# Implementing Intrusion Detection Systems and Intrusion Prevention Systems in pfSense

## Introduction

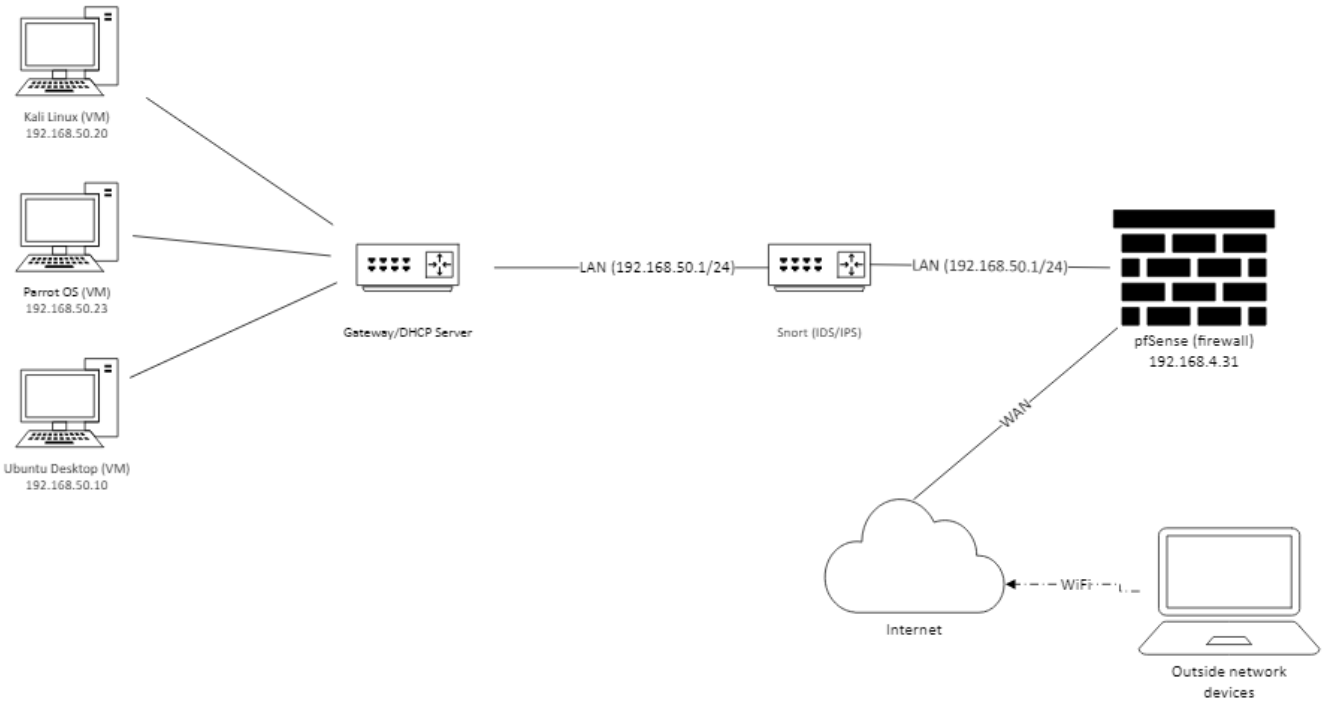
In 2023, there were 2,365 cyberattacks, with a total of 343,338,964 victims (St. John, 2024). Cyber threats are becoming more complicated over time and the role of a firewall, and an intrusion detection system (IDS) and intrusion prevention system (IPS) are becoming more relevant and important in a network environment. While firewalls are meant to block network ports or block data coming and going to a network address (CISA, 2023), IDS and IPS focus more on how to process and react to cyber-attacks when they occur. “An intrusion detection system (IDS) is software that automates the intrusion detection process. An intrusion prevention system (IPS) is software that has all the capabilities of an IDS and can also attempt to stop possible incidents” (Stavroulakis & Stamp, 2010, pp. 177–178). One key difference between the two systems is an IPS can respond to a detected threat by a configured rule made by an end-user (Stavroulakis & Stamp, 2010, pp. 177–178).

