Security in the TCP/IP Protocol Suite

Marcos Luchetti
01194213
School of Cybersecurity, Old Dominion University
Cybersecurity Bachelor of Science
CYSE 450: Ethical Hacking and Penetration Testing
Dr. Rashid Khan
April 5, 2022

Security in the TCP/IP Protocol Suite

ABSTRACT

The ever-increasing presence of the Internet in our everyday lives has introduced security concerns in the framework of computer networking. This framework, often referred to as the OSI model, has seven layers, with each layer being composed of several protocols. For instance, the TCP/IP protocol suite plays a major role in the transmission of data over the internet, and it operates at different layers of the OSI model. Unfortunately, the TCP/IP protocol suite contains many exploitable vulnerabilities, prompting the development of additional security protocols. Security protocols like IPsec and HTTP over SSL/TLS (HTTPS) have enhanced network security and Web communications, adding several different layers of security to the communications process. It is critical that developers understand security in the TCP/IP protocol suite so that they can design systems and/or applications that maintain confidentiality, integrity, and availability while communicating over the network.

Keywords: computer networking, security, OSI model, TCP/IP protocol suite, IPsec

Security in the TCP/IP Protocol Suite

INTRODUCTION

A computer network is essentially a web of host machines, or computers, that communicate with each other. Once they are connected with each other, they can share information and other resources. Much of this information is sensitive, and the security of these systems and protocols is not perfect. Computer networks adopted the TCP/IP protocol suite, which was created in 1980, to handle communications between computers. Times were simpler back then, and there was not as much cause for concern for making these protocols secure. The developers prioritized other functions, since they could not have imagined how much more vast and advanced the Internet would become. This is why the TCP/IP protocol suite is so vulnerable to various attacks.

The TCP/IP protocol suite functions at different layers of the OSI model, referenced below:

Layer	Function	Example
Application (7)	Services that are used with end user applications	SMTP,
Presentation (6)	Formats the data so that it can be viewed by the user Encrypt and decrypt	JPG, GIF, HTTPS, SSL, TLS
Session (5)	Establishes/ends connections between two hosts	NetBIOS, PPTP
Transport (4)	Responsible for the transport protocol and error handling	TCP, UDP
Network (3)	Reads the IP address form the packet.	Routers, Layer 3 Switches
Data Link (2)	Reads the MAC address from the data packet	Switches
Physical (1)	Send data on to the physical wire.	Hubs, NICS, Cable

Fig. 1. The OSI Model by Geekinfo

Security in the TCP/IP Protocol Suite

RELATION ACROSS LAYERS

The functions of the TCP/IP protocol suite include the following: Address Resolution Protocol (ARP) and Ethernet, which function at the Physical and Data Link layers (L1 and L2); the Internet Protocol (IP), and Internet Control Message Protocol (ICMP) which work at the Network layer (L3); the Transmission Control Protocol (TCP) and User Datagram Protocol (UDP), working at the Transport layer (L4), with TCP also operating at the Session layer (L5). There are also protocols that work at the Presentation and Application layers (L6 and L7), which include Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP), File Transfer Protocol (FTP), and more.

The **Session, Presentation and Application layers** are similar in that they are concerned with human interaction and how software applications are implemented (Alotaibi et al., 2017, p. 97). In the older TCP/IP model of network communication, these three layers are grouped together into one, called the Application layer. In regards to computer networking, these layers are concerned with providing network services to applications (p. 97). These layers include protocols such as HTTP, SMTP, FTP, etc. Each protocol serves an important purpose in the transmission of data over the Internet (Chapple & Solomon, 2004, p. 136), with application protocols making sure that data can be moved between hosts, and remote connection is intuitive (Alotaibi et al., 2017, p. 97).

