Blockchain-Enhanced Network Scanning and Monitoring (BENSAM) Framework

Syed Wasif Abbas Hamdani , Kamran Ali , Zia Muhammad *

Posted Date: 26 November 2025

doi: 10.20944/preprints202505.1028.v2

Keywords: blockchain; network security; cybersecurity framework; vulnerability scanning; smar contracts; policy enforcement; iot security; advanced threat detection; tamper-proof auditing; decentralized verification

 doi:10.20944/preprints202505.1028.v2

Article

Blockchain-Enhanced Network Scanning and Monitoring (BENSAM) Framework

Syed Wasif Abbas Hamdani [1], Kamran Ali [2] and Zia Muhammad [3,*]

[1] National University of Sciences and Technology (NUST), Islamabad, 46000, Pakistan
[2] Department of Computer Science, North Dakota State University, Fargo, ND 58102, US
[3] Department of Computing, Design, and Communication, University of Jamestown, Jamestown, ND 58405, USA
* Correspondence: zia.muhammad@uj.edu

Abstract

In recent years, the convergence of advanced technologies has enabled real-time data access and sharing across diverse devices and networks, significantly amplifying cybersecurity risks. For organizations with digital infrastructures, network security is crucial for mitigating potential cyber-attacks. They establish security policies to protect systems and data, but employees may intentionally or unintentionally bypass these policies, rendering the network vulnerable to internal and external threats. Detecting these policy violations is challenging, requiring frequent manual system checks for compliance. This paper addresses key challenges in safeguarding digital assets against evolving threats, including rogue access points, man-in-the-middle attacks, denial-of-service (DoS) incidents, unpatched vulnerabilities, and AI-driven automated exploits. We propose a Blockchain-Enhanced Network Scanning and Monitoring (BENSAM) Framework, a multi-layered system that integrates advanced network scanning with a structured database for asset management, policy driven vulnerability detection, and remediation planning. Key enhancements include device profiling, user activity monitoring, network forensics, intrusion detection capabilities, and multi-format report generation. By incorporating blockchain technology, leveraging immutable ledgers and smart contracts, the framework ensures tamper-proof audit trails, decentralized verification of policy compliance, and automated real-time responses to violations such as alerts, actual device isolation is performed by external controllers like SDN or NAC systems. The research provides a detailed literature review on blockchain applications in domains like IoT, healthcare, and vehicular networks. A working prototype of the proposed BENSAM framework was developed that demonstrates end-to-end network scanning, device profiling, traffic monitoring, policy enforcement, and blockchain-based immutable logging. This implementation is publicly released and is available on GitHub. It analyzes common network vulnerabilities (e.g. open ports, remote access, disabled firewalls), attacks (including spoofing, flooding, DDoS), and outlines policy enforcement methods. Moreover, the framework anticipates emerging challenges from AI-driven attacks such as adversarial evasion, data poisoning, and transformer-based threats, positioning the system for future integration of adaptive mechanisms to counter these advanced intrusions. This blockchain enhanced approach streamlines security analysis, extends framework for AI threat detection with improved accuracy, reduces administrative overhead by integrating multiple security tools into a cohesive, trustworthy, reliable solution.

Keywords: blockchain; network security; cybersecurity framework; vulnerability scanning; smart contracts; policy enforcement; iot security; advanced threat detection; tamper-proof auditing; decentralized verification

1. Introduction

In recent years, with the enriched convergence of the latest technologies whereby users can access, share, and store data through diverse devices and networks in real-time, there has been a significant increase in cybersecurity risks. Organizations face a growing challenge in safeguarding their digital

 doi:10.20944/preprints202505.1028.v2

assets against a spectrum of evolving threats. Network infrastructures are particularly vulnerable, with attackers having sophisticated techniques, such as rogue access points and man-in-the-middle attacks to compromise Wi-Fi networks and steal user credentials. Attackers can successfully impersonate corporate access to steal user credentials, like usernames and passwords, by setting up rogue APs. Denial of service (DoS) and distributed denial of service (DDoS) attacks remain a persistent threat, disrupting network availability by overwhelming target systems with malicious traffic [1].

Enterprise systems can experience severe security threats in the event that vulnerabilities are left unnoticed, leaving significant data exposure to attackers. For enterprises, this can result in prolonged downtime of the systems, which can cause a huge loss of productivity and revenue. The absence of appropriate security measures and particular controls in place to protect sensitive data can lead to an attack. Certain attacks are passive and include observing and data-stealing, whereas active attacks are intended to destroy or corrupt the data and the network infrastructure. Therefore, if the proper security measures are not in place, then the data and networks are vulnerable to any of these attack [2]. The organizations take appropriate measures and install different software to protect their data and systems from different attacks. These security measures do not protect the organizations completely, as every day new attacks are introduced and vulnerabilities are found in security software; hence, this software is required to be patched on time. Any business that is seriously concerned about security can particularly focus on patch management, enhanced cybersecurity measures within the organization, and the possibility of considering cyber insurance to transfer residual risk in case of a cyber attack [3]. A critical vulnerability lies in unpatched software, which can expose enterprise systems to severe security breaches, data breaches, and financial losses. Effective patch management is therefore paramount, yet it presents significant challenges due to the rapid discovery of new vulnerabilities and the complexity of maintaining up-to-date systems [4]. It is observed that security breaches in an enterprise network are mostly caused by the absence of updated patches in an enterprise's operating system (OS) along with other applications. For instance, it requires a large amount of commitment and time to update all of the systems whenever a patch update is available in the IT department [5]. Moreover, the rise of advanced threat actors, who leverage artificial intelligence and machine learning to automate attacks and evade traditional security measures, necessitates a multi-layered security approach [6].

For organizations that have digital infrastructures in place, network security is an essential consideration in coping with certain potential cyberattacks. As it is very difficult to completely secure the organization's network just by installing the software, the organizations also take some other appropriate precautionary measures to ensure the network's security. For this purpose, an organization can make and implement policies to guide and instruct its employees to secure its systems and data. Policies and procedures of any organization must be instigated in conformance with the enterprise's culture and internal practices. This assists in the rapid implementation and adoption of procedures interrelated to the current security standards. James et al., as cited by Taylor [7], state that "The essence of information security is all about people, processes, and controls: the heart of successful security is not pure technology, but a team of well-trained employees who are prepared to use technology as a tool to implement and manage effective IT controls." [8].

Beyond technical controls, organizations must foster a robust security culture through comprehensive policies and employee training. Security policies should be tailored to the organization's unique requirements and aligned with industry best practices, such as those outlined in the NIST Cybersecurity Framework [9]. Some precautions must be taken in the form of a policy to secure the system from intruders and getting hacked. The fundamental elements of an organization's security policy can be about the status of the firewall as enabled and the remote access is disabled. To avoid attackers entering the system through a backdoor, it is necessary to monitor all ports when the system is connected to the Internet. Zenmap [10] can be used to list the open ports on a system along with their associated services and version information. Organizational security policies can be enforced through various mechanisms, ranging from simple employee notifications to more sophisticated automated systems. Traditionally, policies were disseminated through manual instructions, but modern

 doi:10.20944/preprints202505.1028.v2

approaches emphasize automated enforcement. For instance, endpoint detection and response (EDR) agents can be deployed on network systems to monitor user activity and identify policy violations in real-time [11]. Operating system (OS) hardening is another enforcement method, although it must be balanced against user productivity [12]. The hardening of the Operating System (OS) can be another option to enforce the policy but it will also restrict the users to perform some of their management tasks. Davide et al. tried to enforce a network security policy through software-defined networking (SDN) in an organization [13]. However, it requires a change to shift the whole network to SDN by switching all network devices.

A reliable method to audit and evaluate an organization's policies is a network scanner as most of the scanners of network scanners already can check different security aspects like firewall status, shared directories, server recognition, OS detection, remote access detection, and virtual machine recognition in one way or another. We proposea blockchain-enhanced network scanning and monitoring (BENSAM) framework linked with a structured database that enables us to scan assets and asset groups, view vulnerable assets and their complete security information, schedule scans, e-mail scan reports, and take appropriate action to safeguard our assets based on the remediation solutions provided. It can help to recognize available network devices, and services, identify any filtering systems in place and operating Systems (OSs) in use, and discover vulnerabilities to protect the network from potential attacks. Since security is a major concern for organizations, network scanning enables a security analyst to detect devices present over the network that could be more likely to be exploited by hackers. We also proposed some advanced features in the scanner with which its capability can be greatly improved. These features are network forensics, device profiling, user activity monitoring, IDS capability, and report generation of multiple types.

Blockchain technology, known for its decentralized and immutable ledger, offers a transformative approach to enhancing network security and policy compliance in digital enterprises. By integrating blockchain into network security systems, organizations can ensure that logs of network activities and policy violations are tamper-proof, providing a reliable audit trail for forensic analysis and compliance verification. Furthermore, blockchain's smart contracts enable automated enforcement of security policies by generating compliance decisions when violations are detected such as triggering alerts. Actual enforcement actions including device isolation are carried out by external controllers such as SDN systems.This reduces human error, enhances trust in the system, and aligns with the need for a multi-layered security strategy to combat evolving cyber threats. This article presents a comprehensive analysis of organizational network security challenges, identifies critical gaps in existing scanning tools, and introducesa network scanning and monitoring framework framework. By integrating policy-driven scanning, forensic capabilities, and blockchain-based logging, this research outlines essential design features for scalable, compliant, and secure network monitoring systems. It also extends the security model to consider emerging AI-driven attacks, which increasingly bypass traditional defenses and demand adaptive detection mechanisms. The major contributions of this article are summarized below:

1. A detailed overview of organizational network security challenges and the role of policy enforcement in mitigating internal and external threats.
2. Identification of key limitations in existing network scanners regarding policy violation detection, auditability, and continuous compliance assurance.
3. Proposed a policy-aware network scanning and monitoring framework (BENSAM) that integrates device profiling, user activity monitoring, traffic analysis, and network forensics into a unified architecture.
4. Formulation of a structured database model supporting asset-based scanning, scheduled execution, report generation, and vulnerability remediation planning.
5. Integration of a permissioned blockchain (Hyperledger Fabric) to enable immutable logging, secure audit trails, and decentralized verification of policy compliance through smart contracts.

 doi:10.20944/preprints202505.1028.v2

6. Development and public release of a working prototype demonstrating the end-to-end operation of BENSAM, including scanning, profiling, policy enforcement, blockchain anchoring, and automated report generation.
7. Comparative analysis of BENSAM against traditional enforcement and monitoring techniques, emphasizing its scalability, automation, and trustworthiness.

In addition, the framework is designed to be extensible for future integration of AI-driven threat detection and adaptive scanning capabilities, which are identified as potential directions for further research. The rest of the paper is organized as follows: Section II reviews related literature and applications. Section III discusses network vulnerabilities, common and emerging AI-driven attacks, and policy enforcement methods. Section IV describes the core features of a conventional network scanner, while Section V highlights the security aspects that can be assessed through network scanning. Section VI presents the proposed blockchain-enhanced network scanning and monitoring framework, its architecture, components, algorithm, implementation, and preliminary validation. Section VII outlines open challenges and future research directions, including AI-driven threat detection. Finally, Section VIII concludes the paper.

2. Literature Review

Blockchain has emerged as a transformative technology for enhancing security, privacy, and trust in the networks and related domains. Recent research has explored blockchain-based solutions across diverse applications, including healthcare, vehicular networks, wireless sensor systems, smart homes, drones, and satellite communications.

A comprehensive survey by Khan et al. [14] highlights the potential of blockchain to mitigate IoT vulnerabilities, providing a state-of-the-art review of attacks and countermeasures. Similarly, Singh et al. [15] discuss blockchain-related security attacks and propose solutions for distributed IoT environments. Fotohi and Aliee [16] proposed authentication mechanisms using SHA-256 to secure large-scale IoT communication, improving scalability. Additionally, Rani et al. [17] explored IoT-based software-defined networks, introducing a blockchain-based framework to strengthen data integrity. Hsiao and Sung [18] also emphasized blockchain's role in securing wireless sensor networks against intrusions.

The healthcare domain has seen significant adoption of blockchain for data security. Khan et al. [19] developed a blockchain-based framework for safeguarding sensitive data in healthcare IoT systems. Sharma et al. [20] focused on improving the security of medical big data using blockchain technology, ensuring data confidentiality and resilience. Furthermore, Nguyen et al. [21] introduced BEdgeHealth, a decentralized blockchain-enabled edge architecture for IoMT (Internet of Medical Things) networks.

Vehicular Ad Hoc Networks (VANETs) present unique security challenges. Tandon et al. [22] proposed D-BLAC, a dual blockchain architecture for authentication and communication in VANETs. Grover [23] provided a comprehensive review of blockchain's role in securing VANETs, while Bendiab et al. [24] examined autonomous vehicles, integrating blockchain and artificial intelligence to address critical security issues in intelligent transportation systems.

Farooq et al. [25] explored blockchain-enabled smart home networks enhanced with fused machine learning techniques, addressing cybersecurity risks in residential IoT ecosystems. Hizal et al. [26] similarly studied blockchain-based security mechanisms in IDS research centers, highlighting practical applications in IoT environments.

Blockchain has also been applied in drone and satellite communications. Lv et al. [27] analyzed the use of blockchain to safeguard the privacy of drone big data. Li et al. [28] proposed blockchain-based methods to secure satellite networks, ensuring both efficiency and robustness of data transmissions. Beyond classical applications, researchers are exploring advanced approaches. Dhar et al. [29] introduced a hybrid model combining blockchain with quantum cryptography to secure IoT devices, aiming to achieve future-proof security mechanisms.

 doi:10.20944/preprints202505.1028.v2

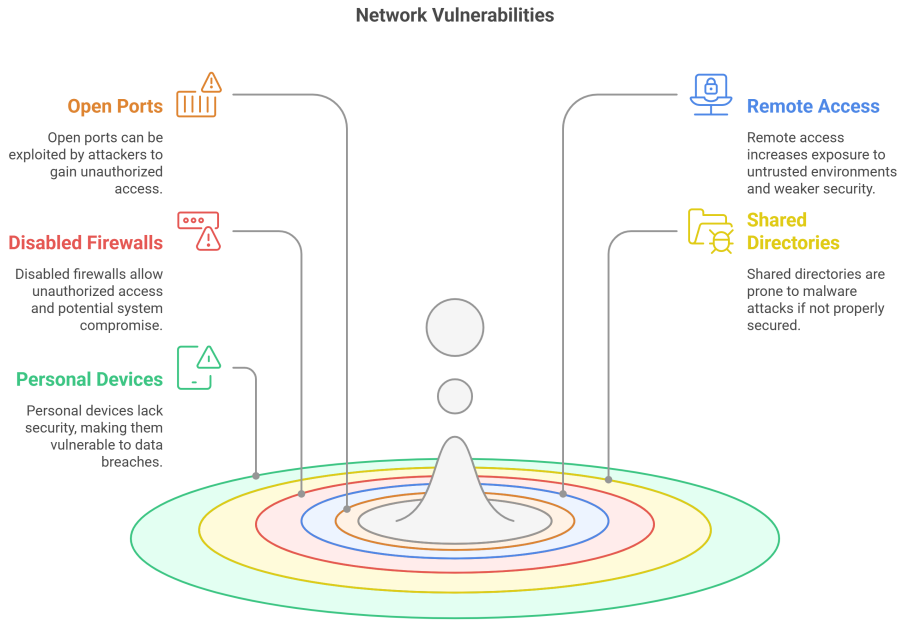

Figure 1. Illustration of Common Network Vulnerabilities Including Open Ports, Remote Access, Disabled Firewalls, and Shared Directories in Enterprise Networks.

