UNIVERSITY OF TARTU
Institute of Computer Science
Cybersecurity Curriculum

Juri Djomin

Evaluating SAMR Enumeration Exposure in Multi-Forest Active Directory

Master's Thesis (21 ECTS)

Supervisors:

Raimundas Matulevičius, PhD
Juhan Aasaru, MSc

Tartu 2025

Evaluating SAMR Enumeration Exposure in Multi-Forest Active Directory

Abstract:
Security Account Manager Remote (SAMR) enumeration can be used as a reconnaissance technique against Active Directory. Existing enumeration tools expose only a subset of protocol operations and seldom measure how inter-forest trust configuration or Access Control List (ACL) settings limit cross-forest data exposure. A comparative analysis of nine public SAMR enumeration utilities highlights these limitations and frames the design goals of this work. This thesis presents `samr-enum`, a client that implements a broad set of the SAMR enumeration operations relevant to directory reconnaissance and returns output with suitable detail. The tool is exercised in a multi-forest laboratory environment containing tens of thousands of directory objects configured under multiple trust and ACL settings. The experiments demonstrate that `samr-enum` honours these configurations. Under forest-wide authentication a non-administrative account enumerated all user and group objects. The same invocation returned no objects when selective authentication was enabled without the *Allow to authenticate* right. Attribute statistics followed the same pattern, confirming that the client reliably mirrors visibility boundaries imposed by trust scope and ACL entries. Experimental logs, operation numbers (*OpNums*), success counts, and object totals are included on a GitHub project page for verification. The study provides administrators and auditors with a procedure and a software instrument to check how trust design and ACL changes influence information exposure through SAMR.

Keywords:
Security Account Manager Remote Protocol (SAMR), Active Directory Enumeration, Cross-Forest Trusts, Selective and Forest-Wide Authentication, Access Control Lists (ACL), Information Exposure Assessment, Multi-Forest Environments, Security Tool Development.

CERCS: P170 - Computer science, numerical analysis, system, control

SAMR-loendusründe käigus andmete paljastumise hindamine Active Directory mitmemetsa keskkonnas

Lühikokkuvõte:
Security Account Manager Remote (SAMR) loetlemist võib kasutada luuretehnikana Active Directory (AD) keskkondade vastu. Olemasolevad päringutööriistad toetavad vaid osa SAMR-protokolli kõigist võimalikest operatsioonidest ja ei paku piisavat tuge, et välja selgitada, kas AD-metsasisesed (inter-forest) usalduskonfiguratsioonid või juurdepääsu kontrollnimekirjade (Access Control List) seaded on piisavad metsadevahelise (cross-forest) andmelekkekoormuse maandamiseks. Põhjalik võrdlev analüüs üheksa avaliku SAMR-loendustööriista kohta toob need piirangud esile ja raamib käesoleva töö eesmärgid. Käesolev magistritöö tutvustab uut tööriista `samr-enum` — klientrakendust, mis realiseerib 65,2% SAMR-i loendusoperatsioonidest, mis on asjakohased kataloogiandmete tuvastamiseks, ja tagastab tulemuse piisava detailsusega.

Tööriista on katsetatud mitmemetsa (multi-forest) laborikeskkonnas, milles oli kümneid tuhandeid kataloogiobjekte erinevate usaldus- ja ACL-konfiguratsioonidega. Katsed näitasid, et `samr-enum` järgib rangelt usalduse ulatust ja ACL-sid: metsaülese (forest-wide) autentimise puhul tagastas klient kõigi kasutaja- ja grupiobjektide täieliku loendi, samas kui selective authentication ilma *Allow to authenticate* õiguseta andis tühja tulemuse. Sama muster ilmnes atribuutide statistikas, kinnitades, et rakendus peegeldab täpselt nähtavuse piire, mida trust'i seaded ja ACL-id määravad. Eksperimentaallogid, protokollioperatsioonide numbrid (OpNum-id), edukate päringute arvud ja objektide koguhulgad on avaldatud kontrollimiseks projekti GitHubi lehel. Uuring pakub süsteemiadministraatoritele ja infoturbe spetsialistidele metoodikat ning töövahendit, millega hinnata, kuidas usalduse (*trust*) disain ja ACL-seadistuste muutmine mõjutavad andmete kättesaadavust SAMR-protokolli kaudu.

Võtmesõnad:
Security Account Manager Remote Protocol (SAMR), Active Directory loendusrünne (enumeration), Active Directory metsade vahelised usaldusseaded, Active Directory metsaülene autentimine, selektiivne autentimine, juurdepääsu kontrollnimekirjad, informatsiooni kättesaadavuse hindamine, Active Directory mitmemetsa keskkonnad, turvatööriistade arendus

CERCS: P170 - Arvutiteadus, numbriline analüüs, süsteem, juhtimine

Acknowledgements

I would like to express my sincere gratitude to my academic supervisors, **Dr. Raimundas Matulevičius** and **Juhan Aasaru, MSc**, for their invaluable guidance, patience, and time. Over a period of more than eight months, they consistently found time to discuss progress, maintain motivation, and steer research in the right direction. Their professionalism and supportive attitude made working on this thesis both productive and inspiring.

I am also deeply grateful to my family for their support in all its forms—both obvious and delightfully invisible (such as the sudden appearance of a cup of coffee right when LaTeX decides to crash). Your patience with endless SAMR-related talks and your shared appreciation for peaceful Wi-Fi made this work possible.

Contents

List of Abbreviations

ACL	**A**ccess **C**ontrol **L**ist
ACE	**A**ccess **C**ontrol **E**ntry
AD	**A**ctive **D**irectory
AP-REQ	**A**uthenticator and **P**rincipal **REQ**uest
AV	**A**nti**V**irus
DACL	**D**iscretionary **A**ccess **C**ontrol **L**ist
DCE	**D**istributed **C**omputing **E**nvironment
FSP	**F**oreign **S**ecurity **P**rincipal
gMSA/MSA	group **M**anaged **S**ervice **A**ccount
LDAP	**L**ightweight **D**irectory **A**ccess **P**rotocol
NTLM	**N**T **L**AN **M**anager
OpNum	**Op**eration **Num**ber
RID	**R**elative **I**dentifier
RPC	**R**emote **P**rocedure **C**all
SAM	**S**ecurity **A**ccount **M**anager
SAMR	**S**ecurity **A**ccount **M**anager **R**emote protocol
SID	**S**ecurity **I**dentifier
SMB	**S**erver **M**essage **B**lock
SPN	**S**ervice **P**rincipal **N**ame
TGT	**T**icket **G**ranting **T**icket
VLAN	**V**irtual **LAN**

1 Introduction

1.1 Motivation

Threat actors increasingly leverage SAMR for reconnaissance in Active Directory environments. This protocol enables remote queries of user and group accounts, often even by low-privileged users, resulting in lists of privileged domain administrators and group memberships. This information provides a blueprint for the domain structure[1] and primary targets, serving as an initial foothold toward a complete domain compromise. Real-world incidents confirm the risk. In 2021, attackers first compromised a domain controller and then used SAMR queries over SMB to retrieve a list of sensitive accounts [1]. An analysis of the 2024 BlackSuit ransomware incident revealed that the attackers deployed SharpHound to query the SAMR service on multiple domain controllers, extracting user and group data [2]. In both cases, SAMR reconnaissance enabled adversaries to identify high-value accounts and relationships, which in turn accelerated their lateral movement and facilitated domain-wide compromise. These examples illustrate how SAMR enumeration is not a benign query but a deliberate tactic that materially helps attackers map out and take over Active Directory domains.

In multi-domain or multi-forest deployments, the dangers of SAMR exposure are amplified. A compromise in one domain can be leveraged to enumerate and exploit accounts in a trusted domain or forest via cross-domain SAMR queries [3]. Security researchers have demonstrated *trust-hopping* attacks. After compromising one forest, an attacker can use SAMR to enumerate the local Administrator group on domain controllers in another forest [3]. This cross-forest reconnaissance effectively undermines the intended security boundary of separate Active Directory forests, as an attacker can discover and target privileged accounts across trust relationships. The potential for such cascading compromise highlights why a systematic measurement of SAMR exposure is necessary. However, many organisations do not adequately assess their Active Directory for these hidden exposures [4]. A quantitative assessment of where SAMR enumeration is permitted, especially across domain and forest boundaries, helps defenders gauge exposure and apply hardening measures, such as restricting remote SAMR calls. This research is motivated by the need to illuminate the extent of SAMR enumeration in real environments and to provide data-driven insight into mitigating the associated cross-forest security risks.

[1] See Glossary for definition of domain, forest, and trust

1.2 Problem Statement

In multi-forest Active Directory environments, the behaviour of SAMR when enumerating directory information across forest trust boundaries remains insufficiently understood. To address this issue, the study introduces a structured and tool-based approach. The approach is evaluated in a controlled multi-forest laboratory under varying trust relationships and ACL settings. Through systematic experiments and qualitative analysis, the research highlights the practical behaviours and limitations of SAMR in multi-forest scenarios, thereby providing insight into its cross-forest capabilities and constraints.

> How can a structured tool-based approach be employed to observe and compare SAMR enumeration behaviour in a multi-forest Active Directory environment under different trust relationships and access control configurations?

Two supporting sub-questions further guide this inquiry:

1. How do different inter-forest authentication scopes, specifically forest-wide authentication versus selective authentication, influence the scope of cross-forest SAMR enumeration?

2. How do variations in Access Control List permissions change the directory information that a non-privileged user can retrieve across forest boundaries using SAMR?

1.3 Objectives and Scope

The aim of this thesis is to examine the behaviour of SAMR in a multi-forest Active Directory environment and to measure how trust settings and access-control rights influence the volume of information that the protocol reveals. In order to reach this aim the study (i) reviews academic literature, white papers and vendor documentation, (ii) evaluates existing SAMR enumeration utilities, (iii) designs and implements the `samr-enum` tool, (iv) builds a laboratory Active Directory that contains several forests, (v) conducts a set of controlled experiments that vary trust configuration and discretionary access-control lists, and (vi) analyses the resulting data.

The scope of the research is deliberately bounded. All experiments use Windows Server 2022 domain controllers, a two-way trust, and read-only SAMR operations; the study does not attempt privilege escalation, vulnerability discovery, or performance benchmarking. LDAP, Kerberos, and other directory interfaces are referenced only as background and are not instrumented. Results therefore generalise to scenarios where organisations wish to understand SAMR visibility within

a comparable multi-forest topology, but they do not extend to legacy operating systems, write-capable SAMR calls, or mixed on-premises and cloud domains.

1.4 Contributions

The research introduces `samr-enum`, a custom tool for systematically enumerating information via SAMR in Active Directory. The tool addresses a practical gap in existing auditing techniques by enabling structured measurement of SAMR enumeration. Specifically, `samr-enum` can operate under varied trust relationships and access control list (ACL) conditions in multi-forest Active Directory environments.

For administrators and security auditors, `samr-enum` provides a method to assess and understand the exposure of SAMR-based information in complex Active Directory environments. The contributions of the work are mainly practical. It offers a structured approach to document and quantify existing SAMR enumeration behaviours in different trust and ACL scenarios. In particular, the work does not introduce any new security vulnerabilities. Instead, it systematically documents how different trust relationships and ACL settings influence the information that can be retrieved through SAMR.

1.5 Declaration on Generative AI

I hereby declare that during the preparation of this thesis several tools powered by artificial intelligence were employed in supporting roles. Specifically, OpenAI ChatGPT (using GPT-4o, o1 and o3 models) and the DeepSeek R1 system were used to help structure program code, resolve programming errors, and suggest improvements in language. In addition, Grammarly was used to correct grammar and refine writing style, and the SciSpace platform helped with the literature search.

I affirm that these tools were not used to generate any substantive content of the thesis; they served strictly as auxiliary aids. All essential aspects of the research (including the design of the experiments, the development of the application, the data analysis, and the interpretation of the results) were performed independently by me as the author.

2 Literature Review

Introduction. This chapter provides the foundational knowledge needed to understand SAMR enumeration in multi-forest Active Directory (AD) environments. Examines previous work and key concepts that support the objective of research of creating and evaluating `samr-enum`. The focus is on the literature related to measuring and assessing the risk of information exposure through SAMR queries. Consequently, the review emphasises security implications and risk assessment rather than the design of new attack techniques. By framing the discussion in terms of information exposure, this chapter ensures that the development of `samr-enum` is grounded in understanding how SAMR-based enumeration can affect security in multiple Active Directory forests.

2.1 Active Directory Enumeration: Concepts and Threat Context

Active Directory enumeration is increasingly recognised as a critical reconnaissance step in multi-stage attack chains. Mokhtar et al. (2022) characterise domain enumeration as an initial phase that adversaries undertake after gaining a foothold, using it to locate weak points and privileged accounts for subsequent privilege escalation [5]. Their work aimed to provide information on the criticality and detection of AD attacks by reviewing known attack vectors and simulating two representative privilege-escalation scenarios to examine Windows event log signatures. A key finding from this study is that most domain reconnaissance can be executed by standard user accounts using native tools (e.g., `net.exe`) or well-known offensive tools (such as PowerView or BloodHound), providing extensive information on domains, trusts, privileged groups, configurations, and other potential targets. This underscores the ease and breadth of malicious AD enumeration. However, the paper is primarily a qualitative survey; it does not quantify how prevalent or exposed real-world AD environments are to such enumeration tactics, instead focusing on general insights and detection recommendations.

A more recent industry threat-research post by the Palo Alto Networks Unit 42 team (2024) examines LDAP-based AD enumeration as a growing threat and detection challenge [6]. It presents real-world evidence that advanced threat actors (including nation-state APT groups and ransomware operators) abuse LDAP queries for internal reconnaissance, mapping out user accounts, group memberships, and other directory data as part of the attack discovery phase. By analysing incident investigations, Unit 42 researchers identify common malicious LDAP query patterns and tools – for example, repurposed admin utilities such as AdFind (often renamed by attackers to evade detection) and the BloodHound/SharpHound toolkit – that are commonly employed to enumerate AD environments. In particular,

the report highlights that distinguishing such malicious enumeration from normal administrative traffic is difficult due to the sheer volume of legitimate directory queries in enterprise domain controller logs. This creates a gap in detection coverage because LDAP-based reconnaissance can blend with routine operations. Although Unit 42 analysis offers practical detection guidance, it does not provide a structured quantification of enumeration exposure. In particular, neither it nor the study by Mokhtar et al. systematically measures the extent of SAMR enumeration exposure across multi-forest Active Directory environments, a notable gap that motivates the present thesis.

2.2 SAMR Fundamentals

Recent work confirms that SAMR remains an attractive target for attackers because it exposes user and group information over the network. Kabibo (2024) demonstrates that legacy *null-session* behaviour can be revived: By analysing the RPC traffic to the `\PIPE\samr` interface and abusing an undocumented bind sequence, the author anonymously enumerated domain users on fully patched domain controllers [7]. The experiment shows that, despite default hardening, SAMR can still disclose directory objects when misconfigurations persist; however, the attack is successful only on hosts that allow anonymous pipe access, limiting its generality. Defatsch (2022) provides a complementary case study of SharpHound, showing how the tool retrieves local group membership across the network via SAMR calls such as `GetMembersInAlias` [8]. Trace analysis confirms that, starting with Windows 10 v1607 and Windows Server 2016, remote SAMR calls to workstations and member servers are permitted only to administrators, so low-privileged users receive access-denied responses; on domain controllers these queries remain allowed by default unless an explicit hardening policy (for example, *Network access: Restrict clients allowed to make remote SAM calls*) is applied; SharpHound must then return to slower, less complete LDAP or Group Policy parsing. Together, these studies highlight the dual nature of SAMR: powerful enumeration when the interface is reachable and sharply reduced visibility when operating system safeguards are enabled.

The manipulation potential of the protocol extends beyond the enumeration. Castro and Cárdenas (2024) examine the SAMR account management functions of SAMR and introduce *invisible accounts*: by hijacking relative identifiers (RIDs) or injecting credentials directly into the SAM database, they create stealth administrative users that the standard SAMR listing fails to reveal [9]. Although these attacks require prior local privilege, they expose a gap between SAMR's design, returning objects by RID, and higher-level security checks that assume a one-to-one mapping between account names and identifiers. Collectively, the papers underline a key research gap: none offers a systematic measurement of

SAMR exposure across modern multi-forest environments. Addressing this gap is the objective of the present investigation, which employs the `samr-enum` tool to quantify the information SAMR discloses under various trust and access control conditions.

2.3 Cross-Forest Trusts and Access-Control Principles

Schroeder (2018) and Mollema (2021) independently show - through lab-based exploit execution and network trace analysis - that a two-way forest trust is penetrable: the first work uses a printer-spooler flaw plus unconstrained delegation to impersonate privileged accounts across forests, while the second forges inter-realm Kerberos tickets (CVE-2020-0665) to bypass SID filtering and obtain elevated rights in a trusting forest [10] [11]. Robbins and Schroeder (2018) extend the risk by manipulating discretionary access control lists: In test domains, they grant a low-privilege (or foreign) principal the *WriteDACL* right, thus embedding hidden administrative privileges that standard queries overlook [3]. All three studies rely on practical proofs of concept against specific misconfigurations, and none quantifies how common such weaknesses—or the related information exposure via protocols like SAMR—are in production multi-forest environments. The absence of structured measurement motivates the present thesis, which employs `samr-enum` to evaluate how trust links and ACL settings together determine the amount of directory data revealed across forest boundaries.

2.4 Enumeration Tool Landscape and Limitations

A survey of contemporary tools that perform enumeration exclusively through the SAMR interface illustrates both current capabilities and their defensive constraints. A 2022 technical analysis of CrackMapExec (CME) confirmed that its SAMR module extracts user and local administrator groups via the `\PIPE\samr` interface. However, the post-2016 Microsoft policy *Restrict clients allowed to make remote SAM calls* blocks such queries for non-admins, limiting CME to systems without that restriction [12]. Enum4linux-NG, a 2020 rewrite of the long-standing enum4linux script, was profiled in its release note, which details a Python wrapper around Samba's rpcclient; black-box trials demonstrated reliable extraction of domain user and group RIDs through `SamrEnumerate*` calls, yet the note concedes that the tool returns empty sets when SMB signing is enforced or anonymous sessions are refused, limiting its usefulness to legacy or misconfigured networks [13]. Finally, Rapid7's 2023 update of Metasploit's auxiliary module `samr_account` documents controlled experiments against Windows Server 2022: the module retrieves basic account metadata via `SamrQueryDisplayInformation`, but cannot enumerate nested group memberships or domain-local groups and therefore recommends supplementing

results with LDAP queries [14]. Together, these evaluations, packet capture analysis for CME, wrapped library inspection for Enum4linux-NG, and controlled testing for Metasploit show that modern SAMR tools automate enumeration, but are increasingly curtailed by hardened defaults and yield only fragmentary data in multi-forest contexts, leaving a measurement gap that the present study seeks to address through systematic, cross-forest assessment.

2.5 Identified Research Gap

Existing literature confirms that SAMR can reveal extensive account and group information, yet prior studies focus either on isolated misconfigurations (printer-spooler delegation, forged SID filtering, over-permissive ACLs) or on single-domain reconnaissance tools whose coverage is fragmentary once modern *restrict remote SAM* policies apply. None of these works measures, in a controlled and repeatable manner, how the volume and type of data returned by SAMR vary when trust scopes (forest-wide versus selective authentication) intersect with customised ACLs across multiple forests. This lack of quantitative cross-forest evidence defines the research gap that the present study addresses by end-to-end instrumenting SAMR and recording object counts, attribute visibility, and protocol responses under systematically altered trust and permission conditions.

3 Comparative Analysis of Existing SAMR Enumeration Tools

Introduction. This section provides a systematic evaluation of existing SAMR enumeration tools, analysing the functionality and coverage of each tool using various criteria relevant to Active Directory environments. The goal is to understand the capabilities and limitations of each tool, informing the design of a new tool to meet the gaps identified in existing solutions.

