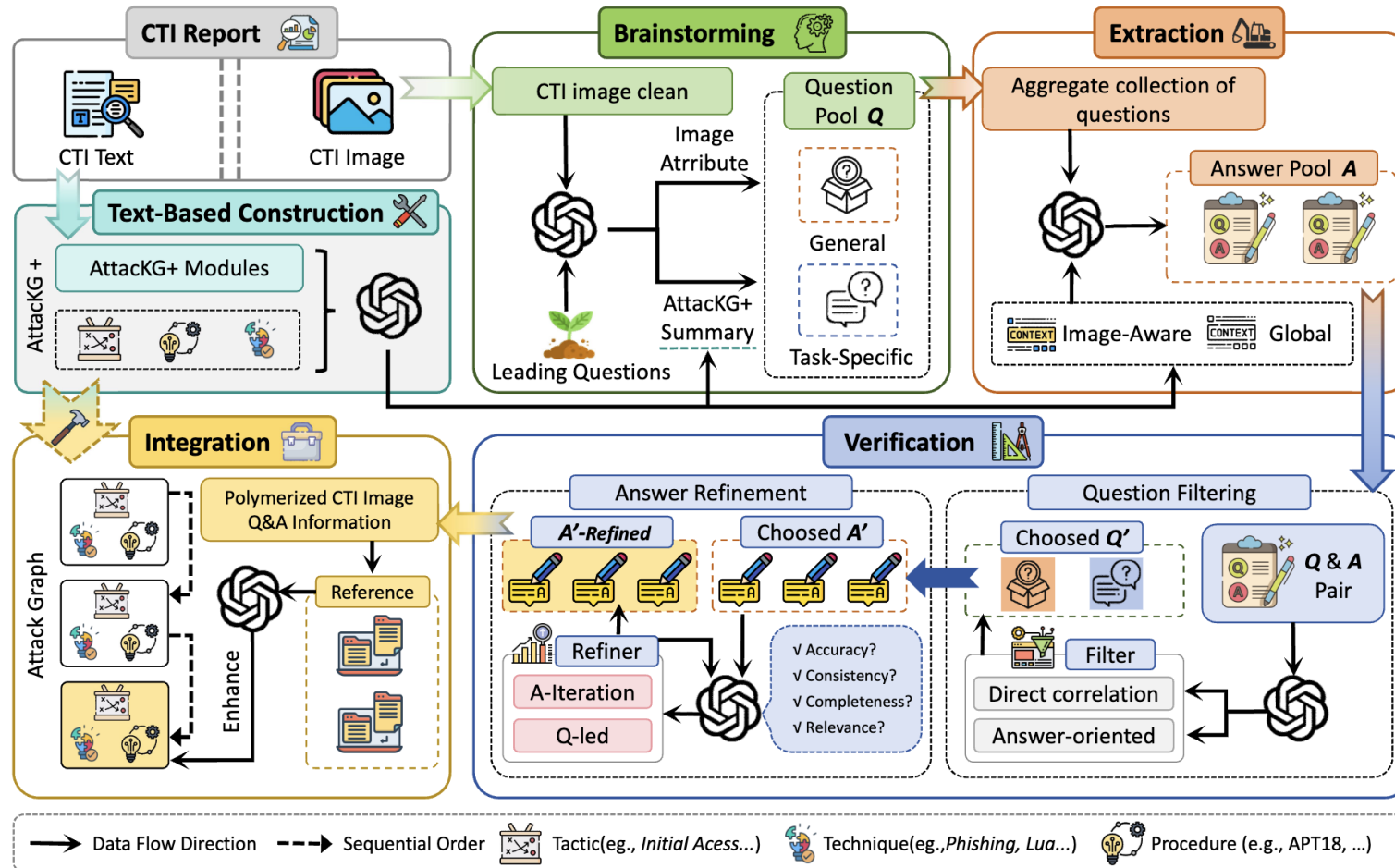
MM-AttackKG recovers attack evidence from both report text and images.

- Motivation: text-only preprocessing discards evidence embedded in images.



MM-AttackKG: integrating visual evidence

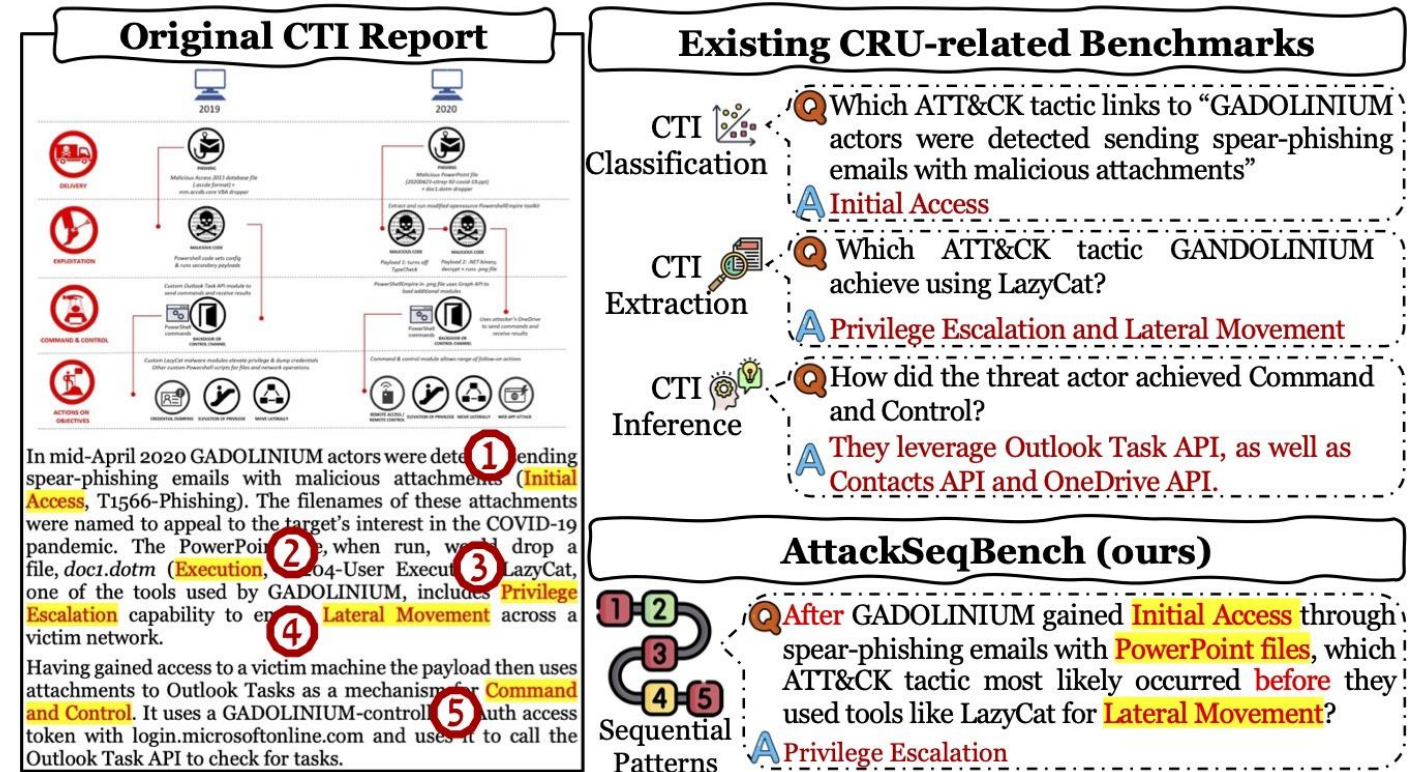
MM-AttackKG recovers attack evidence from both report text and images.



