IEEE World Forum on Internet of Things

IEEE 11th World Forum on Internet of Things

JUMPSEC

University of Glasgow

School of Computing Science

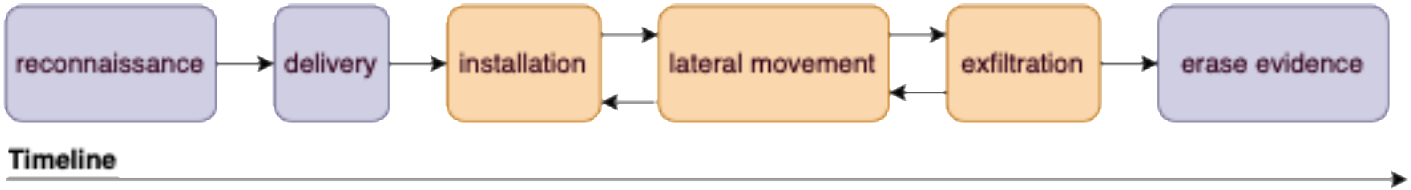
Advanced Persistent Threats based on Supply Chain Vulnerabilities: Challenges, Solutions and Future Directions

Zhuoran Tan, Shameem Puthiya Parambath,
Christos Anagnostopoulos, Jeremy Singer, and Angelos K. Marnerides

01

APT Attack Lifecycle

APT Decomposition



Reconnaissance: Information gathering, OSINT, network mapping, supplier profiling

Delivery: Payload transmission, phishing, malicious updates, software compromise

Installation: Execution, persistence, privilege escalation, backdoor deployment

Lateral Movement: Credential theft, internal propagation, network traversal, vendor trust abuse

Exfiltration: Data theft, command and control, covert transfer, supply chain channel exploitation

Erase Evidence: Log deletion, artifact removal, defense evasion, impact mitigation

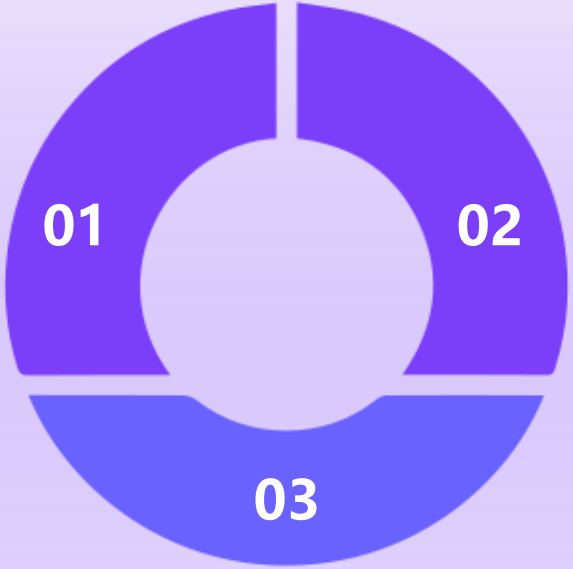
Cyber Supply Chain Compromise

02

CSC Primary Components

Software and Hardware

Covers both **software supply chains** (open- and closed-source) and **hardware components** (firmware, routers, IoT, etc.).



Manufacturing and Development Processes

Encompasses **DevSecOps** and **MLSecOps** practices integrating security into software development and AI-driven systems.

Logistics and Third-Party Ecosystems

Includes **transportation, distribution, and external service providers** (e.g., cloud, outsourced IT).

Classification of Supply Chain Compromises

01

Source Compromise.

Software
Hardware
Thrid-partiy providers
Human element

02

Build Compromise

Software
Hardware
Infrastructure

03

Transform compromise

Digital distribution
Physical distribution
Third-party providers

04

Usage compromise

Availability/integrity
Performance
Privacy
Integrated systems

AI component threats

1

Software Components

AI introduces new risks like data/model vulnerabilities and privacy issues, complicating supply chain security.

2

Hardware-Based AI

AI hardware components— processors, edge devices, embedded systems. Threats tampering, reverse engineering, and side-channel attacks

3

Detection/Prevention

Integrity check / model scan / data monitoring, but still facing challenges at system level

MITRE/SLSA frameworks

SLSA for Supply Chain Security.

SLSA enhances integrity via tamper-proof builds and provenance tracking, mitigating update poisoning risks.

