TALLINN UNIVERSITY OF TECHNOLOGY
School of Information Technologies

Grete Ohak 232632IVCM

Enhancing Small Business Cybersecurity with Multi-Layer Threat Blocking and Retrospective Hunting

Master's thesis

Supervisors: Shaymaa Mamdouh Khalil

Cyber Security MSc

Mert Meiessaar

Cyber Security MSc

Tallinn 2026

TALLINNA TEHNIKAÜLIKOOL
Infotehnoloogia teaduskond

Grete Ohak 232632IVCM

Väikeettevõtete küberturbe tõhustamine mitmekihilise ohublokeerimise ja retrospektiivse logianalüüsi kaudu

Magistritöö

Juhendaja: Shaymaa Mamdouh Khalil

Cyber Security MSc

Mert Meiessaar

Cyber Security MSc

Tallinn 2026

Author’s declaration of originality

I hereby certify that I am the sole author of this thesis. All the used materials, references to the literature and the work of others have been referred to. This thesis has not been presented for examination anywhere else.

Author: Grete Ohak

[04.01.2026]

Tallinn 2026

Abstract

Small and medium-sized enterprises (SMEs) face increasing exposure to web-based threats but often lack the budget and expertise required for enterprise-grade security solutions. This thesis investigates whether cyber threat intelligence (CTI) from the Malware Information Sharing Platform (MISP) can be operationalised across multiple network-layer defences using lightweight, open-source tools. Building on earlier work that demonstrated CTI-driven DNS filtering, this research adds proxy-based URL enforcement and retrospective log analysis (retrohunt) to create a unified, automated framework aimed at improving detection coverage and incident visibility for SMEs.

The prototype was implemented on a single virtual machine using Pi-hole for domain blocking, Squid for URL-path enforcement with HTTPS interception, and Python scripts for CTI ingestion, synchronisation, and exact-match retrospective detection. The system was evaluated through controlled experiments, an automated test script, and a small user study.

The results show that domain and URL indicators from MISP can be transformed into accurate enforcement rules, with no false positives when warninglists are applied. The retrohunt module reliably identified earlier DNS and proxy requests associated with indicators added after the traffic occurred, improving the timeliness of incident discovery without SIEM infrastructure. Automation operated correctly under scheduled execution, and users were able to deploy the system on modest hardware.

The study demonstrates that multi-layer CTI operationalisation is feasible and practically valuable for SMEs, offering a low-cost, open-source alternative to commercial secure web gateways.

This thesis is written in English and is 50 pages long, including 7 chapters, 1 figure and 2 tables.

Lühikokkuvõte

Väikeettevõtete küberturbe tõhustamine mitmekihilise ohublokeerimise ja retrospektiivse logianalüüsi kaudu

Väikesed ja keskmise suurusega ettevõtted (VKEd) seisavad silmitsi kasvava veebiohtude survega, kuid neil puuduvad sageli ressursid ja oskused ettevõttekesksete küberturbelahenduste kasutuselevõtuks. Käesolev magistritöö uurib, kas pahatahtliku tegevuse vastane küberturbeinfo (CTI) platvormist MISP saab tõhusalt rakendada mitmes võrgukihis, kasutades vaid avatud lähtekoodiga ja kergekaalulisi tööriistu. Varasemale CTI-põhisele DNS-filtreerimise uurimusele tuginedes laiendatakse raamistikku lisades veebiproksi URL-tasemel filtreerimise ning ajalooliste logide vastega tagantjärele analüüsiga (retrohunt), eesmärgiga parandada VKEde avastamistäpsust ja nähtavust.

Prototüüp rakendati ühel virtuaalmasinal, kasutades Pi-hole'i domeenipõhiseks blokeerimiseks, Squidi HTTPS-i läbivaatusega URL-filtreerimiseks ning Pythonis kirjutatud skripte CTI hankimiseks, sünkroonimiseks ja retrospektiivseks tuvastamiseks. Süsteemi hinnati kontrollitud katsete, automaatse testskripti ja väikese kasutajauuringu abil.

Tulemused näitavad, et MISP-i domeeni- ja URL-indikaatorid on võimalik teisendada täpseteks blokeerimisreegliteks ning hoiatusnimekirjade kasutamisel ei esinenud vale-positiivseid tulemusi. Retrohundi moodul tuvastas usaldusväärselt varasemad DNS-i ja veebiproksi päringud, mis vastasid hiljem lisatud indikaatoritele, parandades seeläbi intsidentide avastamise õigeaegsust ilma SIEM-lahendusteta. Automaatne sünkroonimine toimis õigesti ning kasutajad said süsteemi paigaldada piiratud riistvaraga.

Kokkuvõttes kinnitab töö, et mitmekihiline CTI-põhine kaitse on VKEdele tehniliselt teostatav ja praktiliselt väärtuslik, pakkudes avatud lähtekoodiga ja madala kuluga alternatiivi kommertslikele turbelahendustele.

Lõputöö on kirjutatud Inglise keeles ning sisaldab teksti 50 leheküljel, 7 peatükki, 1 joonis, 2 tabelit.

List of abbreviations and terms

ACL	Access Control List
API	Application Programming Interface
CA	Certificate Authority
CERN	European Organization for Nuclear Research
CERT	Computer Emergency Response Team
CNAME	Canonical Name Record
CONNECT	HTTP CONNECT Request
CSF	Cybersecurity Framework
CSIRT	Computer Security Incident Response Team
CTI	Cyber Threat Intelligence
CTIMP	Cyber Threat Intelligence Management Platform
DNS	Domain Name System
DoH	DNS over HTTPS
DoT	DNS over TLS
DSR	Design Science Research
ELK	Elasticsearch, Logstash, Kibana
ENISA	European Union Agency for Cybersecurity
FTL	Faster-Than-Light
GEIGER	GEIGER Project (EU SME Cybersecurity Initiative)
HTTP	Hypertext Transfer Protocol
HTTPS	Hypertext Transfer Protocol Secure
IDS	Intrusion Detection System
IoC	Indicators of Compromise
IT	Information Technology
JSON	JavaScript Object Notation
MISP	Malware Information Sharing Platform
NATO	North Atlantic Treaty Organization
NGFW	Next-Generation Firewall
NIDS	Network Intrusion Detection System
NIST	National Institute of Standards and Technology

NXDOMAIN	Non-Existent Domain
OSINT	Open Source Intelligence
OTX	Open Threat Exchange
Pi-hole	Network-wide DNS Sinkhole
RDP	Remote Desktop Protocol
REST API	Representational State Transfer Application Programming Interface
RPZ	Response Policy Zones
SASE	Secure Access Service Edge
SIEM	Security Information and Event Management
SME	Small and medium-sized enterprises
SNI	Server Name Indication
SOC	Security Operations Center
SSH	Secure Shell
SSL	Secure Sockets Layer
STIX	The Structured Threat Information Expression
SWG	Secure Web Gateway
TAXII	Trusted Automated eXchange of Intelligence Information
TCP	Transmission Control Protocol
TLS	Transport Layer Security
TTL	Time To Live
URL	Uniform Resource Locator
VM	Virtual Machine
VPN	Virtual Private Networks

Table of contents

List of figures

List of tables

1 Introduction

Small and medium-sized enterprises (SMEs) form the backbone of modern economies, yet they are now among the most frequent targets of cyberattacks that were once primarily aimed at large corporations. Recent data show that around 43 % of all cyberattacks are directed at small businesses, yet only 14 % are prepared to defend against them [1]. Similar trends are reflected in European reports, where SMEs face increasing exposure to phishing, ransomware, and supply-chain incidents [2,3].

Limited budgets, a lack of in-house expertise, and growing reliance on cloud and web services make these organisations particularly vulnerable to link-based phishing and malicious URLs (Uniform Resource Locator), which now dominate modern attack campaigns. Proofpoint's 2025 Human Factor report found that malicious URLs are used four times more often than attachments in phishing messages [4] while IT Pro reported more than 3.7 billion URL-based threats globally in a single year [5]. These realities create a pressing need for affordable, automated defences that can address both domain-level and URL-level threats, areas that this thesis directly examines.

This thesis continues and expands the defended research of Mert Meiessaar [6], which demonstrated that integrating public Cyber Threat Intelligence (CTI) from the Malware Information Sharing Platform (MISP) with Pi-hole can deliver a cost-effective DNS-level (Domain Name System) protection mechanism for small businesses. While that work proved the technical feasibility and value of CTI-driven DNS filtering, it also revealed significant limitations.

DNS filtering remains a fundamental first line of defence, because every web connection begins with a domain lookup; understanding and extending this layer is essential before addressing more advanced URL-level and retrospective detection gaps. DNS filtering alone, however, cannot block malicious URLs hidden within legitimate domains or uncover earlier exposures once new IoCs (Indicators of Compromise) are published.

Building on that foundation, this research shifts the emphasis from domain-based blocking to URL-level threat prevention and retrospective threat discovery, extending

CTI automation beyond real-time filtering into historical analysis and contextual detection.

The present research addresses these limitations by introducing two complementary layers that extend the existing MISP–Pi-hole (Network-wide DNS Sinkhole) system. The first enhancement, proxy-based URL filtering, enables fine-grained inspection of web traffic at the application layer, even when encrypted. Because the proxy performs SSL-bump (Secure Sockets Layer Interception), URL-level indicators from MISP can be enforced consistently for both HTTP (Hypertext Transfer Protocol) and HTTPS (Hypertext Transfer Protocol Secure) traffic. This ensures that malicious pages hosted on legitimate domains are detected and blocked. The second addition, retrospective hunting, re-analyses historical DNS and proxy logs to identify past connections to domains or URLs that later appear in threat intelligence feeds.

Together, these extensions form a multi-layered, fully open-source cybersecurity framework tailored for SMEs, combining real-time prevention and historical detection within a lightweight, automated environment. This layered design provides small organisations with affordable visibility normally found only in enterprise systems, stronger day-to-day protection and the ability to learn from past incidents without costly commercial tools.

1.1 Problem and Motivation

SMEs remain disproportionately exposed to cyber threats but continue to struggle with the cost and operational complexity of enterprise-grade security tools. Reports such as the ENISA (European Union Agency for Cybersecurity) Threat Landscape 2024 and 2025 highlight that phishing, ransomware, and credential theft continue to dominate incidents among SMEs, while limited staff and budgets constrain adoption of managed SOC (Security Operations Center) or SIEM (Security Information and Event Management) solutions [2,3,7].

DNS-level filtering, implemented through lightweight tools such as Pi-hole and powered by public CTI feeds from the MISP, has already demonstrated a viable first line of defence for SMEs [6]. However, this protection operates only at the point of domain resolution

and cannot block malicious URLs hidden within otherwise legitimate domains, nor can it retrospectively detect compromises once new IoCs are released.

This research directly addresses those shortcomings by extending the MISP–Pi-hole framework with two additional capabilities: proxy-based URL filtering and retrospective log analysis (retrohunt). Together, these functions enable detection of web-based threats that evade domain-only controls and allow SMEs to re-examine historical traffic when new threat data becomes available. The aim is to broaden detection coverage and reduce incident discovery time without adding substantial technical or financial burden.

By introducing proxy-level inspection and retrospective analytics, the system extends protection from preventive filtering to active threat discovery, closing the blind spots most relevant to phishing, malware delivery, and cloud-service abuse. This approach aligns with modern CTI guidance that encourages defenders to use new indicators for retrospective analysis to uncover prior compromise [8]. For SMEs, the result is a practical, open-source method to achieve layered detection and forensic insight without enterprise infrastructure or licensing costs.

1.2 Thesis Proposition and Hypothesis

This thesis proposes to design, implement, and evaluate a multi-layered open-source cybersecurity system that integrates CTI-driven DNS filtering, proxy-level URL blocking, and retrospective hunting (retrohunt) into a unified, automated framework for SMEs. The proposition is that combining these complementary layers enables SMEs to achieve broader and more timely detection of web-based threats using only open-source tools.

The hypothesis is that extending the existing MISP–Pi-hole pipeline with proxy and retrohunt components will (a) increase detection coverage for malicious domains and URLs, (b) improve the timeliness of incident discovery by enabling retrospective identification of previously undetected threats, and (c) maintain affordability and usability suitable for SME environments.

1.3 Research Questions

The research is guided by one primary and three secondary questions designed to evaluate both the technical feasibility and the practical value of the proposed system.

Primary Research Question:

RQ: How effectively can cyber threat intelligence from MISP be operationalised across DNS filtering, proxy-based URL enforcement, and retrospective log analysis in a lightweight, open-source framework suitable for SMEs?

Secondary Research Questions:

SRQ1: How accurately do the DNS and proxy layers enforce domain and URL indicators derived from MISP?

SRQ2: To what extent can retrospective matching of DNS and proxy logs reveal earlier visits to malicious domains or URLs using updated CTI?

SRQ3: What level of automation can be achieved to ensure continuous CTI ingestion, enforcement, and retrospective analysis with minimal administrative effort?

SRQ4: What are the resource, configuration, and operational requirements for deploying the integrated system in SME-like environments?

These questions aim to determine whether a layered, open-source design can deliver measurable gains in detection and resilience while remaining practical for organisations with limited technical resources.

1.4 Aim and Objectives

The aim of this thesis is to design, implement, and evaluate a fully open-source cybersecurity framework that extends CTI-driven DNS filtering with proxy-level URL enforcement and retrospective threat hunting (retrohunt). The framework seeks to demonstrate that SMEs can achieve multi-layered protection, combining real-time blocking with historical detection, while remaining affordable, lightweight, and easy to maintain.

Building directly on Meiessaar's [6] validated MISP–Pi-hole model, the present research extends its functionality through practical automation and forensic capabilities aligned with modern web-based threat dynamics. The framework is assessed in controlled laboratory and SME-like environments, focusing on its technical performance, detection coverage, and operational feasibility.

