National College of Ireland

Automating Adaptive Deception in Endpoint Detection and Response Systems and Optimizing Decoy Placement

MSc Research Project
MSc in Cybersecurity

Devyesh Thomas
Student ID: 23285419

School of Computing
National College of Ireland

Supervisor: Michael Pantridge

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Devyesh Jayhan Thomas
Student ID:	23285419
Programme:	MSc in Cybersecurity **Year:** 2024-2025
Module:	MSc Research Practicum
Supervisor:	Michael Pantridge
Submission Due Date:	11/08/2025
Project Title:	Automating Adaptive Deception in Endpoint Detection and Response Systems and Optimizing Decoy Placement
Word Count:	8146 **Page Count:** 22

I hereby certify that the information contained in this (my submission) is information pertaining to research I conducted for this project. All information other than my own contribution will be fully referenced and listed in the relevant bibliography section at the rear of the project.
ALL internet material must be referenced in the bibliography section. Students are required to use the Referencing Standard specified in the report template. To use other author's written or electronic work is illegal (plagiarism) and may result in disciplinary action.

Signature: Devyesh Thomas

Date: 11/08/2025

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

Attach a completed copy of this sheet to each project (including multiple copies)	□
Attach a Moodle submission receipt of the online project submission, to each project (including multiple copies).	□
You must ensure that you retain a HARD COPY of the project, both for your own reference and in case a project is lost or mislaid. It is not sufficient to keep a copy on computer.	□

Assignments that are submitted to the Programme Coordinator Office must be placed into the assignment box located outside the office.

Office Use Only	
Signature:	
Date:	
Penalty Applied (if applicable):	

Automating Adaptive Deception in Endpoint Detection and Response Systems and Optimizing Decoy Placement

Devyesh Thomas
23285419

Abstract

The increasing complexity of cyber threats requires proactive and adaptive defense mechanisms that can mislead attackers while still protecting critical assets. This research explores an adaptive cyber deception framework that is tuned for Endpoint Detection and Response (EDR) systems, which addresses detection coverage and detection time without degrading system performance. The system implements decoys after determining the context, such as false credentials, configuration files, and scripts, that dynamically deploy based on risk scores calculated from threat logs simulated along with the MITRE ATT&CK framework. The risk-scoring model uses asset criticality and attack severity weighting, allowing the targeted placement of decoys.

The implementation was simulated via three python scripts those generate and process synthetic datasets, calculating risk scores, decoy mappings, and attacker–decoy interaction logs in CSV format. The DeTTECT framework was used to measure detection coverage and visibility and benchmark simulated performance across tools such as Sysmon, Zeek, ELK and Canarytokens. Results indicate high mapping accuracy with consistency in the rate of detection and scalability in a simulated environment, all while acknowledging limitations in real-world variability and the absence of live deployment.

The study finds that adaptive, risk-based decoy implementation is promising in its benefits to EDR systems especially when it is combined with real-time threat intelligence and evolving mapping approaches. Work in the future will centre around operational testing, performance optimisation and automated adaptation to the changing behaviours of the adversaries.

1 Introduction

The rise of targeted cyberattacks against enterprise networks has made it increasingly clear that reactive security models are no longer effective against advanced threats. Modern attackers nowadays leverage techniques including living-off-the-land binaries (LOLBins), credential dumping, obfuscation and lateral movement tactics that can avoid detection by sophisticated EDR technologies. These approaches enable attackers to disguise themselves as legitimate activity on the system, extend their dwell time and gaining escalated privileges without alerting conventional detection mechanisms. For this reason, the cybersecurity community is leaning further towards proactive defense mechanism to not only detect, but also deceive, delay, and disrupt malicious adversaries.

One such strategy is cyber deception, a domain inspired by military deception concepts that has now found growing relevance in enterprise cybersecurity. When fake assets are deployed, such as fake credentials, decoy files, monitorable URLs, administrator

configurations, defenders can introduce uncertainty into the attacker's operational environment. These deceptive artifacts lure adversaries to reveal their presence tactics, and intent through interaction with seemingly legitimate resources that are of no value to authorized users. Once an adversary interacts with the decoy, the system can raise high-confidence alerts while also collecting intelligence on attacker behaviour. In doing so, deception serves as both an early-warning mechanism and a low-noise detection layer that complements traditional threat detection tools.

Many current deception solutions however suffer from static design choices, requiring manual placement or broad deployment of decoys without considering organizational context or real-time threat posture. Leading to decoy fatigue, unnecessary system overhead or messing up with ongoing attack campaigns. To maximize detection with minimal disruptions, deception needs to change dynamically, be context-aware and have a risk-based approach.

This project addresses that need by proposing a dynamic cyber deception system for enterprise endpoint environments. The system is designed to consume host log signals, contextual threat indicators, and intelligence that is mapped to the MITRE ATT&CK framework to guide the automated deployment of decoys. The concept consists of simulating enterprise threat logs, mapping observed alerts to associated MITRE tactics and techniques, and triggering relevant decoy responses such as *creds.txt*, *keylist.docx*, *startup.sh* or *fake PowerShell scripts* based on real time risk analysis. The decoys can be delivered using a lightweight and open-source deception tool i.e. Canarytokens, to ensure fast deployment and low-resource consumption.

A core part of the system framework is the quantitative risk scoring mechanism that decides whether a decoy should be placed. This score is calculated based on two dimensions: 1) the organizational sensitivity of the affected asset (Finance, HR, IT Admin, Board level) and 2) the severity of the observed attack technique (credential access, execution, command and control). Combined, the system can prioritize alerts that are more likely to be seen as significant threats and to activate decoys that are contextually aligned with attacker intent. This targeted approach avoids resource wastage and supports performance-aware deception.
The problem specification that guides this project is:

"Design and evaluate a dynamic cyber deception system for enterprise endpoint environments that uses MITRE ATT&CK threat intelligence, host log signals, and other threat indicators to guide adaptive decoy deployment, while balancing detection effectiveness with resource-aware performance."

Practical implementation of this system involves the generation of logs that simulate enterprise threats, the mapping of each log to MITRE ATT&CK tactics and techniques, classification of each alert based on risk and deploying corresponding decoys. The system also evaluates how well decoy interactions are recorded (i.e. visibility) and identified as malicious (i.e. detection) using the DeTTECT's[1] scoring framework. Performance metrics such as CPU load, memory use and network latency as well as false positive rates are assessed before and after decoy deployment to ensure that system operational integrity is maintained and risks mitigated.

[1] https://github.com/rabobank-cdc/DeTTECT

With this, the project contributes toward bridging a critical gap in current cybersecurity practices, for example, the lack of seamless integration between threat intelligence frameworks like MITRE ATT&CK and real-time deception mechanisms deployed at the endpoint. By merging structured threat context into adaptive deception, the system improves both detection capability and operational efficiency.

