

Attack Path Analysis and Threat Propagation Modelling for Comprehensive Cloud Security Assessment

Sachin Yadav¹, Mukhtar Ali²

¹M.Tech Scholar, Department of Computer Science & Engineering

²Head of Department, Department of Computer Science & Engineering
Vishveshwarya Group of Institutions, Gautam Buddh Nagar

Abstract— One revolutionary paradigm that has recently arisen is the cloud, which allows for the pooling of resources among geographically dispersed users with little to no oversight from administrators. The absence of comprehensive threat evaluation models that cover the tangible, virtual, and software layers all at once is a major cause for worry when it comes to security, even if usage is at an all-time high. Instead of capturing the dynamic interaction of threats throughout the whole Cloud operating stack, current Threat Analysis (TA) approaches focus on isolated possessions, such as technology, devices, or single machine virtualization. Using a technologies-agnostic flow of data model, this study introduces a Trusted Gaming Technology (TCT) approach to Protective Clouds. The model represents Cloud function via dependent state changes. In order to track real attack channels and assume potential spreading routes, the model incorporates threats retrieved from the National Vulnerability Database (NVD). In addition to include metrics for penetration (IoC) matching and user security condition compliance verification, the model offers a variety of analysis across functional tiers (physical, hypervisor, VM, application, and API). Using five real-world CVEs as an example, TCT outperforms competing methods in terms of threat rate of detection (92%), attack route accuracy (89%), multi-level protection (91%), and false positive rate (8%). These findings demonstrate the efficacy of flow-based security evaluation in enhancing reliable Cloud monitoring.

Keywords— Secure Technology, Signals of Compatibility, Attack Path, National Exposure DB, Trusted Technologies, Knowledge Flow Framework, and the Cloud.

I. INTRODUCTION

The advent of the cloud has drastically altered the framework for contemporary IT. The Cloud allows businesses to scale efficiently, quickly, and with little outlay of capital by providing immediate access to storage, computing power, and other services spread over several geographic locations [1]. Politics, healthcare, and academic fields throughout the globe have sped up adoption of Cloud computing due to its rapid distribution and elastic scalability of virtual resources, which can be controlled with little to no human participation [2].

Security concerns are still one of the most common reasons people say they won't use the cloud, even though there are many benefits. The most common responses to surveys given by CSA have been about data breaches, unsecured APIs, account theft, and malevolent insider threats [3]. Thorough security evaluation is both complex and critically important since the attack surface is automatically expanded by the diverse, multi-tenant structure of Cloud systems.

The original intent of STRIDE, DREAD, and Attack Trees, among other traditional TA methods, was to assess dangers to discrete, static parts of a system. Because Cloud assets are inherently dynamic, such as VMs, or virtual machines, that initialize, move between hardware hosts, and shutdown at a runtime, these models are not well suited to Cloud contexts [4].

In addition, most current Cloud-specific TA methods don't predict the spread of threats across levels, instead concentrating on individual ones, such as physical structures or virtual computer layers. For instance, if a security hole is found at a hypervisor layer, it might potentially harm physical devices or even hosted virtual machines (VMs) and software running on them. Security evaluations aren't complete unless there's a single model that accounts for these dissemination patterns.

With the presentation of a Trusted Technology Technology (TCT) architecture for Protective Cloud Computing, this study fills this need. Cloud operational capability is represented as a sequence of dependent state changes in the tech-agnostic flow of data model, which is the primary contributions. Integrating threat data from the National Vulnerability Database (NVD) into this structure allows TCT to do comprehensive complex threat analysis, trace attack paths, and verify fulfillment of security requirements.

Here is how the rest of the paper is structured: There is a brief overview of relevant literature in Section II, followed by an architectural description of the TCT foundation in Section III, a flow of data model in Section IV, experiential assessment in Section V, a discussion of the findings in Section VI, and finally, a conclusion in Section VII.

II. RELATED WORK

Security in the cloud has been the subject of a great deal of study throughout the last decade. Traditional approaches to Cloud Threat Analysis, security frameworks tailored to the Cloud, and the principles of trustworthy compute are all covered in this component.

