EPiC Series in Computing

Volume 111, 2026, Pages 17–32

Proceedings of The 21st Asia Joint Conference on Information Security

Attacker Profiling through MTD-Elicited Behavioral Evidence and Mutation-Aware IDS Observation

Rui Liu[1] and Hiroshi Koide[2]

[1] Graduate School of Information Science and Electrical Engineering,
Kyushu University, Fukuoka, Japan
xiaorui121212@gmail.com
[2] Research Institute for Information Technology,
Kyushu University, Fukuoka, Japan
koide@cc.kyushu-u.ac.jp

Abstract

Conventional Intrusion Detection Systems (IDSs) are often deployed in static environments, where the defender's view of attackers is largely limited to direct evidence of observed attack attempts, such as request patterns, payload characteristics, and the alerts they trigger. Although such information is useful for detection, it provides only a limited view of how an attack is carried out by an attacker against a target system. Moving Target Defense (MTD) can expand this view by repeatedly invalidating attacker knowledge and forcing adversaries to disclose additional behavioral reactions, such as renewed scanning, access to expired targets and decoy interaction. However, existing studies have mostly discussed such reactions at the level of detection or visibility enhancement, while their potential for further attacker profiling remains insufficiently explored.

This paper proposes an MTD-aware attacker profiling framework that transforms MTD-elicited reactions into structured behavioral evidence. By incorporating MTD service-state knowledge into IDS observation, the framework enables attacker interactions to be interpreted according to their mutation context rather than as isolated traffic events. We evaluate the framework in a controlled Docker-based web testbed using multiple simulated attackers with different behavioral tendencies. The results show that, compared with static IDS observation without MTD, mutation-aware IDS under MTD-driven mutation provides additional behavioral dimensions that make attacker differences more interpretable and distinguishable, enabling profiling-oriented analysis beyond visibility enhancement.

1 Introduction

Intrusion Detection Systems (IDSs) are fundamental components of modern cybersecurity because they monitor traffic and system interactions for indications of malicious activity [2]. Through observed requests, payloads, alert patterns, and attack attempts, IDSs provide useful evidence for recognizing and analyzing ongoing attacks. However, many IDS deployments still operate under largely static observation conditions, where detection logic is developed around a relatively stable attack surface. When that surface remains unchanged, attackers can gradually

T.-T. Tsai, H.K. Kim, Y. Wang and A. Yamada (eds.), AsiaJCIS 2026 (EPiC Series in Computing, vol. 111), pp. 17–32

learn both its structure and the conditions under which detection is likely to occur, making evasion increasingly feasible over time. Consequently, stealthy and adaptive attacks often remain difficult to uncover [3, 7], and the defender's view may remain limited to the attack attempts themselves. Although such observations are valuable, a more complete defense process requires richer information to support later assessment, reasoning, and response.

Moving Target Defense (MTD) has been widely studied as a proactive defense paradigm that reduces attacker certainty by dynamically changing the attack surface [6, 4]. By varying elements such as IP addresses [22, 23, 12], ports [23], and network topologies [11], MTD makes reconnaissance and exploitation more difficult and forces attackers to operate against a less predictable environment. Existing studies have mainly examined MTD from the perspective of resilience improvement, attack-cost elevation, and adversarial disruption. At the same time, because MTD repeatedly invalidates previously acquired attacker knowledge, it can also disturb attacker behavior and generate observable fluctuations in the way attacks are carried out [24].

This suggests substantial potential in combining MTD with IDS. Even so, their coordination has not been sufficiently explored. In many existing studies, IDS is mainly used to support MTD decision-making, such as deciding when mutations should occur or how the environment should be changed [15, 18]. Some studies have also adopted MTD-inspired strategies to improve detection or security effectiveness by introducing uncertainty into system structure [21], switching sensor inputs [8], mutating feature spaces [19], or randomly shifting decision boundaries [10].

In addition, a small number of studies have started to state more explicitly that the value of MTD lies not only in disruption but also in induction [9]. For instance, changes in the attack surface can reveal attacker reactions that are weak or less visible under static observation, thereby creating better detection opportunities and improving IDS visibility [17].

We extend this induction-oriented view by treating MTD as a behavioral elicitation mechanism. In this view, mutation is not merely a defensive refresh of the attack surface, but a controlled change that deliberately invalidates attacker knowledge and prompts attackers to reveal how they search, adapt, select targets, and resume malicious interaction.

However, existing work still largely stops at exposing such reactions for detection opportunities or visibility enhancement. It remains necessary to move from simply exposing attacker reactions to making them analyzable and comparable, so that MTD-elicited behaviors can provide discriminative attacker-related clues for profiling beyond direct attack attempts.

From a profiling perspective, this kind of behavioral exposure is particularly valuable because attacker reactions to mutation are not uniform. In static environments, defender observations are often restricted to sparse attack attempts, which provide only limited clues about how an adversary actually operates. By contrast, when MTD repeatedly invalidates attacker knowledge, attackers may differ in how quickly they reacquire targets, how much they continue to contact expired endpoints, whether they filter decoys, how aggressively they scan, and how persistently they resume malicious interaction. These differences provide behavioral dimensions that are difficult to obtain from attack attempts alone and can serve as a basis for attacker profiling.

Motivated by this observation, we propose an MTD-aware attacker profiling framework in which MTD-elicited state changes serve as behavioral probes for IDS-based observation. Rather than treating MTD only as a defensive mechanism that hides or refreshes the attack surface, the framework intentionally uses mutations to disturb attacker knowledge and elicit observable response patterns. In this setting, MTD dynamically changes real service exposure and decoy exposure, while invalidated endpoints are retained as expired-state knowledge. The mutation-aware IDS interprets subsequent interactions with current services, expired endpoints, or decoy states, and aggregates these mutation-conditioned reactions into attacker-level tendency vectors. We evaluate this design in a controlled experimental setting with simulated attackers

and examine whether actively elicited responses can produce interpretable and distinguishable attacker profiles beyond static IDS observation.

