International Journal of Computer Techniques – Volume 12 Issue 4, July - August - 2025

Smarter Cyber Defence: Using Hierarchical Explainable AI to Detect APT29

[1] Anugra P Jose: VTU26905 ,vtu26905@veltech.edu.in
[2] Dr.Priya .P. Sajan, Senior Project Engineer, C-DAC Thiruvananthapuram, priyasajan@cdac.in

Abstract—**This case study is on solarwinds cyber attack in order to see how supply chain attacks function and how AI can be used to detect them. APT29 also referred to as "cozybear" is a cyber-espionage group attributed to russian government. This enabled attackers to reach U.S. government agency networks and international corporations without being detected for months. The case was examined through digital forensic tools, log analysis and AI based threat detection models. The attack remained unnoticed for months, but AI assisted in determining patterns of compromise. The research targets the technical tactics employed by the attack, such as malware injection, stealth lateral movement, and data exfiltration, while highlighting the significance of Artificial Intelligence (AI) and Machine Learning (ML) in threat identification and digital forensics. AI-powered tools were central to the detection of anomalies, log correlation, and identification Indicators of Compromise (IOCs) that conventional security products missed. Using in-depth forensic analysis and behavioral pattern recognition, this report shows how next-generation cybersecurity defenses — driven by AI and supported by investigative forensics — are crucial in discovering and blocking such advanced persistent threats. A supply chain attack is a form of cyberattack in which hackers target a trusted third-party supplier or vendor to infiltrate into a bigger firm's network. This case demonstrates the essentiality of utilizing AI in cyber security in order to identify APTs. It also underscores the necessity of bolstering supply chain security and constant monitoring to avoid future similar occurrences.**

Index Terms—**Key words: APT29, cozybear , LiteAI-MD**

I. Introduction

APT29Cozy Bear, a sophisticated and stealthy cyber group suspected to be from the Russian intelligence agency, the Federal Security Service (FSB). The threat group is infamous for launching cyber espionage campaigns against high-profile targets like government offices, military units, and global organizations. What makes APT29 unique is its ability to have long-term, hidden access to its victims — a distinguishing characteristic of Advanced Persistent Threats (APTs).

One of the most high-profile APT29 incidents is the SolarWinds supply chain attack, which hit the international headlines in late 2020. SolarWinds is an American software firm that offers IT management solutions to more than 300,000 customers,incuding military organizations, and U.S. federal institutions like the Department of Homeland Security, the Treasury, and the Department of Commerce. FireEye, one of the world's top cybersecurity companies are also their clients. The attackers abused SolarWinds' privileged position by inserting malware into the firm's standard software updates — the Orion network monitoring platform. This advanced supply chain attack occurred in March 2020, but it went unnoticed for almost nine months. The malicious update, which contained the SUNBURST backdoor, was digitally signed and distributed via SolarWinds' routine update channels, enabling it to seamlessly land on customer systems without arousing suspicion.

In December 2020, FireEye noticed the breach after conducting an internal probe of suspicious activity on its network. It was found that the malware had been quietly sniffing out communications, internal reports, security tools, and sensitive business plans. The hackers acquired sweeping unauthorized access to many key networks, several of which were of U.S. national infrastructure and intelligence agencies.

The attack uncovered serious deficiencies in conventional malware discovery mechanisms. In spite of the existence of established security systems, attackers managed to bypass detection by:

Excessive use of digital signatures and reliance on software from validated vendors, thereby enabling the malicious update to circumvent verification processes.

Reactive security models, which react only after the malware has run and demonstrated destructive behavior.

Rigid or too general heuristic rules, which miss subtle or slow-moving threats such as SUNBURST.

Traditional detection methods—signature-based, heuristic-based, and sandbox-based inspection—were inadequate. Signature-based solutions were inadequate since the malware was unknown; heuristic approaches missed it because it loaded slowly; and sandbox tools didn't work since the malware did not start any malicious activities immediately.

The break came when AI and ML-based behavioral analytics tools were utilized. They could scan system logs and pick up on subtle changes in network behavior, looking for deviations from the norm. Such AI-assisted detection was instrumental in uncovering evidence of the APT's presence and its long-term activities.

This event, now known as one of the most landmark 21st-century cybersecurity breaches, has prompted revitalized emphasis on supply chain security and proactive defense practices. It speaks volumes on the need for intelligent, explainable, and proactive malware detection systems — particularly those that can analyze software prior to installation and detect threats that conventional tools may not.