AttackSeqBench: from graphs to sequence reasoning

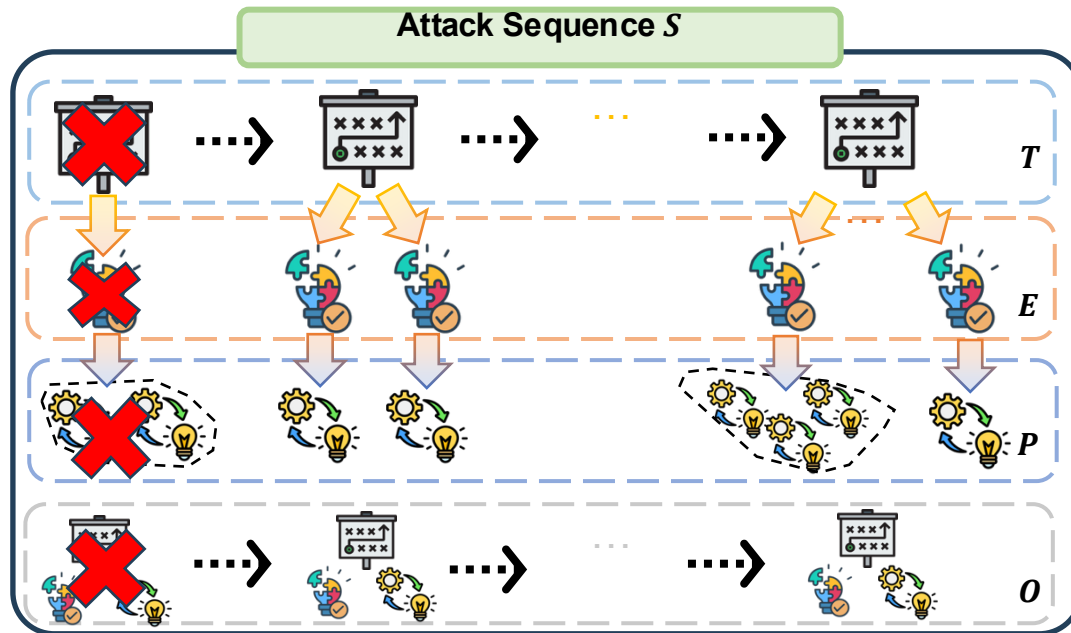
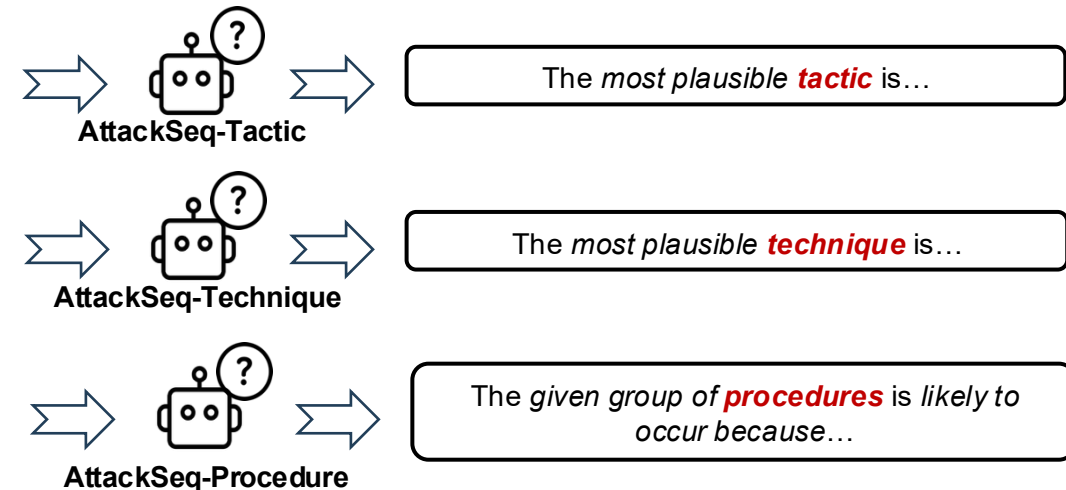
Motivation

- Attacks unfold through dependent steps
 - Extracted facts do not establish dependencies or missing steps
- Extend attack understanding
 - Reason about sequences and predict plausible missing behavior



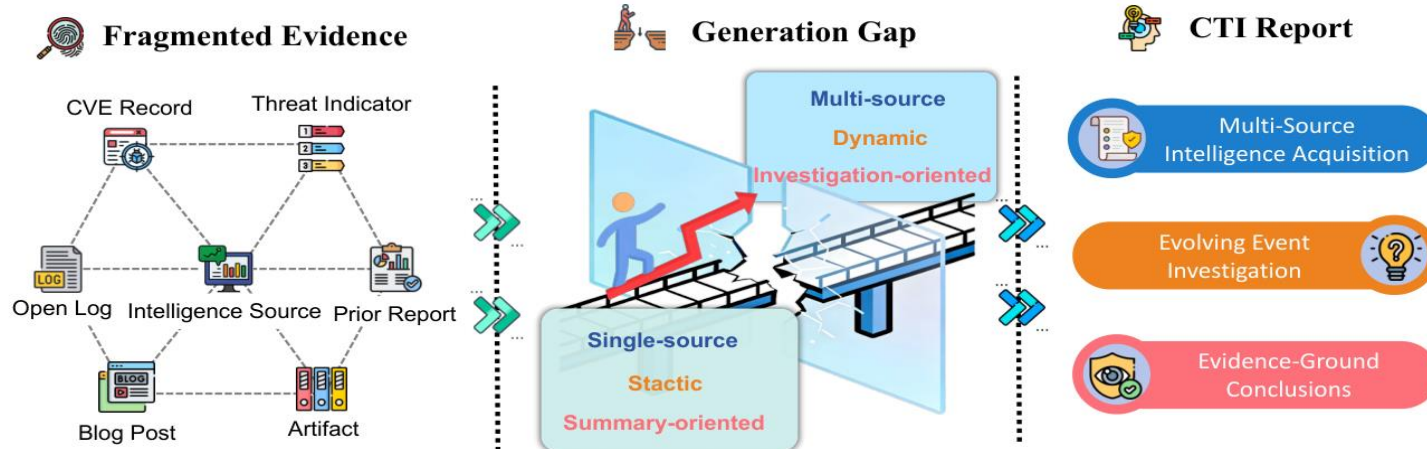
AttackSeqBench: reasoning about missing steps

- Attack sequence prediction
- Three QA tasks at tactic, technique, and procedure levels.
- Infer the most plausible missing TTP from the remaining sequence.



CTI-ReGen: from understanding to report generation

Report generation requires an evolving investigation.