The **Transport layer** is responsible for the flow of data between two hosts, usually a client and server. TCP and UDP are the most commonly used transport protocols on TCP/IP networks, with the former providing a reliable, connection-oriented mechanism for process-to-process communication (Chapple & Solomon, 2004, p. 139), while the latter serves as a connectionless companion to TCP. TCP is used by applications, which need reliability in order to work properly. Applications may use HTTP, SMTP, or FTP to transfer files and other data over the network—all of these rely on TCP to transport the data without loss. UDP is used in those situations where losing data is acceptable, such as when streaming music, video, or transmitting small amounts of data. This helps to relieve TCP of processing even more data, prioritizing the more important data for reliable transmission.

The **Network layer** consists of protocols that help in delivering packets to their set destinations. For instance, IP is responsible for the routing functions of traffic on a TCP/IP network (p. 137). This includes activities such as addressing, routing and transmitting packets

over the network (Alotaibi et al., 2017, p. 98). ICMP provides administrative services to TCP/IP networks, often used to transmit control messages between hosts. These control messages can be used to report on error conditions such as dropped packets, connectivity failure, as well as for redirecting packets via another router. Ethernet works at the Physical layer, sending data in bits. ARP works at the Data Link layer, linking and translating Internet layer addresses to Network Interface layer addresses, such as a MAC address (p. 98).

THREATS AND VULNERABILITIES

Unfortunately, there exist numerous threats that have the potential to exploit many vulnerabilities found in the TCP/IP protocol suite. For instance, HTTP, the default communication protocol used by all web browsers, transmits data in plaintext form. This opens the door to multiple attack vectors. One of these is **session hijacking**, where the attacker steals an HTTP session after capturing packets with a packet sniffer. The attacker obtains the session ID of the conversation between the client and server, bypassing the authentication measures, allowing the attacker to act as the target host and access the user's information. Session hijacking can also be done using an exploit called **cross-site scripting (XSS)**, in which the hacker inserts malicious code in a web application or browser to be executed on the target's client machine. Cookie poisoning is another known attack method of HTTP, in which the attacker steals or modifies a cookie (a user's saved credentials and other information on a given website) from a target user's machine to obtain sensitive information, such as a password and/or username which can be used to access the target user's accounts on certain websites.

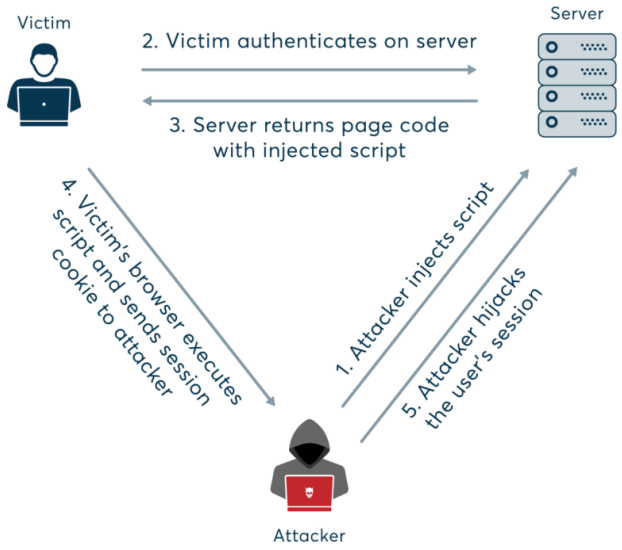

Fig. 2. Session hijacking by Bipin Choudhary

Security in the TCP/IP Protocol Suite

TCP is also prone to many exploitable vulnerabilities. In order for two machines to begin communicating with each other, they must first engage in a "three-way handshake." During this activity, the client sends the server a "SYN" packet, or a synchronization request, to which the server responds with an acknowledgement of the request, or "ACK" packet. The server also sends a SYN packet to synchronize sequence numbers from their end. Lastly, the client responds with an ACK packet, and the two machines can then engage in a reliable, connection-oriented exchange (Basta et al., 2013, p. 96). It is during this "three-way handshake" where things can get problematic for the client/server. A hacker can use a **TCP "SYN" attack** to send many SYN requests to the server, overwhelming it so much that it cannot respond to the client (Alotaibi et al., 2017, p. 100). An attacker could also exploit a TCP connection by way of **TCP sequence number prediction**. This method involves guessing the sequence number of a packet transmitted between a client and server, and then counterfeiting a packet to pretend to be an authorized person after spoofing the IP victim (p. 101).