To achieve this aim, the study pursues five interrelated objectives. First, it develops an integrated prototype that connects the MISP with both DNS and proxy enforcement layers, automatically translating IoCs into actionable blocklists for domains and URLs. Second, it implements and tests proxy-based URL filtering, using open-source tools such as Squid, to detect and block malicious URL connections (for example Google Drive links) that may bypasses DNS-only controls. Third, it introduces a retrospective hunting module that re-scans stored DNS and proxy logs using exact matching against updated CTI indicators to identify earlier visits to malicious domains or URLs. Fourth, evaluate detection correctness, blocking behaviour, and retrospective matching accuracy. Finally, it examines the system's usability and deployment practicality in SME-scale settings with testgroup (4 persons), assessing resource requirements, automation level, and ease of configuration.

These objectives verify whether a layered, automated, and open-source approach can deliver measurable improvements in visibility, detection accuracy, and resilience for SMEs, offering an accessible alternative to costly enterprise-grade defences.

1.5 Scope and Assumptions

This research focuses on SMEs and evaluates an open-source, on-premise cybersecurity framework that integrates the MISP with Pi-hole at the DNS layer and a proxy component such as Squid at the web layer. The scope is confined to network-level enforcement of threat intelligence indicators and retrospective log analysis for missed threats. Endpoint protection, SIEM integration, and commercial intelligence feeds are excluded to maintain cost neutrality and reflect the practical constraints faced by SMEs.

It is assumed that the target environment can host the system on a Raspberry Pi or virtual machine, retain at least two weeks of DNS and proxy logs, and route web traffic through a local proxy for policy enforcement. These assumptions reflect realistic capabilities for

small organisations and ensure that the study remains feasible under typical SME resource limitations.

1.6 Limitations

This work represents a prototype, and long-term maintenance was not evaluated, making sustained operational use potentially challenging. Although the system performs well in controlled conditions, very large MISP deployments may slow API (Application Programming Interface) queries even though the prototype processes only current indicators rather than historical events. TLS (Transport Layer Security) interception relies on a locally trusted root certificate, which in this thesis was configured only for the Firefox instance inside the virtual machine; extending interception to additional devices would require further configuration and development. As a result, the findings should be interpreted as representative of small-office environments rather than large enterprise networks. The use of a locally trusted certificate authority for HTTPS inspection introduces organisational visibility into employee web traffic and therefore requires transparent communication and appropriate policy alignment. Variability in the quality of public CTI feeds may introduce false positives, although this risk is reduced through the use of MISP warninglists, keeping in mind that warninglists themselves may occasionally contain inaccuracies.

1.7 Novelty and contribution

This thesis extends the previously validated MISP–Pi-hole DNS filtering model by introducing proxy-based URL enforcement and retrospective threat hunting (retrohunt) to create a unified, automated, multi-layered defence framework [6]. The novelty lies in demonstrating the combined operation of DNS and proxy enforcement under a shared CTI pipeline that automatically translates indicators into active blocking rules and re-evaluates historical DNS and proxy logs using exact matching when new threat intelligence becomes available. While earlier work confirmed the feasibility of CTI-driven DNS protection for SMEs, this research advances it further by showing that URL-level inspection and retrospective correlation can be implemented entirely with open-source components and minimal infrastructure. The resulting system integrates prevention and post-incident detection within a single workflow, offering small

organisations a practical means to enhance cyber-resilience without the cost or complexity of enterprise solutions.

2 Background and Related Literature

This chapter provides the technical and research foundation for the proposed CTI-driven, multi-layered defence framework. It explains how key technologies, cyber threat intelligence, DNS-based filtering, proxy-level enforcement, and retrospective hunting, operate and how prior studies have applied them. It then summarises the research gap that motivates this work. By linking technical context with related research, the section clarifies both the rationale for each chosen component and how they jointly support SMEs.

2.1 Cyber Threat Landscape for SMEs

Recent evidence confirms that SMEs remain disproportionately targeted by cyberattacks. A systematic review spanning 2017–2023 (77 studies) describes a clear shift by attackers toward SMEs, driven by exploitable structural weaknesses [9]. A 2024 review aligned with the NIST (National Institute of Standards and Technology) & CSF (Cybersecurity Framework) reaches similar conclusions, noting persistent exposure to phishing, ransomware, credential theft, and web-based intrusions alongside weak detect–respond–recover capabilities [10]. A 2025 survey of 63 SMEs finds preparedness gaps are common and that even modest IT (Information Technology) investment strongly predicts better readiness, indicating that under-resourcing continues to shape outcomes [11]. Practitioner-oriented synthesis in 2024 reinforces the high prevalence of phishing and substantial third-party exposure in SME incidents [12]. This continuing trend defines the problem space for this research: smaller organisations face the same classes of attacks as large enterprises but must defend them with far fewer tools and staff.

The underlying causes of this vulnerability are well-established. Budgets are tight, teams are small, and specialist security skills are rare [19]. In a two-year UK field study, observed SMEs spent just under €500 per year on cybersecurity on average, while costs rose quickly once enterprise NGFW (Next-Generation Firewall)/IDS (Intrusion Detection System) features were enabled [13]. Many SMEs cannot afford “advanced firewall and event management” (i.e., SIEM) or higher-tier cloud security add-ons, and managed service models often charge extra for the very protections that make a difference, driving up ongoing costs [14,15]. At the same time, legacy systems, irregular patching, and

limited incident-response or continuity planning remain common among SMEs. Many risk-assessment frameworks assume a level of cybersecurity expertise that small organizations typically lack, while low staff awareness further increases exposure to phishing and other social-engineering attacks [10,11,12]. These limitations illustrate why the present study concentrates on automating security functions that normally demand expert oversight, using open-source platforms capable of operating within restricted budgets.

Attack patterns track these constraints. Phishing and business email compromise dominate initial access; one synthesis reports sources attributing roughly 53% of incidents to phishing and about 94% of malware delivery to email [12]. Ransomware and commodity malware remain widespread, often pairing encryption with data theft and multi-layered extortion [9,10,11,12]. Supply-chain attacks are also common, often involving the compromise of managed service providers or their remote-management tools. These risks are amplified when lower service tiers exclude advanced security features by default. Data breaches frequently result from stolen credentials and the exploitation of exposed remote-access services such as Remote Desktop Protocol (RDP), Virtual Private Networks (VPNs), or misconfigured cloud systems [9,10,12,15]. Because these attacks usually begin with a user resolving or clicking a malicious domain or URL, the network layers responsible for DNS resolution and web access become natural points for prevention, an idea that underpins the layered approach developed later in this thesis.

Under these conditions, traditional enterprise defences fit poorly. Effective NGFW/IDS deployments become expensive as subscriptions and threat-focused features accumulate, outpacing typical SME budgets observed in practice [13]. SIEM requires data ingestion, normalization, detection engineering, and sustained triage, capabilities most SMEs lack [9,10]. Managed SOC offerings often meter advanced capabilities as add-ons and still depend on local containment and integration work that many SMEs cannot execute reliably [10,11,14,15]. These practical barriers directly motivate the thesis focus on scalable, automated mechanisms that extend protection without introducing enterprise-level cost or complexity.

A more practical path is to pair URL-level filtering with CTI-based automation. In experimental evaluations, BIND's DNS Response Policy Zones (RPZ) mechanism demonstrated how structured threat data can drive real-time blocking of known malicious

domains, a concept that later evolved into URL- and proxy-level enforcement models relevant to SME deployments [16]. Complementary research shows that CTI systems designed for SMEs can automatically collect community-shared intelligence (e.g., through MISP), prioritize threats based on local context, and generate concise, actionable recommendations. This reduces the analytical workload and supports automated enforcement across both DNS and URL layers. In addition, threat-based risk-assessment methods can translate local operational data into prioritized, automation-ready actions [17,18]. These prior efforts form the research baseline for this work, which extends them by integrating proxy-based URL filtering and retrospective analysis into a unified, open-source framework suitable for small organisations.

SMEs need lightweight, automated defences, particularly CTI-driven URL filtering complemented by DNS policy enforcement, because this combination balances cost, simplicity, and security effectiveness under real-world SME constraints.

2.2 Cyber Threat Intelligence (CTI) and MISP

CTI refers to the systematic practice of collecting, analysing, and sharing IoCs and their contextual information to transform raw security data into actionable knowledge for prevention, detection, and response. Its core goal is an evidence-driven cycle in which organizations continuously acquire observables, interpret them in light of adversary behaviour, and disseminate timely outputs that other defenders and systems can consume automatically [17,20,21]. CTI therefore forms the intelligence backbone for both proactive blocking and retrospective analysis, supplying the indicators that guide the detection and prevention logic used throughout this research.

Machine-readable intelligence exchange is achieved through formal data standards and platform-specific schemas. The Structured Threat Information Expression (STIX) format provides a shared language for modelling threat entities and their relationships, while the Trusted Automated eXchange of Intelligence Information (TAXII) protocol defines how those objects are transported between systems. In operational environments, many platforms, including MISP, also expose native JSON (JavaScript Object Notation) schemas and REST APIs (Representational State Transfer Application Programming Interface) that underpin day-to-day automation. In practice, these standards act as gateways for interoperability, whereas the platform-specific JSON and API layers form

the practical backbone of most automated CTI workflows [17,20,21,22]. This standards-plus-API combination permits intelligence to flow between tools with minimal manual transformation. Such structured exchange is critical for enabling automated enforcement, where CTI objects can be transformed into DNS or URL blocklists through automated synchronisation workflows. and reused later to drive retrospective scans of historical network logs.

Automation is particularly critical for SMEs. Studies of SME needs consistently argue that these organizations cannot triage large volumes of threat data by hand and therefore require pipelines that ingest CTI, prioritize what matters, and deliver directly actionable, automatically updated outputs suited to limited staff and budgets [17]. Within this thesis, the same automation principle supports both continuous proxy blocklist updates for URL filtering and scheduled retrohunt processes, ensuring defences evolve automatically as new CTI becomes available.

2.2.1 MISP

MISP is an open-source CTI platform that realises this vision both technically and operationally [26]. It comprises a web application backed by a database and a set of background workers responsible for correlation, enrichment, feed ingestion, and synchronisation, all accessible through a comprehensive REST API for integration and automation [22,23]. The platform's data model is built around Events that contain typed Attributes (IoCs) and Objects, structured bundles of related data, enriched with Tags, Taxonomies, and Galaxies (e.g., MITRE ATT&CK techniques or malware families) to provide rich, queryable context. Public feeds and peer-to-peer synchronisation (push, pull, or selective "cherry-pick") are governed by distribution levels and Sharing Groups, while automatic correlation, sightings (including false-positive feedback), and decay or scoring models ensure that indicators remain current and operationally useful [17,20,24,25]. This modular, API-driven design makes MISP particularly suitable for downstream integrations such as automated proxy-based URL enforcement and retrospective log correlation, two functions central to this research.

MISP was selected for this research because it is open-source, free of licensing costs, and widely adopted across the CTI community, including CSIRTs (Computer Security Incident Response Team), CERTs (Computer Emergency Response Team), and major

public-sector organisations such as NATO (North Atlantic Treaty Organization) and the European CERT ecosystem, an indicator of governance maturity and reliability of shared content [17,20,22]. The platform enables robust automation through its REST API and extensive ecosystem, with documented integrations to detection and monitoring systems; academic studies frequently emphasise its use in IDS and NIDS (Network Intrusion Detection System) pipelines [20,22,24,25]. MISP also supports standards-based interoperability via STIX import/export and TAXII components and, in practical deployments, can integrate with network enforcement controls such as DNS resolvers and web proxies, even though such applications are less explored in current academic literature [17,20,21,22]. These integration capabilities make MISP uniquely appropriate for this study, where CTI feeds must be automatically exported to proxy-level URL filters and correlated with historical DNS and proxy logs during retrohunt. For SMEs, MISP's feed mechanisms and federation model are particularly valuable because they provide access to public CTI (OSINT- Open Source Intelligence) without licensing barriers, enabling smaller organisations to join and benefit from trusted intelligence-sharing communities at minimal cost [17,20,22].

MISP represents the most practical CTI platform for SMEs, combining openness, automation, and structured intelligence exchange within a mature, community-supported ecosystem.

2.3 DNS Filtering and Pi-hole

Most network communications begin with the DNS, making name resolution a natural control point for preventive security. When a client resolver queries a recursive server, it first checks the cache and, if needed, performs iterative lookups, root, top-level domain, then authoritative, before caching results per TTL (Time to Live). If the resolver intentionally returns NXDOMAIN (Non-Existent Domain), an empty response, or a sinkhole address (e.g., 0.0.0.0 or ::), the connection attempt is blocked before any TCP (Transmission Control Protocol) or TLS handshake occurs. Applied at a shared resolver, this policy provides uniform protection across devices and applications and, when refreshed with current threat intelligence, offers an effective, low-maintenance defence for organisations, including SMEs [27,28]. DNS filtering thus intercepts threats at the earliest possible stage of network communication, before payload delivery or content

inspection are required. This principle formed the foundation of Mert Meiessaar's successful research [6], which demonstrated that CTI-driven DNS filtering using MISP and Pi-hole can significantly reduce malicious connections for small enterprises while remaining affordable and easy to manage [6].

Pi-hole implements this model by inserting a policy-aware resolver between clients and upstream DNS. Built on dnsmasq and the FTL (Faster-Than-Light) engine, it evaluates queries, including CNAME (Canonical Name Record) chains, against blocklists ("gravity") and, on a match, returns a non-resolving answer to sinkhole the request. Its web dashboard and statistics provide visibility for verification and tuning, while the lightweight footprint enables deployment on Linux or Raspberry Pi with minimal resources. Studies report network-wide blocking of unwanted domains in campus and SME-like environments, confirming its operational feasibility on modest hardware [29,30,31]. The project's open-source documentation further confirms active community support and consistent performance benchmarks [32]. In this thesis, Pi-hole continues to serve as the DNS-level enforcement layer, the first defensive boundary within a multi-layered architecture that adds proxy-based URL filtering and retrospective analysis.