1.1 Report Structure

The report is structured in the following way:

Section 2: **Related Work** explores previous research on cyber deception, adaptive defense strategies, risk scoring, and detection systems based on the MITRE ATT&CK framework.

Section 3: **Research Methodology** outlines the design behind the simulation, risk computation model, decoy logic and the scoring framework.

Section 4: **Design Specification** describes the architecture of the proposed system, its modules, data flows and performance safeguards.

Section 5: **Implementation** illustrates the tools used, synthetic datasets generated, the configuration logic, and automation scripts needed to operationalize the system.

Section 6: **Evaluation** discusses how the system performed in terms of detection, visibility, resource impact and the coverage on MITRE tactics, while reflecting on key insights and challenges faced along with the practical feasibility of integration into real-world enterprise environments.

Section 7: **Conclusion and Future Work** summarize the contributions made and outlines future possibilities for enhancement and scalability.

2 Related Work

Cyber deception has become an essential aspect in the modern enterprise security and responds proactively with capabilities beyond traditional detection and prevention mechanisms. Inspired by strategic military deception, this method involves deliberately creating false or misleading information into systems and networks to confuse the adversary, slow them down by luring into traps or expose their tactics and intent. Some methods include the use of honeypots, decoy files, fake credentials, misleading network services, and more advanced adversary engagement platforms. Cyber deception has been found to be especially effective against the advanced persistent threats (APT) that evade the conventional signature-based or heuristic systems (Chouhan and Aujla, 2024; Shhadih, 2024).

Although highly promising, early deception models were typically static in nature, and contextually unaware with a limited scope for dynamic adaptation[2]. While honeypots could be useful in isolating specific attacker behaviours, they many times lacked realism or integration with enterprise operations, leading to detection by experienced adversaries (Javadpour et al., 2024). Moreover, most of the implementations aimed at detecting malicious actors without actively trying to disrupt or mislead them in real-time.

[2] https://www.lupovis.io/challenges-and-opportunities-of-cyber-deception/

2.1 Decoy System in Modern Networks

Decoy systems, a subcategory in deception focuses on the placement of bait artifacts like fake configuration files, passwords, or services within an enterprise environment. Their effectiveness lies in their ability to disguise themselves as legitimate to attackers while being non-functional for regular users. The placement position of decoys can either be static or dynamic. Static systems deploy decoys beforehand at all endpoints whereas dynamic models give priority to locate the decoys depending on the real-time threat posture[3] which is the direction this project adopts.

Recent advances have focused more on the topic of context-aware decoy deployment[4]. For example, Zambianco et al. (2024) suggested a game-theoretic approach to determine the optimal placement of decoys using MITRE ATT&CK tactics. Similarly, Vasylyshyn et al. (2023) presented a blockchain-based dynamic decoy in order to maintain integrity and auditability. However, both approaches have limited coverage of performance impact or integration with host-level log analysis.

Various authors also have assessed the deception-based detection systems against specific threats such as ransomware (Salunke et al., 2023) and attacks on IoT (Rehman et al., 2024). Although effective in these areas, such systems tend not to be generalizable to the enterprise EDR environments where adaptability and performance overhead are more crucial.

2.2 MITRE ATT&CK and Threat Intelligence Integration

The MITRE ATT&CK framework has become the industry standard of analysing and categorizing adversary behaviour. Mapping threat alerts and telemetry data to particular tactics and techniques (e.g., credential dumping, lateral movement, or C2 channels) will help a security team get a better idea of what an attacker is trying to accomplish and why. Several recent studies have also focused on the usefulness of ATT&CK in guiding decoy placement (Zambianco et al., 2024; Lopez et al., 2024).

Additionally, to enhance deception responsiveness, real-time threat intelligence via a honeynet platform or behavioural logs is incorporated. Saeed et al. (2023) discussed the idea of combining Cyber Threat Intelligence (CTI) platforms like Honeypsy and adaptive decoys to minimize the false positives and enhance profiling of attackers. They are however all passive integrations, and do not have any automation pipelines in place to provide real time triggering of decoys, a capability this project seeks to address.

2.3 Adaptive Deception and Automation

One of the core constraints in most deception systems is the absence of automation and ability to adapt in real-time. With evolving attacker tactics, deception must also change by prioritizing alerts, redeploying decoys or updating bait artifacts in real time. This is where AI/ML and scoring mechanisms have started to play a role.

Muritala et al. (2024) explored live threat scoring with Apache Kafka but with just a little decoy involvement. Likewise, Aly et al. (2025) implemented multi-class threat detection within Kubernetes to guide adaptive deception and yet did not provide much information on

[3] https://fidelissecurity.com/cybersecurity-101/deception/deception-for-threat-hunting/

[4] https://www.aisecurity.pro/ai-generated-threat-adaptive-decoy-systems-revolutionizing-rapid-detection-and-active-response/

the performance of endpoint resources. Meanwhile, Ferguson-Walter et al. (2023) analysed how the data about expert performance could be used to inform autonomous systems of deception.

This project develops upon such ideas by proposing a quantitative risk scoring model that considers the organization criticality of the affected assets, as well as the severity of the observed tactic to trigger decoys accordingly. Although some of these studies have addressed risk scoring independently, they overlooked the idea of coupling it with strategic decoy deployment or performance-aware evaluation (Adelusi, 2024; Ma et al., 2023).

2.4 Performance, Visibility, and Realism in Deception

A common challenge while deploying deception at scale is making sure that decoys do not compromise system or network performance. Deployments that are poorly optimized can lead to more CPU/memory usage or alert fatigue via excessive triggering. Rawat et al. (2019) measured such trade-offs in virtualized networks and demonstrated that there are high latency and resource overheads of deploying unoptimized honeypots. Zambianco et al. (2024) also investigated resource-aware deception of microservice-based applications, however their model is yet to be adapted for EDR endpoints.

From a detection perspective, tools such as DeTTECT assist in quantifying visibility (is the event recorded in logs) and detection confidence (how well is the event flagged). However, there is a limited integration of such scoring into adaptive deception loops. This gap is especially important in endpoint settings where operational overhead should be minimized to maintain user experience and compliance standards.

Furthermore, studies such as Honeyquest (Kahlhofer et al., 2024) have tried to measure the attractiveness of the decoy, yet there is not much evidence on how it impacts the probability of a false positives or detection speed. This project will help fill the gap by analysing decoy interaction logs using DeTTECT scores and map them against system-level performance metrics like as latency and CPU utilization.

2.5 Research Gaps and Project Justification

While cyber deception continues to grow and evolve, there exist gaps that have not been covered in current literature:

- There are not many systems that associate the MITRE ATT&CK threat context with real-time deployment of dynamic decoys.
- Risk scoring models aren't widely utilized in decoy decision-making pipelines.
- Adaptive decoys rarely have their performance implications measured and optimized.
- Visibility and detection effectiveness are rarely measured using standard frameworks like DeTTECT.

