CONTENTS

LIST OF USED ABBREVIATIONS

DPI - Deep Packet Inspection

GFW - Great Firewall of China

HTTPS - Hypertext Transfer Protocol Secure

TCB - TCP Control Block

TCP - Transmission Control Protocol

VPN - Virtual Private Network

INTRODUCTION

In an age when digital communications plays an vital role in modern socienty, ensuring the confidentiality and integrity of network communications has become crucial. The emergence of virtual private networks (VPNs) as a method of securing and privatizing digital communications on the public Internet has been a significant development in this direction. Because of the growing popularity of virtual private networks, governmental agencies and corporate network administrators have increased their efforts to identify and block these encrypted connections. The result has been an arms race between organizations seeking to monitor, control, or censor Internet traffic and VPN service providers competing for the privacy of their users. The implementation of VPN obfuscation, a progressive collection of technologies capable of disguising VPN traffic as routine Internet operations, is central to this dispute. This obfuscation allows unrestricted access to the global digital environment.

VPNs were initially implemented as a simple way to establish secure network connections across the insecure infrastructure of the public Internet. However, the use of VPN technology eventually surpassed the scope of corporate applications. Private users are increasingly adopting virtual private networks (VPNs) for their own benefit, owing to the potential for enhanced privacy, circumvention of surveillance, circumvention of content geo-restrictions, and protection against cyber threats. In response, efforts to undermine the effectiveness of virtual private networks have increased. These initiatives include corporate firewall policies, government censorship initiatives such as China's Great Firewall, and anti-VPN measures implemented by content providers (El-Maghraby et al. 2017).

The proliferation of anti-VPN initiatives has sparked the emergence of a complex area of cybersecurity that uses data packet signatures to distinguish VPN connections. In response, the VPN obfuscation movement has developed a number of countermeasures. As a recent example, we have the Geneva algorithm which employs genetic algorithms to generate dynamic, packet-manipulation-based evasion strategies, providing a new take on DPI evasion (Bock et al. 2019).

Steganography can also be utilized in VPN obfuscation. This method provides a way around DPI systems by embedding VPN traffic inside regular data streams. Steganography enables regular web traffic, like audio or video streams, to conceal VPN packets, making it difficult for DPI tools to discern between encrypted and regular traffic (Kundur & Ahsan 2003). The effectiveness and subtlety of this method are assessed, offering insights into its limitations and useful applications in different network environments.

Another way of obfuscating VPN traffic is through the application of port switching. To avoid detection, a VPN server is configured to use non-standard ports, which can assist in getting around simple DPI systems that keep an eye on popular VPN ports.

Another way of providing VPN obfuscation is the integration of encrypted proxy tunneling software like Shadowsocks and Stunnel. It can be used to make VPN traffic look like HTTPS traffic (Clowwindy et al. n.d.).

All of these methods, excluding port switching, bring with them a trade-off of in connection speed and throughput, though of varying degrees.

The thesis begins with an overview of VPN technology and then describes the various use cases for VPNs as well as the fundamental principles underlying their operation. It then describes the techniques used to detect VPN traffic. The following is a comprehensive examination of various obfuscation methods and their adaptation to evolving VPN traffic detection strategies. The goal of this analysis is to provide insight into VPN detection methodologies and obfuscation techniques through experimentation, qualitative assessment, and technical proficiency in the process developing a methodology by which one can consider selecting an obfuscation method in real-life scenarios.

1. BACKGROUND

1.1. Virtual Private Networks

1.1.1. Definition and Purpose of VPNs

A VPN is a communication system that enables safe and private connections within a public network infrastructure, such as the internet. Its purpose is to facilitate the secure and private transmission of data over a shared network, restricting access to only authorized users within a certain community of interest.(Ferguson & Huston 1998) VPNs ensure anonymity by utilizing techniques such as tunneling and encryption. This allows for the creation of a virtual network that makes use of the existing infrastructure of the public network, while maintaining high levels of security and privacy. A VPN serves the purpose of facilitating safe and confidential communication, which is particularly crucial for the transmission of sensitive data and distant connectivity to private networks.

1.1.2. Types of VPNs

For lone users who must connect to a private network from a distance, **remote access VPNs** are made. This kind is frequently used by people who work from home or telecommute and who need safe remote access to company or personal network resources. The user's device's VPN client software creates a secure connection to a VPN server, which then provides access to the private network. With this configuration, data transported over the internet is guaranteed to be encrypted, protecting its integrity and secrecy.

Site-to-Site VPNs, sometimes referred to as Router-to-Router VPNs, are mostly utilized by big businesses with dispersed offices. The purpose of these VPNs is to link whole networks together. For example, a company connecting its office network in New York to its office network in London might do so by using a Site-to-Site VPN. This sort of VPN can be further divided into two categories: Extranet-Based (which connects to sites outside the organization, such as partners or vendors) and Intranet-Based (which connects to numerous sites within the same organization). (Rathore et al. 2009)

Personal VPN services are sold to individuals by outside companies. These services help people protect their privacy and safety by encrypting their internet data and hiding their IP address from other people, like ISPs. People often use personal VPNs to get around content limits based on location, stay anonymous online, and keep their data safe when they use public Wi-Fi networks.

