National College of Ireland

AN INDEPTH EXPLORATION OF NETWORK FIREWALL PERFORMANCE, MONITORING EFFICACY AND SECURITY

MSc Research Project

Cybersecurity

Karthic Krishna Srekant
Student ID: x22165291

School of Computing

National College of Ireland

Supervisor: Imran Khan

National College of Ireland

MSc Project Submission Sheet

School of Computing

Student Name:	Karthic Krishna Srekant		
Student ID:	x22165291		
Programme:	MSc Cybersecurity	**Year:**	2023 - 2024
Module:	MSc Research Project		
Supervisor:	Imran Khan		
Submission Due Date:	30/1/2024		
Project Title:	**AN INDEPTH EXPLORATION OF NETWORK FIREWALL PERFORMANCE, MONITORING EFFICACY AND SECURITY** **Page Count 26 with cover page**		
Word Count:	8334 without cover page		

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

Signature:	Karthic krishna Srekant
Date:	30/01/2024

PLEASE READ THE FOLLOWING INSTRUCTIONS AND CHECKLIST

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

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

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

AN INDEPTH EXPLORATION OF NETWORK FIREWALL PERFORMANCE, MONITORING EFFICACY AND SECURITY

Karthic Srekant
X22165291

Abstract

This report contains the details of exploration of network firewall performance using wireshark and the snort. The report provide the details of various analysis of many literatures which helps to know about the analysis of different sector of firewall, also this report provide the analysis of wireshark and snort with proper result, Through the analysis the result provide various firewall bypass and breach, this helps to know that how they breach where the attack coming from and where these packets go to, all this things which shows in logs result.

1 Introduction

In the ever-evolving world of information technology, robust network security measures are now important to corporate sustainability. The environment that is favourable to possible cybersecurity concerns is created by the sheer volume and complexity of data exchanged within and across networks, as well as the adoption of the digital era by organizations and institutions. Network security is crucial in this dynamic environment, necessitating the deployment of strong defences to keep sensitive information, valuable assets, and critical systems safe from illegal entry and malicious use. In this context, the firewalls refer to the protection system that performs as an essential guard, properly analysing as well as watching digital boundary networks. Along with this, firewalls are also important for properly controlling and monitoring data flow among internal and external network as it also ensures valid communication that pass through all these virtual getaways. These types of security precautions are also significant for eliminating the risk related to the broad cyber threats, and malware infiltration for any kind of data breach that focuses on preventing unwanted access to the data. Furthermore, firewalls also assist in monitoring the data packets as they help in moving across the whole network and acting in related with a predetermined set of rules and policies to improve the security purpose.

However, a huge range of packet characteristics that consider various sources and proper destination addresses, protocols that are being used and port numbers are also analysed and accessed with the use of this kind of process (Kaspersky, 2023). In addition to this, firewalls also assist in protecting the network from any kind of potential threats by analysing all these rules and at the same time developing a virtual barrier that can selectively block data flow. Mainly, there are various options available within the overall world of firewalls in each and every design in order to meet specific security needs. Stateful review of firewalls also analyses every connection within the greater detailed information that also keeps track of the status of all the active connections. It also assists in making choices over the fly as compared to traditional packet filtering firewalls that directly contribute to examining the individual information packets as per the predetermined parameters. Along with this, by effectively analysing the application level protocols, application layers also offer huge precision towards the types of services that give permission to interact over the firewalls. From all these types of firewall choices, network firewalls are also included as an important part that is used for ensuring the security of a business network infrastructure.

By keeping in mind all the incoming as well as outgoing traffic, all these types of cautions also guardians develop an issue against potential threats that would originate from both outside and in the network as well. By proper understanding of all the intricacies and proper analysis of network firewalls is more significant for developing as well as implementing enough security carriages and this type of understanding is significantly increasing the significance as creativities contend within the continuously changing landscape of cyber threats. Thus, type investigation also focuses on exploring the network firewall security, performance and proper efficacy of monitoring (GeeksforGeeks, 2021). This research is based on resolving the issues and complex nature of the network firewalls with the use of exploring the different forms of effective operational processes and also changing the need for network security. By effectively analysing all the critical aspects, the study aims to provide proper insights that will assist the network security framework. optimisation and also reinforce the organisation's defences against the constant and ever-changing nature of cyber issues within the current digital scenario.

Rationale

The current investigation focuses on the network firewall performance, monitoring efficacy and security that needs to be undertaken within the current scenario of digital interconnection and the increasing threats that arise from cyber attackers. Additionally, network security must be resilient and effective when information technology provides the foundation of all organisational activities. As a researcher, a number of things went into choosing this topic. Firstly, the security of networks is a persistent and significant global concern, offering a wide field for investigation and assessment. It is vital to have a compacted understanding of the structures that strengthen digital perimeters due to the pervasiveness of cyber threats, which makes this a valuable and motivating area of study. The field of safety for networks is dynamic due to the ever-evolving threat landscape and continuous advancements in technology. The researcher's choice is also influenced by how beneficial the network firewall study is. As the globe grows more interconnected, businesses across a range of industries are finding it difficult to fortify their networks against sophisticated cyberattacks. The issue aligns with the researcher's professional and academic objectives and offers a way to effective skills within the network analysis, security of data, and research on cybersecurity. The complexities of network firewalls enable the acquisition of practical knowledge in cybersecurity methods, testing processes, and the thorough analysis of security postures.

Aim

Through a thorough investigation of network firewalls performance, monitoring effectiveness, and security, the main goal of this research is to further enhance our comprehension of these crucial elements in contemporary cybersecurity. The research intends to contrast the effectiveness of these firewall varieties in terms of productivity, latency, and resource usage in addition to evaluating how well they detect and mitigate cyber threats. The study will evaluate and analyze the security postures of multiple firewall architectures to determine the best setup and deployment strategies. In this context, the, research also look at the research's practical implications and provide useful advice for businesses trying to strengthen their cybersecurity defenses.

Research questions

- What are the effectiveness traits of various network firewall structures, and how do their efficacy and efficiency stack up against one another ?
- What are the differences in throughput, latency, and resource usage between various kinds of network firewalls?
- What aspects affect a network firewall's performance, and how can an organization optimize its firewall configurations to improve network performance as a whole?
- How good are firewall monitoring systems in identifying and thwarting various types of online threats?
- How well-suited are various firewall types to recognize and address unusual network activity?
- How can companies improve their overall network security postures by putting these best practices into practice?

2 Related Work

A firewall is a secure Internet gateway that connects a private network to the Internet, states Chadwick (2023). A firewall is composed of several different parts. One example of this is the organization's Internet access security policy. This defines the general level of security that the organization intends when using the Internet. It states that outside users are allowed to join to the company network following a robust authentication procedure. Also, it is required that any corporate information that is not publicly accessible be transmitted via the Internet in a safe and private manner. Also, the policy forbids business customers from using any other services and limits them to sending just electronic mail over the Internet. Technical designs and processes are built to execute and enforce these principles; however, they are susceptible to revisions in response to new technology and system configurations. A firewall system, which consists of hardware and software components such an IP packet filtration router and a host computer running authentication and application filtering software to support overall security measures, makes it easier to implement these regulations and designs. As per Ali Bin Hamid Ali (2011), an essential part of network security is a firewall, which can be acquired as software, hardware, or a mix of the two. Its primary use is in network security, namely in monitoring and controlling data traffic as it moves via a connected router. In order to prevent unauthorised users from entering a Local Area Network (LAN) or other potentially dangerous external sources, such as the Internet, a firewall is installed. Packet filtering is the process of analysing data packets one by one and deciding whether to accept or reject them based on predetermined criteria. Decisions are based on traffic behaviour and history in a stateful analysis, which takes into account the context and condition of current connections. Another firewall method that Ali Bin Hamid Ali (2011) discussed was the use of proxy services, which acted as an intermediate system between users' internal systems and external networks, adding an additional degree of security.

Operating at the application level, application-layer filtering searches for certain patterns or behaviours associated with known data hazards. Fulp and Tarsa (2023) state that firewalls, which are also called packet filters, are essential parts of network security systems that protect against emerging threats. Importantly, these firewalls apply a security policy-based approach to access control, auditing, and traffic management. This policy is simply an organised collection of recommendations that govern what needs to happen when a packet arrives or departs. Following the first-match policy, which is used by several firewall systems like the iptables Linux version, a packet is progressively matched to these rules as it arrives at the firewall until a match is discovered. The firewall conducts the corresponding action once a match is detected, determining whether to allow or deny the packet in line with the standards that have been specified. A higher level of network protection is provided by this logical operation, which checks that network traffic follows the defined security requirements.

As stated by Fulp (2006), a scalable design that integrates several firewalls to increase efficiency and satisfy the needs of expanding network traffic is represented by a function-parallel network firewall. This strategy splits the original security policy's rules between the array of firewalls, placing each firewall in responsibility of carrying out a specific component of the policy. Compared to earlier parallel systems, this method tries to reduce delays and increase performance. The entire processing time is considerably decreased by dividing the workload across numerous firewalls, guaranteeing speedy and effective packet handling throughout the network. This is particularly relevant in circumstances where low latency and high throughput are critical characteristics. A key problem is the necessity for effective communication between the multiple firewalls in the array. In order to ensure the dispersed rules' synchronization and coordination and to allow the complete and logical execution of the security policy as a whole, this communication is essential.