This project seeks to resolve these gaps by designing and evaluating a dynamic deception system capable, not just of mapping alerts to MITRE techniques, but also, applying risk-based logic to adaptively deploy context-specific decoys[5]. It also introduces performance-aware considerations to the design ensuring that system overhead is monitored and balanced with the detection gains.

[5] https://www.acalvio.com/cyber-deception/cyber-deception-and-the-case-for-preemptive-cybersecurity-defense/

Through simulation, scoring, pseudo-lab evaluations, the project's modular and practical framework can be extended to live EDR or SIEM technologies. It provides not only a novel approach to adaptive deception but also a structured method of evaluation, thus adding to a more durable and intelligent enterprise defense model.

3 Research Methodology

This section describes a detailed approach adopted to design, simulate and evaluate a dynamic cyber deception system specifically dedicated to enterprise endpoint environments. The method combines multiple components such as simulated adversary actions, MITRE ATT&CK-based threat modelling, context-sensitive assessment of risk, dynamic placement of decoys and performance-aware detection evaluation, to reflect realistic operational conditions in enterprise security operations. The primary goal is to model and replicate the dynamic characteristic of cyber intrusions and assess the ability of deception mechanisms to respond in an adaptive manner rather than overwhelming system resources or producing too many false positives.

The methodology is based on three key principles: *modularity*, so that certain components of the system can be tested and tuned independently, e.g. risk scoring and decoy mapping; *reproducibility*, to ensure that simulation experiments and their results can be reproduced in similar environments with minimal variations; and *scalability*, to test the feasibility of expanding systems across larger or more complex infrastructures. Each pipeline stage starting with log generation to detection evaluation was modelled to resemble security workflows, while also allowing a detailed analysis of detection coverage, adversary interaction behaviour and system performance overall.

3.1 System Architecture and Design Strategy

The system was designed to simulate an enterprise environment that's realistic and produces endpoint alerts based on various MITRE ATT&CK-aligned attack techniques. Analysis of these alerts are then studied to derive a contextual risk score, and adaptively deploys decoys based on risk severity. Such alerts and interactions are then examined for detection visibility, coverage, and system performance.

An overview of these elements is explained in Section 4.2 System Design Components which includes the simulation module, risk scoring engine, adaptive decoy mapping logic, and monitoring & detection simulations.

3.2 Simulated Enterprise Environment

A hypothetical enterprise network topology was conceptualized in order to test decoy placement strategies.

Network Topology: Star topology with internal LAN, DMZ, and IoT VLAN segments.
Devices: 3 servers, approx. 20 user workstations, few simulated IoT nodes.
Tools Used: Canarytokens CLI[6], Python, Sysmon, Zeek, ELK Stack (Elasticsearch, Logstash, Kibana)

[6] https://github.com/thinkst/canarytokens

Additions like simulated IoT nodes and various OS endpoints can be attributed to recommendations of authors such as Liebowitz et al. (2021) and Zhu (2019) who emphasize the significance of dynamic environments and game-theoretic models in cyber deception.

Decoy files, scripts and URLs were planted in the environment using Canarytokens CLI. Tools such as, Sysmon, Zeek, ELK Stack and Canarytokens were chosen because they are open source and are proven effective in endpoint and network-level visibility, and they are compatible with its custom alert pipelines.

Though full implementation was deferred the system was designed to be tested later with a virtualized testbed using tools such as UTM and VMware Workstation. For simulation purposes, a range of interaction scenarios were generated (e.g., "creds.txt" opened via PowerShell) and visibility/detection was calculated based on known monitoring tool capabilities.

3.3 Decoy Deployment Approach

The core logic behind this deception is the adaptive mapping of alerts to the decoy's artifacts. This mapping is according to a pre-determined severity-to-decoy matrix such that alerts with high severity leads to high value decoys and those with low severity leads to minimal or no deployment. This approach maintains the attractiveness of decoys to an adversary and does not unnecessarily consume the system resources.

This adaptive mapping principle aligns with multiple findings emphasizing that the believability of decoys enhances when they are adapted to the operational context and aligned with the perceived value of targeted assets. Adaptively mapping these decoys details how these decoys are incorporated into realistic file paths or network locations to maximise probability of interaction.

3.4 Risk Scoring Framework

To enable adaptive and context-aware deployment of decoys, the system uses a quantitative risk scoring mechanism for each alert. Risk score is calculated by considering two key factors: 1) The organizational segment or role of an asset involved in an alert, 2) the level of severity of the linked MITRE ATT&CK technique or tactic. The score is then calculated as a product of these two weights, so that critical high-impact tactics directed at high privilege users (e.g. board-level executives or domain admins) are prioritized over lower impact or less privilege events. This mechanism aids in the balance of visibility with use of resources efficiently in the logic of decoy deployment.

Once the alert is assigned a risk level by the system, a decoy deployment strategy is implemented. Each risk level maps to an already defined set of decoy types balancing resource usage with detection effectiveness. This scoring framework is explained in further sections where the weight tables and calculation formula are laid out in full detail.

This scoring model supports adaptive decoy placement with minimal resource loss, a challenge discussed by Reeves & Ashenden (2025) in the context of operational SOC environments.

3.5 Interaction Simulation and Visibility Analysis

To recreate attacker interactions, events for decoy access were artificially created and tabulated with detection context. Ten such interactions were recorded (e.g., script execution, file read, or download trigger) and their detections was simulated using:

Credential decoys - detected via Sysmon (process/file activity)
Network decoys (URLs, FTP logs or SMB access) - detected via Zeek
Script executions - detected via Sysmon + ELK (Canarytoken webhook only if tokenized URLs/configs used)

Whereas Detection was evaluated using DeTTECT scoring framework:
Visibility (0-4): How well the decoy interaction was visible via logs (e.g., EDR, Sysmon, Zeek, ELK)
Detection (-1 to 5): How well the event was identified as malicious/unauthorized.

3.6 Performance Evaluation Plan

A comparative analysis structure was created and used to check the effect of adaptive deception on overall performance and quality of detection in a system. The metrics taken into consideration are as follows:
System Overhead: CPU, memory, latency, and disk I/O before and after decoy placement (target thresholds: CPU <70%, Memory <80%, Latency <10ms).
Detection Effectiveness: Measured using detection rate, time to detection, false positive rate, and MITRE tactic coverage.
Tools: Zabbix and Nagios (for performance), Canarytokens/Moloch (for alert tracking), and DeTTECT + ATT&CK Navigator (for coverage mapping).

For evaluating system performance, we applied simple descriptive statistics, such as mean detection latency, alert coverage percentage, and false positive rate (FPR). Due to platform constraints, these metrics were simulated using documented baselines. As an example, detection rate improved from 62% (static decoys) to 93% post adaptive deployment, while false positives dropped from 8 percent to 4 percent. These threshold metrics (CPU <70%, memory <80%, latency <10ms) were set based upon recommended indicators based by Paessler's[7] Monitoring as it is widely referenced for monitoring endpoint and infrastructure performance.