02

01

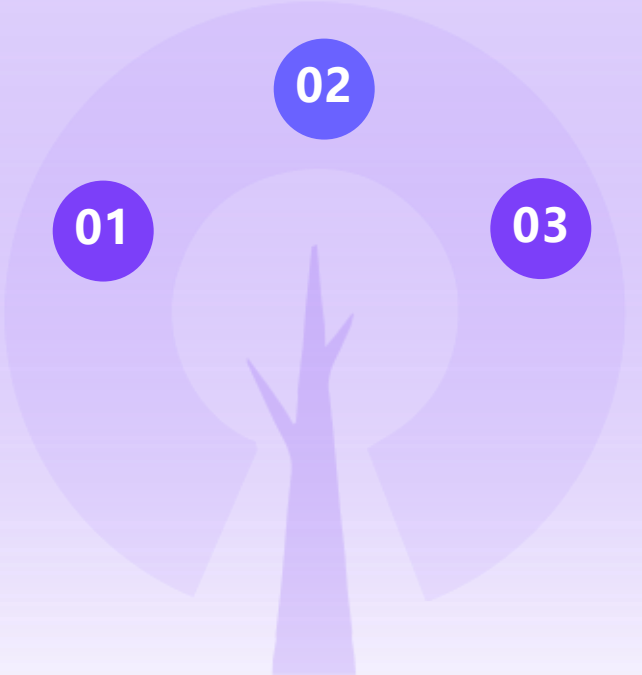
MITRE Framework for SCV-APT.

MITRE framework details tactics like reconnaissance and lateral movement in supply chain attacks, aiding threat modeling.

03

Defense Gaps in Current Frameworks.

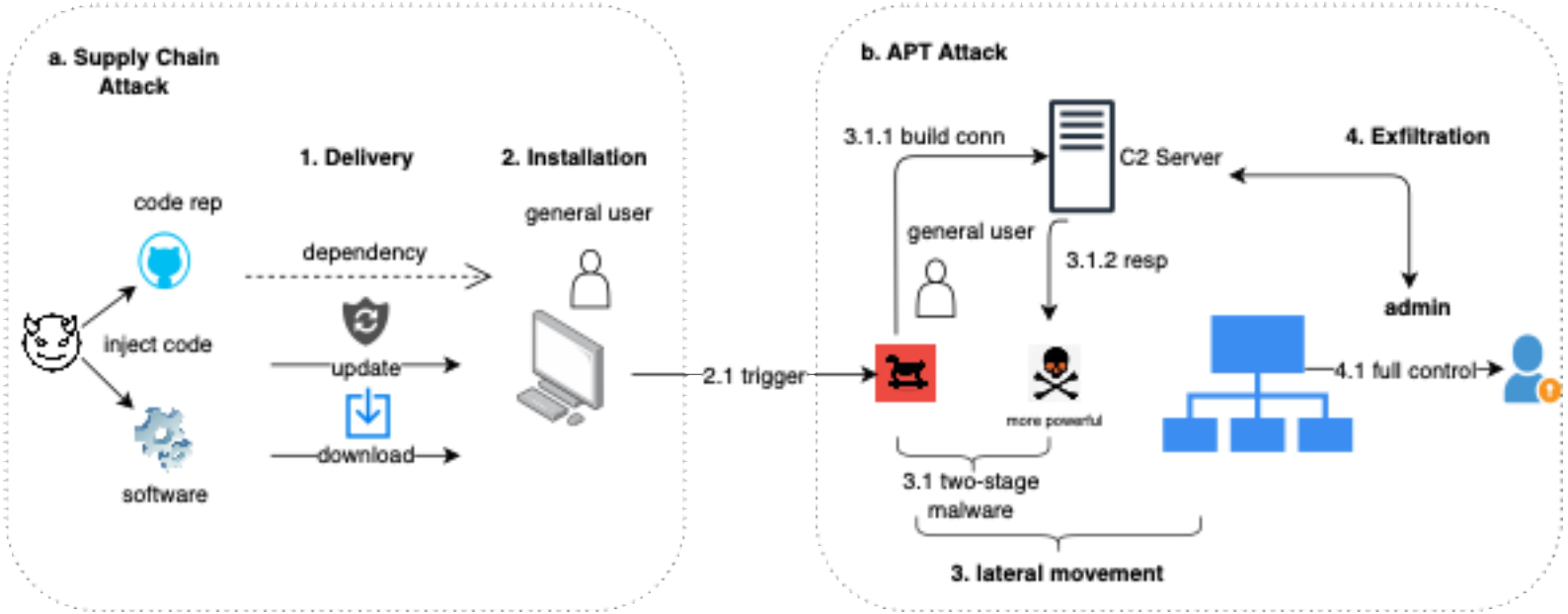
Existing frameworks lack coverage for AI/ML threats and long-term APT evasion techniques.



