*Journal of Computer Science*

Research Article

# Enhancing Network Security through Accurate Asset Discovery Using Lightweight Agent-Based Approach

**Ishu Sharma[1, 3], Ahthasham Sajid[2, 3], Mazliham Mohd Suud[3] and Muhammad Mansoor Alam[3, 4]**

[1]*School of Engineering and Technology, CGC University Mohali – 140307 and Research Management Centre, Multimedia University, Cyberjaya, Selangor, Malaysia*
[2]*Department of Computer Science, Fazaia Bilquis College of Education for Women, PAF, Nur Khan , Rawalpindi, Air University, Islamabad, Pakistan*
[3]*Research Management Centre, Multimedia University, Cyberjaya, Selangor, Malaysia*
[4]*Faculty of Computing, Riphah International University, Islamabad, Pakistan*

*Article history*
Received: 08-06-2025
Revised: 18-07-2025
Accepted: 01-08-2025

**Corresponding Author:**
Ishu Sharma
School of Engineering and Technology, CGC University Mohali – 140307 and Research Management Centre, Multimedia University, Cyberjaya, Selangor, Malaysia
Email: ishu.sharma001@gmail.com

**Abstract:** Any organization's network administrators and security professionals must have an accurate and real state of assets connected to the network to devise different policies for securing critical resources by identifying vulnerabilities. Asset discovery is a challenging and critical task in vulnerability assessment. The vulnerability of the organization's network can be correctly analysed only with the accurate prediction of connected devices. In this paper, the recent research activities in asset identification are explored to identify the gaps for future research directions. This paper provides a systematic review of the techniques that can be utilized for asset identification tasks in the organization. The comparative analysis presented in this paper lists the parameters required to choose a specific policy for network discovery. The proposed lightweight agent-based technique is inspired by socket communication and can fetch the accurately installed application software details on remote devices. The proposed method is tested on the campus area network by mapping Nmap Scanning results with the results achieved from the proposed methodology to optimize the network discovery task. The state-of-the-art network discovery tool Nmap provides probability-based results, and our proposed lightweight agent-based technique can enhance the finding of Nmap with exact details on IT assets.

**Keywords:** IT Asset Discovery, Network Security, Lightweight Agent Based Approach, Network Discovery Optimization, Hybrid Discovery Methodologies

## Introduction

*Context and Motivation*

Network discovery is the foremost step for network administrators of an organization to understand the level of network security in the system. Asset scanning or discovery enables users to keep track of all resources of an organization. Efficient management of assets in an organization ensures reliable computing infrastructure for supporting the goal of the firm. The new trend in the industry emphasizes the Bring Your Own Device (BYOD) approach to achieving optimized resource management. However, this approach comes with the challenge of discovering all connected devices in the network. There is also a requirement to get all the details of services running in all connected devices to ensure the authenticity of network users. The level of security can only be applied or measured when there is the availability of exact information about all the assets of the network. The risk analysis approach can predict the threat in the network by having clear visibility of all the associated assets of the network. The standard ISO/IEC 27001 (International Organization for Standardization, 2020) also considers the assets identification process as a crucial and important phase of the cybersecurity domain.

The inventory of the assets in an organization keeps updating with the entry and exit of the devices. Interestingly Information Technology (IT) asset discovery is targeted by both network administrators and hackers. The visibility of the network assets allows one to have control over the network whether he is an administrator or hacker. Generic devices like cameras,

laptops, desktops, printers, sensors etc can be connected to the organization's network and are capable of sharing network information with others through the Internet. Some devices can enter the network as intruders to fetch maximum information about the infrastructure, so the security policy implied by the organization should be strong enough to track such connectivity. Hence, efficient asset discovery is the backbone of the network security domain.

Many methods and tools can be employed to fetch the network details. The challenges of cybersecurity in a digitized world increase daily and hence network administrators and managers must devise optimized policies for securing all sensitive resources or payloads. It is always good to use multiple tools to get complete visibility of a network. The network administrators and managers need to know about the features associated with each asset discovery policy to make the system stand against intruders. Many researchers are also looking to contribute to the field of building an optimized solution for accurate and reliable asset identification. For moving in the same direction, one must clearly understand the existing techniques. In this paper, we have presented the latest techniques being used for asset identification, the limitations of these techniques and the future scope in this area. Reference (Ghadi et al., 2024) describes how machine learning methods are used in wireless sensor networks to overcome security issues. Our research is further motivated by the issues and solutions related to IoT security found in Reference (Mazhar et al., 2023a), which suggests that artificial intelligence can be a key factor in overcoming the difficulties related to medical data extraction. Reference (Mazhar et al., 2023b) proposed smart grid security using machine learning algorithms; however, gaps still exist due to the lack of accurate information about the network. The cybersecurity governed by artificial intelligence can be effective only with the right network discovery information. The basic network results lack the exact details of application software and build versions.

The databases like national vulnerability database and google hacking database provide the vulnerability data that are included in particular application software and build version. The prevention and mitigation methods cannot be employed without having accurate information of the assets in the network. Hence, it is critical to get the precise information of all the assets in the network. The existing methods or tools work on the probabilistic results of the network scanning. The methods based on artificial intelligence also work with traffic patterns to classify normal and malicious packets but the vulnerabilities related to application software, web application programming interface and mobile applications can be easily exploited by cyberattackers without getting alert or action from artificial intelligence methods. Our research work targets this crucial requirement of the cybersecurity domain and provides the open source solution to the community without having any need to paid software or tools.

### *Research Contributions*

This paper presents three primary research contributions. First, it identifies critical gaps in existing methodologies for IT asset discovery, highlighting limitations that hinder comprehensive network visibility. To address these shortcomings, the study introduces a novel lightweight agent-based approach designed to enhance network discovery outcomes without relying on any commercial third-party automated discovery tools. Finally, the proposed model is integrated with Zenmap, an open-source tool, to aggregate and refine discovery results, enabling the retrieval of precise details about installed application software, including build versions, on remote Windows-based devices.

## *Literature Survey*

### *Current Research*

Reliable asset identification has become a necessity in the cybersecurity domain for the proper management of the network in any organization. Researchers are actively working to devise an optimized solution to get the actual state of the network despite the number of users in the network or the size of the network. In this section, the latest research work for asset discovery is explained in detail to understand the different policies used by the researchers for complete network scanning. Chatzipoulidis et al. (2015) mentioned that in the proposed work of vulnerability analysis network security tools (Kaushik et al., 2020) Nmap and Nikto2 (Ma et al., 2012) have been used to identify the operating system and web servers respectively. Risk prediction methodology is proposed for the e-banking sector and software platforms are identified by Common Platform Enumeration (CPE) specifications. The historical rate of vulnerability occurrences is analysed based on the category of the software platform. The distribution fitting procedure is used to predict the vulnerability of the e-banking sector. Industrial processes (Coffey et al., 2018) are controlled by a combination of software and hardware elements in the form of Supervisory Control and Data Acquisition (SCADA). In this work, the authors analysed the network scanning tools for the discovery of assets and services in the SCADA systems. The laboratory setup detailed in the paper consists of a small SCADA system to replicate the functionality of the Programmable Logic Controller (PLC) and Human Machine Interface (HMI) device. Network scanning tools Nmap, Zmap, Tshark (Tsoukalos, 2015) as well as python scripts with UDP Denial of Service (DoS) are utilized to scan the SCADA network.

Martínez et al. (2020) used the Nmap tool to scan all the open ports in the cyber space-oriented network. The

port scanning performed in the proposed approach is capable of getting information about the service version and operating system. The details collected by the proposed method are stored in the internal repository with the asset criticality score. The scoring system is based on traffic, vulnerability, asset task, and task severity. In Wang (2020), the topology discovery technique Simple Network Management Protocol (SNMP) has been analysed in laboratory set-up and improvement to the existing protocol has been proposed. The proposed approach suggests making changes in the packet structure of SNMP protocol to handle the discovery of heterogeneity of the network. The improvement consists of changes in data acquisition method, router IP addressing and improvement of system topology level.

The Internet of Things (IoT) is a collection of multiple devices forming a direct or indirect network with heterogeneous devices. Feng et al. (2018) proposed a rule-based method for device discovery in IoT networks. An Acquisitional Rule-based Engine (ARE) forms rules for device discovery without using any training dataset. This method is dependent on the responses from the application layer and product information stored in device start-up webpages. The rule-based engine works in four components transaction collection, rule mining, storage, and plan executor. The description given on the product start-up webpages can be retrieved using web crawlers. An automated rule miner module is built upon as part of the second component of the rule-based engine. The discovered rules are stored in the rule database concerning timestamps. The fourth component works as the supervisor for the rule-based engine. In the experimental setup, multiple honeypots are deployed to capture malicious activities on an external network. The proposed method is used to identify the compromised IoT devices by inputting scanned IP addresses.