A **Mobile VPN** maintains a consistent connection as the user moves and changes network connections, making it ideal for professionals who frequently switch between Wi-Fi and cellular data networks. Unlike traditional VPNs, which can drop connections during such transitions, Mo-

bile VPNs ensure uninterrupted connectivity and application session persistence. (Alshalan et al. 2016)

Hardware VPNs offer a dedicated standalone device with dedicated processors to handle VPN functions. This type offers robust security, high performance, and ease of maintenance, making it suitable for large businesses. However, they can be more expensive than software-based solutions and might require more technical expertise for setup and management. A valid example of this may be a pure hardware implementation of WireGuard on FPGA. (Liu et al. 2023)

Secure Sockets Layer (SSL) VPNs provide secure remote access to an organization's internal network and applications. Unlike traditional VPNs that require installing specific client software, SSL VPNs can be accessed using a standard web browser, providing a more flexible solution for users. They are particularly useful for providing access to web applications and services. (Sun 2011)

1.1.3. VPN Protocols and Standards

Different protocols and standards, each with specific functionality and security levels, are at the core of VPN technology. Understanding VPN protocol obfuscation and Deep Packet Inspection evasion requires to have a solid understanding of these protocols:

- **OpenVPN**: An exceptionally configurable and resilient protocol. It is known for its open-source implementation of security and performance technologies, including the OpenSSL encryption library. OpenVPN is a highly adaptable software that supports a wide range of encryption algorithms and can circumvent firewalls. It is an ideal candidate for research into obfuscation and DPI evasion techniques due to its adaptability.

- **IPSec/IKEv2**: Because of its reliability and speed, Internet Protocol Security (IPSec) and Internet Key Exchange version 2 (IKEv2) provide robust security. However, it is less suitable for in-depth obfuscation research because to its intricacy and the need for a third-party client on some platforms.

- **L2TP/IPSec**: Because IPSec and Layer 2 Tunneling Protocol (L2TP) work so well together, many devices employ them together. But because L2TP/IPSec depends on fixed ports, it is less suitable for obfuscation research because it is more prone to DPI and simpler to stop.

- **PPTP**: One of the first VPN methods was Point-to-Point Tunneling Protocol (PPTP). PPTP is not recommended for secure communications because it has known security gaps and encryption that is not very strong. This makes it less relevant for current study into obfuscation and DPI evasion.

- **SSTP**: Secure Socket Tunneling Protocol is well-known for its capacity to circumvent the majority of firewalls. However, because it is Microsoft's proprietary software, it does not have the openness and public inspection that are required for conducting in-depth study on obfuscation.

- **WireGuard**: WireGuard, a more recent addition, provides both simplicity and high-speed performance. Although it shows potential, its relative newness has limited its testing in various circumstances, especially in situations involving obfuscation.

Taking these things into account, OpenVPN becomes the main topic of experimentation into hiding VPN protocols and getting around DPI. Because it is open-source, it can be studied and configured in a lot of detail, which is necessary for studying effective obfuscation methods, in addition OpenVPN is widely used and supports many different types of encryption, it is also a good choice for this study.

1.1.4. Security Mechanisms in VPNs and DPI challenges

VPN's build safe, encrypted tunnels that shield users from all kinds of risks, like as censorship, data theft, and spying. However, more sophisticated Deep Packet Inspection (DPI) methods are posing a threat to VPN efficacy. We examine the main VPN security features below, along with how DPI may compromise their efficacy.

Encryption

- **Function**: Encryption is the most important part of VPN security. It changes data into a coded form that can only be read by someone with the right decoding key. AES (Advanced Encryption Standard) and TLS (Transport Layer Security) are two common types of encryption.

- **DPI challenge**: DPI can look at patterns of encrypted data to find VPN use. DPI can't decrypt the data, but it might be able to slow down or stop encrypted traffic, especially if the encryption protocol is well known and easy to spot.

Tunneling Protocols

- **Function**: OpenVPN, L2TP/IPsec, and IKEv2 are examples of tunneling protocols that are utilized for the purpose of encapsulating and securely transmitting data over the internet networks. In addition to ensuring that data is kept private and unaltered during the transmission process, they define the manner in which packets are transmitted.

- **DPI challenge**: DPI solutions have the capability to detect and classify network traffic originating from widely used tunneling protocols. DPI can identify and impede VPN traffic by analyzing packet headers and sizes, particularly when conventional ports and protocols are employed.