Supply Chain based APT Attacks

03

Threat Model

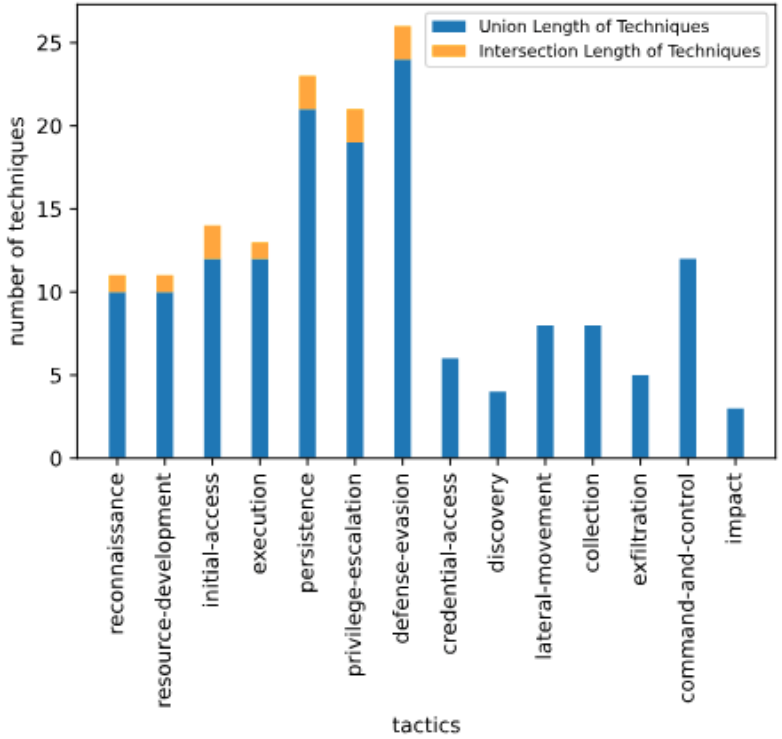


Technique Insight

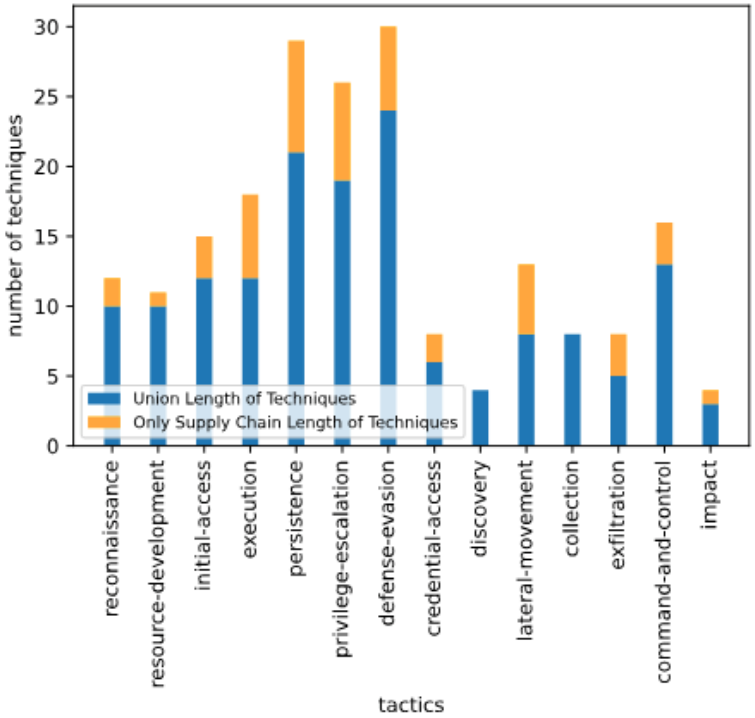
TABLE II: Supply Chain Based APT Attacks