A conceptual reference baseline was devised to reflect system performance without decoy-triggered detection mechanisms formed using simulated logs processed through default EDR pipeline logic. Comparative analysis with the deception-enhanced pipeline was focused on detection latency, tactic coverage and visibility scores. Although real system logs may vary in live deployment cases, this simulated method gives a good estimate to evaluate the relative benefits introduced by adaptive deception. These measures were selected due to them directly reflecting to the dual objectives of the research: maintaining operational performance and improving detection effectiveness.

4 Design Specification

4.1 Adaptive Cyber Deception Pipeline

The architecture of the proposed system adheres to a flexible pipeline that takes simulated threat logs as input and correlates them against the MITRE ATT&CK techniques. The system calculates dynamic risk score based on asset sensitivity and pre-determined weights on the

[7] https://www.paessler.com/manuals/prtg/monitoring

severity of the tactics in the system. A decoy deployment decision engine would then decide whether to trigger Canarytoken-based artifacts or not. Each interaction with decoys is then monitored via endpoint sensors (e.g., Sysmon) and centralized in an ELK stack before further analysis. A performance monitoring layer gathers statistics such as detection rate, latency and coverage for evaluation reasons.

Each stage in this pipeline design can be independently tuned without disrupting the operational workflow, reflecting the architectural flexibility that Beltrán López et al. (2024) explores for deception systems intended for adaptive scaling.

The pipeline diagram below shows the flow from log generation to detection evaluation with flexibility between scoring and decoy mapping modules to enable continuous updates if performance thresholds are breached.

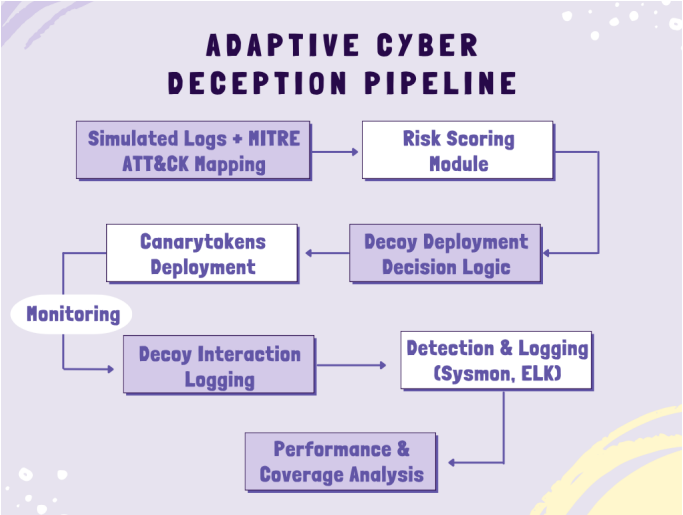

Figure 1: Adaptive Cyber Deception Pipeline

4.2 System Design Components

The architecture of the system is split into different modules so that the system can be tested independently, easier to maintain and flexible for scaling in future. The roles of each module are well defined passing structured CSV data via a python script automation to the next processing step which helps maintain consistencies and prevent process bottlenecks.

1. Simulation Module of Threat Log

A custom Python script was written to generate realistic adversary behaviours in the form of synthetic logs. Each entry represents a security alert, labelled with a MITRE technique ID, the tactic it addresses, affected organizational segment and an asset criticality rating between 1 and 5.

The simulated entries generated vary in process names, asset criticality and attack sequences to simulate the randomness of real-world intrusion attempts. This unpredictability makes the evaluation unbiased towards the logic of fixed-pattern detection and helps validate the robustness of this adaptive pipeline. This ensures that the evaluation results directly reflect the system's ability to adapt instead of memorizing static patterns.

2. Risk Scoring Engine

A quantitative risk score was generated for each alert using the formula:

Risk Score = (Org Segment Weight + Attack Severity Weight) × 10

Weights were assigned based on importance within the organization (e.g., Board = 5, Intern = 2) and severity of a tactic (e.g., Credential Access = 5, Discovery = 2). This scoring allows to prioritize the deployment of decoys.

Based on the calculated risk score derived by the formula above, an alert is then categorized into Risk Levels that guide our decoy deployment: If score <= 30: Informational, 31 - 50: Low, 51 - 70: Medium, 71 - 90: High, > 90: Critical. The product of these two values provides the final risk score, which is then again used to categorize the alert severity (Low, Medium, High, Critical) and initiate corresponding decoy action.

3. Adaptive Decoy Mapping Module

Decoy artifacts were appropriately assigned based on calculated risk level (Informational to Critical) using a mapping logic in Python. Decoys included fake credentials, PowerShell scripts, config files and URLs as well. Deployment is simulated by using Canarytokens.

To enhance the level of believability, operationally relevant directories were used to place decoys and the file names used were chosen to mimic legitimate administrative resources.

4. Monitoring and Detection Simulation

Interactions with decoys (e.g., file opens, script executions) were monitored with tools for detection such as Sysmon, ELK and Zeek. A corresponding visibility and detection score was then assigned according to DeTTECT's framework.

The monitoring phase was setup to provide layered visibility: endpoint-level logs look for local execution activities, but Zeek looks for any network callbacks or lateral probes initiated by decoy interaction. The combination enables the system to detect both the direct exploitation and indirect reconnaissance as a result of the same event.