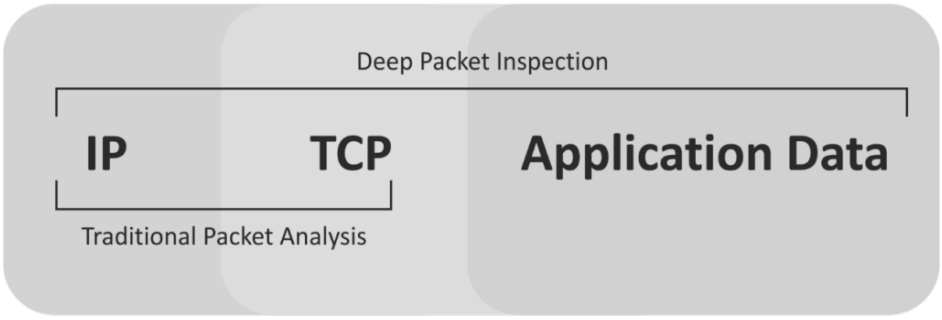
Authentication

- **Function**: VPNs utilize authentication methods to authenticate the identities of users and devices. This procedure frequently includes the use of certificates or credentials to guarantee that only authorized users have access to the VPN.
- **DPI challenge**: DPI doesn't directly affect authentication, but it can prevent users from connecting to VPN servers, which negates the purpose of authentication.

IP Masking

- **Function**: By substituting the IP address of the VPN server for the user's original IP address, VPNs disguise their identity. This aids in preserving anonymity and eschewing limitations based on location.
- **DPI challenge**: It is possible to distinguish between VPN traffic and ordinary traffic using advanced DPI techniques, even when IP masking is in use. This opens the door to the prospect of excluding or restricting VPN traffic depending on its unique properties.

1.2. Deep packet inspection



1.1. Figure. **Inspection area of DPI**

Deep Packet Inspection (DPI) is a form of network packet filtering that examines the data part (and often the header) of a packet as it passes an inspection point. As can be seen on the

figure above, DPI examines the packet's data payload in depth as opposed to conventional packet filtering, which simply looks at the header section, allowing for more complex packet decision-making. DPI can detect, identify, classify, reroute, or prevent packets containing particular data or code payloads thanks to this sophisticated technique.

Due to its great versatility, DPI technology is employed in many different contexts, such as traffic management, censorship, eavesdropping, and network security. It is an essential tool for internet service providers and regulatory bodies since it can identify and manage apps using the network even in the face of encryption and port-hopping schemes.

1.2.1. DPI in Relation to VPNs

DPI is important when it comes to VPNs. In order to protect privacy and get around censorship or geographical limitations, VPNs usually encrypt data. Nonetheless, DPI can be used to examine encrypted packet metadata, including timing, size, destination, and protocol. This makes it possible to identify VPN activity even when the content is encrypted.

Certain DPI techniques can detect the signatures of the encryption protocols used by VPNs, allowing them to differentiate between conventional and VPN traffic. This capability is essential in situations when the use of VPNs is prohibited or closely watched, like in nations with severe internet censorship.

1.2.2. DPI's Challenges for VPNs

VPN technologies have substantial hurdles due to the efficacy of DPI:

- **Detection and Blocking**: If Advanced DPI detects that a VPN is being used, it has the ability to either block or limit data, making the VPN useless for getting around censorship or geographical restrictions.
- **Avoiding Encryption**: Although DPI is unable to decrypt data, it can identify and potentially block VPN services based on educated estimates about the encrypted traffic's nature.
- **Adaptive Countermeasures**: VPN service providers are compelled to constantly modify and introduce fresh methods to avoid being detected by DPI, resulting in a never-ending game of cat and mouse between VPN providers and DPI-using companies.

1.3. OpenVPN identification

A first impression, may lead one to wonder how DPI systems are capable of differentiating between legitimate encrypted data such as secure HTTPS access to a website and VPN traffic. There are several techniques that DPI can use to identify OpenVPN traffic (Pang et al. 2013):

- **Protocol Signature Identification**: OpenVPN traffic can be recognized by DPI systems by looking for distinctive signatures in the packet headers. This is a conventional method that might work less well for OpenVPN because of its encryption and inconsistent port usage.
- **Heuristic analysis**: In this method, traffic patterns are examined to find traits common to VPN traffic, such as regular timing and packet sizes.
- **IP address and port inspection**: OpenVPN typically uses TCP or UDP port 1194, but it also has the option to use variable ports, which reduces the efficacy of this approach.
- **TLS Handshake Analysis**: VPN traffic, including OpenVPN, can be identified by examining the TLS handshake procedure, paying particular attention to the usage of particular cipher suites or TLS versions.
- **Timing and Packet Size**: DPI can identify consistent timing intervals and packet sizes in OpenVPN traffic.
- **Analysis of Encryption Patterns**: Even though OpenVPN encrypts its traffic, some encryption patterns might still be recognizable.
- **Behavioral Analysis**: VPN usage can be inferred by observing general network behavior, such as persistent, long-lasting connections.
- **Sequence Number and Acknowledgment Number Analysis**: Analyzing these numbers in TCP-based OpenVPN traffic can reveal VPN usage patterns.
- **Opcode Analysis in Packets**: the first 10 packets of a communication can be used as statistical characteristics for early detection of OpenVPN tunnels. By using the five bits of opcode in each packet to classify different types of packets.