Name	Foothold Establishment	Delivery	Installation	Lateral Movement	Exfiltration	Erase Evidence
SolarWinds [78]	Drive-by Compromise, Trusted Relationship, Malicious Code Injection(MCI), Compromise Update Mechanism (CUM), Compromise Build Environment (CBE)	Update	Remote Services, Activate Backdoor	Pass the Hash, Remote Services, Web Protocols, Valid Accounts	Exfiltration Over C2 Channel	DLS, Modify System Timestamps, Obfuscate Activities
CCleaner [79]	Trusted Relationship, CUM, UA, MCI, CBE	Update	Upon Download/Execute	N/A	Exfiltration Over C2 Channel	N/A
ASUS [80]	UA, Trusted Relationship	Software Package, Update	Initial Setup or Update	N/A	N/A	N/A
NotPetya [81]	HU, Trusted Relationship	Update	Exploitation for Privilege Escalation	Remote Services: SMB/Windows Admin Shares, Lateral Tool Transfer	Data Encrypted for Impact	Data Destruction
Kaseya VSA [82]	Trusted Relationship, ZDV, UA	Trusted Relationship	Automatic	Update	Hijack Execution Flow: DLL Side-Loading	N/A
Accellion FTA [83]	Trusted Relationship, ZDV, UA	Web Shell	Create or Modify System Process, Create Account	N/A	Exfiltration Over Alternative Protocol	N/A
Codecov [84]	Drive-by Compromise, UA	Bash Uploader	Backdoor	N/A	N/A	N/A
HAFNIUM [85]	Trusted Relationship, Exploit Public-Facing Application, ZDV, UA	Web Shell	Create or Modify System Process, Create Account, Command and Scripting Interpreter: Windows Command Shell	OS Credential Dumping: Security Account Manager, Remote Services: SMB/Windows Admin Shares	Exfiltration Over Web Service: Aspera Faspex	N/A

Technique Insight



Comparison of Union and Intersection Techniques in Tactics between all APTs



Comparison of Union Techniques in all APTs and Techniques only in APT based on Supply Chain

SCVs as primary attack vector

01 SCVs as APT primary vector.

Supply chain vulnerabilities enable stealthy, long-term APT attacks, exploiting trusted updates and dependencies for widespread infiltration.

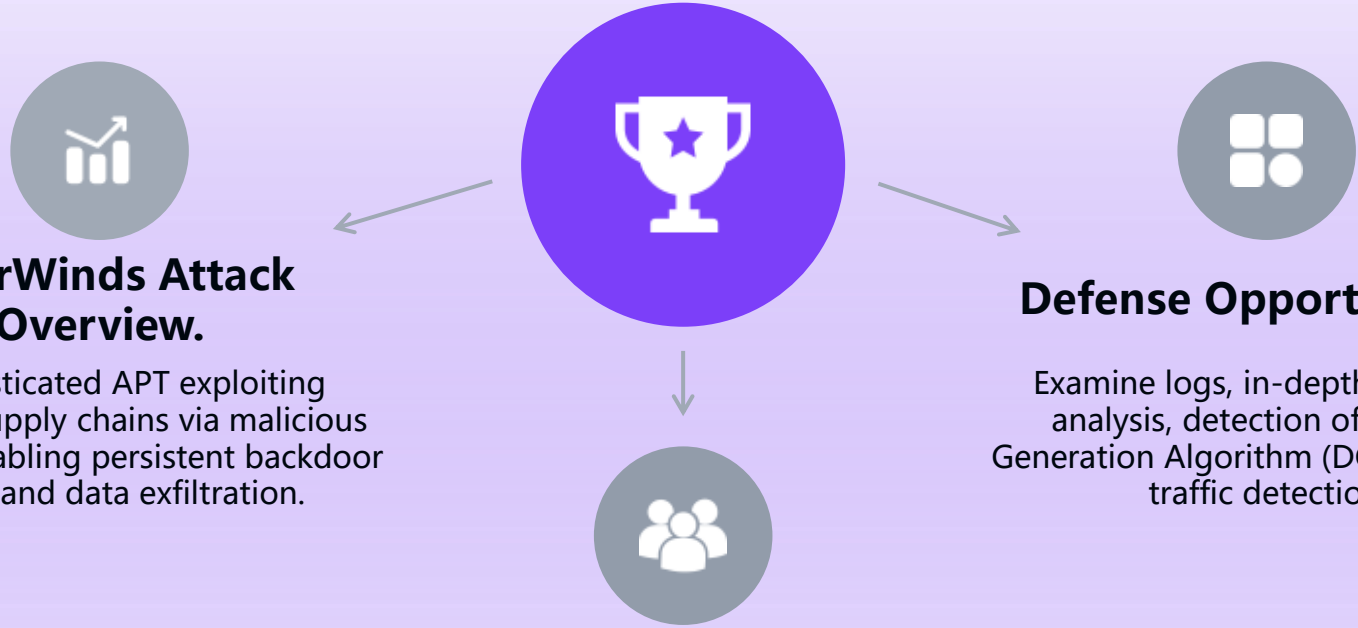
02 Unique SCV attack tactics.

Leverages trust in third-party components, update poisoning, and lateral movement via compromised dependencies, differing from traditional APTs.

03 Defense challenges and solutions.

Current research lacks suitable dataset, efficient solutions, strong attention to counter evolving SCV-APT techniques at early stages.