Jung et al. (2021) proposed UDP based methods for IT asset discovery for Internet of Things devices. The proposed algorithm works in two phases: Primary scan and auxiliary scan. In the primary scan, the unicast packet is sent into the network to fetch device information accessing the SSDP protocol. The second phase of the proposed method uses a unicast packet to fetch device information with protocols NetBIOS Name Service (NBNS) (Vazquez, 2019) and Multicast DNS (MDNS). An experimental setup was taken with real-life IoT devices to test the proposed algorithm and results are being compared with the Nmap tool in terms of speed and accuracy of results. Zhu et al. (2021) proposed WND-Identifier for reliable identification of wireless devices without setting the network card to promiscuous mode. WND-Identifier works in three phases: Database collection, device identification based on rules and text classification based on text. In the first phase database related to the device, identification is collected from the protocol which carries the device-related information. The information related to cameras and routers is accessed through the management webpage of these devices during the second phase of the proposed method. K-Nearest Neighbours (KNN) classifier is utilized for labelling the connected devices and at the same time, malicious devices can be detected with text-based classification.

Nmap is a powerful network scanning tool, and it is widely used by system administrators. Redondo and Cuesta (2019) developed a web application to provide immediate access to most of the features of Nmap without using complex scripts. The limitations of the official interface of Nmap, Zenmap (Rahalkar, 2019) are discussed in the paper and that sets the objective for the new interface for the tool. NmapGUI allows users to create multiple scans based on customized scripts. Zheng et al. (2021) made a tool WebHunt for assessing the network security of campus networks. The assets are discovered in the proposed tool using the reverse resolution of domain names, and detection of live devices and software are identified with fingerprinting. The user is required to enter seed links and rules based on the network settings and the crawler will start working to obtain the network information. The forward and reverse resolution for DNS names is being utilized to have IT addresses corresponding to the used domains in the campus network. Further active scanning is used to identify the connected devices in the network. The port scanning task has been performed by using the status of default ports, Masscan and Nmap tool. The service and version of the devices are fetched using socket connection establishment and Nmap. Web fingerprint identification is utilized to get the details of web servers. WebHunt tool has been experimented with for the campus network of seven universities in China and the result comparison is presented with tools Shodan, FOFA (Li et al., 2020) and Zoomeye (Tundis, 2018).

The generic behaviour of the active probing or scanning method is dependent on ICMP packets. The request and reply approach with the ICMP packet make the processing time-consuming. He et al. (2017) proposed an automatic discovery method based on the MDNS protocol. The proposed method can be implemented for small networks in the absence of a DNS server. The authors presented a graphical user interface to run the MDNS protocol for automatic network discovery. As the Graphical User Interface (GUI) starts, the multicast system sends messages to all connected devices on the Local Area Network (LAN). The data is collected for the IP address and services on the devices. The result analysis shows that the MDNS method takes 30% less time in automatic discovery as compared to the Internet Control Message Protocol (ICMP packet).

The updated information of contextual information is required by network security personnel at different levels. Husák et al. (2021) proposed the same issue to give time to time contextual information about the network. The

proposed design uses a combination of active and passive network scanning. Netflow and Nmap tools are utilized for passive and active scanning respectively. The Operating System identification of the assets is fetched using Passive Operating system fingerprinting. The proposed system uses a scanner module to handle large networks and it divides the network into a specific range of IP addresses. Later, the output from multiple modules is merged. The additional information about the open ports is achieved from the Nmap tool. The Webchecker module checks the adoption of HTTP and HTTPS for different websites. Content Management System has also been included in the proposed model using the tool WhatWeb. SNMP protocol is generally used to fetch network discovery-related information for small networks as it may result in latency for large networks. Espinel Villalobos et al. (2021) proposed a multi-layer method for large networks using the basics of SNMP protocol. The testing of the proposed algorithm is executed on campus networks. In the experimental setup, the services are installed on the virtual server only with a private cloud server. The method consists of three layers: Data collection, processing and application layer. The collector agents can fetch the number of users' information at the peak hour of the network as well. The density of each building can also be categorized by the proposed method. The authors used the python library for the implementation of the multi-agent SNMP protocol.

Wang et al. (2022) presented a method for IP-based core network for asset discovery using multiple autoencoders. In the first step, data is collected from the network about the connected assets. Data pre-processing is achieved by using feature engineering on the collected data. The proposed method uses pre-trained multiple autoencoders for network discovery. The unbalanced autoencoders are used for the construction of the neural network. The unlabeled asset data can pre-train the neural network using the unsupervised technique. The hidden output of one autoencoder is assigned as input to the next autoencoder. The experimental setup is tested with the proposed technique using a prototype model. The fingerprinting method is widely used to identify the Operating system, applications and services running on the connected devices. Lastovicka et al. (2020) proposed a method to handle fingerprinting in an encrypted communication scenario. A machine learning model is trained to Transport Layer Security (TLS) handshake to identify the operating system. A comparative analysis of the discussed research work has been presented for a better understanding of the discussed methodologies. The comparative analysis gives a more systematic view of choosing among different policies that have been employed for asset identification tasks.

The comparison of OCR tools like Tesseract and Google Cloud Vision API by Mazhar et al. in their study on Thai vehicle registration certificates demonstrates the potential for significant improvements in accuracy through machine learning. This finding is crucial for our research as it points toward the necessity of adopting advanced computational methods, such as those discussed in Ghadi et al. (2024), for enhancing the security and efficiency of wireless sensor networks. Such methodologies could be adapted for OCR applications in the medical field, suggesting a promising avenue for our proposed integration of OCR and RAKE algorithms.

### *Research Gap*

The current survey and study, based on the literature review on IT asset discovery, reveal several issues that this paper aims to address. There has been some improvement in the available techniques as far as finding network assets is concerned; however, some challenges still remain, especially in dealing with the complicated organizational networks that change over time.

In the first place, the present study suggests that much research depends on such customary tools as Nmap, which are good enough but not efficient in portraying a detailed and full image of all network assets, particularly under the BYOD regime and the growing IoT devices trend. This gap serves as an indicator of a need for a more sophisticated and dynamic approach to asset discovery due to the advancement of technology in organizational networks.

Moreover, the incorporation of state-of-the-art computational techniques, including machine learning and artificial intelligence, in the identification of assets is an area that has not yet been extensively explored. The capabilities of these technologies to improve the precision and speed at which asset recognition procedures are carried out have not yet been fully tapped. This space is seen as a clear sign of the scope of innovation in making use of these breakthrough technologies in asset discovery approaches. Another point is the inadequacy of scalability and adaptability in present asset discovery systems, which can accommodate the increasing requirements of organizations. Since organizations are constantly changing, their scope of asset discovery should also change so that no network resources will be missing.

To recapitulate, one of the main findings from this research is the absence of a comprehensive approach, where different discovery methods can be used together to reveal IT assets based on their intrinsic qualities. It becomes apparent that there is such an active/passive mix model, with advanced data analysis methods, that would yield an accurate and dependable network inventory.

Addressing these gaps, this paper specifically focuses on the issues identified in the literature and proposes solutions to bridge them. Through a novel agent-based lightweight design, we intend to augment the network discovery results substantially. Our approach not only

overcomes the drawbacks of conventional discovery tools but also facilitates the incorporation of sophisticated computational techniques within asset discovery procedures. Also, the scalability and adaptability of our proposed model would be highly effective, regardless of the different contexts or sizes of organizations. Lastly, we have put forward an integrated method that combines multiple discovery methods to make a complete solution, which is beyond the competence level of previous single-method ways. It results that research lacunae elucidated in the current scenario of IT asset detection could be considered a reasonable ground for justification for our study's input. Having eliminated these gaps, we aim at enhancing the level of IT asset discovery so that any organization could receive a considerably better solution in terms of its being more precise, faster, and scalable while managing its network assets.

### *Preliminary Studies*

Before applying or choosing any policy for network security, administrators are required to clearly understand the type of users that connect to the network and what kind of access is allowed to them. The process of network discovery can be explored in a systematic way to know the category of the output with the various methods. The different perspectives of network discovery are detailed below:

### *Type of Network Scanning*

Active Scanning: The liveness of assets can be monitored by using active probing or active network scanning. The probing can be simple ping requests to complex requests about the network. This type of scanning works with request and reply format. Active scanning is used to fetch the end-to-end results with appropriate statistical information. The information about the port number, service running, and system details are captured by creating a connection through request packets.

Passive Scanning: This type of scanning does not create additional network traffic to scan assets or services (Natu and Sethi, 2006). The traffic among different points is inspected by observing different packet headers. The traffic flow and information of some assets may not be guaranteed always; hence a complete picture of the network cannot be obtained by using passive monitoring. The tools and techniques sued for active and passive scanning of the network are listed in Table 1.

The different protocols used in the communication store different types of information in the packet headers. The process of network scanning targets to fetch these network protocol packets to understand the details of any network. active or passive scanning will aim to target these packet headers either with the request-reply packet or by silently inspecting the network. In Table 2, the type of information that can be fetched from the different protocols is listed.