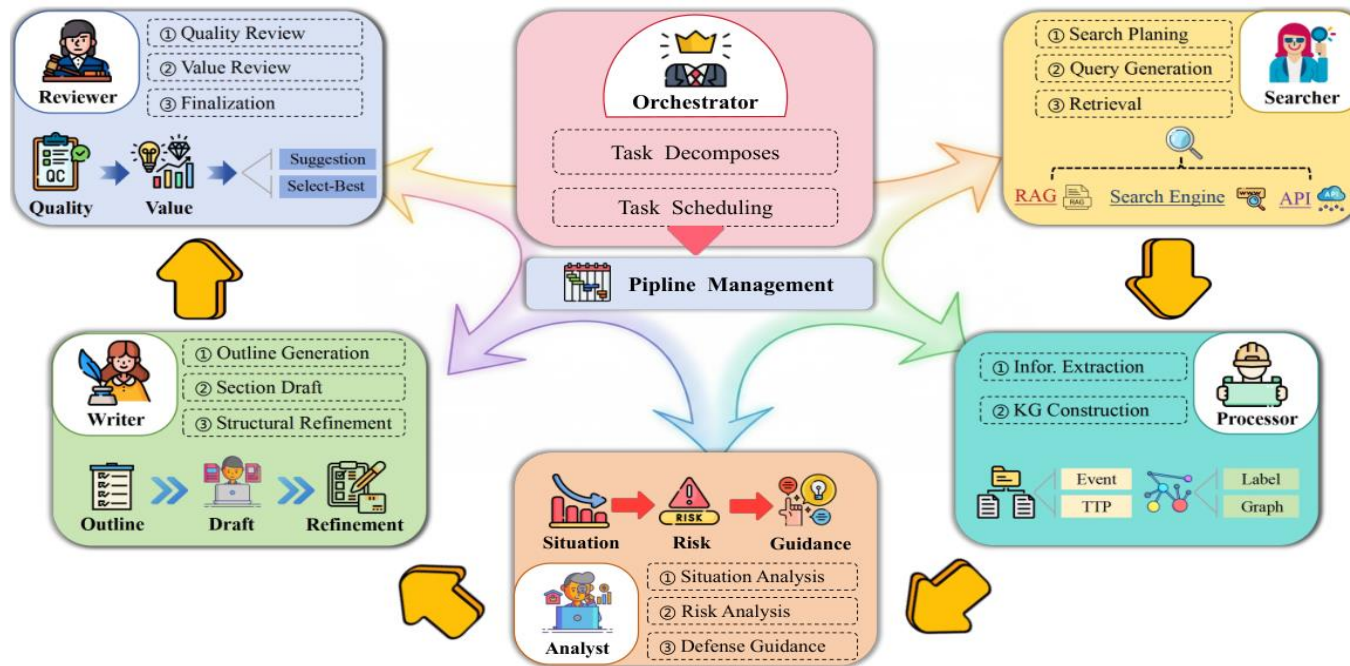


- Fragmented input
 - Evidence arrives from reports, advisories, artifacts, and updates.
- Evolving understanding
 - New information may revise the event rather than merely add detail.
- Defensible output
 - Conclusions need sources, uncertainty, and defensive relevance.

This motivates a shared event model between acquisition, reasoning, writing, and review.

CTI-ReGen: a shared event model

Agents need a shared model of the event.

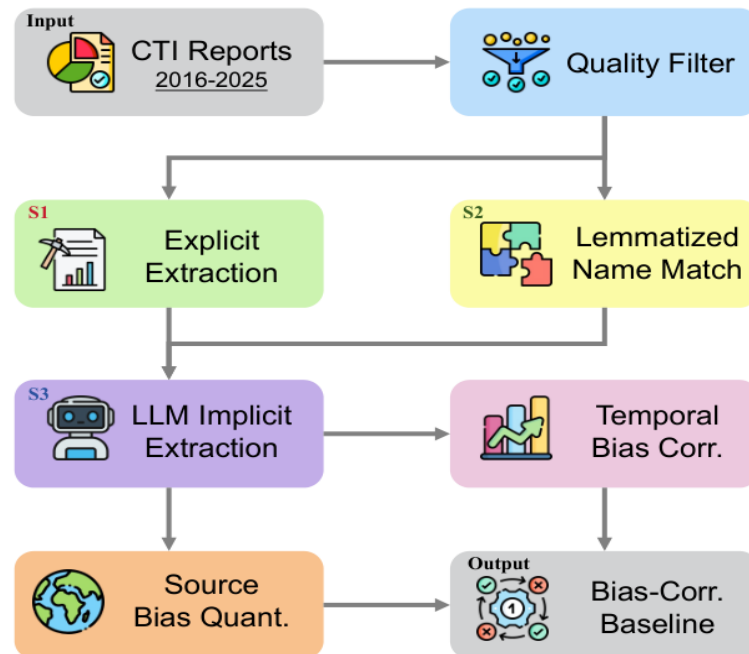


- The gap
 - Independent summaries cannot maintain one coherent investigation.
- The response
 - A shared event model connects evidence, relations, and provenance.
- Why this matters
 - Generation follows an evolving understanding that can be checked.

The graph organizes evidence, while original source artifacts remain the verification anchor.

TTPDecade: a decade of threat intelligence

The contribution is a set of findings about the reported threat landscape.

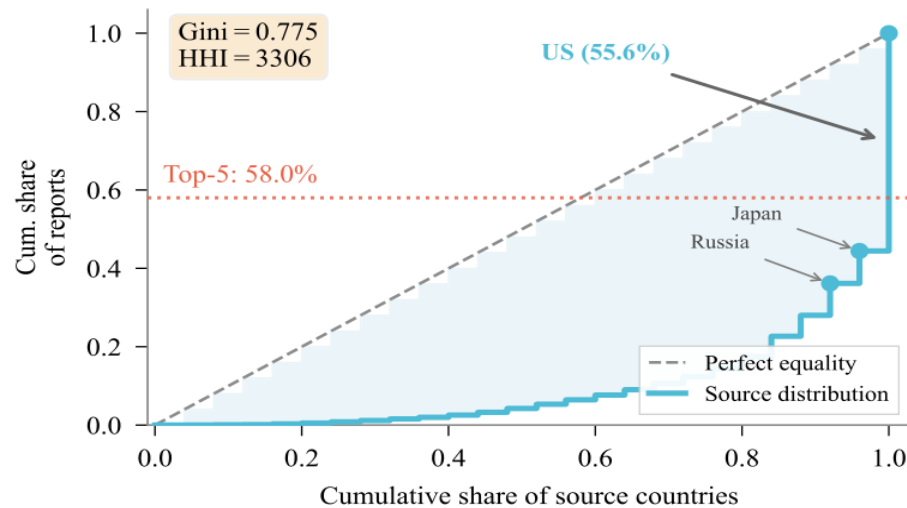
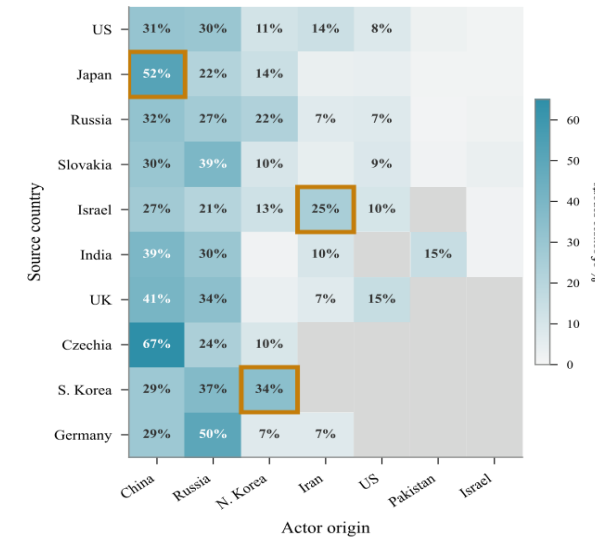


- A decade of evidence
 - 7,587 reports in 2016–2025, within a collected corpus of 8,336.
- Findings to follow
 - Vocabulary growth, reporting concentration, recurring attack paths, and codification lag.
- Why the findings matter
 - Raw frequency can misdirect defensive priorities unless the observation process is understood.

The central question is which patterns survive changes in vocabulary, sources, and extraction.

TTPDecade: reporting bias shapes the landscape

Publisher geography and language affect which threats appear common.



55.6% from US sources

94.2% in English

Among reports with attributable publisher country.

Language coverage leaves systematic blind spots.