SolarWinds case study



SolarWinds Attack Overview.

A sophisticated APT exploiting software supply chains via malicious updates, enabling persistent backdoor access and data exfiltration.

Defense Opportunities.

Examine logs, in-depth malware analysis, detection of Domain Generation Algorithm (DGA), network traffic detection.

Key Attack Techniques.

Leveraged trusted vendor relationships, update poisoning, and lateral movement to evade detection and escalate privileges.

04

Defense Strategies

Classification of Defense Methods

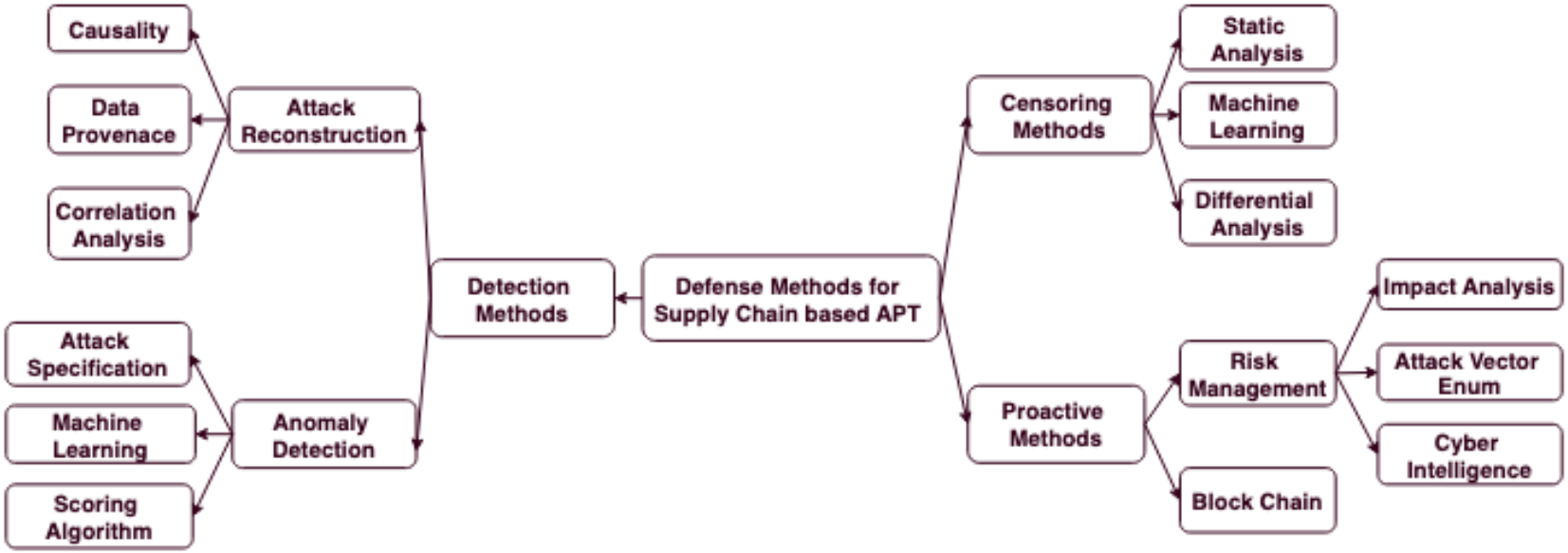


Fig. 7: Classification of Defense Methods

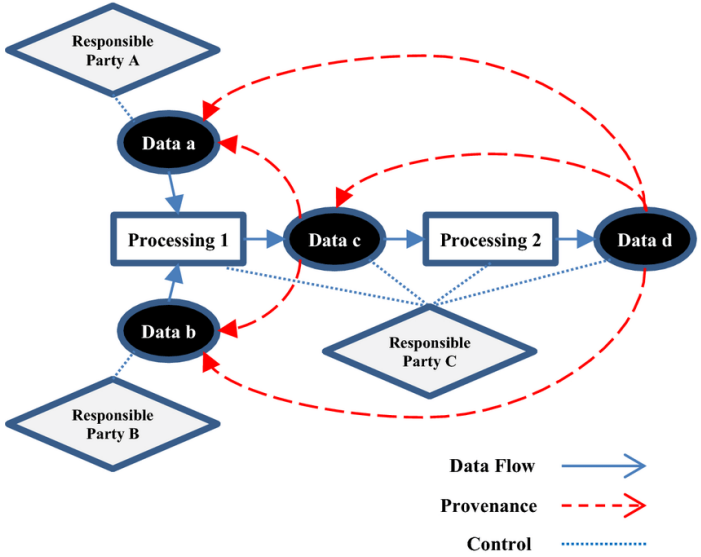
Detection-based Method – Attack Reconstruction

- **Definition**:
  - Reconstructing attack paths or graphs by analyzing raw data sources or correlating alerts from intrusion detection systems.