Beyond manual configuration, Pi-hole's command-line and API interfaces enable automation: scripts can update blocklists, modify domains, and export configurations, supporting CTI-driven enforcement. Indicators from platforms such as MISP are normalised to DNS-relevant entities, deduplicated, and ingested automatically, as studies show that resolver policies achieve best accuracy with curated feeds [27,28,31]. Upstream resolvers such as Unbound improve privacy but do not enforce blocklists, while managed services like Quad9 or Cisco Umbrella trade local control for curated intelligence (as described in their official documentation). For SMEs, Pi-hole's open-source model, minimal resource demands, and automation capability make it a pragmatic enforcement engine, provided that policies prevent bypass via direct DNS or encrypted DoH (DNS over HTTPS)/DoT (DNS over TLS) [27,28,29,30,31]. In this work, Pi-hole's automation interfaces are extended to synchronise MISP threat feeds across both the DNS and proxy layers and to support retrospective hunting.

Pi-hole remains an ideal first-line control within the layered defence model developed in this thesis: technically efficient, simple to deploy, cost-free, and capable of delivering real-time, CTI-driven DNS protection across SME networks [27,28,29,30,31]. While

DNS filtering effectively blocks known malicious domains, its scope is limited to domain-level granularity. The following sections extend this foundation with proxy-based URL filtering and retrospective threat analysis to detect malicious activity hidden within legitimate services and to reveal earlier compromises.

2.4 Proxy-Based Threat Filtering

A proxy server acts as an intermediary between client devices and external web resources: clients send web requests to the proxy, which evaluates policy, inspects traffic where appropriate, forwards approved requests, and returns responses. In HTTP, the proxy receives full absolute-form URLs and can decide whether to allow or block a request before any fetch occurs. In HTTPS, clients typically establish a TLS tunnel using the CONNECT method, revealing only the destination authority and port. To view URL paths or payloads, the proxy must terminate TLS for the client and re-establish it toward the destination, a process that enables full Layer-7 inspection but extends the trust boundary and therefore requires strict certificate validation and governance [33,34,35,36]. In this thesis, building on Meiessaar's prototype [6], a second enforcement point is introduced that evaluates web requests at URL and content level rather than only by domain.

DNS-level filtering alone is inherently limited because it blocks entire domains but cannot detect malicious URLs or payloads residing under legitimate hosts. Modern attacks frequently exploit trusted domains, cloud services, and content-delivery platforms where only specific paths or files are harmful. A proxy closes this gap by applying hostname and URL policies during live web transactions: it can enforce host-based rules through Server Name Indication (SNI) without decryption, or, where justified, perform TLS interception to analyse paths, headers, and content. SNI-based filtering is lightweight but restricted to hostname visibility, whereas interception enables precise URL-level control at the cost of added complexity [33,34,37]. This enforcement layer is central to the proposed framework, translating CTI-derived domain and URL indicators from MISP into Squid access-control rules that block requests to known malicious URLs, including those hidden within otherwise legitimate domains, which DNS filtering alone cannot identify.

Open-source proxies align well with SME constraints. Squid offers caching, access-control lists, authentication, and optional HTTPS interception, with operational evidence

of scalability and acceptable performance on low-cost hardware [57]. Nginx provides an event-driven architecture well suited to lightweight forward-proxy use and efficient SNI-based blocking without decryption. Privoxy supports privacy-oriented and content-focused filtering and is often chained upstream for handling CONNECT requests. In practice, these components can co-exist with a DNS sinkhole such as Pi-hole on the same virtual machine or single-board computer, and multiple case studies confirm feasible performance in SME and campus environments [38]. In this research, such proxies form the enforcement tier above Pi-hole, enabling layered, CTI-driven blocking that extends protection into encrypted and application-level traffic in environments where the interception CA (Certificate Authority) is trusted. Squid was ultimately selected as the primary proxy in this work due to its mature feature set, integration flexibility, and strong community support, which together make it the most practical and automation-friendly choice for implementing CTI-driven policies in SME environments.

This proxy layer expands visibility beyond DNS, detecting URL-level threats, redirections, and post-resolution compromise indicators. Scripts synchronise CTI-derived URL indicators from MISP, normalise their format, and generate proxy-native rules to enforce intelligence-driven blocking on encrypted traffic, thereby covering DNS blind spots [34,37]. These same IoCs are also reused in retrospective hunting, where historical proxy logs are correlated with later intelligence to identify missed infections.

Open-source proxies were chosen over commercial Secure Web Gateway (SWG) or SASE (Secure Access Service Edge) solutions because they are cost-free, transparent, and locally managed, avoiding routine third-party data sharing and allowing auditable TLS handling, an important safeguard given repeated findings of weak validation and poor TLS hygiene in many interception appliances. Integrating a proxy with DNS filtering therefore provides SMEs with a second, deeper barrier that catches phishing, malware delivery, and other web-based threats that DNS alone cannot, while preserving the affordability and operational simplicity required in small-business environments [36,39].

2.5 Retrospective Hunting (Retrohunt)

Retrospective hunting (retrohunt) refers to the re-analysis of historical DNS and proxy telemetry using newly acquired CTI. Its purpose is to identify malicious domains or URLs

that were not detectable at the time of collection. For SMEs, this approach leverages data they already possess and compensates for the absence of continuous monitoring by correlating past network activity with updated indicators from MISP or other STIX-formatted feeds. The results can be visualised on timelines that clarify the sequence and scope of each incident [17,40,41]. Within the framework of this thesis, retrohunt complements the real-time enforcement layers, DNS filtering and proxy-based URL blocking, by introducing a post-event detection capability that reveals missed or delayed compromises. This retrospective process effectively extends the visibility window for SMEs, turning existing logs into a low-cost forensic asset rather than requiring continuous monitoring infrastructure.

A practical workflow involves collecting and retaining resolver and proxy logs, normalising hostnames and URLs, and ensuring that timestamps are handled in a consistent format so that historical activity can be re-evaluated when threat intelligence changes [42]. CTI is ingested via MISP, and domains and URLs are deterministically matched against the normalised log fields. Hits in each dataset can then be reprocessed whenever new indicators become available. This process can be automated with lightweight Python scripts or scaled using SIEM/ ELK (Elasticsearch, Logstash, Kibana) pipelines; existing MISP-to-Elastic integrations and CTI-to-query workflows demonstrate that such retrospective matching methods are both practical and transferable to network telemetry [40,43]. In this research, the implementation follows the same general principle but prioritises SME feasibility: lightweight scripts replace full SIEM pipelines while maintaining indicator accuracy and repeatability.

From an operational standpoint, multi-week or multi-month log retention is recommended to capture delayed indicators and attacker dwell time. Large-scale studies confirm that retrospective processing over substantial DNS and proxy corpora is feasible [44]. False positives can be reduced by curating CTI feeds, maintaining allowlists and warninglists, and prioritising rare destinations or anomalous user-agent and beaconing patterns; such measures significantly improve triage quality [28,44]. Results should include first- and last-seen timestamps, affected clients, and correlated artefacts, with sightings submitted back to MISP to enhance collective intelligence over time [45]. This iterative enrichment loop not only refines local detection but also contributes validated

sightings back into community intelligence feeds, reinforcing the collaborative nature of MISP-based CTI.

For SMEs, the benefit is tangible: affordable forensic insight, evidence-based incident response, and improved situational awareness without the complexity or cost of enterprise-grade SIEMs. Retroactive DNS-centric analyses have shown that subtle or long-dwell attacks are often confirmed only through historical reprocessing, underscoring the importance of this capability [46,47]. In the proposed architecture, retrohunt transforms static DNS and proxy telemetry into actionable intelligence, adding a forensic and temporal dimension that broadens detection coverage and strengthens long-term resilience for SMEs. Together with DNS and proxy enforcement, retrohunt completes the layered defence model developed in this thesis, providing both preventive and retrospective visibility using only open-source tools.

2.6 Continuation of Mert Meiessaar's Work

This thesis builds directly on Mert Meiessaar's defended work "How to Protect Small Businesses Using Public Cyber Threat Intelligence", which demonstrated that integrating the MISP with Pi-hole can deliver affordable, automated DNS-based protection for SMEs[6]. His system proved technically feasible and effective for blocking known malicious domains in real time using public CTI feeds, offering a low-cost defence mechanism accessible even to organizations with minimal IT capacity. However, his study also recognized key limitations: DNS filtering alone cannot detect malicious URLs hidden within legitimate domains, lacks visibility into encrypted HTTPS traffic, and provides no retrospective capability once new indicators are published.

To address these gaps, the present research extends Meiessaar's validated MISP–Pi-hole framework with two additional layers: a proxy-based filtering module for URL-level inspection and a retrospective hunting mechanism for re-evaluating historical DNS and proxy logs using updated threat intelligence. These enhancements transform a single-layer DNS defence into a multi-layered, CTI-driven architecture that strengthens detection coverage and forensic insight while preserving the openness, simplicity, and low operational cost central to the original design.

2.7 Related work

Existing academic work on CTI has largely concentrated on ingestion, correlation and analyst support rather than on tightly integrated, automated operationalisation across concrete enforcement layers. Early systems ingest STIX/TAXII feeds or interact with MISP and place indicators alongside logs in centralised platforms such as Elasticsearch or big-data stacks, enabling correlation and visualisation but stopping short of automated network-level blocking. For example, Ravi Kumar and Chaudhary retrieve STIX indicators via TAXII and index them with DNS, proxy and firewall logs in ELK, with the stated goal of "blacklisting malicious traffic", yet their implementation is limited to centralised storage and dashboards, without indicator lifecycle management or concrete DNS/proxy rule generation [48]. At CERN (European Organization for Nuclear Research), Panero et al. integrate a local MISP instance into a Kafka/Spark pipeline to enrich and correlate large-scale logs with IoCs [49]. Similarly, Serketzis et al. normalise and filter IoCs from multiple sources, proposing a model in which CTI helps pre-select relevant audit logs for digital forensics [50], while Papanikolaou et al.'s CTI management platform for industrial environments ingests MISP feeds, aggregates and exports custom knowledge as STIX 2.x and generates SIGMA rules for SIEM backends [51]. GEIGER Project (EU SME Cybersecurity Initiative), targeting SMEs, also consumes MISP JSON, extracts attributes and applies rule-based scoring to produce SME-friendly recommendations and APIs [17]. Across these works, CTI operationalisation remains SIEM- and analyst-centric: CTI is used to drive alerts, visualisations or rule generation for generic event correlation, but not to drive concrete, synchronised changes in DNS resolvers or web proxies.

A second line of work demonstrates end-to-end automation from external threat feeds to enforcement but restricts itself to IDS and generic firewall controls. IDS+OSINT automatically collects information from 49 OSINT feeds and transforms the resulting IoCs into IDS rules and blacklists that are deployed on a network IDS, detecting various malicious activities in real time [52]. Paulins likewise uses IntelMQ and OSINT (including MISP and OTX- Open Threat Exchange) to generate Suricata rules and IP blacklists that are pushed to a Palo Alto firewall, with alerts stored and visualised in ELK [53]. UNI-CERT, designed for the educational sector, automates the reception and sharing of IoCs in STIX format and claims to "update network configuration dynamically

to block traffic to malicious hosts" [54], while self-healing architectures based on STIX data propose automatic configuration corrections on security appliances [55] and are later integrated into CTIMP (Cyber Threat Intelligence Management Platform) as SSH-based (Secure Shell) remedial actions [51]. Although these systems clearly move beyond pure analysis to actual CTI-driven enforcement, their targets are IDS signatures, IP-based firewall rules or abstract "security appliances". None of them models the specific requirements of DNS or HTTP(S) proxy enforcement: there is no automated CTI → DNS blocklist (e.g. Pi-hole or RPZ) translation, no handling of domain versus URL indicators in resolver policy, and no generation of proxy-side ACLs (Access Control List), URL-path rules or SSL-bump decisions from CTI.

With regard to retrospective hunting, several authors recognise that CTI should inform post-hoc analysis of historical logs. Ravi Kumar and Chaudhary store both CTI and logs from proxies, DNS servers and firewalls in Elasticsearch, which conceptually permits querying past traffic for IoC matches [48]. Serketzis et al.'s threat-intelligence-informed digital forensics readiness model explicitly uses normalised IoCs to triage historical audit logs and surface events of interest to investigators [50]. CTIMP generates SIGMA rules from CTI that can be applied to stored logs in a SIEM [51], and the CERN big-data pipeline can in principle reprocess historical data with updated IoC sets using Spark [49]. Yet in all these cases, retrospective analysis is generic and only loosely coupled to CTI lifecycle. There is no dedicated mechanism for replaying new CTI over specifically DNS query logs and decrypted proxy URL logs, no explicit host and path canonicalization aligned with the enforcement path, and no automated creation of structured retrospective detection events back into a CTI platform such as MISP.

Finally, while some systems are explicitly tailored to constrained environments such as SMEs or specific sectors, they still fall short of providing an end-to-end, open-source enforcement pipeline. GEIGER and its successor focus on integrating public CTI into SME monitoring to reduce false positives and provide understandable guidance, but they deliberately leave the choice and configuration of downstream controls to the SME's existing tools [17,59]. CTIMP and UNI-CERT likewise rely on open-source components and automate significant parts of CTI handling and rule generation, yet neither delivers a packaged combination of CTI ingestion, DNS resolver enforcement, HTTPS proxy

interception and retrospective hunting that can be deployed with minimal manual tuning in small organisations [51,54].