### *Operating System Fingerprinting*

Vulnerability analysis of the network can only be performed with the exact asset identification. The National Vulnerability Database (NVD) (National Institute of Standards and Technology, 2022), maintains records of Common Vulnerabilities and Exposures (CVE) for each vulnerability. The vulnerability is associated with different Operating systems and applications. The detection of the operating system (Kaushik et al., 2022b) of the discovered devices in the network is necessary to have further details about the vulnerability of the network in any organization.

**Table 1:** Techniques/Tools Used for Active and Passive Network Scanning

| Type of Scan | Techniques/Tools |
|---|---|
| Active Scanning | ICMP (Wang and Zhang 2020), TCP Three-way handshake, Nmap (Lyon, 2008), Zmap (Durumeric et al., 2013) |
| Passive Scanning | P0f (Barnes and Crowley, 2013), Search Engines like Shodan (Genge and Enăchescu, 2016, Al-Alami et al., 2017), who-is information (Kuyama et al., 2016), Censys (Arnaert et al., 2016, Lee et al., 2017), Netcraft (Devi and Kumar, 2020), Netflow (Berthier et al., 2010) |

**Table 2:** Type of Information with Protocol/Service Packet Header

| Protocol/Service | Type of Information |
|---|---|
| Address Resolution Protocol (ARP) | Internet Protocol (IP) Address, Media Access Control Address (MAC address) |
| Dynamic Host Configuration Protocol (DHCP)/DHCPv6 | Host Name, Vendor, Domain Name |
| Multicast Domain Name System (mDNS) | Qname, Rrname |
| BROWSER | Source Name |
| User Datagram Protocol (UDP) | MAC Address |
| Simple Service Discovery Protocol (SSDP) | User Agent |
| Link-Local Multicast Name Resolution (LLMNR) | Qname |

The operating system can be detected using the fingerprinting method. The traditional method is to utilize packet fields of TCP packets and make a comparison with the operating system database (Matsunaka et al., 2013). The fingerprinting approach can be further divided into two categories active fingerprinting and passive fingerprinting.

Active Fingerprinting: Active fingerprinting works with reply and request packets among the device performing the task of asset identification and the connected host in the network. A three-way TCP handshake with acknowledgement can be used effectively for fetching the information embedded in packets of protocols Address Resolution Protocol (ARP), User Datagram Protocol (UDP), Internet Control Message Protocol (ICMP) and Transmission Control Protocol (TCP). The queries related to Domain Name System (DNS) can identify the Android Operating system (Lyu et al., 2022). Time to Live (TTL) is an important parameter to have accurate information about the Operating system. BROWSER protocol by Microsoft also stores information about the operating system (Mehdi and Starly, 2020).

Passive Fingerprinting: Passive fingerprinting works with tools without any interaction among hosts. Ettercap (Ornaghi and Valleri, 2020), is a tool for man-in-the-middle attack (Kaushik et al., 2022b) and it utilizes its database to detect operating systems in the network. The tool is used for the analysis of different network protocols and is capable of intercepting network traffic. P0f (Kali Linux Project Team, 2021) is the most advanced tool for passive fingerprinting using TCP packets. Passive fingerprinting takes time to execute, and the results obtained from these tools are dependent on their own datasets.

Figure 1 shows an example of communication among a web server, a machine deployed for asset identification and an HTTPS server to spoof messages for getting the device details of the webserver.

### *Service/Application Discovery*

Like operating system identification, the detection of running service or application on host devices connected to the network is an important criterion to analyse the vulnerability of the network. The information about open ports can get detail about the services used on the devices.

For example, port 22 indicates the working of the Secure SHell (SSH) service for a device in the network. Internet Protocol (IP) flow fingerprinting (Vermeer et al., 2021) can get information about the running applications. The information stored in the higher-level protocol is fetched to get the details about running the application and service.

The process of asset identification surrounds network scanning whether it is active or passive. The identification process includes getting details about IP addresses, MAC addresses, type of operating system, open ports, services running etc. The accuracy of the vulnerability analysis of the network is directly proportional to the accuracy of asset identification.

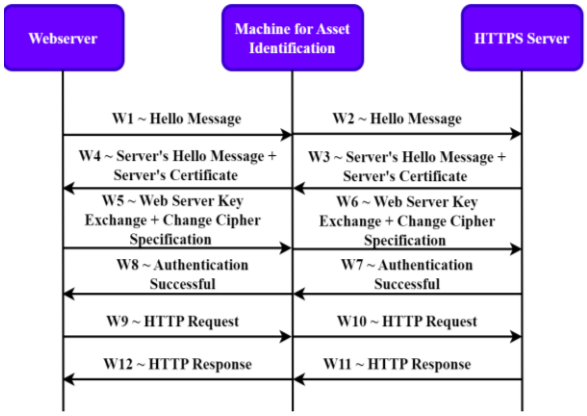


**Fig. 1:** Example of SSL Spoofing for Fingerprinting of Web Server

Active and passive scanning both have their corresponding pros and cons. The choice of type of scanning depends on the requirements of the organizations. Generally, passive scanning takes higher latency. The next section covers the latest research articles focused on IT asset identification for different types of networks.

### *Comparative Analysis of Methodologies for IT Asset Discovery*

The comparative analysis of the discussed research work on asset identification taken from the latest research articles has been shown in Table 3. The comparison is drawn based on the type of discovery, direct protocol used for asset discovery, and tools used. The research gap is also briefly mentioned in the table. Most large organizations work on different types of devices together, the network can be the setup of devices like Cameras, Printers, laptops, desktops, and Sensors. The approaches used for IoT Discovery (Feng et al., 2018; Jung et al., 2021) IP-based Core networks (Wang et al., 2022) and small LAN-based networks (He et al., 2017) can be used only for some special cases. The era of digitization is growing exponentially, and it demands to have optimized solutions to deal with the diversity of the network for asset identification (Wu and Wang, 2023).

Nmap tool is widely used in the literature as well as in organizations to get network discovery details. The tool gives the result with ambiguity, many researchers worked to provide the solution for enhanced results with Nmap. Chatzipoulidis et al. (2015) used the Nikto2 tool as well as Nmap to have reliable information about web servers specifically. Coffey et al. (2018) combined the results from Python UDP-DoS script, Nmap, Zmap and Tshark for SCADA systems. That clearly shows that no single approach is capable of fetching the complete picture of the network. The aggregation of the fetched data is again a challenging task concerning the reliability of the results in the asset identification process.

**Table 3:** Comparative Analysis of Methodologies for IT Asset Discovery

| Type of Discovery | Direct Protocol Used for Asset Discovery | Tools Used for Asset Discovery | Research Gap |
|---|---|---|---|
| Operating System, Web Server (Chatzipoulidis et al., 2015) | None | Nmap, Nikto2 | Only limited resources are considered in the network |
| Device Information (Coffey et al., 2018) | Python UDP-DoS script | Nmap, Zmap, Tshark | Python script can cause congestion in the network. No single network scanning tool can fetch the complete information of the assets. |
| Service Version, Operating System, Common Platform Enumeration (Martínez et al 2020) | None | Nmap | Undiscovered assets are required to be entered manually into the system |
| Operating System, Services (Wang, 2020) | Improved SNMP | None | SNMP services are to be configured through agents in the network. An internal subnet can be discovered only by using an ICMP packet. |
| IoT Device Discovery (Feng et al., 2018) | Application Layer Protocols | None | IoT devices connected to external networks are focused on research work, however majority of the devices are connected to a home network or firewall. |
| Device Type of IoT devices (Jung et al., 2021) | SSDP, NBNS, MDNS | None | The experimental setup consists of many devices connected to an internal network in the same domain. |
| Device Identification (Zhu et al., 2021) | DHCPv6, DHCP, BROWSER, IGMP, UDP, mDNS, LLMNR, NBNS and SSDP | None | WND Identifier achieves 85.2% accuracy in identifying the model details of different connected devices and there is a scope of research work to improve the identification process. |
| Network Scanning (Redondo and Cuesta, 2019) | Management website of Camera and Routers | NmapGUI | Many more features are required to be incorporated into the web application to cover all the functionalities of Nmap |
| Network Scanning (Zheng et al., 2021) | WebHunt | None | The accuracy of Seed and the rule provided by the user is a big challenge in the devised tool. |
| Automatic Discovery (He et al., 2017) | MDNS | None | The proposed method can only work for small networks in the absence of a DHCP server. |
| Network Scanning (Husák et al., 2021) | OS Fingerprinting | Netflow, Webchecker, WhatWeb | The data combined from multiple sources can be analysed only with Visualization support |
| Campus Network Scanning (Espinel Villalobos et al., 2021) | Multi-Agent SNMP | None | The method is fit to only get the count of users on the campus network. |
| Asset Identification in IP based core network (Wang et al., 2022) | Pretrained Multiple Encoders | None | The work is novel in the field of IP based core networks. But very few organizations are using entirely IP based core networks. |
| Operating System Identification (Salih et al., 2021) | TLS Handshake | None | An automated method is required to collect data for training the machine learning model. |

Including a Denial-of-Service attack in the network may affect the performance of the network. Martínez et al. (2020) used a manual method to enter undiscovered devices in the network with the output received from the Nmap tool. The same method cannot be utilized in the case of large networks and even small networks will not prefer to perform the manual entry using a network scanning tool. NmapGUI (Redondo and Cuesta, 2019) tool can provide an easy interface to the new users of Nmap in terms of script building but the limitation of not being able to discover all devices in the network remains the same.