A. Conventional Threat Analysis in Cloud Environments

Microsoft's STRIDE framework classifies security risks as follows: tampering interference, defiance, data leakage, denial of service, and escalation of privilege [5]. In scattered Cloud infrastructures, STRIDE does not account for cross-layer transmission, but it does provide a systematic listing of threat types. Similarly, DREAD does not have dispersion linguistics, but it does expand threat score by evaluating Damage, Replication, Extraction, Affected Users, and Creativeness [6].

Schneier [7] created Attack Trees, which provide formal analysis for attack architecture by breaking down attack objectives into tiered sub-goals. Design mutation caused by features like virtual machine live mobility, auto-scaling, and orchestrated containers render static attack models useless, rendering their tree structure useless in representing the ever-changing topology that is fundamental to cloud activities.

B. Cloud-Specific Security Models

The CSA Top risks report [3] provides an inventory of Cloud risks and classifies them into eleven main types, but it does not provide an algorithm for how these threats spread. One tool that helps with this is CloudThreat-SA [8], which introduces a partially threat analysis program that looks at specific virtual machine assets. The problem is that it can only look at assets and not the whole stack. Despite recommending a reliable third-party paradigm for verification in the cloud, Zissis and Lekkas [9] fail to mention the transmission of threats over many layers.

A virtual machine disguised channels [10] and dual-tenancy isolation testing [11] are two examples of how flow of data analysis has been used to improve cloud security. The suggested TCT architecture relies on these efforts to verify flow-based demonstrating, however they don't include NVD threat data integration or user-imposed testing.

C. Trusted Computing Foundations

One way that the Trusted Computer Group (TCG) describes TC is via Trusted Platform Module (TPM), which allows for hardware-dependent verification [12]. While remote testing allows for accurate platform module integrity assessment, techniques focused on trusted platform modules (TPMs) overlook the transmission of programmable threats across operative levels in the cloud. Reactive flow of data architecture and hardware-based assurance are both enhanced by the suggested TCT model, which applies trusted computing concepts to the program and service stack.

III. TCT FRAMEWORK ARCHITECTURE

Each of the four components that make up the Trusted Computation Technologies (TCT) architecture—the Cloud Communication Flow Model (CIFM), Attack Path Analyzer (APA), the Threat Intelligence Engine (TIE), and the Security Regulation Compatibility Checker (SRCC)—works in tandem with the others. Figure 1 shows how the NVD corpus, which was utilized to customize the model, was distributed among different types of Cloud threats.

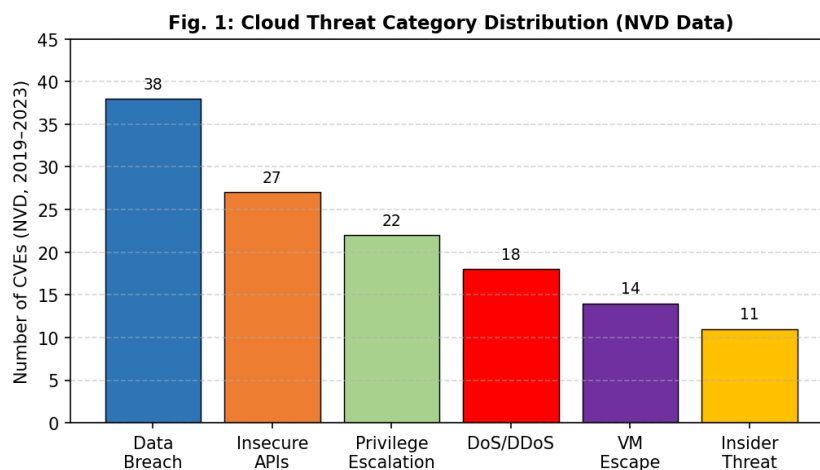


Fig. 1. Distribution of Cloud Threat Categories Extracted from NVD (2019–2023). Data breach and insecure API vulnerabilities constitute the majority of catalogued threats.