The main contributions of this work are as follows:

- We formulate mutation-conditioned attacker profiling as a problem of transforming MTD-elicited attacker reactions into discriminative behavioral evidence, positioning MTD not only as a defense mechanism but also as a behavioral elicitation mechanism.
- We propose an MTD-aware attacker profiling framework that couples mutation-state knowledge with IDS observation, organizes mutation-elicited reactions into session-level evidence, and aggregates them into a 13-dimensional attacker-level tendency vector.
- We conduct controlled experiments with simulated attackers and show the proposed framework reveals distinguishable and interpretable tendency patterns across attacker types, supporting profiling-oriented attacker analysis beyond static IDS observation.

2 Background and Related Work

2.1 Intrusion Detection System

IDSs are developed to monitor and analyze network traffic or system behavior for indications of malicious activity [2]. They are commonly divided into signature-based and anomaly-based categories [14]. Signature-based IDSs, such as Snort [20], identify known attack patterns by matching observed traffic against predefined rules or signatures. These systems are effective against well-understood threats, but they have difficulty handling previously unseen or intentionally obfuscated attacks. In contrast, anomaly-based IDSs learn patterns of normal behavior and raise alerts when deviations are observed, providing stronger adaptability but often introducing more false positives [13].

Despite their broad adoption, IDSs still face limitations in dynamic or adversarial settings. Stealthy attacks may imitate benign traffic patterns and become difficult to separate from legitimate behavior [5]. In addition, static rule sets, fixed thresholds, or fixed feature representations can make IDSs susceptible to evasion when attackers adapt their behavior to avoid alert.

Recent studies have attempted to improve intrusion detection by refining feature representations and combining multiple detection models. Liu et al. [16] employed feature subspace decomposition and weighted decision fusion to improve multi-class intrusion detection. Their method captures localized feature dependencies across different attack categories and integrates multiple classifiers to obtain more discriminative decisions. Ahmad et al. [1] focused on fusion-based intrusion detection in heterogeneous network environments. By combining common features from multiple datasets with multiple deep learning models, their method improves the stability of detection performance across different traffic sources.

Although these studies advance detection capability, the information available to IDSs under static observation conditions is still largely limited to traffic and access events directly observed against the exposed system. IDSs can analyze requests, features, payloads, and alerts associated with attack attempts, but these observations mainly describe the attack or access itself. As a result, the defender obtains only a limited set of information dimensions for understanding attacker behavior. This motivates the use of MTD as a complementary mechanism to expand the defender's observation space.

2.2 Moving Target Defense

MTD is a proactive security paradigm intended to improve system resilience by continuously altering attack surfaces and internal configurations [6]. By introducing uncertainty and dynamism into the system's externally observable behavior, MTD seeks to interfere with the reconnaissance, planning, and exploitation stages of cyber attacks [4]. Representative MTD techniques include changing configurations such as IP addresses [22, 23, 12], service ports [23], and network topologies [11]. By repeatedly modifying these parameters, MTD increases the complexity and cost of attack planning and invalidates previously collected reconnaissance information.

A number of studies have focused on how to optimize mutation decisions and improve the effectiveness of MTD. Zhang et al. [24] proposed a deep reinforcement learning-based MTD framework to disrupt adversarial network reconnaissance. Their approach models the attacker–defender interaction and learns obfuscation actions that mislead scanning and reduce attacker inference accuracy while considering defense cost. Yoon et al. [22] proposed an attack graph-based MTD framework for software-defined networks. Their method estimates host exploitability and attack paths, and then prioritizes mutation actions for more critical or vulnerable hosts.

These studies show that MTD can reduce attacker certainty and improve system resilience. However, their main focus remains on mutation strategy, attack-cost elevation, and defense optimization. The attacker reactions produced by mutation, such as renewed scanning, repeated connection attempts, or interaction with invalidated targets, are usually treated as side effects of disruption rather than as analyzable behavioral evidence. From the perspective of this work, these reactions are important because they may provide additional attacker-side clues that are difficult to obtain from static IDS observation alone.

2.3 Integration of MTD with IDS

Although IDS and MTD have both been studied extensively, research on their integration is still relatively limited. In many existing approaches, IDS is used mainly to support MTD decision-making. For example, alerts, anomaly information, or risk estimates produced by IDS can be used to determine mutation timing or mutation-target selection [15, 18]. In such designs, IDS contributes useful input to MTD control, but the IDS itself does not necessarily obtain richer attacker-oriented information from the mutation process.

Several studies have attempted to connect MTD-inspired dynamism with IDS-related mechanisms. Sengupta et al. [21] proposed an MTD strategy for optimizing IDS placement in cloud environments. By dynamically switching among IDS deployment configurations, their approach obscures detection surfaces and improves security effectiveness while considering performance cost. Osei et al. [19] introduced a reinforcement learning-based MTD framework to enhance anomaly-based IDSs in IoT networks. Their method dynamically shuffles feature combinations across distributed IDS components, introducing uncertainty to disrupt adversarial reconnaissance and maintain detection robustness. Giraldo et al. [8] proposed a switching-based MTD mechanism for control systems, where sensor-data availability is randomly varied to improve the visibility of stealthy sensor attacks. By changing what information is available to the observer, the method increases attack detectability while maintaining control-system requirements.

These studies demonstrate that dynamic configurations can strengthen IDS-related defense by changing deployment, sensing, feature representation, or detection conditions. However, their primary goal remains detection improvement, robustness enhancement, or system security optimization. They do not mainly ask whether the attacker behaviors elicited by mutation can be further organized and interpreted as behavioral evidence.

A smaller line of work has started to emphasize that MTD can provide value not only

through disruption but also through induction. Griffioen et al. [9] discussed how changes in the attack surface can force adversaries to reveal behaviors that would be less visible in static environments. Liu et al. [17] further showed that MTD-induced reactions can improve IDS visibility by exposing attacker behaviors such as renewed probing and interactions with changed or invalidated exposure. This direction is closely related to the motivation of this paper, because it treats MTD not only as a protection mechanism, but also as a way to bring adaptive attacker behavior into view.

Nevertheless, existing work in this direction still largely remains at the level of detection opportunities or visibility enhancement. Improved visibility is an important step, but it does not by itself explain how the revealed reactions should be organized, compared, or interpreted at the attacker level. In contrast, this work investigates whether MTD-elicited reactions observed by a mutation-aware IDS can be transformed into structured behavioral evidence for profiling-oriented analysis. By moving from exposing attacker reactions to interpreting them as attacker-level tendencies, the proposed framework extends MTD–IDS integration beyond detection and toward attacker profiling.