- **Core Technologies**:
  - Data Provenance – tracking data origin and flow
  - Causality Analysis – identifying cause-effect relationships
  - Correlation Analysis – linking related events across systems

- **Purpose**:
  - Enhances visibility and reduces **false positives** in anomaly-based APT detection

Attack Reconstruction- Data Provenance



Concept: Documents the **origin, lineage, and transformation** of data throughout its lifecycle

Use in CyberSecurity: Enables tracing of attack activities and data manipulation across distributed systems

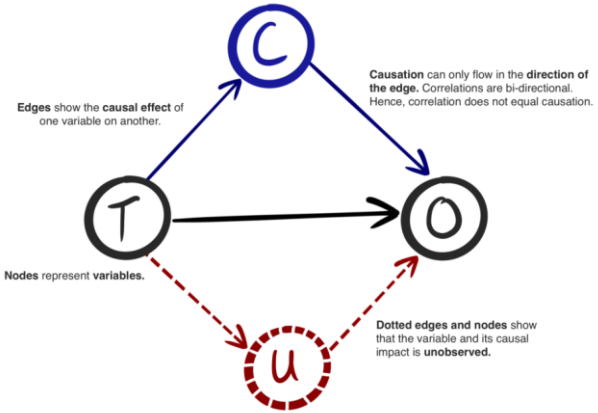
Key Systems & Approaches:

- **CONAN (Xiong et al.)** – State-based, lightweight provenance framework; retains only 0.1% of events → **low FPR** and efficient detection

- **ATLAS (Alsaheel et al.)** – Combines **NLP + ML (LSTM)** for event sequence analysis; reconstructs causal attack graphs; mitigates **alert fatigue**

- **TRACE (Irshad et al.)** – Enterprise-level provenance system; integrates static analysis and causal graphs; achieves **80% APT detection coverage**

Attack Reconstruction - Causality



Definition: Understanding **cause–effect links** between events via causal **discovery** (no prior structure) and **inference** (based on prior assumptions)

Applications in Cybersecurity: Models attack logic and progression to improve **APT detection and response**.

Representative Systems:

- **SteinerLog (Bhattarai et al.)** – Real-time APT reconstruction using **Steiner tree + hierarchical traversal**; assigns **risk scores** to anomalies

- **ATLAS (Alsaheel et al.)** – Builds causal graphs from audit logs to counter alert fatigue.

- **TRACE (Irshad et al.)** – Correlates multi-host provenance data into **enterprise-wide causal graphs**.

Attack Reconstruction – Correlation Analysis

Definition: Identifies **hidden relationships** within large datasets through **co-occurrence patterns**

Key Research & Systems:

- **Poirot (Yang et al., 2022):** Combines **causality** and **correlation** to detect long-term APTs. Uses **Latent Dirichlet Allocation (LDA)** to model semantic context and latent attack intent.
- **MNEMOSYNE (Allen et al., 2020**): **Postmortem forensic tool** for **watering hole attacks** (early APT stage). Applies **versioning** and **user-level analysis** to detect compromised sites and behavioral changes.

Detection-based Method - Anomaly Detection

- **Purpose:** Identify abnormal data instances or patterns that deviate from expected behavior.
- **Three main detection method categories:**
- **- Attack Specification**
- **- Machine Learning Applications**
- **- Scoring Algorithm**

Anomaly Detection – Attack Specification

Definition: Rule-based detection using empirical indicators (IOCs/IOAs)

Examples:

- **HAWK-EYE (Alageel et al., 2021):** Uses DNS semantic and structural features for APT C2 domain classification; complements NIDS.
- **Hopper (Grant et al., 2021):** Detects lateral movement using rule-based anomaly scoring from log data.
- **Wilkens et al. (2021):** Builds APT scenario graphs from alert sequences to reduce noise and capture network-based APT stages.

Advantages: High accuracy for known threats.

Limitations: Requires manual rule creation post-incident, Labor-intensive setup, Ineffective against unknown or novel attacks.

Anomaly Detection - Machine Learning Applications

Purpose: Use ML/DL models to learn features and detect APT anomalies across data sources

Representative Studies:

- King et al. (2023): *Euler* system—uses GNN + RNN in distributed setup for scalable, temporal lateral movement detection.