SNMP is also widely used by network administrators (Chen and Zhou, 2023) to have brief information about the network. The entire information of the network cannot be retrieved with even improved SNMP (Wang, 2020) as well. Espinel et al. (2021) suggested a multi-agent-based network for device discovery in the network. Agent-based discovery is not feasible in many organizations, hence need for another improved approach arises. The objective of the research paper by Lastovicka et al. (2020) was to introduce the machine learning domain to handle encrypted data signals. The proposed method used in Zhu et al. (2021); Zheng et al. (2021) emphasizes using a combination of multiple approaches to have a real state of the network. The packet headers of different protocols store different types of information about the network. The collective information retrieved from these packets can give a reliable state of the network.

WebHunt tool experiments in the campus networks and the same tool can be tested with another type of organization as well (Miyajima et al., 2023). The accuracy associated with the provided seed and rule determines the authenticity of the network discovery (Ajiboye et al., 2023). Automation with seed and rule generation can make the system more error-free. The packet headers of multiple protocols fetched in Zhu et al. (2021) open a new area for different organizations to customize the asset identification process according to the needs of the organization. Alturkistani and El-Affendi (2022) explored the optimization of cybersecurity incident response using deep reinforcement learning. The use of reinforcement learning for dynamic decision-making could have complemented the proposed method by integrating predictive analytics. Ahmed and Khorsheed (2022) discussed the open network structure and smart network solutions for sharing cybersecurity within the 5G framework. This research work also supported the need for scalable and flexible network security solutions. Ahmed Abdirahman et al. (2024) focused on predicting vulnerability severity using natural language processing. Their approach to analyzing vulnerability descriptions provided a method for assessing potential threats. Basuki and Adriansyah (2023) addressed the optimization of response times in vulnerability management systems. Our research work improves the response time by fetching right information at right time. Widjajarto et al. (2021) examined vulnerability and risk assessment frameworks, specifically comparing VulnOS and Vulnix using the STRIDE model.

### *Proposed Light Weight Agent-Based Approach*

The recent work in the direction of asset discovery has been discussed in previous section and the comparative study of the studied literature shows that Nmap is majorly used for accomplishing the network discovery task. The proposed light weight agent-based approach targets optimizing the network discovery process by improvising remote device discovery results. In Figure 2 the suitable network architecture for the Campus Area network is presented with the proposed light weight agent-based approach with a dedicated network discovery server. Nmap tool can discover the active remote devices and provides logs of IP Addresses, MAC addresses, device operating system information, running services and open Ports. The vulnerability assessment is a crucial task while designing the security policies of an organisation. The vulnerability can be assessed accurately only after getting the complete details of peers connected to the organization network. The organizations like schools, colleges, health centres and universities connect to the campus area network and the different users of the campus area network use their own devices inside the organization. These users can work on multiple application software and these applications can be vulnerable to the entire organization. Cyber attackers target the vulnerable devices of organizations to get the access inside the network and execute malicious activity to steal the crucial data or to halt the working of the entire network by attacks like DDoS or Ransomware. The network administrator or security professionals of organizations need to take all preventive steps to prevent cyber threats in the organization. On the other hand, if any device is operating with pirated rights and using the device connecting to the campus area network, the application software company can put hefty fines to the organization. Hence, it is highly required to discover the assets connected to the campus area network accurately.

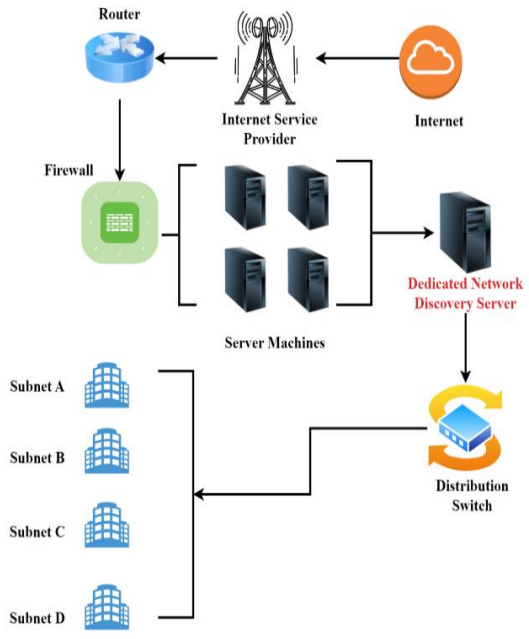


**Fig. 2:** Proposed Network Architecture for Optimizing Network Discovery

The proposed lightweight agent-based approach is also explained with pseudo-code in Algorithm 1. A dedicated server for network discovery is placed in between the network design of the campus area network. The proposed approach initiates with the scanning of the remote devices with the Nmap tool on the dedicated server. The Nmap results are stored in .xml format, these results are converted into .csv format using support from Python modules. The schedule of scanning the network can be decided by the network administrators or security officials of the organization considering the severity of the situation. The logs for IP address, MAC address, operating system, open ports and service are maintained in the .csv file after converting Nmap XML results into CSV format. The network is scanned for different subnets and these results can be aggregated using data visualization software.

Algorithm 1

Pseudocode Code - Light Weight Agent-Based Approach for Network Discovery

1. START → Routine IT Asset Discovery
2. START Nmap → nmap -T4 -A -v IP Range // Routine Route Discovery Process
3. CALL → XML_to_CSV() // Converting Nmap route discovery results to CSV format
4. FILTER → Window-based IP Address and Port Number
5. SUDO TERMINATE Connectivity for Filtered IP Addresses
6. CREATE SOCKET with web-based Login Page// Connectivity between Dedicated Server and Remote Peer device
7. RECEIVE (application_software.txt) → Dedicated Server // The details of application software with build information is fetched through remote peer to dedicated route discovery server
8. Map Nmap logs with socket connectivity results.
9. CALL → Data Preprocessing

After getting the details from Nmap network scanning, the dedicated network discovery initiates the socket connectivity with the peer devices in the form of a login page. In campus area networks, the devices can access the Internet only after successful login access. The proposed approach is limited to the window-based operating system only. The details of the Windows-based Operating system are fetched using the Nmap tool and the details are further used for expiring the login access of the specified IP Addresses. When the device user requests again for login, the socket connectivity is initiated with the dedicated network discovery server.

Socket communication utilizes the Windows Management Instrumentation (WMI) command-line utility for getting the details of all application software running in the remote device. The details of the application software with the exact build number are stored in the text file and transmitted to the dedicated network discovery server before closing the socket connection. The text file received at the dedicated network discovery server is converted to .csv format for further analysis. The socket connectivity at the dedicated network discovery server and the remote machine is explained with Algorithm 2 and Algorithm 3 respectively.

Algorithm 2

Server: Socket Creation

1. socket.socket(socket.AF_INET, socket.SOCK_STREAM) // Initialize TCP Socket
2. server.bind(ADDR) // Bind IP Address and Port Number
3. server.listen() // Set Server in Listening Mode
4. [NEW CONNECTION] {addr} // Update Status with Connection Request Approval
5. conn.recv(SIZE).decode(FORMAT) // Receive the File from Remote Client
6. file.close() //Close the file
7. conn.close() //Close Socket

Algorithm 3

Client : Socket Connectivity

1. Data = subprocess.check_output(['wmic', 'product', 'get', 'name']) // Write Data in file from wmic
2. socket.socket(socket.AF_INET, socket.SOCK_STREAM) // Initialize TCP Socket
3. client.connect(ADDR) // Connect to Server
4. client.send("detail.txt".encode(FORMAT)) //Send wmic data to Server
5. file.close() //Close the file
6. conn.close() //Close Socket

## Results and Discussion

The experimental study focused on implementing a lightweight agent-based approach for optimizing network discovery in the campus area network of the University. The key components of the network, including firewalls, computer servers, routers, and switches, were considered. The proposed approach was tested using a dedicated network discovery server installed before the distributed switch in the network architecture, as outlined in the proposed model. The study's foundation lies in the defined network parameters, as summarized in Table 4. The campus area network, consisting of five subnets with enabled inter-subnet communication, employs DHCP for IP address assignment and enforces a network login policy. A crucial aspect is the explicit mention and sharing of the usage of the agent-based approach for network discovery in the organization's IT policy. This ensures that the deployment aligns with organizational guidelines and standards.