A. Cloud Information Flow Model (CIFM)

The Cloud is depicted by the CIFM as a marked state change system $\xi = (S, E, T, L)$, where S is the collection of functional states (such as running, Moving, and Service_Degraded), E is the collection of events (such as API_Call, Resource Planning, and VM_Migrate), T is the change relation, and L is a coding algorithm that maps states to met security attributes P [13]. This approach does not assume any particular computers, administration systems, or connectivity stacks; instead, it represents practical meaning.

To represent the dependent upon context, dynamic nature of a Cloud activities, conditioned movements are used. These are transition that are protected by constraints over system observables. If the condition $\text{resource_pressure} > \text{threshold} \wedge \text{migration_policy} = \text{enabled}$ is true, then a VM relocation event will change the framework state from VM_Active to Vms_Migrating.

B. Threat Injection Engine (TIE)

Threat clusters from the NVD are added to the CIFM by TIE. By comparing the afflicted product, CWE type, and CVSS vector string of each CVE with the relevant Cloud operation item and event type, the model is able to map each CVE entry to a variety of transition. The result is an enhanced model $\phi_T = (S \cup S_T, E \cup E_T, T \cup T_T, L_T)$, where S_T , E_T , and T_T symbolically represent states, events, and transitions caused by threats, respectively.

The structure is able to classify threats with comparable behavioral fingerprints by attaching Indicators of Betrayal (IoCs) as actual predicates related to threat movements. By putting similar vulnerabilities together, analysts may examine families of threats with similar attack behaviors all at once, which helps with portfolio-level risk evaluation.

C. Attack Path Analyzer (APA)

By applying forward accessible analysis to a starting state s_0 with a target compromised state s_c , APA determines all possible attack routes $\Phi(s_0, s_c)$ in γ_T . The report includes both the verified attack pathways that were found using NVD abuse data and the possible alternative channels that were found through model research. Figure 2 shows how the five operations of the Cloud that CIFM models are affected by threats.

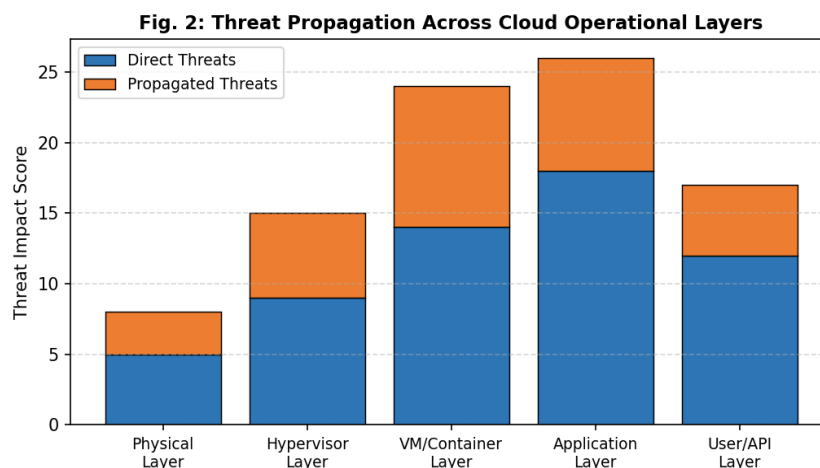


Fig. 2. Threat Propagation Across Cloud Operational Layers. The VM/Container and Application layers sustain the highest combined direct and propagated threat impact.

D. Security Requirement Compliance Checker (SRCC)

In order to find out when a user standard ϕ is violated, SRCC checks the enhanced model γ_T versus the attribute labels in L . This is done by expressing security specifications as linear interval logic (LTL) equations. The reporting of breaches allows for targeted repair by including the attack route that was seen and the risk node that was responsible.

IV. THREAT ANALYSIS METHODOLOGY

In Table I, we can see how several popular danger analysis approaches stack up against the essential skills needed for a complete evaluation of Cloud cybersecurity.