II. Literature Survey

Cybersecurity defenses are seriously challenged by Advanced Persistent Threats (APTs), like APT29. By interfering with the update process of SolarWinds Orion, a reliable IT management tool, the SUNBURST backdoor—which was ascribed to APT29—entered the software supply chain. The drawbacks of using only signature-based malware detection systems were made clear by this attack. Conventional antivirus software, such as Microsoft Defender and Avast, works by identifying known patterns of malicious files or predefined signatures. They are effective at identifying known threats, but they frequently miss zero-day malware or modified versions that are made to evade detection.

modern security tools try to detect malware by watching how it behaves. Platforms like Cuckoo Sandbox run suspicious files in a safe, isolated environment to see what they do. This works well for catching active threats — but smart malware like SUNBURST knows how to hide. It stayed quiet for 14 days after installation, long enough to avoid detection by sandbox tools, which usually only observe for a few minutes. Other tools, like VirusTotal or MITRE ATTCK, help by matching files against known malware patterns (called Indicators of Compromise or IoCs). These are helpful for identifying threats that have already been discovered. But they're reactive, meaning they only work after the malware is known and reported. Plus, using them often means uploading files to third-party servers, which can raise privacy concerns, especially when the files are internal, confidential, or proprietary.

Platforms like Cuckoo Sandbox use behavior-based detection techniques like sandboxing, which provide dynamic analysis of file behavior in isolated environments. Organizations can correlate malware with known Indicators of Compromise (IoCs) with the help of threat intelligence platforms like Virus Total, MITRE ATTCK, and AlienVault OTX. Nevertheless, they rely on global threat databases and are reactive in nature. Because they frequently call for uploading private files to servers run by third parties, they also give rise to privacy concerns .

cybersecurity vendors have incorporated machine learning (ML) and artificial intelligence (AI) into their detection engines to overcome these obstacles. Although these tools are available, there isn't a lightweight, AI-integrated platform designed especially for software update pre-installation malware detection in the current systems. This makes room for innovative solutions like the suggested system (Lite AI-MD), a low-cost, AI-based malware scanner that can detect threats before they are installed, especially insettings with limited resources.

III. EXISTING SYSTEM

Current Malware Detection Mechanisms and Their Weaknesses The malware detection environment as it exists today utilizes a mix of methods like signature-based, machine learning-based behavioral analysis, and sandboxing techniques. These are utilized by mainstream antivirus programs like Microsoft Defender, Norton, McAfee, Kaspersky, Avast, and Bitdefender, and have, in some measure, been successful in detecting well-known and somewhat changed malware threats. With the advent of Advanced Persistent Threats (APTs) and AI-created malware, though, these are increasingly inadequate.

Current Methods in Application: Signature-Based Detection: The most popular technique employed in conventional antivirus programs banks on a database of established malware signatures (digital patterns specific to every malware). If the file is associated with a known signature, it is identified as malicious. Products such as Microsoft Defender Antivirus and Norton significantly depend on this method. This method, although being fast and light, cannot identify new or unidentified malware (so-called zero-day attacks) since there is no signature for them yet.

Heuristic and Behavioral Analysis Using AI/ML: Certain contemporary systems utilize machine learning models to scan the system's activity logs, identify suspicious sign-ins, unusual system activities, and other anomalies. These models are trained on extensive datasets of known malware. Yet, if malware is crafted to simulate legitimate actions or remain dormant for extended durations, as in the case of the SUNBURST attack, it may go undetected.

Sandbox-Based Detection: In this method, a suspicious file is run in a simulated environment (sandbox) to monitor its activity without endangering the host system. Although valuable in runtime anomaly detection, this technique has major limitations:

Time-based evasion: Advanced malware such as SUNBURST is dormant for a few days after installation, simply evading brief sandbox monitoring times.

High resource utilization: Executing each update or application in a sandbox utilizes processing resources and memory, making it infeasible for schools, SMEs, or low-resource settings.

Key Limitations: Too much dependence on Signature Databases: Malware that is new, polymorphic (self-modifying), or AI-created won't have any known signature match, rendering signature-based systems blind to them.

AI Model False Positives/Negatives: Smarter AI-powered systems are not infallible. They may flag clean files as threats (false positives) or miss malicious ones (false negatives), causing either unnecessary disruption or missed breaches.

AI-Generated Malware: Attackers are beginning to employ AI to train malware to evade AI detectors — a sinister development. These types of malware may mimic normal behavior, delay execution, or encrypt code so that detection is all but impossible.

Delayed Activation: Malware that remains dormant for many days is especially insidious. SUNBURST, for instance, executed 14 days after installation, long after most detection mechanisms would have considered it clean.

Resource Inefficiency: Sandbox-like systems which would be able to identify sophisticated behavior are too resource-intensive to operate on each file or update — particularly in schools, small businesses, or regions with restricted technical infrastructure.

Example – Microsoft Defender Antivirus: Microsoft Defender is a signature-based, mostly used protection mechanism. Though it successfully prevents known threats due to its extensive malware database, it has trouble dealing with zero-day malware, artificially generated attacks, or compromised malware that slightly varies from known patterns. This leaves it exposed to advanced methods employed by actors such as APT29.

IV. PROPOSED SYSTEM

Legacy malware detection tools are largely reactive, i.e., they scan files upon installation or at runtime. This leaves a vulnerable time window in which malware goes unnoticed and can execute, exfiltrate data, or gain persistence. Particularly for instances like the SUNBURST backdoor from APT29, in which the malicious code slept for two weeks after installation, reactive detection mechanisms fall short. Additionally, it is prone to missing new, obfuscated, or AI-avoiding threats, depending on known signatures or simple heuristics. To address such limitations, we introduce a hybrid, lightweight, AI-based malware detection system, titled LiteAI-MD, for deployment within SMEs (Small and Medium Enterprises) and educational organizations, which usually lack enterprise-scale cybersecurity infrastructure.