3 Proposed Method

3.1 Framework Overview

This work proposes an MTD-aware attacker profiling framework that uses MTD-driven state changes both to elicit attacker reactions and to provide state context for IDS-based observation. In the framework, mutation is not treated only as a defensive refresh of the attack surface. Instead, each mutation acts as a controlled state change that invalidates attacker knowledge and prompts the attacker to reveal response patterns, such as renewed scanning, contact with expired endpoints, decoy interaction, or resumed malicious interaction.

As shown in Fig. 1, the proposed method proceeds in four stages. First, the MTD controller applies mutations to change the exposed attack surface and constructs mutation-aware state knowledge, including target service, decoy, and expired states. Second, attacker interactions elicited after mutation are interpreted with reference to this evolving state knowledge. Third, the interpreted reactions are organized as session-level behavioral episodes. Finally, evidence accumulated across sessions is aggregated into attacker-level behavioral tendencies that support interpretable attacker differentiation and profiling-oriented analysis.

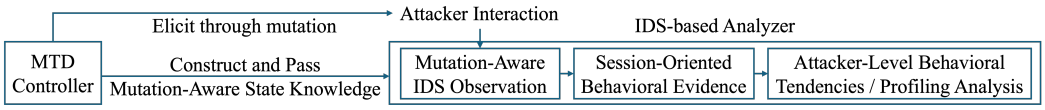

Figure 1: Overview of the proposed MTD-aware attacker profiling framework

3.2 Interaction Elicitation and State Knowledge Construction

The first stage of the framework is to use MTD to elicit attacker interactions while constructing mutation-aware state knowledge for later IDS-based observation. At each mutation, the MTD component updates the valid service endpoint and refreshes the decoy states. By invalidating previously acquired attacker knowledge, these changes may elicit reactions such as renewed scanning, continued contact with no-longer-valid endpoints, or interaction with decoy states.

At the same time, the defender records the resulting service-state history as mutation-aware state knowledge. This knowledge includes the currently valid service endpoint, active decoy states, and endpoints that were valid before mutation but have since become expired. The previously valid endpoint is no longer treated as current exposure, but its historical status is retained by the defender. This allows later interactions with that endpoint to be interpreted not as ordinary failed access events, but as expired-state interactions that reflect how the attacker responds to knowledge invalidation.

The framework requires maintaining mappings among current, expired, and decoy states. The memory overhead and the cost of correlating IDS events with state history grow with the retained mutation history, the IDS-log retention window, and traffic volume. In our prototype, this overhead remains limited by the bounded run-level history; however, large-scale or high-frequency mutation settings would require bounded state retention, log aggregation, or efficient indexing.

3.3 Session-Oriented Behavioral Observation

Given the mutation-aware state knowledge described above, the next stage is to align IDS-based observation with the attacker interactions elicited by mutation. Instead of treating traffic, alerts, and access events as independent records, the framework interprets attacker interactions with reference to the service state associated with the time of observation. In this way, interactions with current services, decoy states, and expired endpoints can be distinguished as different behavioral responses to mutation.

The framework then organizes these contextualized observations at the session level. A session is treated as a response episode associated with a MTD-driven state transition, in which the attacker may attempt to recover from the loss of previously valid knowledge. Within each session, the framework observes scanning activity, expired-target contact, decoy interaction, reacquisition effort, and resumed attack interaction. These observations provide the intermediate behavioral evidence used for attacker-level aggregation.

3.4 From Session Evidence to Attacker Profile Vector

A single session may be affected by local fluctuations, such as mutation timing, scan timing, or temporary target-selection outcomes. Therefore, the framework aggregates session-level observations over multiple mutation cycles for each observed attacker entity. The resulting behavioral dimensions summarize how the attacker tends to react to repeated knowledge invalidation, including adaptation to mutation, expired-state interaction, decoy treatment, reacquisition and scanning behavior, persistence, and payload-side characteristics.

In this work, an attacker profile is represented as a 13-dimensional behavioral tendency vector:

$$\mathbf{P}_a = (s_{a,1}, s_{a,2}, \ldots, s_{a,13}),$$

where a denotes an observed attacker entity and each component $s_{a,j}$ is a normalized behavioral score derived from mutation-aware IDS evidence. This vector provides a computable and reproducible basis for comparing attacker tendencies. Radar charts are used only to visualize the underlying 13-dimensional profile vector for qualitative profiling-oriented interpretation.

The resulting attacker-level tendencies can support subsequent assessment, prioritization, response planning, and adaptive defense adjustment.

4 Threat Model

We consider an external attacker whose objective is to discover the currently valid vulnerable service endpoint and repeatedly conduct malicious interactions against the target system. The attacker has no privileged access to the defender side and can interact with the system only through externally reachable interfaces. The attacker does not know the runtime service state maintained by MTD, including the current real service endpoint, decoy endpoints, expired-state history, mutation schedule, or the defender-side synchronization between state changes and IDS interpretation. Although the attacker may understand the general nature of the target service, it must infer usable targets from external interaction outcomes as the attack surface changes.

Under this threat model, a typical attacker follows a repeated *scan–attack–reacquire* cycle. The attacker scans the exposed port space to locate candidate targets, conducts malicious interaction against selected targets, and, after mutation invalidates previously acquired knowledge, returns to search and reacquire the true service. Different attackers may vary in scanning style, target filtering, failure tolerance, persistence, and payload-side behavior, but share the same need to operate under incomplete knowledge of the changing service state.

5 Experiments

5.1 Experimental Objectives

The experiments assess whether the proposed framework can extract profiling-oriented behavioral evidence from MTD-elicited attacker reactions, with three objectives: **observation-space expansion**, examining whether mutation-aware IDS provides behavioral evidence unavailable under static IDS-only observation; **single-policy consistency**, examining whether a single dominant policy change is reflected in the corresponding tendency-vector dimensions; and **composite-behavior interpretability**, examining whether the representation remains interpretable when multiple behavioral factors are combined.

5.2 Experimental Environment

The experiments are conducted in a controlled Docker-based testbed to isolate attacker reactions to MTD events and enable reproducible evaluation. As shown in Fig. 2, the environment consists of three main parts: a configurable attacker module, an MTD-controlled target system, and an offline IDS-based analyzer.