These sophisticated methods, which focuses on statistical traits and traffic fingerprints, especially the opcode analysis, are very successful in identifying OpenVPN traffic, with low false positive rates and high accuracy (Pang et al. 2013). In this context, the goal of VPN obfuscation can be described as attepting to bypass detection by trying to render as many as possible of the above mentioned techniques, as ineffectual.

1.4. Packet manipulation

One of the most important tools for the design of hardware implementation for the DPI is the finite state machine. This system design, in certain cases, may be exploited by utilizing packet manipulation techniques like packet fragmentation. (AbuHmed et al. 2008) These attacks are even

more relevant in the context of secure network environments, where traditional intrusion detection methods are employed.

In order to avoid detection by DPI systems, the following techniques may be employed:

- **Fragmentation and noise padding**: Harmful data is divided into smaller pieces and mixed in with ”noise” data, which is random or irrelevant data that doesn't interfere with the payload's ability to function. The idea is to trick DPI systems into thinking the fragmented packets are harmless, since these systems usually look for patterns in data to identify malicious activity. The attackers can effectively mask the malicious content, preventing it from being identified and prevented, by fragmenting the data and adding noise. DPI systems may find it especially difficult to defeat this strategy since it requires them to precisely reassemble and examine all of the packets in order to find the harmful content that is hidden—a laborious and resource-intensive procedure. (Yoo & Ahmed 2019)

- **TCB Creation Evasion**: The first SYN packet that opens a TCP connection is manipulated in this tactic. A false state is produced in the TCP Control Block (TCB) of the censoring system by changing specific fields in this packet. Actual data packets are able to evade filtering without being noticed because of this manipulation, which makes the censorship monitor think that the link is either nonexistent or has already ended. (Aceto & Pescapé 2015)

- **Data Reassembly Evasion**: This method takes advantage of flaws in the way censorship systems put back together broken TCP segments. When pieces of data are sent purposely out of order or overlapping, it makes it harder for the filtering system to correctly put these pieces back together and analyze them. This means that private information can get through the system that blocks them without being found or stopped. (Khattak et al. 2013)

- **TCB Teardown Evasion**: This method employs deliberately constructed packets, such as RST (reset), RST/ACK (reset acknowledgment), or FIN (finish), to prematurely end the Transmission Control Block in the censoring system. This premature termination results in the censorship system ignoring any subsequent packets from that connection. Nevertheless, the client-server link remains operational, enabling uninterrupted data flow without any intervention from censorship. (Papadogiannaki & Ioannidis 2021)

- **Resync and Desync**: In this approach, particular packets are transmitted in order to initiate a re-synchronization of the censoring system's transmission control block with the specific TCP connection. Following this, an out-of-window sequence packet is transmitted, which ultimately results in the desynchronization of the censorship system from the link that is

actually being employed. It is possible for data packets to avoid detection and censorship as a result of this difference. (Wang et al. 2017)

These serve as an example of how packet manipulation methods are capable of bypassing DPI. However, it should be noted that these tactics are, by far, not consistently effective and require a fair amount of manual labor to discover the specific packet sendings order, degree of fragmentation or noise required to bypass the DPI, if at all possible.

1.5. Protocol obfuscation

Protocol obfuscation refers to the strategies used to change or conceal the properties of internet protocols in order to escape detection, censorship, or intervention from third parties. This is especially crucial in situations when internet access is tightly controlled or limited. Obfuscation prevents data packets from being detected by filters or detection systems that depend on established protocol signatures or patterns.

1.5.1. General Methods of Protocol Obfuscation

- **Randomization**: In order to hinder automated systems from identifying patterns, random elements are incorporated into the communication protocol. For example, it may be challenging for Deep Packet Inspection (DPI) to recognize the traffic as being part of a particular protocol when packet sizes or timing change. (Kailanya et al. 2022)

- **Protocol Mimicry**: By transforming the traffic of one protocol into that of another, more widely used protocol (such as HTTPS), users are able to evade filters that target particular types of data. A VPN might make its traffic look like normal HTTPS traffic, for instance. (He & Chen 2016)

- **Header Manipulation**: Systems that use header analysis can't figure out what the data bits are if the header information is changed or encrypted. Changing port numbers or encrypting the whole header are two examples of this. (Munshi 2023)

- **Traffic Pattern Masking**: This technique entails modifying the attributes of traffic, such as its quantity, frequency, or trajectory, in order to conceal its true nature. It can be especially efficient against systems that examine traffic patterns over a period of time. (Iacovazzi & Baiocchi 2014)