The installation method may be exacerbated by the intricacy of firewall intercommunication, which might create extra delays. The function-parallel network firewall architecture is still an attractive alternative for enterprises wanting to achieve a balance between performance and scalability in the face of increasing cybersecurity threats and expanding network needs, despite these obstacles. As stated in Markham and Payne (2001), network Edge Security is a breakthrough approach of network security that delivers a new distributed firewall architecture intended particularly to solve internal threat challenges and keep up with growing technology trends. Compared to standard firewall installations, Network Edge Security methodically disperses security controls along the network edge, delivering a more extensive and detailed barrier against probable attacks originating from within the firm. This innovative design is highly flexible to the changing demands of current technological breakthroughs, such as virtual private networks

(VPNs), mobile computing, and business-to-business computing. as described by Yue et al. (2009) In order to eliminate insider threats and stay up with the newest technology developments, such as virtual private networks (VPNs), mobile computing, and business-to-company computing, Network Edge Security is an advanced decentralised firewall design.

This unique technology, which varies from standard firewalls, strategically distributes security measures at the network edge to offer a more complicated and thorough protection against internal threats. This design delivers enhanced visibility into network behaviour by spreading security controls to the network edge, permitting proactive threat detection and prevention. The campus network follows a hierarchical structure, which is a regularly utilised approach in both campus and corporate networks, according to the research done by Bin Ali et al. (2013). The modular topology of this design gives building blocks for straightforward network development. With 950 million individuals accessing the internet globally, there are over 225 security breach incidences per day, which underscores the critical need for effective security measures. Campus networks are constructed in a hierarchical form to promote predictable traffic patterns and make measurement, understanding, and troubleshooting simpler. Scalability and stability are enhanced by this strategy, which increases the functionality and performance of devices dependent to where they are in the network hierarchy. Because of the design's flexibility, adding or removing modules may be done fast and simply without needing a major redesign or revamp of the network. This flexibility supports efficient network administration, fault isolation, and troubleshooting—all critical components of a secure and strong campus network design.

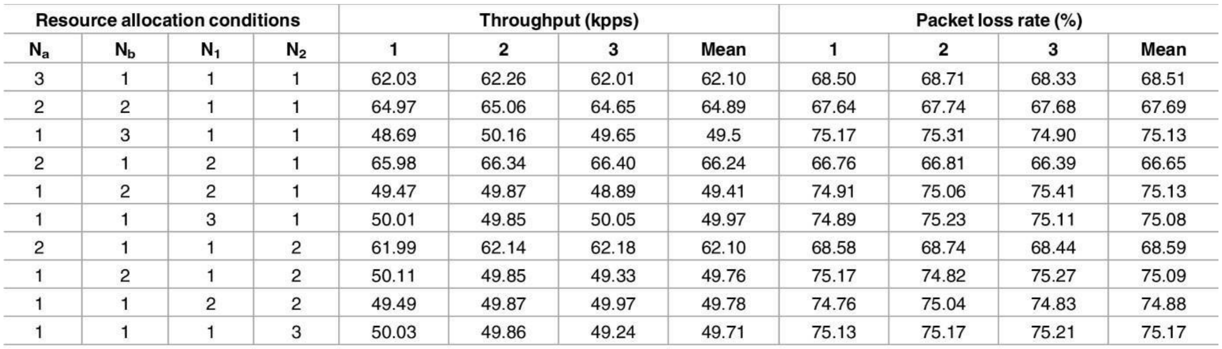
As stated by Maheshwari and Dagale (2018) by linking distant smart devices and relying largely on sensing, communication, and real-time data processing with scarcely any human interaction, the Internet of Things (IoT) is changing global connection. However, due of their intrinsic limitations—limited memory, low processing power, and dependence on battery-powered operations—and their accessibility to the internet, IoT devices are susceptible to a number of assaults, such as floods, Man-in-the-Middle (MIM), Denial of Service (DOS), and Sybil.. Different techniques are essential in light of these issues in order to ensure meaningful security for IoT applications and devices. In this research, we present an innovative firewall and secure communication architecture tailored for Internet of Things applications. With our solution, the computational strain from IoT devices is offloaded by introducing a dedicated server entity to the network. This inventive design offers a solid security architecture ideal for a range of applications within the IoT ecosystem, solving the special issues provided by the Internet of Things.

2.1 Network firewall performance

Firewalls have developed from their conventional role as perimeter devices for data centres to become crucial components woven throughout the whole network fabric, stretching from edge to edge, according to Sheth & Thakker (2011). The purpose of this strategic deployment is to offer pervasive and layered protection. Modern firewalls must combine dynamic policy-based security in addition to static defences to enable application intelligence, quick scalability, high availability, and optimal performance. This modification emphasizes a comprehensive approach that satisfies a range of security needs throughout the network, representing the next stage in the development of firewall technology. According to (Sheth & Thakker, 2011), the relevance of network firewall design quality has been reinforced by contemporary regulatory frameworks such the Sarbanes-Oxley Act, CobiT, the Payment Card Industry Data Security Standard (PCI DSS), and the NIST standard.

These laws' compliance requirements place a special emphasis on firewall configuration, management, and audit procedures. Recent assessments have shown that Cisco ASA distinguishes itself in this environment by outperforming its competitors across a range of performance metrics. Notably, OpenBSD PF and Checkpoint SPLAT both perform admirably and competitively, meeting the changing needs for reliable and legally compliant network firewall solutions. based on (Lyu & Lau, 2001), when they study firewall security and how it relates to distributed system performance, experiments are run to evaluate the effects of seven different security layers. These security tiers are systematically developed, created, put into practice, and evaluated in separate phases in an experimental setting.. The experimental results challenge the widely held assumption that increased security will always end in decreased performance. The study shows that there isn't always a positive correlation between security and performance in firewall testing. Instead, enhanced security has a minimal effect on performance under certain conditions.

The results of the study highlight the complex nature of the link between security and performance in the context of firewall setups within distributed systems, proposing that the intuitive notion that security and performance must be traded off does not always hold true. According to the test results, no firewall showed continuous tolerance against Distributed Denial of Service (DDoS) attacks done by Sheth & Thakker, (2013). In comparison with Cisco and PF, Checkpoint originally showed resistance and permitted valid traffic at a larger percentage of DDoS. Still, Checkpoint's CPU usage was greater than Cisco ASA and PF firewalls'. It took only a short while for all three firewalls to become inaccessible, meaning that there was little time to properly respond to DDoS attacks. The experiments showed that during a DDoS attack, each set of packets used state-table resources, causing a bottleneck, even though the firewalls were stateful. As a result, under intense DDoS attacks, all three firewalls' state-table resources quickly ran out, making them inaccessible. Sheth and Thakker (2013) carried out more tests to improve firewall performance in response to these difficulties by modifying the TCP Opening timer during SYN Flood attacks. In order to regulate state table entries and lessen the effect of DDoS attacks, a number of settings were changed. The aim was to enhance the firewall's capacity to manage and fend off DDoS attacks by incorporating intelligence into its configuration.



Resource allocation conditions				Throughput (kpps)				Packet loss rate (%)			
N_a	N_b	N_1	N_2	1	2	3	Mean	1	2	3	Mean
3	1	1	1	62.03	62.26	62.01	62.10	68.50	68.71	68.33	68.51
2	2	1	1	64.97	65.06	64.65	64.89	67.64	67.74	67.68	67.69
1	3	1	1	48.69	50.16	49.65	49.5	75.17	75.31	74.90	75.13
2	1	2	1	65.98	66.34	66.40	66.24	66.76	66.81	66.39	66.65
1	2	2	1	49.47	49.87	48.89	49.41	74.91	75.06	75.41	75.13
1	1	3	1	50.01	49.85	50.05	49.97	74.89	75.23	75.11	75.08
2	1	1	2	61.99	62.14	62.18	62.10	68.58	68.74	68.44	68.59
1	2	1	2	50.11	49.85	49.33	49.76	75.17	74.82	75.27	75.09
1	1	2	2	49.49	49.87	49.97	49.78	74.76	75.04	74.83	74.88
1	1	1	3	50.03	49.86	49.24	49.71	75.13	75.17	75.21	75.17

doi:10.1371/journal.pone.0167280.t001

Figure 1Perfromace evaluation model for application layer firewall

Firewalls have developed from their conventional role as perimeter devices for data centres to become crucial components woven throughout the whole network fabric, stretching from edge to edge, according to Sheth & Thakker (2011). The purpose of this strategic deployment is to offer pervasive and layered protection. Modern firewalls must combine dynamic policy-based security in addition to static defences to enable application intelligence, quick scalability, high availability, and optimal performance. This modification emphasizes a comprehensive approach that satisfies a range of security needs throughout the network, representing the next stage in the development of firewall technology. According to (Sheth & Thakker, 2011), the relevance of network firewall design quality has been reinforced by contemporary regulatory frameworks such the Sarbanes-Oxley Act, CobiT, the Payment Card Industry Data Security Standard (PCI DSS), and the NIST standard.

2.2 Monitoring and security efficiency

The research of Lee et al., (2005) asserts that four concurrent views—real-time traffic, visual signature, statistics, and IDS alarm—are offered by VisualFirewall to handle firewall setup and network monitoring difficulties. These views, which are particularly developed for system managers, provide varied time-scales and degrees of information for both passive and active monitoring. Even non-experts may readily discern between benign and malicious traffic patterns thanks of the tool's user-friendly UI. Visual Firewall provides administrators the opportunity to observe how firewalls react to packets, allowing them to setup and monitor network security with confidence. As to Grammatikakis et al. (2014), it is obvious that there is significant interest in trusted computing in the embedded arena, particularly when it comes to Multiprocessor System-on-Chip (MPSoC) architectures. Every processing unit in these systems, like a CPU, has the capacity to seek access to physical resources like memory or I/O units.

The study to far has concentrated on safeguarding MPSoCs, there is a distinct hole in the literature when it comes to source protection as opposed to target protection. Malicious material might be easily detected and destroyed at the network interface to stop it before it ever reaches to the on-chip network. By cutting any security concerns at their source, this proactive technique strives to strengthen MPSoCs' overall security posture. Moreover, network performance may improve from the implementation of a security mechanism at the network interface. This technology saves power waste associated with superfluous transmissions and shortens the packet delivery time from start to finish by keeping harmful material out of the on-chip network. As a consequence of the rising usage of the Internet and information systems as well as the broad supply of services via web applications, hazards and threats associated to these advances have expanded considerably, according to Ghanbari et al. (2015).