Against this backdrop, the present thesis advances the state of the art in several tightly coupled ways. First, it operationalises CTI from MISP end-to-end: indicators are automatically ingested, updated, filtered using MISP warninglists, normalised and deduplicated, and then directly translated into enforcement artefacts rather than merely into analytical rules or dashboards. Second, it realises a dual enforcement stack in which the same CTI feed drives both a DNS enforcement layer (Pi-hole) and an HTTP(S) proxy enforcement layer (Squid). On the DNS side, domain IoCs are automatically converted into Pi-hole blocklists to achieve real-time blocking, with all queries logged for later correlation. On the proxy side, the system performs full HTTPS interception (SSL-bump), inspects decrypted URLs, and enforces CTI-derived policies at the URL-path level through automatically generated Squid ACLs, thereby enabling real-time blocking of malicious HTTP and HTTPS URLs. Crucially, domain and URL indicators are handled consistently across these layers through host and path canonicalization and a shared internal representation, ensuring fully synchronised multi-layer enforcement rather than ad hoc, layer-specific lists.

Third, the thesis introduces a CTI-driven retrohunt capability specialised for DNS and proxy telemetry. Historical DNS logs and decrypted proxy URL logs are periodically re-analysed against newly ingested indicators, using the same canonicalization and matching semantics as the live enforcement path. Matches produce structured retrospective detection events that are automatically written back into MISP, closing the loop between CTI, enforcement and historical analysis. Finally, all components—from CTI ingestion and processing to DNS/proxy rule generation, SSL-bump configuration and retrohunt—are orchestrated by Python scripts without manual rule editing, and built entirely from open-source tools. This yields a low-cost, low-complexity pipeline that is explicitly designed for SMEs yet embodies capabilities—synchronised DNS and proxy enforcement driven by MISP CTI, plus DNS/proxy-level retrohunt—that are not present in existing academic systems.

3 Research Methodology

This chapter presents the methodological approach used to design, construct, and evaluate the multi-layer CTI-driven defence system developed in this thesis. Because Chapter 4 provides the technical implementation details, the focus here is on the research logic, the sequence of activities, and the design-science reasoning that guided the creation and validation of the prototype.

3.1 Methodological Approach

The study follows a combined Design Science Research (DSR) and experimental evaluation approach. DSR is well suited for research that develops innovative IT artifacts to address real organisational problems. Following the structure proposed by Hevner [58], the research identifies a relevant SME cybersecurity problem, develops an artifact in the form of a multi-layer CTI operationalisation pipeline, demonstrates its functionality, and evaluates its utility through controlled experiments and test-group feedback. This perspective provides a structured foundation for motivating the artifact, justifying its design, and analysing its practical contribution.

Complementing DSR, the experimental methodology focuses on producing controlled, repeatable conditions for assessing enforcement correctness, retrospective detection capability, automation reliability, and SME-level feasibility. The methodological process consists of four steps, illustrated in Figure 1.

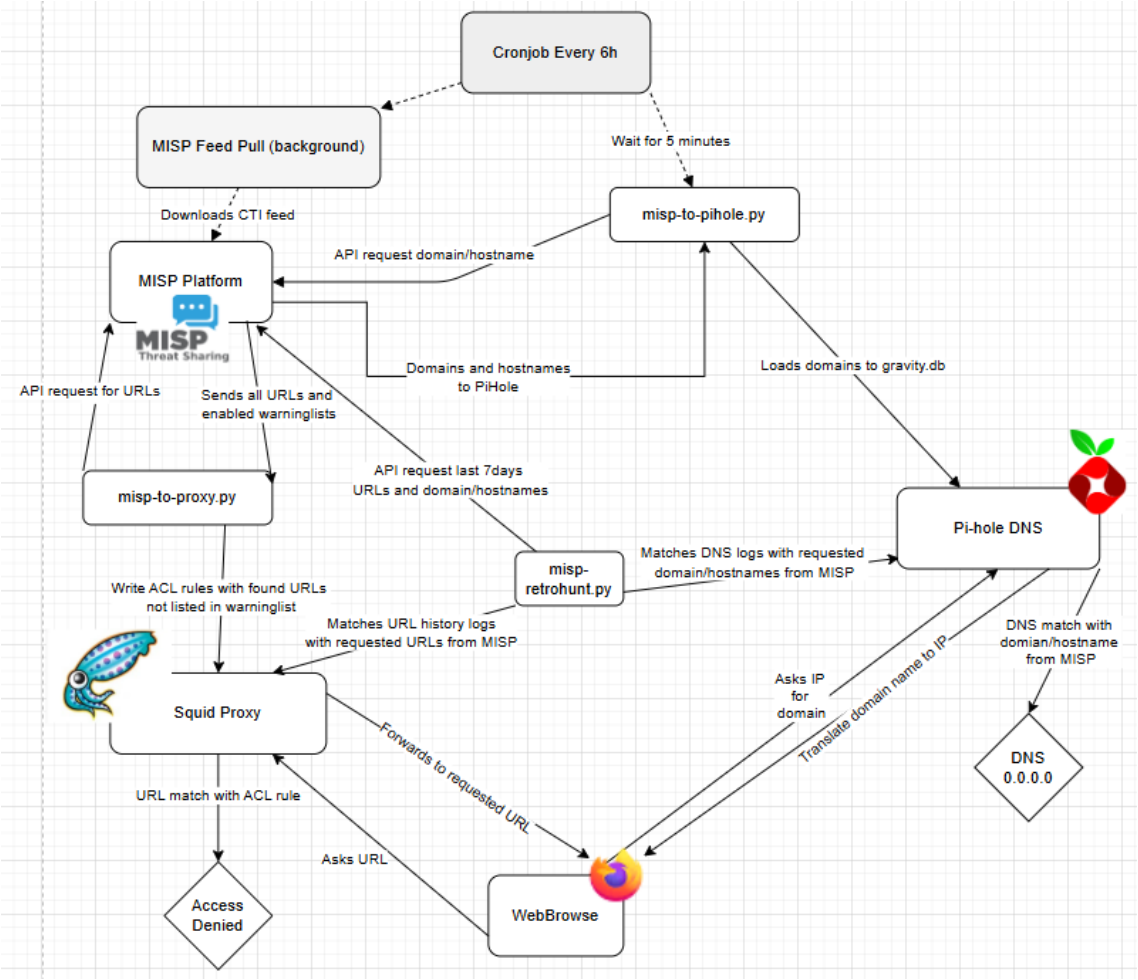

Figure 1. Prototype design

3.2 System Architecture Design

The first step involved defining an architecture capable of operationalising cyber threat intelligence across three enforcement and visibility layers: DNS filtering, proxy-based URL enforcement, and retrospective analysis. This stage focused on conceptual data flows, selecting appropriate open-source components, and determining how IoCs move from MISP to enforcement points and ultimately into retrohunt logic. Hardware and software choices were guided by SME constraints—lightweight deployment, low cost, transparency, and minimal administrative overhead. Technical configuration details are covered in Chapter 4; here the emphasis is on the architectural relationships and rationale.

3.3 CTI Operationalisation Across Layers

The second step established how CTI would be transformed into actionable policies at each layer. This required defining which indicator types are relevant (domains and URLs), how they should be normalised, and how warninglists should be applied to reduce false positives. The methodology clarified the logical mapping from indicators to DNS blocklists, proxy ACL rules, and retrohunt match criteria, ensuring that all three layers operate on a consistent intelligence set. This step describes the conceptual transformation rather than specific script logic, which appears in Chapter 4.

3.4 Controlled Traffic Generation and Log Collection

To evaluate enforcement correctness and retrospective detection, controlled experiments were designed to generate reproducible DNS and HTTP(S) traffic. This included interactions with both benign and malicious domains and URLs, enabling observation of blocking behaviour and false-positive handling. All evaluation data was captured using Pi-hole's DNS logs and Squid's proxy logs to reflect realistic SME constraints, avoiding packet-capture tools or SIEM platforms. The goal of this step was to produce datasets suitable for assessing enforcement accuracy and validating the retrohunt workflow.

3.5 Evaluation and Analysis

The final methodological step involved evaluating the system according to the research questions. This included assessing:

- SRQ1: DNS and proxy enforcement accuracy
- SRQ2: The effectiveness of retrospective matching using updated CTI
- SRQ3: The reliability and sufficiency of automation
- SRQ4: Deployment practicality and resource requirements in SME-like settings

Evaluation drew on functional tests, repeated runs of the enforcement pipeline, retrospective matching outputs, and qualitative feedback from a small test group. This step focuses on how evaluation decisions were made, while the specific results and observations are presented in Chapter 5.

4 Implementation

This chapter describes the practical implementation of the multi-layer CTI-driven defence system developed in this thesis. Whereas Chapter 3 outlined the conceptual research methodology, the present chapter explains how each architectural component was deployed and integrated on a single Ubuntu 24.04 virtual machine (using Windows 11, VMware 17.6.0, CPU type i7-1280p, VM settings- 2 cores + 4GB RAM + 30GB disc), reflecting the lightweight and resource-constrained environments typical of SMEs. Scripts and additional text can be found in Github repository, reference number 56.

4.1 Environment Setup

This section outlines how the test environment was prepared and configured to support controlled, repeatable experimentation. All components are installed on virtual machine to ensure isolation, consistency and ease of reversion during iterative testing.

4.1.1 Virtual Machine Configuration

All system components, including MISP, Pi-hole, Squid, and the automation scripts, were installed on a single Ubuntu virtual machine. Running all services locally simplified the prototype setup, reduced hardware requirements, and enabled a controlled and repeatable environment for testing and evaluation. This configuration also mirrors realistic SME deployments where a single server or virtual machine often hosts multiple security components.

4.1.2 Operating System Preparation

The virtual machine was prepared using standard Ubuntu package repositories. Dependencies for MISP, Pi-hole, and Squid were installed using official installation scripts. Default configurations were retained unless modifications were required for CTI ingestion, DNS enforcement, proxy filtering, or TLS interception.

4.1.3 Component Installation Workflow

Installation was orchestrated via a unified setup script that deployed MISP, Pi-hole, and Squid sequentially. After installation, the MISP administrator account was initialised, and

a dedicated API key was created for use by the automation scripts. Pi-hole was configured with DNS logging enabled, while Squid was enabled with SSL-bump for HTTPS inspection within this controlled environment. Running all components on one VM (Virtual Machine) ensured consistent testing conditions and reduced configuration complexity.

4.1.4 Overview of the pipeline scripts

The following table summarises the three automation scripts developed for this thesis. Each script implements a distinct stage of the CTI-driven defence pipeline, covering installation, DNS enforcement, proxy enforcement, and retrospective analysis. Together, they form the core operational workflow of the prototype. Shown in Table 1.

Table 1. Overview of the pipeline scripts.

Script Name	Description	Related Services
install.sh	End-to-end installer: sets up MISP (via official INSTALL.ubuntu2404.sh), installs & configures Pi-hole (web on port 8080, DNS logging), installs Squid with SSL-bump CA, MISP URL/domain list files, and Firefox CA policy. At the end it prints URLs and credentials/info for MISP, Pi-hole and the proxy.	MISP, Pi-hole (FTL/dnsmasq), Squid, system CA store, Firefox CA
misp-to-pihole	Pulls IoCs from MISP (domains/hostnames from enabled feeds/events), applies optional warninglists to avoid FPs, and loads the resulting domains into Pi-hole's blocklists so Pi-hole will sinkhole those domains. Intended to be run periodically (e.g. via cron every 6h).	MISP API, Pi-hole (gravity.db)
misp-to-proxy.py	Pulls URL-type IoCs from MISP and converts them to regex patterns, then updates the Squid URL regex list used by the proxy ACL so that HTTP/HTTPS requests matching these IoCs are blocked at the web proxy. Designed to be rerun after new/updated MISP events.	MISP API, Squid (ACLs, config reload), (misp_blocked_url_regex.txt)
misp-retrohunt.py	Implements the "retrohunt" logic: reads MISP IoCs and searches historical Squid URL logs and Pi-hole DNS logs to find where those IoCs were seen in the past; then creates new MISP events summarising matches for proxy and DNS, so analysts see which hosts contacted malicious domains/URLs.	MISP API, Squid logs (url-only.log), Pi-hole logs (pihole.log)
test_pipeline.py	Test/validation harness for the thesis pipeline: orchestrates running the MISP→Pi-hole sync, MISP→Squid sync, and retrohunt, likely while measuring CPU/RAM/IO and verifying that	Orchestrates MISP, Pi-hole, Squid and OS metrics / subprocesses

Script Name	Description	Related Services
	blocking and retrohunt behave as expected end-to-end. (Inferred from repo description & naming.)	

4.2 DNS Enforcement Implementation

The DNS enforcement layer operationalises domain-based threat intelligence by synchronising MISP attributes with Pi-hole's internal blocklist. This functionality is implemented in the misp-to-pihole.py script [56], which runs entirely on the same virtual machine as MISP and Pi-hole. The script retrieves indicators, normalises them, applies warninglists, updates Pi-hole's gravity.db database, and triggers a DNS blocklist rebuild.

4.2.1 Retrieving Domain Indicators from MISP

The script queries the MISP REST API for attributes of type domain and hostname using the restSearch endpoint, retrieving only indicators marked with to_ids = True. This ensures that only actionable IoCs are considered for enforcement. MISP warninglists are fetched in advance and transformed into a local whitelist set. Each warninglist entry is parsed, and values resembling domain names are added to an ignore set used later to prevent known benign domains from being blocked.

4.2.2 Normalisation, Deduplication, and Warninglist Filtering

Each retrieved attribute value is normalised through trimming, lowercasing, and removal of trailing dots. Duplicate values are collapsed into a set. The script then filters out domains that appear in the warninglist or match a warninglist domain suffix. This avoids false positives such as widely used cloud platforms or known benign infrastructure. The output of this stage is a clean, deduplicated list of malicious domains suitable for enforcement.

4.2.3 Updating Pi-hole's Domain Blocklist

Pi-hole stores custom blocklist entries in the SQLite database gravity.db, specifically in the domain list table. Before inserting new domain indicators, the script removes all previously inserted MISP-derived entries by matching rows via the comment field

"Synced from MISP" and the configured domain list type. This prevents stale or outdated IoCs from persisting across synchronisation cycles.