1.5.2. Focused methods for practical use

Cosidering the wide range of techniques available, the selection was limited to four specific methods for experimentation:

- **Port Switching**: To get around port-based filtering systems, it's easy but effective to change the port of the VPN server. Numerous DPI systems are set up to either watch or restrict traffic on particular ports that are known to be utilized by VPNs. The VPN traffic can get around these filters by periodically switching the server port. This approach is a sensible option for experimentation because it is simple to use and evaluate. (Crawford 2019)

- **Steganography**: Data concealment is the approach of encoding sensitive information within seemingly innocuous data. It entails hiding the communication within seemingly innocuous data streams, such video or picture streams, within the framework of internet protocols. Systems that aren't built to identify such advanced concealment strategies can be severely harmed by this. To learn how well obfuscation holds up against sophisticated DPI systems, experiments will be conducted with steganography. (Handel & Sandford 1996)

- **Geneva Automated Censorship Evasion**: Geneva is an innovative method that makes use of genetic algorithms to automatically look for new ways to circumvent censorship. Geneva has the capacity to keep one step ahead of censoring systems if it continues to evolve and adapt to the environment with ongoing action. In the subject of protocol obfuscation and censorship evasion, testing Geneva provides a chance to investigate the ways in which automated systems might make a contribution to the field. (Bock et al. 2019)

- **Encryption Tunneling (utilizing stunnel or shadowsocks)**: This solution involves encrypting the communication, making it impossible for DPI devices to examine the data packets' contents. Tools such as stunnel and shadowsocks are designed to conceal and obscure traffic, offering an extra degree of security. This strategy is especially useful in circumstances where DPI systems are powerful and capable of performing in-depth analysis. The use of these tools for experimentation can provide useful information about the effectiveness of encryption-based obfuscation approaches. (Zhao et al. 2018)

To summarize, the discipline of protocol obfuscation provides a variety of methods to avoid censorship and surveillance. The four selected approaches for experimentation - port switching, steganography, Geneva automatic censorship evasion, and encrypted tunneling - possess distinct advantages and cater to various aspects of obfuscation. Their choice to conduct experiments with these subjects is warranted due to their significance in modern situations when internet censorship and surveillance are becoming more advanced.

1.6. Case Studies

There exist numerous real-life examples of DPI being deployed and used on a significant scale. Some of the most notorious being:

- **The Great Firewall of China (GFW)**: China's Great Firewall is among the most well-known applications of DPI technology. Among other censoring tools, it uses advanced DPI techniques to identify and restrict VPN connections. VPN service providers frequently need to improve their techniques on a regular basis to avoid being detected.
- **Iran's Internet Censorship**: Especially during political turmoil, Iran's government use DPI to monitor and regulate internet traffic, including the detection and blocking of VPN usage.
- **Corporate Network Management**: By keeping an eye out for unauthorised VPN use that might go around corporate security measures, DPI is utilised in a business context to guarantee security and compliance.
- **Russia's Telegram Ban**: Using DPI, Russia attempted to ban the messaging service Telegram. Telegram, however, circumvented this restriction using a variety of strategies, including VPNs, illustrating the continuous conflict between DPI implementation and VPN evasion approaches.

DPI not only poses a serious threat to VPN technologies but also propels the ongoing development of VPN obfuscation strategies including port switching, network steganography, and sophisticated tunnelling methods like Geneva, Shadowsocks, and Stunnel.

1.7. Early DPI Evasion Methods

Online security and privacy were issues that only a small minority of people worried about in the early days of the internet. But as digital technology developed, so did the techniques for keeping an eye on and filtering online behavior. Deep Packet Inspection started to spread widely and revolutionized the way that institutions and governments could monitor and regulate internet traffic. At first, it was possible to avoid DPI by utilizing networks like Tor or non-standard protocols. However, as DPI technologies advanced, these strategies encountered more difficulties, prompting creative solutions like Tor bridges. (Knapp & Langill 2015)

1.7.1. Rise of DPI and Intial Evasion Tactics

Since Deep packet inspection goes farther than conventional packet filtering by checking the data part (content) of packets as they pass, rather than only the header. Governments and ISPs were able to regulate and control internet traffic with an unparalleled level of precision.

Many people, concerned about their right to privacy and freedom of expression, sought out ways to circumvent DPI as it gained traction. The use of non-standard protocols was one strategy. The original intention of DPI filters was to identify and regulate conventional internet traffic; however, users were able to circumvent these filters by employing less popular or custom-built protocols.

The Tor network was another effective solution against DPI. The Onion Router, or Tor for short, is a system that aims to hide one's online identity. It conceals the user's location and activity from network monitors by rerouting their web traffic through a network of relays. At first, it seemed like a good way to avoid DPI—just connect to Tor and use it. The traffic looked like typical encrypted web traffic.

1.7.2. Advancement of DPI technologies

But things started to change as DPI tools got better. DPI systems got smarter and could look at more protocols and even decrypted traffic trends. They started to notice the signs of Tor activity and the use of non-standard protocols, which made these ways of hiding from surveillance less effective.