Particularly, in the last few years, a number of websites—including governmental websites, online businesses, and portals—have been the focus of illicit hacking efforts and penetration assaults. In addition to incurring enormous financial losses, these invasions frequently damage the image of enterprises and, in certain instances, national interests. This accessibility is advantageous, particularly given the worldwide nature of the Internet and the demand that websites be available around-the-clock internationally. But it also creates flaws, exposing up internet programmes to assaults and targeting. In response to these challenges, web application-specific firewalls, or Web Application Firewalls (WAFs), have arisen as an important and relatively new breakthrough in the area of cybersecurity. These technologies serve as a preventive step against internet invasions and attacks. Web application firewalls (WAFs) are vital for defending against a range of cyberthreats because they allow the execution of security restrictions between Internet applications and end users.

According to Shu et al., (2011), classical firewalls function by filtering known attack types and blocking packets based on IP addresses or ports that have been allocated. This strategy, however, only offers static and limited protection. The article proposes a security architecture based on Secure Router, a double-homed host that operates as a firewall with the capacity to dynamically update its rule set. This modification takes happen in answer to alarms received from credible intrusion detection systems within the protected local area network. The consistent interface of this framework with algorithm-independent intrusion detection modules is another essential characteristic noted by Shu et al., (2011). By boosting flexibility and extensibility, this design approach helps the security system to better react to new threats. A packet recording mechanism that enables offline network traffic analysis simpler is also incorporated in the framework. This logging capability helps to uncover and grasp probable security vulnerabilities by enabling a more detailed examination of the network's activities. Eventually, the purpose of the Secure Router-based method that is being presented is to deliver a security solution that is more responsive and dynamic than traditional firewalls.

Author	**Title**	**Methodology**	**Finding**
Ali Bin Hamid Ali, F. (2011)	'A Study of Technology in firewall system', *2011 IEEE Symposium on Business, Engineering and Industrial Applications (ISBEIA)* [Preprint].	proxy-based firewalls, packet filtering	Whether it be software or hardware, firewalls protect networks by controlling traffic and blocking illegal access using a variety of techniques. Security policies are impacted by choices made about firewalls.

Fulp, E.W. (2006)	‘An independent function-parallel firewall architecture for high-speed networks (short paper)’, *Information and Communications Security*, pp. 292–301.	firewall intercommunication	Outperforming existing systems, the autonomous, accept-set distributed function-parallel firewall matrix provides speed, integrity preservation, and service differentiation.
Fulp, E.W. and Tarsa, S.J. (2023)	*Network Firewall Policy Tries* [Preprint].	parallel network firewall architecture	An independent, accept-set distributed function-parallel firewall array outperforms existing systems in terms of speed, integrity preservation, and service differentiation.
Markham, T. and Payne, C. (2001)	‘Security at the Network Edge: A distributed firewall architecture’, *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX’01* [Preprint].	Network Edge Security	An independent, accept-set distributed function-parallel firewall array outperforms existing systems in terms of speed, integrity preservation, and service differentiation.
Bin Ali, M.N., Rahman, M.L. and Hossain, S.A. (2013)	‘Network Architecture and security issues in campus networks’, *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* [Preprint].	Network Architecture Design	The suggested campus network design places an emphasis on security while also addressing the various demands of the institution and guaranteeing flexibility and resistance to future cyberattacks.

Maheshwari, N. and Dagale, H. (2018)	‘Secure communication and firewall architecture for IOT Applications’, *2018 10th International Conference on Communication Systems &amp; Networks (COMSNETS)* [Preprint].	Novel Secure Communication and Firewall Architecture	By outsourcing computing to a server, the proposed IoT security architecture strengthens device defense against weaknesses. A comparison of the two approaches using DTLS is provided.
Sheth, C. and Thakker, R. (2013)	‘Performance evaluation and comparison of network firewalls under ddos attack’, *International Journal of Computer Network and Information Security*, 5(12), pp. 60–67.	FTP (File Transfer Protocol) servers	Hackers can hack FTP servers on port 21. Solution: For security, use firewalls, limit client access, and use port knocking authentication.
Lyu, M.R. and Lau, L.K.Y. (2001)	‘Firewall security: Policies, testing and performance evaluation’, *Proceedings 24th Annual International Computer Software and Applications Conference. COMPSAC2000* [Preprint].	Performance Metrics	The effect of firewall security on performance varies; performance is not always negatively impacted by more security. There is a trade-off; a connection isn't necessarily inverted.
Lee, C.P. *et al.*	‘Visual firewall:	firewall security	Performance is not

(2005)	Real-time Network Security Monitor’, *IEEE Workshop on Visualization for Computer Security, 2005.*	levels, conducting experiments, collecting data	always negatively impacted by firewall security; improved security doesn't always have the same effect. Performance and security trade-offs are scenario-dependent.
Grammatikakis, M.D. *et al.* (2014)	‘Security Effectiveness and a hardware firewall for mpsocs’, *2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS)* [Preprint].	Multi-Processor System-on-Chip (MPSoC).	A hardware NoC firewall that is implemented at the segment level and has deny rules increases MPSoC security by lowering power consumption and delay while improving network speed.
Ghanbari, Z. *et al.* (2015)	‘Comparative approach to web application firewalls’, *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)* [Preprint].	Web Application Firewalls (WAF)	Growing dependence on the internet also means growing cyberthreats. Web application firewalls, or WAFs, are essential for all-around security and for thwarting online application threats.

3 Research Methodology

This study shows that the mixed-method design is a reliable and comprehensive strategy for investigating the details of network firewall performance, monitoring efficacy, and security. By employing numerical data to examine crucial performance metrics like throughput, delay, and packet loss rates, firewall efficiency may be evaluated on a quantitative basis. Simultaneously, indicators such as the number of security incidents detected, false positives, and false negatives are employed to gauge the effectiveness of the monitoring system. Qualitative approaches are employed to supplement these quantitative measurements and capture the subjective aspects of the investigation.

Network administrators, security analysts, and end users can all provide valuable insights into the usability and efficacy of the firewall system through surveys and interviews. Via the analysis of cybersecurity specialists' experiences and relevant literature evaluations, qualitative research enhances our understanding of the broader context around security threats. The combination of various qualitative and quantitative data sets facilitates it and raises the study's overall reliability and validity. The research's conclusions are enhanced and more informed decision-making is made possible by the application of a mixed-method methodology. When numerical data shows a significant false positive rate for the monitoring system (Sreekumar, 2023).

3.1 Research Tools

3.1.1 Wire shark

Wireshark, a well-known network protocol analyzer, is crucial to network research since it covers security, efficacy of monitoring, and performance analysis of network firewalls. Analysis and Capture of Packets: Wireshark captures and analyzes network traffic in real time. It allows researchers to examine individual data packets, providing them with detailed information on the communication occurring between devices connected to the network. Decoding Protocols: With Wireshark, a wide range of network protocols may be decoded and shown. Its revelation of the structure and content of each packet aids researchers in their understanding of the communication protocols used in the network.

Network issues can be identified by analyzing packet-level data with Wireshark. In order to evaluate firewalls and track system performance, researchers must be able to spot anomalies, errors, or inefficiencies in network data. Researchers may view network behavior in real time with Wireshark's monitoring tools. It is essential to comprehend how firewalls and monitoring systems respond to traffic issues occurring in real time. Wireshark facilitates the analysis of offline packet data. Researchers' capacity to review and assess data that has been acquired quickly allows for a thorough analysis of network events and potential security issues. Wireshark aids in the identification of performance issues by exposing patterns, bottlenecks, or anomalies in network traffic (Gautam, 2023).

3.1.2 Snort

An open-source prevention and detection of intrusions system called Snort is useful for studying the performance, security, and efficacy of network firewalls. Snort records network activity and continuously examines packets. Using signature-based detection, the system analyzes network traffic to a database of known attack patterns in order to pinpoint known threats. Snort analyzes network protocols to search for anomalies and possible security problems. Based on pre-established rules, Snort creates logs and alerts in response to suspected or malicious activities.

Since its main function is intrusion detection, Snort is a useful tool for identifying and resolving any security vulnerabilities. Snort makes it easier to analyze traffic in-depth, which aids researchers in understanding the many types and trends of network activity. Researchers can immediately respond to security issues and assess the effectiveness of their monitoring systems since Snort detects attacks in real time. By adding and modifying rules in Snort, researchers can tailor the system to the particulars of their network and study goals. Researchers will have access to comprehensive logs and real-time alerts because of Snort, which also makes incident response easier by enabling quick analysis and correction of security incidents (Zenarmor, 2023).

3.2 Evaluation or analysis

Performance evaluation

Important metrics like throughput, delay, and resource consumption are carefully assessed using careful approaches in the quantitative performance evaluation of the network firewall under various real-world scenarios. A comprehensive battery of testing is conducted with real-world stimuli to ascertain how well the firewall can recognize and stop different kinds of attacks.

Throughput Analysis:

One quantitative method to assess throughput is to measure the rate at which data passes through the firewall. A computer model of actual traffic situations is available, encompassing both common network functions and possible vectors of attack (Superfastcpa, 2023).

Measurement of Latency:

The delay the firewall adds when processing network traffic is one of the key factors affecting performance. Tests of delay are conducted under a variety of real-world conditions, including simulated attack traffic and regular traffic. The evaluation takes into account a variety of package sizes, traffic patterns, and network loads.