This project will use an open-source IDS and IPS software called Snort. Snort is a third-party package that can be installed on pfSense. “Snort has three primary uses: As a packet sniffer like tcpdump, as a packet logger – which is useful for network traffic debugging, or it can be used as a full-blown network intrusion prevention system. Snort can be downloaded and configured for personal, and business use alike” (*Snort - Network Intrusion Detection & Prevention System*, n.d.). Snort also provides community rulesets, which are “developed, tested, and approved by Cisco Talos” (*Snort - Network Intrusion Detection & Prevention System*, n.d.). These rules can be enabled or disabled at any time for each different network connection. Snort being open source, provides an opportunity for users to make changes to meet their needs.

The purpose of this research is to analyze how effective both systems are when it comes to responding to a cyber threat in a home or small office network environment. At the same time, the impact on network security that both systems can bring to the environment. Not only will the purpose of this project be to analyze the functionalities of IDS and IPS systems, but to gather more experience in network administration and managing virtual environments.

## Network Architecture and Design

The network architecture and environment are shown in Figure 1. The environment simulates a small office or home network environment running on a virtual machine software called Oracle VirtualBox. The virtual environment is composed of three internally connected computer systems running different operating systems (OS). The OS running on each virtual system is Kali Linux, Parrot OS, and Ubuntu Desktop OS. The pfSense firewall software is running on a standalone virtual machine (VM) with network adapters configured to simulate a gateway, a local-area network (LAN), and a wide-area network (WAN). The IDS and IPS are both configured through an installed third-party package called Snort. Finally, one device outside the virtual environment was used to conduct common network attacks against the firewall to test the IDS and IPS.




(Figure 1)

## Timeline

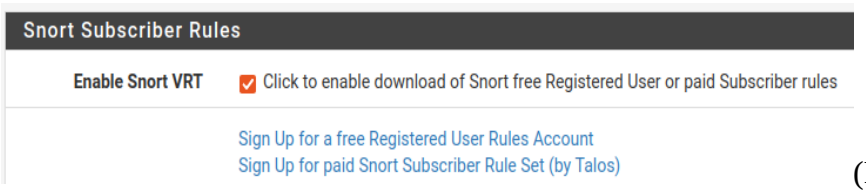
| Phase | Tasks | Start Date | End Date | Duration |
|---|---|---|---|---|
| Planning and Research | Research IDS and IPS options, define objectives, brainstorm experiments | 09/19/2024 | 09/26/2024 | 7 Days |
| Design | Draft architecture and configure IDS/IPS software. Review any network attack rulings. | 09/27/2024 | 09/29/2024 | 3 Days |
| Experiment | Perform common network attacks against the IPS/IPS system. Change the rulings and test security | 09/30/2024 | 10/04/2024 | 5 Days |
| Documentation | Document findings and IDS/IPS information | 10/052024 | 10/06/2024 | 1 Day |

| | | | | |
|---|---|---|---|---|
| Review | Develop final reports | 10/06/2024 | 10/17/2024 | 15 Days |

## Implementation Process

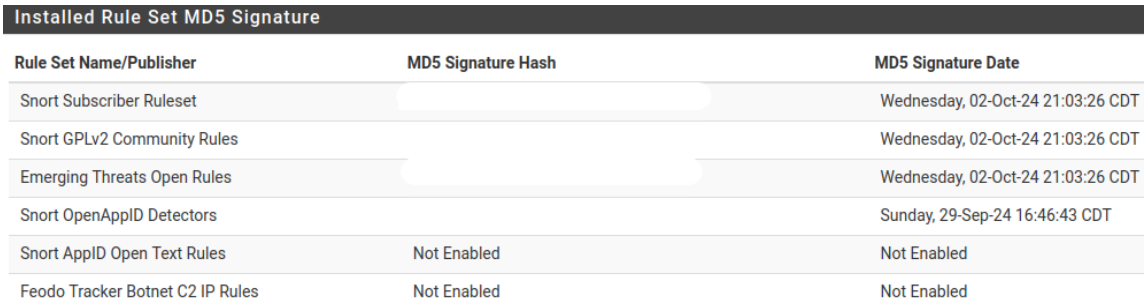
An IDS and IPS can be configured through network threat detection rules. Snort provides community-established network threat detection rules that allow users to upload a file and filter what specific rules are needed (*Snort - Network Intrusion Detection & Prevention System*, n.d.). For example, Snort will provide a list of malware-related rules that will look for network traffic that can potentially be related to trojan or ransomware or block network traffic that is related to scanning a device to provide end-user information regarding open ports or the type of operating system.