Long lists of known IP addresses for Tor relays and entry nodes started to be used. They could stop the Tor network or keep an eye on people who tried to use it with these lists. Advanced DPI systems were also able to recognize Tor's unique traffic patterns, such as its uniform packet sizes and timing. So, using Tor or other obscure methods was no longer a surefire way to get around DPI.(Dingledine & Mathewson 2006)

In response to these developments in DPI, the Tor Project created additional ways to assist users in maintaining their anonymity while still gaining access to the network. The introduction of Tor bridges was one of the those ways.

Tor bridges are alternate Tor network entry points that are not included in the public Tor directory. Because of this obscurity, they are less likely to be blocked or monitored than conventional Tor relays. Users in countries where the internet is strictly censored could use these bridges to connect to the Tor network without drawing attention to themselves.

Furthermore, the Tor Project launched pluggable transports, a technique meant to change the look of Tor traffic, making it difficult for DPI systems to recognize. Obfsproxy (obfuscated proxy), for example, disguises Tor traffic as conventional, innocuous-looking internet traffic. This makes DPI detection much more difficult. (Loshin 2013)

Meek, another pluggable transport, employs a method known as domain fronting to make Tor traffic appear to be connecting with a prominent website such as Google or Amazon. This strategy takes use of the fact that restricting traffic to these important sites would cause enormous

collateral harm, rendering censorship an unfeasible choice.

1.7.3. The Ongoing Cat-and-Mouse Game

The progression of DPI and the accompanying advancements in Tor technology symbolize an ongoing struggle between surveillance parties and proponents of internet privacy. As the technologies for DPI become more sophisticated, the techniques for circumventing them also become increasingly more complex. The implementation of Tor bridges and pluggable transports serving as a valid example of this continuing conflict.

Nevertheless, the usability of these approaches may differ depending on the geographical area and the particular DPI technology being employed. Tor bridges and other obfuscation techniques remain effective for evading detection in certain regions. In some cases, particularly in countries with sufficient government funding for advanced Deep Packet Inspection technologies, even these advanced methods encounter difficulties.(Winter & Lindskog 2012)

1.8. Modern DPI techniques

Recent studies have taken an in-depth examination of the latest techniques employed by the Great Firewall of China to identify and prevent fully encrypted internet traffic. The main techniques are as so (Wu et al. 2023):

- **Heuristic Rules for Exemption**: Based on the assumption that some types of traffic are unlikely to be fully encrypted, the GFW uses a set of heuristic rules to exempt them from blocking. This includes the fraction, position, and maximum contiguous count of ASCII characters; a rough entropy test based on the fraction of bits set; and rules based on common protocol fingerprints.

- **Entropy-Based Blocking**: The GFW denies access to connections based on the entropy of the client's initial TCP payload. As a sign of encrypted data, connections with a specific range of bits set per byte are blocked.

- **Exemption for ASCII Characters**: If the first six bytes are printable, more than half of the bytes are printable, or more than 20 consecutive printable bytes are present, the GFW will not block connections related to ASCII characters in the payload.

- **Protocol Exemptions**: Certain protocols—most notably TLS and HTTP—are specifically spared from blocking because their initial bytes match patterns that these protocols are known to use.

- **Blocking Mechanism**: The client's packets are dropped and never reach the server when the GFW detects encrypted traffic. This prevents further traffic from occurring.
- **Limited Scope and Probabilistic Blocking**: The GFW uses probabilistic blocking, which means that not all connections that satisfy the blocking requirements are obstructed. Blocking is purposefully restricted to particular IP ranges of well-known data centers.
- **UDP Traffic**: Sending UDP datagrams with a random payload does not result in blocking under the new censorship system, which is restricted to TCP.
- **Active Probing System**: Using a payload length-based rule in addition to similar rules, the GFW's active probing system operates in tandem with the traffic analysis system.

These methods highlight the GFW's sophisticated approach to censoring encrypted traffic while allowing standard internet communications. This aids in understanding the censorship mechanism and informs strategies to circumvent these blocks.

1.9. Use of modern DPI techniques

Since modern Deep Packet Inspection techniques have improved over time, effectively combating older methods used to evade such monitoring in the ever-changing world of internet restriction and spying, the capacity of authorities, such as those in charge of the Great Firewall of China, to identify and stop the many strategies that have historically been used to avoid censorship and surveillance has been greatly improved by recent developments in DPI technology.

- **Advanced Signature Detection**: Older ways of getting around security measures often relied on hiding traffic signatures to look like allowed protocols or hide what kind of data was being sent. These days, DPI systems use more complicated algorithms that can better find trends and oddities in data packets. This includes being able to find small changes in how a protocol works, strange headers, and even patterns that could mean steganography or the use of hidden channels.
- **Behavioral Analysis and Anomaly Detection**: Newer DPI methods use behavioral analysis instead of just looking at the static features of traffic. This lets the systems figure out what normal network behavior looks like and spot changes that might be signs of attempts to get around the system. Anomaly detection algorithms can spot strange traffic patterns, like packet sizes that aren't normal, data amounts that don't match up, or unexpected use of protocols.