Configurable Attacker Module. The attacker module is external to the target system and generates controlled attacker-side interactions through the exposed interface. Each run consists of multiple sessions, following a *scan–attack–reacquire* cycle. In each session, the attacker scans the candidate port range, selects reachable targets, conducts Shellshock-like interaction, and reacquires targets after mutation invalidates previous observations.

The scan stage supports both a built-in connect-based scanner and external scanning tools such as `nmap`, `masscan`, and `unicornscan`. The built-in scanner provides lightweight customized scanning logic, while the external tools provide off-the-shelf scanning behavior. The candidate port range and scan aggressiveness can be configured for different experimental settings.

The module also includes several optional mechanisms. *Smart scanning* uses previously observed open-port information to narrow the search range in later sessions and expands the range when the target is not found. It also adjusts the scan rate according to the current

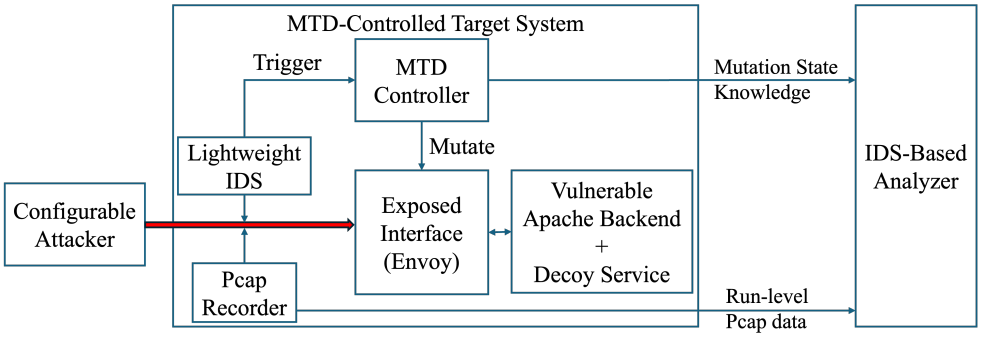

Figure 2: Overview of the Docker-based experimental environment

search range, typically resulting in a lower scan rate than full-range aggressive scanning. *Belief retention* carries previously observed target candidates into later sessions. *Surface classification* performs lightweight response-based filtering before Shellshock-like interaction, using simple externally observable properties such as reachability, HTTP-like behavior, and decoy-indicator strings. *Failure-threshold-based target abandonment* controls how many failed attempts are tolerated before the attacker stops interacting with a target.

The Shellshock-like interaction stage sends HTTP requests containing safe, non-executable Shellshock-like markers. These markers are used only as detectable traces and do not execute real commands. Their intent type and diversity are controlled through probability-based configurations, including the probabilities of exploration-, exploit-, and foothold-oriented markers, as well as the probability of adding randomized payload suffixes to simulate payload variation. Using these functions, the same attacker module is used to instantiate the simple attacker, simple-policy attackers, and composite attackers in the following experiments.

MTD-Controlled Target System. The MTD-controlled target system is a containerized web environment that exposes a mutable external access interface to the attacker. Internally, it contains an Apache-based vulnerable backend, lightweight decoy services, an MTD controller, an online lightweight IDS, and a PCAP recorder. The backend runs Apache 2.4.52 with CGI enabled, and the vulnerable CGI endpoint is executed through Bash 4.3, providing a Shellshock-related web-facing attack surface. In this study, Shellshock-like interaction is used as a stable and reproducible IDS-observable attack pattern.

The backend service is not directly exposed to attackers. Instead, Envoy v1.30-latest forwards traffic from the currently exposed service port to the internal Apache backend, allowing the externally visible access point to be changed by the MTD mechanism. Within a predefined port range, the MTD controller maintains one current web port and nine decoy ports. The decoys are lightweight TCP/HTTP-like services with configurable HTTP-like responses.

Mutation is event-driven. When the online lightweight IDS detects a Shellshock-like interaction with the current web port, the MTD controller assigns a new current web port, regenerates decoy ports, rewrites the Envoy configuration, and records the state transition. Previously used web ports become unreachable and are retained as expired states in the defender-side log. The controller provides mutation-state knowledge for later analysis, while run-level traffic is collected by the PCAP recorder.

Offline IDS-Based Analyzer. The offline IDS-based analyzer processes the recorded PCAP, Suricata alerts, and mutation-state logs after each run. Using the mutation-state knowledge provided by the MTD-controlled target system, the analyzer reconstructs sessions according to mutation events and interprets traffic with reference to current, expired, and decoy states.

Based on this mutation-aware interpretation, the analyzer extracts session-level observations such as scanning behavior, target reacquisition, expired-target access, decoy interaction, and Shellshock-like traces. These observations are then aggregated into attacker-level behavioral features for each source IP by averaging session-level feature values across repeated sessions.

5.3 Experimental Design

The experimental design is organized into three parts. Part I compares static IDS-only observation with MTD-aware IDS observation. Part II evaluates simple-policy attackers by modifying one major behavioral setting at a time. Part III evaluates composite attackers that combine multiple behavioral tendencies.

Across all experiments, the same vulnerable backend and IDS-based analysis pipeline are used. For the MTD-based experiments, the recorded PCAP, Suricata alerts, and mutation-state logs are processed under mutation-aware conditions. Attacker sessions are reconstructed according to mutation events, traffic is interpreted with reference to current, expired, and decoy states, and the resulting observations are aggregated into attacker-level behavioral features.

Part I: IDS-Only Static Baseline and MTD-Aware Observation. Part I addresses the first objective, **observation-space expansion**, by comparing a static IDS-only baseline with MTD-aware IDS observation.

For the no-MTD baseline, we use a simplified static attacker. The vulnerable service is exposed on a fixed port, set to port 80 in our experiment, and the attacker repeatedly sends inert Shellshock-like HTTP requests to that port until the predefined attempt limit is reached. This attacker does not perform scanning, target reacquisition, belief retention, surface classification, or failure-driven adaptation because there is no mutation in the environment.