While several prior works have explored blockchain-based logging and compliance frameworks, e.g., hyperledger-backed append only SIEM logs, certificate transparency style auditability mechanisms, and tamper-evident log structures, these approaches primarily focus on ensuring data integrity and traceability. Although these frameworks demonstrate the benefits of blockchain for auditability, they generally lack integration with real-time network scanning and automated policy enforcement. In contrast, the proposed BENSAM framework integrates real-time network scanning with blockchain auditing and policy enforcement. It uniquely integrates automated scanning results with smart contract–driven compliance verification, supports hybrid storage combining on-chain hashes with off-chain detailed logs, and enables verifiable audit trails across multiple scanning agents.

3. Network Vulnerabilities, Attacks, and Policy Enforcement Methods

An enterprise network may have various weaknesses that can cause different attacks to disrupt organizational services or steal sensitive information. To avoid such types of threats, organizations define their procedures and policies to reduce vulnerabilities. The implementation of these procedures is an administrative task that can be accomplished using different mechanisms. In this section, various network vulnerabilities, policy enforcement methods, and common attacks are discussed.

3.1. Common Network Vulnerabilities

All operating systems (OS) have built-in certain essential security measures that guard against various vulnerabilities, however, any change in the configurations to meet user requirements can make the OS vulnerable without being noticed by the organization. Figure 1 highlights typical vulnerabilities in enterprise networks that can be easily exploited by attackers like open ports, remote access, shared directories, and disabled firewalls. This demonstrates the points that BENSAM aims to monitor and secure. The following is a brief description of some of these common vulnerabilities.

3.1.1. Open Ports

In computer networks, an open port is a communication endpoint for accepting incoming connections, usually used by the server applications to cope with requests from remote hosts or clients.

 doi:10.20944/preprints202505.1028.v2

Nevertheless, open ports can also accept connections by malevolent clients if these are not protected carefully by exposing prospective weaknesses in the server-side software to remote exploitation. Such an intrinsic vulnerability has always escorted the practice of open ports throughout the account of network services, thus, opening the doors for huge numbers of severe network attacks including TCP SYN flooding attacks. [30]. Smartphone OS also inherits the support of open ports, and since smart devices are significantly dissimilar from the traditional server machines' availability and performance guarantees, therefore, limited information is available about how smart device applications utilize open ports and what the security consequences accordingly [31].

3.1.2. Remote Access

Remote access can be defined as the ability of an enterprise's users to access its office's computing resources from different locations other than the organization's premises. This provision of accessing resources remotely is increasingly common due to the widespread accessibility of smart devices and Internet access. An extensive selection of client hosts across the organization along with several hosts outside the enterprise's control are also reachable through remote access and these hosts usually have weaker safety measures e.g. physical security controls, no corporate-level antivirus tools and firewalls in place. as compared to the systems of an Enterprise. Similarly, organizations do not manage several devices and mostly remote communications are done over untrusted channels. It is quite possible that the remote devices of the client may be used in hostile environments that may not be configured appropriately.

3.1.3. Disabled Firewall

Presently, the protection of data and sensitive information has become a major challenge. Nearly, all the private and public institution's secured data is connected to the internet for diverse purposes. The attackers have many more occasions for getting access to this sensitive information through the Internet. Attacking against a particular network and digital infrastructure is comparatively easy since any network can be accessed from anywhere in the world via the Internet. Therefore, a firewall is an essential and integral part of the organization for protecting systems [32]. Firewall blocks unauthorized attempts to gain control or crash the systems. However, if the firewall is disabled then these systems become vulnerable which can result in numerous unmanaged exceptions. Additionally, this scenario is particularly favorable for the attackers to scan the entire system and may set backdoors without being noticed by the system user which can result in system compromise.

3.1.4. Shared Directories

Over the enterprise network, there can be a commonplace in the digital infrastructure where most of the common resources are shared within the organization. These resources can be accessed by different employees as and when required. Such shared directories can be vulnerable to various cyber threats because a shared directory is accessible over the network therefore, any malware could be placed which can compromise the system and thus overall network. In order to prevent such attacks, it is essential for an organization to keep an eye on shared directories over the network and make them accessible to authorized users only. Advanced IP scanner [33] has the ability to detect shared folders.

3.1.5. Personal Devices

Personal devices such as smartphones, tablets, iPADs, laptops. while connected to organizational networks are vulnerable to various cyber-attacks as these devices usually lack proper security software for their protection. These devices can have viruses or worms and can copy or transfer data against an organization's policy. Bringing personal devices to the organization is the death of perimeter security provided by the organization's border firewall.

 doi:10.20944/preprints202505.1028.v2

3.2. *Common and Emerging Attacks via Network*

Attackers gained sufficient skills over the years to find system vulnerabilities, and also utilize advanced intrusion mechanisms that are hard to identify and trace. Network security tools are not only suitable for recognizing security violations successfully but also useful in monitoring attempts to disrupt security [34]. Figure 2 shows that networks are becoming even more vulnerable to a broader variety of security threats and emphasizes how this threat landscape motivates proactive scanning and automated policy verification. One of the major network requirements is to have several internet access points for private and public networks, therefore, it is essential to secure these networks. Some common attacks on networks are discussed below.

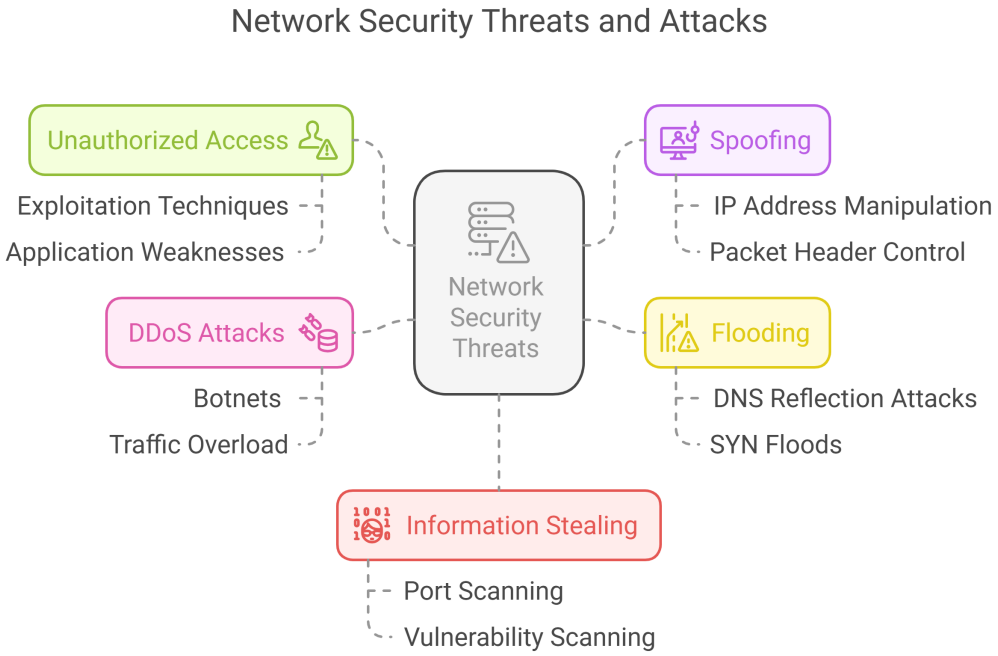

Figure 2. Diagram of Common Network Security Threats and Attacks Such as Unauthorized Access, Spoofing, Flooding, and DDoS in Cybersecurity.

3.2.1. Unauthorized Access

Unauthorized access attacks are intended to provide an attacker with safe access to targeted systems without permission. These attacks usually gain the advantage of the existing vulnerabilities in a targeted system by using well-known exploitation techniques like, scripts or hacking tools, against the targeted system. This contains unauthorized reading, copying, writing, deleting, or sharing information that usually is not available to an attacker. System access is generally extended by making use of identified application weaknesses that could offer partial or full access to a specific system. Likewise, access to a system can be acquired through back doors set up by an attacker or poor application structure during prior system setup [35].

Blockchain can mitigate unauthorized access by implementing decentralized identity management, where user credentials are verified across multiple nodes, reducing reliance on a single point of failure and enhancing access control integrity [36].

3.2.2. Spoofing

IP spoofing happens when the malevolent program generates its packets and does not mention the actual source IP address within the headers of such packets. It is not very difficult to produce individual packets with thorough control in the header of IP and transmit via the same network if one gains sufficient rights in the OS [37]. The intruder captures the IP address of the source system

 doi:10.20944/preprints202505.1028.v2

and places this IP address on the packet headers that are being transmitted toward the destination system, thereby evolving the destination machine by ensuring it is an authentic source machine for the attacker that transmits the packets on-demand [8] [38].

By leveraging blockchain for secure data sharing and verification of packet origins, the risk of spoofing can be reduced. Immutable records of legitimate IP addresses stored on the blockchain can help distinguish authentic traffic from spoofed packets.

3.2.3. Flooding

Generally, malevolent software events are furtive, and hence recognition of such actions is a challenging task. An initiating relation can be supposed to be causal and to develop a time-based affiliation between such events, e.g., in the scenario of a spoofed DDOS flooding attack, the intruder handles a three-way handshake process. During such an attack, the total number of spoofed IP addresses and the open ports utilized by the intruder undergo a causal relationship. DNS reflection attacks and TCP SYN floods are common examples of flooding. Currently, most of the DNS reflection attacks are instigated by spoofing the source IP address to overflow the Internet. For example, SYN floods are spoofed TCP floods, where the source of IP packets looks to be different from their concrete origin. In case the servers are compromised, they can also transfer spoofed packets to generate a large attack. According to Verisign [39], in the second-to-last quarter of the year 2016, there was an enormous strength TCP-SYN flood attack comprising 60 Gigabytes (GB) per second with 150 million packets per second. It was quite a bigger attack than the earlier biggest attack with 125 million packets per second during the last quarter of 2015 [40].

3.2.4. Distributed Denial of Service (DDoS) Attack

The primary objective of DDoS attacks is to deny authentic users' access to different resources of the enterprise network. Figure 3 shows the structure of a distributed denial-of-service (DDoS) attack, depicting how an intruder coordinates zombies and handlers to create a botnet. This demonstrates the type of network disruption that BENSAM's blockchain-based monitoring and tamper-evident logging aim to detect and trace.

 doi:10.20944/preprints202505.1028.v2

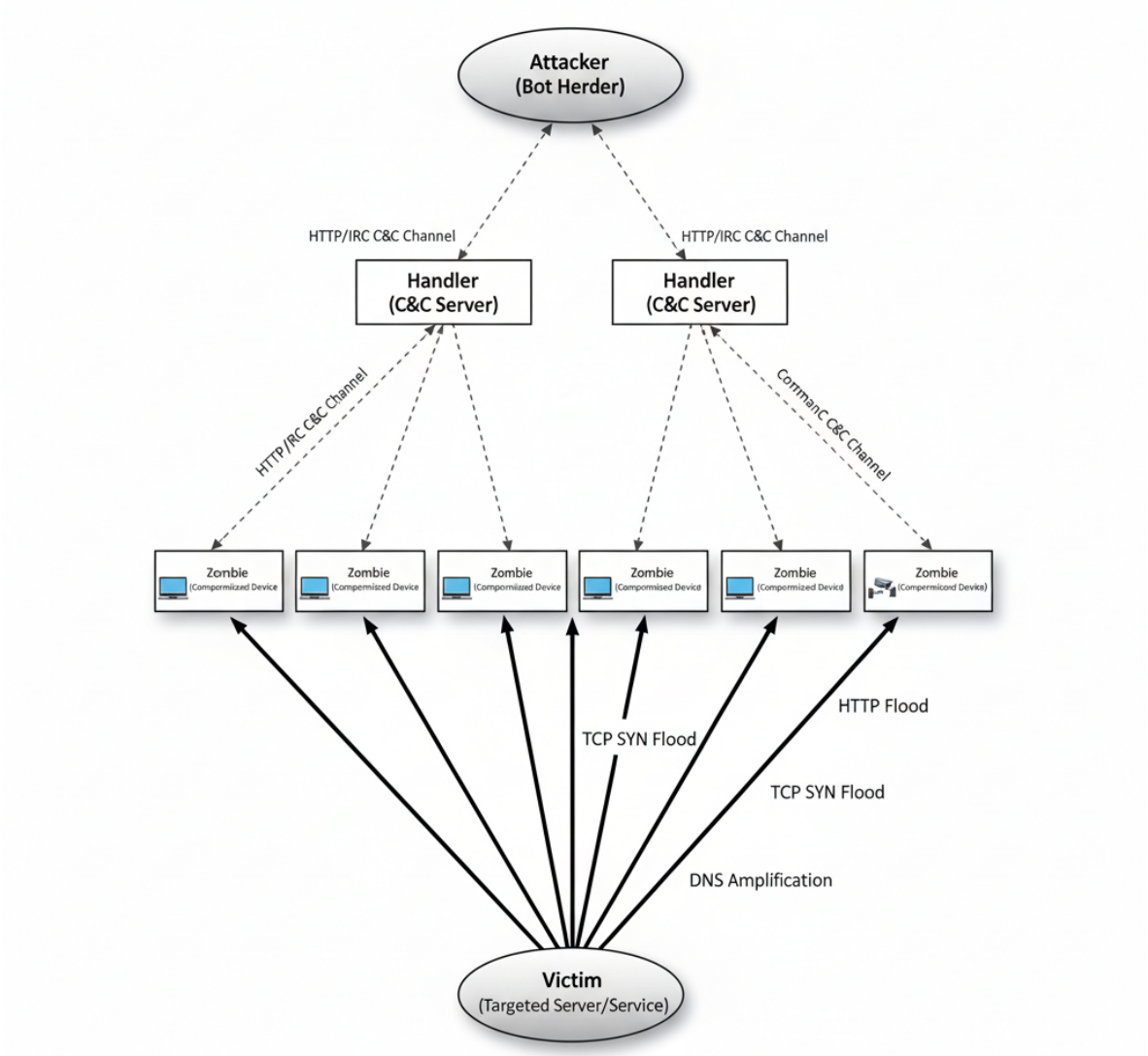

Figure 3. Visual Representation of a Distributed Denial of Service (DDoS) Attack Using Zombies and Handlers to Create a Botnet for Network Disruption.