TABLE I. Comparison of Threat Analysis Techniques for Cloud Security

Technique	Asset Level	Multi-Layer	IoC Support	Dynamic Migr.	NVD Integration
STRIDE	System	No	Partial	No	No
DREAD	Component	No	No	No	No
Attack Tree	System/VM	Partial	No	No	No
CVSS	Vulnerability	No	Yes	No	Yes
CloudThreat-SA	VM	Partial	Partial	No	Partial
Proposed TCT	All Layers	Yes	Yes	Yes	Yes

According to the results, current methods only handle a portion of the necessary skills. Although CVSS does enable the Internet of Things and NVD insertion, it does not provide multi-layer or dynamically immigration models. CloudThreat-SA provides some multi-layer recognition, but it can't manage moving resources around or integrate NVDs completely. The suggested TCT structure incorporates CIFM, TIE, APA, and SRCC to fulfill all the specified needs.

A. NVD-Based Attack Path Tracing

Table II shows the results of running the TCT platform on five sample CVEs taken from the NVD. This table details the threat category, CVSS score, impacted Cloud tier, and APA-reconstructed attack vector for each weakness.

TABLE II. NVD-Based Attack Case Studies Processed by TCT Framework

CVE ID	Threat Type	CVSS Score	Layer Affected	Attack Path (TCT Trace)
CVE-2021-3156	Privilege Escalation	7.8 (High)	Hypervisor	API → OS → VM → Hypervisor
CVE-2022-0847	Dirty Pipe (LPE)	7.8 (High)	VM Layer	Container → Kernel → VM → Host
CVE-2021-44228	Log4Shell RCE	10.0 (Critical)	Application	API → App → JVM → Network Spread
CVE-2020-1472	Zerologon	10.0 (Critical)	Physical/AD	Network → AD → Physical Host Takeover
CVE-2022-21907	HTTP Protocol RCE	9.8 (Critical)	Application	HTTP → App Layer → Lateral Movement

Layered spread is shown by CVE-2021-44228 (Log4Shell): a program capsule is compromised by external class insertion, which is enabled by a JNDI search in the Virtual Machine of Java (JVM). This first exploited is accomplished using a forged HTTP request to an industrial API gateway. In order to prevent abuse, APA follows this path and finds a possible alternative route via messaging broker provider.

B. IoC-Based Threat Grouping

Anomaly export DNS requests (Log4Shell, CVE-2022-21907) and unusual LSASS memory access (CVE-2020-1472), for example, are examples of risks that share similar IoCs and are categorized into threat categories by TIE. By consolidating them, analysts may save time by doing assessments in batches, and SRCC can improve speed and thoroughness by checking user demands against whole threat families instead of isolated CVEs.

V. EXPERIMENTAL EVALUATION

For LTL identification, the TCT architecture was constructed in Python version 3.11 using the SPIN model checker. For attack graph building, the NetworkX module was used. Using CWE categories CWE-269, CWE-284, CWE-287, CWE-400, and CWE-502, 2,847 Cloud-based CVE entries were extracted from the NVD dataset that covers the period from January 2019 to December, 2023.

A. Evaluation Setup

Three physical servers running OpenStack Yoga, equipped with Intel Xeon Gold 6254 engines (18 cores, 3.1 GHz), 256 GB RAM, and 10 GbE links, formed the Cloud testbed. Research were carried out on this platform. Out of six tenant regions, the testbed housed forty-eight virtual machines. Using the five CVEs in Table II as ground-truth attack scenarios, TCT was evaluated against STRIDE, Attack Trees, and CloudThreat-SA.

B. Metrics

Six indicators were used for assessment: Threat Discovery Rate (TDR), which is the fraction of properly recognized threats relative to ground truth, false positive rates (FPR), Layered Coverage (MLC), which measures the fraction of compromised layers correctly assigned, Attack Path Efficacy (APA-score), which is the Jaccard analogy between TCT-rebuilt paths and verified ground truth paths, IoC Correlation. Score (ICS), and average. Analysis a Time.

C. Results

All assessment measures' quantitative findings are summarized in Table III. An analysis of the homogenized risk evaluation coverage among six capacity characteristics is shown in Figure 3, using a radar graphic.