To gather inputs for the proposed lightweight agent-based approach, the Nmap tool was employed for network discovery without relying on commercialized automated tools. The five subnets were subjected to a thorough network scan, which recorded important information such IP addresses, MAC addresses, services that were active, open ports, and operating system kinds. The basis for further investigation was the Nmap findings, which were kept in XML format for every subnet. Python was used to transform the XML results into CSV format for processing more quickly.

This process simplified the data and made it possible to combine the outcomes from many subnets. A more thorough examination was made possible by the detailed picture of the network's state that the generated dataset offered. The identification and distribution of operating systems inside the campus area network was a crucial component of the research. The frequency of different operating systems found using Nmap and the lightweight agent-based technique is graphically shown in Figure 3. Understanding the variety of the network's operational environment requires knowledge of this information. Its ability to delve further into the specifics of installed application software on devices running window-based operating systems was further expanded by the lightweight agent-based technology. By using socket connectivity, the method connected to certain distant devices and extracted useful data about the program environment. Ensuring that the agent-based strategy is in line with the organization's IT policy is a crucial aspect of its deployment.

Transparency and adherence to set principles are ensured by the policy framework's clear acknowledgment and dissemination of this technique. This proactive strategy supports the development of a secure and well-managed network environment through the implementation of a specialized network discovery server, as illustrated in Figure 4. The substance of the communication process is captured in this snapshot, which is essential for obtaining comprehensive details on installed application software on distant devices. The server machine starts the listening mode as soon as it creates the socket, preparing the system for possible client machine activity. By using this proactive approach, the server may reply to socket requests quickly, facilitating a smooth and effective channel of communication. Assuring that the server is prepared to interact with distant peers looking for information about application software, the listening mode is an essential first step. The server goes into listening mode and listens for requests from peers that are remotely located. These peers might be devices that are using the lightweight agent-based technique.

**Table 4:** Network Parameters

| Parameter | Values |
|---|---|
| Network Type | Campus Area Network |
| Number of Subnets | 5 |
| Inter-Subnet Communication | Enabled |
| IP Address Assignment | DHCP Server |
| Network Login Policy | Enabled |
| Usage of Agent-Based Approach for Network Discovery | Mentioned and Shared in Organization IT Policy |

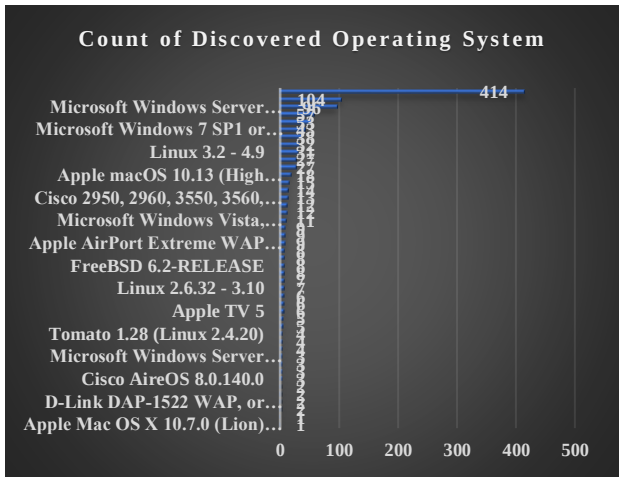


**Fig. 3:** Count of Discovered Operating System with Nmap Network Discovery Tool

```
In [1]: runfile('C:/Users/Ishu Sharma/Desktop/Cybersecurity/
Code/master3.py', wdir='C:/Users/Ishu Sharma/Desktop/
Cybersecurity/Code')
[STARTING] Server is starting.
[LISTENING] Server is listening.
[NEW CONNECTION] ('172.20.10.8', 3274) connected.
[RECV] Receiving the filename.
[RECV] Receiving the file data.
[DISCONNECTED] ('172.20.10.8', 3274) disconnected.
```

**Fig. 4:** Snapshot of Socket Creation on Dedicated Network Discovery Server

Upon receiving a socket request, the server examines it and grants permission to start a conversation. By establishing a secure communication channel, this approval method helps to prevent unwanted access and preserves the network's integrity. After establishing the socket connection and approving the request, the file transmission step starts.

This crucial stage enables data to be sent from the remote computer to the server, especially providing detailed information about all installed applications. In addition to the program names, the data payload contains version information, settings, and any other relevant

information that is considered necessary for a thorough comprehension of the software environment. The information regarding installed application software is carefully bundled to further improve the usefulness of the provided data. Information like the program's name, version number, installation date, and any particular parameters that affect how the application behaves are all included in this. Because of the data's granularity, network administrators are guaranteed to have a thorough understanding of the situation, which enables them to make well-informed choices on software management and security. Security is the most important factor to take into account at all times. Robust encryption algorithms are used by the lightweight agent-based approach to ensure the safe communication of sensitive application software data. By protecting the data from possible eavesdropping or manipulation, this makes sure that the data's integrity is preserved during transmission. The procedure that follows is designed to be scalable in order to function in a changing network environment. The lightweight agent-based technology scalable to easily enable simultaneous communication with several distant peers as the number of devices using it rises. For large-scale network infrastructures with a variety of devices coexisting and each adding to the total complexity of the network environment, this scalability is essential.

The final step of the lightweight agent-based method involves displaying the installed application software information, which is received in a text file, as illustrated in Figure 5. This comprehensive snapshot offers a thorough inventory of the software environment by representing a single distant computer. The procedure flows to many distant computers with ease, each of which adds to the overall knowledge of the software ecosystem on the network. The application software information that are received and are first saved in text files are transformed in order to improve their compatibility and integration with the network data that already exists.

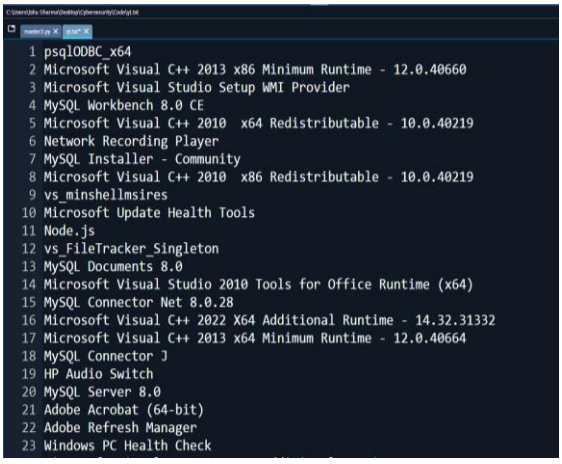


**Fig. 5:** Snapshot of Text File Received with Socket Communication at Dedicated Network Discovery Server

The text files are methodically transformed to the widely used and flexible.csv file format. This conversion enables smooth mapping to the outcomes of Nmap network scanning in addition to facilitating effective data handling and storage.

The mapped findings combine information from lightweight agent-based application software discovery and active scanning to offer a synchronized picture of the network. Network administrators may find possible security flaws, guarantee software compliance, and expedite software upgrades with the help of this strategic mapping. For example, by comparing the application software data obtained by the lightweight agent with the open ports found by Nmap, one might uncover security vulnerabilities related to certain software versions or configurations. Continuous network optimization is made possible by the lightweight agent-based approach's iterative nature and incorporation of Nmap data. Network administrators may monitor the efficacy of security measures, make targeted modifications, and adjust to changing network dynamics with the help of the synchronized data. This flexible strategy supports the ideas of ongoing development, resulting in a network architecture that is robust and sensitive to new threats.

Table 5 shows the comparative analysis of the existing OCS inventory tool and our proposed lightweight agent-based approach based on multiple parameters. Originally created for network discovery, the lightweight agent-based method may also be used to find endpoints running susceptible application software. This proactive strategy improves the security posture of the network by identifying possible areas of exploitation and enabling prompt correction. A further module for specialized vulnerability detection is incorporated, building upon the lightweight agent architecture already in place. To find software versions with known vulnerabilities, this module combines vulnerability databases and signature-based analysis. The agent's lightweight design guarantees low system resource consumption while strengthening the network's defenses against any security breaches. The expanded strategy includes ongoing device-wide monitoring of installed application software. Periodically, the lightweight agent checks for vulnerabilities by comparing the software versions it finds to a database of known flaws.

Network administrators are able to evaluate the risk level connected to every device by monitoring for instances of software that is susceptible and triggering alarms. A thorough understanding of the network's security environment is provided by the graph in Figures 6-7, which displays trends in the number of devices with vulnerable software found, devices scanned, and the number of high-risk vulnerabilities over the course of 15-time intervals.

Variations in the number of devices running susceptible software are consistent with the dynamics of the entire scanning process and provide information about how the network is changing.

**Table 5:** Comparison Analysis of Existing OCS inventory and proposed approach

| Tool | Agent Based | Pre-Installation Requirement | Installed Application Software | Support for BYOD |
|---|---|---|---|---|
| OCS Inventory | Yes | Required | Can be Fetched accurately by the agent installed on Client Machine | Not Suitable |
| Proposed Lightweight Agent-Based Approach | Yes, Web-based agent | Not Required | Can be Fetched accurately by Socket Communication | Suitable |

The "High-Risk Vulnerabilities" feature was added to identify vulnerabilities that represent a serious risk to the network. The growing and decrease patterns in high-risk vulnerabilities highlight how crucial it is to give remediation operations top priority in order to quickly resolve serious security issues.