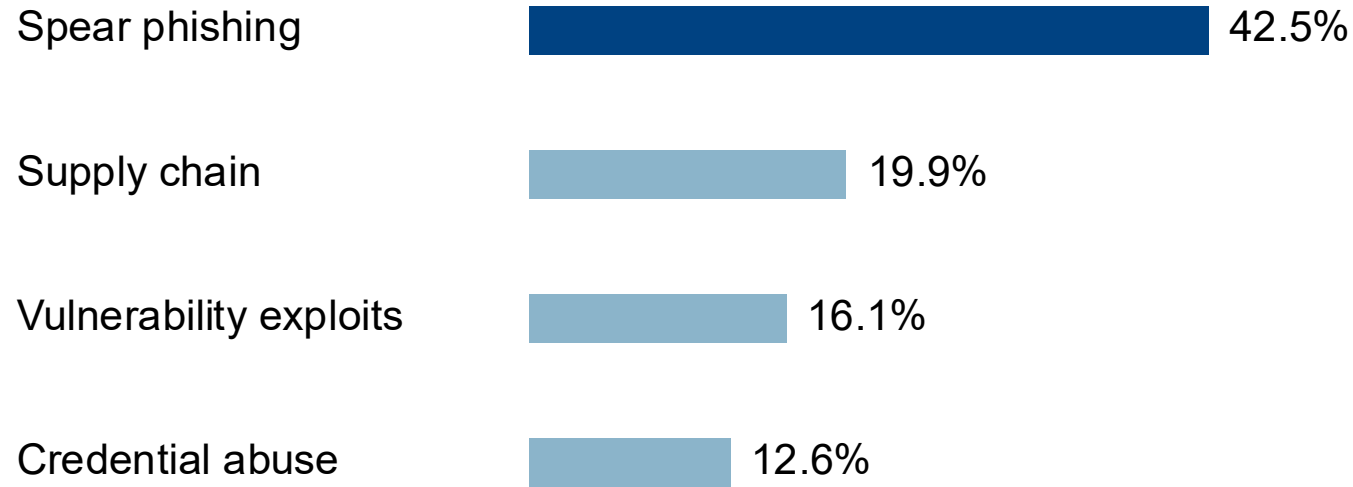
• Regional reporting focus

– Sources emphasize different attributed actor sets.

Diversify sources and inspect source–actor coverage before turning frequency into a priority.

TTPDecade: familiar entry paths

These percentages describe mapped Initial Access mentions in the corpus.



- A recognizable attack surface
 - Defenses still need to cover people, trusted software, exposed services, and identities.
- A practical use of CTI
 - Select relevant test scenarios and detection coverage, then check local exposure.

Use the corpus to inform coverage of established entry paths, with source-aware interpretation.

LLM for Cybersecurity: Attack Surface Management

1



Threat Intelligence Analysis

Information extraction
Knowledge graph construction
Event detection and forecasting
Temporal analysis
Report generation

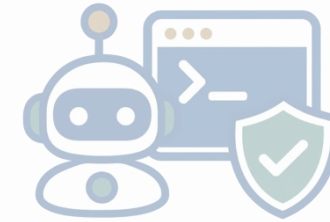
2



Attack Surface Management

Representation learning
Contrastive learning
Graph reasoning
Graph RAG

3



Penetration Testing

Agentic systems
Agent post training
Dynamic environment construction

What is attack surface management?

Attack surface management links discovery to remediation.



Discover

Domains, cloud endpoints,
web apps, and services



Attribute

Owner, operator,
and dependencies



Prioritize

Exposure, consequence,
and remediation



Monitor

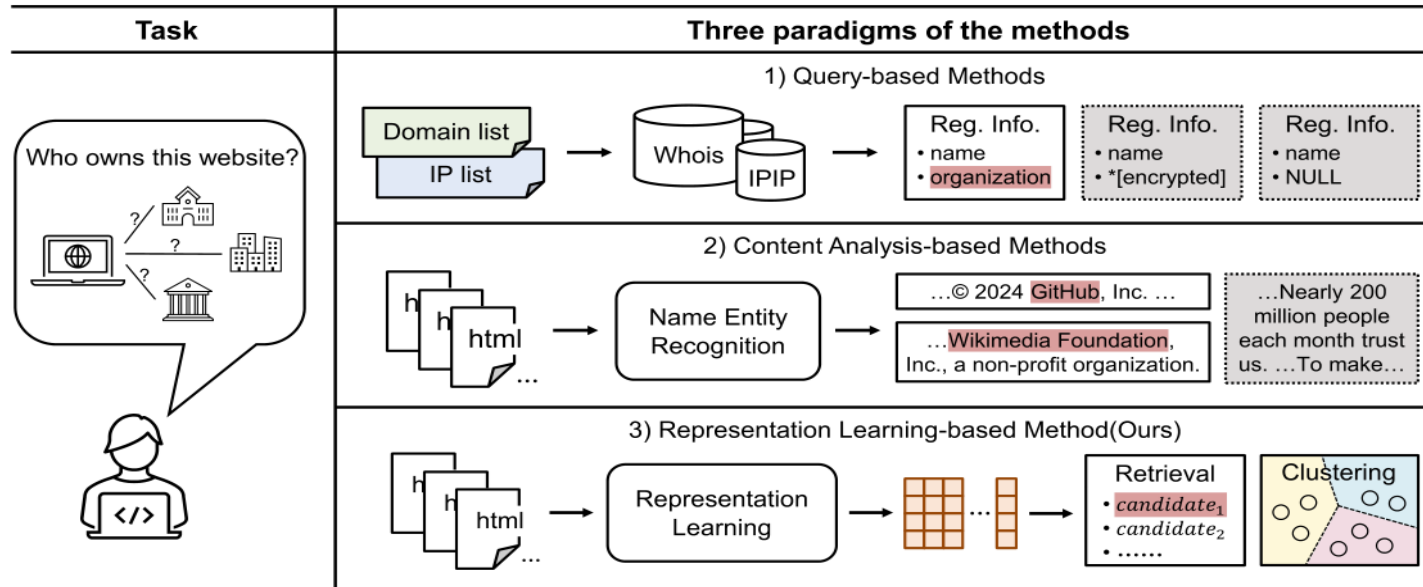
New assets and
changing configurations

Website-owner identification contributes to attribution and remediation routing.

Our contribution addresses one essential link: connecting a website to its real-world owner.

Website ownership: who is responsible?

A hosting provider may not be the website owner.



The task

Connect a website to the real-world organization or individual behind it.

The gap

A registrar may be a provider. The actual owner may not be named.

The value

Build inventory and route notifications to someone able to remediate.

Ownership requires evidence about responsibility, beyond a visible organization name.

Website ownership: sparse and ambiguous evidence

Ownership evidence can be sparse or ambiguous.

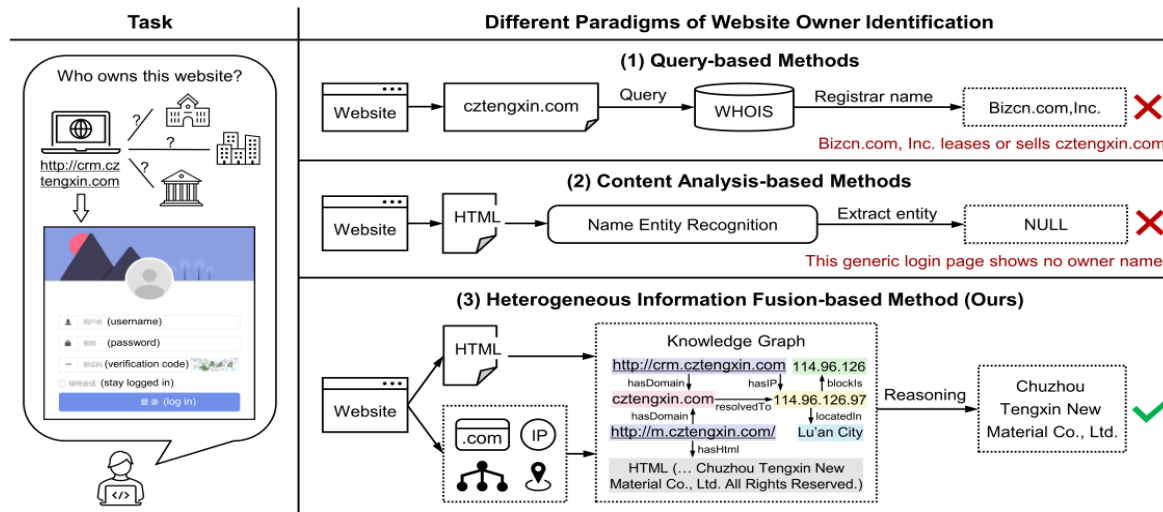


Fig. 1. Left: Task illustration of WOI, where the input is a website and the output is its owner. Right: Paradigm comparison between our proposed approach (Heterogeneous Information Fusion-based Method) and existing methods (Query-based and Content Analysis-based Methods).

A sparse login page

The target provides little direct owner information.