RPC Header (Transport Layer)

- UUID (Interface ID)
- Packet Type
- Call ID
- Fragment Length
- Authentication Length

SAMR Request Header

- Function Code (OpNum) <– **Comparison Criterion**
- Access Mask
- Desired Access <– **Comparison Criterion**
- Context Handle

SAMR Request Body

- Object Handle
- Input Parameters

- Specific object identifiers (e.g., RIDs, SIDs)
- Domain/User/Group identifiers

- Output Parameters

Table 1. Structure of SAMR Request Packet

RPC Header (Transport Layer)

- UUID (Interface ID)
- Packet Type
- Call ID
- Fragment Length
- Authentication Length

SAMR Response Header

- Status Code
- Context Handle (Returned)
- OpNum (Echoed from Request)
- Access Mask (Returned)

SAMR Response Body

- Object Handle (Returned)
- Output Parameters <− **Comparison Criterion**

- Specific object data (e.g., RIDs, SIDs)
- Domain/User/Group details

- Result Flags
- Error Information (if applicable)

Table 2. Structure of SAMR Response Packet

3.1 Key Criteria for SAMR Tool Evaluation

This section defines the evaluation criteria for SAMR enumeration tools, focusing on their ability to extract directory information in multi-forest Active Directory environments. The criteria are derived from the structure of the SAMR request and response packets, as summarized in Table 1 and Table 2. A detailed explanation of the fields referenced in these tables is available in Appendix A.

3.1.1 Cross-Forest Support

Multi-forest environments often rely on trust relationships and authentication settings, influencing data accessibility across forest boundaries. This criterion

evaluates each tool's ability to perform enumeration across a single forest trust, assessing how well it handles this configuration. By focusing on one forest trust, the analysis will explore the effectiveness of each tool's cross-forest enumeration capabilities in a simplified but representative multi-forest setup. This approach allows for practical insight into tool functionality within cross-forest configurations while limiting the complexity of testing multiple trust types.

3.1.2 OpNum Coverage

The range of SAMR operations, identified by their OpNums, is central to understanding how each tool gathers different data types. This criterion assesses whether each tool supports the SAMR operations necessary for cross-forest enumeration, including user, group, and domain enumeration, as well as retrieving security settings and password policies. The evaluation of OpNum coverage is crucial as it determines the range of enumeration functions supported by each tool, which affects its ability to retrieve complete domain data.

3.1.3 Access Scope Compliance

The criterion examines whether the requested access rights of each tool are aligned with the access levels specified for each SAMR operation in the protocol documentation. This criterion evaluates how tools set the access mask in the SAMR request packet and whether they request only the permissions listed as necessary for each specific OpNum operation. Lai and Zhang (2006) [15] highlight the essential importance of enforcing the principle of least privilege to minimise unnecessary access requests. Their quantitative framework underscores that excessive permissions, analogous to privilege leaks in role-based access control, not only violate security principles but also increase the potential for misuse or unintended consequences. Tools that adhere to the specified permissions avoid unnecessary access requests, reducing security risks associated with over-privileged enumeration in Active Directory environments. This approach highlights tools that request excessive permissions, potentially increasing security exposure by going beyond the documented access requirements.

3.1.4 Data Parsing and Accuracy

Accurate parsing of SAMR responses ensures that the data collected from the enumeration can be trusted. SAMR responses include complex data structures that vary depending on the operation requested. This criterion examines the ability of each tool to correctly interpret and organise these responses, ensuring precision and completeness in the output. Reliable data parsing is essential in large or

multi-forest setups, where incomplete or misinterpreted data could lead to gaps or inaccuracies in analysis.

3.1.5 Authentication Protocol Support

Authentication protocol support is a key parameter in evaluating tool compatibility in Active Directory environments. NTLM and Kerberos are the primary authentication protocols used within these environments, and the support of each tool for one or both protocols affects its ability to perform SAMR enumeration in different contexts.

The tools used for SAMR enumeration must accommodate authentication protocols configured in the target domain. Since Kerberos is the default mechanism in Windows AD environments, as emphasized by Díaz et al. (2021) [16], its correct handling is critical for tool reliability in realistic deployments.

NTLM, though a legacy protocol, remains necessary in environments with older systems or specific configurations that do not allow Kerberos. Identifying which tools support NTLM, Kerberos, or both protocols provides information on the adaptability of each tool, particularly in multi-forest or cross-domain setups where domain policies may require a specific protocol.

This criterion helps assess the operational flexibility of SAMR enumeration tools within environments that use mixed authentication protocols. By examining authentication support, this research evaluates the suitability of each tool in environments where specific protocols are required to access resources across domain boundaries. This aligns with the research goal of understanding the capabilities and limits of SAMR enumeration tools in various Active Directory structures.

3.1.6 Access Level Requirements

This criterion examines the access privileges required by SAMR enumeration tools to operate within a domain environment. Assesses whether a tool functions with standard user permissions or requires elevated (administrative or root) access. Tools that can perform operations with standard permissions are noted for their ability to operate in environments with limited administrative access. In contrast, tools that require elevated privileges may provide broader SAMR access but can pose security risks in restricted contexts.

This criterion is essential to understand the balance between usability and security. Tools with minimal access requirements support the principle of least privilege, making them suitable for environments with strict security policies. Conversely, tools that need administrative access may offer more extensive functionality, but are less adaptable in scenarios where administrative privileges are restricted.

3.2 Comparison Methods

This section introduces the criteria for evaluating SAMR enumeration tools and outlines how each criterion is measured with respect to the tool's ability to extract data across forest boundaries. These criteria guide the detection of capability gaps between tools and determine whether `samr-enum` can realise functions not yet provided elsewhere. Each criterion examines a specific aspect of tool functionality and data retrieval capability, particularly within multi-forest environments. The evaluation methods align with a taxonomy of reconnaissance techniques, which categorises enumeration as essential for assessing exposure and improving the security posture of Active Directory environments. Previous research highlights that enumeration methods vary significantly in approach, complexity, and reliance on protocol-specific operations [17]. This framework was used to inform the categorisation of tools in this investigation, ensuring that comparisons accurately reflect their capabilities to support reconnaissance objectives.

The comparison methods are based on recommendations by Barman et al. [18], who propose a structured enumeration framework that ensures consistency and thoroughness throughout the evaluation phases. Their framework emphasises the importance of systematically organising the enumeration process through clear documentation, segmentation of tasks, and iterative refinement of outputs. By incorporating tools such as Nmap and Netcat, Barman et al. outline how diverse enumeration approaches can be unified within a single, methodical workflow to reduce inconsistencies and improve efficiency. Applying these principles to SAMR enumeration, this study adopts a similar focus on structured workflows, ensuring that each tool's results are compared in a consistent manner across predefined criteria such as completeness, accuracy, and protocol-specific capabilities. Additionally, integration of documentation and segmentation processes aligns with the goal of capturing detailed outputs and minimising errors during cross-tool evaluation.

The laboratory environment for this criterion consists of two forests connected through a forest-wide trust, as shown in Figure 1. The network and environment configuration details are documented in the repository and are referenced in the Appendix B.

`Domain-z` was designed to include data with special characters, fields in foreign languages, and entries with long names and descriptions. This setup ensures that the tools' capabilities in handling non-standard and complex data formats can be evaluated. `Domain-x`, populated with 20 000 user accounts, 10 000 groups, and 10 000 computer accounts, allows evaluation of tooling capability at scale. This large dataset allows one to assess whether the tools can accurately retrieve and display all information without errors or omissions.

The server configurations, including group policies, authentication settings, and

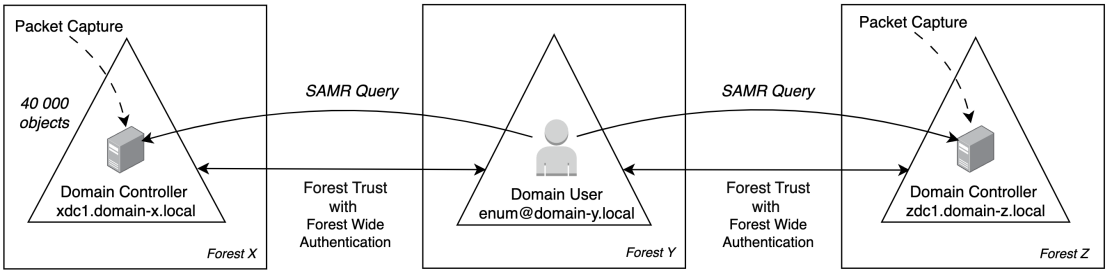

Figure 1. Enumeration Vectors and Trust Relationships

the default domain schema, were not altered to optimise tool performance. This decision reflected real-world conditions, as the default settings represent the most common production setups. Furthermore, adapting configurations to specific tools would introduce bias and invalidate the comparative nature of the study.

Wireshark was installed on both domain controllers to analyse the traffic generated during the enumeration activities. The network configuration ensured that clients and servers were located in different network segments, communicating through a router. This setup mimics real-world network segmentation and enables detailed analysis of traffic from SAMR. References to the captured traffic files and the outputs generated by the evaluated tools have been provided in the Appendix B for further analysis and verification.

3.2.1 Cross-Forest Support

The cross-forest support criterion evaluates the capability of each tool to perform SAMR enumeration across domains within a two-way forest trust configured in two ways for forest-wide authentication. Enumeration requests were executed by a nonprivileged account in `domain-y` to evaluate the ability of the tools to function under common real-world conditions. This specific configuration was selected to isolate the inherent capability of each tool to operate across forests without relying on elevated permissions, to enable an empirical evaluation of cross-forest functionality.

The tools were classified as supporting or not supporting the cross-forest enumeration based on their ability to retrieve data across forest boundaries for all operations tested.

In cases where a tool does not natively support Windows or was originally designed for Linux environments (e.g., Linux-based tools later adapted for Windows), enumeration was conducted from an Ubuntu Linux client. This client was not joined to the domain but specified `domain-y` credentials to authenticate and perform enumeration against foreign domain controllers. This approach allowed for evaluating tools' flexibility and adaptability in non-domain-joined configurations.

Furthermore, the commonly used `net.exe` tool, although lacking cross-forest request support, was included in the analysis. Despite its limitations, this tool was evaluated due to its widespread use among administrators and importance in understanding baseline capabilities.

The evaluation was conducted using either the tools' official documentation or by executing commands directed at foreign domain controllers when the documentation did not explicitly mention cross-forest support. For this criterion, traffic capture and detailed network analysis were not required, as the assessment focused solely on the functional capabilities of the tools and their documented behavior.

3.2.2 OpNum Coverage

The OpNum coverage criterion evaluates the range of SAMR operations supported by each SAMR enumeration tool in collecting data from Active Directory within multi-forest configurations. The analysis focuses on the types and number of SAMR operations supported by the tools, as detailed in Table 5 in the Appendix C.

To assess OpNum coverage, each tool is systematically tested through the execution of supported OpNums, and the resulting network traffic is captured and analysed in Wireshark. This captures the actual support for each tool for SAMR operations and is validated against the tool documentation to confirm the accuracy of OpNum functionality.

In evaluating how each tool handles SAMR operations, the initial connection OpNum itself can vary depending on the server version of Windows. This variation in `SamrConnect` versions generally does not change the core data fields or the general enumeration scope, but can affect whether a given tool successfully connects or negotiates a fully compatible RPC session. The `SamrConnect` call itself exists in multiple versions (`SamrConnect1-5`), and a server's operating system may dictate which variant is used. In this study, scan analysis only records the `SamrConnect` version ultimately used by each tool during testing, not whether the tool can adapt to different versions. Consequently, tools that fail to align their `SamrConnect` variant with the target server might appear to have restricted OpNum coverage in certain environments, even if they support a wider range of calls.

Coverage is measured as a percentage of relevant SAMR OpNums supported by each tool. The levels are categorised as follows: *high* coverage" (80% or more relevant OpNums), *moderate coverage* (50%-79% of relevant OpNums) and *low coverage* (below 50% of relevant OpNums).

3.2.3 Access Scope Compliance

This criterion evaluates whether the access rights requested by each tool are aligned with the permissions specified for each SAMR OpNum in the protocol

documentation. The goal is to identify instances where tools request permissions that fall outside the defined access masks, which may indicate a lack of protocol adherence or introduce unnecessary exposure.
For each tool, the requested *Desired Access* bits are compared with the permissions allowed for each OpNum as defined in the SAMR specification. Tools that request only valid, specification-aligned permissions are marked as *compliant*, while those that include access bits beyond the defined scope are marked as *over-permissioned*.

3.2.4 Data Parsing and Accuracy

The data parsing and accuracy criterion evaluates whether each tool can correctly interpret and present the SAMR response data according to the protocol specifications. Given the structural complexity and operational variability of SAMR responses, this criterion is critical to ensuring the correctness and completeness of the enumeration output. Incorrect or incomplete parsing may result in misleading analysis outcomes, particularly in environments involving large datasets or multi-forest configurations.

The evaluation focuses on the tool's ability to handle SAMR responses that include diverse attributes and complex data formats. The dataset in `domain-x`, populated using the BadBlood tool [19], was specifically designed to test these capabilities. It contains 20 000 user accounts, 10 000 groups, and 10 000 computer accounts, representing a realistic and challenging Active Directory environment. This dataset incorporates attributes such as short and long names, special characters (e.g., # $ % ^ & * \‘ + = _ - <>, . ? / "), and names in foreign languages such as Russian and Chinese. To evaluate edge cases, the `sAMAccountName` attribute was configured for computer accounts with 15-character names. These configurations simulate the variety of data encountered in real-world scenarios.

To ensure accurate parsing and reliable analysis, evaluation methods are inspired by the formal approach to data accuracy evaluation proposed by Athamena et al. [20]. Their method of fragmenting data into areas with uniform accuracy values allows inaccuracies to be isolated and systematically addressed. This framework is particularly relevant when analysing SAMR responses with diverse data attributes and complex formats, as it emphasises the importance of accurate labelling and selective analysis to mitigate errors in parsing.

The output of each tool was compared against the expected structure and values defined in the SAMR documentation. Completeness and accuracy metrics were used to assess the quality of the tool output. Completeness measures whether all fields expected for a particular SAMR operation are retrieved, while accuracy evaluates whether the retrieved values match the expected baseline values. Tools that consistently retrieved and accurately parsed all required fields were marked as *Accurate*. In contrast, tools that produced missing or erroneous data were

categorised as *Incomplete* or *Inaccurate*.

This criterion evaluates whether tools can correctly and consistently parse complex and heterogeneous datasets, with a focus on parsing correctness and robustness under realistic operational conditions. The results highlight the capabilities and limitations of each tool in environments requiring accurate interpretation of the SAMR response data.

3.2.5 Authentication Protocol Support

This criterion assesses the authentication protocols supported by each SAMR enumeration tool to evaluate their compatibility with various domain security configurations. The evaluation involved reviewing the documentation of the tool and capturing network traffic to determine which authentication protocols were used during SAMR operations. This approach made it possible to identify the default authentication protocol employed by each tool (e.g., NTLM, Kerberos) without modifying domain-level authentication policies, thereby avoiding potential issues with cross-forest trust or system stability.

The compatibility of each tool was evaluated based on both the authentication protocols explicitly documented and those observed during network traffic analysis. Tools that adjusted their authentication behaviour in response to domain policy were classified as exhibiting *protocol adaptability*. In contrast, tools that explicitly supported multiple authentication mechanisms, such as NTLM and Kerberos, were marked as *multi-authentication compatible*. No additional authentication protocols were observed beyond those explicitly documented or used by default. This approach provided a clear understanding of the flexibility of each tool's in relation to domain security configurations and its ability to function under varying authentication settings.

3.2.6 Access Level Requirements

Access level requirements were evaluated by testing the ability of each tool's to perform SAMR operations under standard and elevated permissions. The tools were categorised according to the lowest access level required for all SAMR operations tested. Tools that require elevated permissions for any operation were classified as *elevated access required,* while those capable of performing all tested operations with standard permissions were marked as *standard access sufficient.*

This evaluation documents the minimum access requirements for SAMR enumeration, providing clarity on how tools adhere to least privilege principles. Understanding access level requirements helps identify tools that align with security policies, particularly in production environments where access levels must be tightly controlled. Tools that work with standard permissions reduce the risks associated

with escalation of privileges and unauthorised access. These findings offer insights for administrators and researchers into how privilege levels impact data accessibility and tool design.

3.3 Selection of Tools

The selection of tools to evaluate SAMR enumeration capabilities focused on their applicability in modern multi-forest Active Directory environments. Although an essential consideration was the ability to support cross-forest SAMR requests, exceptions were made for specific tools to analyse their behaviour in local SAMR scenarios. Detailed version and configuration information for each evaluated tool is provided in the Appendix D, see Table 7.

For example, `net.exe`, a built-in Windows utility, was included in the evaluation despite its inability to perform cross-forest SAMR queries. Its analysis provides insight into how SAMR operations are implemented by the original developer of the protocol, offering a reference point for correct or expected behaviour. Although limited to single-domain contexts, `net.exe` serves as a baseline to compare third-party tools and assess deviations from Microsoft's native implementation.

In contrast, tools identified as potentially unsafe for use were excluded from the study. For example, RPC Investigator, known for its ability to analyse Windows RPC (Remote Procedure Call) services, including SAMR, was scanned using the TotalVirus online service. The tool received alerts from 20 out of 57 anti-malware providers, raising concerns about its safety despite being available on GitHub[2]. Such exclusions ensure that the research only includes tools that can be deployed securely in controlled environments.

Additionally, tools such as PowerSploit and PowerView were excluded. PowerSploit, which is no longer actively maintained, lacks support for cross-forest SAMR queries and does not align with the focus on modern tools. Similarly, PowerView, including its modern fork (PowerView2), does not offer functionality for targeting foreign domain controllers across forests using SAMR, limiting its utility in multi-forest scenarios.

By prioritizing tools capable of addressing local and cross-forest SAMR queries where possible, this research aims to analyse the enumeration behaviour of SAMRs in diverse configurations. The selected tools enable a comparative analysis of protocol interactions and security implications in local and multi-forest Active Directory environments. Detailed evaluations of the selected tools are presented in subsequent sections.

[2]GitHub. *RPC Investigator repository.* Available: url-https://github.com/trailofbits/RpcInvestigator

3.3.1 Net Commands

The `net user` and `net group` commands are Microsoft-developed built-in Windows utilities used to retrieve and display information about user and group accounts on both local and domain systems. Designed as straightforward command-line tools, these commands enable administrators to list and manage accounts within a Windows environment. The tools are installed as part of the Windows operating system, and no additional setup or installation is required.[3][4]

When executed with the `/domain` flag, `net user` can query user accounts from an Active Directory domain, while `net group` provides details about domain groups. Both commands have a limited scope, focusing on user- and group-specific information rather than broader data fields related to SAMR. They operate exclusively on Windows, require basic permissions, and are suitable for fundamental enumeration tasks in single-domain contexts. The reliance of these tools on minimal access rights and the absence of support for multi-domain environments limit their applicability to local account management rather than comprehensive SAMR enumeration.

3.3.2 Enum4linux

Enum4linux is a Linux-based tool licenced under GPL, used for the enumeration of Windows systems using SMB and SAMR protocols to collect user, group and shared resource information.[5] The tool relies on specified access rights to perform data collection tasks and operates exclusively on Linux. Documentation is available through community resources, and the tool has been widely used for initial data collection and reconnaissance processes. Although primarily employed in scenarios involving single-domain environments, its capabilities were evaluated in this research to assess its functionality in multi-domain and cross-forest contexts.