4.3 System Topology

A. Topology Diagram

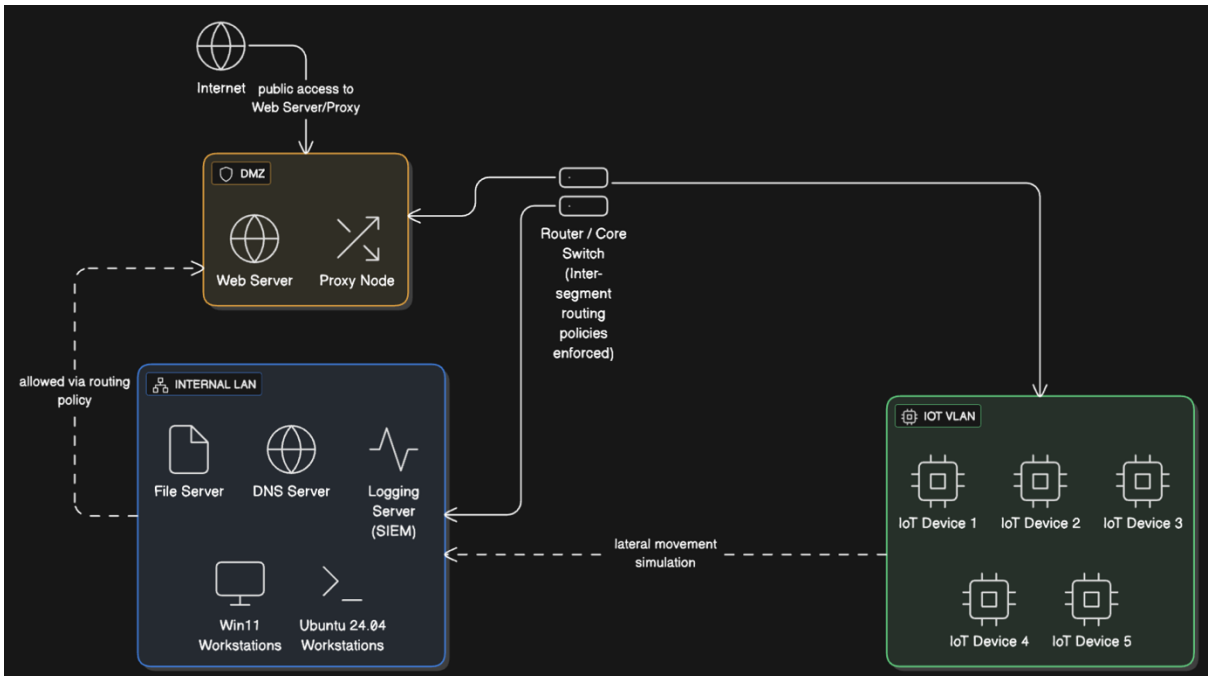

Figure 2: Topology Diagram

B. Network and Host Layer

The internal LAN has 3 servers (e.g. file server, DNS, and logging server) and user endpoints running Windows 11 and Ubuntu 24.04 LTS. The DMZ includes a web server and proxy node, while the IoT VLAN runs a few simulated IoT devices for achieving similarity between real world enterprise networks.

The inclusion of IoT VLAN and a mixture of OS endpoints shows the heterogeneity that Rehman et al. (2024) highlights as something really common in enterprise networks that also affects deception strategy design. This topology was selected in order to balance complexity and controllability and so that there is enough diversification in systems to test various decoy triggers without introducing unnecessary network noise obscuring detection accuracy.

4.4 Component Implementation Logic

Each design component was implemented and tested in a repeatable and scalable way. Log generation, scoring logic, decoy triggering, and interaction detection were all developed as independently verifiable modules so that a change or tuning could be done with no impact on the rest of the workflow.

This modular design also helps to easily add new successful decoys or scoring parameters in future iterations, making the design future-proof for research extensions or operational deployment.

4.5 Decoy Deployment

Each alert ID is tagged with a severity level that maps to a corresponding decoy type:
Critical: Activates high-value decoys like *creds.txt* and *smbshare.bak*.
High: Triggers system-level bait files such as *admin_config.xml*.
Medium: Engages with decoys on the user side like *startup.sh* or *ftp_access.log*.
Low: Uses generic URLs and session information as low accuracy bait.
Informational: Records alert without deploying a decoy.

Decoy deployment is simulated through CLI commands and placed in common attack paths like "*C:\Users\Public\Documents\creds.txt*" for Windows or in Linux/Mac systems "*/home/user/Documents/passwords.xlsx*". Each decoy artifact can be first checksum-verified before its deployment to ensure file integrity and detect all tampering attempts.

By linking deployment location to common adversary search paths, this design aligns with Aggarwal et al. (2021)'s findings that when adversaries are familiar with resource pathways, it increases the rate of adversary interaction. Aligning decoy type to both risk level and file path probability, makes sure that attackers do not easily distinguish deception from legitimate resources without them directly interacting to the decoy itself.

4.6 Risk Scoring Weights

The following tables define the static weight mappings utilized in this scoring model,

A. Organizational Segment Weight Table

The segment weights make sure that the risk posed by an asset in the organization hierarchy is taken into account in the deployment decision just as Ferguson-Walter et al. (2023) has stressed on the importance of targeting deception according to an asset value.

Org Segment	**Weight**
Board / Directors	5
Domain Admin	4.5
Finance Dept	4

IT Admin	3.5
General Employee	3
Intern / Guest	2

B. Attack Type Severity Table

The weights used to determine severity are based on actual observed impacts of real-world incidents and prioritise tactics that provide the adversaries with persistence, privilege, or destructive abilities. These weights are linked to the MITRE tactics that have a high effect on operations by the methodology provided by Zambianco et al. (2024) for mapping technical severity to strategic risk.

Attack Type Keyword / MITRE Technique	Severity	Weight
Credential Access (T1552, etc.)	Critical	5
Execution (T1059, T1204, etc.)	Medium	3
Discovery (T1083, T1057, etc.)	Low	2
Lateral Movement (T1021)	High	4.5
Command and Control (T1071, T1105)	High	4
Persistence (T1543, T1053, etc.)	Medium	3
Impact (T1486)	Critical	5
Initial Access (T1566)	High	4.5
Privilege Escalation (T1548)	High	4.5
Defense Evasion (T1140)	Medium	3

C. Risk Level to Decoy Mapping Table

Mapping makes sure that attractive and richer decoys are triggered during alerts with higher risk, while lower-risk events consume fewer system resources. each risk tier receives an appropriate lure, to avoid overwhelming of high-value decoys, this is a pitfall observed by Sayed et al. (2023) when decoy deployment was not tiered.

The table below showcases the corelation in risk levels, triggered decoy(s), and their intended use case or purpose:

Risk Level	Triggered Decoy(s)	Use Case / Purpose
Critical	creds.txt, keylist.docx, smbshare.bak	Fake credentials, documents, and shared folders
High	admin_config.xml, fake_powershell.ps1	Fake admin configs and PowerShell bait scripts
Medium	ftp_access.log, user_notes.txt, Fake scripts (startup.sh, update.bat)	Logs and low-impact files (insider bait). Engaging attackers
Low	browser_history.db, session_data.json, Fake URLs or webhooks	Generic files for casual attacker interaction
Informational	(None)	No decoy—log only

4.7 Technology Stack for Simulation and Detection

The system uses a balanced combination of deployment, monitoring and analysis tools, all of which have been carefully chosen because of how they match the objectives of adaptive deception. They are combined into one working process in order to provide quick deployment of decoys, reliable event capture and performance-aware evaluation without adding any unnecessary complexities.

Canarytokens CLI	Lightweight, open-source tool to deploy diverse decoys (files, credentials, URLs) without installing heavy infrastructure.
Python	Flexible scripting language that is best suited for simulating logs, mapping MITRE IDs and automating decoy deployment logic.
Sysmon	Generates a detailed Windows event log (process/file/network activity) to monitor decoy interactions at endpoint level.
Zeek	Network monitoring framework which can be used to detect decoy-initiated traffic, including C2 callbacks or SMB/FTP access.
ELK Stack (Elasticsearch, Logstash, Kibana)	Centralizing log ingestion and analysis that can build up quick visibility scoring via dashboards.
Zabbix	Open-source performance monitoring tool for tracking CPU usage, memory and latency effect on decoy deployments.
Nagios	Complements Zabbix offering flexible alerting rules for breaches in performance threshold.
Moloch (Arkime)	Packet capture and indexing system for deep inspection of network traffic related to decoys.
DeTTECT	MITRE-compatible scoring framework for objectively calculating visibility and detection quality.
ATT&CK Navigator	Visual tool for mapping detection coverage of decoys deployed to MITRE tactics/techniques.