Evidence beyond the page

A related website can provide additional ownership clues.

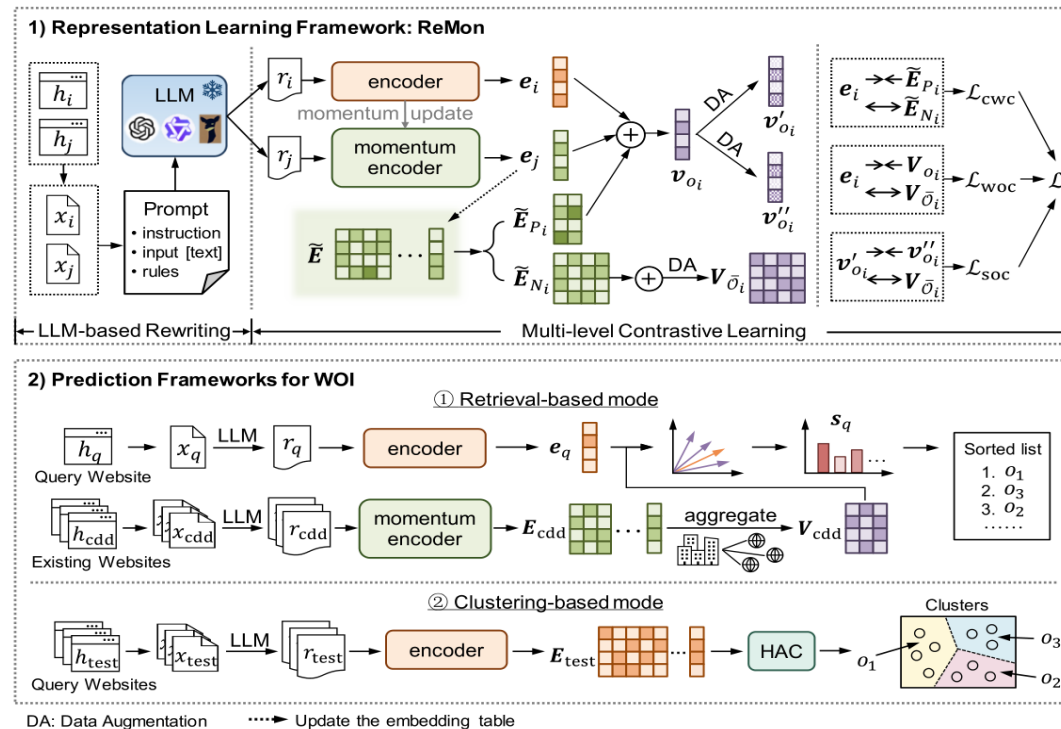
The challenge

Some graph paths are useful. Shared infrastructure can also mislead.

This motivates ownership intelligence: learn relationships, resolve multilingual names, and reason over linked evidence.

ReMon: from names to ownership representations

The owner's name may be absent from the page.

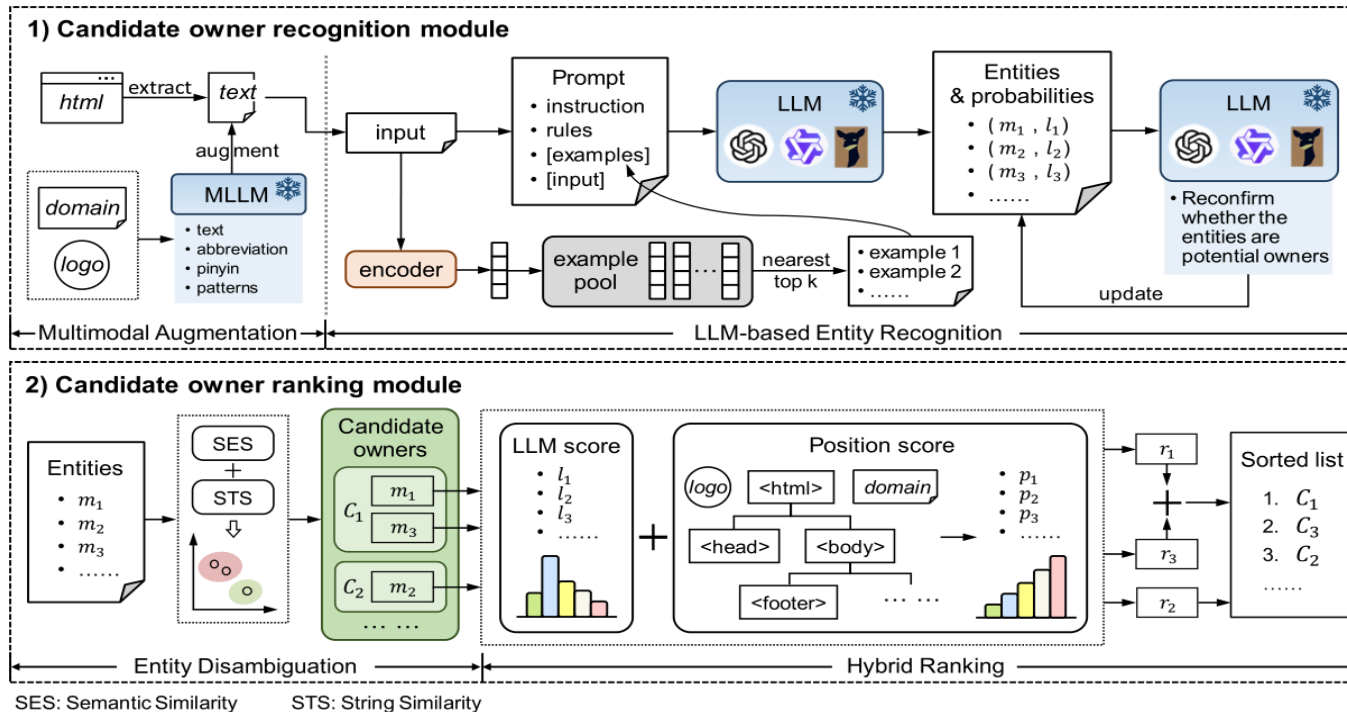


- The gap
 - WHOIS can be incomplete, and the owner name may be absent from the page.
- The response
 - Learn ownership-sensitive representations after LLM-based content rewriting.
- Why it fits
 - Website and owner relationships support retrieval or clustering, beyond explicit-name extraction.

Representations handle missing names. Explicit owner evidence still needs careful multilingual reading.

OwnerHunter: multilingual owner identification

The full page helps distinguish owners across languages.