To start, the Snort package needed to be installed to configure the IDS and IPS in the network environment. By going to the pfSense Package Manager, Snort was able to be installed. After installation, the first step was to configure the Snort interfaces and configure what network threat detection rules were to be installed. Snort provides a list of pre-configured rules. See Figure 2.




(Figure 2)

The Snort rules are certified rulesets that are distributed free of charge without any Snort Subscriber License restrictions. Finally, the installation process is to install the rules to the Snort IDS and IPS. See Figure 3.




(Figure 3)

The final step in the implementation process was to configure the network threat detection rules for each network interface, the LAN and the WAN. Having different rules configured for each network interface allows users to have strict network detection outside the firewall. After configuring the rules for each interface, the rules were analyzed to see what information was scanned. Figure 4 is a rule that will deny any network packets that are related to a potential denial-of-service (DOS) attack.

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| ✓ | ⚠ | 1 | 2017722 | tcp | $EXTERNAL_NET | any | $HTTP_SERVERS | $HTTP_PORTS | ET DOS Trojan.BlackRev V1.Botnet HTTP Login POST Flood Traffic Inbound |
| ✓ | ⚠ | 1 | 2017918 | udp | any | any | any | 123 | ET DOS Possible NTP DDoS Inbound Frequent Un-Authed MON_LIST Requests IMPL 0x02 |
| ✓ | ⚠ | 1 | 2017920 | udp | $HOME_NET | 123 | $EXTERNAL_NET | any | ET DOS Possible NTP DDoS Multiple MON_LIST Seq 0 Response Spanning Multiple Packets IMPL 0x02 |
| ✓ | ⚠ | 1 | 2017921 | udp | $HOME_NET | 123 | $EXTERNAL_NET | any | ET DOS Possible NTP DDoS Multiple MON_LIST Seq 0 Response Spanning Multiple Packets IMPL 0x03 |
| ✓ | ⚠ | 1 | 2017965 | udp | any | 123 | any | 0:1023 | ET DOS Likely NTP DDoS In Progress MON_LIST Response to Non-Ephemeral Port IMPL 0x02 |
| ✓ | ⚠ | 1 | 2018978 | tcp | $EXTERNAL_NET | any | $HTTP_SERVERS | $HTTP_PORTS | ET DOS HOIC with booster inbound |

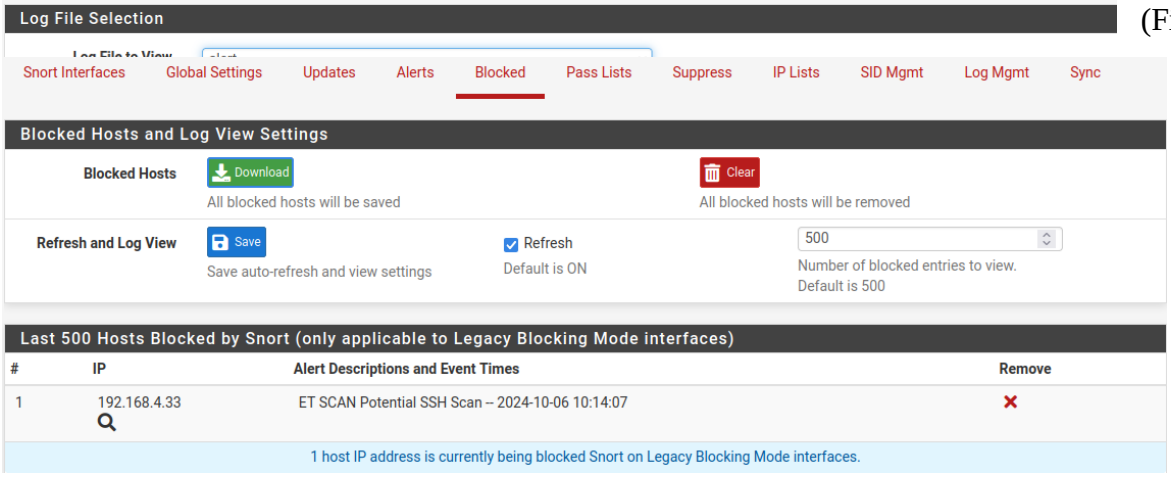
(Figure 4)