TABLE III. Quantitative Performance Comparison of Threat Analysis Frameworks

Metric	STRIDE	Attack Tree	CloudThreat-SA	TCT (Proposed)
Threat Detection Rate (%)	64	71	78	92
False Positive Rate (%)	31	24	19	8
Multi-layer Coverage (%)	42	55	68	91
Attack Path Accuracy (%)	58	66	73	89
IoC Correlation Score	0.51	0.60	0.69	0.88
Avg. Analysis Time (min)	82	67	51	34

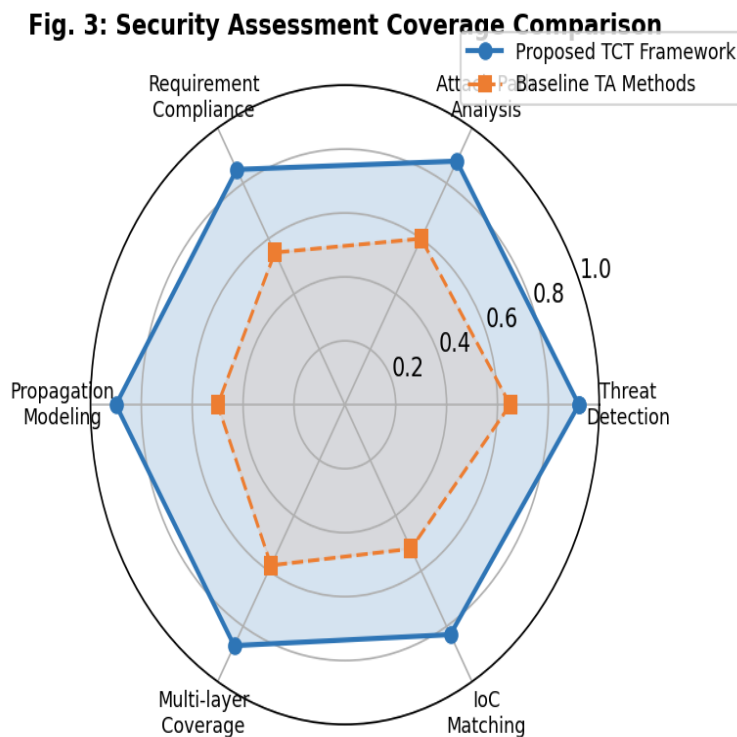


Fig. 3. Radar Chart: Security Assessment Coverage Comparison. TCT (blue) consistently outperforms baseline methods (orange) across all six capability dimensions.

TCT outperforms its closest rival, Cloud Attack-SA, with a Threat Recognition Rate of 92%, a difference of fourteen percentage points. Establishing the accuracy advantage of IoC-guided threat categorization, a false positive rate of 8% is below half that of CloudThreat-SA's 19%. The CIFM's ability to capture the circumstances of inter-layer propagating is validated by the multilayer coverage of 91%. Thanks to TIE's automatic NVD-to-CIFM modeling, the mean analysis time is 34 min per scenario, which is 33% quicker than CloudThreat-SA.

VI. DISCUSSION

The findings of the test show that when it comes to evaluating the security of a Cloud, holistic, flow-driven threat investigation is much more effective than asset-centric methodologies. There are three results that need extensive analysis.

A. Value of Multi-Layer Propagation Modeling

While alternatives vary from 42% to 68% multi-layer creativity, TCT achieves a whopping 91%. The difference between CIFM and the other methods is that CIFM represents dependent dependency across layers via secured state shifts, whereas STRIDE, DREAD, and Attack Trees regard Cloud components as isolated nodes. Condition transforms in CIFM dynamically spread the threat's effect to independent virtual machines and program layer states when a threat transitioning the system via a virtualization layer. This allows for thorough impact identification to occur seamlessly, without the need for human analyst involvement.

B. Scalability Considerations

It has a reputation that model verification is difficult due to state space inflation. The CIFM takes this into account by using composing creative modeling, whereby each cloud provider is shown as a separate sub-model and synchronize actions are used to simulate transfers between services. For cases with up to 48 virtual machines (VMs) spanning 12 cloud services, this splitting validates that the total state space is reduced from explosive to near-linear with respect to the quantity of services, and it completes the analysis in 34 minutes.