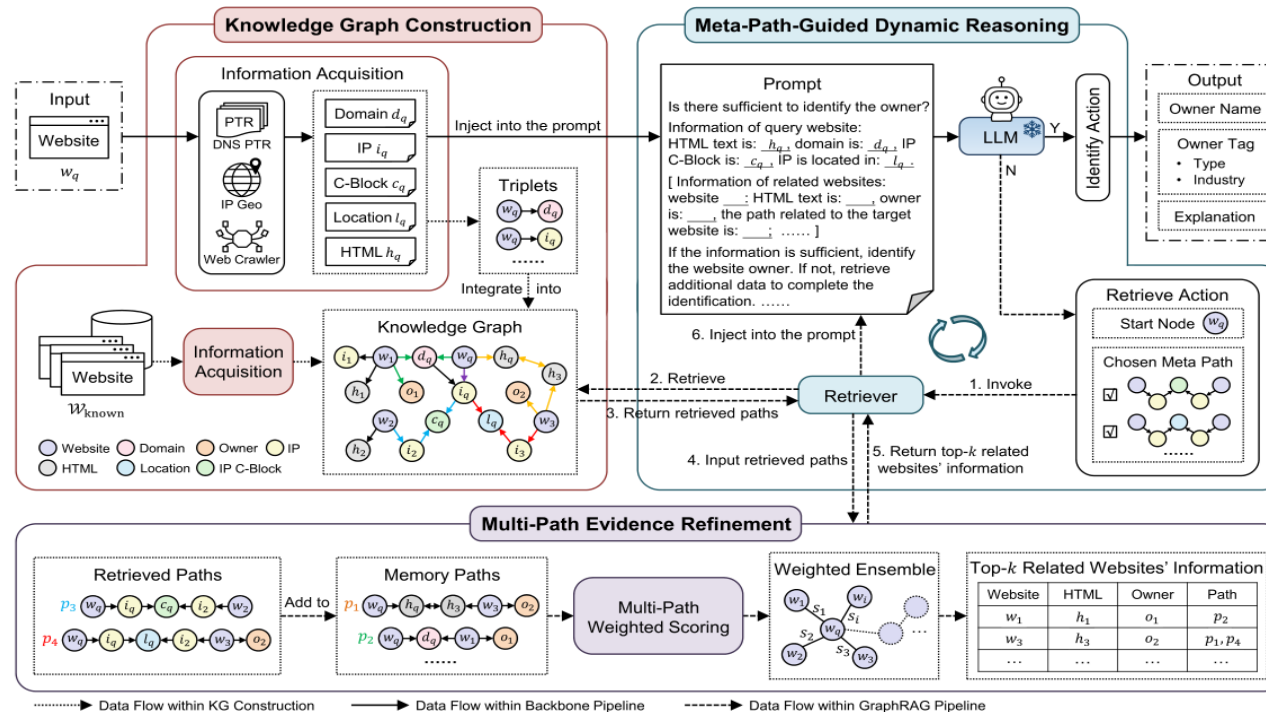


- The gap
 - Local name extraction confuses owners with partners, and struggles across languages.
- The response
 - Read the whole page with few examples, augmented cues, and candidate verification.
- Why ranking matters
 - Resolve aliases and competing names before selecting the owner.

Whole-page understanding helps. Sparse pages may still require evidence from related websites.

MetaRAG: from sparse pages to graph evidence

Relevant graph evidence can supplement a sparse page.



- The gap
 - A sparse page can provide too little evidence, while unrestricted graph retrieval adds noise.
- The response
 - Retrieve along ownership-relevant meta-paths and seek more evidence when needed.
- Why it fits
 - Refine multiple paths before inferring ownership, rather than trust shared infrastructure alone.

Ownership inference closes the second research line. We now turn to a separate testing task.

Website ownership: complementary approaches

The three methods address different evidence gaps.

Evidence gap	Approach	Useful signal
Owner not named	ReMon	Learned website and owner relationships
Many names or languages	OwnerHunter	Document-level context and disambiguation
Sparse local content	MetaRAG	Ownership-relevant paths through a graph

Representation learning

Infer ownership when the page omits an explicit owner name.

Language understanding

Distinguish owners from customers, partners, and other mentioned entities.

Graph retrieval

Combine local content with relevant paths through public metadata.

These methods map website ownership. The third research line studies Penetration Testing.

LLM for Cybersecurity: Penetration Testing

1



Threat Intelligence Analysis

Information extraction
Knowledge graph construction
Event detection and forecasting
Temporal analysis
Report generation

2



Attack Surface Management

Representation learning
Contrastive learning
Graph reasoning
Graph RAG

3



Penetration Testing

Agentic systems
Agent post training
Dynamic environment construction

What is penetration testing?

Penetration testing is an authorized simulated attack that evaluates a system's security.

Example: a new customer portal

Business question

Can one tenant access another tenant's invoices?

Agreed test environment

Website and API, synthetic records,
and explicitly authorized test accounts.

A service engagement

Scope the assessment before production release.

A verified finding

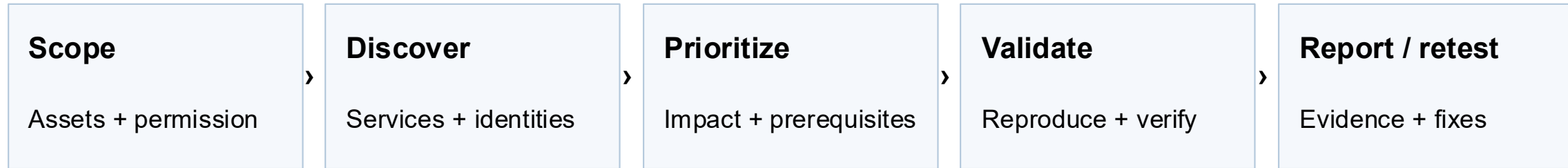
Reproduce the issue and retain target evidence.

An actionable outcome

Explain impact, recommend a fix, and retest.

Why automate penetration testing?

Expert judgment must connect many tests into one defensible assessment.



12 testing categories

OWASP spans authentication, authorization, sessions, input handling, business logic, and more.

1–3 weeks

Illustrative vendor planning range for a SaaS/web application engagement.

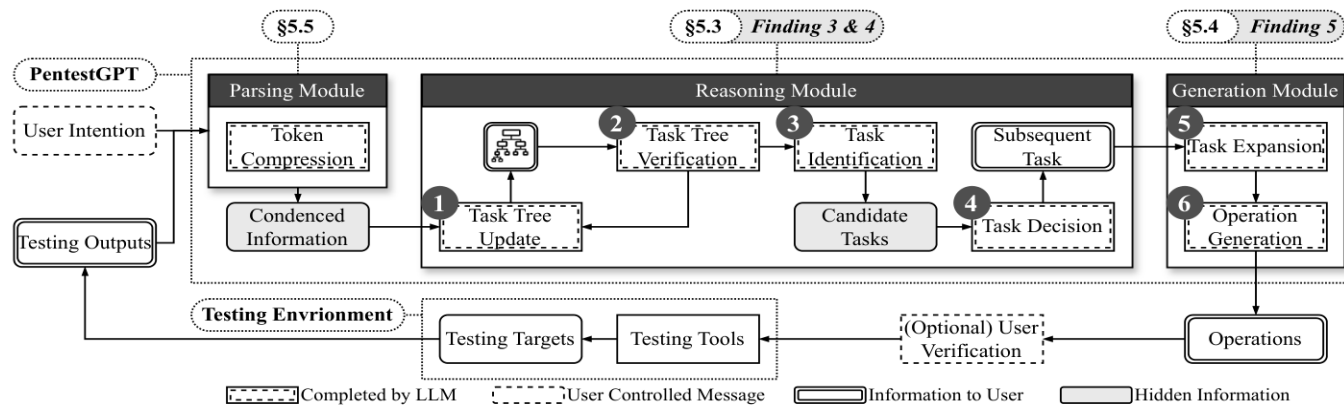
US\$5,000–30,000

A scope-dependent commercial planning range, not a universal price.

Automation can reduce repeated effort and support more frequent testing between releases.

PentestGPT: pioneering LLM-agent penetration testing

An early LLM-agent framework uses task memory and tool feedback to support automated testing.