3.3.3 Enum4linux-ng

Building on Enum4linux, Enum4linux-ng is a more recent fork designed for use with the SMB and SAMR protocols, implemented in Python and released under the GPL licence.[6] Similar to its predecessor, it gathers data on users, groups, shares, and domain policies. Enum4linux-ng extends the functionality by supporting both

[3]Microsoft. *Net user* documentation. Available: `https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc771865(v=ws.11)`

[4]Microsoft. *Net group* documentation. Available: `https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-server-2012-r2-and-2012/cc754051(v=ws.11)`

[5]GitHub. *Enum4linux repository.* Available: `https://github.com/CiscoCXSecurity/enum4linux`

[6]GitHub. *Enum4linux-ng repository.* Available: `https://github.com/cddmp/enum4linux-ng`

Kerberos and NTLM authentication, allowing for broader enumeration capabilities. Produces detailed reports on Active Directory structures by combining SAMR and LDAP protocols and requires appropriate permissions for specific SAMR queries. Its configuration is compatible across Linux-based environments, making it accessible for a range of deployments.

3.3.4 PowerShell

PowerShell provides a range of cmdlets for querying and managing Active Directory, which is included in the Active Directory module. This module, available as part of the Remote Server Administration Tools (RSAT) package on Windows systems, contains 42 cmdlets specifically designed to read or enumerate Active Directory objects such as users, computers, groups, domains, and domain controllers. [7]

Cmdlets such as `Get-ADUser`, `Get-ADGroup`, `Get-ADComputer`, `Get-ADDomain`, `Get-ADDomainController`, and `Get-ADGroupMember` are used to retrieve detailed information about Active Directory objects. The module's extensive coverage of object retrieval tasks makes it a key tool for both administrative and security-related operations in Active Directory environments.

As a native Windows tool, the PowerShell Active Directory module is utilised to automate directory management tasks. Given its capabilities for retrieving information about directory objects, it was selected for this research to analyse the protocols and methods employed by its cmdlets, including their potential interaction with protocols such as SAMR.

This research focuses specifically on the Active Directory module cmdlets and excludes methods like `Get-WmiObject` and `Invoke-WmiMethod`. Although these methods can access similar information, they fall outside the scope of the cmdlet-specific evaluation.

3.3.5 SharpHound

SharpHound is a data collector for Active Directory environments designed to collect data on user, group, and permission relationships to support graph-based analysis tools. Unlike tools limited to LDAP-based enumeration, SharpHound incorporates SAMR queries to retrieve user and group information directly from domain controllers, enabling deeper access to object relationships that may not be exposed via directory services alone. Distributed under the GPL licence, SharpHound includes extensive documentation through developer and community resources to facilitate deployment and use.[8]

[7] Microsoft. *PowerShell Cmdlets for Active Directory.* Available: `https://learn.microsoft.com/en-us/powershell/module/activedirectory`

[8] SharpHound GitHub repository. Available: `https://github.com/BloodHoundAD/SharpHound`

SharpHound supports multi-forest environments and can gather cross-domain data, enabling a detailed view of permissions and relationships across forests. Its integration of SAMR queries enhances its data collection capabilities, making it more versatile for security assessments. SharpHound is compatible with Windows and Linux systems, providing flexibility in deployment configurations.

3.3.6 Metasploit Framework

The Metasploit Framework includes SAMR enumeration modules within its library of penetration testing tools.[9] Developed as an open-source project with additional commercial licensing options, Metasploit offers community and official training documentation. The framework supports multi-forest enumeration and automated data gathering, with modules designed specifically for SAMR queries alongside other network protocols. Metasploit is built for deployment on Windows, macOS, and Linux platforms, offering configuration options to tailor SAMR-related queries across various environments.

3.3.7 CrackMapExec

CrackMapExec (CME) is a GNU GPL-licensed post-exploitation and enumeration framework designed for network reconnaissance in Active Directory environments. Implemented in Python, CME builds on the Impacket library, reusing its core protocol logic for SAMR and SMB interactions. The tool supports multi-forest deployments and is compatible with both Windows and Linux systems. Installation involves cloning the GitHub repository, installing Python dependencies, and setting up environment-specific options. CME provides dedicated commands for user, group, and session enumeration using SAMR operations, and includes additional modules for credential validation and lateral movement. Its Impacket-based architecture ensures protocol fidelity while extending functionality through a modular interface.

3.3.8 Impacket

Impacket is an open-source Python library released under a modified Apache 2.0 licence. Designed for low-level network protocol interaction, it provides tools for SAMR enumeration and Active Directory reconnaissance, including utilities such as `samrdump.py` and `net.py`.[10] While documentation is available, the use of Impacket tools typically requires technical proficiency due to their low-level interface and minimal abstraction.

[9] Rapid7. *Metasploit Framework*. Available: `https://www.metasploit.com`

[10] GitHub. *Impacket Library*. Available: `https://github.com/fortra/impacket`

Impacket is compatible with Windows, macOS, and Linux platforms, making it suitable for diverse network environments. Installation involves cloning the repository, resolving Python dependencies, and executing the desired modules from the command line or embedded scripts.

3.3.9 rpcclient

The `rpcclient` tool is a command-line utility within the Samba suite, designed to interact with Windows RPC services, specifically SAMR [11].Released under the GNU GPL licence, `rpcclient` enables users to query SAMR data directly through various command options. Using the tool's full capabilities requires technical expertise due to its interactive and manual operation mode, which limits automation and requires users to query domains individually, as it does not support enumeration by domain or across the forest. Primarily compatible with Linux and other Unix-based systems, `rpcclient` offers an alternative to SAMR interactions in non-Windows environments.

3.3.10 Summary of Tool Selection and Capabilities

The tools selected for SAMR enumeration represent a range of capabilities, covering different approaches to retrieve Active Directory data. The research aimed to identify tools that could operate in both single-domain and multi-forest environments while adhering to the SAMR's functionality. The prototype `samr-enum`, developed as part of this study, is included in the comparison tables to allow direct benchmarking against existing utilities; its quantitative metrics are taken from the tests presented in Chapter 4.

Tools such as `rpcclient` and Impacket were included due to their ability to perform SAMR queries, including cross-forest enumeration. Their inclusion highlights tools capable of leveraging the protocol for advanced enumeration tasks. In contrast, tools such as `net.exe`, although a standard component in Windows administrative utilities, were found to lack support for cross-forest SAMR queries. Despite this limitation, `net.exe` was included for its role in providing foundational insights into SAMR interactions in single-domain environments.

Specific tools were excluded due to limitations or security concerns. For example, RPC Investigator [12] , although capable of analysing Windows RPC services, was flagged by multiple anti-malware engines during a VirusTotal scan, raising concerns about its operational safety in production environments.

[11]Samba. *rpcclient documentation* Available: `https://www.samba.org/samba/docs/current/man-html/rpcclient.1.html`

[12]`https://github.com/trailofbits/RpcInvestigator`

The Nmap script `smb-enum`, released in 2010[13], was considered for inclusion. However, the script failed to perform the enumeration tasks successfully during the testing. The errors observed, which are detailed in the Appendix (Appendix B), indicated that the script could not negotiate a connection due to dependence on SMBv1, a protocol disabled in modern systems for security reasons. While further analysis of the script's behaviour could provide additional insights, it was not pursued as it failed to establish basic functionality in the research environment. This limitation rendered it incompatible with the research objectives.

This selection process highlights differences in tool capabilities and alignment with modern Active Directory configurations. Evaluation criteria, such as cross-forest support and access-level requirements, provided a structured framework to assess the functionality of each tool. These results form the foundation for identifying gaps in existing tools and exploring opportunities to develop more comprehensive solutions for SAMR enumeration.

3.4 Comparison Results

This section presents the findings of the comparative analysis of SAMR enumeration tools, structured around the evaluation criteria defined in Section 3.1. The evaluation emphasises the tools' ability to retrieve data from multi-forest Active Directory environments using SAMR. A summary of tool features and corresponding evaluation results is presented in Table 3 and Table 4, which also includes the prototype `samr-enum` developed in this research.

Table 3. Comparison of SAMR Enumeration Tools (Part 1)

Tool Name	**Cross-Forest Request Support**	**OpNum Coverage**	**Access Scope Compliance**
Net	Not Supported	Not Applicable	Not Applicable
Enum4linux	Not Supported	Not Applicable	Not Applicable
Enum4linux-ng	Not Supported	Not Applicable	Not Applicable
PowerShell	Supported	Not Applicable	Not Applicable
SharpHound	Supported	Low Coverage (17.3%)	Compliant
Metasploit	Supported	Low Coverage (26%)	Compliant
CrackMapExec	Supported	Low Coverage (26%)	Compliant
Impacket	Supported	Moderate Coverage (56.5%)	Compliant
rpcclient	Supported	High Coverage (82.6%)	Compliant
samr-enum	Supported	Moderate Coverage (65.2%)	Compliant

Several challenges were encountered during the analysis. First, SMB-layer communication captured during enumeration was encrypted, requiring decryption using Wireshark's built-in features. This process ensured that protocol-level interactions could be comprehensively analysed.

[13] `https://nmap.org/nsedoc/scripts/smb-enum-users.html`

Table 4. Comparison of SAMR Enumeration Tools (Part 2)

Tool Name	Data Parsing and Accuracy	Authentication Protocol Support	Access Level Requirements
Net	Not Applicable	Not Applicable	Not Applicable
Enum4linux	Not Applicable	Not Applicable	Not Applicable
Enum4linux-ng	Not Applicable	Not Applicable	Not Applicable
PowerShell	Not Applicable	Not Applicable	Not Applicable
SharpHound	Accurate	Multi-Authentication Compatible	Standard Access Sufficient
Metasploit	Accurate	NTLM	Standard Access Sufficient
CrackMapExec	Accurate	Multi-Authentication Compatible	Standard Access Sufficient
Impacket	Accurate	Multi-Authentication Compatible	Standard Access Sufficient
rpcclient	Accurate	Multi-Authentication Compatible	Standard Access Sufficient
samr-enum	Accurate	Multi-Authentication Compatible	Standard Access Sufficient

Second, the naming conventions used for SAMR fields and OpNums in Wireshark differed significantly from those specified in the SAMR documentation. To address this discrepancy, a mapping was created and published on the project's GitHub page, as referenced in Appendix B. The appendix also includes a correlation between field names in SAMR responses and their corresponding attributes in the Active Directory schema, as these differ significantly and require careful reconciliation for accurate evaluation.

Third, the tools varied in their documentation and transparency about SAMR usage. For example, tools such as Metasploit, CrackMapExec, and Impacket did not specify the exact subcommands or parameters that employed SAMR requests. This required extensive testing of different commands and configurations to isolate SAMR-related functionality. Furthermore, some tools, such as SharpHound and CrackMapExec, used a combination of protocols, including LDAP and Microsoft Network Logon, along with SAMR in their operations. This required a detailed analysis to separate data explicitly related to SAMR, ensuring that the results accurately reflected SAMR-based enumeration capabilities. Identifying the commands and configurations that used SAMR requests was necessary for these tools to focus the evaluation on the protocol of interest.

PowerShell cmdlets from the Active Directory module were also examined. The research found that none of the evaluated cmdlets used SAMR for their operations. Instead, these cmdlets relied on higher-level APIs and protocols, such as Microsoft .NET Naming Service (MS-NNS) and Microsoft .NET Message Framing Protocol (MS-NMF). This abstraction from SAMR reduces direct interaction with lower-level protocols, aligning with security practices by minimising exposure to vulnerabilities inherent in SAMR. Although PowerShell provides extensive enumeration capabilities for Active Directory objects such as users, computers, and groups, its exclusion of SAMR-based operations underscores the design's focus on administrative tasks rather than low-level protocol interactions.

The analysis revealed discrepancies in the naming conventions between Wire-

shark and the SAMR documentation. For example, Wireshark labels the Access Mask `0x00000200` as `USER ACCESS GET ATTRIBUTES`, while the protocol specifies it under `DOMAIN_LOOKUP`. Similarly, OpNums in Wireshark, such as `EnumDomains`, correspond to `SamrEnumerateDomainsInSamServer` in the protocol.

To address these inconsistencies, this study mapped OpNums using their numeric identifiers and correlated Access Mask values by identifying matching byte patterns in traffic captures. This mapping process ensured an accurate interpretation of tool outputs against the protocol specifications and underscored the challenges of aligning tool-specific representations with standardised documentation.

3.4.1 Cross-Forest Support Criterion

The evaluation of cross-forest support assessed each tool's ability to perform SAMR enumeration across domains within a forest trust. The testing environment included two one-way forest trusts configured between `domain-y`, `domain-z`, and `domain-x`, with enumeration attempts that target remote domain controllers such as `zdc1.domain-z` and `xdc1.domain-x`. Tools that did not support cross-forest requests were tested only within the same domain, targeting `ydc1.domain-y`.

Several tools, including `rpcclient`, Impacket, SharpHound, Metasploit, and CrackMapExec, demonstrated the ability to retrieve data across forest boundaries using SAMR requests. This functionality aligns with the study's focus on multi-forest environments and highlights the effectiveness of these tools to leverage SAMR on trusted relationships. In contrast, tools such as `net.exe` and Enum4linux were limited to single-domain operations and could not list objects in trusted forests.

Although Enum4linux-ng internally utilises `rpcclient` for its operations, it does not expose functionality or parameters for cross-forest enumeration. This limitation prevents it from leveraging the cross-forest support inherent in `rpcclient`. As a result, Enum4linux-ng is similarly restricted to single-domain operations.

PowerShell cmdlets from the Active Directory module supported cross-forest environments but relied on higher-level APIs such as Microsoft.NET Naming Service (MS-NNS) rather than SAMR. Although these cmdlets offer extensive enumeration capabilities, their abstraction from SAMR reduces their relevance for protocol-specific evaluations.

Tools that do not support cross-forest enumeration, such as `net.exe`, Enum4linux, and Enum4linux-ng, were excluded from the analysis according to other criteria. This decision ensures that the evaluation is focused on tools that align with the study objectives of evaluating SAMR functionality in multi-forest environments.

These findings emphasise the importance of cross-forest support for comprehensive Active Directory reconnaissance and highlight functional gaps in tools that lack this capability. The ability to traverse forest trusts using SAMR requests remains a critical feature for addressing the complexity of multi-forest environments.

3.4.2 OpNum Coverage Criterion

The evaluation of the OpNum coverage criterion aimed to determine the breadth and relevance of supported SAMR operations (OpNums) by each tool, focusing on their ability to retrieve Active Directory data in alignment with the SAMR specifications. The findings highlight notable differences in OpNum coverage, data retrieval accuracy, and relevance to enumeration and security tasks. For a complete breakdown of OpNum coverage by each tool, refer to Appendix E.

OpNum Analysis and Tool Capabilities

Tools such as `rpcclient` and Impacket supported various OpNums, including key enumeration operations. SharpHound, in contrast, relied only on a few SAMR OpNums, such as `SamrEnumerateUsersInDomain` (OpNum 13), supplementing its output with LDAP queries. Although this approach ensured detailed results, it underscores the importance of segregating SAMR-derived data from LDAP responses to maintain accuracy in security and administrative analyses.

Metasploit modules exhibited limited OpNum usage, restricting their ability to support bulk enumeration tasks. These differences in OpNum coverage emphasise the varying utility of these tools in specific enumeration scenarios. In particular, discrepancies were observed in the data retrieved using certain OpNums. For example, `SamrQueryInformationDomain` (OpNum 8) returned more user data (30 000 records) than retrieved via `Get-ADUser` cmdlets (20 000 records). The scope of this research did not include a further analysis of the disparity. However, it highlights the potential for SAMR operations to retrieve additional data, such as deleted objects or other hidden records. This observation underscores the need for further investigation into the behaviour of SAMR and its implications for data security and integrity.

Critical OpNums for Enumeration and Security

Certain OpNums were identified as essential for their role in the enumeration and security assessments. These include:

- `SamrEnumerateDomainsInSamServer` (OpNum 6): Serves as the foundation for discovering domain information within a SAM server, an essential step in understanding the overall structure of an Active Directory environment.
- `SamrEnumerateUsersInDomain` (OpNum 13) and `SamrEnumerateGroupsInDomain` (OpNum 11): Retrieval of user and group data within a domain helps identify potential targets for privilege escalation or further reconnaissance.
- `SamrQueryInformationUser` (OpNum 36): Describes user-specific attributes,

such as logon hours and account status, which are critical for administrative assessments and potential abuse in targeted attacks.

- `SamrValidatePasswordPolicy` (OpNum 57): Validates passwords against domain policies, useful for identifying weak password configurations that could be exploited in attacks.

- `SamrQueryDisplayInformation` (OpNum 40): Enables enumeration of accounts in a structured and filtered manner, valuable for scaling attacks in environments with numerous users and groups.

These OpNums provide essential data for reconnaissance and attack planning, particularly in multi-forest environments.

Multi-Forest Context and Additional Considerations

In multi-forest environments, OpNums facilitating cross-domain enumeration are particularly significant:

- `SamrEnumerateDomainsInSamServer` (OpNum 6), `SamrEnumerateUsersInDomain` (OpNum 13), and `SamrEnumerateGroupsInDomain` (OpNum 11) collectively provide foundational data to enumerate domains, users, and groups in a foreign forest.

- `SamrValidatePasswordPolicy` (OpNum 57) becomes critical to evaluating password policies in trusted forests, contributing to a comprehensive review of security posture.

- `SamrQueryDisplayInformation` (OpNum 40) is invaluable for handling environments with large user and group populations, particularly where network latency or policy restrictions affect enumeration efficiency.

These OpNums align with the goals of comprehensive enumeration while supporting lateral movement strategies in multi-forest environments. The retrieval of hidden or additional data via certain OpNums, such as `SamrQueryInformationDomain`, highlights the need for tools to delineate data boundaries and ensure careful compliance with security policies.

Findings and Recommendations

Tools supporting key OpNums provide a robust foundation for Active Directory reconnaissance and security assessments. Recommendations for tool development include the following:

- Prioritizing support for enumeration-related OpNums while ensuring adherence to read-only permissions.

- Enhance clarity in output by distinguishing data retrieved via SAMR from other protocols, such as LDAP.
- Investigating the implications of additional data retrieved by certain OpNums, such as `SamrQueryInformationDomain`, to understand their potential to expose deleted or hidden objects.

This structured approach improves the applicability of a tool's in system administration and auditing tasks, aligning with the revised focus of the thesis.

3.4.3 Access Scope Compliance Criterion

The analysis of five SAMR enumeration tools - Impacket, CrackMapExec, Metasploit, SharpHound, and Samba's `rpcclient` – indicates that all operate within the limits of the SAMR specification. Each tool establishes RPC connections to the Security Account Manager (SAM) and requests a handle with specific access rights by providing a *DesiredAccess* mask in the SAMR calls. This is in line with the design of the protocol: for example, the MS-SAMR documentation for `SamrOpenUser` explicitly states that the client provides an `ACCESS_MASK` to indicate the requested access for the handle. In practice, CrackMapExec, Impacket, `samrdump.py` and `rpcclient` set this mask to the special value `MAXIMUM_ALLOWED` (0x02000000) when opening SAMR *server*, *domain*, *user*, *groups* and *aliases* objects. According to the MS-SAMR specification, the `MAXIMUM_ALLOWED` flag tells the server to grant the highest level of permissions that the caller is entitled to on the target object. This usage is *explicitly permitted* according to the protocol. In fact, when a client's *DesiredAccess* includes `MAXIMUM_ALLOWED`, the server will grant a handle with the access rights the client's security context authorises for that object. This behaviour contrasts with a minimal or specific *DesiredAccess* request: if a client asks for particular rights and any requested right is not actually allowed to that client, the server will reject the request with an `STATUS_ACCESS_DENIED` error. By using `MAXIMUM_ALLOWED`, these tools avoid such denials – they obtain a handle with all privileges that the connecting user is entitled to and no more.

None of the tools exceeded or violated the SAMR allowances.
The `MAXIMUM_ALLOWED` is a valid flag defined in the protocol, and the server enforces the actual permissions internally. For instance, Samba's `rpcclient` tool uses `MAXIMUM_ALLOWED` in its `SamrConnect` and `SamrOpenDomain` calls by default, meaning it asks for the broadest access, but ultimately receives only what the user can legitimately have. Impacket and CrackMapExec exhibit the same pattern, as does SharpHound's SAMR-based enumeration module. All of these tools adhere to the protocol's rules for handle acquisition; the difference lies in the scope of access they request, not in the exploitation of unauthorised rights.