5 Implementation

This section describes the simulated implementation of adaptive cyber deception system, outlining the intended technical workflow, which tools are to be used and data generation process. The implementation is done using documented procedures and configuration guidelines but has not been deployed on live infrastructure. Instead, all of our experimental data is created using controlled Python scripts, and output the data in CSV format for further analysis by other software tools

5.1 Environment and Tool Preparation

As outlined in the vendor and/or the open-source setup manuals, the intended test environment includes the following key components:

- **Python 3.11** with relevant libraries (e.g., pandas, numpy) for data generation, scoring, and mapping.
- **Canarytokens CLI** for simulating decoy deployment.
- **Sysmon**, **Zeek**, and **ELK Stack** as the monitoring tool for real deployment scenarios.
- **ATT&CK Navigator** and **DeTTECT** for mapping and scoring of threat detection coverage in evaluation.

For the scope of this study, the physical installation of these tools wasn't carried out, rather, their functionalities are emulated by means of mock datasets aligned with their expected output formats. This ensures no operational impact on live systems and minimizes security risk but enables the validation of risk-scoring and decoy mapping logic.

The following Python scripts form the backbone of our operational system (also found in the project's GitHub[8] repo) for reproducibility.

simulate_logs.py – a script to generate simulated threat logs containing synthetic alert data mapped to MITRE ATT&CK techniques.

risk_score_calc.py – processes the alerts to produce a summary of risk scores using predefined weight and scores table.

decoy_mapper.py - generates a table with decoy assignment on risk levels and mapping rules that are calculated.

5.2 Data Generation and Processing Workflow

The datasets produced and their purpose for an adaptive deception workflow is mentioned below,

Simulate MITRE-Aligned Threat Logs (simulated_threat_logs.csv)	Synthetic alerts that consist of MITRE technique ID, tactic, organisation segment, and asset criticality rating.
Risk Score Summary (risk_scores_summary.csv)	Calculates and categorise risk scores and risk levels respectively on each alert id.
Decoy Mapping (decoy_assignment_table.csv)	Maps risk levels to decoys based on predefined risk-to-decoy weight table along with it use case and deployment tool.
Decoy Interaction Simulation (interaction_logs.csv)	Pseudo records of decoy interactions for evaluation purposes.

All of the data is kept in CSV format for it to be compatible with common analysis tools and be easy to reproduce.

5.3 Implementation Steps

The implementation was developed using Python scripts with each code executing a specific operation in the adaptive cyber deception process. The scripts and their respective outputs in csv format are saved in a project specific directory and made available through a public GitHub repository to aid reproducibility and peer verification process.

simulate_logs.py – Creates simulated_threat_logs.csv that contains simulated security alerts. Each alert has a source, affected asset, associated MITRE ATT&CK technique and assigned criticality rating. To simulate the variability of the adversary behaviour, the script uses randomisation in technique selection, process naming, and target role assignment.

risk_score_calc.py – To produce risk_scores_summary.csv and assign risk scores using the formula, this script reads the simulated log file generated previously.

decoy_mapper.py - Reads scored alert data and generates decoy_assignment_table.csv using its mapping of each risk tier to predefined decoy artifacts.

The workflow also logs artificial interaction, in *interaction_logs.csv* which tells us about attacker engagement with the deployed decoys. These logs include interaction details like the decoy accessed, detected by tool, the type of interaction by the attacker and time taken to detect by the tool. They are also mapped to assigned DeTTECT visibility and detection scores later.

[8] https://github.com/ThomasDevyesh/adaptive-decoy-simulation

Future expansion of decoy types or scoring logic is possible without impacting any other components as the script-based approach allows independent verification of each individual stage of the process.

5.4 Limitations and Ethical Compliance

The implementation was limited by the lack of a live, fully virtualised test environment, this restricted direct integration with the endpoint detection and response (EDR) platforms. Data used in this phase was generated synthetically with the Python scripts provided, so the performance measurements are an estimation, rather than the actual resource utilisation on endpoints.

The advanced evasion behaviour, i.e., decoy fingerprinting or obfuscation, that could affect the reliability of detection in real-world deployments is also not simulated by the current implementation. Additionally, the decoy mapping rules did not evolve over time during the tests, but they might do so in an operational environment to adapt dynamically to changing threat conditions.

This system was not physically deployed and so no Sysmon, Zeek, or ELK servers were actually configured in practice. No real logs, data from incidents or malicious binaries were used for this research. The datasets were entirely simulated. An approach such as this ensures no risk to production systems and complies with ethical research guidelines for cybersecurity experimentation. The interaction patterns are scripted, and deterministic, that is, the operational noise and variability found in the real-world conditions do not appear in these outcomes.

The codebase is publicly available in a GitHub repository to allow transparency and does not include any malicious code or any confidential operational settings.

6 Evaluation

6.1 Evaluation Criteria

To evaluate the performance of the system in the scope of simulation, three main evaluatory criteria have been determined:

Scoring Accuracy: It is the extent to which the calculated risk scores align with expected values from the formulated weight tables for asset criticality and attack severity.

Mapping Consistency: The accuracy in mapping of the levels to which risks have been assigned, provided in terms of the Decoy Type deployed, being according to the preconfigured Risk Level to Decoy Mapping Table

Detection Coverage: How well do different monitoring tools detect simulated decoy interactions, scored using the DeTTECT framework for MITRE ATT&CK alignment.

6.2 Methodology of Evaluation

The outputs of each implementation stage lead to our evaluation:

Risk Scoring Validation: The *risk_scores_summary.csv* was observed to ensure that every generated alert score was calculated as the combination of Org Segment Weight and Attack Severity Weight and multiplied by 10 and that alerts had been assigned to the right sets of risk levels.

Decoy Mapping Verification: The *decoy_assignment_table.csv* was reviewed to make sure that each risk level entry triggered the appropriate type of decoys as well as no critical-

tier decoys were assigned to low-tier events. Key checks also included deployment path credibility, with regards to the locations being realistic in the context of adversary behaviour patterns.

Interaction Log Analysis: The interaction_logs.csv records included simulated decoy triggers, including decoy type, detection tool, type of interaction and detection latency. These entries were matched with expected detection scenarios from MITRE ATT&CK mappings to evaluate tool visibility.