The proposed method is further investigated against tow well established tools Nmap and OCS inventory NG. The experiments were conducted with 100 windows machine including physical and virtual machines. The firewall security is disabled with the administrative privileges on the machines. The complete configuration of the experimental set up is given in Table 6.

The investigation is based on five metrics: Accuracy, execution speed, performance overhead, network overhead, and detection accuracy. The measurement techniques used for each metric are outlined in Tables 7-8.

The proposed approach achieved accuracy of 96%, where Nmap's accuracy is 88% and OCS Inventory NG achieves 97%.

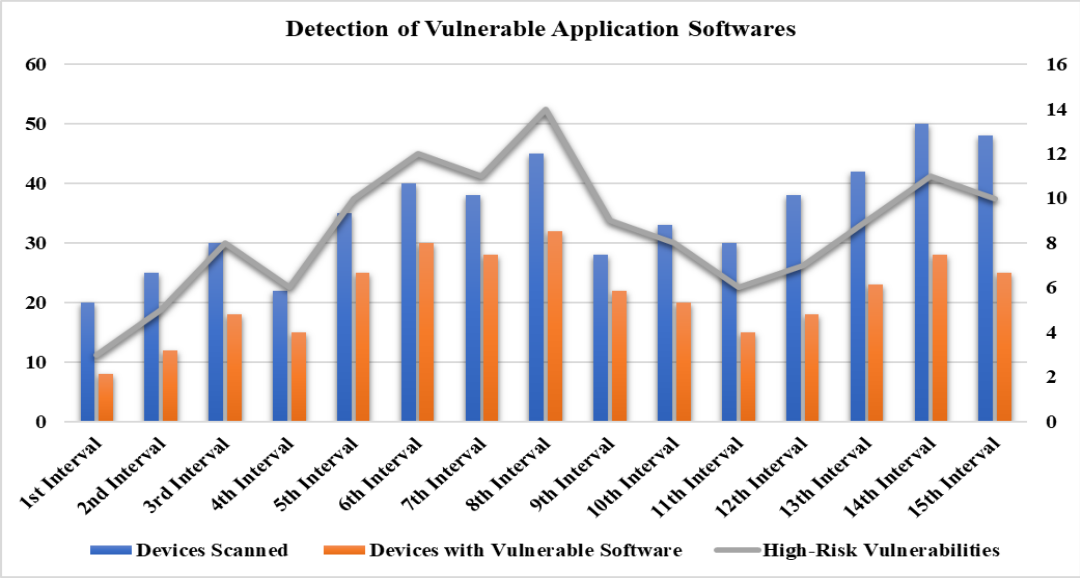


**Fig. 6:** Comparative Analysis of Detection of Vulnerable Application Software in Campus Area Network (15 Intervals)

**Table 6:** Experimental Setup

| Component | Details |
|---|---|
| Test Environment | Isolated network with 100 Windows 10 Pro machines (physical & virtual) |
| Server System | Windows 11 Pro |
| Client Systems | Windows 10 Pro (Physical + Virtual Machines) |
| Network Configuration | 1 Gbps Ethernet |
| Firewall Settings | Disabled |
| User Privileges | Administrative access |

**Table 7:** Details of Tools Configuration

| Tool | Description |
|---|---|
| Nmap | Version 7.94<br>command used: nmap -T4 -A -v 192.168.0.0/24 |
| OCS Inventory NG | Version 2.12, server setup (Apache + MySQL + OCS Reports); agents installed on clients |
| Proposed Approach | Python 3.10 + wmic + socket programming |
| File Transfers | Socket-based transfer of application_software.txt files |
| Monitoring Tools | Psutil, Windows Task Manager, Wireshark |

**Table 8:** Measurement Approach

| Metric | Measurement Approach |
|---|---|
| Accuracy (%) | Total discovered nodes / Total active devices × 100 |
| Speed (sec) | Time taken from discovery start to data collection completion |
| Performance Overhead | Client CPU usage tracked using psutil |
| Network Overhead | Average data transmitted/received per client through Wireshark |
| Detection Accuracy (%) | Matched software entries vs. installed list |

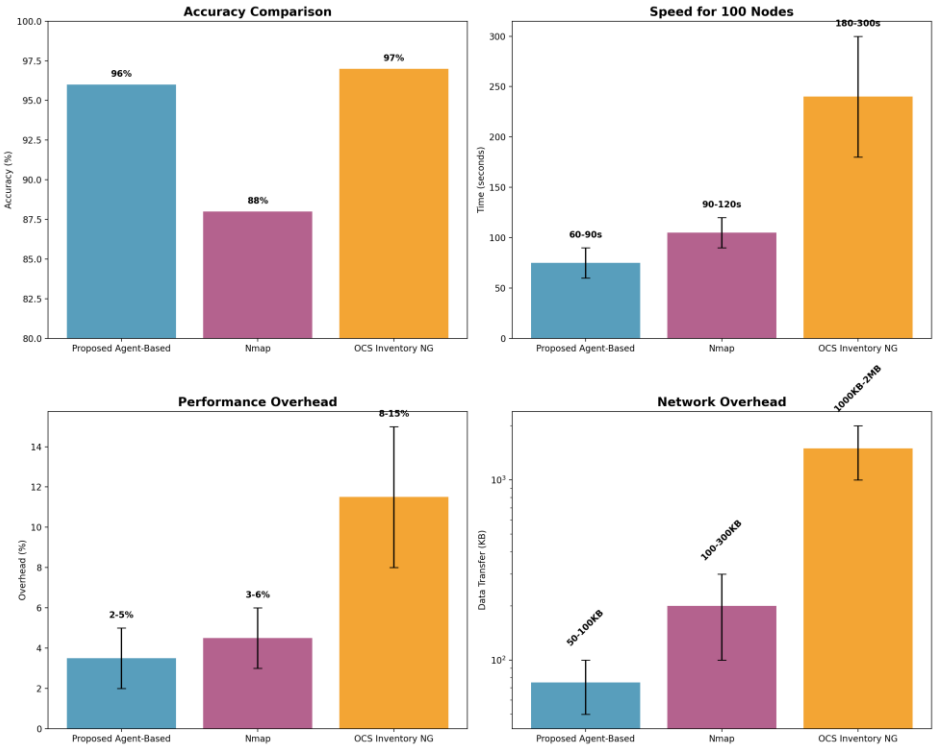


**Fig. 7:** Performance Evaluation of Proposed approach, Nmap and OCS Inventory NG

The detection accuracy for installed applications is also highly effective at 95–97%, aligning well with well-established tool OCS's performance. In terms of speed, the proposed approach taken 60–90 seconds for network discovery, which is faster than Nmap and OCS Inventory NG. This advantage is impactful due to its socket-based communication at the lighter stage in comparison with tools like OCS Inventory NG. Resource efficiency is also improved with the proposed approach. CPU usage remained between 2–5% during network discovery phase. This is lower than Nmap and OCS Inventory NG. Network overhead followed same pattern: The proposed approach required 50–100 KB of data transmission per node, while Nmap ranged between 100–300 KB, and OCS Inventory NG ranged 1–2 MB per node. The experimental results proves that the proposed method offers enhanced accuracy, speed, and efficiency in the process of network discovery.

The expanded lightweight agent-based method works well for classifying vulnerabilities according to risk categories and for locating devices with software vulnerabilities. This enables network managers to take a proactive approach, concentrating on minimizing the most serious security risks and lowering the network's overall risk profile. The scalability of the security measures is shown by the association between the number of devices examined and instances of susceptible software. The method adjusts as the network grows or shrinks, making sure that security precautions grow in step with the network's growth. Emerging high-risk vulnerabilities may be identified thanks to the expanded approach's continuous monitoring capabilities. The network is kept safe from changing security risks by the adaptive defensive system, which dynamically modifies detection settings.

The proposed light-weight agent-based approach for asset discovery is significant in maintaining the security of network devices and architecture. It is capable to provide more accurate and comprehensive information about the devices and the installed software of all the devices of the network. This enhanced visibility is vital for effective vulnerability assessment, enabling organizations working with campus area network to better

identify and address cybersecurity threats. Moreover, the proposed approach helps in ensuring compliance with software licensing agreements, which helps timely mitigation of legal risks associated with unauthorized software use in associated devices. The approach's scalability can be enhanced with the growth of the network without any additional cost that makes the model scalable and easy to deploy. The solution is applicable from small scale networks to the large-scale networks. The encryption methodologies for data protection for network devices enhances the confidentiality, integrity and availability of the campus area network.