From a security perspective, this behaviour reflects a deviation from the principle of least privilege. Ideally, an enumeration client would request only the minimal permissions necessary for the task (for example, only the read or list access required to enumerate users or groups) rather than requesting every possible permission. However, in our observations, the tools favoured convenience and completeness by using `MAXIMUM_ALLOWED` to ensure that all needed rights are obtained in one request. This approach is functionally correct, but grants the tool a broader access scope than is needed for simple queries. As a detection criterion, therefore, a SAMR client that consistently requests broad *maximum allowed* access on SAM objects can be flagged as exhibiting excessive permission usage. In other words, the tool does not limit itself to the least privileges required for enumeration, which is unusual for well-behaved system components. While this practice does not break any protocol rules, it is a distinguishable pattern: a legitimate query may request only specific enumerate permissions, whereas these enumeration tools uniformly request the maximum allowed access. Identifying this pattern in SAMR traffic can thus serve as an indicator of potential enumeration activity by third-party tools, since the client is deliberately asking for all possible rights (a sign of an aggressive or generic enumeration technique) instead of adhering to a minimal privilege approach. In summary, the detection criterion for *access scope compliance* is met when a SAMR client's access masks are unscoped (e.g., using `MAXIMUM_ALLOWED` for handle requests) despite the fact that only read or list permissions are needed. All tested tools triggered this criterion by design, underscoring that they operate within protocol specifications but not optimised for least-privilege usage, although within protocol compliance. This insight refines our understanding: the behaviour of tools is an allowable use of SAMR, not an exploitation of a protocol flaw, and any detection mechanism must treat it as a heuristic sign of the usage of non-minimal privileges rather than a policy violation.

3.4.4 Data Parsing and Accuracy Criterion

The evaluation of this criterion aimed to determine how well each tool parsed and displayed SAMR responses while ensuring consistency with protocol specifications. The parsing accuracy was assessed through tests on multilingual and symbol-rich entries to simulate diverse production environments.

Most tools displayed parsed data accurately without errors, ensuring readability in different languages and character sets. However, the outputs were not always complete. For example, while SAMR responses contained all fields, tools such as Metasploit provided minimal details in their output, such as essential attributes, limiting their utility in scenarios requiring detailed enumeration.

Certain tools, such as SharpHound and CrackMapExec, leveraged multiple protocols, including LDAP, alongside SAMR. This dual-protocol approach required

additional analysis to isolate SAMR-derived data from LDAP responses, complicating the evaluation of SAMR-specific functionality. Despite this, such tools offered comprehensive summaries that integrated information from multiple sources, enhancing their utility in broader reconnaissance tasks.

Some tools presented raw or unparsed data directly from SAMR responses. For example, Impacket's `net.py` displayed the `Logon Hours Allowed` field in its raw hexadecimal format (`00003f00003f00003f00003f00003f00003f00003f`). Although this output is suitable for programmatic processing, it is less interpretable for manual analysis, highlighting the need for tools to provide user-friendly output alongside raw data.

Furthermore, `rpcclient` provided data in human-readable and hexadecimal formats for fields such as user group memberships and security descriptors. This dual representation is valuable for low-level protocol analysis, but may overwhelm users seeking only high-level information.

Although most tools supported user and group enumeration, only a subset provided explicit support for computer enumeration. For example, `rpcclient` focused primarily on users and groups, with no apparent support to enumerate computer accounts. In contrast, tools like Metasploit required specific modules for user and computer enumeration, limiting their ability to enumerate all object types in a single operation comprehensively. CrackMapExec demonstrated broader capabilities, supporting the enumeration of users, groups, and computers within its SAMR and SMB functionalities.

Enumeration capabilities also varied significantly. Although most tools supported bulk enumeration, Metasploit modules were limited to targeted queries for specific users, groups, or computers. This restricted scope limits its applicability in scenarios that require extensive data collection, such as analysing all domain objects.

These findings emphasise the need for tools to balance parsing accuracy, completeness, and user-friendly output. Recommendations for improvement include enhancing field interpretation, providing more precise distinctions between data retrieved via SAMR and other protocols, and enabling detailed and summarized outputs to cater to diverse user requirements.

3.4.5 Authentication Protocol Support Criterion

The authentication protocol support criterion evaluates whether tools can authenticate using NTLM, Kerberos, or both protocols. These authentication methods are essential for interacting with domain controllers in modern Active Directory environments, with NTLM often used for legacy compatibility and Kerberos preferred for its enhanced security and efficiency.

The analysis revealed that most tools supporting cross-forest SAMR requests

supported NTLM and Kerberos authentication methods. For example, `rpcclient` and Impacket demonstrated compatibility with both protocols, providing flexibility in environments where either authentication method might be required. Similarly, tools like CrackMapExec and SharpHound leveraged multi-authentication compatibility to enhance their enumeration capabilities across trusted forests.
In contrast, the Metasploit modules that interact with SAMR are based on NTLM authentication. Although the Metasploit framework broadly supports Kerberos for other operations, such as service ticket authentication and exploitation modules, its SAMR-related modules, such as `"auxiliary/scanner/smb/smb_enumusers"` and `"auxiliary/admin/dcerpc/samr_account"` do not explicitly document support for Kerberos. This limitation highlights the need for practical testing or source code verification to confirm Kerberos compatibility within specific modules.

These findings highlight the importance of multi-authentication compatibility in enabling SAMR enumeration across diverse Active Directory configurations. The tools supporting NTLM and Kerberos demonstrate enhanced adaptability and alignment with modern security practices. In contrast, tools restricted to NTLM may face compatibility issues in environments that enforce Kerberos-only authentication, whether due to domain controller configuration or organisational security policy.

3.4.6 Access Level Requirements Criterion

The access level requirements criterion evaluated whether tools required standard or elevated privileges to perform SAMR enumeration tasks. Testing was conducted using non-privileged user accounts in both the domain where the user resided and the domain being enumerated. These accounts did not have special permissions or group memberships beyond those assigned by default.

As seen in the results table, none of the tools required elevated privileges on the local system to perform SAMR enumeration tasks. Tools such as SharpHound, CrackMapExec, Impacket, and `rpcclient` successfully retrieved SAMR data fields using standard user permissions, adhering to the principle of least privilege. These tools demonstrated the ability to operate without administrative privileges, making them suitable for environments with strict access control policies.

However, while these tools do not require privileged access to execute SAMR-related operations, some may require administrative privileges for installation. For instance, tools such as CrackMapExec and Metasploit often require elevated access during setup to install dependencies or configure the environment. Although it does not directly impact their run-time permissions, this requirement could limit their usability in environments with strict installation policies.

These findings emphasise the importance of access-level requirements in evaluating tool effectiveness. Tools that operate with standard user permissions align more

closely with security best practices, particularly in environments prioritizing the principle of least privilege. Although installation privileges are outside the primary focus of this criterion, they remain a logistical consideration when deploying these tools.

3.5 Limitations of the Comparison

Although the research provides valuable information on the capabilities and limitations of existing SAMR enumeration tools, several constraints in the experimental setup and evaluation process introduce limitations that should be considered:

- **IPv4-Only Configuration:** The experimental environment used only IPv4 networking, excluding scenarios that involve IPv6 or dual stack configurations. This may limit the applicability of the findings in environments where IPv6 is deployed.
- **Default Domain Schema:** The Active Directory schema was maintained in its default state without modifications. Custom schema extensions and non-standard configurations, which are prevalent in certain enterprise environments, were not evaluated.
- **Data Population Constraints:** To test tool capabilities with long names and special characters, domain data were populated using Active Directory Users and Computers (ADUC). As ADUC imposes character limits shorter than the maximums defined in the schema, the tests may not fully reflect real-world constraints.
- **Protocol Compliance of Servers:** The research did not validate whether domain controllers strictly adhered to the SAMR specification when responding to tool requests. For example, whether the servers returned all requested data fields or behaved correctly in edge cases remains undetermined.
- **No Policy Adjustments:** Default server policies were used throughout the evaluation. Modifications such as enabling SMBv1 for compatibility, enforcing specific protocols (e.g., SMBv3), or implementing security hardening measures were not considered. These factors could affect the performance and functionality of the tools in real-world deployments.

These limitations underline the scope of the research and suggest areas for further exploration in future studies. Addressing these constraints could provide a more comprehensive understanding of SAMR enumeration tools and their behaviour in diverse configurations.

4 SAMR Enumeration Tool

Introduction. Enumerating directory information in Active Directory is particularly challenging in complex multi-forest domains where multiple domains are linked by trust relationships. Existing enumeration methods frequently struggle in such environments, revealing structural shortcomings that impede a complete and reliable audit of accounts and configurations. In practice, many tools produce inconsistent representations of the same data across different domains and often lack comprehensive support for the full range of enumeration paths needed to capture all relevant details. These issues are compounded when traversing trust relationships: cross-domain queries may fail or return partial results, especially under restrictive permission settings or selective authentication between forests. The net effect is that security analysts and administrators are left with an incomplete and fragmented view of the directory in multi-forest scenarios, highlighting the need for a more systematic solution.

The `samr-enum` tool was developed to address these gaps and limitations. It is designed to eliminate ambiguities in the output by presenting the enumerated data in a consistent, structured manner, thus avoiding the misinterpretations caused by earlier tools. Crucially, the tool operates effectively across domain and forest boundaries, mitigating the operational hurdles that hinder existing approaches in trust-linked environments. In summary, `samr-enum` directly tackles the core problems of inconsistency, incompleteness, and cross-domain limitations in Active Directory enumeration, providing a clear and factual basis for the detailed techniques discussed in the following chapter.

4.1 Design of the Tool

The `samr-enum` tool was designed with a clear architectural structure to maximise both the coverage of SAMR and the maintainability of the code. At a high level, the tool workflow is organised into distinct phases: connection establishment, enumeration actions, and output handling. In the connection phase, the tool establishes a session to the SAMR interface on a target host. We chose to build this on top of the Impacket library, which provides a Python implementation of the MSRPC protocols (including SAMR) [21]. The modular design of Impacket, offering high-level classes for common operations, greatly facilitated our development, as it allowed the tool to interface with SAMR using tested library calls rather than crafting raw packets from scratch. This approach takes advantage of Impacket's strengths (it is widely used by security researchers to create custom tools [21]) and ensures compliance with the low-level protocol requirements without re-implementing them. Architecturally, the tool uses Impacket's DCERPC `samr` bindings to perform calls such as `SamrConnect`, `SamrOpenDomain` and various enumeration functions. Each of

these calls is abstracted into functions within our tool, forming a logical sequence that mirrors the steps a manual enumerator or script would take. For example, upon connecting to the SAMR service of a target server, the tool enumerates the available domains (usually the built-in domain and the primary domain on a domain controller), then opens the target domain and, in turn, enumerates users, groups, and aliases. This process is illustrated in Figure 3; see Appendix F for the complete SAMR session workflow with all RPC calls. The structured sequence is implemented in a modular way, so each step can be individually debugged or extended.

We selected Python as the implementation language for several reasons. First, Python's simplicity and readability are advantageous for an academic and security context – it allows rapid prototyping and straightforward integration of libraries such as Impacket. Second, using Python aligns with Impacket itself (which is written in Python), ensuring seamless integration. The choice of Python also makes the tool cross-platform by nature; it can run on any system with a Python 3 interpreter, which is ideal for flexibility (e.g., a security engineer might run it from a Kali Linux or directly on a Windows host with minimal changes). Furthermore, Python's rich ecosystem for data handling (JSON, CSV libraries, etc.) facilitates the multiformat output feature of the tool.

A crucial aspect of the design was the correct implementation of desired access rights for each SAMR operation in accordance with the Microsoft SAMR specification. SAMR requires that when a handle is opened (for a domain, user, group, etc.), the requester specifies an access mask indicating what operations it intends to perform. If this mask is too limited, subsequent queries may fail to retrieve needed info; if it is too broad, the server may reject the request with `STATUS_ACCESS_DENIED`. The tool carefully sets the appropriate access flags for each call. For example, when opening a domain handle for enumeration, it requests the minimal rights necessary to enumerate users and groups. When opening a user handle to query detailed information, it requests user-level read access. This approach was guided directly by the protocol documentation to avoid errors. According to the MS-SAMR specification, if a client requests access bits that are not granted, the server *must* return an access denied error [22]. Thus, if a tool erroneously uses an administrator-level access mask for a normal query, an unprivileged session will fail even if a lesser access could have succeeded. Such implementation errors are avoided in the design of `samr-enum`. In contrast to tools such as `rpcclient` (which, as noted, perform a `LsaOpenPolicy` with broad access rights, even for null sessions), `samr-enum` applies a minimal access approach by using a sequence of SAMR calls with access masks tailored to the specific requirements of each operation. The design philosophy here is to request only the permissions required for each operation, which maximises the likelihood of success for accounts with limited rights. This aligns with the principle

of *least privilege* and improves compatibility with environments that enforce strict access controls, such as those encountered in audits or vulnerability assessments.

Security features are built into the tool design from the ground up. A notable design choice was to prioritise encrypted communications with the target whenever supported. SAMR runs on top of SMB named pipes (or DCERPC over SMB), which supports both signing and encryption. By default, the tool requests at least SMB signing to ensure message integrity and enables SMB encryption when available to protect the SAMR-named pipe communication. If SMB encryption is successfully negotiated (e.g., via SMB 3.x), SAMR traffic, including sensitive enumeration data, is protected in transit and not sent in plaintext. Additionally, Impacket can enforce encryption at the RPC layer using security contexts (e.g., `RPC_C_AUTHN_LEVEL_PKT_PRIVACY`), which encrypts the RPC payload itself using session keys derived from NTLM or Kerberos authentication. In practical terms, a flag in the DCERPC binding (equivalent to the `seal` option in Samba's `rpcclient`) is used to ensure that all SAMR calls are encrypted on the wire. Alongside encryption, the tool uses standard SMB authentication channels, which means that it benefits from any security that SMB provides (for instance, if the domain requires SMB signing, the tool will comply).

Finally, the design includes a focus on usability and safety. Usability is addressed by providing clear command-line options and help messages (the tool has a built-in `help` page detailing usage). Safety is addressed by measures like input validation (ensuring, for example, that the domain name or IP address provided is in a valid format, to avoid unintended behaviour) and by not performing any write operations via SAMR – the tool is strictly read-only in its use of the protocol, thus eliminating the risk of accidental changes on the target.

4.2 Implementation

The implementation of `samr-enum` closely follows the design principles defined earlier in this chapter, translating them into a functional tool. The tool is implemented in Python 3 and is based primarily on Impacket 0.12. Impacket 0.12 was chosen because this version includes up-to-date support for the required SAMR functions and provides full Kerberos authentication support. In setting up the development environment, certain dependencies of the system were required. For instance, on a Linux system, installing Impacket's DCERPC and Kerberos features may require Python development headers and Kerberos libraries. Packages such as `python3-dev` and `libkrb5-dev` must be installed prior to Impacket. This is because Impacket (and related libraries such as `pyasn1` or `gssapi`) can compile portions of code or need C headers for encryption and Kerberos functionality [23]. Ensuring these dependencies are met is part of the tool implementation instructions.

In terms of code structure, the tool is organised into modules corresponding

to functional areas: argument parsing and setup, connection and authentication, enumeration logic, and output formatting. We utilised Python's argparse library to handle command-line options, allowing users to specify credentials, target specification, and output file paths, among other options. The authentication mechanisms supported by the tool include both Kerberos and NTLM. For Kerberos, the tool can take advantage of Impacket's ability to use existing Kerberos tickets (for example, if the user sets the KRB5CCNAME environment variable or specifies `auth=kerberos` for Kerberos authentication, the tool will attempt to get a TGT and use Kerberos AP-REQ for the SAMR connection). If Kerberos is not desired or is not possible, the tool goes back to NTLM authentication using the provided credentials. This dual support was straightforward to implement via Impacket's high-level API, which abstracts the differences – essentially, one can specify a flag for Kerberos, and Impacket handles the lower-level details. This approach is similar to how Samba's `rpcclient` allows choosing the authentication method [24].

We validated the tool on both Linux and Windows platforms to ensure cross-platform compatibility. On Linux (Ubuntu 22.04.5 LTS), the tool is executed directly with Python once the required dependencies have been installed. On Windows, we tested running the tool with a native Python installation. Minor adjustments (such as command-line argument parsing and handling of shell quoting) were confirmed to work properly on Windows. The ability to run on Windows means that a security auditor could execute the enumeration from a domain-joined machine (using local Python) if needed, which could be useful in certain internal assessment scenarios.

For testing and development, a laboratory environment was established. For a complete description of the laboratory environment, refer to Appendix B. This lab included multiple Windows Server instances configured in a multi-domain, multi-forest layout. Specifically, we created three separate Active Directory forests, *Forest X*, *Forest Y*, and *Forest Z* to simulate a multi-forest enterprise environment. *Forest X* contains numerous user, computer, and group objects to determine whether enumeration tools can retrieve complete data from servers. *Forest Z* stores users, groups, and computers with non-standard attribute values, including multiple languages, extended lengths, and special characters. *Forest Y* serves as the source of the user account that authenticates with the domain controllers in the other forests and performs the enumeration tasks. Each forest is set at the Windows Server 2016 forest functional level and configured with forest-wide trusts, enabling cross-forest testing of SAMR enumeration. In this environment, the test setup included various user accounts with different privilege levels, including domain administrators, regular domain users, and some test accounts with no privileges in the opposite forest. Connecting through Kerberos across a forest trust, one must specify the service principal name (SPN) of the target domain controller

in the other forest; the implementation of our tool automatically constructs the appropriate SPN for the SAMR service (usually of the form `cifs/` for SMB). Impacket automatically handles Kerberos cross-domain referrals internally as long as the correct target principal is used; accordingly, our code ensures that the target domain is correctly identified when using Kerberos authentication.

Debugging and error handling were critical implementation considerations, given the complexity of DCE/RPC calls and the variety of environments. The tool provides a verbose logging mode (`debug`) that outputs detailed information for each step, including the executed SAMR call and the returned result or error code. This was particularly useful during development – for instance, if a call failed with STATUS_ACCESS_DENIED, the debug output would show exactly at which function this occurred, allowing us to quickly adjust the access flags or call order if we found a logic mistake. We also used `try-except` blocks around the main segments of the enumeration process to catch unexpected exceptions (such as network timeouts or parsing issues). In case of an exception, the tool will report the error and attempt to continue (if it is not fatal) or cleanly close the SAMR session and exit. In particular, the tool is designed to handle partial data gracefully. For example, if enumerating user accounts succeeds but querying detailed information for a particular user fails (perhaps due to that user having an unconventional attribute that triggers an Impacket parsing quirk), the tool will log the error for that user and continue with others, ensuring the overall enumeration is not derailed. This thorough error handling increases the reliability of the tool in real-world use.

Finally, the implementation provides multiple output formats to facilitate analysis. By default, key results are printed on the console in a human-readable form (grouped by categories like *Users*, *Groups*, etc.). Users may also specify output files: a plain text report, a CSV file, or a JSON file. CSV and JSON export use Python's built-in `csv` and `json` modules. The CSV output arranges the data into rows (for example, one row per user with attributes like *RID*, *username*, *disabled status*). The JSON output uses dictionaries (e.g., a dictionary of users, each with its properties as subfields). This format is useful for programmatic analysis; scripts can ingest it to cross-reference known vulnerable accounts or populate audit reports. These output functions do not interfere with the enumeration logic, as they operate on collected data. This separation of concerns makes it easier to extend the output options in the future (e.g., XML support could be added without altering the core enumeration).