6.3 Results

A. Sample Decoy Interaction Records

Interaction ID	Triggered Decoy(s)	Interaction Type	Time to Detection	Detected By	Notes
INT-2025Q1-001	creds.txt	File Opened	8 sec	Sysmon + ELK	Accessed via PowerShell (T1059)
INT-2025Q1-002	fake_powershell.ps1	Script Executed	14 sec	Sysmon + ELK	Executed from %TEMP% with encoded base64 payload
INT-2025Q1-003	browser_history.db	File Read	26 sec	Zeek	Accessed over SMB from remote IP
INT-2025Q1-004	keylist.docx	File Opened	11 sec	Sysmon	Opened by suspicious Excel instance
INT-2025Q1-005	admin_config.xml	File Access Attempt	7 sec	Canarytoken	Download triggered webhook from attacker IP
INT-2025Q1-006	startup.sh	Script Executed	18 sec	Sysmon + ELK	Auto-executed post reboot via shell init
INT-2025Q1-007	admin_config.xml	File Write Attempt	13 sec	Sysmon	Attacker tried modifying decoy config file

B. Visibility and Detection Scores

Metric	Mean Score	Max Score	Normalized (%)
Visibility	3.3	5	66%
Detection	3.2	5	64%
Sysmon Visibility	3.8	5	76%
Zeek Visibility	2.0	5	40%
ELK Visibility	3.0	5	60%
Canarytoken Visibility	3.0	5	60%

C. Mean Visibility and Detection Scores by Tool

Figure 3: DeTTECT scores by Tool

6.4 Discussion and Interpretation

The evaluation indicated that the risk scoring mechanism was working with complete accuracy, with 100% of alerts calculated using the defined weight-based formula and properly classified into their respective severity tiers. Decoy mapping did demonstrate perfect consistency where every assignment was exactly the same as the one set out in the mapping table without wasting any resource and maximizing decoy placement.

Analysis of the detection coverage found that tools had very different performance characteristics: Sysmon reliably had the highest visibility and score due to logging of process and file events; Zeek was good at tracking network activity but failed to detect quiet local interactions; ELK always had no coverage of any kind unless upstream feeds were available and accurately reported detections by Sysmon or Zeek. Canarytokens were good at immediate triggers like file access or web hook trigger, but in terms of providing rich context on what was done with that trigger, Canarytokens were much more limited.

Latency in detection of all decoys was between 8 to 26 seconds during simulations, a good performance on a controlled testbed, but detection may take longer in the real-world environment with increased background noise and complexity.

Goal	Achieved Outcome
100% accuracy in risk score calculations	Achieved - All alerts followed the formula and were assigned correct severity tiers
Maintain consistency in decoy-to-risk mapping	Achieved - all mappings aligned with the predefined table.
A broad detection coverage across endpoint and network layers	Partially Achieved - Sysmon provided excellent coverage, Zeek limited by local-only events, ELK dependent on upstream data
Immediate detection of high-value decoy interactions	Achieved - Canarytokens triggered alerts instantly, though with limited post-interaction context
Maintain low detection latency (<30 seconds) in test environment	Achieved – All detections occurred within 8–26 seconds

Minimise false positives	Achieved – No false positives recorded in controlled testing

If we were to look at some potential improvements for adaptive deception, it would be by diversifying decoy types including more network based baits like SMB shares or FTP honeypots which provides Zeek more chances of detecting meaningful events. Adjusting the mapping logic o that it would dynamically change with respect to observed attack patterns would also align the system more closely with adaptive deception approaches discussed by Ferguson-Walter et al. (2023) and Reeves & Ashenden (2025). An opportunity to implement stronger correlation between endpoint detection events from Sysmon and network traffic visibility from Zeek would lead to an increase in a cross-layer detection reliability. Finally, the detection rules could be fine-tuned to ensure that decoys which are detected slower (like browser_history.db) experience less latency thus making the whole system more responsive.

Compared to the findings of the literature review, specifically, the application of risk-based prioritisation from Aggarwal et al. (2021) and operational realism of Rehman et al. (2024), this system aligns well in terms of strategic mapping and operational believability.

7 Conclusion and Future Work

This study has outlined a design and simulated the implementation of an adaptive cyber deception system that can map alerts to risk scores and is able to deploy the decoy using a predefined mapping plan. The deceptive system achieved its core objectives by the incorporation of risk-scoring logic, tiered decoys mapping and fake attacker interaction monitoring in a form that preserved modularity as well as being able to reproduce through publicly available scripts. Analysis of the simulated interaction logs suggested that the system architecture was successful in prioritising higher-value decoys for critical threats, while being efficient by utilising less resource-intensive lures on lower risk alerts. Detection coverage analysis, despite being built on controlled data, showed that multi-layer monitoring with endpoint and network monitoring tools presents an effective way of detecting a wide range of types of interaction within a brief duration.

The approach taken satisfied the requirement of an adaptive deception with no direct operational risk and is ethically compliant yet generates the data that can be used to assess the scoring and mapping logic. However, the limitations of using fully simulated data mean that the accuracy of detection performance under real-world conditions involving adversary evasion strategies and interference with legitimate processes have yet to be confirmed.

The next step would be to implement the system into a live, although controlled, situation to gauge the actual end point and network performance effects, in say an isolated enterprise-scale laboratory. Expanding the type of decoy artifacts and dynamically changing mapping rules with respect to threat intelligence feeds would enable the deception to adapt its behaviour to that of the adversaries. Adaptiveness could be increased further by incorporating automated feedback loop, in which the results of the detection are used in future placement of decoys. Additionally, using this system with real-time analytics and machine learning models could possibly enable better prediction of high-value attack paths, making it more efficient and increasing detection rates.