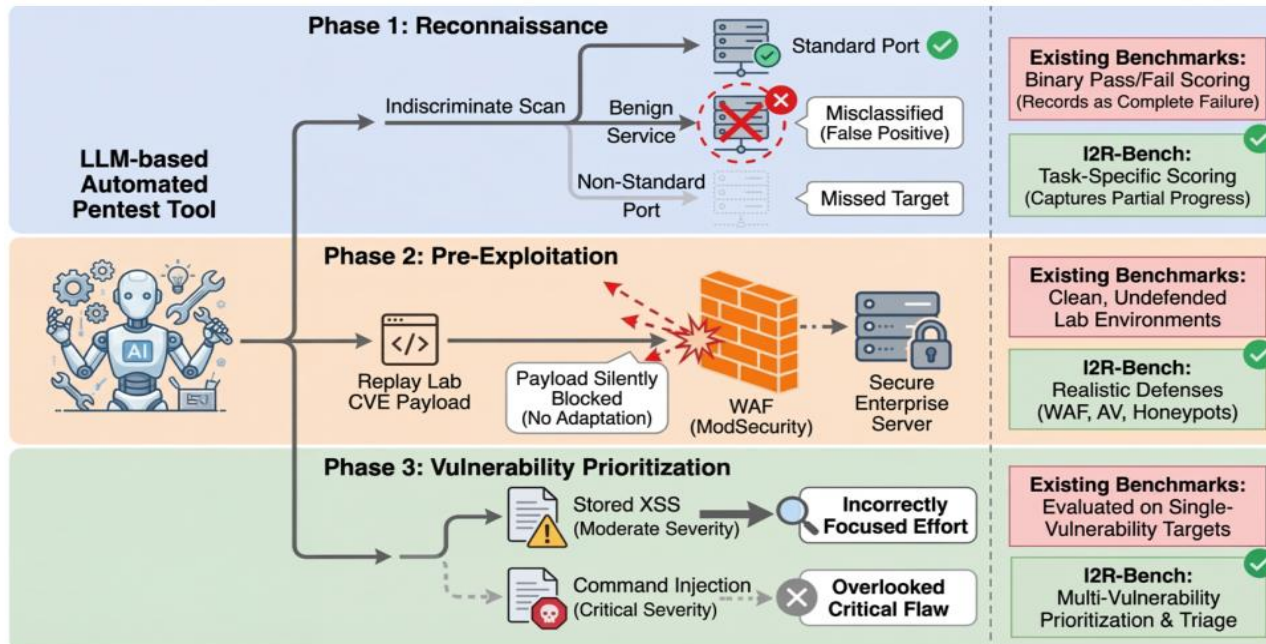


- The problem
 - Long conversations lose the overall engagement state and earlier evidence.
- The response
 - Separate the task tree, detailed actions, and interpretation of tool output.
- The remaining question
 - Does the testing protocol expose realistic dependencies and defenses?

PentestAgent and VulnBot extend this direction with agentic and collaborative workflows.

Gaps in current agentic penetration testing

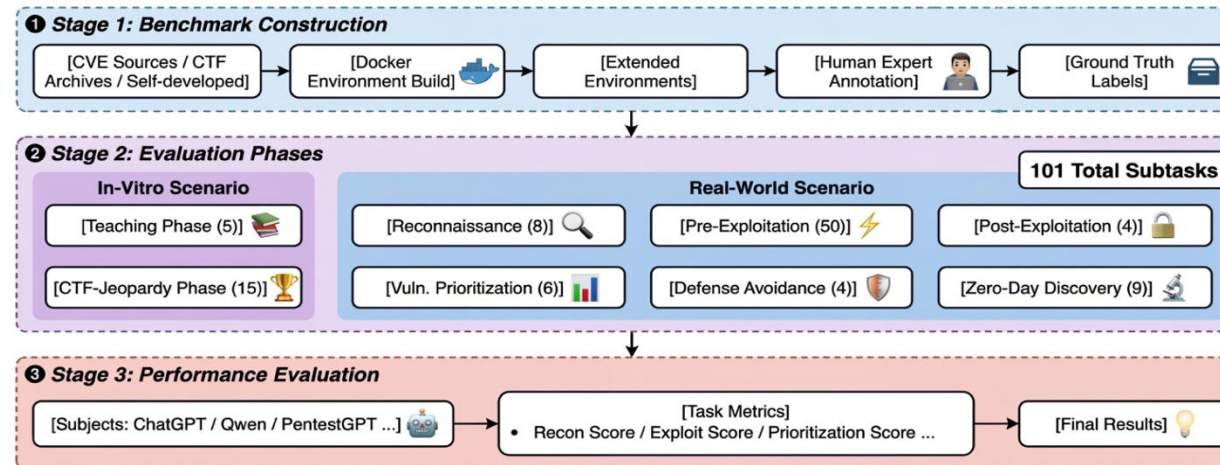
A solved isolated challenge does not establish a complete penetration test.



- Narrow workflow coverage
 - Missing dependencies, competing weaknesses, and realistic defensive controls.
- Simplified information
 - Known hints or privileged target context can remove the discovery burden.
- Overly coarse success
 - A flag or pass/fail cannot represent discovered sets, ranked priorities, and partial exploitation.

Our proposal: I2RBench - In-Vitro to Real-World

To address isolated-task coverage, I2RBench evaluates eight testing phases with task-specific scoring.



Coverage gap

Isolated tasks miss phases such as prioritization and defense avoidance.

A broader lifecycle reveals which capabilities an apparently strong agent still lacks.

Scoring gap

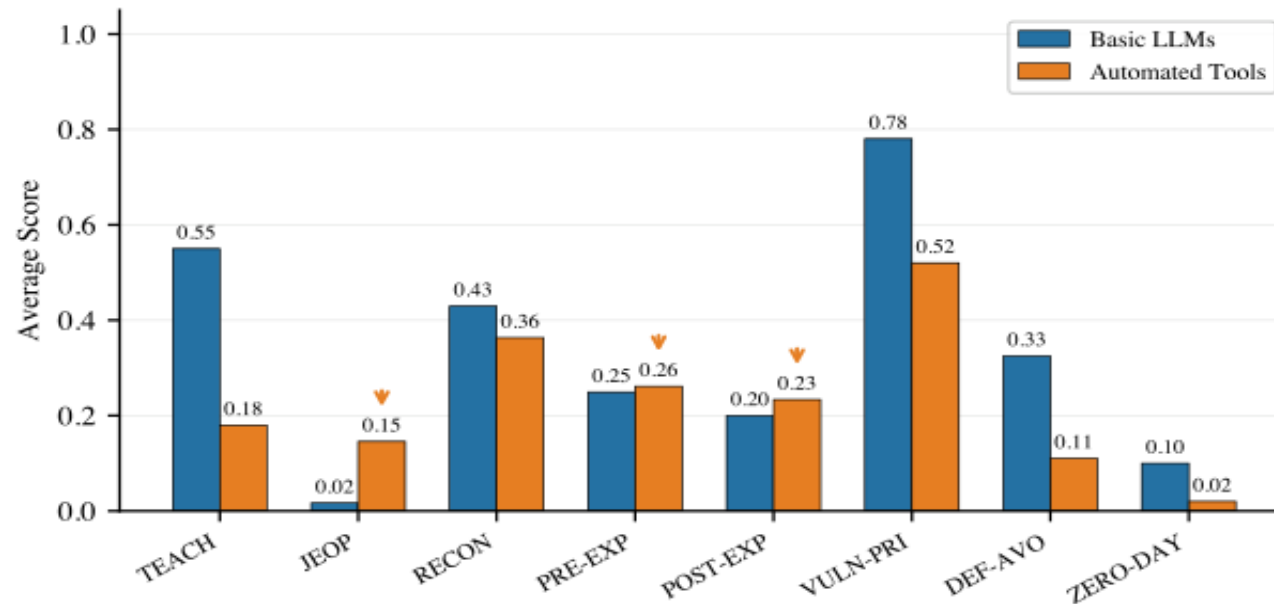
Discovered sets, ranked issues, and exploit outcomes need different metrics.

Response

101 subtasks across eight phases, with phase-specific assessment.

Key Findings: orchestration depends on the phase

Tool orchestration helps some testing phases and constrains others.