4.3 Tool Features and Capabilities

Microsoft's documentation notes that SAMR allows even low-privileged users to list accounts, groups, and group memberships from the local SAM database or Active Directory [25]. To take advantage of this for multi-domain and forest reconnaissance, `samr-enum` implements a broad set of SAMR OpNums that are not consistently supported across existing tools. For example, it invokes methods such as `SamrEnumerateUsersInDomain` (OpNum13) to list all user accounts in a domain[26], `SamrEnumerateGroupsInDomain` (OpNum11) for global groups, and `SamrEnumerateAliasesInDomain` (OpNum15) for local groups (aliases)[27]. Although these operations are partially implemented in some tools, `samr-enum` integrates them into a unified workflow that simplifies and consolidates enumeration. This capability is especially impactful in multi-forest environments, where domain local groups can contain users from other trusted domains or forests (stored as foreign security principals)[28]. Enumerating alias memberships through SAMR reveals those external SIDs, ensuring that accounts crossing forest boundaries are not overlooked. In contrast, many tools either omit these functions or require multiple manual steps to invoke them. For instance, Impacket's `samrdump` enumerates domain users and basic attributes, but lacks explicit support for group membership enumeration [29]. Likewise, Samba's `rpcclient` utility supports these operations through manual commands (e.g. `enumdomusers`, `enumdomgroups`) [24], but requires the operator to run them separately. By integrating a wide range of enumeration-relevant OpNums into a single automated process, `samr-enum` offers a consolidated view of domain structure—an approach aligned with the thesis objective of supporting comprehensive enumeration in trust-linked Active Directory environments.

Although multiple enumeration tools were evaluated in Section 3, the Table 6 focuses on `rpcclient` (from Samba) and Impacket's `samrdump` for specific reasons. First, `rpcclient` exhibits broad OpNum coverage and command-based control over SAMR functions, making it a relevant benchmark for protocol-level capabilities. Second, `samrdump` relies on the same Impacket library used by `samr-enum`, which allows for a direct comparison of how different client implementations handle the same underlying RPC calls. By contrasting `samr-enum` with these two tools, the table highlights differences in the completeness of the data, the authentication support, and the behaviour of the transport layer relevant to the multi-forest enumeration. Key aspects include supported SAMR operations, the ability to list foreign security principals, and how each tool manages encrypted SMB sessions. The observed differences underscore how `samr-enum` addresses gaps that exist in other enumeration utilities. For example, the table shows that `rpcclient` offers broad functionality, but relies on manual commands, which can leave parts of a domain unqueried unless the operator executes each relevant step. Meanwhile, `samrdump`

automates basic user enumeration but does not enumerate group memberships by default, which may lead to incomplete results in multi-domain or multi-forest scenarios. In contrast, `samr-enum` combines a wider set of SAMR calls into a single process, capturing user accounts, groups, and membership relationships with minimal user interaction. This integrated approach is especially valuable in complex Active Directory topologies where trust relationships span multiple forests. In such environments, the absence of even a single group or alias could hinder a comprehensive security assessment. Although six enumeration-related OpNums are not invoked explicitly, their output forms a strict subset of the data returned by calls already implemented. Based on this equivalence, the effective enumeration coverage increases to 21 out of 23 operations ($\approx 91\,\%$ of the SAMR enumeration surface). Section 4.4 lists each omitted procedure and identifies its functional equivalent.

4.4 Testing and Validation

Test Credentials and Trust Configuration

The testing was carried out using a standard (non-administrative) domain user account. Specifically, an account of `domain-y` was used to perform SAMR queries against two other domains (`domain-z` and `domain-x`) in a multi-forest environment. This account had no elevated privileges, ensuring that the results reflect what an ordinary domain user can enumerate across trust boundaries. Active Directory forests were connected by regular two-way forest trusts configured with forest-wide authentication. In other words, each trust allowed all authenticated users from the trusted forest to be recognised in the trusting forest without any per-user access filtering. This setup mirrors a common enterprise trust model and avoids any special access that selective authentication would require, thereby testing the tool under typical cross-forest conditions.

Network Setup and Conditions

All tests were performed on an isolated lab network that covered multiple subnets, with the machines of each domain residing on a dedicated IP subnet. For example, the workstation of the `domain-y` user was on a separate VLAN from the target `domain-x` and `domain-z` domain controllers, exercising the tool across routed network segments. A routing server connected these subnets and facilitated communication between the forests. Network firewalls and packet filters were not placed between subnets, and host-based firewalls were disabled on domain controllers and workstations to eliminate interference. This ensured that SAMR traffic could flow unimpeded and that any failures or limitations in the enumeration

would be attributable to permission or protocol constraints rather than network blocking. Detailed topology information (including IP addressing and VLAN layout) is available in the GitHub repository linked in the Appendix B.

Verification via Network Traffic Analysis

To validate the precision of the `samr-enum` results, each enumeration operation was cross-verified using packet capture analysis. During the operation of the tool, the network traffic was captured with Wireshark on the relevant network interfaces, recording the raw requests and responses. The tool was executed with its option `opnums` to display the SAMR OpNums enabled. Thus, the output included the specific OpNum identifier for each SAMR call made. These OpNum values reported by `samr-enum` were systematically compared with the OpNums observed in the Wireshark trace of actual network traffic. This one-to-one comparison confirmed that the sequence of SAMR calls (for example, `SamrConnect`, `SamrOpenDomain`, `SamrEnumerateUsers`, etc.) and their OpNums in the tool output perfectly matched the RPC operations seen on the wire. The correlation of the tool log with the Wireshark capture verified that no extraneous or missing calls occurred and that the tool interpretation of the protocol was correct. This step also confirmed that the data displayed by the tool for each operation corresponded exactly to the actual responses from the domain controller. In effect, the tool accurately reported the directory information returned over SAMR.

Edge-Case Testing for Special Characters and Unicode Handling

The testing regime also covered directory entries with edge-case naming and attribute scenarios to ensure correct handling of all data types. User and group objects containing special characters in their names (e.g., punctuation or non-alphanumeric symbols) were included in the target domains. This verified that `samr-enum` could retrieve and display such entries without error. Likewise, accounts with unusually long names and those with Unicode characters (for example, names in non-English scripts or with accented characters) were part of the dataset. The tool correctly enumerated these objects and their attributes, demonstrating proper encoding and formatting in the output. No truncation or misinterpretation of multibyte characters was observed. This result indicates that the implementation handles Unicode and long strings in Active Directory attributes as expected. This capability is crucial for use in diverse international or multi-national enterprise environments. The comprehensive synthetic data set (populated through tools such as BadBlood) ensured that these edge cases were present in the directory. This allowed their handling to be validated as part of the overall testing process.

4.5 Limitations and Future Enhancements

IPv4-Centric Testing and IPv6 Considerations

Although `samr-enum` provides substantial improvements over existing tools, its limitations and potential avenues for future enhancement must also be recognized. One known limitation is that the current version has been tested primarily in IPv4 networks. Modern environments could use IPv6, and while in theory Impacket and SMB support IPv6 addressing, we have not verified the tool's behaviour over IPv6. This means that there may be unanticipated issues if one tried to target an IPv6 address (for example, address parsing or network library differences). In practice, most Active Directory environments still support IPv4, so this limitation did not hinder our testing, but it remains an area to expand in the future: ensuring full dual-stack support would make the tool more resilient for any network scenario.

Dependence on Impacket's Protocol Support

Another limitation lies in the reliance on the Impacket library and the alignment of documentation. Impacket is a library used in the research, but it is a reimplementation based on observed behaviour and official documentation; minor discrepancies occur at times. For instance, Microsoft documentation might describe a field or a specific RPC call that Impacket has not implemented or has implemented differently. This consideration was explicitly factored into the development process. One specific example is the use of `SamrConnect5`, a newer call introduced in later Windows versions. Impacket 0.12, at the time of writing, does not explicitly expose a `SamrConnect5` method (it typically uses `SamrConnect` for modern connections). If Microsoft ever requires the use of the newer call for certain scenarios (or deprecates older calls), the tool would need an update to Impacket or a workaround. Similarly, some information available in SAMR might not be parsed out by Impacket's current structures. We noticed, for example, that certain fields related to password policies and extended account attributes are not readily exposed in Impacket's high-level calls (they would require manual DCE/RPC calls with specific info levels). These were beyond our immediate needs, but it means that the tool might not extract everything the SAM database holds if Impacket does not support it. In essence, the capabilities of the tool are limited by what the underlying library can do, and any mismatch between the official protocol and the library's implementation is a potential limitation. However, no mismatches were observed that affect the core functionality. However, reliance on a third-party library remains an inherent risk that must be monitored as Windows or Impacket evolve.

Unsupported OpNums that Offer Unique Value

SAMR defines several enumeration operations not yet implemented in `samr-enum`, which offer capabilities beyond those of calls currently supported. Integrating them could improve the scope of enumeration by exposing unique data, thereby strengthening the tool's coverage in multi-forest Active Directory environments.

- `SamrValidateComputerAccountReuseAttempt` (OpNum 74) checks whether a domain controller permits rejoining a computer account based on its SID. Returns a Boolean indicating whether reuse is allowed. This type of policy-level check is not supported by current enumeration calls, which focus on listing existing accounts rather than validating reuse conditions.

- `SamrAccountIsDelegatedManagedServiceAccount` (OpNum 77) checks whether a given account is a Delegated Managed Service Account (gMSA/dMSA) and whether the caller has permission to use it. It returns two booleans: `Result` (`TRUE` if the `objectClass` is `msDS-DelegatedManagedServiceAccount`) and `authorised` (`TRUE` if the caller is allowed via `msDS-GroupMSAMembership`). This functionality is not provided by standard user or group enumeration calls.

Unsupported OpNums Already Covered by Existing Functionality

Some remaining OpNums are technically unsupported but do not expand `samr-enum`'s capabilities. Their core features overlap with operations already in place, such as enumerating group membership or retrieving domain policy details. Consequently, separate implementation of these calls would produce minimal additional information, making their inclusion a lower priority for future development.

- `SamrGetAliasMembership` (OpNum 16). This method returns the union of all RIDs of the alias group to which a given set of user SIDs belong. In practice, the tool can already derive this information using supported calls: for example, by enumerating all aliases via `SamrEnumerateAliasesInDomain` (OpNum 15) and retrieving each alias's members via `SamrGetMembersInAlias` (OpNum 33), one can determine which aliases a user SID is in. Thus, `SamrGetAliasMembership` provides a more direct way to get alias memberships for a user (local group) but does not expose new data that cannot be obtained by combining existing calls.

- `SamrGetUserDomainPasswordInformation` (OpNum 44) and `SamrGetDomainPasswordInformation` (OpNum 56) overlap with existing functionality. These calls retrieve select password policy settings of the domain.

SamrGetUserDomainPasswordInformation returns the domain password policy (for example, minimum password length and password complexity flags) given a user handle, and SamrGetDomainPasswordInformation does the same without needing a user handle (and with no access check). The tool already supports SamrQueryInformationDomain2 (OpNum 46), which can obtain the domain password policy (the DOMAIN_PASSWORD_INFORMATION fields) including these values. In other words, the minimum length and password properties returned by these calls are already accessible via the domain information query. The main difference is that the SamrGetDomainPasswordInformation method is intended for the convenience of the client (available with minimal rights to let end-users view policy), but it does not provide new policy data beyond what the tool can obtain through the existing domain info query.

- SamrQueryDisplayInformation (OpNums 40, OpNums 48, OpNums 51) methods (versions 1, 2, and 3) return a batch of account entries (users, machines, or groups) in ascending name order along with details such as account RID, name, and attributes. In essence, they are an alternative way to enumerate accounts, optimised for retrieving display data in sorted order. The samr-enum tool already enumerates users and groups via SamrEnumerateUsersInDomain (OpNum 13) and SamrEnumerateGroupsInDomain (OpNum 11), and can obtain detailed attributes for each account using SamrQueryInformationUser2 or group queries (supported OpNums 47 and OpNum 20). For example, fields such as the user's full name or admin comment (which SamrQueryDisplayInformation provides in one go) can be obtained by opening the user and calling SamrQueryInformationUser2. Therefore, while the QueryDisplayInformation calls would simplify getting a sorted, richly detailed account list, they do not provide new information – they package data that the tool can already collect through a combination of enumeration and per-account queries.

- SamrGetDisplayEnumerationIndex (OpNums 41 and OpNum 49) is a helper function that computes the index of a given account name prefix in the sorted account list. It does not return any account data itself, only a position. In context of samr-enum, which typically retrieves all accounts for enumeration, this is not a necessary separate operation – the tool can sort retrieved names locally or simply enumerate all accounts without needing an index lookup. Since SamrQueryDisplayInformation (above) is not essential for new data, the index lookup calls are even less critical. They offer no unique information about SAM objects, merely a convenience for client paging, so implementing them would not expand the tool's capabilities.

- `SamrRidToSid` (OpNum 65) returns a full SID given a relative ID (RID) within a SAM context. Although this call simplifies operation, it does not expose additional directory data. The tool can already determine SIDs for accounts by combining known information: after connecting to a domain, the `SamrLookupDomainInSamServer` (OpNum 5) call yields the domain's SID. With the domain SID in hand, any RID (e.g., from an enumerated user or group) can be converted to a SID by appending it as the RID portion. Thus, `SamrRidToSid` duplicates a computation the tool can perform on its own. There is no unique enumeration benefit in a separate implementation of this call, as it would return the same identifier values the tool can derive with existing calls.

Enhancing Error Handling

Improving error handling and user feedback is another future goal. During testing, we identified scenarios where error messages could be improved for clarity and usability. For example, if a Kerberos login fails because of a clock skew or because the SPN was not found, Impacket returns a low-level error message. We can catch such errors and provide suggestions (for example, *Kerberos authentication failed, please check your clock synchronisation or SPN*). If the tool fails to connect to a host, it prompts the operator to verify network connectivity and firewall settings.

In a multi-forest deployment, the tool might encounter domains or domain controllers with varying configurations (such as older Windows versions or stricter SAMR permissions). Improving exception handling would allow the script to gracefully handle access-denied errors or RPC timeouts – for example, skipping over an unreachable domain controller after logging a warning, rather than aborting the entire enumeration. Similarly, better handling of large domain datasets (tens of thousands of objects) is needed: the current implementation could incorporate chunked retrieval and progress checkpoints to avoid memory issues or partial results if interrupted.

These improvements aim to make the tool more accessible to users with limited experience in Active Directory networking.

Refining Output Parsing and Format Options

Another limitation mentioned is the parsing and formatting of output. Although JSON and CSV output exists, there is room to refine the schema. For instance, group membership is currently shown as an array of RIDs or as names. This format could be unified to always use names or include both representations. Additionally, in large domains, the output files may become substantial. Future versions could

offer filtering options to narrow the scope of enumeration, for example, to include only users with *admin* in their name.

Transport and Authentication Enhancements

Future development could improve authentication and transport flexibility for compatibility with diverse enterprise environments. One enhancement is support for authentication via computer (machine) accounts, useful in administrative contexts such as service-based or scheduled enumeration. Support for multiple transport bindings could also increase reliability. The current implementation uses SMB-based `ncacn_np:` transport for all SAMR queries. Some environments may restrict SMB access while allowing direct RPC over TCP. Adding a fallback to `ncacn_ip_tcp` (RPC over TCP/135) would allow operation in such settings without manual reconfiguration.

Integrating with Complementary Reconnaissance Tools

Integration with external tools that support LDAP-based enumeration, such as BloodHound, could extend the applicability of `samr-enum` in post-enumeration analysis. For example, SAMR-derived local group memberships, often not accessible via standard LDAP queries, could be exported to BloodHound-compatible CSV files to support relationship graph construction. As a possible future enhancement, the tool could include an output mode customized to the expected BloodHound schema or provide an interface to populate a Neo4j database directly.

Conclusions and Future Roadmap

In conclusion, while `samr-enum` in its current state achieves the core objectives and performs reliably, the identified limitations and proposed enhancements define a roadmap for future development. Addressing IPv6 support, aligning with protocol updates, expanding coverage of SAMR operations, and improving usability and interoperability with external analysis tools will help ensure the continued relevance of the tool in modern Active Directory environments. Each of these improvements supports the overarching goal of the thesis: enabling administrators and security professionals to assess account and group visibility across complex multi-forest configurations using precise and verifiable methods.

5 Methods, Experiments, and Discussion

Introduction. This experiment demonstrates how the `samr-enum` tool operates in a multi-forest Active Directory environment. The study records (i) the set of SAMR operations executed by the tool, (ii) the structure of the data returned, and (iii) the tool response to changes in security configurations. `samr-enum` is executed under two trust-authentication configurations—forest-wide and selective—each evaluated across multiple sub-scenarios, including default and hardened ACLs. A quantitative analysis of the output shows how the amount of accessible information varies between configurations and confirms the suitability of the tool for research and audit tasks in multi-forest settings.

5.1 Research Design and Laboratory Setup

This section provides a description of how the experiment was designed, the variables controlled, and how the laboratory environment was configured. The following section describes in detail the data collection and analysis methods. The goal was to measure how reconfiguring a single forest trust between two Active Directory forests and modifying access of the domain objects influence the volume and type of SAMR data disclosed to a non-privileged security principal.

Overall Rationale and Multi-Forest Scope

The experiment was conducted in a multi-forest controlled environment consisting of `domain-a` (*Forest A*), `domain-b` (*Forest B*), and `domain-x` (*Forest X*). For brevity, the *.local* suffix is omitted from domain names throughout this thesis. Each test scenario focused on a forest trust between two of these forests to isolate the impact of that trust configuration on SAMR data exposure. This design maintained a manageable infrastructure while representing a large directory dataset, as one forest contained tens of thousands of objects. The research goal was to observe how a single trust relationship influences SAMR data availability without the confounding effects of multiple simultaneous inter-forest trusts.

Logical Variables and Control Variables

The independent factors in this study were (i) the authentication scope (forest-wide versus selective) and (ii) discretionary access control lists that protect target objects. Each unique combination of these variables constituted an experimental condition. Because altering the forest functional level during this final stage could introduce unrelated feature changes, the forest functional level remained fixed at Windows Server 2016 on both sides. This version was applied consistently at the domain and forest levels to ensure predictable SAMR behaviour across all test

scenarios. Configuration details related to domain controllers, trust structure, network topology, and virtualisation environment are provided in Appendix B.

The dependent measures were (i) the number of objects returned per SAMR OpNum and (ii) the proportion of attributes exposed to the remote caller.

Several control variables were kept constant across both forests to eliminate sources of variability not related to trust configurations. Both domain controllers ran Windows Server 2022 with matching hotfix levels, and the same /24 subnet model was used to maintain comparable network latency. Moreover, IPv6 was disabled on both servers, each domain controller used the same time zone and synchronised its clock with the host server, the Windows Defender Firewall was turned off and no antivirus software was installed. These measures ensured that differences in addressing, clock skew, or security configuration did not influence the experimental results.

Network Topology and Data Corpus

Each forest contained exactly one domain controller: `adc1.domain-a.local` in *Forest A*, `bdc1.domain-b.local` in *Forest B*, and `xdc1.domain-x.local` in *Forest X*. All three servers resided in distinct subnets (192.168.1.0/24, 192.168.2.0/24, and 192.168.10.0/24). A virtual router forwarded traffic between subnets, and host and network firewalls were disabled inside the lab to prevent interference with SAMR traffic.

Forest B contained additional test objects: standard and disabled user accounts, a domain-admin user, service accounts (including an SPN account, one managed service account, and one group Managed Service Account), and several group objects spanning domain-local, global, universal, and distribution scopes. An organisational unit with a customised access control list was also created to evaluate ACL effects on object visibility.