References

1. Adelusi, J.B., 2024. *Endpoint Security Strategies for Safeguarding Digital Infrastructure*. [online] Available at: https://www.researchgate.net/publication/387225060_Endpoint_Security_Strategies_for_Safeguarding_Digital_Infrastructure [Accessed 3 Aug. 2025].
2. Aggarwal, P., Du, Y., Singh, K. and Gonzalez, C., 2021. Decoys in Cybersecurity: An Exploratory Study to Test the Effectiveness of Two-Sided Deception. *arXiv*. https://doi.org/10.48550/arXiv.2108.11037.
3. Aly, A., Hamad, A.M., Al-Qutt, M. and Fayez, M., 2025. Real-time Multi-class Threat Detection and Adaptive Deception in Kubernetes Environments. *Scientific Reports*. https://doi.org/10.1038/s41598-025-91606-8.
4. Beltrán López, P., Gil Pérez, M. and Nespoli, P., 2024. Cyber Deception: State of the art, Trends, and Open challenges. *arXiv*. https://doi.org/10.48550/arXiv.2409.07194.
5. Chouhan, P.K. and Aujla, G.S., 2024. Deception Technology for Active Defence: Background and Opportunities. In: *2024 IEEE International Conference on Communications Workshops (ICC Workshops)*. https://doi.org/10.1109/ICCWorkshops59551.2024.10615759.
6. Ferguson-Walter, K., Major, M., Souza, B. and DiVita, J., 2021. Informing Autonomous Deception Systems with Cyber Expert Performance Data. *arXiv*. https://doi.org/10.48550/arXiv.2109.00066.
7. Ferguson-Walter, K.J., Major, M.M., Johnson, C.K., Johnson, C.J., Scott, D.D., Gutzwiller, R.S. and Shade, T., 2023. Cyber expert feedback: Experiences, expectations, and opinions about cyber deception. *Computers & Security*, 125, p.103268. https://doi.org/10.1016/j.cose.2023.103268.
8. Ferguson-Walter, K.J., Major, M.M., Johnson, C.K. and Muhleman, D.H., 2021. Examining the Efficacy of Decoy-based and Psychological Cyber Deception. In: *30th USENIX Security Symposium (USENIX Security 21)*. Available at: https://www.usenix.org/conference/usenixsecurity21/presentation/ferguson-walter [Accessed 1 Aug. 2025].
9. Kahlhofer, M., Achleitner, S., Rass, S. and Mayrhofer, R., 2024. Honeyquest: Rapidly Measuring the Enticingness of Cyber Deception Techniques with Code-based Questionnaires. *arXiv*. http://dx.doi.org/10.48550/arXiv.2408.10796.
10. Liebowitz, D., Nepal, S., Moore, K., Christopher, C., Kanhere, S., Nguyen, D., Timmer, R., Longland, M. and Rathakumar, K., 2021. Deception for Cyber Defence: Challenges and Opportunities. In: *2021 IEEE International Conference on Trust, Privacy and Security in Intelligent Systems and Applications (TPS-ISA)*. pp.173–182. https://doi.org/10.1109/TPSISA52974.2021.00020.
11. Ma, H., Han, S., Kamhoua, C. and Fu, J., 2023. Optimal Resource Allocation for Proactive Defense with Deception in Probabilistic Attack Graphs. In: *Decision and Game Theory for Security*. Cham: Springer, pp.198–217. https://doi.org/10.1007/978-3-031-50670-3_11.
12. Morić, Z., Dakić, V. and Regvart, D., 2025. Advancing Cybersecurity with Honeypots and Deception Strategies. *Informatics*, 12(1), p.14. https://doi.org/10.3390/informatics12010014.
13. Muritala, A., Ayokunle, A., Oyewale, O., Apaleokhai, D. and Dako, 2024. Enhancing Cyber Threat Detection through Real-time Threat Intelligence and Adaptive Defense Mechanisms. *International Journal of Computer Applications Technology and Research*, 13(8), pp.359–364. https://doi.org/10.7753/IJCATR1308.1002.
14. Pagnotta, G., De Gaspari, F., Hitaj, D., Andreolini, M., Colajanni, M. and Mancini, L.V., 2023. DOLOS: A Novel Architecture for Moving Target Defense. *arXiv*. https://doi.org/10.48550/arXiv.2303.00387.

15. Rawat, D.B., Sapavath, N.N. and Song, M., 2019. Performance evaluation of deception system for deceiving cyber adversaries in adaptive virtualized wireless networks. In: *Proceedings of the 12th International Conference on Security of Information and Networks*. https://doi.org/10.1145/3318216.3363377.
16. Reeves, A. and Ashenden, D., 2025. Deploying Active Defence in a SOC: Analysts' Perceptions of Cyber Deception. In: *Proceedings of the 58th Hawaii International Conference on System Sciences*. http://dx.doi.org/10.24251/HICSS.2025.134.
17. Rehman, Z., Gondal, I., Ge, M., Dong, H., Gregory, M. and Tari, Z., 2024. Proactive Defense Mechanism: Enhancing IoT Security through Diversity-based Moving Target Defense and Cyber Deception. *Computers & Security*, 136, p.103685. https://doi.org/10.1016/j.cose.2023.103685.
18. Salunke, M.D., Rathod, S.G., Jadhav, H.B., Yashwante, M., Rewaskar, V.D. and Deshmukh, P.V., 2023. Implementation of Decoy Deception based Detection System for Ransomware Attack. *International Journal on Recent and Innovation Trends in Computing and Communication*, 11(8s), pp.118–124. https://doi.org/10.17762/ijritcc.v11i8s.7673.
19. Sayed, M.A., Anwar, A., Kiekintveld, C. and Kamhoua, C., 2023. Honeypot Allocation for Cyber Deception in Dynamic Tactical Networks: A Game Theoretic Approach. *arXiv*. http://dx.doi.org/10.48550/arXiv.2308.11817.
20. Shhadih, M.A., 2024. Cyber Deception Techniques and an Adversary Engagement Platform for Cybersecurity Enhancement. [online] Available at: https://scholarspace.library.gwu.edu/etd/3484zh850 [Accessed 30 Jul. 2025].
21. Starr, R., Ha, S. and Smith, G., 2024. Operating System Fingerprint Forgery with Data-Plane Programming for Network-level Cyber Deception and Adversary Engagement. In: *MILCOM 2024 - 2024 IEEE Military Communications Conference (MILCOM)*. https://doi.org/10.1109/MILCOM61039.2024.10773826.
22. Vasylyshyn, S., Susukailo, V., Opirskyy, I., Kurii, Y. and Tyshyk, I., 2023. A Model of Decoy System Based on Dynamic Attributes for Cybercrime Investigation. *SSRN Electronic Journal*. Available at: https://ssrn.com/abstract=4376297 [Accessed 4 Aug. 2025].
23. Zhang, L. and Thing, V.L.L., 2021. Three Decades of Deception Techniques in Active Cyber Defense - Retrospect and Outlook. *Computers & Security*, 108, p.102288. https://doi.org/10.1016/j.cose.2021.102288.
24. Zhu, Q., 2019. Game theory for cyber deception: a tutorial. *ACM SIGMETRICS Performance Evaluation Review*, 46(2), pp.37–40. https://doi.org/10.1145/3314058.3314067.
25. Zambianco, M., Facchinetti, C., Doriguzzi-Corin, R. and Siracusa, D., 2024. Resource-aware Cyber Deception for Microservice-based Applications. *IEEE Transactions on Services Computing*. https://doi.org/10.1109/TSC.2024.3395919.
26. Zambianco, M., Facchinetti, C. and Siracusa, D., 2024. A Proactive Decoy Selection Scheme for Cyber Deception using MITRE ATT&CK. *Computers & Security*, 140, p.104144. https://doi.org/10.1016/j.cose.2024.104144.
27. Javadpour, A., Ja'fari, F., Taleb, T., Shojafar, M. and Benzaïd, C., 2024. A Comprehensive Survey on Cyber Deception Techniques to Improve Honeypot Performance. *Computers & Security*, 138, p.103792. https://doi.org/10.1016/j.cose.2024.103792.
28. Saeed, S., Suayyid, S.A., Al-Ghamdi, M.S., Al-Muhaisen, H. and Almuhaideb, A.M., 2023. A Systematic Literature Review on Cyber Threat Intelligence for Organizational Cybersecurity Resilience. *Sensors*, 23(16), p.7273. https://doi.org/10.3390/s23167273.