A botnet may contain hundreds of compromised sources that create capacious traffic to overwhelm the victim. It is tremendously hard to distinguish genuine traffic from attack traffic. It is quite possible that such sources may be spread all around the world [41–44]. Earlier, the DDoS attacks were launched in four (4) different phases like, trade-off, scanning, deployment, and finally propagation. Steadily over the subsequent years, automation mechanisms have been introduced in each of these phases, yet these phases are similar [40].

1. The intruder gathers information about network configuration through port scanners to recognize existing weaknesses in the network.
2. The intruder exploits such vulnerabilities to launch the attack over the organization's network.
3. In case of a successful attempt of attack, the attacker further installs and sets up additional software to manage uninterrupted network access channels.
4. Finally, the intruder struggles to wash-out any remaining evidence that may be left due to the earlier actions. At this stage, daemons restarted that crashed during the 2nd phase, logs were deleted and various actions were taken accordingly.

3.2.5. Information Stealing

Information gathering about a particular network can be done through network port scanning and vulnerability scanning, however, if this job is done by anonymous persons, these are observed as

 doi:10.20944/preprints202505.1028.v2

the start of an attack. In scanning processes like ping sweeps and port scans, explicit information about particular IP addresses mapped to active hosts and the services they provide, is returned. Similarly, the inverse mapping method collects information about IP addresses that do not map to active hosts and this assists the attacker in emphasizing possible IP addresses. During the footprint phase, the attacker creates a profile of the targeted organization including data such as e-mail servers and the domain name system (DNS) of the organization along with its IP address range. In the scanning phase, the attacker determines details about the listed IP address range which can be accessed online, system architecture, OS information, and the list of services running on each system [45]. However, in the enumeration stage, the attacker gathers data like, routing tables, group names, network users, data of Simple Network Management Protocol (SNMP), and so on [5].

Immutable logging on the blockchain ensures that any attempt to steal information is recorded in a tamper-proof manner, aiding in detection and traceability during forensic investigations [46].

3.2.6. Emerging AI-Driven Threats

In recent years, cyber attackers have begun to exploit advancements in artificial intelligence to launch more sophisticated and evasive attacks. These include AI-generated phishing emails, polymorphic malware that mutates to avoid detection, and adversarial inputs designed to mislead intrusion detection systems. Additionally, large language models (LLMs) can automate reconnaissance by interpreting network responses, generating payloads, or identifying weaknesses more efficiently than traditional scripts. Such AI-driven threats evolve rapidly and are often harder to detect using static rule-based mechanisms. Although the current study focuses on traditional attack vectors, future versions of the proposed system will be extended to address these emerging, intelligent threats through adaptive detection mechanisms and enhanced policy enforcement.

3.3. Policy Enforcement Methods

Organizations define their own procedures in the form of policies to safeguard their data and networks and such policies can be implemented in numerous ways from simple instructions to employees to system hardening or installing of monitoring agents in the individual's machines. Some of the common policy implementation mechanisms are discussed in the following subsections:

3.3.1. Guidelines

Organizations must define and document certain security policies in the form of guidelines. These guidelines can be communicated to the employees through various means like, verbally, through email and organizational documents. The next step is to ensure that all of the designed guidelines are implemented and properly followed by the employees. As in this method, there are no checks and balances so the organizations have to trust their employees that they will not violate the policy which seems impossible.

3.3.2. OS Hardening

It is a method of increasing the security of network infrastructure and an OS to improve effective security. The security of an OS can be reinforced by setting up appropriate configurations, eliminating vulnerable services, updating software, and applying security policies e.g. monitoring user logins and enhancing password strength. A comprehensive set of minimum requirements for OS hardening is proposed based on NIST, FIPS, CC. cybersecurity standards [12]. The complexity of OS hardening is influenced by enterprise policies and the skills of the network administrator [47]. The most common exercise is to follow predefined security guidelines that are executed from time to time to ensure that security procedures are in place. The guidelines list can be executed through various auditing tools like, Nmap, Nessus, Open Vulnerability Assessment Scanner (OpenVAS) [48]. This policy enforcement method fails when an employee requests admin privileges to perform certain tasks such as installing specific software, or modifying system configurations. Likewise, if a hypervisor like, VirtualBox or VMWare, etc. is installed, then the employee has a fully flagged OS under his/her control with

 doi:10.20944/preprints202505.1028.v2

admin privileges, and the activities performed on this virtual machine will not be detected by the organization.

3.3.3. Agent-Based Enforcement

An agent-based solution to be installed on the host to monitor and to keep track of activities like software installations, file downloads, and, logs. This solution helps to monitor the user activities and manage logs that can be used for further analysis. In this method, the organizations install an agent on each system of their network, which will inform the server about any violation done by the user but this method can also be bypassed by the employees by disabling the agent.

3.3.4. Restriction through Software Defined Networking

A modern way to enforce network security policy in an organization is by using software-defined networking (SDN) [13]. Nevertheless, this requires a major transformation by shifting the complete network infrastructure to SDN by replacing all the traditional network devices with OpenFlow-enabled devices.

A trustworthy method to review and enforce an enterprise's procedures is a network scanner that is capable of identifying common vulnerabilities such as a disabled firewall, enabled remote access, virtual machines shared directories. There is a need to develop an advanced network scanner that allows us to scan network assets from time to time, inquire about vulnerable assets and all security information, e-mail scans, and take suitable action to protect assets based on the indemnification solutions provided. As the scanner is an essential tool for a network administrator as well as a penetration tester for the diagnosis and investigation of the enterprise network.

3.3.5. Blockchain-Based Policy Enforcement

Blockchain technology introduces a novel method for policy enforcement by utilizing its decentralized and immutable properties. Security events, such as policy violations detected by the network scanner, can be logged on a blockchain, ensuring they cannot be altered or deleted [49,50]. Smart contracts self executing agreements coded on the blockchain, can automate responses to violations, such as isolating a non-compliant device or notifying administrators, without manual intervention. Additionally, a decentralized network of nodes can verify compliance checks, reducing the risk of manipulation and enhancing trust in the enforcement process. This approach complements traditional methods by providing a robust, automated, and auditable framework for policy adherence [51].

3.4. Advanced AI-Driven Network Attacks

With the increasing use of machine learning and transformer models in network monitoring, attackers have also adopted AI-driven techniques to bypass conventional security mechanisms. These advanced attacks exploit vulnerabilities in ML-based intrusion detection systems (IDS), anomaly detectors, and network traffic classifiers, posing a growing challenge for organizational network security. Some of the notable AI-driven attacks are described below.

3.4.1. Adversarial Attacks on Intrusion Detection Systems (IDS)

Adversarial attacks manipulate network inputs to deceive ML-based IDS models into misclassifying malicious traffic as benign. By carefully perturbing traffic features, attackers can evade detection without triggering alarms. Studies have shown that adversarial inputs can reduce detection rates by over 80 percent, highlighting the limitations of current deep learning–based IDS in real-world scenarios[52].

3.4.2. Data Poisoning Attacks on Network Anomaly Detection Systems

Data poisoning involves introducing malicious or misleading data into the training or monitoring datasets of ML-based security systems. This compromises the model's integrity, resulting in increased false negatives or false positives[53]. Such attacks are particularly effective in distributed or federated

 doi:10.20944/preprints202505.1028.v2

learning systems, where poisoned updates from compromised nodes can degrade overall model performance.

3.4.3. Transformer-Based Evasion Attacks

Transformer architectures, such as BERT and GPT variants, can be leveraged to generate adaptive malicious traffic or highly convincing phishing content. Attackers can subtly alter packet sequences or payloads so that transformer-based IDS misclassifies them as legitimate, evading detection. Recent studies highlight frameworks like TransfficFormer that demonstrate the increasing sophistication of transformer-based evasion techniques[54].

3.4.4. Relevance to Blockchain-Enhanced Policy Enforcement

These AI-driven attacks illustrate the limitations of traditional network monitoring tools in detecting sophisticated threats. By integrating blockchain technology, immutable logging, and smart contract–driven policy enforcement, the proposed network scanner framework ensures a tamper-proof record of device configurations, user activity, and compliance checks. This provides an additional layer of security against adversarial manipulation, poisoning, and transformer-based evasion by maintaining a trusted audit trail and enabling automated policy verification and enforcement.

4. Basic Features of a Network Scanner

Network scanner offers analysis and assessment for different weaknesses of network systems thereby attaining preemptive protection. It enables the network administrator to recognize some unsafe backdoors, vulnerabilities and overcome such weaknesses before the system is broken. Many of the scanning solutions are generally based on packet capturing and packet crafting libraries like, Libpcap and Libnet.

The proposed network scanner will interface with a blockchain network to store and retrieve critical data, such as scan results and compliance statuses. By recording this information on an immutable ledger, the scanner ensures that security assessments are trustworthy and verifiable, enhancing the reliability of basic features like IP scanning, port scanning, and vulnerability detection.

Libnet [55] library is written by Mike D. Schiffman and can generate and send packets for numerous protocols. Especially, such a system gets a fast and suitable capability that is based on Libnet since it is specifically designed for packet crafting. The procedure of creating network packets is streamlined and packets of each distinct protocol can be generated using Libnet. It offers support for common protocols like, TCP, IP, UDP, ARP, MPLS, RARP, ICMP, and so on [56]. Nmap uses its modified version like, Libdnet, for low-level tasks like sending ethernet frames, etc [57].

Libpcap [58,59] is a generic library designed by Steven McCanne, Craig Leres, and Van Jacobson from the Lawrence Berkeley National Laboratory at the University of California, for packet capturing that offers a high-level interface to packet capture systems. It practices the BPF [60] technique of packet filter to receive the packet rapidly. The original filter expression can be associated with building a much more complex filter expression using “not” “and”, and “or”. In the Microsoft Windows platform, the Winpcap utility is based on the Libpcap and uses the NPF mechanism. The network scanning and monitoring tool sits within any enterprise network as illustrated in Figure 4. Network scanning comprises some of the core features like port scanning and vulnerability scanning.

 doi:10.20944/preprints202505.1028.v2

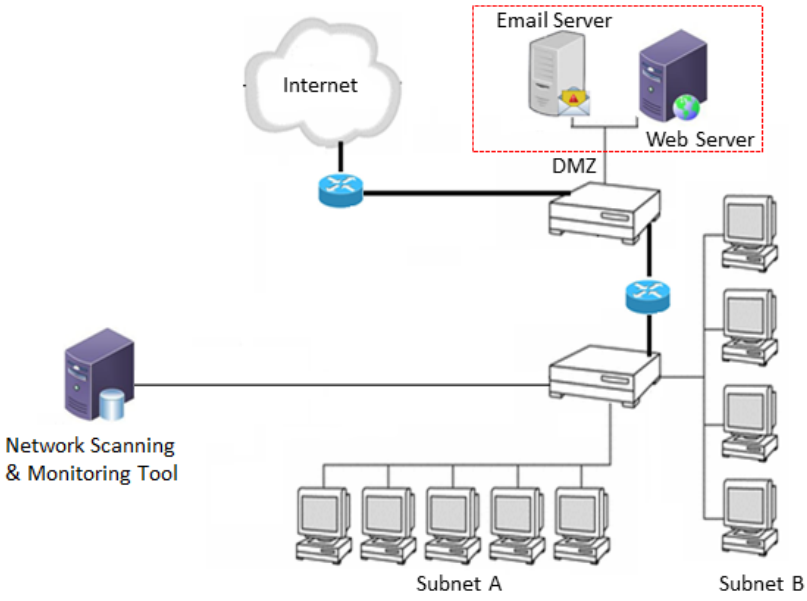

Figure 4. Deployment Architecture of Network Scanning and Monitoring Tools in Enterprise Networks for Enhanced Security and Compliance.

There are several network, IP, Port, and vulnerability scanning tools widely used for security auditing and network scanning. Some of the widely available scanning tools are Nmap [61–63], SolarWinds Port Scanner [64], Advanced IP Scanner [33,65] , Angry IP Scanner [66,67], Free IP Scanner by Eusing [68,69], NetCat [70,71], LanSweeper IP Scanner [72–74], MyLanViewer Network/IP Scanner [75–77], Nessus [78–82] and Slitheris Network Discovery [83]. Some of the most common services these scanners provide are discussed below:

4.1. IP Scan

IP scan is a general scan and is the continuing IT job of investigating an enterprise network to determine IP addresses and discover appropriate information related to these IP addresses. It is useful in finding hidden devices (which do not appear in general searches from operating systems) and discovering new devices over the network. Almost all of the aforementioned widely used scanning tools like, Nmap, SolarWinds port scanner. provide IP scanning features to discover the up hosts on the network.

4.2. Port Scan

In port scanning, data packets are sent over the network to the specified service port numbers (e.g., port number 23 for Telnet, and port number 80 for HTTP.) of a targeted system. The goal of port scanning is to discover open ports of the host and get information about the services running on these ports. It is really valuable to get further details from the remote host since ports are opened by different servers like Email servers, FTP servers, Web servers, and so on. TCP and UDP protocols require ports to communicate through the Internet, and every port is a number that identifies different types of service. The port numbers initiated at 1 up to a total of 65535 ports including port numbers of lower ranges are used for general Internet protocols. Through the port scanning method, it can be identified which ports are open, closed, or filtered of any remote host(s). Nmap, SolarWinds with many other scanners have port-scanning features. Generally, port scanning comprises three types: open, half-open, and stealth scan as briefly described below:

4.2.1. Open Scan

An open scan is a process of discovering open ports in a remote host and is mostly done by the network administrator and pen-testers during the analysis of networks using a TCP connection to the

 doi:10.20944/preprints202505.1028.v2

destination host. Such types of scans are certainly recognized by the firewalls and usually complete the three-way handshake port scan process [84] [85].

4.2.2. Half-Open Scan

It determines if a port is open by executing the first step of a three-way handshake. The aggressor struggles to set up a TCP/IP connection with a server at each potential port and this is carried out by sending a synchronization (SYN) packet to each port of the server.

4.2.3. Stealth Scan

In this type of port scan, a firewall, filter, or router is bypassed, thereby acting as spontaneous network traffic. Various stealth port scan mechanisms are practiced including NULL scan, FIN scan, and XMUS scan [86].

4.3. Banner Grabbing/OS Detection

In network environments, the system vulnerability is associated with the OS and various OS have their security features. Therefore, before inquiring about the security status of the system the information about the OS must also be understood. In network scanning, the identification of OS should be the first step in network security scanning, can discover suitable information about OS like, OS version, classification., and is also very useful for the discovery of OS vulnerabilities. Since distinct OSs have different kernels and implementation styles thereby the OS detection of a remote host becomes integral to acquiring precise mechanisms to discover the OS vulnerabilities [56]. Banner grabbing can be done through Nmap, Zenmap, Nessues. scanning tools. Nmap stresses identifying the accurate OS of a remote host, however, this may not be possible for every host. In such a case, it labels the percentage with the detected OS. Likewise, SolarWinds Port Scanner offers hostname resolution with particular DNS details and in addition to this, it can also discover MAC addresses to extract the OS and its related information.

4.4. Server Recognition

While scanning the remote host, may require verifying whether the remote host is either a server or a client. This can be achieved by extracting information about some common ports like, 8080, 25. For instance, the port for the SMTP mail server is 25, 21 for FTP and 8080 port number for a web server. The presence of one or more of these open ports on a system represents that the system is working as a server. Nevertheless, none of the major scanning tools provide such features to discover the server machine.