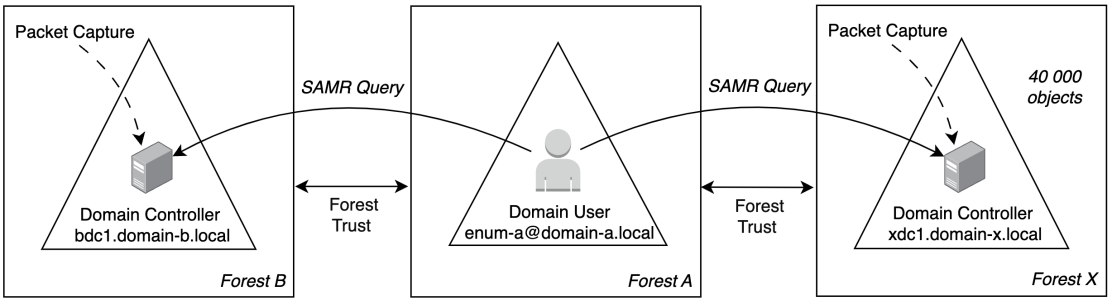

Figure 2. Enumeration Vector

Caller Identity and Non-Privileged Context

The enumeration requests originated from the account `enum-a@domain-a.local`, which was a standard user in *Forest A* without administrative rights in either forest (Figure 2). The `samr-enum` tool was executed on an Ubuntu Desktop 24.04 workstation not joined to any domain; however, the domain credentials were provided as part of the tool's command parameters to authenticate against *Forest B* and *Forest X*. Testing with a non-privileged identity aligned with the threat model of an attacker who has a foothold in one forest and attempts to discover vulnerabilities in another.

Execution Procedure and Environment Reset

For each experimental condition, the existing trust was removed in the Active Directory Domain and Trusts console, then recreated with the required parameters (for example, switching from forest-wide to selective authentication). This process minimised overhead by reusing the same domain controllers instead of introducing new machines or forest configurations. After applying the new trust settings, a reboot of the system ensured that caches, tickets, and name resolution were cleared.

Having established the laboratory design and the procedure for reconfiguring trusts, the following section describes how the data were collected, processed, and statistically analysed to evaluate the differences in SAMR enumeration outcomes.

5.2 Data Collection and Analysis Methods

Data collection in the study consisted of capturing the output of the `samr-enum` tool and inspecting the associated network traffic. These data streams allowed verification of object listings and attribute details returned by SAMR under varying trust configurations.

Tool Execution and Output

Each trust configuration was tested by running the `samr-enum` tool from the Ubuntu workstation outside the target forest. The execution parameters included the target domain name (`bdc1.domain-b.local`), non-privileged user credentials (`enum-a`), and the enumeration object within the domain. The tool displayed user accounts, groups, service accounts, and computer objects, along with attribute details such as group memberships, account statuses, and partial access flags. This text output was saved in a local file for subsequent parsing.

Where relevant, a PowerShell script was executed on `bdc1.domain-b.local` to retrieve the object's access control list (ACL) locally. This allowed validation of *permission* inheritance and object accessibility. The script output and `samr-enum` results are available on the repository's experiment page; see Appendix B.

Packet Capture

In parallel with the execution of the tool, Wireshark was used in the domain controllers `bdc1.domain-b.local` and `xdc1.domain-x.local` to collect network traffic on the relevant Hyper-V switch port. The capture included Remote Procedure Call (RPC) traffic between `xws1` Ubuntu workstation and domain controllers. The resulting PCAP files were stored in PCAPNG format and uploaded to the project repository along with instructions for decrypting the payload.

Data Examination and Verification

The captured packet data was inspected for SAMR segments, which were decrypted in Wireshark using the credentials provided. Each SAMR request or response was matched to the corresponding console output of `samr-enum`, confirming that OpNum sequences and returned object attributes aligned with the tool's reported results. Anomalies such as access-denied errors were noted and cross-referenced with possible ACL misconfigurations.

Repository of artefacts

All captured PCAP files, `samr-enum` outputs, PowerShell ACL listings, and command parameters are available in the GitHub repository (see Appendix B). This enables reproducibility by providing a complete record of each run, including enumeration commands and Wireshark captures. By separating tool execution, network capture, and ACL inspection, the experiment verified that `samr-enum` functions correctly under varying trust and *permission* settings. The tool issued the expected SAMR calls, parsed the responses, and reflected visibility changes per scenario. These controlled conditions confirmed the reliability of the tool's in reproducing and measuring the enumeration results in multi-forest environments.

5.3 Experimental Results

In summary, the baseline experiment demonstrates that `samr-enum` reproduces the expected Active Directory enumeration behaviour under a forest-wide trust with default ACLs. All user accounts (regular, disabled, and administrative) and all non-local groups were discoverable by the tool, consistent with documented SAMR behaviour. At the same time, objects restricted by design to the local domain scope (such as domain-local security groups and distribution lists) remained hidden from the cross-forest query, which matches the established visibility scope of these objects in Active Directory. These findings confirm that the default output of the tool's in a multi-forest scenario aligns with Microsoft's official protocols and security model, even when the target forest contains tens of thousands of objects, and highlight the types of objects that a standard SAMR query will and will not reveal.

5.3.1 Baseline Cross-Forest Enumeration

In the baseline scenario, the `samr-enum` tool was executed by a non-privileged user from `domain-a` to enumerate directory objects in a trusted `domain-b` and `domain-x` under a forest-wide trust with default permissions. This configuration allows all authenticated users from `domain-a` to query `domain-b` and `domain-x` without special access grants. As expected, the tool enumerated all user accounts in the target domain, including default disabled accounts (e.g., *Guest* account) and privileged accounts (e.g., *Domain Admins*). The inclusion of these accounts reflects the default Active Directory behaviour that even disabled or *AdminSDHolder*-protected accounts remain visible to authenticated users when no access control changes are in place. In fact, Microsoft's official SAMR specification confirms that the SamrEnumerateUsersInDomain call *enumerates all users* in a domain, which explains why `samr-enum` returned every user object by default. In particular, the presence of the *AdminSDHolder* protection on privileged accounts did not remove those accounts from the results, indicating that the default *AdminSDHolder* ACL does not restrict read access to the enumeration of the domain in this scenario.

The tool output also demonstrated that both service accounts and Managed Service Accounts (*MSA/gMSA*) are enumerable under default conditions. All service accounts configured in `domain-b` were listed, and the MSA/gMSA objects appeared when the tool queried the computer objects. This behaviour occurs because Active Directory treats MSAs as computer-class objects in the directory schema; thus, a comprehensive SAMR enumeration of computers includes MSAs by design. The `samr-enum` tool successfully retrieved these accounts along with standard user and computer accounts, illustrating its capability to capture a wide range of account types in a multi-forest environment.

The group enumeration in the baseline test revealed that the global and universal security groups in `domain-b` were fully visible to the foreign user, while the domain-local groups were not. All configured global and universal groups (which are meant to be visible throughout the forest-wide by Active Directory design) were returned in the domain group listing. In contrast, domain-local groups were completely absent from the enumeration results. This result is consistent with Active Directory's scope rules: domain local groups are only visible within their home domain and are not by default exposed to users in external domains. For instance, in our lab `domain-b`, 100% of global and universal groups were enumerated, but 0% of domain-local groups were returned to the `domain-a` user. The tool's cross-domain query for *all groups* yielded only security-enabled groups; any distribution-only groups (non-security groups) in `domain-b` were similarly omitted from that broad listing. We confirmed that two distribution groups present in `domain-b` did not appear in the general enumeration output, although they could still be discovered by targeted name queries. This behaviour aligns with the focus of SAMR on security principles and is documented as such, for example, the SAMR call to enumerate local groups (aliases) targets security groups in the domain's SAM database.

Another cross-forest artefact observed in the baseline scenario was the handling of Foreign Security Principals (FSPs). When a `domain-a` user is added to a `domain-b` local group, Active Directory creates an FSP object in `domain-b` to represent the foreign member. In our experiment, the FSP representing the `domain-a` account was present in `domain-b`'s directory (as a member of a domain-local group). However, because the domain-local group itself was not visible across the forest trust, the foreign user could not see that group or its membership through SAMR queries. In effect, the default trust and permission settings concealed the existence of domain-local groups (and thus any FSP memberships in those groups) from the enumerator. This outcome underscores that, under out-of-the-box conditions, a cross-forest enumeration with `samr-enum` can enumerate all standard user and computer accounts and all globally scoped groups, but will exclude any purely domain-scoped groups.

In summary, the baseline experiment demonstrates that `samr-enum` reproduces the expected Active Directory enumeration behaviour under a forest-wide trust with default ACLs. All user accounts (regular, disabled, and administrative) and all nonlocal groups were discoverable by the tool, consistent with documented SAMR behaviour. At the same time, objects restricted by design to the local domain scope (such as domain-local security groups and distribution lists) remained hidden from the cross-forest query, which matches the established visibility scope of these objects in Active Directory. These findings confirm that the default output of the tool's in a multi-forest scenario aligns with Microsoft's official protocols and security model, while also highlighting the types of objects that a standard SAMR

query will and will not reveal. The next sections will examine how this visibility changes when we impose custom ACL restrictions.

5.3.2 Cross-Forest Enumeration with Restrictive ACL Modifications.

In this scenario, `samr-enum` is executed under a forest-wide trust while progressively modifying ACLs in the target domain to evaluate how the tool handles restricted access conditions. The objective is to assess whether the tool correctly issues SAMR queries and reflects changes in object visibility as access rights are incrementally reduced. This configuration tests the tool's ability to enumerate directory objects from a non-privileged foreign context while adhering to domain-defined access constraints.

Documented behaviour. Under default ACLs, a low-privileged user from a trusted forest can enumerate almost all accounts in the target domain, including disabled users (such as built-in *Guest*) and high-privilege accounts. This is expected because, by design, the default directory permissions grant *Authenticated Users* broad read access (e.g. *List Contents* and *Read All Properties*) on user and group objects. Even accounts in protected administrative groups remain readable: the *AdminSDHolder* mechanism will regularly reset any changes to their ACLs back to the standard template, ensuring that membership in a privileged group does not remove the default read permissions. As a result, an authenticated user in one forest can, by default, list all user accounts in a trusted foreign domain. Furthermore, SAMR's group enumeration reflects Active Directory's group scope rules. All global and universal security groups are returned in cross-forest queries, whereas domain-local groups are omitted. This behaviour is explicitly enforced by the protocol filter: `SamrEnumerateGroupsInDomain` only includes groups of type *security global* or *security universal*. The absence of domain-local groups in the results aligns with the intended scope being limited to the resources of the local domain.

Expected Design. Selective removal of default access principals from an object's ACL illustrates the cumulative nature of Active Directory permissions. In our experiments, removing the *Pre-Windows 2000 Compatible Access* principal from a *user* object's ACL hid that user from cross-forest enumeration: the SAMR query returned `STATUS_ACCESS_DENIED` and the object was not listed, even when the *Authenticated Users* entry remained. In contrast, removing only the *Authenticated Users* ACE had no effect because the remaining *Pre-Windows 2000 Compatible Access* principal still provided the necessary read rights.

For *group* objects the effect was stricter: removing either principal in isolation did not revoke read permission, and the foreign user could still enumerate the group. Only when both *Authenticated Users* and *Pre-Windows 2000 Compatible Access*

were removed from a group's ACL did the SAMR call fail with a denied access status. These findings confirm that the two default principals provide overlapping access; both must be eliminated to prevent a group from being returned in cross-forest SAMR queries, whereas removing *Pre-Windows 2000 Compatible Access* alone is sufficient to hide an individual user object.

Granular permission tests further demonstrated that partial rights are insufficient for enumeration. For example, granting a foreign user only a limited permission such as *Read General Information* on a target user did not allow that user to be discovered – the enumeration still failed with an access denied status. However, restoring the full read permissions (e.g. an explicit *Read All Properties* allow ACE) immediately made the object enumerable again. This indicates that SAMR enumeration operations require comprehensive read access to the object's attributes; if any essential attribute or listing right is lacking, the query will not return the object. This behaviour is in line with the protocol's design and Microsoft's documentation that the default ACL (which grants full read) is what permits these queries.

Finally, tightening ACLs on containers can selectively hide all child objects from enumeration, but any object may still be revealed indirectly via memberships, unless those are also restricted. In one experiment, removing the default read ACE for *Pre-Windows 2000 Compatible Access* on a specific organizational unit (OU) successfully prevented a computer account in that organizational unit from appearing in a cross-forest machine list. However, that computer object was a member of a security group, and it remained visible when we enumerated the members of that group. The reason is that SAMR's group membership query reads the membership from the group object (which still had default-readable permissions), not from the member's object. Likewise, a user account with a hardened ACL (no default read principals) was no longer returned in a direct user listing, but could still be seen as a member of a group. These outcomes reflect the expected design of Active Directory: an object's own ACL controls the direct enumeration of that object, while group or membership information is governed by the ACL of the container or group where that information resides. To fully obscure an object from all enumeration vectors, one must therefore harden not only the object itself but also any groups or containers that could reveal its existence.

5.3.3 Selective Authentication

In a selective authentication environment between two Active Directory domains, an external user account from the trusted domain attempted full SAMR enumeration (users, groups, and computers) in the target domain using the `samr-enum` tool under three access conditions: no *Allow to authenticate* permission, *Allow to authenticate* on the domain object, and *Allow to authenticate* in a specific OU. The experimental results show that only with domain-level *Allow to authenticate* rights did the

enumeration succeed, returning all user, group, and computer objects along with their attributes. In contrast, when the account lacked this permission entirely or had it only at the OU level, the SAMR queries returned no user, group, or computer objects; even objects located in the permitted OU were not enumerated. These results indicate that under selective authentication, the *Allow to authenticate* permission must be granted at the domain scope for SAMR enumeration to return any results.

5.3.4 Interpretation of Results

The custom enumeration tool, `samr-enum`, exhibited consistent and correct behaviour in the three experimental scenarios, demonstrating its ability to detect changes in object visibility and attribute exposure under varying trust and ACL conditions. In the forest-wide trust scenario with default ACLs, `samr-enum` enumerated all tested domain-object types: regular user accounts, disabled users, service accounts (including gMSA / MSA) and privileged groups, showing that default cross-forest permissions expose every object class except domain-local groups, which SAMR does not return to a foreign caller by design. When ACLs were selectively tightened in the second experiment (for example, by removing the *Pre-Windows 2000 Compatible Access* entry from *user* objects), the tool's output immediately reflected reduced visibility: previously listed objects disappeared, confirming that fine-grained ACL changes translate directly into enumeration results. Removing only the *Authenticated Users* entry, however, left the object set unchanged, indicating that this principal is redundant when *Pre-Windows 2000 Compatible Access* remains. In every case, `samr-enum` faithfully reported the presence or absence of objects according to the effective permissions, validating that the tool responds precisely to ACL modifications.

The third scenario, which involves selective authentication trusts, further underscored `samr-enum`'s responsiveness to trust configuration changes. With selective authentication enabled (and no special permissions granted), the foreign user had no default right to enumerate the target domain, and accordingly `samr-enum` returned no enumerated objects. This confirms that the tool adequately reflects the severe restriction of object visibility when cross-forest authentication is denied. Once the *Allow to authenticate* permission was granted at the entire domain level for the foreign principal, `samr-enum` produced a comprehensive listing of the users and groups of that domain, demonstrating that it immediately responds to the expanded scope of access. Attempts to grant *Allow to authenticate* in a narrower scope (such as an individual organisational unit) did not yield any partial enumeration – the tool continued to report zero objects from that OU, reinforcing the observation that cross-forest SAMR enumeration is all-or-nothing at the domain level. These results show that `samr-enum` reliably mirrors trust-based access controls: it enumerates

directory objects only to the extent that the authentication settings permit, with no false positives or extraneous data beyond what the security configuration allows.

In all experiments, `samr-enum` proved to be a reliable and comprehensive SAMR query tool. Implements a broad set of SAMR operation calls (OpNums) – including user and group enumerations, alias (local group) listings, and membership retrieval – and processes the responses in a structured format. This coverage ensured that, for each scenario, the tool gathered all accessible information and presented it in an organised manner, facilitating direct comparison between different security settings. The structured output made it clear which objects and attributes were retrievable in each configuration and which were not, thus highlighting the exact impact of each ACL or trust change. In particular, all observed variations in `samr-enum`'s results aligned with the expected behaviour of the underlying Active Directory security model. The tool consistently listed objects when the environment settings allowed it, and just as consistently omitted objects or attributes that had been secured against enumeration. These experimentally verified results illustrate that `samr-enum` can be reliably applied in different Active Directory environments to assess SAMR exposure. In practice, an auditor or administrator can run `samr-enum` under various trust and permission configurations and trust its output to reveal the true extent of the domain object data visible to a given foreign principal. This performance in all three test scenarios confirms the effectiveness of the tool's in detecting even subtle changes in directory visibility, without relying on external heuristics, thus establishing `samr-enum` as a reliable method for evaluating the exposure of SAMR information under various cross-forest conditions.

5.3.5 Risk Analysis and Security Implications

In multi-forest Active Directory environments, trust relationships can inadvertently expand the attack surface if not properly controlled. The risk analysis showed that when two forests are connected by bidirectional trust with forest-wide authentication and default ACLs, a low-privileged user from one forest can enumerate a wide range of objects and attributes in the other forest. This represents a serious reconnaissance risk: an adversary who compromises a low-level account in one domain could harvest user lists, group memberships, and computer accounts from a trusted domain, gaining valuable intelligence for a potential attack. In contrast, a hardened trust configuration – for example, using selective authentication without cross-forest *Allowed to Authenticate* permissions – can significantly reduce this exposure. The difference between these scenarios underscores the importance of systematically assessing SAMR-related exposures and enforcing directory access controls that follow the principle of least privilege.

The `samr-enum` tool supports this risk assessment by providing a methodical way to enumerate directory data under various trust and *permission* settings. As

detailed in subsection 4.3, it implements a wide spectrum of SAMR queries and produces structured, machine-readable output. This enables defenders and auditors to quantify information leakage in each configuration. Rather than relying on assumptions, security teams can run the tool from a low-privileged context in a partner or external forest and directly observe which objects and attributes are exposed. This empirical approach to risk validation is especially useful when verifying mitigation measures. For instance, after administrators tighten discretionary access control lists (DACLs) or switch a forest trust to selective authentication, `samr-enum` can be rerun to confirm the effect: a reduction in returned objects or attributes serves as evidence that the hardening is effective. If any high-value accounts or hidden groups still appear in the results, it suggests incomplete restrictions, prompting further adjustments to security settings. In this way, the tool provides an auditing mechanism to verify whether cross-forest queries are properly constrained and whether SAMR exposure is minimised.

Security teams may embed `samr-enum` within scheduled audits of Active Directory trust relationships. The tool outputs JSON or CSV listings of enumerated users and group visibility. Repeated execution under consistent conditions permits direct comparison between configurations or time points. Teams establish a baseline, then detect deviations when trust settings or Access Control Lists change. A difference report highlights renewed exposure if later administration grants wider *permission* read scopes. This process supports ongoing control of Security Account Manager Remote access across forests.

Experimental runs confirm that the baseline, Restrictive Access Control Lists, and Selective Authentication configurations reduce cross-forest enumeration surface and limit *permission* escalation.

The scope of the laboratory, the use of synthetic identities, and the absence of concurrent domain controller load were acceptable simplifications for the purpose of controlled testing. Although these factors may differ in production environments, the results and methodology of this study remain applicable and may inform real-world deployments, subject to additional validation under specific operational conditions.

5.4 Limitations and Suggestions for Future Research

The experimental Active Directory environment was narrowly scoped. All domain controllers in our laboratory environment ran Windows Server 2022 exclusively, so no other server versions or alternative directory implementations were evaluated. Furthermore, the environment enforced only NTLM authentication for cross-forest requests, whereas in general a forest trust can utilise Kerberos for inter-domain authentication. Authentication for cross-forest requests relied on NTLM because captured traffic can be decrypted more easily in a controlled setting; therefore, the

findings describe SAMR behaviour under NTLM and may not fully represent the conditions where Kerberos is employed.

Another limitation lies in the trust configuration that was examined. The study covered only a two-way transitive forest trust. One-way trust scenarios, where access is unidirectional (e.g., Domain A trusts Domain B), were outside the scope of this study. As a result, the findings do not capture how SAMR enumeration might differ when only one side of a trust is allowed to initiate queries.

The custom enumeration tool, `samr-enum`, also has specific functional limitations. In particular, the tool operates in a read-only capacity and does not implement any write operations via SAMR (e.g., it cannot invoke `SamrSetInformationUser`, which updates user account attributes). Its queries focus on listing users, groups, computers, and their memberships, but it lacks support for retrieving detailed attribute fields of computer account objects. In addition, `samr-enum` communicates exclusively through the named-pipe transport (`\PIPE\samr` via SMB) and does not initiate SAMR connections using RPC over TCP. As a result, code paths or behaviours specific to the TCP-based protocol sequence were not exercised during the experiments.