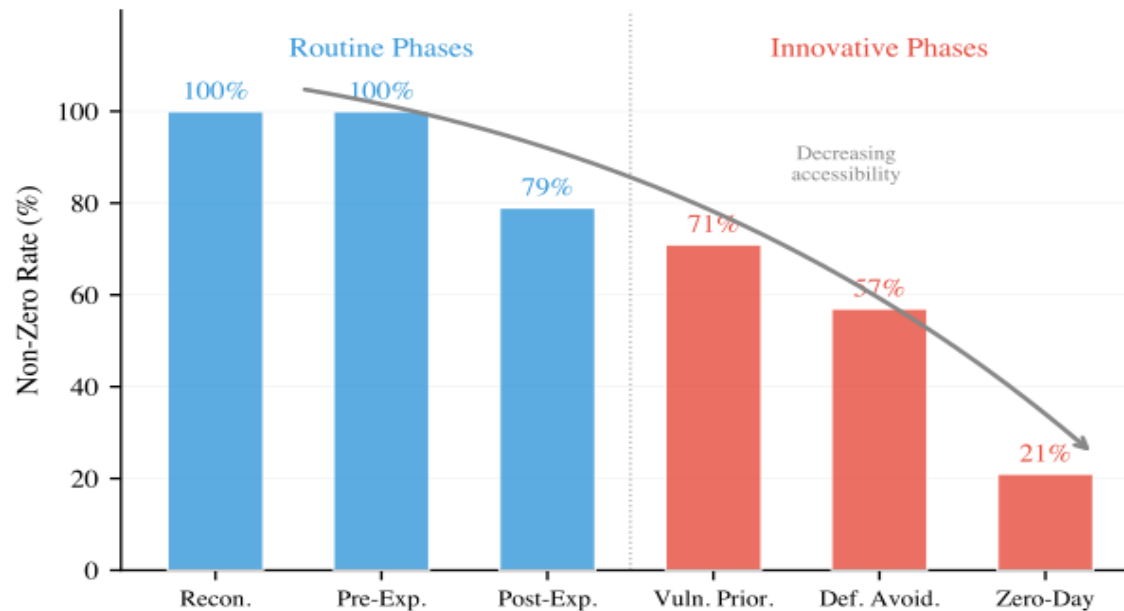


- The question
 - Does an agent framework add value across every phase?
- Draft observation
 - Orchestration helps some action-heavy phases and constrains other capabilities.
- Design implication
 - Link each component to a failure it is meant to solve.

The research problem is to add the right orchestration, rather than assume that a larger framework always helps.

Key Findings: advanced phases remain difficult

Partial progress does not establish a completed penetration test.

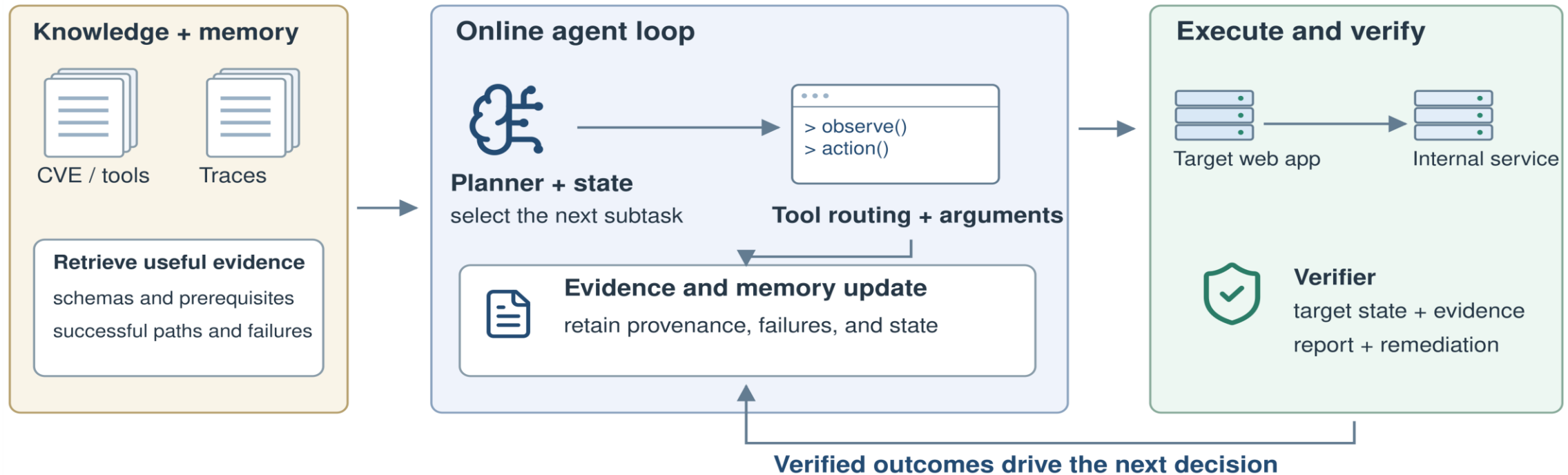


- The question
 - Where does adaptive judgment remain difficult?
- Draft observation
 - Fewer configurations gain any credit in defense avoidance and zero-day tasks.
- Interpretation
 - Nonzero credit measures some progress. It does not establish full exploitation.

These gaps motivate grounded tool arguments, reusable memory, recovery, and independent verification.

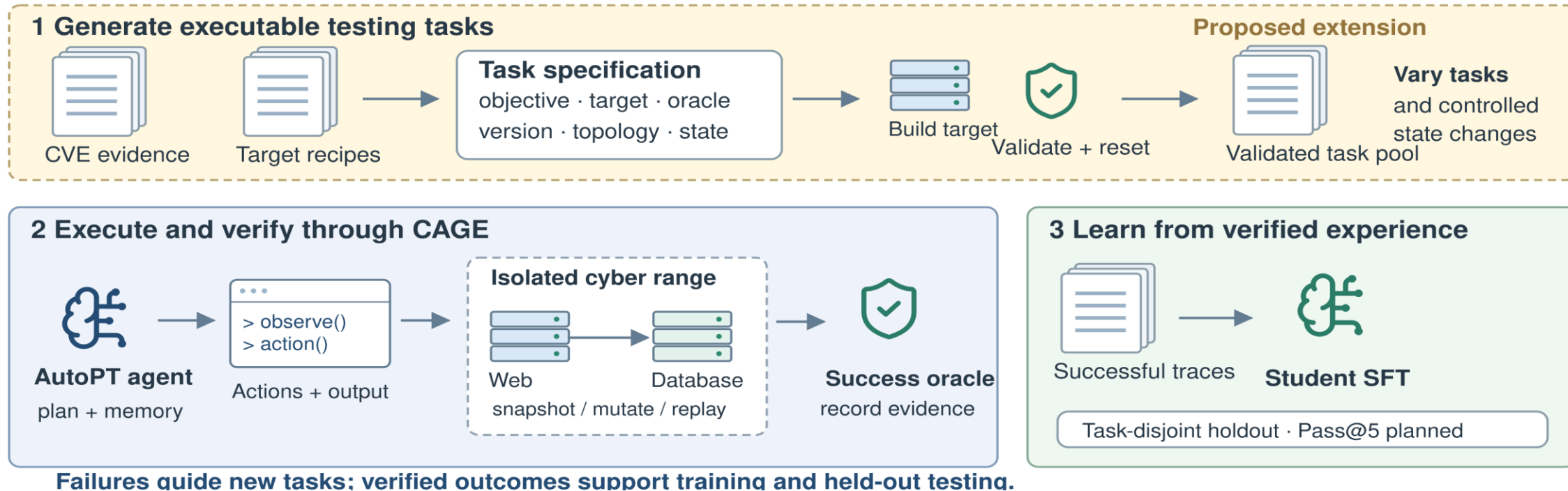
Our proposed system (under development): AutoPT

To sustain long-horizon testing, AutoPT grounds tool use in persistent state and verifies action outcomes.



Key feature: learning through interactions

Motivation: agents need varied situations and verified experience beyond fixed testing tasks.



Summary

Our work connects cybersecurity challenges with AI techniques.

Cybersecurity Challenges

Our Answer: LLM for Cybersecurity

AI Techniques

1. Threat Intelligence Analysis

AttackKG+

MM-AttackKG
AttackSeqBench

CTI-ReGen
TTPDecade

Information extraction
Knowledge graph construction
Event detection and forecasting
Temporal analysis
Report generation

2. Attack Surface Management

ReMon

OwnerHunter

MetaRAG

Representation learning
Contrastive learning
Graph reasoning
Graph RAG

3. Penetration Testing

I2R-Bench
AutoPT

AutoPT
Dynamic
environments

Agentic systems
Agent post training
Dynamic environment
construction

2023

2024

2025

2026

Next steps

Next steps

- Agentic systems improvement
- Environment construction
- Training in dynamic environments
- Reverse engineering

Thanks & Q&A