For the MTD-based setting, we use a **simple attacker**, which corresponds to the simplest case under the threat model discussed above. It scans the full candidate port range from 1 to 65535 using `nmap` with a scan rate of 50000, applies a failure threshold of 5 for target abandonment, and uses exploration-oriented Shellshock-like markers without randomized suffixes. Smart scanning, belief retention, and surface classification are disabled.

Part II: Simple-Policy Attacker Scenarios. Part II addresses the second objective, **single-policy consistency**, by evaluating four attackers that each modify one major setting of the **simple attacker** in Part I while keeping the remaining settings unchanged.

The **stealthy / smart-scanner attacker** enables *smart scanning*, allowing later sessions to adjust the scan range based on previously observed open ports. The **decoy-classification attacker** enables *surface classification*, filtering candidate targets before Shellshock-like interaction. The **stubborn attacker** increases the failure threshold to 50, allowing more repeated failures before target abandonment. The **persistent attacker** increases the number of sessions to 50, producing more repeated scan–attack–reacquire cycles.

Part III: Composite-Behavior Attacker Scenarios. Part III addresses the third objective, **composite-behavior interpretability**, by evaluating simulated attackers that combine multiple behavioral factors. Their configurations are summarized in Table 1.

Table 1: Configurations of Composite Simulated Attackers

Attacker	Sessions	Tool	Smart Scan	Surface Classification	Belief Retention	Fail Threshold	Suffix Prob.	Intent Probability
Composite 1	20	builtin	Yes	Yes	No	2	1.0	`exploration=0, exploit=0.5, foothold=0.5`
Composite 2	20	builtin	No	Yes	No	5	0.3	`exploration=0.5, exploit=0.5, foothold=0`
Composite 3	20	nmap	No	No	Yes	10	0	`exploration=0.5, exploit=0.5, foothold=0`
Composite 4	50	nmap	No	No	No	2	0	`exploration=0, exploit=1, foothold=0`
Composite 5	5	nmap	No	No	No	100	0	`exploration=0, exploit=1, foothold=0`

5.4 Thirteen-Dimensional Behavioral Tendency Representation

Before presenting the results, we describe how the attacker profile vector defined in Section 3.4 is instantiated in the experimental analysis. For each source IP, session-level observations are aggregated mainly using average values across repeated sessions and mapped to the 13 behavioral scores summarized in Table 2. Each metric is mapped to a 0–4 reference score.

The score-mapping thresholds are empirically selected to distinguish qualitatively different attacker reactions under the mutation dynamics and IDS observation granularity of our testbed. They are not intended as universal constants, but as configurable parameters that can be calibrated according to deployment-specific mutation frequency, network latency, and IDS observation granularity.

5.5 Experiment Results and Analysis

5.5.1 IDS-Only Static Baseline and MTD-Aware Observation

The first experiment compares IDS-only static baseline with mutation-aware IDS observation. As shown in Table 3, the static IDS baseline captures direct attack-side evidence against the fixed vulnerable service. The observed information includes the attack source and destination, the Shellshock-like alert count, payload diversity, payload-intent distribution, and payload danger score. However, because the exposed service remains fixed and no mutation context exists, the static baseline does not provide mutation-related evidence such as target reacquisition, expired-target access, decoy interaction, or adaptation after target invalidation.

Fig. 3 shows the behavioral tendency vector of the **simple attacker** under MTD-aware observation. In addition to payload-side characteristics, the radar chart includes MTD-elicited behavioral dimensions that are not available in the static baseline. The attacker shows high scan effort, scan coverage, scan-tool dependency, and scan peak rate, reflecting repeated full-range high-rate scanning for target reacquisition. Decoy discrimination and investigation effort remain low, indicating that the attacker continues Shellshock-like interaction without filtering candidate targets before attack interaction.

The figure also shows non-negligible expired fixation and Shellshock intensity. This is consistent with the **simple attacker** setting: because surface classification is disabled, the attacker continues interacting with discovered candidates, and mutation may invalidate the current web

Table 2: Behavioral Tendency Dimensions and Score Mappings

Dimension	Raw Metric	Score Mapping
MTD awareness (delay)	Delay before scanning resumes after mutation	Piecewise: 5, 15, 60, 180 s
MTD awareness (retry times)	Newly expired TARGET/DECOY access count before the first effective scan window	Piecewise: 10, 30, 100, 300
Expired fixation	Expired port access ratio outside scan windows	Ratio $\times 4$
Decoy discrimination / confusion	Target port preference ratio in Shellshock-like interactions	Baseline ratio = 0.1; above/below baseline indicates discrimination/confusion
Investigation effort	Non-Shellshock SYN count toward current web and decoy targets outside scan windows	Piecewise: 3, 7, 15, 30
Scan effort	Scan packet count before successful target access	(scan packets/65535) $\times$ 4
Scan tool dependency	External scan-tool usage ratio	Ratio $\times 4$
Scan coverage	Scanned-port coverage ratio over the full port range (1–65535)	Ratio $\times 4$
Scan peak rate	Peak scan packet rate	Piecewise: 100, 1000, 10000, 100000
Persistence	Number of repeated attack sessions	Piecewise: 3, 10, 30, 50
Shellshock intensity	Shellshock-like event count per Shellshock-observed session	Piecewise: 2, 5, 10, 20
Shellshock payload diversity	Unique payload count per Shellshock-observed session	Piecewise: 0.2, 0.8, 1.5, 3.0
Shellshock payload danger	Weighted score from payload-intent ratios	exploration = 1, other = 2, exploit = 3, foothold = 4

Table 3: Observation Summary in the Static No-MTD Baseline

Item	Observed Result
Target exposure	Fixed service port 80
Total IDS alerts	30
Alert type	Shellshock-like attempt
Source IP	172.23.0.3
Destination port	80
Unique payload count	10
Payload intent ratio	exploit: 0.60, exploration: 0.30, foothold: 0.10, other: 0.00
Payload danger score	2.50

target while the interaction loop is still ongoing. As a result, some post-mutation interactions are interpreted as expired-target access by the mutation-aware IDS, even though belief retention is disabled.