Attack Simulation Scenarios:

To assess a firewall's effectiveness in identifying and thwarting various cyberthreats, real-world attack scenarios are replicated. To evaluate the firewall's security capabilities, a variety of attack vectors are simulated, such as DDoS attacks, intrusion attempts, and malware injections. Researchers assess a firewall's ability to identify fraudulent activity, neutralize attacks fast, and protect a network from a variety of cyberthreats.

Statistical Analysis:

To assess a firewall's effectiveness in identifying and thwarting various cyberthreats, real-world attack scenarios are replicated. To evaluate the firewall's security capabilities, a variety of attack vectors are simulated, such as DDoS attacks, intrusion attempts, and malware injections. Researchers assess a firewall's ability to identify fraudulent activity, neutralize attacks fast, and protect a network from a variety of cyberthreats.

4 Implementation

4.1 Wireshark

http

The Hypertext Transfer Protocol (HTTP) is a protocol used on the World Wide Web to transport hypermedia content such as HTML files. When a user requests a webpage from a web server, their web browser sends a request message to the server over HTTP, and the server responds with either the desired webpage or an error message. Wireshark is used to get the network protocol that record and analyse network data. It can collect and analyse HTTP traffic, displaying the specifics of individual HTTP requests and answers between a client and a server.

The structure of HTTP communications, typical HTTP methods and status codes, analysing headers and payloads, and comprehending the flow of an HTTP session will all be covered. Wireshark HTTP Traffic Capture

- Setting up Wireshark to collect HTTP traffic.
- Limiting captured traffic to HTTP requests and replies.
- Recognising the many protocols used in an HTTP session

HTTP Message Structure

- Examination of HTTP request and response structure.
- Dissecting the various components of an HTTP message (start line, headers, body).
- HTTP method interpretation (GET, POST, PUT, DELETE) and response status codes

Analysing HTTP Headers - A typical HTTP headers and their importance.

- Deciphering request headers (User-Agent, Referrer, Cookie, and so on).
- Parsing response headers (Content-Type, Content-Length, Server, and so on).
- Examining headers for cache management, compression, and authentication.

HTTP Payload Analysis

- Understanding the type and encoding of HTTP payloads.
- Examining the payload data's structure and content.
- Recognising and analysing embedded payload resources.

Recognising HTTP Session Flow

- Examining the course of an HTTP session from start to finish.
- Recognising the construction and teardown of TCP connections.
- Examining the HTTP request and response sequence inside a session.

HTTP Analysis

- Examining HTTP redirection and learning about the status codes involved.
- Examining HTTP cookies and comprehending their utilisation.

Troubleshooting HTTP Issues

- Recognising and analysing common HTTP errors (404, 500, and etc).
- Investigating network latency and performance problems.
- Recognising probable security flaws in HTTP traffic.

UDP analysis

Applying a filter expression in the Wireshark capture or display filter to filter and focus on UDP packets. To view just UDP packets recorded, use the filter expression 'udp', or apply more specific filters depending on source/destination port, IP addresses, and so on.

TCP analysis

TCP Handshake: The TCP handshake is the first step in establishing a connection between two devices before data transmission can begin. For each packet in the analysis, the source and destination IP addresses, port numbers, sequence numbers, and acknowledgment numbers are presented.

Flow Control and Window Size: To guarantee that data delivery does not overwhelm the receiver, TCP employs flow control algorithms. The receiver advertises the size of its window, which shows how much data it is ready to accept at any one time. The determine the receiver's processing capability for incoming data by using Wireshark to analyze the window size parameter in the TCP header. Larger window sizes indicate a higher ability to accept data, while smaller window widths suggest likely congestion or packet loss.

4.2 Snort

Engine for Rule-Based Detection

Snort uses a rule-based detection engine to identify anomalies in network traffic as well as potential threats. The foundational concepts of Snort's rules, including rule creation, rule selection, and rule actions, will be covered in this section. Additionally, rule matching and alert generating based on matched rules will be covered.

Monitoring and Alerting

In addition to real-time analysis and rule matching for potential criminal investigation and incident response, Snort offers logging services. analyzing the data that Snort has gathered in a variety of log file formats, including the payload, packet headers, and relevant contextual information.

Setup and Configuration of Snort for Traffic Analysis

You will be guided step-by-step through the installation and setup of Snort for traffic analysis in this part. It will include comprehensive instructions on how to configure network interfaces, create rule sets, maximize system performance, and incorporate Snort into an already-existing network architecture.

Live Traffic Capturing and Analysis

The next crucial step after correctly installing Snort is to gather actual traffic for analysis. The discuss various methods, including network taps, span ports, and port mirroring, along with their advantages and disadvantages. In addition, will look into methods for efficiently analyzing traffic that has been collected while maintaining performance in circumstances with a lot of traffic.

Considerations for Performance

Performance of the real-time traffic monitoring system is essential since delays or obstructions could cause alarms to go unnoticed or to be postponed. The evaluation criteria for Snort will be reaction time, CPU load, memory usage, and network throughput.

Packet logging

Capture Network Packets: Gathering network packets is the first stage in packet logging. Snort uses an aggressive network interface to capture network packets. There are several modes in which it can operate, such as inline, passive, and inline with tap. After then, the gathered packets are stored for later analysis.

Snort pre-processes

packets after they have been gathered through processes like IP reassembly, normalization, and defragmentation. This ensures that the network packets are in an analysis-ready format.

Rule Matching

At this point, Snort compares the gathered packets against a predetermined set of rules. Certain patterns or behaviors that are connected to known network attacks or questionable conduct are identified by these criteria. A packet is categorized as an alert or an event of interest when it satisfies a rule.

Logging

After rule matching, Snort logs packet data along with pertinent details including timestamps, source and destination IP addresses, and protocol information. For later analysis, the logged data may be stored locally or sent to a centralized logging system.

Alert Generation

Snort generates warnings whenever a packet fits a specified rule. The severity levels of the alerts can be used to determine the possible danger connected with the observed occurrence. The produced warnings assist security analysts in prioritising their research efforts and responding to potential risks as soon as possible.

Traffic Analysis

Packet recording enables detailed examination of network traffic patterns. Security analysts can discover patterns of normal or suspicious behaviour by studying the logged data. Unusual traffic patterns or flows might signal network misconfigurations, possible assaults, or the existence of malicious activity.

Correlation of Events

The correlation of events across multiple networks is made possible by packet tracking. Analysts can correlate logged data from several network segments or devices to find coordinated attacks, reconnaissance operations, or virus transmission. Event correlation helps to understand the complete threat environment and provides a full picture of the security posture of the network.

Forensic Investigation

Packet logging is extremely important in forensic investigations. The recorded packets include crucial evidence that may be utilised to follow the evolution of an attack and build a timeline of events. Forensic data analysis can give critical information for legal processes, corporate audits, and incident post-mortems.

Content Matching

Criteria for Content Matching

To provide the patterns or signatures it uses to identify harmful behavior in network traffic, Snort uses content matching criteria. A regular expression or byte string can be represented by these signatures. Snort checks packets at the network, transport, and application layers to make sure the content satisfies the specified standards. Among the content matching techniques that can be applied are wildcard matching, regular expression matching, and exact matching.

Rule Enforcement

Snort provides a variety of predefined techniques for handling the detected event when an article match is found. These activities range from straightforward logging to creating alerts and even obstructing traffic. The desired response to an identified threat or penetration attempt dictates the course of action.

Engine of Detection

Network packet analysis and content matching depend heavily on Snort's detection engine. It uses complex algorithms to compare recorded packets from the network interface to the predetermined regulations. Use strategies like rule preparation and effective data structures to improve the content matching process's performance. The detection engine of Snort is built to minimize false positives and false negatives while precisely and effectively identifying potential threats.

Logging and Outputs

Network packet analysis and content matching depend heavily on Snort's detection engine. It uses complex algorithms to compare recorded packets from the network interface to the predetermined regulations. Use strategies like rule preparation and effective data structures to improve the content matching process's performance. The threat detection engine of Snort is built to minimize false positives and false negatives while precisely and effectively identifying potential threats.

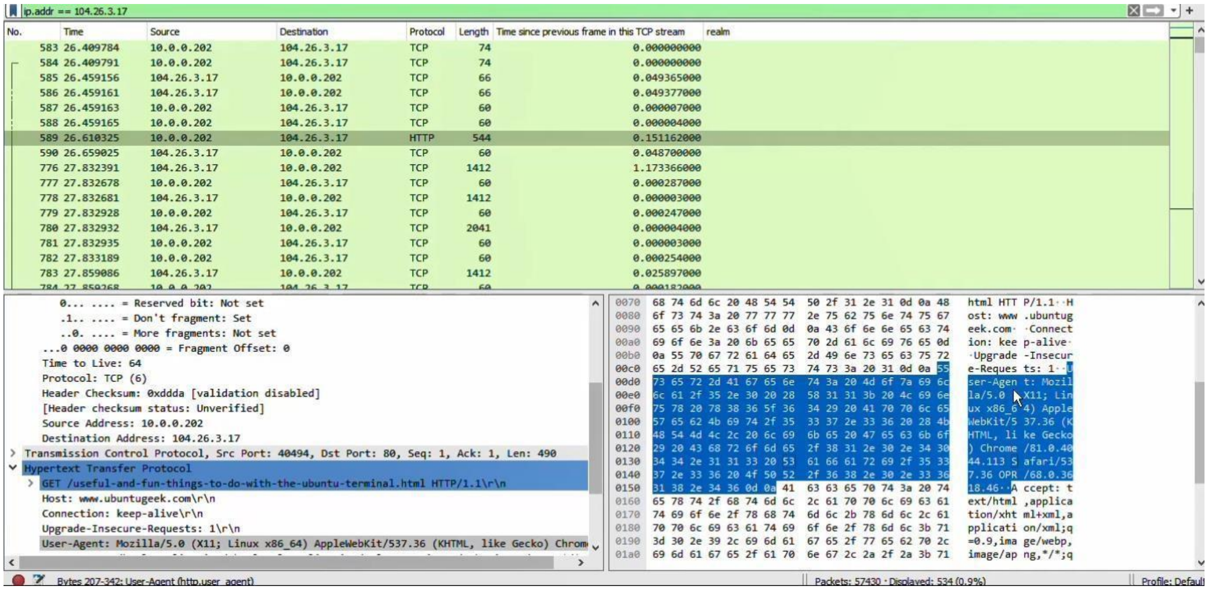
Analysis of protocol

Deconstructing the packets and gathering crucial details about the active protocols are required for this. Snort can interpret packet contents because it contains protocol parsers that understand numerous network protocols' forms and structures. When analyzing HTTP traffic, Snort, for instance, has the ability to extract data about the HTTP method, URL, headers, and any extra parameters included in the request or response.