4.5. Vulnerability Scan

Vulnerability discovery is one of the keys to protect computer systems and such weaknesses can be exploited by hackers to get control of the targeted system. The goal of vulnerability scanning is to discover and fix such weaknesses before attackers utilize them against the machine. Various known weaknesses can be identified by the vulnerability scanners and these scanning tools accomplish this objective by following different techniques. The development of a vulnerability scanning tool is to probe a list of supplied ports of a host and attempts to discover the service running at every port for various known vulnerabilities thereby leading to potential threats to the system [56]. However, in vulnerability scanning, the intention is to identify well-known systems vulnerabilities that are present over the network. It facilitates the discovery of particular weaker spots in an Operating System (OS) and the application software, that can compromise or crash the targeted system [87].
Nessus [82] is one of the well-known tools for vulnerability assessment and is a multithreaded-based tool. The vulnerability data supplied by this tool is compatible with Common Vulnerabilities and Exposures (CVE) [88] [89] which is a publically available famous dictionary for information security. CVE's general identifiers allow data exchange among security products and offer a baseline index for assessing coverage of tools and services. The management of the CVE dictionary is done by

 doi:10.20944/preprints202505.1028.v2

the MITRE Corporation, which generates the list of standardized names for well-known security vulnerabilities and exposures. It is straightforward and facilitates the provision of separate databases to exchange vulnerability data. The vulnerability of the targeted system can be swiftly revised by the CVE-compatible database in case the information includes a CVE token [35]. Likewise, Nmap provides vulnerability scripts, namely Vulscan and Vulners, which enable network administrators to discover related CVE information from the specific local host machines or remote hosts. Vulscan queries CVE from its local databases that are hosted on the machine containing the Nmap client application. The overall vulnerability scan process can be seen in Figure 5.

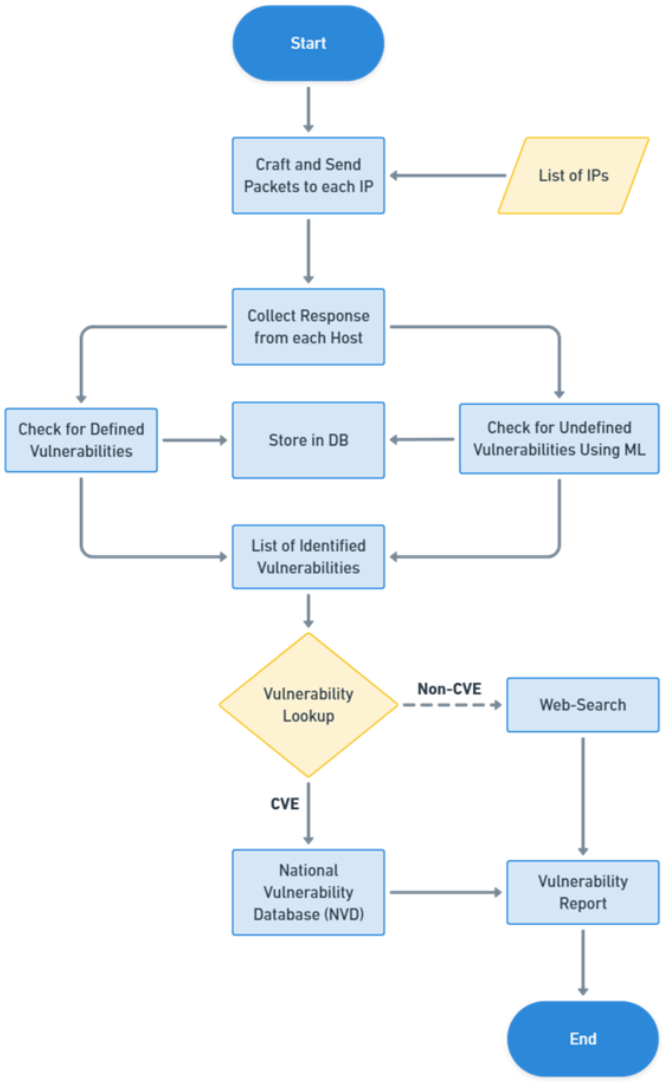

Figure 5. Process Flow of a Vulnerability Scan Identifying Weaknesses in Network Systems and Applications for Cybersecurity Risk Mitigation.

5. Security Aspects Discoverable via Scanner

An attacker discovers the weaknesses using common network and vulnerability scanners before intruding into the system and reveals the security vulnerabilities that compromise the data or steal sensitive information by intrusions. The scanner can verify the integrity of the blockchain ledger used to store security data, ensuring that logs and compliance records remain untampered. It can also monitor the proper execution of smart contracts, confirming that automated policy enforcement actions are triggered correctly. These checks enhance the overall security posture by ensuring the blockchain component operates as intended. A brief overview of some of the common security traits discoverable through a common scanner is given below:

 doi:10.20944/preprints202505.1028.v2

5.1. Firewall Status

A firewall is a perimeter security system installed in a host or network that monitors, controls, and tracks the incoming network traffic, as well as outgoing traffic based on a predefined set of security, controls [90]. It usually generates a blockade between a secure, trusted internal network and an alternative outside network like, the Internet, which is not supposed to be a trusted or secure network. Firewalls are generally categorized into two types like, host-based firewalls and network firewalls. A host-based firewall can be installed and run on a network-connected device or a personal computer and monitor network traffic on that machine. However, the network firewall is a software appliance that can run on a general-purpose hardware-based or hardware firewall system appliances that filter network traffic among two or more networks [32].
Scanners can easily check the status of the firewall on a system via port scanning. For example, Nmap can test and verify the firewall rules and filters while port scanning. It can be verified by simply checking whether the port is open, closed, or filtered. A filtered port means that Nmap is unable to check the status of ports due to firewalls. The TCP ACK scan of Nmap will establish whether network packets can pass through the enterprise firewall unfiltered.

5.2. Remote Access Status

The remote access status of a host can be investigated through a scanner. There are many services that provide remote access such as Windows' RDP, Linux's VNC, Cisco Anyconnect, Teamviewer, and pcAnywhere. The remote access can be verified by examining the corresponding port number for different services such as 3389 for RDP, 5900 for VNC, 1723 for Point-to-Point Tunneling Protocol (PPTP), 5938 for Teamviewer, and 5631-2 for pcAnywhere. To keep the system and data secure, it is essential to make sure that remote access is disabled so that the system can not be accessed remotely. In addition to that network administrators should know that these services may run on any of the available ports, however, the above-mentioned are the default ports for corresponding services.

5.3. Shared Directories

Over the enterprise network, there can be a commonplace in the digital infrastructure where most of the common resources are shared publicly. These resources can be accessed by different employees as and when required. Such a shared directory/file can be vulnerable to various cyber threats, so needs to be protected with some kind of scanning tools. Aside from centralized shared resources, individual employees should not be allowed to share the documents on their systems. To prevent certain threats, it is essential to identify the directories/files shared by the employees on their systems over the network.

5.4. Malicious Services with Open Ports

Services running on the open ports can be easily inquired by the scanner, and information about the services is also available. It is essential to check and verify that there should not be malicious services that are running with the open port(s) of a system.

5.5. Virtual Machines Recognition

A virtual machine has a full OS in control of a system user and can have direct access to the internet so it can have malicious software installed on it with admin privileges. The scanners have the capability of identifying vendor names as well as VM, using the database of vendor lists with defined MAC Address ranges assigned to each vendor. An efficient scanner should identify whether the host is a virtual machine or not.

5.6. IP Conflicts

An IP address is assigned to a system for a specified time period which is renewed when the system maintains the connectivity. If a host is idle for a long time or if a user enables the sleep mode of the system over a longer time, then there may be a possibility that during this time another host joined

 doi:10.20944/preprints202505.1028.v2

the network and assigned the same IP address. When the user disables the sleep mode and starts working on the system, there may be chances of duplicate IP addresses. However, another possibility is that a duplicate IP address to be assigned from an unauthorized or 'rogue' DHCP server connected to a subnet. Duplicate IP addresses are usually automatically found by the operating systems during the DHCP address assignment process.

5.7. Wake on LAN

A system can be switched ON remotely by any other system knowing the MAC address over the network. This can only be done if this functionality is enabled on a system. A scanner should verify that the wake on LAN property should be disabled to avoid any potential cyber-attacks.
As discussed in Section II, network security policy enforcement is the key to basic security measures. If the employees of organizations do not follow and apply the security policy on their systems then border firewalls become useless. As malicious software can penetrate the network from personal devices like laptops or cell phones of employees. Therefore, these devices should be monitored regularly. most organization relies on a central protection system and leave the employee systems unchecked, as it is very difficult to check each system individually for policy violations. In the next section, we discuss the advanced features of the proposed scanner.

6. Blockchain-Enhanced Network Scanning and Monitoring (BENSAM) Framework

This section presents a conceptual framework for secure, policy-aware network scanning and monitoring, with blockchain technology integrated as a core architectural component. For instance, instead of implementing a complete scanner, we design a modular system that supports advanced features such as immutable event logging, decentralized compliance verification, and smart contract-based policy enforcement.

The integrity of network event logs is a cornerstone of any robust cybersecurity strategy, serving as a critical resource for incident response, forensic analysis, and regulatory compliance. However, traditional centralized systems for network scanning, event logging, and policy management are fraught with inherent vulnerabilities. Centralized log servers represent a single point of failure and are a prime target for sophisticated attackers, including insider threats, who may seek to tamper with, erase, or forge log entries to conceal their activities. The lack of a verifiable, immutable audit trail undermines the foundational principles of non-repudiation and accountability.

Furthermore, traditional compliance and security policy enforcement often rely on manual, reactive processes. Audits are conducted periodically, and policy adherence is difficult to verify in real-time, leaving organizations susceptible to violations that go undetected for extended periods. The immense volume of data generated by modern networks exacerbates these challenges, creating significant scalability burdens and complicating the process of data analysis and threat detection. In light of these systemic issues, a new architectural paradigm is required to provide a secure, scalable, and auditable solution for network security. The Department of Homeland Security's (DHS) exploration of Distributed Ledger Technology (DLT) for creating immutable and publicly verifiable audit logs highlights the pressing need for such a shift in approach

This report presents the Blockchain-Enhanced Network Scanning and Monitoring (BENSAM) Framework, a conceptual blueprint for a next-generation security system. The BENSAM framework leverages a permissioned blockchain, specifically Hyperledger Fabric, as a core architectural component to address the limitations of conventional systems. The framework's design is fundamentally modular, integrating off-the-shelf and custom-built components to support immutable event logging, decentralized compliance verification, and smart contract-based policy enforcement. Instead of relying on a single, monolithic scanner, the system supports a distributed network of scanning agents, which feed security events into a highly secure and verifiable control plane.

The core innovation of the BENSAM framework lies in its hybrid data storage model. Raw network events are stored securely in a traditional, off-chain database for reasons of scalability and

 doi:10.20944/preprints202505.1028.v2

privacy. On the blockchain, only a minimal, cryptographic record of the event is stored, specifically a fixed-length hash of the event data and relevant metadata. This on-chain record serves as an immutable, tamper-proof commitment to the integrity and existence of the off-chain data. Smart contracts, known as chaincode in Hyperledger Fabric, are used to automatically execute and verify compliance policies as part of the transaction process, creating a self-enforcing system of governance.

The design of the BENSAM framework is predicated on the careful selection of technologies that align with the rigorous requirements of enterprise security. A modular architecture was chosen to ensure flexibility and seamless integration with existing IT and security infrastructure. This approach allows for the use of various scanning agents and logging services, adapting to diverse network environments and operational needs.

Hyperledger Fabric was selected as the foundational DLT for several critical reasons. Unlike public, permissionless blockchains like Bitcoin or Ethereum, Hyperledger Fabric is a permissioned network where all participants are known and identifiable. This is a prerequisite for a security framework operating in a regulated, enterprise context where accountability is paramount and anonymity is undesirable. Fabric's support for private channels and private data collections enables the confidentiality of transactions and data, allowing for granular control over who can view specific information, a crucial feature for sensitive security data. The platform's high transaction throughput and low latency, achieved through a unique architecture that separates transaction execution from ordering, make it well-suited for high-volume network event logging. This design choice ensures that the system can scale effectively without the performance bottlenecks seen in other blockchain architectures.

The BENSAM Framework is a conceptual, multi-layered system engineered for comprehensive and verifiable network security as show in Figure 6. The image provided depicts the Blockchain-Enhanced Network Scanning and Monitoring (BENSAM) Framework. The diagram illustrates the core architectural components and the detailed, step-by-step data flow of a security event. On the far left, Network Scanning Agents generate raw security event logs. These logs are fed into the Logging & Event Aggregator Layer, which processes, filters, and hashes the payload. This layer then securely stores the full log payload in an Off-Chain Data Store while creating a transaction proposal containing only the hash of the data, the metadata, and a reference ID. This proposal is sent to the Hyperledger Fabric Peer Nodes, where the Ordering Service packages the transaction, and the Peer Nodes endorse and commit it to the shared, immutable ledger. The Chaincode (Smart Contract), which holds the policy enforcement logic, runs a compliance check on the transaction's metadata before it is committed. Finally, the Monitoring & Audit Layer can query the on-chain hash, retrieve the off-chain data using the reference ID, and re-compute the hash to verify the integrity of the original log, ensuring a tamper-proof and auditable record of all network security events. This framework provides a robust, decentralized solution for maintaining the integrity and confidentiality of security data. Its design addresses the core challenges of scalability, integrity, and compliance through a distributed architecture. The framework is composed of the following primary components, which work in concert to process, secure, and monitor network events:

- Network Scanning Agents: These are the data collection endpoints, deployed on servers, hosts, and network devices to continuously monitor for security-relevant activities.
- Logging Service/Event Aggregator: This component acts as the data ingestion gateway, receiving raw events from the agents, processing them, and preparing them for both on-chain and off-chain storage.
- Policy Controller: A centralized management plane that defines and orchestrates the security and compliance policies that govern the network.
- Hyperledger Fabric Network: The core DLT foundation of the framework, comprising Peers (who maintain the ledger and execute transactions), an Ordering Service (which validates and orders transactions), and a Certificate Authority (CA) that manages identities.
- Off-Chain Data Store: A separate, highly scalable database (e.g., a relational or document database) used to store the full, raw log payloads and other large data assets.

 doi:10.20944/preprints202505.1028.v2

- Audit and Analytics Console: A user-facing interface for security analysts and auditors to query event logs, visualize data, and verify the integrity and compliance of recorded events.