Overall, the comparison shows that static IDS observation mainly provides direct evidence of attack attempts, whereas MTD-aware IDS observation additionally exposes behavioral evidence related to reacquisition, scanning style, expired-target interaction, decoy treatment, and continued interaction after mutation. This result indicates that, compared with the static IDS-only baseline, the proposed framework does not merely increase observable information, but provides additional behavioral dimensions that are difficult to obtain from static IDS-only observation and useful for distinguishing attacker tendencies.

5.5.2 Single-Policy Attacker Scenarios

Starting from the **simple attacker** used in the first experiment, four single-policy attackers are evaluated. Each attacker changes one major behavioral setting, and the **simple attacker**

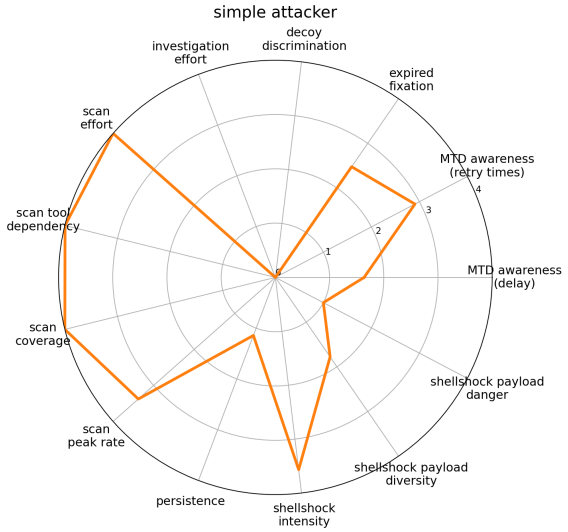

Figure 3: Behavioral tendency vector of the **simple attacker**.

is shown as the dashed baseline in each radar chart. Fig. 4 compares the tendency vectors.

For the **stealthy / smart-scanner attacker**, the main changes appear in the scan-related dimensions. Compared with the **simple attacker**, scan effort, scan coverage, and scan peak rate are reduced, reflecting the narrowed scan range and dynamically adjusted scan rate used after the initial discovery phase.

The **decoy-classification attacker** shows a clear increase in decoy discrimination and investigation effort. At the same time, Shellshock intensity and expired fixation are lower than those of the **simple attacker**, indicating that lightweight *surface classification* reduces unnecessary attack interaction with decoy or invalidated targets.

The **stubborn attacker** is mainly characterized by stronger mutation-reaction evidence. Its MTD-awareness delay, retry-times score, and expired-fixation score are higher than those of the **simple attacker**, showing that the larger failure threshold leads to longer continued interaction with failed or invalidated targets.

The **persistent attacker** is most clearly separated by the persistence dimension, corresponding to its larger number of repeated sessions. The larger session setting also makes Shellshock intensity more stable at the attacker level. In shorter runs, observer-side session reconstruction may be affected by boundary effects when the last attacker-side session triggers mutation and subsequent failure-threshold-driven attempts are assigned to an additional reconstructed session. With more repeated sessions, this effect becomes less dominant. However, payload diversity remains low because randomized payload suffixes are not used, so additional sessions mainly repeat a limited set of Shellshock-like markers rather than introducing new payload variants.

Overall, the results show that the tendency patterns reflected in the radar charts are consistent with the behavioral settings of the corresponding attackers. Rather than merely summarizing attack volume, the proposed representation clearly captures differences in policy-dependent behavioral tendencies. This indicates that the proposed framework can extract readable evidence for distinguishing attackers with different behavioral policies, supporting subsequent profiling-oriented analysis.

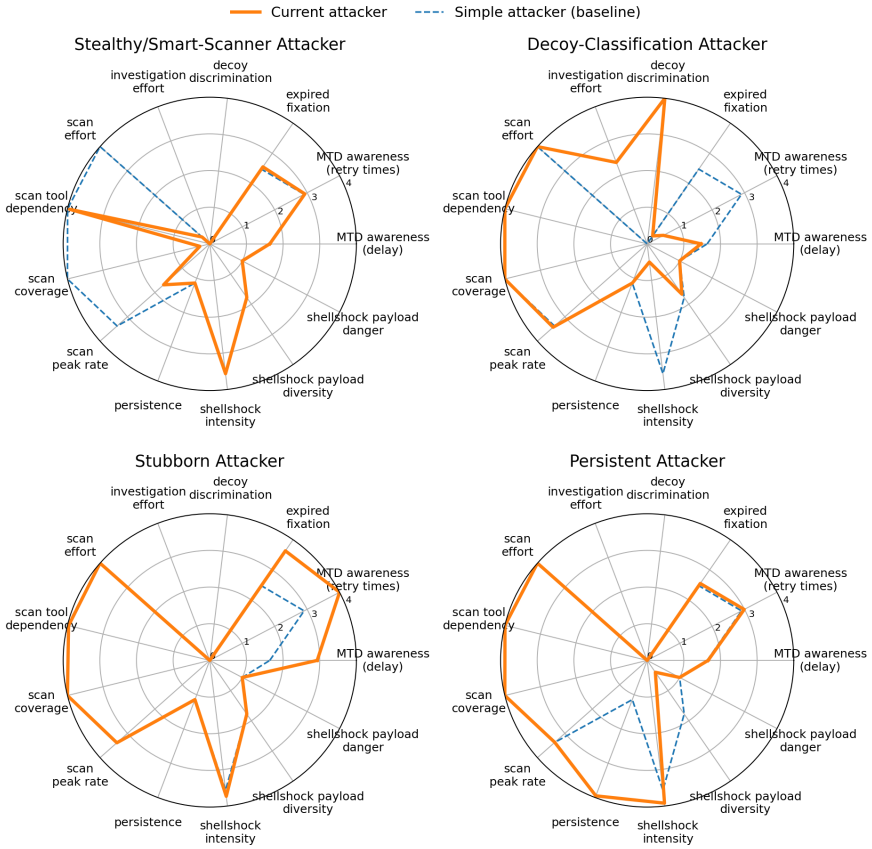

Figure 4: Behavioral tendency vectors of single-policy attackers

5.5.3 Composite-Behavior Attacker Scenarios

Fig. 5 shows the behavioral tendency vectors of the composite attackers, with the **simple attacker** shown as the dashed baseline.