After clearing old entries, the script inserts each domain using INSERT OR IGNORE to avoid duplicates. If Pi-hole's schema includes a group_id column, the entry is associated with the appropriate group; otherwise, a simplified schema is used. When all entries have been written, the database changes are committed.

4.2.4 Gravity Rebuild and Enforcement Behaviour

Once the database is updated, Pi-hole's blocklist is recompiled by invoking pihole -g, which refreshes gravity, re-evaluates all lists, and applies domain blocking rules immediately. This ensures that any DNS query to a malicious domain subsequently resolves to Pi-hole's sinkhole address, effectively preventing outbound connections at the DNS layer.

4.2.5 DNS Logging for Enforcement and Retrohunt

Pi-hole's query log records all DNS resolution attempts made within the virtual machine, including timestamps, queried domains, and whether queries were blocked or allowed. These logs serve two purposes: validating the correctness of DNS enforcement during testing, and providing input for the retrospective hunting module, which re-evaluates historical queries when CTI updates introduce new indicators.

4.3 Proxy Enforcement Implementation

The proxy enforcement layer operationalises URL-level threat intelligence by translating MISP URL indicators into Squid access-control rules. This layer provides visibility and blocking capabilities beyond DNS filtering by inspecting full URLs inside HTTPS sessions where the browser trusts the proxy's interception certificate. The entire proxy component runs on the same virtual machine as MISP and Pi-hole.

4.3.1 Retrieving URL Indicators from MISP

The misp-to-proxy.py script [56] retrieves MISP attributes of type URL via the REST API. Indicators are processed only if they are marked with to_ids = True, ensuring that only actionable IoCs are considered. Warninglists are applied to suppress known benign

URLs and to avoid enforcement of widely used domains or services that would otherwise produce excessive false positives.

The script outputs a cleaned list of malicious URLs that will be enforced through Squid's ACL mechanism.

4.3.2 URL Normalisation and Parsing

Each retrieved URL undergoes structural normalisation. The script extracts:

- hostname (used for host-level matching), and
- path component (used to block specific malicious locations within otherwise benign domains).

This separation allows the proxy layer to block malicious URLs even when the domain itself is legitimate, a capability DNS filtering cannot provide.

Malformed or incomplete URLs are ignored, ensuring that only syntactically valid entries are used to generate enforcement rules.

4.3.3 Generating Squid ACL Rules

The matching. Squid's access-control system supports matching on both hostnames and URL paths. The script generates two types of ACL entries:

1. Host-based ACLs for URLs whose entire domain should be blocked.

2. URL-path ACLs for cases where the domain is benign but a specific path is malicious (e.g., https://drive.google.com/malware/...).

The script writes these ACL definitions into Squid's configuration directory and appends a unified http_access deny rule referencing them. All rules are tagged with a consistent identifier (e.g., "Synced from MISP") to allow overwriting during subsequent synchronisation cycles.

No manual modification of Squid configuration files is required; the script generates and manages all CTI-derived rules.

4.3.4 Applying Enforcement Rules

After the ACL files are updated, Squid is reloaded, allowing new rules to take effect without interrupting the proxy service. Reloading preserves active sessions while ensuring that the updated CTI is immediately enforceable.

Because the browser on the VM is explicitly configured to use the proxy, all HTTP and HTTPS traffic flows through Squid, enabling complete enforcement coverage within the test environment.

4.3.5 HTTPS Inspection via SSL-Bump

In the controlled prototype, Squid is configured to perform SSL-Bump. The browser trusts a locally generated root certificate, allowing Squid to decrypt HTTPS traffic, inspect full URLs, and re-encrypt the connection. This enables enforcement of URL-path IoCs that would otherwise be hidden inside encrypted traffic.

This capability is limited to environments where clients explicitly trust the local certificate authority; no transparent TLS interception is implemented.

4.3.6 Proxy Logging

Squid logs all HTTP and HTTPS requests processed through the proxy. When SSL-Bump is active, log entries include:

- the full requested URL,
- the timestamp,
- the browser's IP (local VM),
- the allowed/denied status.

These logs serve two purposes:

1. Verification of enforcement correctness during DNS and proxy evaluation.

2. Input for the retrohunt module, which matches historical URL requests against newly added IoCs.

The logging behaviour allows deterministic evaluation, as all generated traffic originates from and is recorded within the same VM.

4.3.7 Summary of Proxy Enforcement Constraints

The proxy layer enforces URL-based IoCs with high accuracy but is dependent on explicit browser proxy settings and TLS trust configuration. Enforcement is restricted to exact hostname and path matching and does not include content inspection, behavioural detection, or correlation with DNS logs. These constraints align with the SME-driven lightweight design but limit applicability in larger or more complex environments.

4.4 Retrospective Hunting Implementation

The retrospective hunting module ("retrohunt") provides post-event visibility by re-analysing historical DNS and proxy logs using updated threat intelligence from MISP. Its goal is to identify earlier connections to domains or URLs that were not known to be malicious at the time of access. This component reuses the same indicator types and normalisation logic applied in real-time enforcement, ensuring consistent interpretation across all detection layers. The entire retrohunt process runs locally on the same virtual machine.

4.4.1 Log Collection and Parsing

The misp-retrohunt.py script [56] reads DNS logs from Pi-hole and proxy logs from Squid directly from their default file locations on the VM. Each log line is processed to extract:

- timestamp,
- queried domain (from Pi-hole), or
- full URL (from Squid, including HTTPS URLs revealed via SSL-Bump).

Timestamps are converted to a consistent format, and domain or URL strings are normalised using the same transformations applied during CTI ingestion. This ensures that comparisons between CTI-derived indicators and historical log entries behave reliably.

No packet capture or deep logging infrastructure is used; the script operates solely on lightweight text-based resolver and proxy logs, mirroring realistic SME capabilities.

4.4.2 Retrieving Updated CTI for Retrohunt

The script queries MISP for current indicators of type domain, hostname, and URL. Indicators are filtered for to_ids = True and passed through warninglist logic if enabled. This ensures retrohunt matches reflect the same indicator set used by the DNS and proxy enforcement layers.

Normalisation produces two indicator sets:

- domain IoCs – used for DNS log matching
- URL IoCs – used for proxy log matching

The script does not attempt to infer or derive additional indicators from events; it operates only on explicitly provided attributes.

4.4.3 Exact-Match Retrospective Analysis

Retrohunt performs exact string matching only:

- A DNS log entry is considered a hit if its queried domain matches an indicator domain exactly (or matches a subdomain of that indicator, depending on Pi-hole formatting).
- A proxy log entry is considered a hit if the full URL string matches the normalised indicator URL.

No fuzzy matching, behavioural reasoning, SNI extraction, DNS–HTTP correlation, or multi-log join logic is implemented. This conservative approach is intentional: it maintains transparency, reduces false positives, and remains aligned with SME resource constraints.

Despite its simplicity, exact match retrohunt successfully identifies connections to malicious indicators that predate CTI ingestion.

4.4.4 Result Output and MISP Event Creation

For each retrohunt execution, the script prints a summary of all matched historical entries found in DNS and proxy logs. To maintain traceability within the CTI ecosystem, the script automatically creates two new MISP events:

1. A DNS retrohunt event – containing matched domains with timestamps

2. A proxy retrohunt event – containing matched URLs with timestamps

Each event includes metadata noting the retrohunt execution time and the source logs used. This approach allows analysts to review historical detections directly in MISP and aligns with the platform's workflow for sightings and event provenance.

Event creation does not modify existing CTI; it simply records retrospective observations.

4.4.5 Retrohunt Scheduling and Log Retention

Retrohunt can be executed manually or scheduled periodically. Because all DNS and proxy logs are stored locally on the VM, the accuracy of retrohunt depends on the organisation's log retention period. The prototype environment-maintained logs for approximately two weeks, but SMEs could extend retention based on available storage. No centralised log management or SIEM infrastructure is required.

4.4.6 Constraints and Limitations

The retrohunt module is intentionally simple and carries several limitations. It performs exact string matching only, without correlation or pattern inference. It does not reconstruct sequences of requests or join DNS queries with subsequent HTTP connections. It depends on local Pi-hole and Squid logs only, meaning detections reflect activity within the single VM environment. It does not analyse or classify payloads, TLS features, or behavioural indicators.

Despite these constraints, the module provides meaningful retrospective visibility, enabling SMEs to identify missed connections using the same CTI used for real-time enforcement.

4.5 Automation and Scheduling

Automation in the prototype is designed to minimise administrative overhead while ensuring that DNS and proxy enforcement layers remain synchronised with the latest threat intelligence available in MISP. Because all components run on a single virtual machine, automation is handled through lightweight scheduling and script execution rather than distributed orchestration.

4.5.1 Automating MISP Feed Updates

MISP feed updates are triggered using its built-in command-line interface. A cron job periodically executes: “sudo -u www-data /var/www/MISP/app/Console/cake Server fetchFeed 1 all”.

This command retrieves all enabled feeds and updates the MISP database. Since MISP feed ingestion may take several minutes, the prototype includes a brief delay before updating enforcement layers to ensure that feed data has been fully processed. No continuous or event-driven feed polling is implemented; updates occur strictly according to the cron schedule.

4.5.2 Automated DNS and Proxy Enforcement Refresh

Automation needs. Once MISP has finished ingesting its feeds, the same cron job executes the domain and URL synchronisation scripts: misp-to-pihole.py and misp-to-proxy.py Each script retrieves updated indicators, applies warninglists, performs normalisation, and updates Pi-hole and Squid configurations accordingly.

The DNS script modifies Pi-hole’s SQLite database and triggers a gravity rebuild (pihole -g), while the proxy script regenerates Squid ACL files and reloads the proxy service. These steps ensure that enforcement rules remain aligned with the latest CTI without requiring administrator involvement.

The automation is periodic, not real-time: enforcement accuracy therefore depends on the frequency of scheduled updates (e.g., every six hours in the prototype).

4.5.3 Automated Retrohunt Execution

Retrospective analysis can also be scheduled, although it is not tied to feed updates by default. A nightly cron entry may execute the misp-retrohunt.py script, which:

1. Reads existing DNS and proxy logs from the VM,

2. Retrieves the latest indicators from MISP,

3. Performs exact matching, and

4. Generates new MISP events documenting historical matches.

Because the prototype uses only local logs, the retrohunt module does not require centralised log aggregation or SIEM infrastructure. The correctness of retrospective results depends primarily on the length of the log retention window configured in Pi-hole and Squid.

4.5.4 Automation Constraints

Automation in the prototype is intentionally simple to remain realistic for SME deployment. Key constraints include:

- No real-time enforcement updates—only scheduled synchronisation.
- No distributed orchestration across multiple hosts (single-VM only).
- Squid enforcement depends on the client's browser being manually configured to use the proxy.
- TLS inspection only functions when clients trust the VM's local certificate authority.
- Retrohunt accuracy depends entirely on available local logs.

Despite these limitations, the automation model is sufficient to demonstrate continuous, low-maintenance CTI operationalisation across DNS, proxy, and retrohunt layers.

4.6 Summary of Implementation Constraints

The prototype operates entirely on a single VM, which simplifies experimentation but does not reflect distributed production deployments. Enforcement depends on local browser configuration for proxy use, and SSL inspection requires trusting the VM-generated certificate. Retrohunt is limited to exact matches within logs stored on the same machine. These constraints align with SME resource limitations but restrict broader scalability.

5 Results

This chapter presents the results of evaluating the prototype system across DNS enforcement, proxy-based URL blocking, retrospective detection, automation behaviour, and SME usability. All experiments were conducted on a single virtual machine hosting MISP, Pi-hole, Squid, and the automation scripts, ensuring a controlled and reproducible environment.

5.1 Domain Enforcement via Pi-hole

The DNS layer demonstrated consistent enforcement of MISP-derived domain indicators. After synchronisation, Pi-hole correctly populated its gravity.db database with the normalised indicators and sinkholed all queries to malicious domains. Queries to malicious test domains returned Pi-hole's blocking response, while benign test domains resolved normally. Pi-hole's DNS logs showed clear block events with timestamps and decision flags, confirming that the synchronised blocklist was being applied as intended.

When the synchronisation script was run with warninglists enabled, known benign domains that commonly appear in CTI feeds were removed before insertion into Pi-hole. In this configuration, the test group did not observe any false positives, even when experimenting with widely used domains such as Google properties. In contrast, when warninglists were disabled, Pi-hole enforced all raw indicators from MISP, including domains that were not actually malicious in the test context. This led to unnecessary blocking and confusion for users interpreting the blocklists. These observations confirm that warninglist filtering is essential for keeping automated enforcement safe and interpretable in SME environments.

Overall, the DNS layer enforced CTI-derived domain indicators with high accuracy and predictable behaviour, supporting SRQ1 for domain-based enforcement.

5.2 URL Enforcement via Squid Proxy

The proxy layer enforced URL-path indicators with consistent behaviour inside the controlled environment. With the browser configured to trust the proxy's interception

certificate, Squid performed SSL-Bump on HTTPS traffic and logged full URLs, allowing precise path-based blocking of malicious locations hosted under otherwise legitimate domains.

In the test scenarios, requests to malicious URLs were immediately denied with a Squid-generated block page, while requests to benign URLs proceeded without interruption. URL-path indicators were correctly enforced even when the underlying domain was legitimate, such as when a synthetic malicious path was hosted on a widely used service. Squid's access logs contained full URLs and enforcement decisions, which matched expectations across repeated runs. No incorrect allow or deny decisions were observed in the test cases, indicating high enforcement correctness at the proxy layer and further supporting SRQ1.