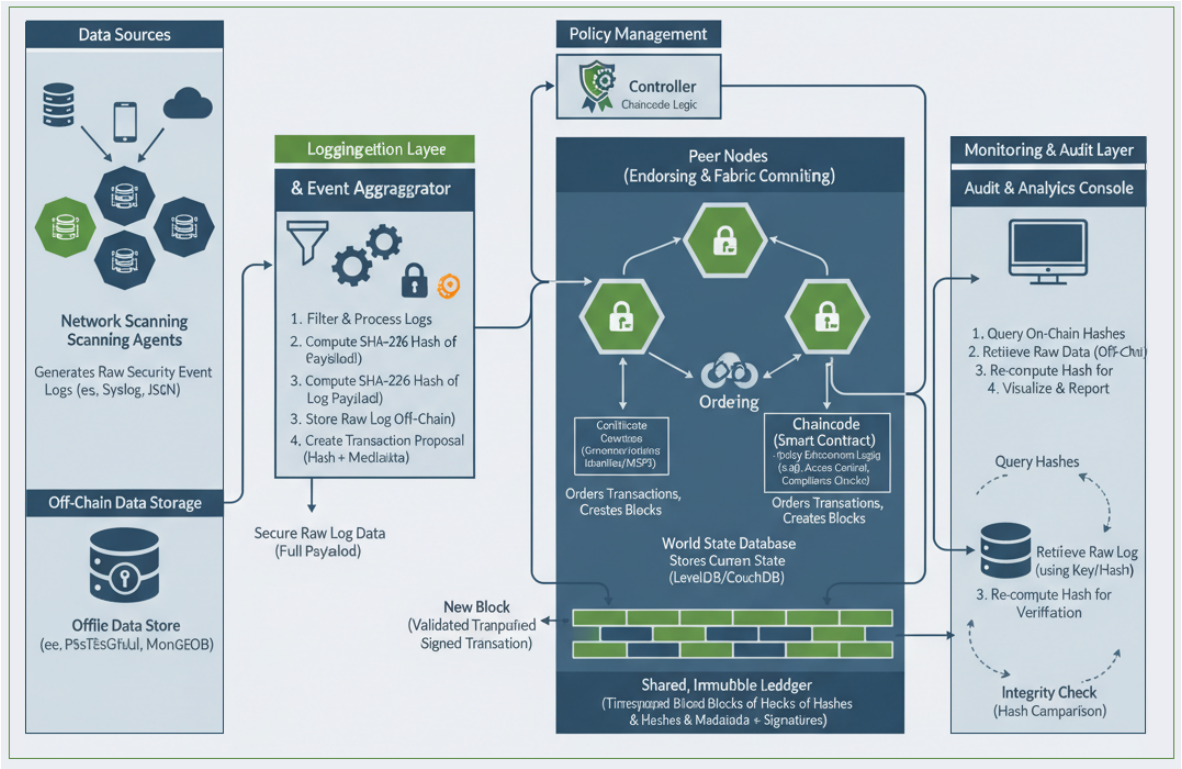

Figure 6. Blockchain-Enhanced Network Scanning and Monitoring (BENSAM) Framework. The diagram illustrates the core architectural components and the step-by-step data flow of a security event.

Table 1 provides an overview of BENSAM framework component roles. The operational flow of the framework is designed for efficiency and security. Network Scanning Agents continuously monitor their environments and, upon detecting a security event, generate a structured log entry. This log is then sent to the Logging Service. The Logging Service performs two critical, near simultaneous functions. First, it securely stores the full, raw log payload in the Off-Chain Data Store. Second, it generates a cryptographic hash of the event data, which acts as a unique digital fingerprint. This hash, along with a minimal set of metadata (e.g., source agent ID, timestamp, and event type), is packaged as a transaction proposal and submitted to the Hyperledger Fabric network. It is important to note that while the Policy Controller's chaincode generates compliance and enforcement decisions, the actual actions on network devices such as alerts or device isolation are carried out by external controllers including NAC, SDN controllers, or EDR systems.

The transaction is processed by the Fabric network's peers and ordering service. During this process, a smart contract (chaincode) automatically executes compliance checks against the transaction data, a process which is then endorsed by the relevant peers. Once the transaction is validated and committed to a new block, the immutable record of the event's hash, its metadata, and the signatures of the endorsing peers are permanently stored on the distributed ledger. The Audit and Analytics Console can then be used to query this on-chain ledger, retrieve the hashes, and use them to verify the integrity of the corresponding off-chain log data.

The architecture is a novel integration of existing, proven technologies. It is not a new scanner, but rather a new, secure control plane for managing scanner output. This hybrid approach, which places a lightweight, verifiable hash on-chain while keeping the raw data off-chain, is a direct solution to the performance and privacy limitations that would arise from attempting to store large network event logs directly on a blockchain. It provides a practical and scalable foundation for a system that can handle the continuous, high-volume data stream of a large enterprise network.

 doi:10.20944/preprints202505.1028.v2

Table 1. BENSAM Framework Component Roles

Component	Key Responsibilities	Input/Output
Network Scanning Agent	Continuously monitors endpoints, servers, and network devices for security events, misconfigurations, and vulnerabilities.	Inputs: Local system/network data. Outputs: Structured raw log data.
Logging Service	Acts as the data ingestion point; aggregates and filters raw logs; generates a cryptographic hash of each log; and securely stores the raw data off-chain.	Inputs: Raw log events from agents. Outputs: (1) Hashed log events (sent to blockchain). (2) Full raw logs (sent to off-chain store).
Policy Controller	Defines, manages, and updates network security policies. Translates high-level rules into executable chaincode logic for compliance verification.	Inputs: Policy definitions. Outputs: Policy parameters for chaincode.
Hyperledger Fabric Network	The distributed ledger for storing immutable, verifiable records of event hashes and metadata. Enforces policies via chaincode and consensus.	Inputs: Hashed transaction proposals. Outputs: Immutable, validated blocks containing event hashes.
Off-Chain Data Store	Provides scalable and private storage for the full, raw log payloads and other large data. The primary data is retrieved from here during an audit.	Inputs: Raw log data from the Logging Service. Outputs: Raw log data to the Audit Console (on request).
Audit & Analytics Console	Provides a user interface for security analysts and auditors. Queries the blockchain for immutable hashes and retrieves the corresponding raw data from off-chain storage for verification and analysis.	Inputs: Queries from auditors. Outputs: Visualizations, reports, and audit verification.

The following pseudocode formalizes the operational workflow of the BENSAM framework. It details how scanning agents, logging services, device profiling, traffic monitoring, and smart contract-based policy enforcement interact to produce immutable logs and compliance reports (see Algorithm 1).

6.1. Immutable Logging and Hybrid Storage Architecture

Within the proposed framework, the blockchain functions as a trustworthy audit trail for critical events. Specifically:

- Device profiling data (e.g. device types, MAC/IP addresses, OS/software) are hashed and committed to the blockchain to ensure historical integrity.
- Traffic monitoring summaries such as packet metadata, are logged immutably to enable post-event validation and auditing.
- Policy compliance results and enforcement actions are recorded on-chain to create verifiable compliance histories.

To mitigate blockchain storage constraints, we propose a hybrid architecture: detailed records are stored off-chain (e.g. CouchDB), while their hashes and references are anchored on-chain. This model balances tamper resistance with performance and scalability.

 doi:10.20944/preprints202505.1028.v2

Algorithm 1 Blockchain-Enhanced Network Scanning and Monitoring Algorithm with Immutable Logging and Smart Contract-Based Policy Enforcement

Initialize: Network Scanner with Database, Blockchain Channel, and Smart Contract Interface
Input: Network Range, Scan Frequency, Policy Rules (Smart Contracts)
Output: Device List, Traffic Logs, Immutable Blockchain Logs, Compliance Reports
procedure NETWORKSCAN
    Retrieve stored device profiles from database
    Scan for active devices on the network
    for each detected device **do**
        if device is new **then**
            Generate alert and add to database
            call DEVICEPROFILING(device)
        else
            Update last seen timestamp
        end if
    end for
end procedure
procedure DEVICEPROFILING(device)
    Identify device type (Laptop, Mobile, Printer)
    Retrieve MAC and IP addresses
    Check OS and installed software
    Store profile in database
    Record profiling event on blockchain (immutable log)
end procedure
procedure TRAFFICMONITORING
    Capture and log network traffic (Packet Headers Only)
    for each packet **do**
        Extract Source and Destination IP, MAC, and Ports
        Log activity in database
        Log traffic metadata to blockchain for tamper-proof audit trail
    end for
end procedure
procedure POLICYENFORCEMENT
    for each device in network **do**
        Trigger smart contract with current device and traffic state
        if smart contract returns violation **then**
            Generate alert and log violation in database
            Record violation on blockchain (for compliance traceability)
        end if
    end for
end procedure
procedure GENERATEREPORTS
    Retrieve logs and blockchain records
    Compile network status, immutable events, and compliance violations
    Export report in PDF format
end procedure

 doi:10.20944/preprints202505.1028.v2

6.2. Smart Contract Role in Security Automation

Smart contracts play a central role in automating the framework's monitoring and enforcement logic. Deploying on authorized peers in the Fabric network, they perform the following key functions:

- Initiating scan operations periodically or in response to specific triggers (e.g., new device discovery),
- Verifying device configurations and usage behavior against predefined policy templates,
- Logging violations and generating tamper-proof alerts,
- Triggering enforcement mechanisms, such as isolating non-compliant devices or escalating issues to administrators.

Each contract invocation is recorded on-chain, ensuring transparency and accountability across the security lifecycle.

6.3. Alignment with Security Standards

The framework design aligns with key controls from the NIST SP 800-53 Rev. 5 catalog, providing conceptual support for regulatory and operational security requirements:

- AU-2 (Audit Events): Immutable blockchain records of scan and enforcement events,
- SI-4 (System Monitoring): Continuous tracking of network activity via trusted smart contracts,
- SC-12 (Cryptographic Key Establishment): Secure peer and user identity validation using digital certificates,
- SC-28 (Protection of Information at Rest): Cryptographic integrity of off-chain data via hashed blockchain references.

These mappings demonstrate the readiness of the framework to serve as a secure foundation for network monitoring solutions in regulated environments.

6.4. Implementation and Preliminary Validation

To validate the conceptual design of the proposed BENSAM Framework, we developed a working prototype and released its source code publicly on GitHub at https://github.com/swhamdani/BENSAM-Framework. The prototype demonstrates the end-to-end workflow of the framework, including network scanning, device profiling, traffic monitoring, policy enforcement, blockchain logging, and audit verification on static test data, following the architecture described above and illustrated in Figure 6. The project structure of this implementation is listed below:

```
BENSAM-Framework/
|-- core.py                # Main orchestration pipeline
|-- audit.py               # Smart contract and blockchain logging
|-- chaincode/bensam.go    # Prototype chaincode for policy checks
|-- interfaces.py          # Abstract interfaces for modularity
|-- tests/test_core.py     # Automated tests using pytest
|-- output/bensam_report_20251030_140037.json  # Execution output
```

Execution and Output: Running `core.py` executes the main orchestration pipeline, performing network scanning, device profiling, traffic monitoring, and policy enforcement. The `audit.py` module handles blockchain logging and smart contract-based compliance checks, simulating Hyperledger Fabric behavior. The prototype Go chaincode in `chaincode/bensam.go` implements policy enforcement rules and interacts with `audit.py` to validate transactions.

A JSON output file (`output/bensam_report_20251030_140037.json`) is automatically generated, summarizing device attributes, traffic events, detected policy violations, and blockchain reference IDs. For example, this report identifies four network devices (laptop, printer, router, and IoT camera) and flags two policy violations: unauthorized external communication from a printer and an open port on an IoT device.

 doi:10.20944/preprints202505.1028.v2

Automated tests were executed using `pytest` in PyCharm IDE to verify module interactions and execution integrity. The following console log shows the complete output of `test_core.py` execution:

This proof-of-concept confirms the technical feasibility of the proposed architecture and demonstrates the potential for auditable, tamper-resistant network monitoring. Future work will extend this prototype with live network data, real Hyperledger Fabric deployment, and performance benchmarking.

As shown in Figures 7 and 8, the console log illustrates the end-to-end execution of the BENSAM prototype, fully aligned with the proposed framework architecture. During the test, the framework successfully performed network scanning and device profiling, identifying four active devices from static data including HP EliteBook (laptop), Office Printer, Main Router, and Smart Camera (IoT). Each device profile was logged immutably to the blockchain, demonstrating the framework's tamper-resistant audit capability.

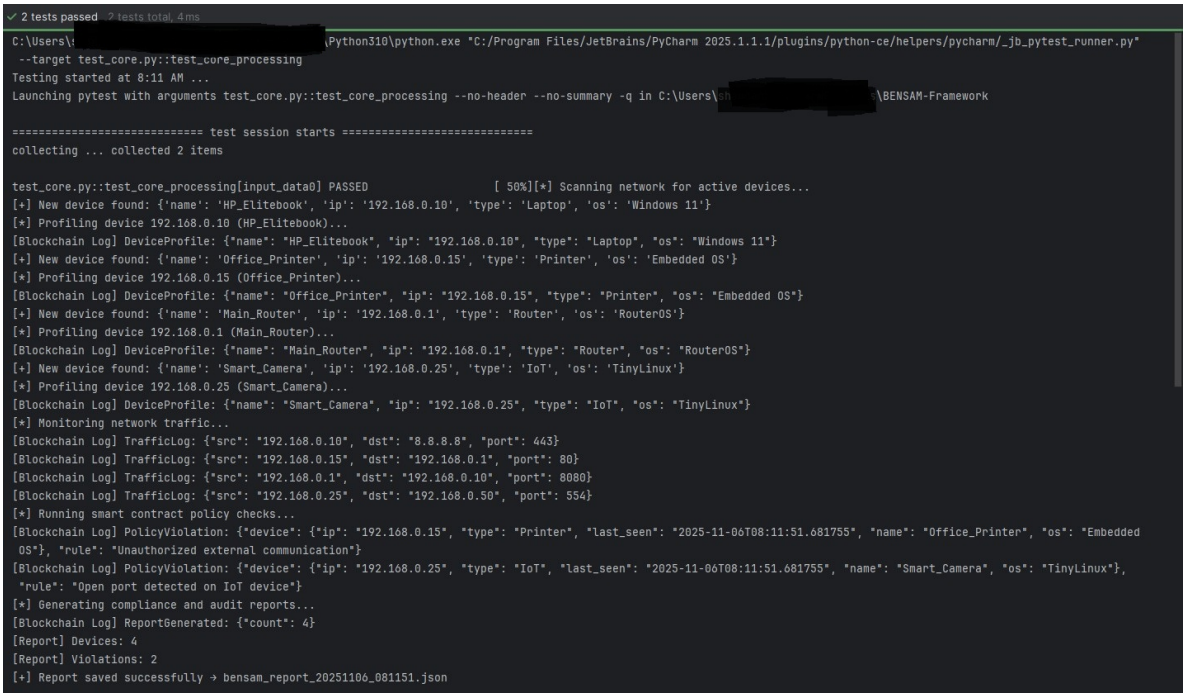

Figure 7. Console log output from `pytest` execution of `test_core.py` in PyCharm IDE, demonstrating successful device discovery, policy enforcement, and blockchain logging.