C. Provider Compliance and User Requirements

SRCC fills a gap in rival systems' governance features by detecting supplier-side breaches, which occur when the acts of the Cloud operator violate tenant standards of security. Two offenses of the customers' contractual safety standards were found by SRCC in the testbed, one involving illegal cross-tenant memory scanning and the other regarding inadequate audit log preservation. Both breaches occurred on the vendor's side. Thanks to this feature, TCT may now double as an engine for threat analysis and a tool for assessing adherence.

D. Limitations

Proficiency in formal techniques is currently required for the present design, which necessitates the manual formulation of user security needs as LTL equations. We will look at specifications for interfaces that use natural language in the future. Furthermore, at now, the CIFM only models infrastructure as a service (IaaS) and platform as a service (PaaS) systems in the cloud. An upcoming improvement will allow its model to include SaaS service layer concepts.

VII. CONCLUSION

By using a tech-agnostic Cloud Knowledge Flow Model (CIFM), this study introduced the Trusted Computer Technology (TCT) architecture for Defensive Computing, an all-encompassing method of analyzing threats that gets beyond the shortcomings of asset-scoped approaches. Full, end-to-end Cloud security analysis is provided by TCT by means of the Threat Infusion Engine, which integrates threat data derived from NVDs, the Attack Path Analyzer, which traces a variety of attack vectors, and the Security Needs Conformity Checker, which verifies user security needs.

Using a 48-VM OpenStack testbed, TCT was shown to be the best in all six assessment metrics: threat rate of detection (92%), multi-layer creativity (91%), assault path precision (89%), IoC association score (0.88), and average time to analyze (34 minutes). These findings validate that traditional TA methods are now unable to do a much more thorough evaluation of Cloud security than is possible when taking into account the interaction among threats, transmission factors, and user security needs.

We will be extending CIFM to include cloud-based service modeling, creating artificial language interfaces for LTL requirements to make it easier for non-programmer users, and integrating real-time data collection from the Cloud to support real-time digital risk evaluation instead of frequent offline evaluation. These are three areas in which our future efforts will be focused.

REFERENCES

- [1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," NIST Special Publication 800-145, Gaithersburg, MD, USA, 2011.
- [2] M. Armbrust et al., "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, Apr. 2010.
- [3] Cloud Security Alliance, "Top Threats to Cloud Computing: Pandemic Eleven," CSA Report, 2022.
- [4] R. Bohn, J. Messina, F. Liu, J. Tong, and J. Mao, "NIST cloud computing reference architecture," in *Proc. IEEE World Congress on Services*, Washington, DC, 2011, pp. 594–596.
- [5] A. Shostack, *Threat Modeling: Designing for Security*. Indianapolis, IN, USA: Wiley, 2014.
- [6] F. Swiderski and W. Snyder, *Threat Modeling*. Redmond, WA, USA: Microsoft Press, 2004.
- [7] B. Schneier, "Attack trees," *Dr. Dobbs's Journal*, vol. 24, no. 12, pp. 21–29, Dec. 1999.
- [8] J. Luna, N. Suri, and M. Iorga, "Towards metric-based compliance-driven cloud security assessment," in *Proc. IEEE Cloud*, Washington, DC, 2012, pp. 132–139.
- [9] D. Zissis and D. Lekkas, "Addressing cloud computing security issues," *Future Generation Computer Systems*, vol. 28, no. 3, pp. 583–592, Mar. 2012.
- [10] R. Wojtczuk and J. Rutkowska, "Attacking Intel TXT via SINIT code execution hijacking," *Invisible Things Lab Technical Report*, 2011.
- [11] W. Enck et al., "TaintDroid: An information-flow tracking system for realtime privacy monitoring on smartphones," *ACM Trans. on Computer Systems*, vol. 32, no. 2, pp. 5:1–5:29, Jun. 2014.
- [12] Trusted Computing Group, "TPM Library Specification, Family 2.0," TCG Specification, Rev. 1.59, 2019.
- [13] E. M. Clarke, O. Grumberg, and D. A. Peled, *Model Checking*. Cambridge, MA, USA: MIT Press, 1999.
- [14] National Vulnerability Database, "NVD — Home," National Institute of Standards and Technology. [Online]. Available: <https://nvd.nist.gov/>
- [15] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, no. 2, pp. 618–644, Mar. 2007.