In addition, the tool exhibits two issues under certain conditions. First, when enumerating a domain controller with large numbers of objects (e.g., `xdc1.domain-x.local`) using the `enumerate=summary` parameter to gather aggregate counts of users, groups, and computers, it returns no results. In contrast, retrieving the full lists of users, groups, or computers individually from the same domain is completed successfully. Second, if a queried object does not exist in the directory, the tool terminates with a generic error message instead of explicitly indicating that no such object was found. These limitations are planned to be addressed in future versions of `samr-enum`.

Finally, several potential scenarios were outside the scope of this research. We did not evaluate how SAMR enumeration behaves under an account with elevated privileges (for instance, using an account in the *Domain Admins* group to perform the queries), so any differences in object visibility or access due to higher privilege levels remain unexamined. Likewise, the study did not cover various Active Directory configurations beyond the basic two-way trust used. Security features such as SID filtering on trust links were not enabled or tested. SID filtering is a mechanism that removes certain security identifiers across trust boundaries to prevent illicit privilege inheritance; therefore, its effect on the results of the enumeration was not observed. Furthermore, specialised multi-forest topologies like the resource forest model (where one forest contains resources and trusts the accounts from another forest) were not represented in our lab setup. In summary, the conclusions drawn are limited to the specific environment and conditions tested and do not encompass the above untested scenarios.

Future work can extend the `samr-enum` tool's capabilities by incorporating additional, underutilised SAMR operations that were beyond the scope of the current evaluation. In particular, implementing support for rarely invoked calls such as `SamrValidateComputerAccountReuseAttempt` (OpNum 74) and `SamrAccountIsDelegatedManagedServiceAccount` (OpNum 77) is a logical next step. These methods (which validate computer account reuse attempts and delegated service account status, respectively) have only recently been introduced and remain underexplored in practice. By extending `samr-enum` to cover these calls, researchers can observe their behaviour in real-world scenarios, especially across domain trust boundaries. Testing the tool in cross-domain or cross-forest trust contexts will reveal whether these newer SAMR calls face restrictions or inconsistencies when invoked between trusted domains, thus deepening our understanding of SAMR behaviour in multi-domain environments.

Another recommended enhancement is to refine the handling of errors and improve the clarity of the SAMR response output, particularly in cases where the server returns partial results or nuanced error codes. In addition, structured error codes (such as specific `NTSTATUS` values signaling denied access, invalid parameters, or other RPC-level issues) should be captured and decoded into meaningful messages. Providing more granular feedback, for example, distinguishing *access denied to attribute X* from a generic failure would help users pinpoint permission limitations or misconfigurations encountered during enumeration. Improving the fidelity of error reporting in this way will make the tool more effective for troubleshooting complex SAMR query scenarios. A future enhancement could focus on richer diagnostics for negative responses. For example, `samr-enum` could map frequently observed `NTSTATUS` error codes—such as `STATUS_ACCESS_DENIED`, `STATUS_INVALID_HANDLE`, or `STATUS_INVALID_INFO_CLASS`—to concise explanations in its structured output, helping operators identify whether a failure stems from insufficient permissions, an incorrect object handle, or an unsupported information class. Adding optional verbosity levels would allow an auditor to request full RPC header details when troubleshooting, while keeping the default output compact during routine assessments.

Finally, we suggest evaluating how fine-grained object-level permissions impact the results of SAMR-based enumeration. The experiment would involve assigning minimal directory rights to a user and observing how much information `samr-enum` can obtain with those limited privileges. For example, a domain user could be granted only the *ListContents* permission on an Organizational Unit (OU) or only the *Read General Information* property set on specific objects, without broader read access. Running `samr-enum` under these constraints will show which objects and attributes become visible when user permissions are restricted to such narrow scopes. This is motivated by real-world scenarios in which an account might

have the ability to list the existence of an object, but not read its properties (or vice versa). Empirical data from these tests would clarify the behaviour of SAMR in partial access situations. For instance, previous work has shown that completely removing an account's *ListContents* right on an OU can hide objects from being listed by the account. Conversely, if *ListContents* is allowed but most read permissions are withheld, one would expect the user to see that an object exists (e.g., its name) without being able to retrieve detailed attributes. By systematically varying such fine-grained ACL settings (for example, granting only the right to list objects or only to read a subset of attributes) and capturing the output of `samr-enum`, future research can map specific permission combinations to their enumeration results. This knowledge would not only validate the principle of least privilege in the context of SAMR queries but also guide improvements to the tool (such as warning when results may be partial due to limited rights). In summary, exploring these targeted extensions - broader SAMR call coverage, improved error reporting, protocol behaviour comparison, and fine-grained permission effects - will directly build upon the experimental findings and further strengthen the utility of `samr-enum` as a research and auditing tool.

6 Conclusion

6.1 Summary

In conclusion, this thesis addressed the challenge of SAMR enumeration in multiple Active Directory forests by developing the `samr-enum` tool and systematically analysing its use under various conditions. The work was motivated by limitations in existing utilities and by the need to measure how multi-forest settings influence information exposure through SAMR. The tool implements several SAMR operations and outputs structured results suitable for auditing, thereby enabling repeatable security assessments. Experiments in a multi-forest laboratory demonstrate that both inter-forest trust configuration and access-control lists change the number of objects and attributes returned by SAMR queries. This evidence answers the main research question on systematic observation of SAMR, as well as the subsidiary questions about the effects of trust configuration and access rights. Consequently, the research offers academic insight into the cross-forest SAMR enumeration and practical means of evaluating Active Directory security.

The `samr-enum` source code is available in a public GitHub repository[14], ensuring transparent review and reuse. A request to Kali Linux maintainers to include the tool in the distribution received a positive response, confirming that a properly packaged release will be accepted[15]. This outreach demonstrates external interest and supports the tool's adoption in security auditing workflows.

6.2 Limitations

This research was conducted under controlled conditions that inherently constrain the generalizability of its findings. All experiments used an IPv4 network; IPv6 operation and dual-stack scenarios were not examined. The evaluated cross-forest relationship was a two-way transitive trust between three forests with default trust settings. One-way trust directions and more complex multi-forest topologies (such as a resource forest model) were outside our scope. In addition, all enumeration tests were performed using low-privileged user accounts. The behaviour was not observed with elevated privileges (for example, using an account in the *Domain Admins* group) or in environments with customised hardened ACLs. Consequently, the conclusions drawn reflect this specific environment and may not directly apply to fundamentally different Active Directory deployments.

The custom tool `samr-enum` also has inherent limitations that affected the scope of our analysis. It was designed for read-only enumeration and does not implement

[14] `https://github.com/studylab1/samr-enum`

[15] See discussion thread on the Kali Linux forum: `https://forums.kali.org/t/requesting-feedback-samr-enum-new-samr-enumeration-tool-for-kali-linux/4440`

any SAMR write operations, which means the study did not explore modifications or administrative actions via SAMR. Certain less common SAMR calls (including some newly introduced in recent Windows versions) were not supported, constraining the breadth of SAMR data examined. Furthermore, `samr-enum` communicates exclusively over the named-pipe interface of the SMB (NCACN_NP) and was not tested with the alternative RPC over TCP transport. This design choice, combined with the tool dependence on an external RPC library, means that no protocol-specific behaviours or unimplemented features were captured in our results (for instance, subtle differences in IPv6 handling or newer SAMR functions).

6.3 Future work

Implementation-specific tasks remain in Sections 4.5 and 5.4. The following lines outline wider research directions.
An avenue for future research is to compare the enumeration behaviour of SAMR and LDAP under identical ACL conditions. Parallel experiments with both protocols, using the same access-control settings, would highlight differences in the directory information returned by each method. This approach would clarify whether SAMR reveals any objects or attributes not available via LDAP (and vice versa) when the underlying *permissions* are identical.

Another possible avenue involves integrating the results of `samr-enum` into broader auditing ecosystems. For instance, the enumeration output could be formatted for ingestion by Security Information and Event Management (SIEM) systems, allowing the data to be correlated with other security logs. Furthermore, automating the difference analysis between `samr-enum` outputs from successive configuration snapshots could help identify changes in Active Directory settings or unexpected exposure of objects over time. These integrations could make the tool's findings more directly applicable to routine audits and continuous security monitoring, extending the practical impact of the enumeration study.

References

[1] Darktrace Ltd. Detecting unknown ransomware: a Darktrace case study; 2022. [cited 2025-04-18]. [Internet]. Cambridge (UK): Darktrace. Available from: https://www.darktrace.com/blog/detecting-the-unknown-revealing-uncategorised-ransomware-using-darktrace.

[2] The DFIR Report. BlackSuit ransomware incident analysis; 2024. [cited 2025-04-22]. [Internet]. [place unknown]: The DFIR Report. Available from: https://thedfirreport.com/2024/08/26/blacksuit-ransomware.

[3] Schroeder W. A guide to attacking domain trusts; 2017. [cited 2025-04-11]. [Internet]. San Francisco (CA): Medium. Available from: https://harmj0y.medium.com/a-guide-to-attacking-domain-trusts-ef5f8992bb9d.

[4] Tenable Inc . Active Directory is now in the ransomware crosshairs.; 2021. [cited 2025-04-22]. [Internet]. Columbia (MD): Tenable. Available from: https://www.tenable.com/blog/active-directory-is-now-in-the-ransomware-crosshairs.

[5] MA MBJAEMA. Active Directory attacks—steps, types, and signatures. Electronics. 2022;11(16):2629. [Internet]. Basel (CH): MDPI; [cited 2025-04-22]. Available from: https://www.mdpi.com/2079-9292/11/16/2629.

[6] T SSRSF. LDAP enumeration: unveiling the double-edged sword of Active Directory; 2024. [cited 2025-03-17]. [Internet]. Santa Clara (CA): Palo Alto Networks Unit 42. Available from: https://unit42.paloaltonetworks.com/lightweight-directory-access-protocol-based-attacks.

[7] Securelist Research Team. A journey into forgotten Null Session and MS-RPC interfaces; 2024. [cited 2025-04-16]. [Internet]. Woburn (MA): Securelist. Available from: https://securelist.com/no-auth-domain-information-enumeration/112629.

[8] Defatsch S. BloodHound inner workings and limitations—part 1: user rights enumeration through SAMR and GPOLocalGroup; 2022. [cited 2024-10-05]. [Internet]. Jona (CH): Compass Security Blog. Available from: https://blog.compass-security.com/2022/05/bloodhound-inner-workings-part-1.

[9] A CSC. Ghost in the SAM: stealthy, robust, and privileged persistence through invisible accounts. Proceedings of the ACM Conference on Computer and Communications Security. 2024:in press.

[10] W S. Not a security boundary: breaking forest trusts; 2018. [cited 2025-02-20]. [Internet]. Washington (DC): SpecterOps. Available from: https://posts.specterops.io/not-a-security-boundary-breaking-forest-trusts-cd125829518d.

[11] Dirkjanm io. Active Directory forest trusts part 2: trust transitivity and finding a trust bypass; 2021. [cited 2025-03-23]. [Internet]. [place unknown]: Dirkjanm.io. Available from: https://dirkjanm.io/active-directory-forest-trusts-part-two-trust-transitivity/.

[12] N A. CrackMapExec in action: enumerating Windows networks (part 1); 2023. [cited 2025-03-18]. [Internet]. San Francisco (CA): Medium. Available from: https://medium.com/r3d-buck3t/crackmapexec-in-action-enumerating-windows-networks-part-1-3a6a7e5644e9.

[13] Kitploit. Enum4linux-NG: a next-generation version of Enum4linux with JSON/YAML export; 2020. [cited 2025-04-20]. [Internet]. [place unknown]: Kitploit. Available from: https://www.kitploit.com/2020/12/enum4linux-ng-next-generation-version.html.

[14] Rapid7. Metasploit SAMR account enumeration module; n.d. [cited 2025-04-16]. [Internet]. Boston (MA): Rapid7. Available from: https://www.rapid7.com/db/modules/auxiliary/admin/dcerpc/samr_account.

[15] Lai C, N ZC. Quantitative analysis and enforcement of the principle of least privilege in role-based access control. In: Proceedings of the International Conference on Security and Cryptography (SECRYPT 2006). Setúbal (PT): INSTICC Press; 2006. p. 69-74. [cited 2025-01-09]. Available from: https://www.scopus.com/record/display.uri?eid=2-s2.0-77954092715&origin=inward&txGid=b8ed3e9b1c97fb2bab50ef292a6499cd.

[16] Carlos DM, Ramon BHJ, Javier BH, Antonio SMJ, Gamez Gomez N. On attacking Kerberos authentication protocol in Windows Active Directory services: a practical survey. IEEE Access. 2021;9:109289-319. [Internet]. [cited 2025-01-09]. Available from: https://doi.org/10.1109/ACCESS.2021.3101446.

[17] Roy S, Nazia S, C AJ, Christopher K, Laszka A. Survey and taxonomy of adversarial reconnaissance techniques. ACM Computing Surveys. 2023;55(6). [Internet]. [cited 2025-01-08]. Available from: https://doi.org/10.1145/3538704.

[18] Fouz B, Nora A, Hamda A, Mahra A, Ikuesan R. A methodical framework for conducting reconnaissance and enumeration in the ethical hacking lifecycle.

In: Proceedings of the 22nd European Conference on Information Warfare and Security (ECCWS). Aberystwyth (UK): Academic Conferences International; 2023. p. 54-64. [cited 2025-01-09]. Available from: https://www.academic-conferences.org/conferences/eccws/eccws-future-and-past/.

[19] Rowe D. BadBlood: Active Directory security test environment data population; 2020. [cited 2024-10-29]. [Internet]. San Francisco (CA): GitHub. Available from: https://github.com/davidprowe/BadBlood.

[20] Belkacem A, Houhamdi Z. Formal approach to data accuracy evaluation. Informatica. 2022;46(2):243-58. [Internet]. [cited 2025-01-08]. Available from: https://doi.org/10.31449/inf.v46i2.3027.

[21] Tait R. Hunting Impacket: part 3; 2024. [cited 2025-03-17]. [Internet]. Reston (VA): SnapAttack Blog. Available from: https://blog.snapattack.com/hunting-impacket-part-3-9c2680fd9265.

[22] Microsoft Corporation. MS-SAMR: common processing for group, alias and user; 2022. [cited 2025-03-15]. [Internet]. Redmond (WA): Microsoft. Available from: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-samr/fb25440b-269d-4220-99f2-8b29841cb1ce.

[23] Kali Linux Project. Pywerview package sources (kali/master branch); n.d. [cited 2025-03-18]. [Internet]. [place unknown]: GitLab. Available from: https://gitlab.com/kalilinux/packages/pywerview/-/tree/kali/master.

[24] Samba Project. rpcclient manual page; n.d. [cited 2025-03-17]. [Internet]. London: Samba.org. Available from: https://www.samba.org/samba/docs/4.17/man-html/rpcclient.1.html.

[25] Microsoft Corporation. Network access: restrict clients allowed to make remote calls to SAM; 2018. [cited 2024-06-22]. [Internet]. Redmond (WA): Microsoft. Available from: https://learn.microsoft.com/en-us/previous-versions/windows/it-pro/windows-10/security/threat-protection/security-policy-settings/network-access-restrict-clients-allowed-to-make-remote-sam-calls#related-events.

[26] Microsoft Corporation. MS-SAMR: SamrEnumerateUsersInDomain (Opnum 13); 2021. [cited 2025-03-17]. [Internet]. Redmond (WA): Microsoft. Available from: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-samr/6bdc92c0-c692-4ffb-9de7-65858b68da75.

[27] Microsoft Corporation. MS-SAMR: SamrEnumerateAliasesInDomain (Opnum 15); 2021. [cited 2025-03-17]. [Internet]. Redmond (WA): Microsoft. Available

from: https://learn.microsoft.com/en-us/openspecs/windows_protocols/ms-samr/ce340cff-edef-4356-ace0-33d2874d306b.

[28] Schroeder W. A guide to attacking domain trusts; 2017. [cited 2025-03-17]. [Internet]. [place unknown]: Harmj0y Blog. Available from: https://blog.harmj0y.net/redteaming/a-guide-to-attacking-domain-trusts.

[29] 0xdf Blog. SMB enumeration cheatsheet; 2024. [cited 2025-03-17]. [Internet]. [place unknown]: 0xdf. Available from: https://0xdf.gitlab.io/cheatsheets/smb-enum.

A SAMR/RPC Packet Fields

This appendix provides a description of the fields listed in the SAMR Request and Response Packet tables.

RPC Header (Transport Layer)

- **UUID (Interface ID):** A unique identifier for the interface being invoked, ensuring the correct binding to the RPC service.
- **Packet Type:** Specifies the type of packet (e.g., request, response, fault) used in communication.
- **Call ID:** A unique identifier for the specific RPC call, ensuring that responses match the correct requests.
- **Fragment Length:** Indicates the size of the packet fragment, used to handle large requests or responses.
- **Authentication Length:** Specifies the length of the authentication token, if included in the RPC packet.

SAMR Request Header

- **Function Code (OpNum):** Identifies the specific SAMR operation requested (e.g., user enumeration or group enumeration).
- **Access Mask:** Specifies the level of access permissions required for the requested operation.
- **Desired Access:** Represents the permissions requested by the client for the operation.
- **Context Handle:** An identifier for the context of the operation, used for stateful communications.

SAMR Request Body

- **Object Handle:** A reference to the queried or manipulated object (e.g., user, group).
- **Input Parameters:** Contains parameters sent by the client to specify the operation's context or target:
  - **Specific object identifiers:** Includes RIDs (Relative Identifiers) and SIDs (Security Identifiers).
  - **Domain/User/Group identifiers:** Identifies specific entities within the domain.
- **Output Parameters:** Contains data returned by the server in response to the request.

SAMR Response Header

- **Status Code:** Indicates the outcome of the operation (e.g., success, failure).
- **Context Handle (Returned):** An identifier for the context of the operation returned by the server.

- **OpNum (Echoed from Request):** The OpNum echoed back to confirm the specific SAMR function invoked.
- **Access Mask (Returned):** Specifies the permissions granted for the requested operation.

SAMR Response Body

- **Object Handle (Returned):** A reference to the object queried or manipulated in the operation.
- **Output Parameters:** Contains the data retrieved:
  - **Specific object data:** Includes RIDs (Relative Identifiers) and SIDs (Security Identifiers).
  - **Domain/User/Group details:** Provides detailed information about the queried entities.
- **Result Flags:** Indicates specific conditions or metadata about the result.
- **Error Information (if applicable):** Provides details about any errors encountered during the operation.

B Supplementary Resources

The GitHub repository that supports this research contains detailed data, including tool version numbers, OpNum capabilities, and comparative analysis details. The following are links to the relevant sections of the repository:

- **Main Page:** Summarises the objective of the project and outlines the laboratory architecture. It also lists the key repository artefacts and notes that the enumeration tool `samr-enum` itself is maintained separately in the `samr-enum` project. Access the main page here: `https://github.com/studylab1/SAMR-Enum-Lab/blob/main/README.md`.
- **Laboratory Setup:** Provides a complete specification of the Hyper-V laboratory, including IP subnets, domain-controller versions, and two-way transitive trust settings. The document is intended to enable for exact replication of the experimental conditions. Access it here: `https://github.com/studylab1/SAMR-Enum-Lab/blob/main/Laboratory_Setup.md`.
- **Tools Comparison:** Presents a comparative matrix of SAMR-enumeration tools, detailing their OpNum coverage, authentication methods, and data-parsing features. A separate section maps the SAMR field names observed in Wireshark to the terminology used in the official protocol documentation. Access it here: `https://github.com/studylab1/SAMR-Enum-Lab/blob/main/Tools_Comparison.md`.
- **Resources:** Provides structured access to all experimental artefacts, including full command listings with parameters and downloadable PCAPNG traffic captures for each tested scenario. The page also documents the exact SAMR OpNums used during the testing. Additionally, it contains a section detailing PowerShell AD module cmdlets used for tool comparison. Access it here: `https://github.com/studylab1/SAMR-Enum-Lab/blob/main/Resources.md`.
- **Experiment:** Documents the final test scenarios, including `samr-enum` command sequences, network capture links, and output logs. It shows how changes in trust configurations and ACLs influence the domain controller's responses. Access it here: `https://github.com/studylab1/SAMR-Enum-Lab/blob/main/Experiment.md`.