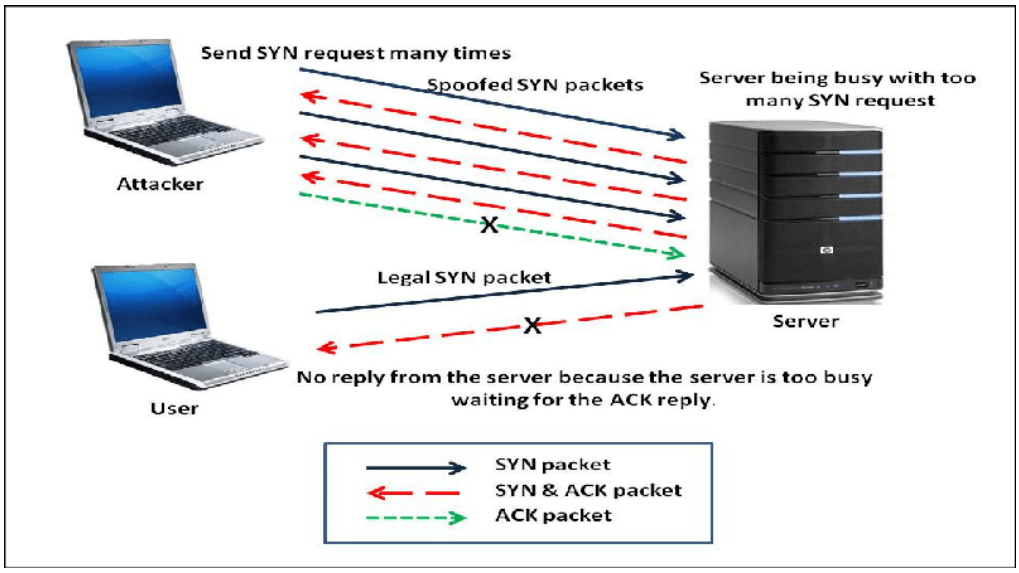

Fig. 3. TCP SYN flood attack by ResearchGate

The Internet Protocol is open to many types of attacks as well. **IP spoofing attacks** are used to hide the identity of the IP sender. The attacker generates the wrong source IP address to fool servers. This method can be used to overload targets with traffic from multiple spoofed addresses, rendering them unusable. This can be done to take down servers that host websites and other resources, but it can also be targeted at clients in a network. The clients get flooded with traffic, hindering user productivity. This type of attack is referred to as a **denial of service**

(DoS) attack, and it is one of the most infamous attacks on the Internet. A **distributed denial of service (DDoS) attack**, however, is done using multiple computers and Internet connections, usually in a botnet, whereas DoS uses just one host and one Internet connection. This attack can cause companies to have unavailability of services, hurting their revenue and reputation with customers. Moreover, ICMP is prone to **Smurf attacks**, in which an attacker spoofs an ICMP packet's source address to send a broadcast to all the computers on a network. If a network's firewall accepts ICMP packets, then this broadcast will heavily congest the victim's network, seriously inhibiting productivity. There exists a similar type of attack called a **fraggle attack**, but it instead affects machines over UDP.