Proposed System Architecture and Workflow LiteAI-MD integrates three essential elements — static AI analysis, cloud-based dynamic sandboxing, and threat intelligence comparison — into one lightweight, scalable framework.

1. Pre-Installation AI-Based Static Analysis The process starts by scanning software update files before they are installed on the local machine.

It employs pretrained machine learning models, trained on real-world malware datasets as well as synthetically generated (AI-generated) threats, to estimate if a file has suspicious properties.

The model examines code patterns, file organization, metadata, and feature vectors like entropy, section mismatches, and embedded URLs.

This aids in the detection of threats independent of conventional signature matching.

2. Cloud-Based Sandboxing (Dynamic Analysis) The suspected update file is submitted to a secure cloud-based sandbox environment.

In the sandbox, the update is run to observe real-time activity.

If the file tries to execute with a delay, communicates with suspicious IP addresses, injects into system processes, or displays concealed network activity, it is detected.

Launching the sandbox within the cloud diminishes local storage and processing loads, making it deployable in environments with limited resources.

3. Threat Intelligence Comparison The system utilizes APIs such as VirusTotal, ThreatConnect, and MITRE ATTACK to compare hashes and behavioral patterns of the file against a global threat database.

Apart from public IoC feeds, it also employs a hierarchical behavioral scoring system to determine the risk level on the basis of:

·Existenceofknownmalwarecodefragments
·Suspiciouscontactof IPordomain
·Encryptedpayloadswithinexecutables
·Unusualdelaysinexecution
·Unusualfilesize

Each file is given a numeric risk score (e.g., 0–100), indicating its threat level.

4. Explainable Output and Web Interface LiteAI-MD features an easy-to-use web-based dashboard where software updates are uploaded and detailed scan reports are displayed.

The system gives reasons why each risk score using explainable AI methods such as feature importance analysis so that IT admins or researchers can believe and understand the system's decision

V. Conclusion

The emergence of advanced cyberattacks, including Advanced Persistent Threats (APTs) such as APT29 and the SUNBURST backdoor, has revealed inherent weaknesses in the traditional malware detection approach. The currently existing system depends on signature based analysis and log analysis after installation, which can be bypassed a stealthy malware easily. Unlike the existing system proposed system analyses the behavior of malware before installation which is more advanced.

LiteAI-MD is programmed to scan software update files prior to their execution, allowing for early threat detection and interception of malicious code from entering the system in the first place. Utilizing static analysis, AI-powered classification, and threat intelligence queries, it detects known and unknown threats in a timely and cost-effective manner. The system includes hierarchical explainability, enabling users to know why a file was detected as malicious — which enhances transparency and trust in the decision-making process.

Through filling the key gaps in available tools and providing proactive defense, LiteAI-MD adds a pragmatic and wise solution for protecting software supply chains from dynamic APT methods.

REFERENCES

[1] Wikipedia, "Cozy Bear," Wikipedia, 2024.[Online].Available: https://en.wikipedia.org/wiki/Cozy*Bear*

[2] M. Cobb, "SolarWinds hack explained: Everything you need to know," TechTarget, 2023.[Online].Available: https://www.techtarget.com/whatis/feature/SolarWinds-hack-explained-Everything-you-need-to-know

[3] S. Rashid, "Limitations of Signature-BasedMalwareDetection,"IEEETransactions on Cybersecurity, vol. 8, no. 3, pp. 25–30, 2019.

[4] H. Singh, R. Rawat, and A. K. Pandey, “Evasion Techniques Used by Modern Malware: A Sandbox Bypass Case Study,” International Journal of Network Security, vol. 23, no. 4, pp. 650–657, 2021

[5] ReversingLabs, “Software Supply Chain Security Solutions,” 2023. [Online]. Available: https://www.reversinglabs.com

[6] T. Balarabe, “The SolarWinds Hack: Implications, Lessons, and Future Cybersecurity Strategy,” Medium, Jan. 2024. [Online]. Available: https://medium.com/@tahirbalarabe2/the-solarwinds-hack-implications-lessons-and-future-cybersecurity-strategy-08a0afa48637

[7] Snyk, “Software Composition Analysis Tools,” 2022. [Online]. Available: https://snyk.io

[8] JFrog, “Xray: Security Scanning for Software Artifacts,”2023.[Online].Available: https://jfrog.com/xray/

[9] T. Ahmad, ”Artificial Intelligence for Cybersecurity: A Review,” Journal of Information Security, vol. 12, no. 2, pp. 109–129, 2021. doi: 10.4236/jis.2021.122007

[10] R. Khandelwal and M. Gupta, ”AI-based Malware Detection Techniques: Challenges and Future Directions,” IEEE Access, vol. 10, pp. 78431–78445, 2022. doi: 10.1109/ACCESS.2022.3191762