 doi:10.20944/preprints202505.1028.v2

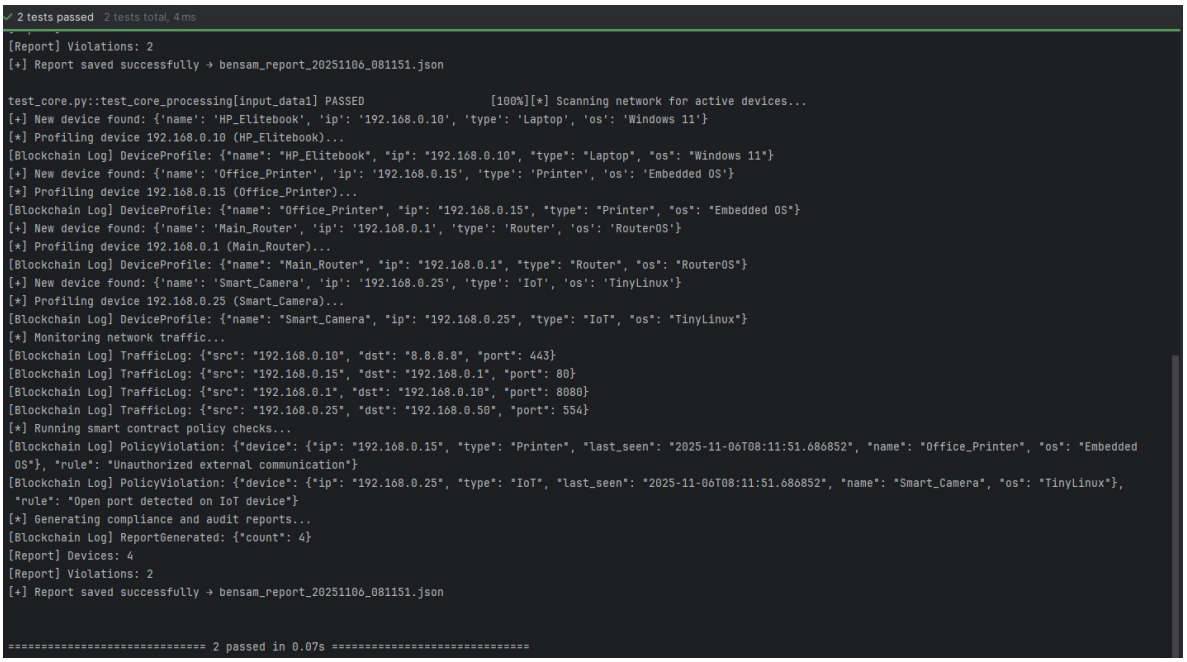

Figure 8. Continuation of console log showing policy violations detected and audit report generation confirming prototype execution integrity.

Subsequently, network traffic monitoring and smart contract-based policy enforcement were executed, detecting two policy violations i.e. unauthorized external communication from the printer and an open port on the IoT camera. Finally, the compliance and audit report was generated, summarizing the four devices, the detected two violations, and associated blockchain reference IDs. These results validate the functional integrity of the BENSAM framework and demonstrate its ability to provide auditable, end-to-end security monitoring in a controlled test environment.

6.5. Feasibility and Future Directions

Although a full-scale deployment is beyond the scope of this work, the conceptual design and the prototype implementation demonstrate that the proposed architecture is technically feasible using existing enterprise practices. Performance concerns such as latency and throughput can be addressed through:

- Event batching and aggregation prior to blockchain submission,
- Selective on-chain recording of high-value metadata only,
- Distributed off-chain storage for log-heavy operations.

The working prototype, described in above section, has been validated on static test data, producing JSON reports (`output/bensam_report_20251030_140037.json`) that demonstrate the end-to-end workflow of scanning, profiling, policy enforcement, and blockchain logging.

Future work includes deploying the framework in a controlled Hyperledger Fabric environment to evaluate operational feasibility under realistic network conditions, integrating live network data, and performing detailed performance benchmarking.

6.6. Local Database

The traditional scanner does not have a structured database to store the scanning results; therefore, the results of the previous scans cannot be retrieved or queried if required. So, linking a scanner with a structured database is the first step toward the advanced scanner and will be very useful in order to associate current results with previous scans. Furthermore, the linked database can store and provide the complete connection history of a specific device with associated IP addresses and timestamps.

Instead of relying solely on a traditional structured database like SQL, the proposed scanner integrates a blockchain-based database to store scan results, device profiles, and activity logs. This ensures

 doi:10.20944/preprints202505.1028.v2

immutability and transparency of the data, allowing multiple stakeholders to verify records without the risk of tampering. The blockchain complements the local database by providing a decentralized, secure storage layer for critical security information.

6.7. Scheduled Scanning

A scheduled scan is a network audit that is scheduled to run automatically on a specific date/time and at a specific frequency. A network scanner should have this feature to scan the network automatically. Scheduled scans can be set to execute once a day or periodically with different parameters. Scanning during working hours can help to monitor and track connected devices. Scanning results can be stored in a database and maintained accordingly. The main benefit of scheduled scanning is regular inventory checks and employees' activities during working hours. The connection history of connected devices in a network can be stored for future reference, and event logs of connected devices are maintained regularly.

Scheduled scanning is enhanced by blockchain through the use of smart contracts. These contracts can be programmed to trigger scans at specified intervals, log the results on the blockchain, and automatically initiate remediation actions if policy violations are detected. This automation reduces administrative overhead and ensures consistent monitoring.

6.8. Device Profiling

It is quite useful for enterprises to know about each device connected to their network, especially in situations where there is no defined policy of an enterprise on Bringing Your Own Device (BYOD). Hence, the personal devices of the employees will be directly connected to the enterprise network without any restriction. Therefore, device profiling is very important for an organization in order to know about the attached devices, their types (like laptop, cellphone, tab.), OS installed, first discovery on the network, and the owner's name as well. Device profiling is illustrated in Figure 9. This feature will enable an organization to identify devices. This can help the network administrator to check if any malicious system or disgruntled employee is attached to the network. Hence, it would help prevent any potential attack or theft of information. The addition of any new system (host) or the removal of the existing system can also be tracked through this network inventory.

Device profiles are stored on the blockchain, making them tamper-proof and verifiable across the network. Each profile, including device type, OS, and owner details, is recorded as a blockchain transaction, ensuring an auditable history of device connections and compliance status.

6.9. New Device Discovery

On day-to-day network scanning, it is obvious to maintain and manage the inventory of all network devices. The list of newly discovered devices can be extracted from the list of active hosts during the device discovery process, as shown in Figure 9. Therefore, whenever a device is discovered for the first time, a scanner should generate an alert and try to get more and more information about it. This information can be stored in the database after verification and filling in the empty attributes.

When a new device is discovered, its details are added to the blockchain via a transaction. Smart contracts can then automatically verify compliance with organizational policies, such as checking for disabled remote access or enabled firewalls, and flag non-compliant devices for immediate action.

 doi:10.20944/preprints202505.1028.v2

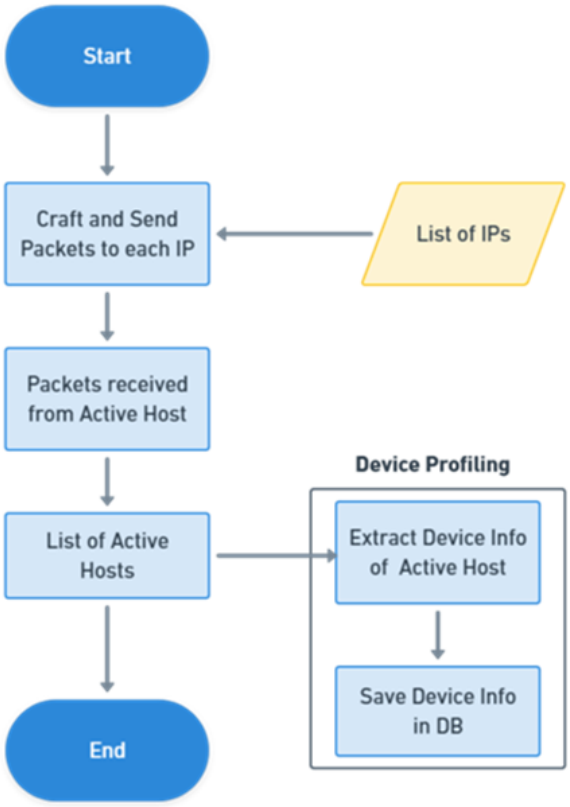

Figure 9. Device Discovery and Profiling Process in Network Security for Identifying and Managing Connected Devices in Enterprise Environments.

6.10. Traffic Monitoring

Although traffic monitoring is out of the scope of a network scanner, it will be very useful if a limited traffic monitoring functionality is added to it as almost all the scanners already have a Libpcap library to decode packets. This purpose can be achieved by connecting the scanner to a switch's mirroring port with a dedicated line to transfer a copy of each packet passing through the network toward the scanner. Only the packet header data is enough for analysis and record purposes and there is no need to inspect the packet payload as it requires heavy processing and machine learning algorithms.

Traffic logs, including packet header data (MAC addresses, IP addresses, and ports), are recorded on the blockchain. This provides an immutable record of network activities, enabling reliable analysis and traceability while preventing log manipulation by malicious actors.

From packet header data we can easily extract the MAC addresses, IP addresses, and port numbers, which can help us to recognize the devices and services they are using. The device monitoring process is illustrated in Figure 10.

 doi:10.20944/preprints202505.1028.v2

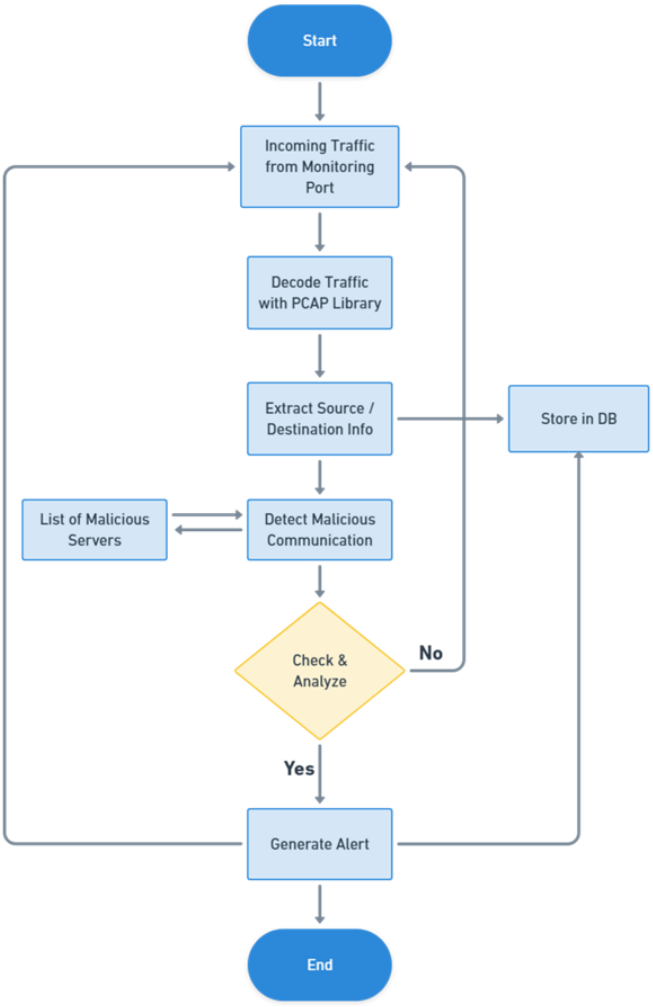

Figure 10. Real-Time Device Monitoring in Network Security Systems for Tracking and Analyzing Network Traffic and User Activity.

6.11. User Activity Logs

The user activity logs are important to know about a user's activeness or idleness on a network in order to evaluate their performance. The limited traffic monitoring feature of a scanner enables an organization to record the employee's activity hours and match it with the attendance management system. Furthermore, this feature can also help to detect any activity after office hours on a system or network.

User activity logs are secured on the blockchain, ensuring that records of employee actions, such as login times or policy violations cannot be altered. This enhances accountability and supports forensic investigations by providing a tamper proof activity history.

6.12. Network Forensics

Keeping a record of network activities (like, information about connected devices, service details, user activities, and various events) is getting important day by day. The linking of a database with a network scanner makes it more powerful to collect and store these records regularly. These records can be very helpful and can provide limited help (like, device type, OS info, IP address, services) in the network forensics process in case of any security incident or a cyber-attack. Therefore, network forensics feature availability in the advanced scanner will be quite useful for all kinds of organizations nowadays.

 doi:10.20944/preprints202505.1028.v2

The blockchain's immutable nature makes it ideal for network forensics. Records of device connections, traffic patterns, and security events stored on the blockchain provide a reliable, unalterable evidence base, facilitating detailed analysis following a security incident.

6.13. IDS/IPS Capability Detection

The purpose of the IDS/IPS is the timely identification and prevention of potential attacks from the internet. Nevertheless, the management of IDS/IPS systems may not be affordable for small to medium-sized enterprises. Again, this feature is out of the scope of a network scanner, but most of the scanners have the capability to craft network packets with the support of various libraries. Therefore, a scanner can be used to create spoofed packets to launch a flooding attack against a system in order to assess its capability to detect and prevent attacks. This feature will easily access the resistance of a network asset against flooding, spoofing, and denial of service attacks.

Results of IDS/IPS capability tests are logged on the blockchain, creating a verifiable history of the system's security posture. Smart contracts can analyze these results and trigger alerts or mitigation steps if vulnerabilities are detected, enhancing proactive defense.

6.14. Blockchain Integrity Checks

The scanner can verify the integrity of the blockchain ledger used to store security data, ensuring that logs and compliance records remain untampered. It can also monitor the proper execution of smart contracts, confirming that automated policy enforcement actions are triggered correctly. These checks enhance the overall security posture by ensuring the blockchain component operates as intended.

6.15. Smart Enforcement

As discussed in section II, network security policy enforcement is the key to basic security measures. A scanner can be a very useful tool to check security violations, like, shared remote access, firewall disabling, directory sharing, and connecting a personal device to a secured organizational network. Smart contracts on the blockchain are used to automatically check for policy violations. When a scanner detects an issue, such as a disabled firewall or an unauthorized device, the smart contract immediately takes action such as triggering an alert, logging the incident, or isolating the device. This reduces manual work and speeds up response time. Blockchain-enabled smart contracts automate policy enforcement by executing predefined actions when violations are detected. For example, if a scanner identifies a disabled firewall, a smart contract can isolate the device and log the event on the blockchain, ensuring rapid response and an audit able record of compliance actions.

6.16. Decentralized Compliance

Instead of relying on a central authority, the framework uses blockchain to allow multiple parties e.g. security tools, admins, or auditors to verify if the network is following its security policies. Since all critical logs and policy checks are recorded on a shared blockchain, everyone can independently verify compliance without any chance of data manipulation.

6.17. Algorithmic Complexity Analysis

The proposed blockchain-enhanced network scanning algorithm consists of various modular procedures, including device discovery, profiling, traffic monitoring, policy enforcement, and report generation. Let n denote the number of active devices in the network, and t denote the number of traffic packets monitored during a scan cycle.