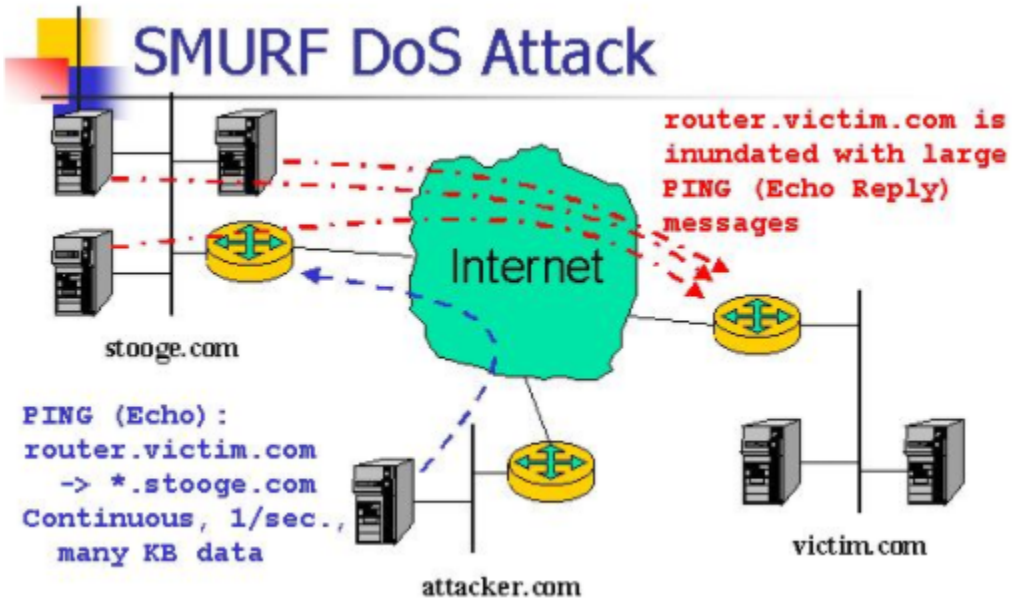

Fig. 4. Smurf DoS attack by Gary Kessler

RESULTS

To address these vulnerabilities, security developers have implemented additional security systems and protocols to counter online threats. One such protocol is called Internet Protocol Security (IPsec), which implements several layers of security into the communications process (Chapple & Solomon, 2004, p. 153). This is accomplished through the use of the Internet Security Association and Key Management Protocol (ISAKMP), the Authentication Header (AH), and the Encapsulating Security Payload (ESP). Other protocols that secure Web connections include the Secure Sockets Layer (SSL), which has evolved into Transport Layer Security (TLS), and Secure-HTTP (HTTP-S) (p. 153).

Security in the TCP/IP Protocol Suite

FINDINGS

IPsec addresses the need for three security requirements: Virtual Private Networks (VPNs), application-level security, and routing security. IPsec is most often used in VPNs, as it is not an adequate solution for application level security or routing security and must be combined with other security measures in order to be most effective. There are two modes of operation in IPsec: **Transport Mode** and **Tunnel Mode**. In Transport Mode, the source and destination hosts are responsible for performing all cryptographic operations, and encrypted data is transmitted through a tunnel that is created with the Layer 2 Tunneling Protocol (Thomas & Elbirt, 2004, p. 39). The data, now in ciphertext, is created by the source host and retrieved by the destination host, establishing security of communications from end-to-end. In Tunnel Mode, there are special gateways that perform cryptographic processing for the source and destination hosts. Tunnels are created between gateways, establishing gateway-to-gateway security (p. 39). In either of these modes, all gateways must be able to verify the integrity of a packet and to authenticate it at both ends, discarding invalid packets.

IPsec makes use of two types of data packet encodings: the **Authentication Header (AH)** and the **Encapsulating Security Payload (ESP)**, which give data network-level security. The AH data packet encoding provides authenticity and integrity of the packet by assigning keyed hash functions, or MACs, which produce checksums created by applying a keyed authentication scheme to a message (p. 39). The AH also prevents illegal modification and is able to provide anti-replay security. AH is also important in that it can establish security for multiple hosts, gateways, or multiple hosts and gateways having implemented AH. The ESP header provides encryption, data encapsulation, and data confidentiality (p. 40). The ESP header provides data encryption by using symmetric key algorithms.