Composite Attacker 1 shows a tendency pattern with reduced scan effort, scan coverage, and scan-tool dependency, while decoy discrimination, investigation effort, payload diversity, and payload danger are higher than those of the **simple attacker**. This pattern suggests that the attacker does not rely on broad external scanning alone, but narrows the reacquisition process, filters targets before attack interaction, and uses more varied and higher-risk payload markers. From a profiling perspective, this supports the inference of a selective high-risk attacker with relatively stealthy reacquisition behavior.

Composite Attacker 2 presents a different pattern. Its decoy discrimination and investigation effort are high, but scan effort and scan coverage also remain high. This indicates an attacker that still performs broad target reacquisition, while also applying target filtering before attack interaction. Its payload-side risk is higher than that of the **simple attacker**, but less prominent than that of **Composite Attacker 1**. This tendency pattern supports the inference of a broad and relatively noisy scanner with target-filtering capability and moderate payload-side risk.

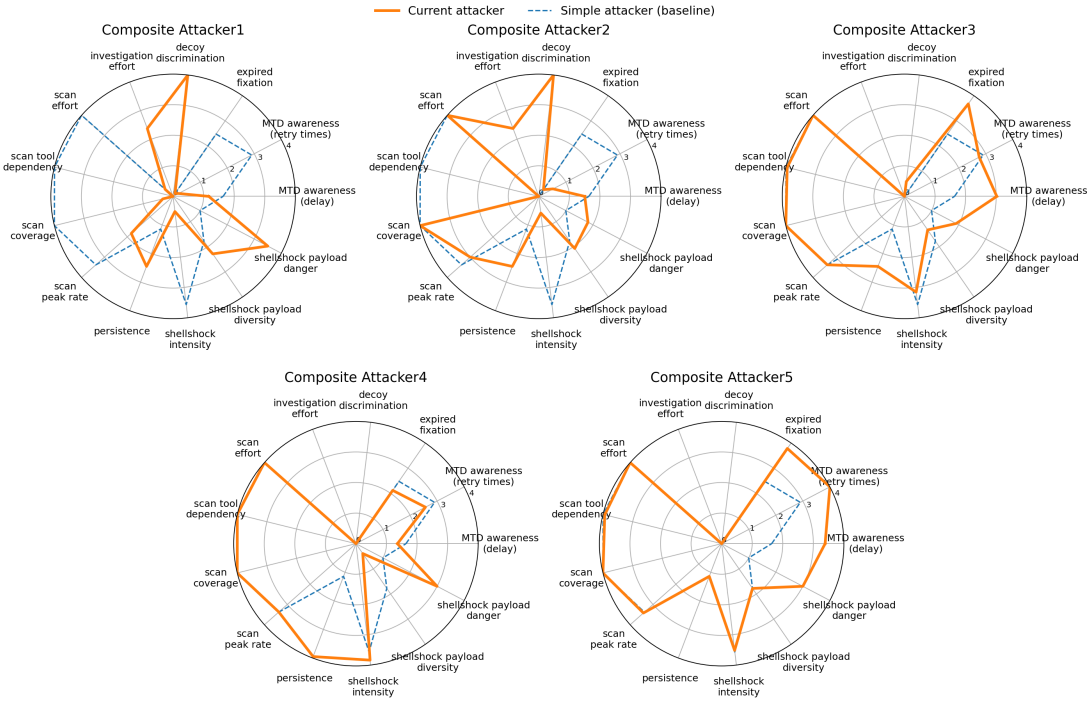

Figure 5: Behavioral tendency vectors of composite attackers

Composite Attacker 3 is characterized by high scan effort, scan coverage, and scan-tool dependency, together with stronger mutation-reaction evidence. In particular, the higher MTD-awareness delay, retry-times score, and expired-fixation score indicate that the attacker continues to rely on previously acquired target knowledge even after mutation has invalidated it. Because decoy discrimination remains low, the attacker appears less selective toward deceptive exposure. This pattern supports the inference of a noisy broad external scanner with stale-target retention.

Composite Attacker 4 is mainly distinguished by high persistence and high payload danger. The tendency vector indicates repeated long-horizon interaction and consistently high-risk payload intent, while payload diversity remains limited. This suggests an attacker whose main behavioral characteristics are not target filtering or payload variation, but repeated attack continuation with dangerous intent. From a profiling perspective, this pattern supports the inference of a persistent and conspicuous exploit-oriented attacker.

Composite Attacker 5 shows the strongest mutation-reaction pattern among the composite attackers. Its high MTD-awareness delay, retry-times score, and expired-fixation score indicate delayed target abandonment and strong continued interaction with invalidated targets. At the same time, its scan-related dimensions and payload danger remain high, suggesting broad reacquisition behavior combined with high-risk attack intent. This tendency pattern supports the inference of an aggressive, slow-adapting attacker reluctant to abandon failed targets.

Overall, the composite attackers occupy distinct regions of the behavioral tendency space, showing that the framework remains interpretable when multiple factors are combined and supports profiling-oriented differentiation beyond attack volume.

6 Conclusion and Future Work

This paper presented an MTD-aware attacker profiling framework that uses MTD not only as a defensive mechanism, but also as a mechanism for eliciting attacker responses for mutation-aware IDS observation. By intentionally invalidating attacker knowledge and interpreting the resulting reactions with mutation-state knowledge, the proposed framework organizes elicited attacker behaviors into 13-dimensional behavioral evidence, enabling subsequent analysis.

Experiments in a controlled web-based testbed show that MTD-aware observation exposes behavioral dimensions that are difficult to obtain from static IDS observation without MTD. The proposed representation also reveals interpretable and distinguishable tendency patterns among simple-policy attackers and supports profiling-oriented inference over composite simulated attackers. Overall, this work shows that MTD-elicited reactions can be transformed from additional visibility into discriminative attacker-level evidence for profiling-oriented analysis.

Several directions remain open for future work. First, beyond the Shellshock-like HTTP setting, the framework should be evaluated in noisier and more realistic environments involving background traffic, network latency, unexpected disconnections, multi-source attackers, and more diverse attack types, including slow-rate and multi-stage application-layer attacks. Second, the tendency vectors could be combined with clustering, semi-supervised learning, or other profiling-support methods for more systematic attacker grouping and profile inference. Finally, a profiling-aware adaptive defense loop could be investigated, where inferred attacker tendencies guide mutation frequency, decoy deployment, detection focus, and response policies.