## Conclusion and Future Scope

The heart of an organization's security strategy depends upon the accurate asset identification and classification. The complete knowledge of the network allows network administrators to keep updated record of network assets to implement a reliable security policy. The comparative analysis of existing methodologies presents the limitations of the various tools and applications in building right security architecture. The tools like Nmap are also ineffective for implementing independent result-oriented approach.

It has been noticed by researchers that while using Nmap information as a initial step, can generate leads for the next steps of the process of ethical hacking or implementing security policies. As far as network discovery initiatives are concerned, the utilization of a hybrid method substantially reinforces the efficiency of such actions. Comparing our study with others, it can be said that the use of a set of different techniques adapted for the architecture of the network and type of applications can provide maximum accuracy in asset identification and classification. Specifically, those methodologies that use multiple approaches for discovering assets give birth to new domains that open up more accurate ways to locate assets on networks depending on their environment. This idea is also being explored by some researchers through IP-based core networks, which might indicate yet another way of detecting assets in similar organizational setups as governmental bodies.

The proposed method can be integrated with artificial intelligence methodologies for boosting the entire security architecture from risk management to disaster recovery process. Adding compatibility for more operating systems, like Linux and macOS, would increase its adaptability and increase in the market for various organizations. The efficiency of the proposed approach can be increased by using data visualization and artificial intelligence methods. By sending out notifications and taking corrective action promptly, automated response systems can also accelerate the cybersecurity issues.

One of the most encouraging ways of increasing asset discovery's accuracy is to apply neural networks and progress further into more complex techniques that lie within the machine learning domain. The present research, therefore, demonstrated that asset discovery can be optimized by combining multiple methods like using scanning tools as well as active probing through protocols for data retrieval, which improves precision greatly. Our lightweight agent-based approach is considered to accurately distinguish remote devices across a campus network by using socket communication. For instance, the network discovery server that we have configured has been initiating regular scans to keep track of information which is later refined via client-server socket communication. This process guarantees the comprehensive retrieval of remote device details, thus giving security experts a clear picture while compiling effective vulnerability assessment reports.

Despite of this, the current solution is oriented mainly to Windows OS, so it demonstrates considerable space for further enlargement in the future towards covering all OS. Moreover, machine learning methods' involvement as a part of asset identification creation process can be an interesting direction for further exploration that gives great opportunities. For asset identification, this paper conducted a detailed comparative analysis with an emphasis on recent work on this topic and articulated areas that need attention by future researchers.

## Acknowledgment

This research was supported by the Faculty of Computing and Informatics, Multimedia University, Malaysia. The authors thank the university's network team for facilitating access to the experimental infrastructure. We are also grateful to the reviewers for their constructive comments, which have significantly improved the quality of this paper.

## Funding Information

This research was funded by Multimedia University.

## Author's Contributions

**Ishu Sharma:** Primarily responsible for the conceptualization of the study, development of the methodology, and drafting the original manuscript.

**Ahthasham Sajid:** Contributed to the software implementation, data validation, formal analysis, and data curation.

**Mazliham Mohd Suud**: Provided supervision throughout the project, oversaw project administration, and contributed to the review and editing of the manuscript.

**Muhammad Mansoor Alam:** Involved in acquiring resources, creating visualizations, contributing to the manuscript's review and editing, and securing funding for

the research.

All authors have read and approved the final version of the manuscript.

## Ethics

The authors declare that there are no ethical issues associated with this research. They remain committed to addressing any ethical concerns that may arise following the publication of this manuscript.

## References

Ahmed Abdirahman, A., Osman Hashi, A., Romo Rodriguez, O. E., & Abdirahman Elmi, M. (2024). Prediction of vulnerability severity using vulnerability description with natural language processing and deep learning. *International Journal of Electrical and Computer Engineering (IJECE)*, *14*(4), 4551–4562. https://doi.org/10.11591/ijece.v14i4.pp4551-4562

Ahmed, A. K., & Khorsheed, A. A. (2022). Open network structure and smart network to sharing cybersecurity within the 5G network. *Indonesian Journal of Electrical Engineering and Computer Science*, *27*(1), 573–582. https://doi.org/10.11591/ijeecs.v27.i1.pp573-582

Ajiboye, M., Ajiboye, O., Aleburu, D., Olayiwola, A., Olayiwola, D., & Ajose, S. (2023). Dimensionality Reduction for Deep Learning Based Intrusion Detection Systems for IoT. *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2023*, 76–81.

Al-Alami, H., Hadi, A., & Al-Bahadili, H. (2017). Vulnerability scanning of IoT devices in Jordan using Shodan. *Proceeding of the 2017 2nd International Conference on the Applications of Information Technology in Developing Renewable Energy Processes & Systems (IT-DREPS)*, 1–6. https://doi.org/10.1109/it-dreps.2017.8277814

Alturkistani, H., & El-Affendi, M. A. (2022). Optimizing cybersecurity incident response decisions using deep reinforcement learning. *International Journal of Electrical and Computer Engineering (IJECE)*, *12*(6), 6768–6776. https://doi.org/10.11591/ijece.v12i6.pp6768-6776

Arnaert, M., Bertrand, Y., & Boudaoud, K. (2016). Modeling Vulnerable Internet of Things on SHODAN and CENSYS: An Ontology for Cyber Security. *Proceedings of the Tenth International Conference on Emerging Security Information, Systems and Technologies*, 299–302.

Barnes, J., & Crowley, P. (2013). k-p0f: A high-throughput kernel passive OS fingerprinter. *Proceeding of the ACM/IEEE Symposium Architecture for Networking and Communications Systems, ANCS*, 113–114. https://doi.org/10.1109/ANCS.2013.6665187

Basuki, A., & Adriansyah, A. (2023). Response time optimization for vulnerability management system by combining the benchmarking and scenario planning models. *International Journal of Electrical and Computer Engineering (IJECE)*, *13*(1), 561–570. https://doi.org/10.11591/ijece.v13i1.pp561-570

Berthier, R., Cukier, M., Hiltunen, M., Kormann, D., Vesonder, G., & Sheleheda, D. (2010). Nfsight: netflow-based network awareness tool. *Proceedings of the 22nd USENIX Security Symposium*, 119–134.

Chatzipoulidis, A., Michalopoulos, D., & Mavridis, I. (2015). Information infrastructure risk prediction through platform vulnerability analysis. *Journal of Systems and Software*, *106*, 28–41. https://doi.org/10.1016/j.jss.2015.04.062

Chen, L., & Zhou, L. (2023). CrackSegConnect: a Crack Inpainting Network based on Segmentation Model. *Engineering Letters*, *31*(1), 255–261.

Coffey, K., Smith, R., Maglaras, L., & Janicke, H. (2018). Vulnerability Analysis of Network Scanning on SCADA Systems. *Security and Communication Networks*, *2018*, 1–21. https://doi.org/10.1155/2018/3794603

Devi, S. R., & Kumar, M. M. (2020). Testing for Security Weakness of Web Applications using Ethical Hacking. *Proceedings of the 2020 4th International Conference on Trends in Electronics and Informatics (ICOEI) (48184)*, 354–361. https://doi.org/10.1109/icoei48184.2020.9143018

Durumeric, Z., Wustrow, E., & Halderman, A. (2013). ZMap: Fast Internet-wide Scanning and Its Security Applications. *IEEE Pervasive Computing*, *12*(3), 605–619. https://doi.org/10.1109/mprv.2013.43

Espinel Villalobos, R. I., Ardila Triana, E., Zarate Ceballos, H., & Ortiz Triviño, J. E. (2021). Design and Implementation of Network Monitoring System for Campus Infrastructure Using Software Agents. *Ingeniería e Investigación*, *42*(1), e87564. https://doi.org/10.15446/ing.investig.v42n1.87564

Feng, X., Li, Q., Wang, H., & Sun, L. (2018). Acquisitional Rule-based Engine for Discovering {Internet-of-Things} Devices. *Proceedings of the 27th USENIX Security Symposium*, 327–341.

Genge, B., & Enăchescu, C. (2016). ShoVAT: Shodan-based vulnerability assessment tool for Internet-facing services. *Security and Communication Networks*, *9*(15), 2696–2714. https://doi.org/10.1002/sec.1262

Ghadi, Y. Y., Mazhar, Tehseen, Al Shloul, T., Shahzad, T., Ahmad Salaria, U., & Ahmed, A. (2024). Machine Learning Solutions for the Security of Wireless Sensor Networks: A Review. *IEEE Access*, *12*, 12699–12719. https://doi.org/10.1109/ACCESS.2024.3355312

He, J., Zhang, Y., & Yuan, X. (2017). MDNS based automatic discovery method in optical NMS. *Proceedings of the 2017 16th International Conference on Optical Communications and Networks*, 1–3. https://doi.org/10.1109/ICOCN.2017.8121329

Husák, M., Laštovička, M., & Tovarňák, D. (2021). System for Continuous Collection of Contextual Information for Network Security Management and Incident Handling. *Proceedings of the 16th International Conference on Availability, Reliability and Security*, 1–8. https://doi.org/10.1145/3465481.3470037

International Organization for Standardization. (2020). *Information security, cybersecurity and privacy protection — Information security management systems — Requirements*. https://doi.org/10.3403/30379032

Jung, H.-C., Jo, H., & Lee, H. (2021). UDP-Based Active Scan for IoT Security (UAIS). *KSII Transactions on Internet and Information Systems*, *15*(1), 20–34.