C Supplementary Tables

Table 5. Descriptions of SAMR Enumeration OpNums

OpNum	Name	Description
0	SamrConnect	Establishes an initial connection to the SAM server, allowing subsequent operations to be performed.
1	SamrCloseHandle	Closes an open handle to a SAM object, releasing the associated resources.
3	SamrQuerySecurityObject	Retrieves security information for a specified SAM object, such as permissions and access control details.
5	SamrLookupDomainInSamServer	Resolves a domain name to its corresponding SID within the SAM server.
6	SamrEnumerateDomainsInSamServer	Lists all domains managed by the SAM server.
7	SamrOpenDomain	Opens a handle to a specific domain for further operations.
8	SamrQueryInformationDomain	Retrieves specific information about a domain, such as security policies or account statistics.
11	SamrEnumerateGroupsInDomain	Retrieves a list of group names and their relative identifiers (RIDs) within a specific domain.
13	SamrEnumerateUsersInDomain	Retrieves user account names and their RIDs within a specific domain.
15	SamrEnumerateAliasesInDomain	Lists alias groups (local groups) within a domain along with their RIDs.
16	SamrGetAliasMembership	Shows alias memberships for a specific user or SID.
17	SamrLookupNamesInDomain	Converts account names into SIDs within a domain.
18	SamrLookupIdsInDomain	Maps SIDs back to account names.
19	SamrOpenGroup	Opens a handle to a specific group for further operations.
20	SamrQueryInformationGroup	Retrieves information about a specific group in a domain.
25	SamrGetMembersInGroup	Retrieves the list of members' RIDs for a given group.
27	SamrOpenAlias	Opens a handle to a specific alias (local group) for further operations.
28	SamrQueryInformationAlias	Retrieves detailed information about an alias (local group), such as its description and member statistics, depending on the requested information level.
33	SamrGetMembersInAlias	Retrieves a list of members for a specified alias (local group).
34	SamrOpenUser	Opens a handle on a specific user account for further operations.
36	SamrQueryInformationUser	Retrieves detailed information on a specific user account.
39	SamrGetGroupsForUser	Lists all group memberships for a specified user.
40	SamrQueryDisplayInformation	Provides display information (e.g., names) for a set of domain accounts, such as users or groups.

41	SamrGetDisplayEnumerationIndex	Retrieves the display index for paginated enumerations.
44	SamrGetUserDomainPasswordInformation	Retrieves select password policy information for a user without requiring a domain handle.
46	SamrQueryInformationDomain2	Retrieves display information (e.g., names, account descriptions) for domain accounts. This operation is similar to SamrQueryDisplayInformation, but allows for extended querying.
47	SamrQueryInformationUser2	Provides additional detailed information about a user account, similar to SamrQueryInformationUser.
48	SamrQueryDisplayInformation2	Retrieves display information for domain accounts (e.g., users, groups) in a paginated format.
49	SamrGetDisplayEnumerationIndex2	Retrieves the display index for paginated enumerations in scenarios requiring extended enumeration.
51	SamrQueryDisplayInformation3	Enables detailed and filtered queries for large-scale user, group, or machine account enumeration.
56	SamrGetDomainPasswordInformation	Retrieves password policy information for the domain.
57	SamrConnect2	Establishes a connection to the SAM server, specifically optimised for certain environments or use cases.
62	SamrConnect4	Establishes a connection to the SAM server using extended security negotiation parameters, offering additional features compared to SamrConnect.
64	SamrConnect5	Establishes a connection to the SAM server for domain enumeration and lookup.
65	SamrRidToSid	Converts a relative identifier (RID) to a security identifier (SID) within the domain.
74	SamrValidateComputerAccountReuseAttempt	Validates whether a computer account reuse attempt complies with domain policies, returning a status code that indicates if reuse is permitted.
77	SamrAccountIsDelegatedManagedServiceAccount	Determines if a computer account is a Delegated Managed Service Account (gMSA), returning a flag indicating its managed service status.

Table 6. Comparison of `samr-enum`, `rpcclient`, and `samrdump`

Feature	**`samr-enum` (This work)**	**`rpcclient` (Samba)**	**`samrdump` (Impacket)**
OpNum Coverage	Extended coverage of enumeration calls: users, global groups, aliases, and membership (via multiple `Samr*` methods). Also queries selected domain information (e.g., policy).	Broad command set for SAMR (users, groups, aliases), but each must be invoked manually. Supports almost all SAMR calls through separate commands.	Limited coverage: enumerates users, but does not enumerate group memberships by default. Could miss inter-forest references.
Data Parsing and & Accuracy	Comprehensive: retrieves all accounts in the target domain(s), including references between domains (for example, foreign security principles). Ensures no objects are missed by iterating through all SAMR enumeration handles until completion.	Accurate per query, but completeness depends on the user executing all necessary commands. Output is raw and requires parsing; risk of missing data if certain groups or aliases are not queried.	Focuses on user lists and attributes. However, it does not by default enumerate membership details or cross-forest references.
Authentication Protocol & Support	Supports standard Windows auth methods: NTLM and Kerberos (if provided a ticket or using domain credentials). Can utilise null sessions where policies allow. Handles encryption/signing requirements automatically via libraries.	Supports NTLM authentication (password or hash) and can leverage Kerberos (`-use-kerberos`). Also, allows null/builtin accounts if permitted. SMB signing/encryption can be enabled via flags. Credentials must be supplied for each invocation.	Supports NTLM (explicit creds or pass-the-hash) and optional Kerberos (Impacket's `-k` flag). Can attempt anonymous binding if not restricted. Uses the underlying Impacket SMB stack for signing or encryption as required.

D Tool Versions and Specifications

The following table provides version numbers and implementation notes for the tools evaluated during this research.

Table 7. Versions and Specifications of Evaluated Tools

Tool Name	**Version**	**Additional Notes**
Net	Built-in	Windows 11 Enterprise x86-64 (version 23H2, OS build 22631.4317).
Enum4linux	0.9.1	A legacy enumeration tool commonly used for SMB and NetBIOS reconnaissance. While functional, it lacks updates and advanced features found in modern tools.
Enum4linux-ng	1.3.4	A modernised version of Enum4linux, with improved enumeration capabilities, formatting, and support for newer SMB versions.
PowerShell	1.0.1.0	ActiveDirectory Module. Windows 11 Enterprise x86-64 (version 23H2, OS build 22631.4317). Cmdlets do not use the SAMR protocol but rely on MS-NNS and MS-NMF.
SharpHound	2.5.9	Part of the BloodHound project. Designed for AD data collection and attack path visualisation.
Metasploit	6.4.41 dev	Penetration testing framework with SAMR modules to enumerate users and computers on remote domain controllers.
CrackMapExec	6.1.0 – John Wick	Post-exploitation and lateral movement tool with SAMR-based enumeration for identifying domain objects.
Impacket	0.12.0	Python library with tools like `samrdump.py` and `net.py` for SAMR enumeration.
rpcclient	4.15.13	Samba suite command-line tool for MS-RPC interaction over SMB. Supports user, group, and domain queries.
samr-enum	1.2.0	Developed as part of this thesis. Uses Impacket to enumerate users, groups, computers, and policies. Supports NTLM and Kerberos. Logs OpNums and supports TXT, CSV, and JSON output.

E Detailed OpNum Coverage by Tools

The evaluation results in this section are based on cross-forest SAMR requests. Table 8 details the connection operations (OpNums 0 `SamrConnect`, 57 `SamrConnect2`, 62 `SamrConnect4`, and 64 `SamrConnect5`) and handle management (OpNums 1 `SamrCloseHandle`, 7 `SamrOpenDomain`, 19 `SamrOpenGroup`, 27 `SamrOpenAlias`, and 34 `SamrOpenUser`).

Table 9 presents the domain enumeration and query operations (OpNums 6 `SamrEnumerateDomainsInSamServer`, 5 `SamrLookupDomainInSamServer`, 8 `SamrQueryInformationDomain`, 46 `SamrQueryInformationDomain2`, and 3 `SamrQuerySecurityObject`) alongside group and alias operations (OpNums 11 `SamrEnumerateGroupsInDomain`, 15 `SamrEnumerateAliasesInDomain`, 20 `SamrQueryInformationGroup`, 28 `SamrQueryInformationAlias`, 25 `SamrGetMembersInGroup`, 33 `SamrGetMembersInAlias`, 17 `SamrLookupNamesInDomain`, 18 `SamrLookupIdsInDomain`, and 16 `SamrGetAliasMembership`).

Table 10 summarises user operations (OpNums 13 `SamrEnumerateUsersInDomain`, 36 `SamrQueryInformationUser`, 47 `SamrQueryInformationUser2`, 39 `SamrGetGroupsForUser`, and 44 `SamrGetUserDomainPasswordInformation`), display and lookup operations (OpNums 40 `SamrQueryDisplayInformation`, 48 `SamrQueryDisplayInformation2`, 51 `SamrQueryDisplayInformation3`, 41 `SamrGetDisplayEnumerationIndex`, 49 `SamrGetDisplayEnumerationIndex2`, and 65 `SamrRidToSid`), and password and policy operations (OpNums 56 `SamrGetDomainPasswordInformation`, 74 `SamrValidateComputerAccountReuseAttempt`, and 77 `SamrAccountIsDelegatedManagedServiceAccount`), thereby highlighting the compatibility and supported functionality across the evaluated tools.

● - supported
❍ - not supported

Table 8. Connection Operations and Handle Management

	Connection Operations				**Handle Management**				
Tool \ OpNum	**0**	**57**	**62**	**64**	**1**	**7**	**19**	**27**	**34**
Net	❍	❍	❍	❍	❍	❍	❍	❍	❍
Enum4linux	❍	❍	❍	❍	❍	❍	❍	❍	❍
Enum4linux-ng	❍	❍	❍	❍	❍	❍	❍	❍	❍
PowerShell	❍	❍	❍	❍	❍	❍	❍	❍	❍
SharpHound	❍	❍	❍	●	●	●	❍	●	❍
Metasploit	●	❍	❍	❍	●	●	❍	❍	❍
CrackMapExec	●	●	❍	❍	●	●	❍	❍	●
Impacket	●	●	❍	❍	●	●	●	●	●
rpcclient	●	❍	❍	●	❍	●	●	●	●
samr-enum	●	❍	❍	❍	●	●	●	●	●

Table 9. Domain Enumeration and Query Operations, and Group and Alias Operations

Tool \ OpNum	**Domain Enumeration and Query Operations**					**Group and Alias Operations**								
	6	**5**	**8**	**46**	**3**	**11**	**15**	**20**	**28**	**25**	**33**	**17**	**18**	**16**
Net	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
Enum4linux	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
Enum4linux-ng	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
PowerShell	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
SharpHound	●	●	❍	❍	❍	❍	●	❍	❍	❍	●	❍	❍	❍
Metasploit	●	●	●	❍	❍	❍	❍	❍	❍	❍	❍	●	❍	❍
CrackMapExec	●	●	❍	●	❍	❍	●	❍	❍	❍	❍	❍	❍	❍
Impacket	●	●	❍	❍	❍	●	●	❍	❍	●	●	●	●	●
rpcclient	●	●	●	❍	●	●	●	●	❍	●	●	●	●	●
samr-enum	●	●	❍	●	❍	●	●	●	●	●	●	●	●	❍

Table 10. User Operations, Display and Lookup Operations, and Password and Policy Operations

Tool \ OpNum	**User Operations**					**Display and Lookup Operations**						**Domain Enumeration and Query Operations**		
	13	**36**	**47**	**39**	**44**	**40**	**48**	**51**	**41**	**49**	**65**	**56**	**74**	**77**
Net	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
Enum4linux	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
Enum4linux-ng	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
PowerShell	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
SharpHound	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
Metasploit	●	❍	❍	❍	❍	❍	❍	❍	❍	❍	●	❍	❍	❍
CrackMapExec	●	❍	●	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍
Impacket	●	❍	●	●	❍	❍	❍	❍	❍	❍	●	❍	❍	❍
rpcclient	●	●	❍	●	●	●	●	●	●	❍	❍	●	❍	❍
samr-enum	●	❍	●	●	❍	❍	❍	❍	❍	❍	❍	❍	❍	❍

F SAMR Session Workflow for samr-enum

F.1 Purpose

This appendix documents the exact SMB/RPC dialogue shown in Figure 3 and generated by `samr-enum` v2.0.1 during the command:

```
python samr-enum.py target=bdc1.domain-b.local \
    username=enum-a password=LabAdm1! \
    enumerate=account-details acl user=adminb1
```

F.2 Environment

Client Ubuntu 22.04.5, Python 3.11, `samr-enum` commit `c6175ae`.

Server Windows Server 2022 DC (`bdc1.domain-b.local`, forest functional level 2016).

Authentication NTLMv2 over SMB 3 with signing.

F.3 Legend

- **OpNum**: ordinal number of the RPC procedure (per MS-SAMR).
- **Handle Srv/Dom**: context handles returned by the corresponding `SamrConnect5` and `SamrOpenDomain` calls.
- **SMB2 Session Setup**: NTLM challenge–response exchange.
- **`IPC$`**: special share used for named-pipe RPC (e.g. `\PIPE\samr`).

F.4 Key OpNums

Table 11 provides short descriptions of the OpNums used in the communication.

F.5 Sequence Description

After SMB negotiation the client binds to the SAMR interface over the `\\IPC$` named pipe and issues `SamrConnect5` (OpNum 64) to acquire *ServerHandle*. With this handle, it enumerates the server's SAM domains, resolves the SID of `domain-b`, and finally opens a *DomainHandle* (OpNum 7) using minimal access (`DOMAIN_LIST_ACCOUNTS`). This establishes the context for the subsequent user, group, and alias enumeration steps (omitted here for brevity).

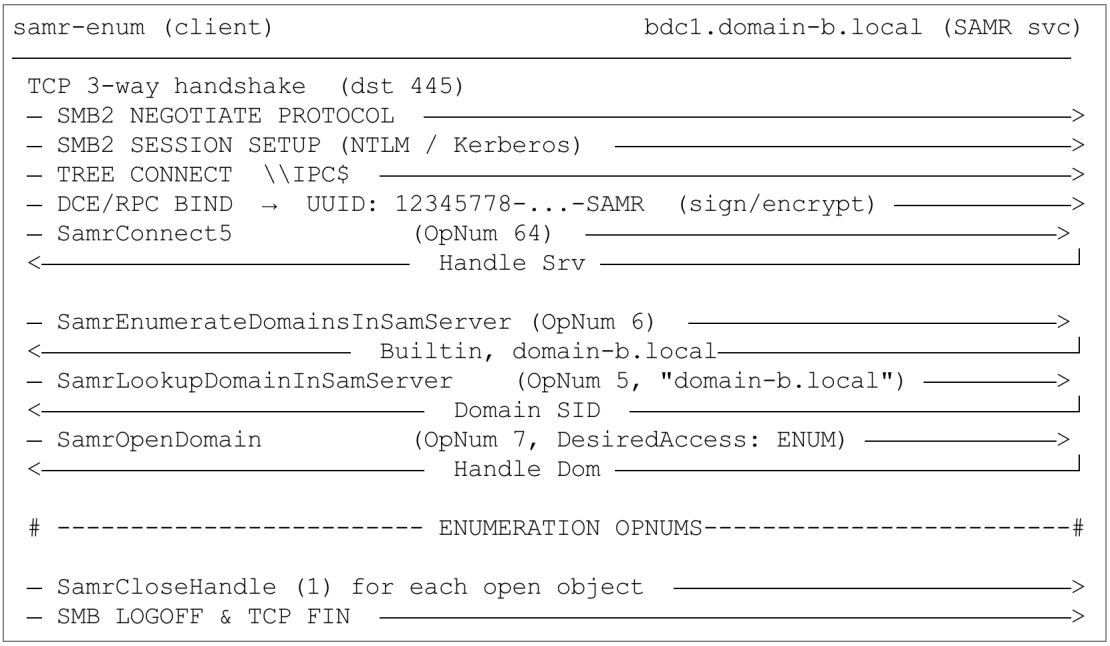

Figure 3. SAMR Session Workflow

OpNum	RPC Method	Purpose
64	`SamrConnect5`	Obtain a *ServerHandle* for subsequent SAM operations.
6	`SamrEnumerateDomainsInSamServer`	Retrieve the list of SAM domains available on the server.
5	`SamrLookupDomainInSamServer`	Resolve the SID of a specific domain by name.
7	`SamrOpenDomain`	Open a *DomainHandle* with the least privilege needed for enumeration.
1	`SamrCloseHandle`	Release any SAMR context handle (user, group, domain, or server).

Table 11. Core RPC operations used in the connection phase

F.6 Artifacts

- Packet capture: https://github.com/studylab1/SAMR-Enum-Lab/raw/refs/heads/main/Resources/acl_0.7_adminb1.pcapng
- Verbose tool output: https://github.com/studylab1/SAMR-Enum-Lab/raw/refs/heads/main/Resources/acl_0.7_adminb1.txt

Glossary

Trust

A *trust* is a formal security relationship between two Active Directory domains or forests. Each trust is characterised by three parameters: *direction* (one-way or two-way), *transitivity* (whether the relationship extends beyond the two partners) and *scope of authentication* (forest-wide or selective). When a trust is in place, credentials issued in one domain or forest can be recognised in the other in accordance with these parameters, enabling cross-boundary access while day-to-day administration remains isolated.

Domain and Forest

A *domain* is a logical collection of directory objects (for example, users and computers) that share a common database and security policies. A *forest* is a set of one or more domains that share a common schema, configuration and global catalogue. Domains in the same forest trust each other automatically; relationships between separate forests rely on explicit trusts.

Forest-wide Authentication

The default trust mode in which any authenticated user from the trusting forest may present credentials in the trusted forest. Unless additional Access Control List (ACL) hardening is applied, Server Message Block Remote-Procedure-Call (SMB/RPC) services such as SAMR return almost the full directory listing to such callers.

Selective Authentication in Active Directory

A trust mode that blocks authentication requests from an external domain or forest unless the caller is granted the explicit *Allow to authenticate* permission on the target domain. Without this right, each SAMR query issued across the trust is rejected, even if other read permissions exist.

Foreign Security Principal (FSP)

A placeholder object that represents an account from another forest or domain once it is added to a domain-local group. The FSP stores the Security Identifier (SID) of the original account and is created automatically. During cross-forest SAMR enumeration, FSP objects remain hidden if the corresponding domain-local group is not visible to the querying principal.

Licence

Non-exclusive licence to reproduce thesis and make thesis public

I, Juri Djomin,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to reproduce, for the purpose of preservation, including for adding to the DSpace digital archives until the expiry of the term of copyright,

   Evaluating SAMR Enumeration Exposure in Multi-Forest Active Directory,

   supervised by Dr. Raimundas Matulevičius and Juhan Aasaru, MSc.

2. grant the University of Tartu a permit to make the thesis specified in point 1 available to the public via the web environment of the University of Tartu, including via the DSpace digital archives, under the Creative Commons licence CC BY-NC-ND 3.0, which allows, by giving appropriate credit to the author, to reproduce, distribute the work and communicate it to the public, and prohibits the creation of derivative works and any commercial use of the work until the expiry of the term of copyright.

3. am aware of the author retains the rights specified in p. 1 and 2.

4. confirm that granting the non-exclusive licence does not infringe other persons' intellectual property rights or rights arising from the personal data protection legislation.

Juri Djomin

15 May 2025