## Analysis of Results

The first experiment was conducting a common network attack to test how pfSense relays the information. The common network attacks include port scanning, ping flood attack, and denial-of-service (DOS) attacks. To do this, a Kali Linux VM was configured on a device running outside the firewall. Kali Linux provides a wide range of pre-installed tools for ethical hackers. For example, the tool hping3 provides the ability to send a large amount of network packets to a system through an IP address. The following command was used to execute a network packet flood attack from the Kali Linux VM (Kali, 2024).

```
sudo hping3 -S -p 22 -flood -V 192.168.4.31
```

After running the command, the Snort WAN logs generated the following information in Figure 5. Snort detected that a potential network scan was being conducted. This is an example of the IDS feature in Snort taking into effect. While IDS detects the attack, IPS in this experiment reacted to the network attack and immediately blocked the Kali Linux VM under the Blocked Host list in Figure 6. What this means is that pfSense and Snort will block all network communications from the Kali Linux VM system.




(Figure 5)



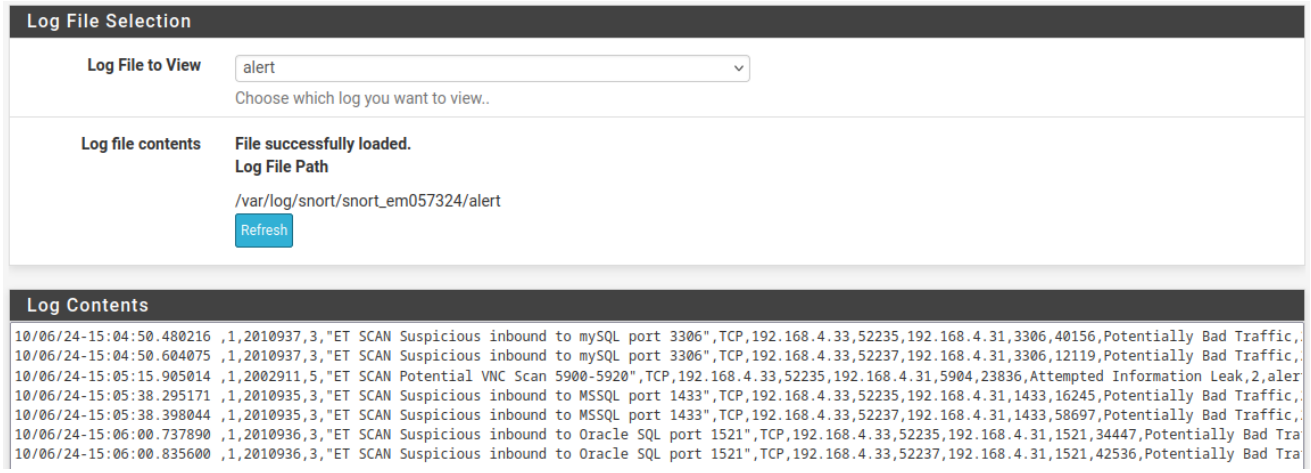

(Figure 6)

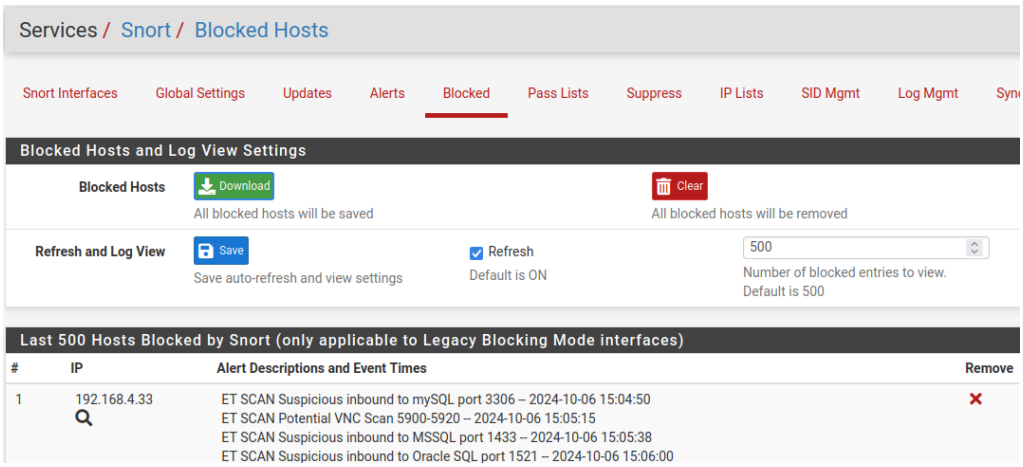
The second experiment focused on another common network using the common network scanner Nmap. “Nmap is short for Network Mapper. It is an open-source Linux command-line tool that is used to scan IP addresses and ports in a network and to detect installed applications” (Shivanandhan, 2020). Malicious actors will use Nmap to find open network ports and possible vulnerabilities on devices. To start, this experiment uses a Kali Linux VM that is configured outside the firewall with an IP address of 192.168.4.33. In the Kalix Linux VM, the following Nmap command was executed.

```
sudo nmap -sS 192.168.4.31
```