Kali Linux Project Team. (2021). p0f Kali Linux Tools. *Kali Linux Tools Documentation*. https://www.kali.org/tools/p0f/

Kaushik, K., Punhani, I., Sharma, S., & Martolia, M. (2022a). An Advanced Approach for performing Cyber Fraud using Banner Grabbing. *Proceeding of the 2022 5th International Conference on Contemporary Computing and Informatics (IC3I)*, 298–302. https://doi.org/10.1109/ic3i56241.2022.10072445

Kaushik, K., Singh, V., & Manikandan, V. P. (2022b). A Novel Approach for an Automated Advanced MITM Attack on IoT Networks. *Advancements in Interdisciplinary Research*, *1738*, 60–71. https://doi.org/10.1007/978-3-031-23724-9_6

Kaushik, K., Tanwar, R., & Awasthi, A. K. (2020). Security Tools. *Book Chapter in a CRC Press/Taylor & Francis Publication*, 181–188. https://doi.org/10.1201/9781003045854-13

Kuyama, M., Kakizaki, Y., & Sasaki, R. (2016). Method for Detecting a Malicious Domain by Using WHOIS and DNS Features. *Proceedings of the Third International Conference on Digital Security and Forensics*, 74–88.

Lastovicka, M., Spacek, S., Velan, P., & Celeda, P. (2020). Using TLS Fingerprints for OS Identification in Encrypted Traffic. *Proceedings of the NOMS 2020 – 2020 IEEE/IFIP Network Operations and Management Symposium*, 1–6. https://doi.org/10.1109/noms47738.2020.9110319

Lee, S., Shin, S.-H., & Roh, B. (2017). Abnormal Behavior-Based Detection of Shodan and Censys-Like Scanning. *Proceedings of the 2017 Ninth International Conference on Ubiquitous and Future Networks (ICUFN)*, 1–6. https://doi.org/10.1109/icufn.2017.7993960

Li, R., Shen, M., Yu, H., Li, C., Duan, P., & Zhu, L. (2020). A Survey on Cyberspace Search Engines. *Cyberspace Safety and Security*, 206–214. https://doi.org/10.1007/978-981-33-4922-3_15

Lyon, G. F. (2008). *Nmap Network Scanning: The Official Nmap Project Guide to Network Discovery and Security Scanning*.

Lyu, M., Gharakheili, H. H., Russell, C., & Sivaraman, V. (2022). Analyzing Enterprise DNS Traffic to Classify Assets and Track Cyber-Health. *ArXiv*, 15–20. https://doi.org/10.48550/arXiv.2201.07352

Ma, K., Sun, R., & Abraham, A. (2012). Toward a lightweight framework for monitoring public clouds. *Proceeding of the 2012 Fourth International Conference on Computational Aspects of Social Networks (CASoN)*, 71–76. https://doi.org/10.1109/CASoN.2012.6412429

Martínez, Á. L., & Vázquez González, V. A. (2020). A novel automatic discovery system of critical assets in cyberspace-oriented military missions. *Proceeding of the ACM International Conference Proceeding*, 1–8.

Matsunaka, T., Yamada, A., & Kubota, A. (2013). Passive OS Fingerprinting by DNS Traffic Analysis. *Proceeding of the 2013 IEEE 27th International Conference on Advanced Information Networking and Applications (AINA)*, 243–250. https://doi.org/10.1109/aina.2013.119

Mazhar, T., Talpur, D. B., Al Shloul, T., & Yasin Ghadi, Y. (2023a). Analysis of IoT Security Challenges and Its Solutions Using Artificial Intelligence. *Brain Sciences*, *13*(4), 683. https://doi.org/10.3390/brainsci13040683

Mazhar, T., Irfan, H. M., Khan, S., Haq, I., Ullah, I., Iqbal, M., & Hamam, H. (2023b). Analysis of Cyber Security Attacks and Its Solutions for the Smart grid Using Machine Learning and Blockchain Methods. *Future Internet*, *15*(2), 83. https://doi.org/10.3390/fi15020083

Mehdi, N., & Starly, B. (2020). Witness Box Protocol: Automatic machine identification and authentication in industry 4.0. *Computers in Industry*, *123*, 103340. https://doi.org/10.1016/j.compind.2020.103340

Miyajima, H., Shigei, N., Miyajima, H., & Shiratori, N. (2023). Neural Gas and k-means Methods with Reduced Communication Costs for Secure Distributed Processing. *Lecture Notes in Engineering and Computer Science: Proceedings of The International MultiConference of Engineers and Computer Scientists 2023*, 19–24.

National Institute of Standards and Technology. (2022). National Vulnerability Database. *U.S. Department of Commerce*. https://nvd.nist.gov/?utm_source=copilot.com

Natu, M., & Sethi, A. S. (2006). Active Probing Approach for Fault Localization in Computer Networks. *Proceeding of the 2006 4th IEEE/IFIP Workshop on End-to-End Monitoring Techniques and Services*, 25–33. https://doi.org/10.1109/e2emon.2006.1651276

Ornaghi, A., & Valleri, M. (2020). About the Ettercap Project. *Ettercap Project Official Website*. https://www.ettercap-project.org/about.html

Rahalkar, S. (2019). Introduction to NMAP. *Quick Start Guide to Penetration Testing*, 1–45. https://doi.org/10.1007/978-1-4842-4270-4_1

Redondo, J. M., & Cuesta, D. (2019). Towards improving productivity in NMAP security audits. *Journal of Web Engineering*, *18*(7), 539–578.

Salih, S., Hamdan, M., Abdelmaboud, A., Abdelaziz, A., Abdelsalam, S., Althobaiti, M. M., Cheikhrouhou, O., Hamam, H., & Alotaibi, F. (2021). Prioritising Organisational Factors Impacting Cloud ERP Adoption and the Critical Issues Related to Security, Usability, and Vendors: A Systematic Literature Review. *Sensors*, *21*(24), 8391. https://doi.org/10.3390/s21248391

Tsoukalos, M. (2015). Using tshark to watch and inspect network traffic. *Linux Journal, 2015*, 1–254.

Tundis, A., Mazurczyk, W., & Mühlhäuser, M. (2018). A review of network vulnerabilities scanning tools. *Proceedings of the 13th International Conference on Availability, Reliability and Security*, 1–10. https://doi.org/10.1145/3230833.3233287

Vazquez, A. (2019). NetBIOS and WINS. *Practical LPIC-3 300*, 499–514. https://doi.org/10.1007/978-1-4842-4473-9_19

Vermeer, M., West, J., Cuevas, A., Niu, S., Christin, N., Van Eeten, M., Fiebig, T., Gañán, C., & Moore, T. (2021). SoK: A Framework for Asset Discovery: Systematizing Advances in Network Measurements for Protecting Organizations. *Proceeding of the 2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, 440–456. https://doi.org/10.1109/eurosp51992.2021.00037

Wang, H. (2020). Improvement and implementation of Wireless Network Topology System based on SNMP protocol for router equipment. *Computer Communications*, *151*, 10–18. https://doi.org/10.1016/j.comcom.2019.12.038

Wang, Y., & Zhang, Q. (2020). Brief Introduction of Network Security Asset Management for Banks. *Cyber Security*, *1299*, 215–221. https://doi.org/10.1007/978-981-33-4922-3_16

Wang, Y., Chang, P., Wang, H., Ding, Y., & Sun, R. (2022). MAE-CAD: An IP-Based Core Network Asset Discovery Technology Based on Multiple Autoencoders. *Security and Communication Networks*, *2022*, 1–13. https://doi.org/10.1155/2022/6854344

Widjajarto, A., Lubis, M., & Ayuningtyas, V. (2021). Vulnerability and risk assessment for operating system (OS) with framework STRIDE: comparison between VulnOS and Vulnix. *Indonesian Journal of Electrical Engineering and Computer Science*, *23*(3), 1643–1653. https://doi.org/10.11591/ijeecs.v23.i3.pp1643-1653

Wu, J., & Wang, J. (2023). An Energy-Efficient Embedded System Platform for Energy-Critical Real-Time Tasks. *Engineering Letters*, *31*(1), 105–112.

Zheng, R., Ma, H., Wang, Q., Fu, J., & Jiang, Z. (2021). Assessing the Security of Campus Networks: The Case of Seven Universities. *Sensors*, *21*(1), 306. https://doi.org/10.3390/s21010306

Zhu, F., Liu, L., Hu, S., Lv, T., & Ye, R. (2021). WND-Identifier: Automated and Efficient Identification of Wireless Network Devices. *Security and Communication Networks*, *2021*, 1–16. https://doi.org/10.1155/2021/9069123