Rule matching

Deconstructing the packets and gathering crucial details about the active protocols are required for this. Snort can interpret packet contents because it contains protocol parsers that understand numerous network protocols' forms and structures. When analyzing HTTP traffic, Snort, for instance, has the ability to extract data about the HTTP method, URL, headers, and any extra parameters included in the request or response.

4.3 Results of implementation



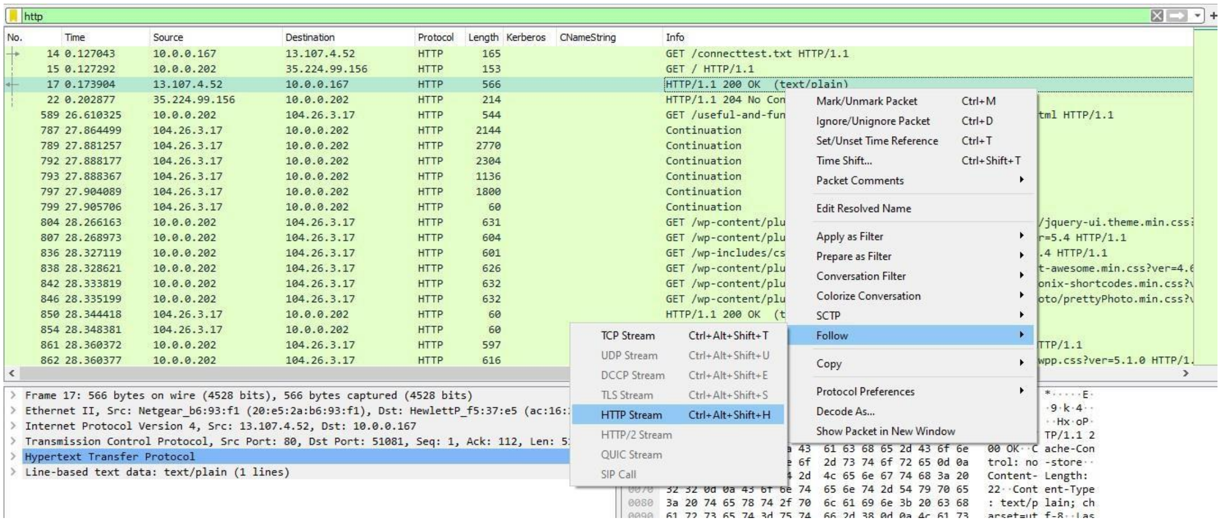
```
ip.addr == 104.26.3.17

No.   Time        Source       Destination  Protocol Length Time since previous frame in this TCP stream  realm
  583 26.409784   10.0.0.202   104.26.3.17  TCP        74   0.000000000
  584 26.409791   10.0.0.202   104.26.3.17  TCP        74   0.000000000
  585 26.459156   104.26.3.17  10.0.0.202   TCP        66   0.049365000
  586 26.459161   104.26.3.17  10.0.0.202   TCP        66   0.049377000
  587 26.459163   10.0.0.202   104.26.3.17  TCP        60   0.000007000
  588 26.459165   10.0.0.202   104.26.3.17  TCP        60   0.000004000
  589 26.610325   10.0.0.202   104.26.3.17  HTTP      544   0.151162000
  590 26.659025   104.26.3.17  10.0.0.202   TCP        60   0.048700000
  776 27.832391   104.26.3.17  10.0.0.202   TCP      1412   1.173366000
  777 27.832678   10.0.0.202   104.26.3.17  TCP        60   0.000287000
  778 27.832681   104.26.3.17  10.0.0.202   TCP      1412   0.000003000
  779 27.832928   10.0.0.202   104.26.3.17  TCP        60   0.000247000
  780 27.832932   104.26.3.17  10.0.0.202   TCP      2041   0.000004000
  781 27.832935   10.0.0.202   104.26.3.17  TCP        60   0.000003000
  782 27.833189   10.0.0.202   104.26.3.17  TCP        60   0.000254000
  783 27.859086   104.26.3.17  10.0.0.202   TCP      1412   0.025897000
  784 27.859268   10.0.0.202   104.26.3.17  TCP        60   0.000182000
```

```
      0... .... = Reserved bit: Not set
      .1.. .... = Don't fragment: Set
      ..0. .... = More fragments: Not set
    ...0 0000 0000 0000 = Fragment Offset: 0
    Time to Live: 64
    Protocol: TCP (6)
    Header Checksum: 0xddda [validation disabled]
    [Header checksum status: Unverified]
    Source Address: 10.0.0.202
    Destination Address: 104.26.3.17
> Transmission Control Protocol, Src Port: 40494, Dst Port: 80, Seq: 1, Ack: 1, Len: 490
v Hypertext Transfer Protocol
  > GET /useful-and-fun-things-to-do-with-the-ubuntu-terminal.html HTTP/1.1\r\n
    Host: www.ubuntugeek.com\r\n
    Connection: keep-alive\r\n
    Upgrade-Insecure-Requests: 1\r\n
    User-Agent: Mozilla/5.0 (X11; Linux x86_64) AppleWebKit/537.36 (KHTML, like Gecko) Chrom
```

```
0070  68 74 6d 6c 20 48 54 54  50 2f 31 2e 31 0d 0a 48   html HTT P/1.1··H
0080  6f 73 74 3a 20 77 77 77  2e 75 62 75 6e 74 75 67   ost: www .ubuntug
0090  65 65 6b 2e 63 6f 6d 0d  0a 43 6f 6e 6e 65 63 74   eek.com· ·Connect
00a0  69 6f 6e 3a 20 6b 65 65  70 2d 61 6c 69 76 65 0d   ion: kee p-alive·
00b0  0a 55 70 67 72 61 64 65  2d 49 6e 73 65 63 75 72   ·Upgrade -Insecur
00c0  65 2d 52 65 71 75 65 73  74 73 3a 20 31 0d 0a 55   e-Reques ts: 1··U
00d0  73 65 72 2d 41 67 65 6e  74 3a 20 4d 6f 7a 69 6c   ser-Agen t: Mozil
00e0  6c 61 2f 35 2e 30 20 28  58 31 31 3b 20 4c 69 6e   la/5.0 ( X11; Lin
00f0  75 78 20 78 38 36 5f 36  34 29 20 41 70 70 6c 65   ux x86_6 4) Apple
0100  57 65 62 4b 69 74 2f 35  33 37 2e 33 36 20 28 4b   WebKit/5 37.36 (K
0110  48 54 4d 4c 2c 20 6c 69  6b 65 20 47 65 63 6b 6f   HTML, li ke Gecko
0120  29 20 43 68 72 6f 6d 65  2f 38 31 2e 30 2e 34 30   ) Chrome /81.0.40
0130  34 34 2e 31 31 33 20 53  61 66 61 72 69 2f 35 33   44.113 S afari/53
0140  37 2e 33 36 20 4f 50 52  2f 36 38 2e 30 2e 33 36   7.36 OPR /68.0.36
0150  31 38 2e 34 36 0d 0a 41  63 63 65 70 74 3a 20 74   18.46··A ccept: t
0160  65 78 74 2f 68 74 6d 6c  2c 61 70 70 6c 69 63 61   ext/html ,applica
0170  74 69 6f 6e 2f 78 68 74  6d 6c 2b 78 6d 6c 2c 61   tion/xht ml+xml,a
0180  70 70 6c 69 63 61 74 69  6f 6e 2f 78 6d 6c 3b 71   pplicati on/xml;q
0190  3d 30 2e 39 2c 69 6d 61  67 65 2f 77 65 62 70 2c   =0.9,ima ge/webp,
01a0  69 6d 61 67 65 2f 61 70  6e 67 2c 2a 2f 2a 3b 71   image/ap ng,*/*;q
```

```
Bytes 207-342: User-Agent (http.user_agent)     Packets: 57430 · Displayed: 534 (0.9%)     Profile: Default
```