- NetworkScan: Iterates through each detected device to check profile status and update records. The time complexity is $O(n)$.
- DeviceProfiling: Executed only for new devices, performing OS and software checks, identity mapping, and blockchain logging. Each profiling operation is performed in constant time per device, yielding a worst-case complexity of $O(n)$.

 doi:10.20944/preprints202505.1028.v2

- TrafficMonitoring: Captures and inspects packet headers, storing metadata in both the local database and blockchain. This results in linear complexity with respect to traffic volume: $O(t)$.
- PolicyEnforcement: For every device, a smart contract is triggered to verify compliance. Assuming constant-time contract evaluation per device, the complexity is $O(n)$.
- GenerateReports: Aggregates and formats logs from both storage layers. In a typical scan cycle, the operation scales with both devices and traffic logs, resulting in $O(n + t)$.

Overall Complexity: The total time complexity per scan cycle is $O(n + t)$, demonstrating linear scalability. The modular design allows for distributed or parallel processing of scanning and logging tasks. Blockchain integration introduces a fixed per-transaction overhead, which is manageable under typical network loads and does not hinder near real-time monitoring. This efficiency confirms the practicality of the approach for medium to large-scale organizational networks.

6.18. Customized Report Generation

All the aforementioned features of the scanning tool may not be very effective without the generation of comprehensive reports of the various network scans, monitoring activities, irregularities, policy violations, and other issues. An advanced scanning tool must provide comprehensive and multipurpose customized reports that are exportable in various formats, such as PDF, web-based. These reports can be shared with the organization's management and leadership to inform them about any malicious activity in the network, like spoofing, flooding. will be described in the irregularities report. In addition to that, many custom reports can also be generated on a daily, weekly, or monthly basis as per the demands of the organization. For example, a general scan report should be generated on a daily basis having information about various in-depth scans. Similarly, the policy enforcement reports in case of any non-compliance with the organizational policy detected from any of the devices in the network must be reported on a daily basis.

Reports are generated based on data retrieved from the blockchain, ensuring accuracy and integrity. These blockchain-backed reports, covering scan results, violations, and forensic data, can be shared with management in formats like PDF, providing a trustworthy overview of network security status.

The vulnerability reports can be generated weekly and contain detailed information about each connected device and patchable vulnerabilities found in these devices. The monitoring activity report that contains information on the host usage and its last active time can be generated every month. The proposed system will get the activity logs from the database saved by the limited traffic monitoring feature as discussed in section 5.6. In case the vulnerability is identified a host should monitor after informing the related person, whether the patches are installed or not. In the latter case, a report can be generated and sent to the organizational management for corresponding actions.

doi:10.20944/preprints202505.1028.v2

Table 2. Comparative Analysis Table of Existing Network, IP, Port, and Vulnerability Scanners Highlighting Features Like Firewall Status, Remote Desktop Status, and Policy Enforcement.

Name	FWS	RDS	VMD	UP	DI	CR	VD	NF	IDSE	SS	SD	PE	TM
NMAP	✓	×	✓	✓	✓	×	✓	×	×	×	×	×	×
SolarWinds Scanner	×	✓	×	×	✓	✓	✓	×	×	×	×	×	×
Advanced IP Scanner	×	✓	×	×	✓	✓	×	×	×	×	×	×	×
Angry IP Scanner	×	✓	×	×	✓	✓	×	×	×	×	×	×	×
Eusing IP Scanner	✓	×	×	×	✓	✓	×	×	×	×	×	×	×
NetCat	✓	✓	×	×	✓	✓	×	×	×	×	×	×	×
LanSweeper IP Scanner	✓	✓	✓	×	✓	×	✓	×	×	✓	×	×	×
MyLanViewer	✓	✓	×	✓	✓	✓	×	×	×	×	×	×	×
Slitheris Network Discovery	✓	×	✓	✓	✓	×	✓	×	×	×	×	×	×
Proposed Advanced Scanner	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

 doi:10.20944/preprints202505.1028.v2

In Table 2, a comparative analysis of existing network, port, IP, and vulnerability scanners is provided. It includes thirteen (13) features for comparison, covering both basic and advanced functions found in the latest scanning tools. This comparison highlights features that various tools provide including device discovery, vulnerability scanning, and basic monitoring but none integrate policy enforcement, blockchain logging, or decentralized verification capabilities. In this table, each checkmark (✓) indicates that the corresponding scanner supports the given feature, while a cross (×) indicates that the feature is not supported. The assessment was conducted based on official tool documentation, user manuals, and publicly available specifications. It is important to note that it focuses only on conventional scanning tools and does not include blockchain-based audit or compliance management solutions. Existing blockchain-based logging frameworks (e.g., Hyperledger-based audit chains or Ethereum-based compliance proofs) typically address data integrity and traceability but lack direct integration with real-time network scanning or policy enforcement mechanisms. In contrast, the proposed BENSAM framework uniquely combines network scanning with blockchain-based auditing and compliance verification. It integrates smart contract–driven policy enforcement, on-chain logging of scan results, and a hybrid storage model combining blockchain immutability with scalable off-chain data. This design enables verifiable, tamper-evident audit trails across multiple scanning agents, which is an advancement not addressed by existing blockchain logging or traditional scanning tools.

7. Open Challenges and Future Directions

This section outlines the open research challenges and future directions of the BENSAM framework, including considerations for AI-driven threat detection and adaptive network defense. A working prototype of the BENSAM framework has been implemented and is publicly available on GitHub, demonstrating end-to-end network scanning, device profiling, traffic monitoring, policy enforcement, and blockchain-based logging. While this prototype validates the feasibility of the design, several open challenges remain for full-scale deployment and operational evaluation. The framework currently incorporates essential features such as scheduled scanning, traffic monitoring, and device profiling, along with the integration of blockchain technology, which further enhances its capabilities by enabling decentralized verification, unalterable logging, and smart contracts. There are particular research challenges that are open and need to be addressed for further extension and are summarized in Table 3. These open challenges and future directions include:

- A software solution must be developed that implements all the proposed features in the form of a standalone advanced network scanning tool.
- Selecting and integrating a suitable blockchain platform (e.g., Ethereum or Hyperledger) to support smart contracts and ensure performance efficiency.
- Evaluating the proposed scanner in real-world conditions to measure its latency, scalability, and overall security performance.
- Investigating the use of advanced cryptographic techniques, such as zero-knowledge proofs, to enhance privacy without compromising enforcement capability.
- Reducing manual oversight traditionally required in compliance enforcement through automation and smart contract-based policy verification.
- Addressing AI-driven security threats that are becoming more common in modern networks, such as phishing created by language models, intelligent scanning tools, and attacks designed to bypass detection systems. These threats require new detection strategies that can adapt over time and use blockchain logging for tracing unusual behavior.

doi:10.20944/preprints202505.1028.v2

Table 3. Open Research Challenges and Future Directions.

Open Challenges	Description	Solution
Lack of integrated software solution	There is currently no standalone tool that combines device profiling, scheduled scanning, traffic monitoring, and blockchain-based policy enforcement into a unified system.	Development of a comprehensive network scanning tool
Blockchain platform selection	Selecting an appropriate blockchain platform (e.g. Ethereum or Hyperledger) is essential to support smart contracts while maintaining performance efficiency and scalability.	Evaluation and integration of a suitable blockchain platform
Performance evaluation in real-world settings	The proposed scanner must be tested under realistic network environments to validate its latency, scalability, and security.	Benchmark testing in operational scenarios
Privacy-preserving enforcement	Cryptographic methods like zero-knowledge proofs are needed to enhance user privacy without weakening policy enforcement.	Integration of advanced cryptographic techniques
Manual oversight in compliance enforcement	Traditional policy enforcement requires human monitoring, which is error-prone and inefficient.	Automation via smart contracts for policy verification
Emerging AI-driven threats	AI-generated phishing, intelligent evasion, and automated reconnaissance are becoming common. These threats are harder to detect using static rules and can bypass traditional defenses.	Use adaptive detection models and blockchain logging to trace unusual activity and improve response

 doi:10.20944/preprints202505.1028.v2

While the BENSAM Framework presents a robust and transformative approach to network security, it is important to acknowledge its limitations and outline future directions for research and development. The integrity of the system relies on the cryptographic link between the on-chain hash and the off-chain data. However, the availability of the raw off-chain data is not guaranteed by the blockchain itself. If the off-chain database is compromised or goes offline, the raw log files may be inaccessible. Future work could explore the integration of decentralized file systems, such as the InterPlanetary File System (IPFS), which has been explored in other DLT applications, to provide enhanced availability and censorship resistance for the off-chain data.

Additionally, the security of the framework remains dependent on the robustness of its chaincode. Vulnerabilities in the smart contract code could be exploited, highlighting the need for thorough and ongoing auditing by professional security experts prior to deployment. However, the working prototype demonstrates end-to-end functionality, integrating the multiple components i.e. network scanning agents, logging services, and the Hyperledger Fabric network, into a fully operational system requires careful coordination and technical expertise. Future work should focus on developing standardized APIs, modular interfaces, and automated deployment procedures to simplify integration, enhance maintainability, and facilitate wider adoption of the framework in realistic network environments.

8. Conclusion

Security policies play a critical role in safeguarding organizational networks, yet enforcing them at a granular level remains a persistent challenge. This paper introduced a blockchain-enhanced network scanning and monitoring (BENSAM) framework equipped with features such as device profiling, scheduled scanning, and traffic monitoring to improve the detection of policy violations. The integration of blockchain technology enhances this framework by enabling immutable logging, decentralized verification, and automated policy enforcement through smart contracts. These capabilities collectively ensure tamper-resistant records, minimize manual oversight, and strengthen trust in compliance processes. Compared to traditional approaches like OS hardening or SDN-based enforcement, the proposed blockchain-enhanced solution offers a more scalable, transparent, and cost-effective model for network policy enforcement. By addressing both traditional network vulnerabilities and emerging AI-driven attacks, the proposed blockchain-enhanced scanner provides a unified, tamper-proof framework for automated policy enforcement and resilient security management. A prototype implementation of the BENSAM framework has been developed, confirming the feasibility of the proposed architecture. This prototype validates end-to-end network scanning, policy enforcement, and blockchain-based immutable logging, providing a foundation for future experimental evaluation and extension to live network environments. By combining adaptive detection mechanisms with blockchain-based traceability, the proposed solution can evolve to meet modern security challenges more effectively. Ultimately, this work demonstrates how blockchain and intelligent detection can converge to create a more trustworthy and future-ready foundation for organizational cybersecurity.

Abbreviations

The following abbreviations are used in this manuscript:

 doi:10.20944/preprints202505.1028.v2

SYN	Synchronize
CR	Custom Reports
UP	User Profiling
CC	Common Criteria
FWS	Firewall Status
OS	Operating System
NF	Network Forensics
IP	Internet Protocol
PE	Policy Enforcement
SS	Scheduled Scanning
DNS	Domain Name System
TM	Traffic Monitoring
SD	Structured Database
IDSE	IDS/IPS Evaluation
VMD	Three letter acronym
RDS	Remote Desktop Status
VD	Vulnerability Detection
UDP	User Datagram Protocol
NAC	Network Access Control
VMD	Virtual Machine Detection
VMD	Virtual Machine Detection
ARP	Address Resolution Protocol
SDN	Software-Defined Networking
TCP	Transmission Control Protocol
MPLS	Multiprotocol Label Switching
EDR	Endpoint Detection and Response
ICMP	Internet Control Message Protocol
RARP	Reverse Address Resolution Protocol
FIPS	Federal Information Processing Standard
NIST	National Institute of Standards and Technology
BENSAM	Blockchain-Enhanced Network Scanning and Monitoring

References

1. Douligeris, C.; Mitrokotsa, A. DDoS Attacks and Defense Mechanisms: Classification and State-of-the-Art. *Computer Science Review* **2022**, *44*, 100458.
2. Saleem, B.; Ahmed, M.; Zahra, M.; Hassan, F.; Iqbal, M.A.; Muhammad, Z. A survey of cybersecurity laws, regulations, and policies in technologically advanced nations: A case study of Pakistan to bridge the gap. *International Cybersecurity Law Review* **2024**, *5*, 533–561.

 doi:10.20944/preprints202505.1028.v2

3. Muhammad, Z.; Straub, J. An Analysis of Cyber Threats and the Protective Role of Cyber Insurance in the US Market. In Proceedings of the World Congress in Computer Science, Computer Engineering & Applied Computing. Springer, 2024, pp. 259–272.
4. Dissanayake, N.; Jayatilaka, A.; Zahedi, M.; Babar, M.A. Software security patch management-A systematic literature review of challenges, approaches, tools and practices. *Information and Software Technology* **2022**, *144*, 106771.
5. Rahman, A.; Kawshik, K.R.; Sourav, A.A.; Gaji, A.A. Advanced Network Scanning. *American Journal of Engineering Research (AJER)* **2016**, *5*, 38–42.
6. Buczak, A.L.; Guven, E. A Survey of Data Mining and Machine Learning Methods for Cyber Security Intrusion Detection. *IEEE Communications Surveys & Tutorials* **2021**, *23*, 1121–1153.
7. Taylor, R.W.; Fritsch, E.J.; Liederbach, J. *Digital crime and digital terrorism*; Prentice Hall Press, 2014.
8. Zhang, C.; Hu, G.; Chen, G.; Sangaiah, A.K.; Zhang, P.; Yan, X.; Jiang, W. Towards a SDN-based integrated architecture for mitigating IP spoofing attack. *IEEE Access* **2017**, *6*, 22764–22777.
9. National Institute of Standards and Technology. Framework for Improving Critical Infrastructure Cybersecurity, Version 1.1. https://www.nist.gov/cyberframework/csf-11-archive, 2018. Accessed on 5 Jan 2025.
10. Nmap Project. Zenmap—Official Cross-Platform Nmap Security Scanner GUI. https://nmap.org/zenmap/, 2024. Accessed on 29 November 2024.
11. Sarker, I.H.; Kayes, A.; Badsha, S.; Alqahtani, H.; Watters, P.; Ng, A. Cybersecurity data science: an overview from machine learning perspective. *Journal of Big data* **2020**, *7*, 1–29.
12. Hamdani, S.W.A.; Abbas, H.; Janjua, A.R.; Shahid, W.B.; Amjad, M.F.; Malik, J.; Murtaza, M.H.; Atiquzzaman, M.; Khan, A.W. Cybersecurity standards in the context of operating system: Practical aspects, analysis, and comparisons. *ACM Computing Surveys (CSUR)* **2021**, *54*, 1–36.
13. Berardi, D.; Callegati, F.; Melis, A.; Prandini, M. Security network policy enforcement through a SDN framework. In Proceedings of the 2018 28th International Telecommunication Networks and Applications Conference (ITNAC). IEEE, 2018, pp. 1–4.
14. Khan, A.A.; Laghari, A.A.; Shaikh, Z.A.; Dacko-Pikiewicz, Z.; Kot, S. Internet of Things (IoT) security with blockchain technology: A state-of-the-art review. *IEEE Access* **2022**, *10*, 122679–122695.
15. Singh, S.; Hosen, A.S.; Yoon, B. Blockchain security attacks, challenges, and solutions for the future distributed iot network. *Ieee Access* **2021**, *9*, 13938–13959.
16. Fotohi, R.; Aliee, F.S. Securing communication between things using blockchain technology based on authentication and SHA-256 to improving scalability in large-scale IoT. *Computer Networks* **2021**, *197*, 108331.
17. Rani, S.; Babbar, H.; Srivastava, G.; Gadekallu, T.R.; Dhiman, G. Security framework for internet-of-things-based software-defined networks using blockchain. *IEEE Internet of Things Journal* **2022**, *10*, 6074–6081.
18. Hsiao, S.J.; Sung, W.T. Employing blockchain technology to strengthen security of wireless sensor networks. *IEEe Access* **2021**, *9*, 72326–72341.
19. Khan, A.A.; Bourouis, S.; Kamruzzaman, M.; Hadjouni, M.; Shaikh, Z.A.; Laghari, A.A.; Elmannai, H.; Dhahbi, S. Data security in healthcare industrial internet of things with blockchain. *IEEE Sensors Journal* **2023**, *23*, 25144–25151.
20. Sharma, P.; Borah, M.D.; Namasudra, S. Improving security of medical big data by using Blockchain technology. *Computers & Electrical Engineering* **2021**, *96*, 107529.
21. Nguyen, D.C.; Pathirana, P.N.; Ding, M.; Seneviratne, A. BEdgeHealth: A decentralized architecture for edge-based IoMT networks using blockchain. *IEEE Internet of Things Journal* **2021**, *8*, 11743–11757.
22. Tandon, R.; Verma, A.; Gupta, P. D-BLAC: A dual blockchain-based decentralized architecture for authentication and communication in VANET. *Expert Systems With Applications* **2024**, *237*, 121461.
23. Grover, J. Security of Vehicular Ad Hoc Networks using blockchain: A comprehensive review. *Vehicular Communications* **2022**, *34*, 100458.
24. Bendiab, G.; Hameurlaine, A.; Germanos, G.; Kolokotronis, N.; Shiaeles, S. Autonomous vehicles security: Challenges and solutions using blockchain and artificial intelligence. *IEEE Transactions on Intelligent Transportation Systems* **2023**, *24*, 3614–3637.
25. Farooq, M.S.; Khan, S.; Rehman, A.; Abbas, S.; Khan, M.A.; Hwang, S.O. Blockchain-based smart home networks security empowered with fused machine learning. *Sensors* **2022**, *22*, 4522.
26. Hızal, S.; Akhter, A.S.; Çavuşoğlu, Ü.; Akgün, D. Blockchain-based IoT security solutions for IDS research centers. *Internet of Things* **2024**, *27*, 101307.