In Snort, an alert was generated about a suspicious scan inbound to the firewall IP address from a device. Figure 7 provides an example of the alert Snort generates. Due to the configuration of IPS, the device was added to the Blocked Host list with a description of the alerts and the times when the alerts were generated. See Figure 8 for an example. In a real-life scenario, if the malicious actor were to try and conduct the scan again, they would receive no response from Nmap since all the network traffic contacting Snort is being blocked.

(Figure 7)




(Figure 8)

## Roadblocks and Solutions

After integrating Snort into the pfSense firewall for my research project, I encountered an unexpected roadblock. The Snort system was blocking all network traffic that left or came into the firewall. For example, devices outside the firewall were unable to ping the pfSense IP address and the virtual systems inside the firewall were unable to connect to an external web server. After further analysis, all the rules for both the LAN and WAN environment were enabled, blocking all network traffic. To work around this roadblock, I did a reinstall of pfSense on the virtual software and performed a factory reset. After factory resetting pfSense, all the Snort network rules had to be reconfigured. Overall, I learned from this roadblock is to not enable all the Snort network rules by default. Instead, disable all the rules and manually review what rules to enable (*Packages — IDS / IPS | PfSense Documentation*, n.d.).

During the process of configuring the LAN interface in Snort, I encountered difficulties with the configuration process. Despite following multiple tutorials online and documentation, I was unable to achieve the outcome of having Snort IDS and IPS rules configured for the LAN interface. The issue originates from trying to perform an experiment where I replicate a malicious actor inside a network environment conducting a network scan using Nmap. When executing the Nmap command, Snort would not generate logs and was unable to apply the rules. I explored multiple resources online like YouTube tutorials and online forums; however, none of the suggestions were effective in resolving the issue. In conclusion, I was unable to determine the root cause of the problem and implement Snort in the LAN environment.

## Future Improvements

After analyzing how the experiments were performed and how the network interfaces were configured, one future improvement is to read more documentation on the effects of implementing Snort to pfSense. Once the Snort packages were installed and configured, the network configuration for the VMs changed, making it so that the VMs on the LAN were not able to access web servers outside the firewall. With access to web servers, other kinds of experiments could have been performed to test Snort and its capabilities.

## Conclusion

In conclusion, the objective of the project was to implement IDS and IPS software in the pfSense firewall to understand how network security can be enhanced. The IDS and IPS software came from a third-party software called Snort that was installed in the pfSense firewall. Not only was the project's objective to understand network security when detecting malicious attacks, but it was also to build a better understanding of managing virtual systems and gain more experience in network administration.

The project proved the capabilities Snort provides in a network environment. Snort's IDS and IPS were able to detect malicious attacks and act against the attack right away. In a small office or home network environment, individuals can deny attacks from malicious actors or recognize when a suspicious file, possibly containing malware, is entering the network environment. Another benefit of having an IDS and IPS in the network environment is preventing an individual from connecting to a malicious website and asking for credentials. These kinds of features are not provided in many out-of-the-box firewall applications.

Despite the roadblocks encountered, particularly Snort not functioning properly in the LAN environment, I was still able to learn more about the IDS and IPS rules in the WAN environment and how the rules function. Understanding how to configure an IDS and IPS enhances my job skills and qualifications in network administration when it comes to job searching. Knowing how to manage multiple virtual systems is not only a good skill but a good thing to have to conduct cybersecurity labs in a safe environment that will not affect personal systems. In the future, this will help in the job field because if an employer needs testing done in a safe environment, configuring a virtual environment provides the possibility. In conclusion, the roadblocks and the project provided an opportunity to enhance my cybersecurity skills.