Figure 2 Ip configuration analysis



```
http

No.   Time        Source         Destination    Protocol Length Kerberos CNameString Info
   14 0.127043    10.0.0.167     13.107.4.52    HTTP      165                        GET /connecttest.txt HTTP/1.1
   15 0.127292    10.0.0.202     35.224.99.156  HTTP      153                        GET / HTTP/1.1
   17 0.173904    13.107.4.52    10.0.0.167     HTTP      566                        HTTP/1.1 200 OK  (text/plain)
   22 0.202877    35.224.99.156  10.0.0.202     HTTP      214                        HTTP/1.1 204 No Con
  589 26.610325   10.0.0.202     104.26.3.17    HTTP      544                        GET /useful-and-fun                           tml HTTP/1.1
  787 27.864499   104.26.3.17    10.0.0.202     HTTP     2144                        Continuation
  789 27.881257   104.26.3.17    10.0.0.202     HTTP     2770                        Continuation
  792 27.888177   104.26.3.17    10.0.0.202     HTTP     2304                        Continuation
  793 27.888367   104.26.3.17    10.0.0.202     HTTP     1136                        Continuation
  797 27.904089   104.26.3.17    10.0.0.202     HTTP     1800                        Continuation
  799 27.905706   104.26.3.17    10.0.0.202     HTTP       60                        Continuation
  804 28.266163   10.0.0.202     104.26.3.17    HTTP      631                        GET /wp-content/plu                           /jquery-ui.theme.min.css
  807 28.268973   10.0.0.202     104.26.3.17    HTTP      604                        GET /wp-content/plu                           r=5.4 HTTP/1.1
  836 28.327119   10.0.0.202     104.26.3.17    HTTP      601                        GET /wp-includes/cs                           .4 HTTP/1.1
  838 28.328621   10.0.0.202     104.26.3.17    HTTP      626                        GET /wp-content/plu                           t-awesome.min.css?ver=4.6
  842 28.333819   10.0.0.202     104.26.3.17    HTTP      632                        GET /wp-content/plu                           onix-shortcodes.min.css?\
  846 28.335199   10.0.0.202     104.26.3.17    HTTP      632                        GET /wp-content/plu                           oto/prettyPhoto.min.css?\
  850 28.344418   104.26.3.17    10.0.0.202     HTTP       60                        HTTP/1.1 200 OK  (t
  854 28.348381   104.26.3.17    10.0.0.202     HTTP       60
  861 28.360372   10.0.0.202     104.26.3.17    HTTP      597                                                                      TTP/1.1
  862 28.360377   10.0.0.202     104.26.3.17    HTTP      616                                                                      wpp.css?ver=5.1.0 HTTP/1.
```

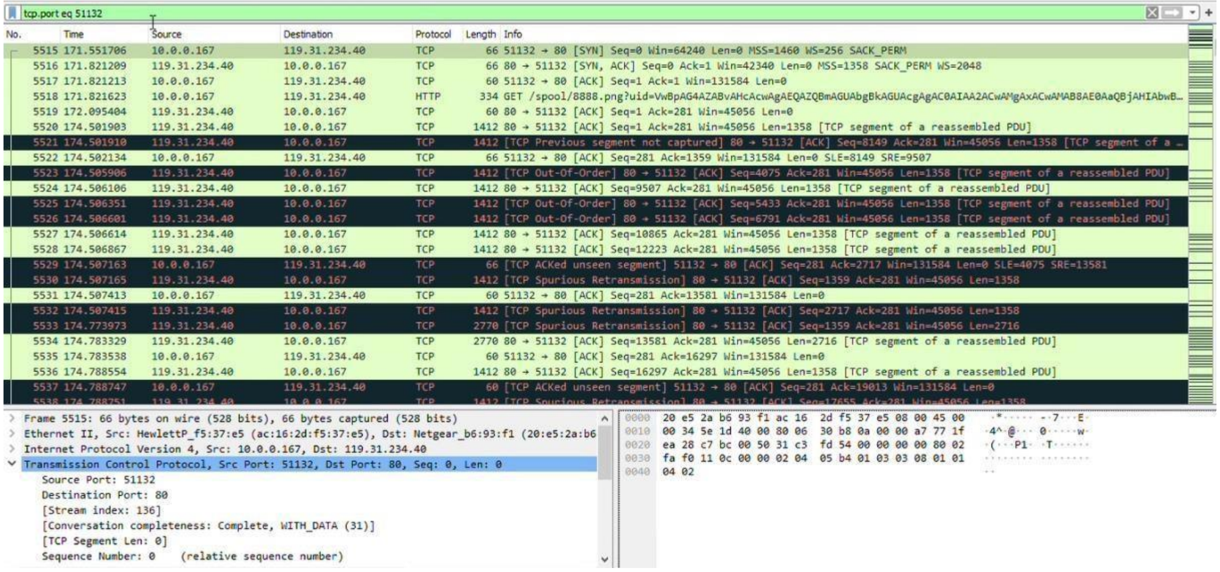
```
> Frame 17: 566 bytes on wire (4528 bits), 566 bytes captured (4528 bits)
> Ethernet II, Src: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1), Dst: HewlettP_f5:37:e5 (ac:16:
> Internet Protocol Version 4, Src: 13.107.4.52, Dst: 10.0.0.167
> Transmission Control Protocol, Src Port: 80, Dst Port: 51081, Seq: 1, Ack: 112, Len: 5
> Hypertext Transfer Protocol
> Line-based text data: text/plain (1 lines)
```

```
                                                     *·····E·
                                                     ·9·k·4··
                                                     ··Hx·oP·
                                                     TP/1.1 2
      a 43  61 63 68 65 2d 43 6f 6e   00 OK··C ache-Con
      e 6f  2d 73 74 6f 72 65 0d 0a   trol: no -store··
      4 2d  4c 65 6e 67 74 68 3a 20   Content- Length:
0070  32 32 0d 0a 43 6f 6e 74  65 6e 74 2d 54 79 70 65   22··Cont ent-Type
0080  3a 20 74 65 78 74 2f 70  6c 61 69 6e 3b 20 63 68   : text/p lain; ch
0090  61 72 73 65 74 3d 75 74  66 2d 38 0d 0a 4c 61 73   arset=ut f-8··Las
```

Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
DCCP Stream Ctrl+Alt+Shift+E
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
SIP Call

Figure 3 data import HTTP analysis

The "http" filter is a sophisticated tool that enables network administrators and developers to monitor and analyse HTTP traffic in order to diagnose and repair HTTP protocol issues. The ability to monitor and debug HTTP traffic becomes increasingly crucial in assuring optimum performance and functionality as online applications develop more sophisticated and feature-rich. Debugging is a popular use for the "http" filter. Debugging HTTP-related problems often demands evaluating the complete context of a request and its corresponding response. The "http" filter gives a broad variety of filtering options for isolating particular kinds of traffic. Filtering based on request or response protocols, status codes, or even certain phrases within the sent data is an example of this. These filtering options allow the network to focus on particular aspects of HTTP traffic, making analysis more targeted and efficient. The "http" filter is vital for security and compliance in addition to debugging and performance monitoring. This allows them to take proactive steps to prevent any attacks and secure the application's and its data's integrity and confidentiality.