5.3 Interaction Between DNS and Proxy Layers

The combined DNS and proxy enforcement layers provided complementary protection. The DNS layer blocked malicious domains at resolution time, preventing any subsequent HTTP(S) traffic to those hosts, whereas the proxy layer blocked specific URL paths in cases where only part of a domain was malicious. When a domain was present both as a domain IoC and as part of a URL IoC, Pi-hole blocked the DNS query before Squid was involved, which is expected given the order of operations. No inconsistent behaviours or conflicts were observed between the layers. This confirms that layered CTI operationalisation can extend visibility and control beyond DNS-only filtering and contributes to answering the primary research question regarding multi-layer effectiveness.

5.4 Retrospective Detection (Retrohunt)

The retrohunt module successfully identified earlier DNS and proxy requests that matched indicators introduced into MISP after the traffic occurred. Using exact string matching on normalised DNS queries and URLs, the script reliably rediscovered prior activity that would not have been blocked at the time of access because the corresponding IoCs were not yet present.

For DNS, retrohunt correctly flagged all historical queries to the synthetic malicious domains generated during the tests. The detections were reproducible across multiple runs, showing that the matching logic and log parsing behaved deterministically. For the proxy layer, retrohunt rediscovered all HTTP(S) requests to the malicious URLs included in the test traffic. Each match corresponded exactly to a previously logged URL, and no spurious hits were produced. In both cases, the script created dedicated MISP events summarising the retrospective findings, one event for DNS matches and one for proxy matches, making historical detections visible within the same CTI environment.

These results confirm that exact-match retrospective analysis is sufficient to reveal earlier visits to malicious domains and URLs in this context, thereby supporting SRQ2.

5.5 Automation Behaviour

Automation was evaluated by scheduling MISP feed updates and enforcement synchronisation scripts via cron. Periodic feed updates successfully pulled new CTI into MISP, after which the DNS and proxy synchronisation scripts refreshed Pi-hole and Squid configurations without manual intervention. Across several cycles, the system consistently reflected the latest indicators: newly added malicious domains and URLs were enforced after the next scheduled run, and removed or warninglist-matching indicators no longer appeared in the blocklists.

Retrohunt was also executed periodically in a scheduled mode. Each run scanned the locally retained DNS and proxy logs, matched them against the current indicator set, and generated updated retrohunt events in MISP. The correctness of these results depended primarily on the log retention window rather than the scheduling frequency. No missed or duplicate retrohunt events were observed within the test period.

These observations support SRQ3 by demonstrating that a meaningful level of automation can be achieved through simple cron-based scheduling and script-driven updates. While this does not provide real-time SOC functionality, it is sufficient for SMEs seeking low-maintenance CTI operationalisation.

5.6 Test Group Evaluation

A small user evaluation was conducted to assess usability and behaviour under conditions closer to real SME practice. Four participants, organised into two groups of two, deployed the system on their own Ubuntu machines using the thesis GitHub repository. Both groups followed the same installation guide but differed in one key configuration: one group ran the system with MISP warninglists enabled, and the other group disabled warninglists during synchronisation. Both groups used CTI feeds like CIRCL OSINT Feed and The Botvrij.eu Data.

The group that disabled warninglists experienced numerous false positives. Commonly used domains, including Google-related services, were blocked because they appeared as raw indicators in MISP but were not actually malicious in the test context. Participants reported confusion when reviewing blocklists and logs, as the system appeared overly aggressive and its decisions were harder to interpret.

In contrast, the group with warninglists enabled saw no false positives during their tests. Execution of the synchronisation scripts was perceived as slightly slower due to the additional filtering work, but the resulting blocklists were cleaner and easier to understand. Participants in this group reported that the system's behaviour seemed reasonable and that they could clearly distinguish intended malicious blocks from normal traffic. Across both groups, participants suggested improvements such as more explanatory status messages, clearer error reporting, and a simple dashboard or summary page showing which indicators were enforced and why.

These findings support SRQ4 by demonstrating that the system can be deployed and operated by typical IT staff in SME-like environments, while also highlighting the importance of warninglists and the potential benefits of further usability enhancements.

5.7 Test Script Validation

To further validate that the integrated prototype behaves as intended, a dedicated end-to-end test script (test_pipeline.py) [56] was executed to assess the entire CTI operationalisation chain in a controlled and reproducible manner. The script generates synthetic malicious and benign indicators, creates a corresponding MISP test event,

triggers the DNS and proxy synchronisation processes, and finally evaluates enforcement and retrospective detection. The output confirmed that the system translated all synthetic malicious domains and URLs into active Pi-hole and Squid enforcement rules, with malicious domains consistently sinkholed and malicious URLs blocked by the proxy. Benign indicators included in the warninglist were correctly excluded from enforcement, demonstrating that warninglists prevent false positives even when indicators resemble legitimate internal domains. The retrohunt stage rediscovered the synthetic malicious requests in both DNS and proxy logs and pushed the results into new MISP events, confirming the reliability of exact-match retrospective detection. The successful execution of the test script provides strong evidence that the prototype performs end-to-end CTI ingestion, enforcement, and retrospective analysis in a manner fully aligned with the research questions. Specifically, it demonstrates high enforcement accuracy (SRQ1), reliable retrospective visibility (SRQ2), correct operation of automated synchronisation steps (SRQ3), and practical deployability in SME-like environments (SRQ4). The test therefore confirms that the prototype functions as designed and that all core components operate cohesively when subjected to a synthetic but comprehensive evaluation pipeline.

5.8 Performance Test Script

To evaluate the scalability and operational limits of the proposed multi-layer threat enforcement prototype, a controlled load test was conducted using a custom performance testing script (sme_load_test.py) [56]. The test environment simulated an SME network by generating concurrent endpoint activity against the integrated Pi-hole (DNS filtering) and Squid proxy (HTTPS filtering with SSL interception). The objective was to assess both enforcement accuracy and system behaviour under increasing load. Command with parameters to run the sme_load_test.py: "./sme_load_test.py --devices [x] --requests 200 --cpu-load 20 --mem-mb 50 --monitor-interval 10 --bad-fraction 0.1". The value "x" was 15, 30, 60 or 75.

Each simulated endpoint generated 200 DNS queries and 200 HTTPS requests, resulting in an equal volume of DNS and HTTPS traffic. A fixed fraction of the generated traffic (bad-fraction = 0.1) intentionally targeted known malicious Indicators of Compromise (IoCs) sourced from MISP, while the remaining traffic represented benign requests. To approximate realistic endpoint behaviour, each simulated client additionally imposed

synthetic system pressure of approximately 20% CPU utilisation and 50 MB RAM consumption. System performance metrics were sampled at 10-second intervals throughout each test run.

Four test scenarios were executed with 15, 30, 60, and 75 concurrent endpoints. Across all scenarios, the system demonstrated 100% enforcement accuracy for malicious IoCs at both the DNS and HTTPS layers. All malicious DNS queries were blocked by Pi-hole (e.g., 307/307, 642/642, 1227/1227, and 1444/1444), and all malicious HTTPS requests were denied by the Squid proxy under SSL-bump, confirming correct propagation of MISP-derived indicators into both enforcement mechanisms. No false negatives were observed for IoC-based traffic in any test scenario.

As endpoint count increased, performance degradation followed a predictable and linear pattern. Average DNS resolution latency increased from 72.6 ms (p95: 186.7 ms) at 15 endpoints to 865.7 ms (p95: 2465.9 ms) at 75 endpoints. Similarly, HTTPS request latency increased from an average of 319.0 ms (p95: 530.7 ms) to 1817.3 ms (p95: 4620.4 ms). Throughput remained relatively stable up to 60 endpoints (approximately 24.5–30.4 requests per second) but declined at 75 endpoints (20.4 requests per second), indicating that the system approached saturation under the combined effects of concurrency and synthetic resource pressure.

An important observation concerns the reliability of benign HTTPS traffic under higher load. While IoC enforcement remained fully effective, the number of blocked “good” HTTPS requests increased with endpoint count (52, 175, 394, and 369 respectively). These blocks were not caused by false IoC matches but rather by proxy-level failures such as timeouts, TLS handshake errors, or resource exhaustion under contention. Such responses were recorded by the test logic as blocked requests, highlighting a distinction between security correctness (which remained intact) and service quality under stress.

Table 2. Load test for DNS and HTTPS

Metric ↓ \ Devices →	**15**	**30**	**60**	**75**
Requests per device	200	200	200	200

Metric ↓ \ Devices →	15	30	60	75
Total requests (DNS+HTTP)	3000	6000	12000	15000
Elapsed time (s)	122.55	206.65	395.15	734.59
Throughput (qps / rps)	24.48	29.03	30.37	20.42
DNS avg latency (ms)	72.56	230.31	598.26	865.66
DNS p95 latency (ms)	186.68	574.36	1548.84	2465.88
DNS blocked (total, %)	307 (10.23%)	642 (10.70%)	1227 (10.22%)	1444 (9.63%)
DNS bad blocked	307 / 307	642 / 642	1227 / 1227	1444 / 1444
DNS good blocked	0 / 2693	0 / 5358	0 / 10773	0 / 13556
HTTPS avg latency (ms)	318.96	480.84	838.52	1817.29
HTTPS p95 latency (ms)	530.74	908.12	1861.52	4620.39
HTTPS blocked (total, %)	359 (11.97%)	817 (13.62%)	1,621 (13.51%)	1,813 (12.09%)
HTTPS bad blocked	307 / 307	642 / 642	1227 / 1227	1444 / 1444
HTTPS good blocked	52 / 2693	175 / 5358	394 / 10773	369 / 13556

The results indicate that the prototype delivers reliable and deterministic enforcement of malicious indicators across both DNS and HTTPS layers, even under elevated load. However, latency growth and reduced reliability for legitimate HTTPS traffic become noticeable beyond approximately 60 concurrent endpoints in the tested configuration. For

SME environments, this suggests that the proposed architecture is well-suited for small to medium-sized deployments, while larger installations would benefit from increased system resources, service separation, or horizontal scaling of proxy and DNS components.

5.9 Summary of Results

The evaluation shows that the prototype successfully operationalises MISP-based threat intelligence across DNS, proxy, and retrospective layers in a single-VM, SME-oriented environment. DNS filtering accurately enforces domain indicators and, when combined with warninglists, can operate without observable false positives in the test scenarios. Proxy-based URL enforcement extends protection to malicious paths embedded within legitimate domains, and SSL-Bump enables this enforcement even for HTTPS traffic when the client trusts the local certificate authority. The retrohunt module reliably identifies earlier requests to malicious domains and URLs using exact matching, demonstrating that historical DNS and proxy logs can be repurposed as a basic forensic asset without SIEM infrastructure. Automation via cron keeps the system in sync with updated CTI, and a small test group confirmed that the prototype is deployable and understandable by non-cybersecurity staff.

In addition to these empirical results, the automated test script further confirmed that the prototype performs end-to-end CTI ingestion, enforcement, and retrospective detection exactly as designed. Its successful execution provides an objective verification that all layers of the system operate cohesively and reinforces the validity of the findings presented in this chapter.

Controlled SME-scale load testing further demonstrated that the prototype maintains 100% enforcement accuracy for malicious indicators at both the DNS and HTTPS layers under increasing concurrency. While performance degradation manifested as increased latency and reduced reliability for benign HTTPS traffic at higher endpoint counts, malicious IoC detection and blocking remained unaffected across all tested scenarios. These results indicate that the system provides reliable protection within typical SME sizing (approximately 15–60 concurrent endpoints under constrained resources), while also highlighting predictable scalability limits that can be addressed through additional resources or component separation.

Collectively, these results support the primary research question and all four secondary research questions, indicating that a layered, open-source CTI operationalisation approach can provide SMEs with practical improvements in visibility, detection accuracy, and incident awareness without imposing enterprise-level complexity or cost.

6 Discussion

This chapter interprets the results presented in Chapter 5 in relation to the research aim, hypothesis, and research questions established in Chapter 1. Whereas the previous chapter focused on empirical observations of the prototype's behaviour, the discussion evaluates what these findings mean for the feasibility, effectiveness, and practical value of multi-layer CTI operationalisation in SME contexts. The analysis considers how well the system fulfils its intended purpose, how the individual components contribute to overall defensive capability, and how the outcomes align with expectations from prior research and the design-science methodology. With this foundation, the following sections validate the hypothesis, answer the research questions, and assess the strengths, limitations, and broader implications of the proposed framework.

6.1 Validation of the Hypothesis

The organisations. The results of this study support the hypothesis that extending the existing MISP–Pi-hole model with proxy-based URL enforcement and retrospective log analysis increases detection coverage for malicious domains and URLs, improves the timeliness of incident discovery through retrospective identification, and remains affordable and usable for SMEs. Across all tests, the three layers behaved deterministically and fulfilled their intended functions: Pi-hole reliably sinkholed malicious domains, Squid enforced URL-path IoCs through HTTPS interception, and the retrohunt module consistently identified earlier DNS and proxy requests corresponding to indicators added to MISP after the activity occurred. Although the study did not measure mean time to discovery numerically, retrospective analysis clearly reduced the delay between compromise and detection by revealing missed earlier requests once new CTI became available.

These outcomes directly address the primary research question by demonstrating that cyber threat intelligence from MISP can be effectively operationalised across DNS filtering, proxy-based URL enforcement, and retrospective log analysis in a lightweight, open-source framework suitable for SMEs. The automated test script further strengthened this conclusion by confirming that the entire pipeline—from indicator creation, to enforcement, to retrohunt event generation—operates correctly under scripted, end-to-

end validation. Overall, the evidence affirms that the prototype meets the hypothesised goals of broader detection coverage, improved detection timeliness via retrospective analysis, and SME-feasible usability.

6.2 Answering the Research Questions

The following subsections interpret the results in direct relation to the primary and secondary research questions introduced in Chapter 1. Each question is addressed using evidence drawn from the enforcement tests, retrohunt analysis, automation behaviour, and user evaluation presented in Chapter 5.

6.2.1 Primary Research Question- RQ

How effectively can cyber threat intelligence from MISP be operationalised across DNS filtering, proxy-based URL enforcement, and retrospective log analysis in a lightweight, open-source framework suitable for SMEs?