- **Machine Learning and AI Integration**: Adding Machine Learning (ML) and Artificial Intelligence (AI) to DPI systems is a big step forward in terms of what they can do. These technologies let the systems learn from the information they handle, which makes them better at finding things over time. In real time, AI algorithms can look at huge amounts of data and find complicated patterns and correlations that humans would never be able to see. (Trivedi & Patel 2016)

- **Encrypted Traffic Analysis**: Many old DPI methods stopped working as much as encryption became more popular as a way to protect user privacy. Modern DPI methods, on the other hand, can look at encrypted data without having to decrypt it first. This is done by looking at statistics, time, and other side-channel data, which lets the authorities guess what kind of data is being encrypted.

- **Active Probing and Response Analysis**: These days' DPI systems don't just watch data; they also probe it and respond to it. These systems can figure out what kind of traffic or application is going on by sending specific packets or requests and looking at the replies. This works especially well against protocols that are made to look like other types of data.(Chakraborty et al. 2022)

All of these DPI advacements have resulted in the gradual decline of non-modern DPI evasion methods. The main methods for evading DPI have been proxies and VPNs. The use of known VPN servers or IP addresses, as well as specific traffic patterns and handshake features, can be detected and blocked by modern DPI techniques.

Even anonymity networks like Tor, which aim to let users browse the internet anonymously and securely, are becoming more and more susceptible to DPI attacks. The unique features of Tor traffic, like packet timing and data transfer sizes, can now be detected by DPI systems, even when Tor employs obfuscation methods like pluggable transports.

Traditional methods of evading DPI also include obfuscation and Secure Shell (SSH) tunneling. Even when communication is disguised or contained within other protocols, contemporary DPI systems can still detect SSH based on its unique handshake and session features.

Methods that masquerade as valid protocols (like HTTPS) in order to conceal harmful or restricted traffic are less successful when faced with current DPI. In order to distinguish between real protocol traffic and imposters, sophisticated algorithms examine time, packet size, and other subtle features. (Pandey et al. 2023)

Data packets' genuine nature might be concealed by using traffic fragmentation and padding, which were common techniques in legacy evasion strategies. In order to determine the true type or content of the traffic, modern DPI can reconstruct broken packets by analyzing the padding.

2. EXPERIMENTATION

2.1. Established testbed

2.2. Port switching

2.3. Steganography

2.4. Geneva

2.5. Encryption tunneling

CONCLUSION

REFERENCES

1. AbuHmed, T., Mohaisen, A. & Nyang, D. (2008), 'A survey on deep packet inspection for intrusion detection systems'.

2. Aceto, G. & Pescapé, A. (2015), 'Internet censorship detection: A survey', *Computer Networks* **83**, 381–421.
   URL: *https://www.sciencedirect.com/science/article/pii/S1389128615000948*

3. Alshalan, A., Pisharody, S. & Huang, D. (2016), 'A survey of mobile vpn technologies', *IEEE Communications Surveys & Tutorials* **18**(2), 1177–1196.

4. Bock, K., Hughey, G., Qiang, X. & Levin, D. (2019), Geneva: Evolving censorship evasion strategies, pp. 2199–2214.

5. Chakraborty, R., Jain, H. & Seo, G.-S. (2022), 'A review of active probing-based system identification techniques with applications in power systems', *International Journal of Electrical Power & Energy Systems* **140**, 108008.
   URL: *https://www.sciencedirect.com/science/article/pii/S0142061522000539*

6. Clowwindy, Madeye & Max, L. (n.d.), 'Shadowsocks'.
   URL: *http://www.shadowsocks.org/*

7. Crawford, D. (2019), 'How to hide openvpn traffic – a beginner's guide'.
   URL: *https://proprivacy.com/vpn/guides/how-to-hide-openvpn-traffic-an-introduction*

8. Dingledine, R. & Mathewson, N. (2006), 'Design of a blocking-resistant anonymity system draft'.

9. El-Maghraby, R. T., Abd Elazim, N. M. & Bahaa-Eldin, A. M. (2017), A survey on deep packet inspection, *in* '2017 12th International Conference on Computer Engineering and Systems (ICCES)', pp. 188–197.

10. Ferguson, P. & Huston, G. (1998), What is a vpn ? — part i.
    URL: *https://api.semanticscholar.org/CorpusID:140112970*

11. Handel, T. G. & Sandford, M. T. (1996), Hiding data in the osi network model, *in* R. Anderson, ed., 'Information Hiding', Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 23–38.