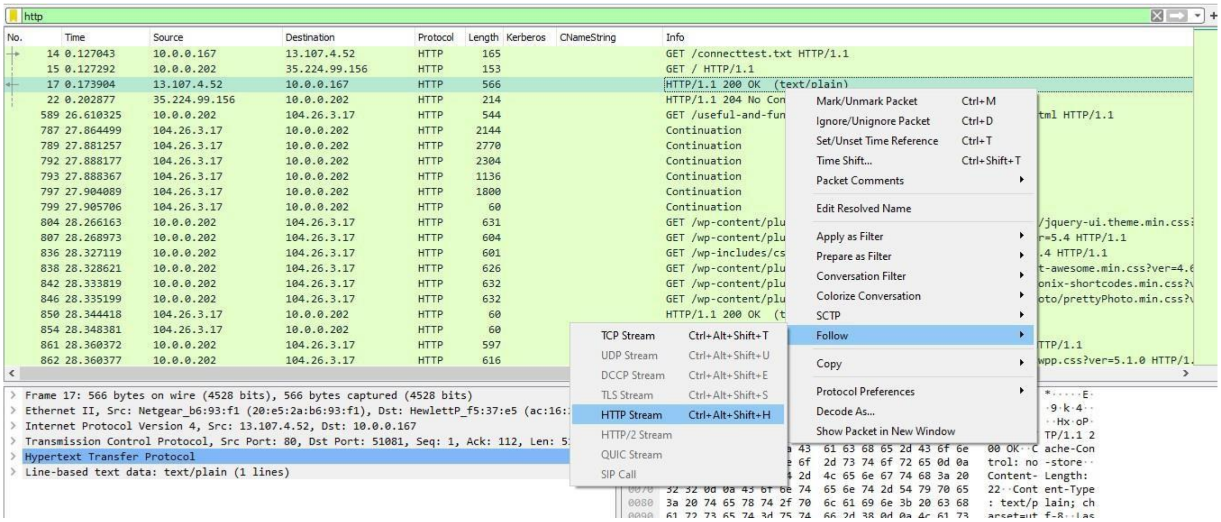


```
tcp.port eq 51132

No.  Time        Source         Destination    Protocol Length Info
5515 171.551706  10.0.0.167     119.31.234.40  TCP      66 51132 → 80 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM
5516 171.821209  119.31.234.40  10.0.0.167     TCP      66 80 → 51132 [SYN, ACK] Seq=0 Ack=1 Win=42340 Len=0 MSS=1358 SACK_PERM WS=2048
5517 171.821213  10.0.0.167     119.31.234.40  TCP      60 51132 → 80 [ACK] Seq=1 Ack=1 Win=131584 Len=0
5518 171.821623  10.0.0.167     119.31.234.40  HTTP     334 GET /spool/8888.png?uid=VwBpAG4AZABvAHcAcwAgAEQAZQBmAGUAbgBkAGUAcgAgAC0AIAA2ACwAMgAxACwAMAB8AE0AaQBjAHIAbwB…
5519 172.095404  119.31.234.40  10.0.0.167     TCP      60 80 → 51132 [ACK] Seq=1 Ack=281 Win=45056 Len=0
5520 174.501903  119.31.234.40  10.0.0.167     TCP      1412 80 → 51132 [ACK] Seq=1 Ack=281 Win=45056 Len=1358 [TCP segment of a reassembled PDU]
5521 174.501910  119.31.234.40  10.0.0.167     TCP      1412 [TCP Previous segment not captured] 80 → 51132 [ACK] Seq=8149 Ack=281 Win=45056 Len=1358 [TCP segment of a …
5522 174.502134  10.0.0.167     119.31.234.40  TCP      66 51132 → 80 [ACK] Seq=281 Ack=1359 Win=131584 Len=0 SLE=8149 SRE=9507
5523 174.505906  119.31.234.40  10.0.0.167     TCP      1412 [TCP Out-Of-Order] 80 → 51132 [ACK] Seq=4075 Ack=281 Win=45056 Len=1358 [TCP segment of a reassembled PDU]
5524 174.506106  119.31.234.40  10.0.0.167     TCP      1412 80 → 51132 [ACK] Seq=9507 Ack=281 Win=45056 Len=1358 [TCP segment of a reassembled PDU]
5525 174.506351  119.31.234.40  10.0.0.167     TCP      1412 [TCP Out-Of-Order] 80 → 51132 [ACK] Seq=5433 Ack=281 Win=45056 Len=1358 [TCP segment of a reassembled PDU]
5526 174.506601  119.31.234.40  10.0.0.167     TCP      1412 [TCP Out-Of-Order] 80 → 51132 [ACK] Seq=6791 Ack=281 Win=45056 Len=1358 [TCP segment of a reassembled PDU]
5527 174.506614  119.31.234.40  10.0.0.167     TCP      1412 80 → 51132 [ACK] Seq=10865 Ack=281 Win=45056 Len=1358 [TCP segment of a reassembled PDU]
5528 174.506867  119.31.234.40  10.0.0.167     TCP      1412 80 → 51132 [ACK] Seq=12223 Ack=281 Win=45056 Len=1358 [TCP segment of a reassembled PDU]
5529 174.507163  10.0.0.167     119.31.234.40  TCP      66 [TCP ACKed unseen segment] 51132 → 80 [ACK] Seq=281 Ack=2717 Win=131584 Len=0 SLE=4075 SRE=13581
5530 174.507165  119.31.234.40  10.0.0.167     TCP      1412 [TCP Spurious Retransmission] 80 → 51132 [ACK] Seq=1359 Ack=281 Win=45056 Len=1358
5531 174.507413  10.0.0.167     119.31.234.40  TCP      60 51132 → 80 [ACK] Seq=281 Ack=13581 Win=131584 Len=0
5532 174.507415  119.31.234.40  10.0.0.167     TCP      1412 [TCP Spurious Retransmission] 80 → 51132 [ACK] Seq=2717 Ack=281 Win=45056 Len=1358
5533 174.773973  119.31.234.40  10.0.0.167     TCP      2770 [TCP Spurious Retransmission] 80 → 51132 [ACK] Seq=1359 Ack=281 Win=45056 Len=2716
5534 174.783329  119.31.234.40  10.0.0.167     TCP      2770 80 → 51132 [ACK] Seq=13581 Ack=281 Win=45056 Len=2716 [TCP segment of a reassembled PDU]
5535 174.783538  10.0.0.167     119.31.234.40  TCP      60 51132 → 80 [ACK] Seq=281 Ack=16297 Win=131584 Len=0
5536 174.788554  119.31.234.40  10.0.0.167     TCP      1412 80 → 51132 [ACK] Seq=16297 Ack=281 Win=45056 Len=1358 [TCP segment of a reassembled PDU]
5537 174.788747  10.0.0.167     119.31.234.40  TCP      60 [TCP ACKed unseen segment] 51132 → 80 [ACK] Seq=281 Ack=19013 Win=131584 Len=0
5538 174.788751  119.31.234.40  10.0.0.167     TCP      1412 [TCP Spurious Retransmission] 80 → 51132 [ACK] Seq=17655 Ack=281 Win=45056 Len=1358

> Frame 5515: 66 bytes on wire (528 bits), 66 bytes captured (528 bits)
> Ethernet II, Src: HewlettP_f5:37:e5 (ac:16:2d:f5:37:e5), Dst: Netgear_b6:93:f1 (20:e5:2a:b6
> Internet Protocol Version 4, Src: 10.0.0.167, Dst: 119.31.234.40
v Transmission Control Protocol, Src Port: 51132, Dst Port: 80, Seq: 0, Len: 0
    Source Port: 51132
    Destination Port: 80
    [Stream index: 136]
    [Conversation completeness: Complete, WITH_DATA (31)]
    [TCP Segment Len: 0]
    Sequence Number: 0    (relative sequence number)

0000  20 e5 2a b6 93 f1 ac 16  2d f5 37 e5 08 00 45 00   ·*······ -·7···E·
0010  00 34 5e 1d 40 00 80 06  30 b8 0a 00 00 a7 77 1f   ·4^·@··· 0·····w·
0020  ea 28 c7 bc 00 50 31 c3  fd 54 00 00 00 00 80 02   ·(···P1· ·T······
0030  fa f0 11 0c 00 00 02 04  05 b4 01 03 03 08 01 01   ········ ········
0040  04 02                                              ··
```

Figure 4 tcp port analysis

Wireshark reveals only network packets that are either heading for or originating from the given port number when the "tcp.port eq" filter is used. TCP ports are numbered from 0 to 65535, with different port numbers reserved for certain protocols or services. By filtering network traffic based on TCP port numbers, analysts may concentrate specifically on the protocols or applications of interest, offering insights that may be exploited for a range of aims such as troubleshooting, performance optimisation, or security research. Using Wireshark's "tcp.port eq" filter, analysts may delve more into the behaviours and characteristics of protocols or applications that utilise specified TCP port numbers. Furthermore, the "tcp.port eq" filter is beneficial in circumstances where network traffic is heavy, making it tough for analysts to sort through all of the packets.

Analysts may substantially minimise the quantity of collected packets by filtering based on TCP port numbers, enabling them to speedily evaluate the relevant packets without being overburdened.


http
Mark/Unmark Packet Ctrl+M
Ignore/Unignore Packet Ctrl+D
Set/Unset Time Reference Ctrl+T
Time Shift... Ctrl+Shift+T
Packet Comments
Edit Resolved Name
Apply as Filter
Prepare as Filter
Conversation Filter
Colorize Conversation
SCTP
Follow
Copy
Protocol Preferences
Decode As...
Show Packet in New Window
TCP Stream Ctrl+Alt+Shift+T
UDP Stream Ctrl+Alt+Shift+U
DCCP Stream Ctrl+Alt+Shift+E
TLS Stream Ctrl+Alt+Shift+S
HTTP Stream Ctrl+Alt+Shift+H
HTTP/2 Stream
QUIC Stream
SIP Call

```
No.   Time       Source         Destination    Protocol Length Kerberos CNameString Info
   14 0.127043   10.0.0.167     13.107.4.52    HTTP      165                        GET /connecttest.txt HTTP/1.1
   15 0.127292   10.0.0.202     35.224.99.156  HTTP      153                        GET / HTTP/1.1
   17 0.173904   13.107.4.52    10.0.0.167     HTTP      566                        HTTP/1.1 200 OK  (text/plain)
   22 0.202877   35.224.99.156  10.0.0.202     HTTP      214                        HTTP/1.1 204 No Con
  589 26.610325  10.0.0.202     104.26.3.17    HTTP      544                        GET /useful-and-fun                            tml HTTP/1.1
  787 27.864499  104.26.3.17    10.0.0.202     HTTP     2144                        Continuation
  789 27.881257  104.26.3.17    10.0.0.202     HTTP     2770                        Continuation
  792 27.888177  104.26.3.17    10.0.0.202     HTTP     2304                        Continuation
  793 27.888367  104.26.3.17    10.0.0.202     HTTP     1136                        Continuation
  797 27.904089  104.26.3.17    10.0.0.202     HTTP     1800                        Continuation
  799 27.905706  104.26.3.17    10.0.0.202     HTTP       60                        Continuation
  804 28.266163  10.0.0.202     104.26.3.17    HTTP      631                        GET /wp-content/plu                            /jquery-ui.theme.min.css
  807 28.268973  10.0.0.202     104.26.3.17    HTTP      604                        GET /wp-content/plu                            r=5.4 HTTP/1.1
  836 28.327119  10.0.0.202     104.26.3.17    HTTP      601                        GET /wp-includes/cs                            .4 HTTP/1.1
  838 28.328621  10.0.0.202     104.26.3.17    HTTP      626                        GET /wp-content/plu                            t-awesome.min.css?ver=4.6
  842 28.333819  10.0.0.202     104.26.3.17    HTTP      632                        GET /wp-content/plu                            onix-shortcodes.min.css?\
  846 28.335199  10.0.0.202     104.26.3.17    HTTP      632                        GET /wp-content/plu                            oto/prettyPhoto.min.css?\
  850 28.344418  104.26.3.17    10.0.0.202     HTTP       60                        HTTP/1.1 200 OK  (t
  854 28.348381  104.26.3.17    10.0.0.202     HTTP       60
  861 28.360372  10.0.0.202     104.26.3.17    HTTP      597                                                                       TTP/1.1
  862 28.360377  10.0.0.202     104.26.3.17    HTTP      616                                                                       wpp.css?ver=5.1.0 HTTP/1.

> Frame 17: 566 bytes on wire (4528 bits), 566 bytes captured (4528 bits)
> Ethernet II, Src: Netgear_b6:93:f1 (20:e5:2a:b6:93:f1), Dst: HewlettP_f5:37:e5 (ac:16:
> Internet Protocol Version 4, Src: 13.107.4.52, Dst: 10.0.0.167
> Transmission Control Protocol, Src Port: 80, Dst Port: 51081, Seq: 1, Ack: 112, Len: 5
> Hypertext Transfer Protocol
> Line-based text data: text/plain (1 lines)

                                                  *·····E·
                                                  ·9·k·4··
                                                  ··Hx·oP·
                                                  TP/1.1 2
      a 43  61 63 68 65 2d 43 6f 6e    00 OK··C ache-Con
      e 6f  2d 73 74 6f 72 65 0d 0a    trol: no -store··
      4 2d  4c 65 6e 67 74 68 3a 20    Content- Length:
0070  32 32 0d 0a 43 6f 6e 74  65 6e 74 2d 54 79 70 65    22··Cont ent-Type
0080  3a 20 74 65 78 74 2f 70  6c 61 69 6e 3b 20 63 68    : text/p lain; ch
0090  61 72 73 65 74 3d 75 74  66 2d 38 0d 0a 4c 61 73    arset=ut f-8··Las
```

Figure 5 HTTP stream

The result reveals by tackling challenges associated to online applications and HTTP-based services, analysts and developers commonly need to concentrate their attention on HTTP interactions between servers and clients, such as web browsers. They may speed up this process by employing the "http" filter, which enables them to inspect and evaluate HTTP exchanges by deleting any unnecessary packets from the recorded stream. When it comes to debugging online programmes, the "http" filter is particularly beneficial. When an application fails, developers may use this filter to isolate and study the HTTP requests and answers made between the client and the server.