The prototype operationalised MISP-derived indicators effectively across all three layers. Indicators were automatically retrieved, normalised, and translated into enforcement rules with consistent behaviour across repeated runs. DNS filtering, proxy-based URL blocking, and retrospective matching each functioned reliably and provided complementary visibility. The system operated entirely on a single VM using only open-source components, demonstrating feasibility for resource-constrained SMEs.

6.2.2 SRQ1: How accurately do the DNS and proxy layers enforce domain and URL indicators derived from MISP?

Testing showed that DNS and proxy enforcement operated with high accuracy. Pi-hole sinkholed all malicious domains, and Squid denied access to all malicious URLs, including those embedded within legitimate domains. Warninglists played a critical role by preventing benign indicators from being misinterpreted as threats; the group with warninglists enabled observed no false positives. Enforcement mismatches or inconsistent behaviour were not observed, confirming precise and deterministic indicator enforcement at both layers.

6.2.3 SRQ2: To what extent can retrospective matching of DNS and proxy logs reveal earlier visits to malicious domains or URLs using updated CTI?

The retrohunt module successfully identified all historical DNS queries and proxy requests associated with malicious indicators introduced after the traffic occurred. Exact matching proved sufficient for the structured nature of DNS and URL data used in this study. The module produced consistent results across repeated runs and generated dedicated MISP events documenting historical matches, enabling analysts to recognise earlier exposures without additional tools. These findings confirm that retrospective matching enhances temporal visibility and improves incident discovery timeliness for SMEs.

6.2.4 SRQ3: What level of automation can be achieved to ensure continuous CTI ingestion, enforcement, and retrospective analysis with minimal administrative effort?

The system achieved a practical level of automation through cron-based scheduling and Python scripts. Feed updates, DNS and proxy synchronisation, and retrohunt execution required no manual intervention once configured. While this does not constitute real-time or event-driven automation, it is appropriate for SMEs and significantly reduces operational overhead. The test script further validated the reliability of automated execution, demonstrating that the entire CTI pipeline can run cohesively in an unattended manner.

6.2.5 SRQ4: What are the resource, configuration, and operational requirements for deploying the integrated system in SME-like environments?

User testing showed that the system runs smoothly on modest hardware, such as a single virtual machine, and can be installed by IT staff without specialist security expertise. The warninglist-enabled configuration produced accurate results without false positives, but required slightly longer synchronisation due to additional filtering. TLS interception required trusting a local certificate authority, which participants found manageable in the test environment. Overall, the deployment complexity and resource requirements were consistent with SME capabilities, supporting feasibility in real-world small-organisation contexts.

6.3 Strengths and Weaknesses of the Proposed System

The framework's primary strength lies in its layered architecture, where DNS filtering, URL-path enforcement, and retrospective analysis jointly enhance detection coverage. All layers rely on open-source technologies, keeping operational costs low and ensuring transparency. Enforcement accuracy was high across all tests, automation reduced manual workload, and retrohunt added valuable forensic capability without requiring SIEM infrastructure.

However, the system remains a prototype. It does not include transparent proxying or automated certificate distribution, requiring manual browser configuration. TLS interception was validated only on a single VM with explicit certificate trust. Retrohunt relies on exact matching and does not perform cross-log correlation or behaviour-level analysis. Scalability under long-term or large-scale MISP deployments was not tested. These limitations do not diminish the demonstrated feasibility but highlight areas for improvement.

6.4 Ethical, Legal, and Operational Implications of HTTPS Interception

The use of HTTPS interception through SSL-Bump introduces important ethical, legal, and operational considerations that must be carefully addressed in SME environments. By requiring client devices to trust a locally issued certificate authority, the organisation gains visibility into employee web traffic, which alters the traditional end-to-end privacy model of HTTPS communication and therefore requires transparent communication, clear acceptable-use policies, and a legitimate security justification. From an operational perspective, HTTPS interception may negatively affect access to certain services, particularly security-sensitive platforms such as online banking, healthcare portals, and Estonian government e-services (e.g., X-Road–connected systems), which often implement certificate pinning or strict trust validation and may refuse proxied connections. In such cases, selective bypass rules or domain-based exclusions are necessary to maintain service availability while preserving security controls elsewhere. Legally, SMEs operating within the European Union must also ensure that HTTPS inspection practices align with GDPR principles, including data minimisation, purpose

limitation, and proportionality. Consequently, while HTTPS interception can significantly enhance threat detection and URL-level enforcement, it should be applied selectively and governed by clear technical, organisational, and legal controls rather than enabled indiscriminately.

6.5 Implications for SMEs

The results show that SMEs can meaningfully enhance their defensive posture using the proposed framework. DNS filtering provides broad protection at low cost, proxy URL enforcement detects threats that would bypass DNS-only controls, and retrohunt allows SMEs to uncover earlier exposures using data they already possess. Together, these capabilities provide a multi-layered defensive model that brings elements of enterprise-grade visibility to small organisations without requiring expensive appliances or commercial threat platforms.

6.6 Considerations for Larger Networks

While components such as CTI ingestion logic and ACL generation may scale conceptually, the system as implemented is not ready for medium or large enterprise networks. Transparent proxying, certificate deployment across multiple devices, load balancing, large-scale log retention, and advanced correlation techniques would be required. Thus, the system is best viewed as an SME-oriented solution with the potential to inform future, more scalable architectural designs.

6.7 Alignment with Prior Research (including Meiessaar 2025)

This research extends Meiessaar's CTI-driven DNS filtering framework by adding URL-path enforcement and retrospective analysis, dimensions not previously operationalised within SME-focused studies. The results align with literature advocating for layered defences and CTI automation while filling a gap in demonstrating that DNS, proxy, and retrospective workflows can be integrated under a unified CTI pipeline. By validating operational feasibility, this work advances prior research beyond conceptual discussions of CTI use in SMEs.

6.8 Novelty Assessment and Contribution

The key contribution of this thesis is a fully open-source, multi-layer CTI operationalisation pipeline tailored for SMEs. While DNS filtering, proxy enforcement, and MISP-based CTI ingestion are not individually novel, their coordinated, automated integration, and empirical validation in an SME-feasible environment, has not been previously documented. The prototype demonstrates that meaningful defence-in-depth and retrospective visibility can be achieved without enterprise infrastructure, providing a replicable model for small organisations.

6.9 Future Work

Future enhancements should focus on improving operational robustness, policy control, and scalability while preserving the lightweight nature of the proposed solution. HTTPS inspection could be refined by introducing explicit bypass mechanisms for security-sensitive services such as online banking platforms, healthcare systems, and government e-services, where interception may be technically incompatible or legally inappropriate. Scalability should be investigated primarily from a hardware and architectural perspective, examining the impact of increased CPU, memory, and service separation on latency and throughput under higher endpoint counts. Finally, usability and external validity could be strengthened through dashboards visualising enforcement and retrohunt results, and through longer-term testing in real SME environments.

7 Conclusion

This thesis demonstrated that cyber threat intelligence from MISP can be effectively operationalised across DNS filtering, proxy-based URL enforcement, and retrospective log analysis in a lightweight, fully open-source framework suitable for small and medium-sized enterprises (SMEs). The integrated prototype showed that domain and URL indicators can be automatically transformed into enforcement rules for Pi-hole and Squid with high accuracy and minimal administrative effort. DNS filtering consistently sinkholed malicious domains, HTTPS interception enabled precise URL-level blocking for threats embedded within legitimate domains when clients explicitly trusted the local certificate authority, and the retrospective hunting module successfully identified earlier DNS and proxy requests corresponding to indicators added to MISP after the activity occurred. Together, these capabilities validate the hypothesis that extending a MISP–Pi-hole pipeline with proxy-based enforcement and retrohunt components increases detection coverage and improves post-event threat visibility in SME environments.

The evaluation results, supported by controlled experiments and automated test scripts, confirmed the functional correctness of all three enforcement layers and their ability to operate cohesively as a single system. The use of warninglists eliminated observable false positives in DNS enforcement within the test scenarios, resulting in predictable and transparent blocking behaviour for SME operators. Automation via cron enabled continuous CTI ingestion, enforcement updates, and retrospective analysis without manual intervention, demonstrating an achievable level of operational simplicity aligned with the constraints identified in the research questions. A small external test group further confirmed that the system can be deployed and operated by IT staff without specialist security training, reinforcing its practical applicability in SME-like environments.

From a performance perspective, the prototype maintained deterministic enforcement of malicious indicators across DNS and HTTPS layers under increasing load, while exhibiting predictable latency growth and reduced reliability for benign HTTPS traffic at higher endpoint counts. These findings indicate that the solution is well suited for typical SME sizes on modest hardware, while also highlighting clear scalability boundaries that

should be addressed through additional resources or architectural separation for larger deployments.

This research contributes a replicable and cost-efficient approach for SMEs seeking to improve cyber resilience through layered CTI-driven defences. By leveraging community threat intelligence and open-source tools, the system provides preventive and retrospective visibility that is commonly associated with enterprise-grade solutions, yet achievable without enterprise-level complexity or cost. Although the prototype does not implement automated certificate distribution, transparent proxying, or long-term production-scale performance validation, it provides clear evidence that multi-layer CTI operationalisation is both technically feasible and operationally valuable for small organisations. Overall, the study demonstrates that SMEs can meaningfully expand detection capability and reduce reliance on commercial security products by adopting a layered, open-source CTI-driven framework.

8 References

[1] U.S. Small Business Administration, “Cyber Safety Tips for Small Business Owners,” SBA.gov, Sep. 2023. Accessed: Oct. 11, 2025. [Online]. Available: https://www.sba.gov/blog/2023/2023-09/cyber-safety-tips-small-business-owners

[2] European Union Agency for Cybersecurity (ENISA), ENISA Threat Landscape 2024. ENISA, 2024. Accessed: Oct. 11, 2025. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2024

[3] European Union Agency for Cybersecurity (ENISA), ENISA Threat Landscape 2025. ENISA, 2025. Accessed: Oct. 11, 2025. [Online]. Available: https://www.enisa.europa.eu/publications/enisa-threat-landscape-2025

[4] Proofpoint, “Human Factor Vol. 2 offers new insights on phishing,” Proofpoint Blog, 2025. Accessed: Oct. 11, 2025. [Online]. Available: https://www.proofpoint.com/us/blog/email-and-cloud-threats/human-factor-vol-2-offers-new-insights-phishing

[5] L. Turner, “Malicious URLs overtake email attachments as the biggest malware threat,” IT Pro, Mar. 2025. Accessed: Oct. 11, 2025. [Online]. Available: https://www.itpro.com/security/cyber-attacks/malicious-urls-overtake-email-attachments-as-the-biggest-malware-threat

[6] M. Meiessaar, How to Protect Small Businesses Using Public Cyber Threat Intelligence. Master’s thesis, Tallinn University of Technology, Tallinn, Estonia, 2025. Accessed: Oct. 11, 2025. [Online]. Available: https://digikogu.taltech.ee/et/Item/ba6e004c-9aae-4eef-ad13-cf5129c98ea5

[7] Verizon, 2025 Data Breach Investigations Report (DBIR). Verizon Enterprise Solutions, 2025. Accessed: Oct. 11, 2025. [Online]. Available: https://www.verizon.com/business/resources/reports/dbir/

[8] MITRE Corporation, Health Delivery Organizations and Ransomware: A White Paper. MITRE, Nov. 2021. Accessed: Oct. 11, 2025. [Online]. Available:

https://healthcyber.mitre.org/wp-content/uploads/2021/11/774099090_WP_-Health-Delivery-Organizations-and-Ransomware_Final-11-23.pdf

[9] C. R. Junior, I. Becker, and S. Johnson, "Unaware, Unfunded and Uneducated: A Systematic Review of SME Cybersecurity," 2023, arXiv. doi: 10.48550/ARXIV.2309.17186.

[10] L. Ambreen, M. Jain, R. K. Yadav, and S. Loonkar, "Effective cybersecurity risk management practices for small and medium-sized enterprises: A comprehensive review," Multidiscip. Rev., vol. 6, p. 2023ss080, May 2024, doi: 10.31893/multirev.2023ss080.

[11] O.I. Enitan, "Enhancing Cybersecurity Readiness in SMEs: Addressing Resource Constraints and Policy Gaps through Scalable Solutions and IT Investments," IJMCIS, vol. 14, no. 1, pp. 1–6, Feb. 2025, doi: 10.30534/ijmcis/2025/011412025.

[12] A. K. Tetteh, "CYBERSECURITY NEEDS FOR SMES," IIS, Volume 25, Issue 1, pp. 235-246, 2024, doi: 10.48009/1_iis_2024_120.

[13] N. Rawindaran, A. Jayal, E. Prakash, and C. Hewage, "Cost Benefits of Using Machine Learning Features in NIDS for Cyber Security in UK Small Medium Enterprises (SME)," Future Internet, vol. 13, no. 8, p. 186, July 2021, doi: 10.3390/fi13080186.

[14] M. Wallang, M. D. K. Shariffuddin, and M. Mokhtar, "CYBER SECURITY IN SMALL AND MEDIUM ENTERPRISES (SMEs)," jgd, vol. 18, no. 1, pp. 75–87, Dec. 2022, doi: 10.32890/jgd2022.18.1.5.

[15] N. Ntingi, S. Von Solms, and J. Du Toit, "Towards an active cyber defence framework for SMMEs in developing countries," eccws, vol. 22, no. 1, pp. 341–348, June 2023, doi: 10.34190/eccws.22.1.1053.

[16] S. Jinu, K. V. Krishnan, P. Yadav, M. Ramanan, and A. S. Revathy, "Blocking malicious domains: An experimental case study on DNS RPZ mechanism," 2024 27th

International Symposium on Wireless Personal Multimedia Communications (WPMC). IEEE, pp. 1–4, Nov. 17, 2024. doi: 10.1109/wpmc63271.2024.10863586.