- Du et al. (2024): *Vul-RAG*—LLM-based RAG framework for code vulnerability detection; builds CVE knowledge base; limited by context length and knowledge base quality.

Challenges:

Black-box nature and low interpretability.

Data bias leading to false positives.

Detection-based Method – Scoring Algorithm

Concept: Assigns numeric scores (anomaly or similarity) to improve detection and interpretability.

Applications:

- **Similarity scores:** Compare devices or codebases (e.g., Wang et al., Yi et al.).

- **Anomaly/risk scores:** Detect abnormal activity or rank threats (e.g., Ho et al., Hassan et al., Bhattarai et al., Li et al.).

Benefits:

Reduces false positives in ML-based systems.

Enhances interpretability.

Detection-based Method – Industrial Detection Solutions

Purpose: Identify real-world APTs exploiting supply chain vulnerabilities.

Examples & Detection Focus:

Behavioral/Process-Based: SolarWinds, NotPetya, Cozy Bear, OilRig.

Hash/Artifact-Based: Kaseya VSA.

Dynamic Code Execution Analysis: Lazarus Group.

Abnormal Behavior & Traffic Analysis: CCleaner, Codecov.

Censoring Methods

- **Focus**: Defense methods targeting **supply chain attacks**, emphasizing **early detection** at the **code-level** (infection point).

- **- Classification:**
  - Differential Analysis
  - Program Analysis
  - Machine Learning

Censoring Methods – Differential Analysis

Concept: Compares benign and infected software versions to detect malicious changes.

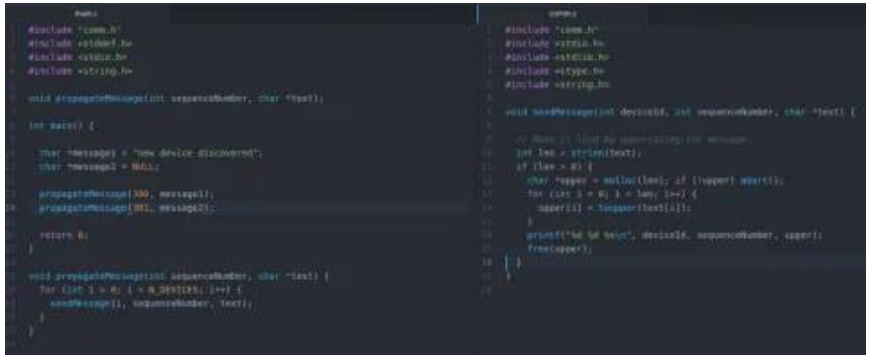
Purpose: Identify anomalies or patterns indicating malicious insertions.

Key Studies:

- **Exorcist (Barr-Smith et al., 2022):** Automated binary-level differential analysis across build versions; combines static, dynamic, and binary analysis; weights results for reliability.

- **Froh et al., 2023:** Uses **CodeQL** for differential static analysis of open-source packages; scores updates by suspicious behavior (network, process, obfuscation, etc.); depends on CodeQL's coverage and accuracy.

Advantages: Identifies abnormal code differences efficiently.



Censoring Methods – Program Analysis

- **Definition**: Examines code through **static** (without execution) and **dynamic** (runtime) analysi.

- **Key Studies:**
  - **Ladisa et al., 2022:** Static analysis for Java OSS repositories; focuses on bytecode and constant pool features; outperforms AVs but limited by undecidability and static constraints.
  - **Wang et al., 2024:** Introduced **vulnerability nets** combining data and control flow graphs; integrates auditor expertise; improves taint-style vulnerability detection but remains labor-intensive.

- **Advantage**: Comprehensive visibility into code logic and runtime behaviors.

Censoring Methods – Machine Learning

- **Concept**: Applies ML to detect malicious patterns directly from code features or learned representations.

- **Key Studies:**
  - **Vul-LMGNN (Liu et al., 2024):** Combines sequence embeddings and GNNs on code property graphs; captures local semantics and structure; limited by GNN depth and integration issues.
  - **DeepSight (Rieger et al., 2022):** FL-based model inspection using cosine distance, NEUPs, and DDifs to identify abnormal updates.

- **Advantages:** Automates large-scale pattern detection; adapts to unseen threats.

Proactive Methods

Definition:

- Defense strategies **implemented before** APT attacks occur or are detected. correlating alerts from intrusion detection systems.
- Aim to assess assets, anticipate threats, and prevent compromise.

Main Approaches:

- **Risk Management** (impact analysis, attack vector enumeration, cyber intelligence)
- **Blockchain-Based Design** (ensuring provenance and integrity)

Risk Management – Impact Analysis

Purpose:

- Evaluate potential consequences of threats on assets and operations.
- Identify exploited vulnerabilities and prioritize mitigation.