12. He, Y. & Chen, M. (2016), 'Protocol mimicry technique and its development', **40**, 1–8.

13. Iacovazzi, A. & Baiocchi, A. (2014), 'Internet traffic privacy enhancement with masking: Optimization and tradeoffs', *Parallel and Distributed Systems, IEEE Transactions on* **25**, 353–362.

14. Kailanya, E., Omamo, A. & Mwadulo, M. (2022), 'Deep packet analysis firewall model', *African Journal of Science, Technology and Social Sciences* **1**(1).
URL: *https://journals.must.ac.ke/index.php/AJSTSS/article/view/62*

15. Khattak, S., Javed, M., Anderson, P. D. & Paxson, V. (2013), Towards illuminating a censorship monitor's model to facilitate evasion, *in* '3rd USENIX Workshop on Free and Open Communications on the Internet (FOCI 13)', USENIX Association, Washington, D.C.
URL: *https://www.usenix.org/conference/foci13/workshop-program/presentation/khattak*

16. Knapp, E. D. & Langill, J. T. (2015), Chapter 3 - industrial cyber security history and trends, *in* E. D. Knapp & J. T. Langill, eds, 'Industrial Network Security (Second Edition)', second edition edn, Syngress, Boston, pp. 41–57.
URL: *https://www.sciencedirect.com/science/article/pii/B9780124201149000034*

17. Kundur, D. & Ahsan, K. (2003), 'Practical internet steganography: Data hiding in ip'.

18. Liu, J., Gao, N., Tu, C., Zhang, Y. & Sun, Y. (2023), A pure hardware design and implementation on fpga of wireguard-based vpn gateway, *in* '2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD)', pp. 1220–1225.

19. Loshin, P. (2013), Chapter 4 - tor relays, bridges, and obfsproxy, *in* P. Loshin, ed., 'Practical Anonymity', Syngress, Boston, pp. 69–80.
URL: *https://www.sciencedirect.com/science/article/pii/B9780124104044000043*

20. Munshi, A. (2023), 'Hybrid detection technique for ip packet header modifications associated with store-and-forward operations', *Applied Sciences* **13**(18).
URL: *https://www.mdpi.com/2076-3417/13/18/10229*

21. Pandey, J., Rai, S. & R, S. (2023), 'Assessment of deep packet inspection system of network traffic and anomaly detection', *International Journal of Scientific Research in Science, Engineering and Technology* pp. 680–688.

22. Pang, Y., Jin, S., Li, S., Li, J. & Ren, H. (2013), Openvpn traffic identification using traffic fingerprints and statistical characteristics, *in* Y. Yuan, X. Wu & Y. Lu, eds,

‘Trustworthy Computing and Services’, Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 443–449.

23. Papadogiannaki, E. & Ioannidis, S. (2021), ‘Acceleration of intrusion detection in encrypted network traffic using heterogeneous hardware’, *Sensors* **21**(4).
URL: *https://www.mdpi.com/1424-8220/21/4/1140*

24. Rathore, M. S., Razzaq, A., Hidell, M. & Sjödin, P. (2009), Site-to-site vpn technologies : A survey.
URL: *https://api.semanticscholar.org/CorpusID:35973654*

25. Sun, S. H. (2011), The advantages and the implementation of ssl vpn, *in* ‘2011 IEEE 2nd International Conference on Software Engineering and Service Science’, pp. 548–551.

26. Trivedi, U. & Patel, M. (2016), A fully automated deep packet inspection verification system with machine learning, *in* ‘2016 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)’, pp. 1–6.

27. Wang, Z., Cao, Y., Qian, Z., Song, C. & Krishnamurthy, S. V. (2017), Your state is not mine: A closer look at evading stateful internet censorship, *in* ‘Proceedings of the 2017 Internet Measurement Conference’, IMC ’17, Association for Computing Machinery, New York, NY, USA, p. 114–127.
URL: *https://doi.org/10.1145/3131365.3131374*

28. Winter, P. & Lindskog, S. (2012), ‘How the great firewall of china is blocking tor’.

29. Wu, M., Sippe, J., Sivakumar, D., Burg, J., Anderson, P., Wang, X., Bock, K., Houmansadr, A., Levin, D. & Wustrow, E. (2023), How the great firewall of china detects and blocks fully encrypted traffic, *in* ‘32nd USENIX Security Symposium (USENIX Security 23)’, USENIX Association, Anaheim, CA, pp. 2653–2670.
URL: *https://www.usenix.org/conference/usenixsecurity23/presentation/wu-mingshi*

30. Yoo, H. & Ahmed, I. (2019), Control logic injection attacks on industrial control systems.

31. Zhao, Y., Ma, X., Li, J., Yu, S. & Li, W. (2018), Revisiting website fingerprinting attacks in real-world scenarios: A case study of shadowsocks, *in* M. H. Au, S. M. Yiu, J. Li, X. Luo, C. Wang, A. Castiglione & K. Kluczniak, eds, ‘Network and System Security’, Springer International Publishing, Cham, pp. 319–336.