[17] M. van Haastrecht et al., “A Shared Cyber Threat Intelligence Solution for SMEs,” Electronics, vol. 10, no. 23, p. 2913, Nov. 2021, doi: 10.3390/electronics10232913.

[18] M. van Haastrecht, I. Sarhan, A. Shojaifar, L. Baumgartner, W. Mallouli, and M. Spruit, “A Threat-Based Cybersecurity Risk Assessment Approach Addressing SME Needs,” Proceedings of the 16th International Conference on Availability, Reliability and Security. ACM, pp. 1–12, Aug. 17, 2021. doi: 10.1145/3465481.3469199.

[19] European Union Agency for Cybersecurity (ENISA), “SMEs Cybersecurity,” ENISA, 2025. Accessed: Dec. 1, 2025. [Online]. Available: https://www.enisa.europa.eu/topics/awareness-and-cyber-hygiene/smes-cybersecurity

[20] S. Bauer, D. Fischer, C. Sauerwein, S. Latzel, D. Stelzer, and R. Breu, “Towards an Evaluation Framework for Threat Intelligence Sharing Platforms,” Proceedings of the Annual Hawaii International Conference on System Sciences. Hawaii International Conference on System Sciences, 2020. doi: 10.24251/hicss.2020.239.

[21] A. de Melo e Silva, J. J. Costa Gondim, R. de Oliveira Albuquerque, and L. J. García Villalba, “A Methodology to Evaluate Standards and Platforms within Cyber Threat Intelligence,” Future Internet, vol. 12, no. 6, p. 108, June 2020, doi: 10.3390/fi12060108.

[22] B. Stojkovski, G. Lenzini, V. Koenig, and S. Rivas, “What’s in a Cyber Threat Intelligence sharing platform?,” Annual Computer Security Applications Conference. ACM, pp. 385–398, Dec. 06, 2021. doi: 10.1145/3485832.3488030.

[23] C. Wagner, A. Dulaunoy, G. Wagener, and A. Iklody, “MISP,” Proceedings of the 2016 ACM on Workshop on Information Sharing and Collaborative Security. ACM, pp. 49–56, Oct. 24, 2016. doi: 10.1145/2994539.2994542.

[24] S. Mokaddem, G. Wagener, A. Dulaunoy, and A. Iklody. 2019. "Taxonomy driven indicator scoring in MISP threat intelligence platforms." ArXiv abs/1902.03914.

[25] A. Iklody, G. Wagener, A. Dulaunoy, S. Mokaddem, and C. Wagner. 2018. "Decaying Indicators of Compromise." ArXiv abs/1803.11052.

[26] MISP Project, “MISP Documentation,” MISP Project, 2025. Accessed: Oct. 11, 2025. [Online]. Available:https://www.misp-project.org/documentation/

[27] R.-J. Hung, C.-C. Hsu, and J.-H. Ho, “Malicious Traffic Blocking Mechanism and Protection Based on DNS,” 2023 IEEE 5th Eurasia Conference on IOT, Communication and Engineering (ECICE). IEEE, pp. 88–91, Oct. 27, 2023. doi: 10.1109/ecice59523.2023.10383095.

[28] J. Magnusson, “Survey and Analysis of DNS Filtering Components,” 2024, arXiv. doi: 10.48550/ARXIV.2401.03864.

[29] S. N. Patil, “PI-HOLES AD BLOCKING SYSTEM USING RASPBERRY PI,” IJSREM, vol. 09, no. 06, pp. 1–9, June 2025, doi: 10.55041/ijsrem49578.

[30] A. D. Yudhistira and R. Harwahyu, “Implementation Strategy Analysis of Network Security using dalo RADIUS and Pi-hole DNS Server to enhance Computer Network Security, Case Study: XYZ as a Fintech Company,” jist, vol. 5, no. 10, pp. 4364–4379, Oct. 2024, doi: 10.59141/jist.v5i10.5321.

[31] A. I. Arkam, M. Yahya, and A. Wahid, “Content Blocking Method To Reduce False Positives Based On Machine Learning,” IOTA, vol. 5, no. 2, pp. 419–435, May 2025, doi: 10.31763/iota.v5i2.935.

[32] Pi-hole, “Pi-hole Documentation,” Pi-hole Documentation, 2025. Accessed: Oct. 11, 2025. [Online]. Available: https://docs.pi-hole.net

[33] A. Mani, T. Vaidya, D. Dworken, and M. Sherr, “An Extensive Evaluation of the Internet’s Open Proxies,” Proceedings of the 34th Annual Computer Security Applications Conference. ACM, pp. 252–265, Dec. 03, 2018. doi: 10.1145/3274694.3274711.

[34] C. Oh, J. Ha, and H. Roh, “A Survey on TLS-Encrypted Malware Network Traffic Analysis Applicable to Security Operations Centers,” Applied Sciences, vol. 12, no. 1, p. 155, Dec. 2021, doi: 10.3390/app12010155.

[35] X. de Carné de Carnavalet and P. C. van Oorschot, “A Survey and Analysis of TLS Interception Mechanisms and Motivations: Exploring how end-to-end TLS is made ‘end-to-me’ for web traffic,” ACM Comput. Surv., vol. 55, no. 13s, pp. 1–40, July 2023, doi: 10.1145/3580522.

[36] L. Waked, M. Mannan, and A. Youssef, “The Sorry State of TLS Security in Enterprise Interception Appliances,” Digital Threats, vol. 1, no. 2, pp. 1–26, May 2020, doi: 10.1145/3372802.

[37] R. A. Fainchtein, A. J. Aviv, M. Sherr, S. Ribaudo, and A. Khullar, “Holes in the Geofence: Privacy Vulnerabilities in ‘Smart’ DNS Services,” Proceedings on Privacy Enhancing Technologies, vol. 2021, no. 2, pp. 151–172, Jan. 2021, doi: 10.2478/popets-2021-0022.

[38] V. K. Maurya, S. Chaudhari, D. K. Sirohi, S. Tomar, A. Rajan, and A. Rawat, “Inimitable Approach to Detect & Quarantine Botnet Malware Infections in Network,” 2021 2nd International Conference on Secure Cyber Computing and Communications (ICSCCC). IEEE, pp. 238–243, May 21, 2021. doi: 10.1109/icsccc51823.2021.9478117.

[39] L. Waked, M. Mannan, and A. Youssef, “To Intercept or Not to Intercept,” Proceedings of the 2018 on Asia Conference on Computer and Communications Security. ACM, pp. 399–412, May 29, 2018. doi: 10.1145/3196494.3196528.

[40] M. Ammi, “Cyber Threat Hunting Case Study using MISP,” JISIS, vol. 13, no. 2, pp. 1–29, May 2023, doi: 10.58346/jisis.2023.i2.001.

[41] L. J. Borges Amaro, B. W. Percilio Azevedo, F. L. Lopes de Mendonca, W. F. Giozza, R. de O. Albuquerque, and L. J. García Villalba, “Methodological Framework to Collect, Process, Analyze and Visualize Cyber Threat Intelligence Data,” Applied Sciences, vol. 12, no. 3, p. 1205, Jan. 2022, doi: 10.3390/app12031205.

[42] K. S. Tharayil, P. Kintis, and A. D. Keromytis, “Augmenting DNS-Based Security with NetFlow,” 2024 4th International Conference on Electrical, Computer, Communications and Mechatronics Engineering (ICECCME). IEEE, pp. 01–06, Nov. 04, 2024. doi: 10.1109/iceccme62383.2024.10797094.

[43] P. Gao et al., “Enabling Efficient Cyber Threat Hunting With Cyber Threat Intelligence,” 2021 IEEE 37th International Conference on Data Engineering (ICDE). IEEE, Apr. 2021. doi: 10.1109/icde51399.2021.00024.

[44] A. Oprea, Z. Li, T.-F. Yen, S. H. Chin, and S. Alrwais, “Detection of Early-Stage Enterprise Infection by Mining Large-Scale Log Data,” 2015 45th Annual IEEE/IFIP International Conference on Dependable Systems and Networks. IEEE, June 2015. doi: 10.1109/dsn.2015.14.

[45] P.-C. Lin et al., “Correlation of cyber threat intelligence with sightings for intelligence assessment and augmentation,” Computer Networks, vol. 228, p. 109736, June 2023, doi: 10.1016/j.comnet.2023.109736.

[46] G. Akiwate et al., “Retroactive identification of targeted DNS infrastructure hijacking,” Proceedings of the 22nd ACM Internet Measurement Conference. ACM, pp. 14–32, Oct. 25, 2022. doi: 10.1145/3517745.3561425.

[47] R. Houser, S. Hao, Z. Li, D. Liu, C. Cotton, and H. Wang, “A Comprehensive Measurement-based Investigation of DNS Hijacking,” 2021 40th International Symposium on Reliable Distributed Systems (SRDS). IEEE, pp. 210–221, Sept. 2021. doi: 10.1109/srds53918.2021.00029.

[48] R. Kumar and P. K. Chaudhary, “Network Security Enhancement using CTI and Log Analysis,” int. jour. eng. com. sci, vol. 7, no. 12, pp. 24430–24432, Dec. 2018, doi: 10.18535/ijecs/v7i12.03.

[49] P. Panero, L. Vâlsan, V. Brillault, and I. C. Schuszter, “Building a large scale Intrusion Detection System using Big Data technologies,” Proceedings of International Symposium on Grids and Clouds 2018 in conjunction with Frontiers in Computational Drug Discovery — PoS(ISGC 2018 & FCDD). Sissa Medialab, p. 014, Dec. 14, 2018. doi: 10.22323/1.327.0014.

[50] N. Serketzis, V. Katos, C. Ilioudis, D. Baltatzis, and G. J. Pangalos, “Actionable threat intelligence for digital forensics readiness,” ICS, vol. 27, no. 2, pp. 273–291, June 2019, doi: 10.1108/ics-09-2018-0110.

[51] A. Papanikolaou, A. Alevizopoulos, C. Ilioudi, K. Demertzis, and K. Rantos, “A Cyber Threat Intelligence Management Platform for Industrial Environments,” Signal Processing and Vision. Academy and Industry Research Collaboration Center (AIRCC), pp. 65–73, Dec. 17, 2022. doi: 10.5121/csit.2022.122206.

[52] I. Vacas, I. Medeiros, and N. Neves, “Detecting Network Threats using OSINT Knowledge-Based IDS,” 2018 14th European Dependable Computing Conference (EDCC). IEEE, pp. 128–135, Sept. 2018. doi: 10.1109/edcc.2018.00031.

[53] N. Paulins, “Improving intrusion detection intelligence by open data usage,” Research for Rural Development, vol. 38. Latvia University of Life Sciences and Technologies, pp. 278–283, Dec. 17, 2023. doi: 10.22616/rrd.29.2023.039.

[54] E. Aljbour, A. Dabit, M. Al-Fayoumi, and Q. A. Al-Haija, “UNI-CERT: A Unified Computer Emergency Response Teams Model for Malware Information Sharing Platform,” 2023 IEEE 5th International Conference on Power, Intelligent Computing and Systems (ICPICS). IEEE, pp. 404–410, July 14, 2023. doi: 10.1109/icpics58376.2023.10235378.

[55] A. Spyros, K. Rantos, A. Papanikolaou, and C. Ilioudis, “An Innovative Self-Healing Approach with STIX Data Utilisation,” Proceedings of the 17th International Joint Conference on e-Business and Telecommunications. SCITEPRESS - Science and Technology Publications, pp. 645–651, 2020. doi: 10.5220/0009893306450651.

[56] G. Ohak, MISP-to-PROXY: Open-source CTI-driven DNS, Proxy, and Retro-Hunt Framework. GitHub repository. Accessed: Jan. 4, 2026. [Online]. Available: https://github.com/ohakgrete/MISP-to-PROXY

[57] Squid Project, Squid: Documentation. Accessed: Oct. 11, 2025. [Online]. Available: https://www.squid-cache.org/Doc/

[58] A. R. Hevner, S. T. March, J. Park, and S. Ram, “Design science in information systems research,” MIS Quarterly, vol. 28, no. 1, pp. 75–105, 2004. Accessed: Oct. 11, 2025. [Online]. Available: https://wise.vub.ac.be/sites/default/files/thesis_info/design_science.pdf

[59] A. Atasi Kalo, M. Schuba, and F. Wiesenfeller, “Open-Source Cyberthreat Intelligence Integration for SMEs,” Proceedings of the 2025 International Conference on Artificial Intelligence, Big Data, Computing and Data Communication Systems. ACM, pp. 1–10, Nov. 25, 2025. doi: 10.1145/3759023.3759108.

Appendix 1 – Non-exclusive licence for reproduction and publication of a graduation thesis[1]

I, Grete Ohak

1 grant Tallinn University of Technology free licence (non-exclusive licence) for my thesis ” Enhancing Small Business Cybersecurity with Multi-Layer Threat Blocking and Retrospective Hunting”, supervised by Shaymaa Mamdouh Khalil and Mert Meiessaar

1.1 to be reproduced for the purposes of preservation and electronic publication of the graduation thesis, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright;

1.2 to be published via the web of Tallinn University of Technology, incl. to be entered in the digital collection of the library of Tallinn University of Technology until expiry of the term of copyright.

2 I am aware that the author also retains the rights specified in clause 1 of the non-exclusive licence.

3 I confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights, the rights arising from the Personal Data Protection Act or rights arising from other legislation.

[1] The non-exclusive licence is not valid during the validity of access restriction indicated in the student's application for restriction on access to the graduation thesis that has been signed by the school's dean, except in case of the university's right to reproduce the thesis for preservation purposes only. If a graduation thesis is based on the joint creative activity of two or more persons and the co-author(s) has/have not granted, by the set deadline, the student defending his/her graduation thesis consent to reproduce and publish the graduation thesis in compliance with clauses 1.1 and 1.2 of the non-exclusive licence, the non-exclusive license shall not be valid for the period.