Key Studies:

Guo et al., 2023: Showed emerging ITs (AI, IoT, blockchain) indirectly strengthen supply chain resilience via internal factors (DEMATEL-ISM approach).

Jiang et al., 2022: Studied pre-trained model (PTM) supply chain risks on platforms like Hugging Face; found poor defense for PTMs and called for automated auditing tools.

Ladisa et al., 2022: Used OSS package download counts to gauge attack influence.

Advantages: Predicts potential impact and prioritizes defenses.

Risk Management - Attack Vector Enumeration

Purpose:

- Identify and classify all possible attack pathways in the supply chain.
- Critical for proactive defense planning.

Key Studies:

Ladisa et al., 2022: Developed a language-independent **attack taxonomy/tree** covering all supply chain stages; validated via expert/developer surveys.

Duan et al., 2020: Analyzed package managers (PyPI, NPM, RubyGems) for security flaws; limited by narrow scope and analysis accuracy.

Eggers et al., 2021: Created **attack surface diagrams** for nuclear-related supply chains to map vulnerabilities between process steps.

Advantages: Provides structured understanding of threat surfaces and improves preparedness.

Proactive Methods – Cyber Intelligence

Purpose:

- Collect, analyze, and interpret actionable data about cyber threats.
- Focused on **supply chain vulnerabilities (SCVs)** and APT-related risks.

Key Studies:

Yin et al., 2022: Developed **MAGCN**, a graph neural network model for link prediction in vulnerability graphs; detects co-exploitation patterns but relies on historical data.‘

Ren et al., 2023: Built a **knowledge graph-based platform** from OSCTI for better APT attribution and proactive defense; combines DL with expert knowledge.

Advantages: Enables proactive defense and contextual understanding of threats.

Proactive Methods – Block-Chain Design

Purpose: Use blockchain to **ensure provenance, authenticity, and integrity** across supply chain data and software.

Key Studies:

- **Bandara et al., 2021 (Let'sTrace):** Combines blockchain, TUF, In-Toto, and federated learning for secure update and development pipelines; prevents tampering and key compromise.
- **Lslam et al., 2022:** Integrated **PUF-enabled RFID** with blockchain for anti-counterfeiting; ensures authentication and secure transactions.

Advantages:

- Ensures immutable traceability and authenticity of assets.
- Reduces single-point failures in supply chain systems.

05

Challenges & Research Opportunities

Key Research Challenges in Supply Chain–Based APT Defense

1. Dataset Limitations

Existing APT datasets simulate **short-term attacks**, not real, **months-long campaigns** (e.g., SolarWinds).

Future Direction:

- Develop **realistic, post-SolarWinds datasets** capturing modern supply chain threat dynamics.
- Establish **public benchmarks** for APT detection research.

2. Multi-Source Fusion

Limited insight into **system-level behaviors** and **cross-domain relationships**.

Future Direction:

- Fuse multi-source data (system, network, process, application).
- Employ graph-based and temporal graph learning to model evolving system states and detect progressive anomalies.

3. Evasion & Privacy Challenges

Models remain **vulnerable to adversarial evasion**, **model theft**, and **privacy leaks**

Future Direction:

- Integrate **privacy-preserving FL** frameworks.
- Use **adversarial training** and **data augmentation** to harden models against evasion.
- Balance **robustness and efficiency** for real-world applicability.

Emerging Research Opportunities

4. Self-Evolving Detection Systems

- Current models prioritize **accuracy on static datasets**, neglecting **continuous adaptation.**

Future Direction:

- Adopt **continual learning** and **lifelong model adaptation** techniques.
- Leverage **feedback loops** to reduce bias and improve resilience over time.

5. Efficient & Explainable Detection

- **LLMs** improve explainability but are **resource-intensive** and **slow to train**.

Future Direction:

- Develop **lightweight LLMs** and **compressed GNNs** for scalable use.
- Explore **efficient retraining** and **distributed training strategies**.
- Focus on **interpretable AI** to improve analyst trust and decision support.

Take Away --- Open Opportunities

- **Data**: Real-world APT & SCV datasets for benchmarking.
- **Fusion**: Cross-domain, temporal, graph-based integration.
- **Privacy**: Resilient, federated, and adversarial-aware learning.
- **Adaptivity**: Self-evolving, feedback-driven detection.
- **Efficiency**: Lightweight, explainable, and scalable AI architectures.

Paper source link

Thanks For Listening!