4.4 Discussion

Firewall generation has grown from traditional perimeter protection devices to crucial components of community architecture that provide layered safety across networks, as mentioned inside the literature. Th research mainly focus how firewalls are complicated devices that use a mixture of software program and hardware components to govern community site visitors and impose protection guidelines. This is particularly clear whilst stateful evaluation and packet filtering are used, which observe records packets and their context to defend community integrity. Furthermore, the proxy offerings and application-layer filtering highlights the development of increasingly complex techniques for mitigating information risks.

The effectiveness and scalability of modern-day firewalls are proven by way of their incorporation into network architecture. In conditions with heavy site visitors, these designs make it easier to distribute the enforcement of safety rules over many devices, enhancing overall performance and decreasing latency. Furthermore, as we will see, the idea of Network Edge Security provides a decentralized technique of deploying firewalls that successfully counters inner threats and keeps up with rising technology like VPNs and cellular computing. Essentially, contemporary firewalls cross beyond their preliminary reason to symbolize a dynamic, policy-driven safety mechanism that is important for protecting network perimeters from new and rising threats. In an ever-changing virtual environment, the planned deployment and ongoing adaption of firewall technologies are important to sustaining robust community security.

4.5 Limitation

The research is limited by the inherent difficulties of researching quickly developing cybersecurity technology. Because the digital world is always changing, it may be challenging to keep up with the latest advances, which might result in knowledge gaps about new risks and defences. Furthermore, given how quickly firewall technology is developing, the research may be limited by the lack of thorough and current literature. The results' generalizability might potentially be affected by the various ways that firewall solutions are implemented in various network topologies and organisational contexts. Lastly, the research's length could not completely account for unanticipated advancements or paradigm changes in cybersecurity, which would restrict a thorough examination of the problems facing network security today.

5 Conclusion and Future Work

Report analysis the malicious network packet, through the network pack in wireshark shows the result which contains the analysis of network firewall and attack through the network, same goes to the snort analysis where it also analysis the same. The findings of the research demonstrate how important Wireshark is for offering in-depth explanations of detected malicious files in the system. whereas Snort, Wireshark not only detect threats but also provides detailed information about the characteristics of these malicious entities like the location, size. One major benefit of using Wireshark is that it may provide information on where these dangers are located in the system as well as additional relevant details which help the experts to resolve that. This information enables network managers and security experts to take immediate action and put custom mitigation methods in place in order to stop the threats. This studys detailed understanding of network security provides a number of suggestions for maximizing the efficiency of network firewalls for the readers and also for organizations looking for such tools. First and foremost, Wireshark should be a part of any organization's security toolkit as it is displayed by the researcher in the above research

with evidence. By using its detailed insights, it can improve the visibility of threats. Also, firewall rules must be updated frequently and monitored continuously in order to properly respond to the changing threat landscape. Also, proactive threat detection can be enhanced by integrating artificial intelligence and machine learning algorithms into firewall systems as these technologies are playing a vital role in enhancing the different working models. This allows for the identification of new threats based on patterns and anomalies. In order to ensure that tools like Wireshark continue to be enhanced and strengthened, increasing their capacity to counter new and emerging cyber threats, cooperation between security experts and the open-source community should be promoted.

6 References

Ali Bin Hamid Ali, F. (2011) ‘A Study of Technology in firewall system’, *2011 IEEE Symposium on Business, Engineering and Industrial Applications (ISBEIA)* [Preprint]. doi:10.1109/isbeia.2011.6088813.

Bin Ali, M.N., Rahman, M.L. and Hossain, S.A. (2013) ‘Network Architecture and security issues in campus networks’, *2013 Fourth International Conference on Computing, Communications and Networking Technologies (ICCCNT)* [Preprint]. doi:10.1109/icccnt.2013.6726595.

Chadwick, D.W. (2023) *Network Firewall Technologies* [Preprint].

Fulp, E.W. (2006) ‘An independent function-parallel firewall architecture for high-speed networks (short paper)’, *Information and Communications Security*, pp. 292–301. doi:10.1007/11935308_21.

Fulp, E.W. and Tarsa, S.J. (2023) *Network Firewall Policy Tries* [Preprint].

Gautam, S. (2023) *What is wireshark? applications, features & how it works*, *KnowledgeHut*. Available at: https://www.knowledgehut.com/blog/security/what-is-wireshark (Accessed: 06 December 2023).

GeeksforGeeks (2021) *The importance of using a firewall*, *GeeksforGeeks*. Available at: https://www.geeksforgeeks.org/the-importance-of-using-a-firewall/ (Accessed: 06 December 2023).

Ghanbari, Z. *et al.* (2015) ‘Comparative approach to web application firewalls’, *2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI)* [Preprint]. doi:10.1109/kbei.2015.7436148.

Grammatikakis, M.D. *et al.* (2014) ‘Security Effectiveness and a hardware firewall for mpsocs’, *2014 IEEE Intl Conf on High Performance Computing and Communications, 2014 IEEE 6th Intl Symp on Cyberspace Safety and Security, 2014 IEEE 11th Intl Conf on Embedded Software and Syst (HPCC,CSS,ICESS)* [Preprint]. doi:10.1109/hpcc.2014.173.

Kaspersky (2023) *What is a Firewall? definition and explanation*, *www.kaspersky.com*. Available at: https://www.kaspersky.com/resource-center/definitions/firewall (Accessed: 06 December 2023).

Lee, C.P. *et al.* (2005) ‘Visual firewall: Real-time Network Security Monitor’, *IEEE Workshop on Visualization for Computer Security, 2005. (VizSEC 05).* [Preprint]. doi:10.1109/vizsec.2005.1532075.

Lyu, M.R. and Lau, L.K.Y. (2001) ‘Firewall security: Policies, testing and performance evaluation’, *Proceedings 24th Annual International Computer Software and Applications Conference. COMPSAC2000* [Preprint]. doi:10.1109/cmpsac.2000.884700.

Maheshwari, N. and Dagale, H. (2018) ‘Secure communication and firewall architecture for IOT Applications’, *2018 10th International Conference on Communication Systems &amp; Networks (COMSNETS)* [Preprint]. doi:10.1109/comsnets.2018.8328215.

Markham, T. and Payne, C. (2001) ‘Security at the Network Edge: A distributed firewall architecture’, *Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX’01* [Preprint]. doi:10.1109/discex.2001.932222.

Sheth, C. and Thakker, R. (2011) ‘Performance evaluation and comparative analysis of network firewalls’, *2011 International Conference on Devices and Communications (ICDeCom)* [Preprint]. doi:10.1109/icdecom.2011.5738566.

Sheth, C. and Thakker, R. (2013) ‘Performance evaluation and comparison of network firewalls under ddos attack’, *International Journal of Computer Network and Information Security*, 5(12), pp. 60–67. doi:10.5815/ijcnis.2013.12.08.

Shu, J., Dai, H.-B. and Zhu, C. (2011) ‘Securouter — A novel dynamic firewall system embedded with IDS integration’, *International Conference on Computer Technology and Development, 3rd (ICCTD 2011)*, pp. 2361–2365. doi:10.1115/1.859919.paper387.

Sreekumar, D. (2023) *What is research methodology? definition, types, and examples: Paperpal, Paperpal Blog*. Available at: https://paperpal.com/blog/academic-writing-guides/what-is-research-methodology#:~:text=A%20research%20methodology%20describes%20the,using%20the%20selected%20research%20instruments. (Accessed: 06 December 2023).

Superfastcpa (2023) *What is throughput analysis?, SuperfastCPA CPA Review*. Available at: https://www.superfastcpa.com/what-is-throughput-analysis/#:~:text=Throughput%20Analysis%20is%20an%20analytical,inputs%2C%20constraints%2C%20or%20configuration. (Accessed: 06 December 2023).

wrike (2023) *What is Agile Methodology in project management?, Versatile & Robust Project Management Software*. Available at: https://www.wrike.com/project-management-guide/faq/what-is-agile-methodology-in-project-management/ (Accessed: 06 December 2023).

Xuan, S. *et al.* (2016) ‘Performance evaluation model for Application Layer Firewalls’, *PLOS ONE*, 11(11). doi:10.1371/journal.pone.0167280.

Yue, X., Chen, W. and Wang, Y. (2009) ‘The research of Firewall Technology in Computer Network Security’, *2009 Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA)* [Preprint]. doi:10.1109/paciia.2009.5406566.

Zenarmor (2023) *Snort ids/IPS explained. what - why you need - how it works, Zenarmor*. Available at: https://www.zenarmor.com/docs/network-security-tutorials/what-is-snort#:~:text=SNORT%20generates%20a%20set%20of,and%20major%20BSD%20operating%20systems. (Accessed: 06 December 2023).