Acknowledgment This study was supported by Hitachi Systems, Ltd. and JSPS KAKENHI Grant Number JP26K14823.

References

[1] Rasheed Ahmad and Izzat Alsmadi. Data fusion and network intrusion detection systems. *Cluster Computing*, 27(6):7493–7519, 2024.

[2] Hashim Albasheer, Maheyzah Md Siraj, Azath Mubarakali, Omer Elsier Tayfour, Sayeed Salih, Mosab Hamdan, Suleman Khan, Anazida Zainal, and Sameer Kamarudeen. Cyber-attack prediction based on network intrusion detection systems for alert correlation techniques: a survey. *Sensors*, 22(4):1494, 2022.

[3] A. Alhashimi, T. Nolte, and A. V. Papadopoulos. Enhancing sensor attack detection and mitigating sensor compromise impact in a switching-based moving target defense. In *Proceedings of the 2024 European Control Conference (ECC)*, pages 2560–2567. IEEE, June 2024.

[4] M. Carvalho and R. Ford. Moving-target defenses for computer networks. *IEEE Security & Privacy*, 12(2):73–76, 2014.

[5] J. Chen, X. Gao, R. Deng, Y. He, C. Fang, and P. Cheng. Generating adversarial examples against machine learning-based intrusion detector in industrial control systems. *IEEE Transactions on Dependable and Secure Computing*, 19(3):1810–1825, 2020.

[6] J. H. Cho, D. P. Sharma, H. Alavizadeh, S. Yoon, N. Ben-Asher, T. J. Moore, and F. F. Nelson. Toward proactive, adaptive defense: A survey on moving target defense. *IEEE Communications Surveys & Tutorials*, 22(1):709–745, 2020.

[7] R. R. dos Santos, E. K. Viegas, and A. O. Santin. Improving intrusion detection confidence through a moving target defense strategy. In *Proceedings of the 2021 IEEE Global Communications Conference (GLOBECOM)*, pages 1–6. IEEE, December 2021.

[8] J Giraldo, A Cardenas, and RG Sanfelice. A switching-based moving target defense against sensor attacks in control systems. *Nonlinear Analysis: Hybrid Systems*, 47:101268, 2023.

[9] Paul Griffioen, Sean Weerakkody, and Bruno Sinopoli. A moving target defense for securing cyber-physical systems. *IEEE Transactions on Automatic Control*, 66(5):2016–2031, 2020.

[10] Ke He, Dan Dongseong Kim, and Muhammad Rizwan Asghar. Mtd-ad: Moving target defense as adversarial defense. *IEEE Transactions on Dependable and Secure Computing*, 2025.

[11] K. Inoue and H. Koide. Detection and isolation malware by dynamic routing moving target defense with proxies. In *Proceedings of the 2022 International Conference on Computational Science and Computational Intelligence (CSCI)*, pages 1071–1075. IEEE, December 2022.

[12] J. H. Jafarian, E. Al-Shaer, and Q. Duan. Openflow random host mutation: Transparent moving target defense using software defined networking. In *Proceedings of the First Workshop on Hot Topics in Software Defined Networks (HotSDN)*, pages 127–132. ACM, August 2012.

[13] V. V. R. P. V. Jyothsna, R. Prasad, and K. M. Prasad. A review of anomaly based intrusion detection systems. *International Journal of Computer Applications*, 28(7):26–35, 2011.

[14] Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. Survey of intrusion detection systems: Techniques, datasets and challenges. *Cybersecurity*, 2(1):1–22, 2019.

[15] M. Kim, J. H. Cho, H. Lim, T. J. Moore, F. F. Nelson, R. K. Ko, and D. D. Kim. Evaluating performance and security of a hybrid moving target defense in sdn environments. In *Proceedings of the 2022 IEEE 22nd International Conference on Software Quality, Reliability and Security (QRS)*, pages 276–286. IEEE, December 2022.

[16] Rui Liu, Yaokai Feng, and Kouichi Sakurai. An approach to multi-class intrusion detection based on feature subspaces and weighted fusion. In *2024 IEEE Conference on Dependable and Secure Computing (DSC)*, pages 139–146. IEEE, 2024.

[17] Rui Liu and Hiroshi Koide. Enhancing ids visibility through mtd-induced behavioral exposure in web applications. In *2025 International Conference on Electrical and Computer Engineering Researches (ICECER)*, pages 1–8. IEEE, 2025.

[18] V. T. Nguyen, R. E. Navas, and G. Doyen. Lightweight security for iot systems leveraging moving target defense and intrusion detection. In *Proceedings of NOMS 2024 - IEEE Network Operations and Management Symposium*, pages 1–6. IEEE, May 2024.

[19] A. Osei, Y. Al Mtawa, and T. Halabi. Mitigating adversarial reconnaissance in iot anomaly detection systems: A moving target defense approach based on reinforcement learning. *EAI Endorsed Transactions on Internet of Things*, 10, 2024.

[20] M. Roesch. Snort: Lightweight intrusion detection for networks. In *Lisa*, volume 99, pages 229–238, November 1999.

[21] S. Sengupta, A. Chowdhary, D. Huang, and S. Kambhampati. Moving target defense for the placement of intrusion detection systems in the cloud. In *Decision and Game Theory for Security: 9th International Conference, GameSec 2018, Seattle, WA, USA, October 29–31, 2018, Proceedings*, volume 11199 of *Lecture Notes in Computer Science*, pages 326–345. Springer International Publishing, 2018.

[22] S. Yoon, J. H. Cho, D. S. Kim, T. J. Moore, F. Free-Nelson, and H. Lim. Attack graph-based moving target defense in software-defined networks. *IEEE Transactions on Network and Service Management*, 17(3):1653–1668, 2020.

[23] H. Q. Zhang, C. Lei, D. X. Chang, and Y. J. Yang. Network moving target defense technique based on collaborative mutation. *Computers & Security*, 70:51–71, 2017.

[24] T. Zhang, C. Xu, J. Shen, X. Kuang, and L. A. Grieco. How to disturb network reconnaissance: A moving target defense approach based on deep reinforcement learning. *IEEE Transactions on Information Forensics and Security*, 18:5735–5748, 2023.