 doi:10.20944/preprints202505.1028.v2

27. Lv, Z.; Qiao, L.; Hossain, M.S.; Choi, B.J. Analysis of using blockchain to protect the privacy of drone big data. *IEEE network* **2021**, *35*, 44–49.
28. Li, C.; Sun, X.; Zhang, Z. Effective methods and performance analysis of a satellite network security mechanism based on blockchain technology. *IEEE Access* **2021**, *9*, 113558–113565.
29. Dhar, S.; Khare, A.; Dwivedi, A.D.; Singh, R. Securing IoT devices: A novel approach using blockchain and quantum cryptography. *Internet of things* **2024**, *25*, 101019.
30. Toyeer-E-Ferdoush.; Rahman, H.; Hasan, M. A convenient way to mitigate DDoS TCP SYN flood attack. *Journal of Discrete Mathematical Sciences and Cryptography* **2022**, *25*, 2069–2077.
31. Jia, Y.J.; Chen, Q.A.; Lin, Y.; Kong, C.; Mao, Z.M. Open doors for bob and mallory: Open port usage in android apps and security implications. In Proceedings of the 2017 IEEE European Symposium on Security and Privacy (EuroS&P). IEEE, 2017, pp. 190–203.
32. Bhakthavatsalam, P.; Malarkodi, B. Analysis of network infrastructure threats using SonicWALL analyser. In Proceedings of the 2016 3rd International Conference on Devices, Circuits and Systems (ICDCS). IEEE, 2016, pp. 6–9.
33. Rahman, A.; Kawshik, K.R.; Sourav, A.A.; Gaji, A.A. Advanced Network Scanning. *American Journal of Engineering Research (AJER)* **2016**, *5*, 38–42.
34. Kumar, S. Classification and detection of computer intrusions. PhD thesis, PhD thesis, Purdue University, 1995.
35. Mohammed, S.A. Designing Rules to Implement Reconnaissance and Unauthorized Access Attacks for Intrusion Detection System. *Iraqi Journal of Information & Communications Technology* **2019**, *2*, 25–43.
36. Shahid, J.Z.; Cimato, S.; Muhammad, Z. A Sharded Blockchain Architecture for Healthcare Data. In Proceedings of the 2024 IEEE 48th Annual Computers, Software, and Applications Conference (COMPSAC). IEEE, 2024, pp. 1794–1799.
37. Lichtblau, F.; Streibelt, F.; Krüger, T.; Richter, P.; Feldmann, A. Detection, classification, and analysis of inter-domain traffic with spoofed source IP addresses. In Proceedings of the Proceedings of the 2017 Internet Measurement Conference, 2017, pp. 86–99.
38. Singh, R.; Thakur, K.; Singh, G.; Gupta, S. Prevention of IP spoofing attack in cyber using artificial Bee colony and artificial neural network. In Proceedings of the Proceedings of the Third International Conference on Advanced Informatics for Computing Research, 2019, pp. 1–10.
39. Verisign, Inc.. Q4 2016 DDoS Trends Report: 167 Percent Increase in Average Peak Attack Size. https://blog.verisign.com/security/q4-2016-ddos-trends-report-167-percent-increase-average-peak-attack-size/, 2017.
40. Deka, R.K.; Bhattacharyya, D.K.; Kalita, J.K. Granger Causality in TCP Flooding Attack. *IJ Network Security* **2019**, *21*, 30–39.
41. Ahmed, A.A.; Zaman, N.A.K. Attack Intention Recognition: A Review. *IJ Network Security* **2017**, *19*, 244–250.
42. Baishya, R.C.; Hoque, N.; Bhattacharyya, D.K. DDoS Attack Detection Using Unique Source IP Deviation. *IJ Network Security* **2017**, *19*, 929–939.
43. Sattar, I.; Shahid, M.; Abbas, Y. A review of techniques to detect and prevent distributed denial of service (DDoS) attack in cloud computing environment. *International Journal of Computer Applications* **2015**, *115*.
44. Sun, J.R.; Hwang, M.S. A New Investigation Approach for Tracing Source IP in DDoS attack from Proxy Server. In Proceedings of the ICS, 2014, pp. 850–857.
45. Jung, J.; et al. Real-time detection of malicious network activity using stochastic models. PhD thesis, Massachusetts Institute of Technology, 2006.
46. Arshad, J.; Talha, M.; Saleem, B.; Shah, Z.; Zaman, H.; Muhammad, Z. A Survey of Bug Bounty Programs in Strengthening Cybersecurity and Privacy in the Blockchain Industry. *Blockchains* **2024**, *2*, 195–216.
47. Teodoro, N.; Gonçalves, L.; Serrão, C. NIST CyberSecurity Framework Compliance: A Generic Model for Dynamic Assessment and Predictive Requirements. In Proceedings of the 2015 IEEE Trustcom/BigDataSE/ISPA. IEEE, 2015, Vol. 1, pp. 418–425.
48. Greenbone Networks GmbH. OpenVAS—Open Vulnerability Assessment Scanner. https://www.openvas.org/, 2024.
49. Irfan, M.; Ali, S.T.; Ijlal, H.S.; Muhammad, Z.; Raza, S. Exploring the synergistic effects of blockchain integration with IoT and AI for enhanced transparency and security in global supply chains.
50. Islam, M.B.E.; Haseeb, M.; Batool, H.; Ahtasham, N.; Muhammad, Z. AI threats to politics, elections, and democracy: a blockchain-based deepfake authenticity verification framework. *Blockchains* **2024**, *2*, 458–481.
51. Daidone, F.; Carminati, B.; Ferrari, E. Blockchain-based privacy enforcement in the IoT domain. *IEEE Transactions on Dependable and Secure Computing* **2021**, *19*, 3887–3898.

 doi:10.20944/preprints202505.1028.v2

52. Sharma, S.; Chen, Z. A Systematic Study of Adversarial Attacks Against Network Intrusion Detection Systems. *Electronics* **2024**, *13*, 5030.
53. Li, Y. Data Poisoning in Network Anomaly Detection Systems. PhD thesis, Carnegie Mellon University, 2024.
54. Du, W.; Xue, J.; Yang, X.; Guo, W.; Gu, D.; Han, W. TransfficFormer: A novel Transformer-based framework to generate evasive malicious traffic. *Knowledge-Based Systems* **2025**, p. 113546.
55. Schiffman, M. The libnet packet construction library. *The Million Packet March* **2005**.
56. Liu, W. Design and implement of common network security scanning system. In Proceedings of the 2009 International Symposium on Intelligent Ubiquitous Computing and Education. IEEE, 2009, pp. 148–151.
57. Calderon, P. *Nmap Network Exploration and Security Auditing Cookbook: Network discovery and security scanning at your fingertips*; Packt Publishing Ltd, 2021.
58. Jacobson, V.; McCanne, S. libpcap: Packet capture library. *Lawrence Berkeley Laboratory, Berkeley, CA* **2009**.
59. Shuguang, W.; Gaogang11, X. libpcap-MT: A General Purpose Packet Capture Library with Multi-Thread. *Journal of Computer Research and Development* **2011**, *5*, 756–764.
60. McCanne, S.; Jacobson, V. The BSD Packet Filter: A New Architecture for User-level Packet Capture. In Proceedings of the USENIX winter, 1993, Vol. 46.
61. Nmap: The Network Mapper, A Powerful Open Source Tool for Network Discovery and Security Auditing. https://nmap.org/, 2024.
62. Shah, M.; Ahmed, S.; Saeed, K.; Junaid, M.; Khan, H.; et al. Penetration Testing Active Reconnaissance Phase–Optimized Port Scanning With Nmap Tool. In Proceedings of the 2019 2nd International Conference on Computing, Mathematics and Engineering Technologies (iCoMET). IEEE, 2019, pp. 1–6.
63. Lyon, G. Nmap: The network mapper–Free security scanner. *Nmap. org* **2016**.
64. Dutta, N.; Jadav, N.; Dutiya, N.; Joshi, D. Using Honeypots for ICS Threats Evaluation. In *Recent Developments on Industrial Control Systems Resilience*; Springer, 2020; pp. 175–196.
65. Advanced IP Scanner — Fast and Reliable Network Scanning Tool. https://www.advanced-ip-scanner.com/, 2024.
66. Angry IP Scanner Project. Angry IP Scanner — Fast and Friendly Network Scanner. https://angryip.org/, 2023.
67. Tian, D. Angry IP Scanner: A Lightweight and Efficient Network Scanning Tool for Cybersecurity Applications. *Cyber Security and Information* **2017**, p. 87.
68. Eusing Software. Free IP Scanner by Eusing — A Fast and Simple Network Scanning Tool. https://www.eusing.com/ipscan/free_ip_scanner.htm, 2022.
69. Baloch, R. *Ethical hacking and penetration testing guide*; CRC Press, 2017.
70. The GNU Netcat, A Versatile Networking Utility for Reading and Writing Data Across Networks. http://netcat.sourceforge.net/, 2021.
71. Kurth, M.; Gras, B.; Andriesse, D.; Giuffrida, C.; Bos, H.; Razavi, K. NetCAT: Practical Cache Attacks from the Network, 2020.
72. LanSweeper. LanSweeper IP Scanner — Comprehensive Network Scanning and Asset Discovery Tool. https://www.lansweeper.com/feature/ip-scanner/, 2022.
73. Erlandson, R. Finding Help and Keeping Up with Changing Technology in Libraries. *Technology for Small and One-Person Libraries: A LITA Guide* **2013**, *21*, 125.
74. HAFSAOUI, M.A.; MANSOUR, H. D 'e development of a computer park management application. PhD thesis, Universit 'e Virtual of Tunis, 2019.
75. MyLanViewer Network/IP Scanner, User-Friendly Network Discovery and Monitoring Tool. http://www.mylanviewer.com/network-ip-scanner.html, 2020.
76. Garcia, H.C. About monitoring the confidentiality of computer systems. *Mathematical machines and systems* **2015**.
77. Dandan, T. LAN scan MyLanViewer. *Cyber security and information* **2017**, p. 94.
78. Tenable Inc.. Nessus — Industry-Leading Vulnerability Assessment Tool. https://www.tenable.com/products/nessus, 2020.
79. Anderson, H. Introduction to Nessus. *SecurityFocus* **2003**. Available at http://www.securityfocus.com/infocus/1741.
80. Wijaya, S.A.A. ATCS System Security Audit Using Nessus. *ATCS* **2017**, *7*.

 doi:10.20944/preprints202505.1028.v2

81. Josephlal, E.F.M.; Adepu, S. Vulnerability Analysis of an Automotive Infotainment System's WIFI Capability. In Proceedings of the 2019 IEEE 19th International Symposium on High Assurance Systems Engineering (HASE). IEEE, 2019, pp. 241–246.
82. Memon, I.; Shaikh, R.A.; Fazal, H.; Tunio, H.; Arain, Q.A. The World of Hacking: A Survey. *University of Sindh Journal of Information and Communication Technology* **2020**, *4*, 31–37.
83. Zhang, W.; Banescu, S.; Pasos, L.; Stewart, S.; Ganesh, V. MPro: Combining Static and Symbolic Analysis for Scalable Testing of Smart Contract. *arXiv preprint arXiv:1911.00570* **2019**.
84. Singh, R.R.; Tomar, D.S. Network forensics: detection and analysis of stealth port scanning attack. *scanning* **2015**, *4*, 8.
85. Patel, S.K.; Sonker, A. Internet protocol identification number based ideal stealth port scan detection using snort. In Proceedings of the 2016 8th International Conference on Computational Intelligence and Communication Networks (CICN). IEEE, 2016, pp. 422–427.
86. Coyle, S. Port Scanning Techniques, Tools, and Detection. *Cybersecurity Journal* **2024**.
87. Qureshi, M.A.; Ahmed, S.; Mehmood, A.; Shaheen, R.; Dildar, M.S. Vulnerability assessment of operating systems in healthcare: exploitation implications techniques and security. *Health Sciences Journal* **2024**, *2*, 104–111.
88. Mitre Corporation. Common Vulnerabilities and Exposures (CVE) Database. https://cve.mitre.org/, 2022.
89. Bonandir, A.; Yussof, S. An analysis of common vulnerability and exposure (CVE) of software products in the year 2016. *International Journal of Advanced Science and Technology* **2018**, *112*, 157–166.
90. Cisco. What is a firewall? A primer on firewalls. https://www.cisco.com/site/us/en/learn/topics/security/what-is-a-firewall.html, 2025.