In the future, IDS and IPS could be enhanced with artificial intelligence (AI) and machine learning (ML). By combining AI and ML, detecting and reacting to malicious attacks can become more efficient and more accurate. Some IDS and IPS software, like Snort, rely on other third-party companies to provide rules for patching vulnerabilities and blocking malicious attacks. Since cybersecurity is a fast-growing industry, AI and ML can help keep up with the industry. Tech Target released a book about *The Future of Intrusion Detection and Prevention Systems*. One key point the authors mentioned is the greater use of honeypots in the near future. “A honeypot is a decoy server that looks and acts like a normal server, but that does not run or support normal server functions. The main purpose of deploying honeypots is to observe the behavior of attackers in a safe environment, one in which there (at least in theory) no threat to normal, operating systems” (Target, n.d.). The book goes on to mention how honeypots can be used to learn more about how malicious actors are attacking systems from an external system. With that, system administrators can create custom rules for the IDS and IPS to block malicious attacks. Overall, AI and ML is a growing topic in the cybersecurity industry as a whole and honeypots could potentially be used more effectively to understand what attacks malicious actors use against systems.

The project highlighted the critical role IDS and IPS can provide in a network environment in terms of enhancing network security. IDS provided the capability of detecting malicious network scans in the experiments and the IPS was able to prevent malicious network scans by blocking the host device (*IPS vs IDS: What’s the Difference and Why It Matters | Tech Impact*, 2023). When configuring both systems, individuals can block malicious attacks such as DOS as well as stop the possibility of a malicious file containing malware entering a system’s hard drive. Firewalls can monitor and reroute network traffic; however, do not allow individuals to act against the network traffic if something malicious happens. With IDS and IPS, individuals can enable or build rules to detect and act against things.

## References

CISA. (2023, February 23). *Understanding Firewalls for Home and Small Office Use | CISA*. Cybersecurity and Infrastructure Security Agency CISA; U.S. Department of Homeland Security. https://www.cisa.gov/news-events/news/understanding-firewalls-home-and-small-office-use

*IPS vs IDS: What’s the Difference and Why It Matters | Tech Impact*. (2023, October 26). Techimpact.org. https://techimpact.org/news/ips-vs-ids-whats-difference-and-why-it-matters

Kali. (2024, May 23). *hping3 | Kali Linux Tools*. Kali Linux. https://www.kali.org/tools/hping3/

*Packages — IDS / IPS | pfSense Documentation*. (n.d.). Docs.netgate.com. https://docs.netgate.com/pfsense/en/latest/packages/snort/index.html

Roesch, M., & Roesch, M. (2014, February 25). *Cisco Announces OpenAppID – the Next Open Source "Game Changer" in Cybersecurity*. Cisco Blogs. https://blogs.cisco.com/security/cisco-announces-openappid-the-next-open-source-game-changer-in-cybersecurity

Shivanandhan, M. (2020, October 2). *What is Nmap and How to Use it – A Tutorial for the Greatest Scanning Tool of All Time*. FreeCodeCamp.org. https://www.freecodecamp.org/news/what-is-nmap-and-how-to-use-it-a-tutorial-for-the-greatest-scanning-tool-of-all-time/

*Snort - Network Intrusion Detection & Prevention System*. (n.d.). Www.snort.org. https://www.snort.org/rules_explanation

Sophos. (n.d.). *IPS and IDS | Intrusion Protection and Detection Explained*. SOPHOS. https://www.sophos.com/en-us/cybersecurity-explained/ips-and-ids

St.John, M. (2024, February 28). *Cybersecurity Stats: Facts And Figures You Should Know – Forbes Advisor*. Www.forbes.com. https://www.forbes.com/advisor/education/it-and-tech/cybersecurity-statistics/

Stavroulakis, P., & Stamp, M. (2010). *Handbook of Information and Communication Security* (pp. 177–178). Berlin, Heidelberg Springer Berlin Heidelberg.

Target, T. (n.d.). *CHAPTER 17 The Future of Intrusion Detection and Prevention 345*.