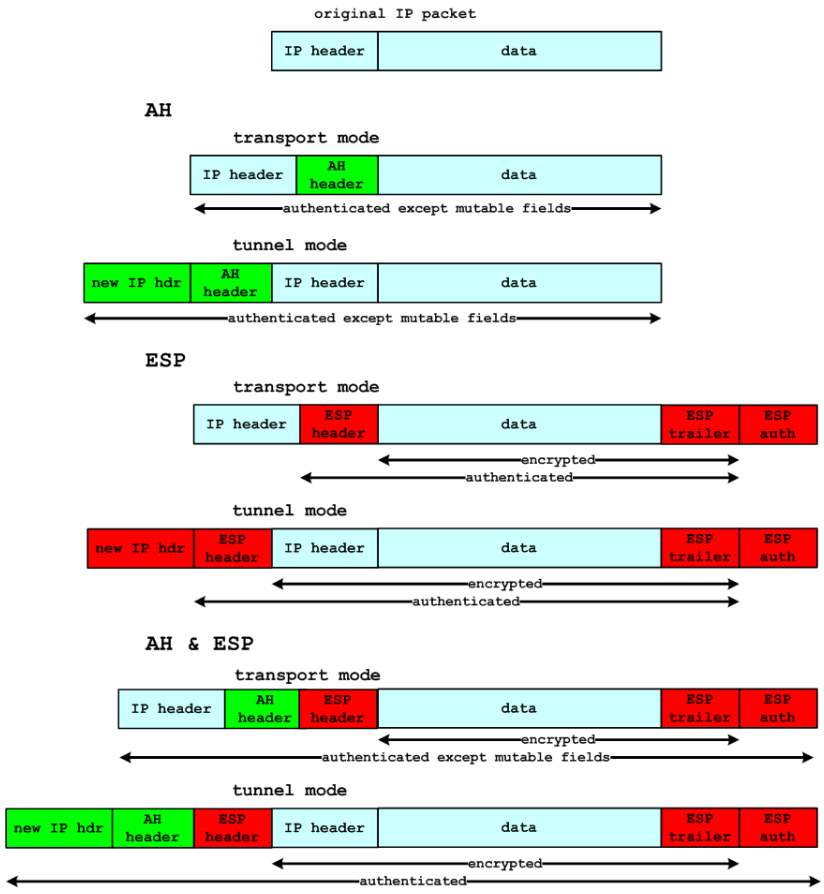

Fig. 5. IPsec AH & ESP header encapsulation by Author Unknown

Another component of IPsec is the **Security Parameter Index (SPI)**, which is an identification tag added to a header of a packet as it makes its way through the many tunnels and gateways. The SPI is essentially aids in checking the integrity of a packet, specifying which algorithms and keys were used by the last system to view the packet (p. 40). **Security Association (SA)** is also an important part of IPsec. This uses the SPI tag that is transferred in the AH and ESP to specify the SA that was used for the packet. This field also includes an IP destination address, which indicates the endpoint. In addition, an **SA database (SAdb)** is used to store all SAs that are used (p. 40). The SAdb makes use of a **Security Policy (SP)** to determine how a router should process a packet—dropping the entire packet, dropping the SA, or replacing it with a different SA. The **SP Database** is used to store all of the SPs.

Implementing IPsec involves having a SAdb with management routines, the IPsec protocol engine itself, as well as cryptographic transforms and algorithms (p. 40). System policies, such as data control policies, are enforced through **SAdb management routines**. For instance, a policy could state that the system will always accept incoming and outgoing IPsec

protected packets as long as they have a relevant SA. Communication with non-IPsec protected systems can be possible by assigning a "NULL SA" as a placeholder for the SAdb. With this policy, the system will only communicate with systems that have SAs or NULL SAs. The **IPsec protocol engine** is made up of two distinct elements: the one for incoming data, and the one for outgoing data. These elements "control the application of the input and output cryptographic algorithms to the data" (p. 40), as well as verifying the integrity of authentication data, changing the IP headers if need be. The **cryptographic algorithms** go hand-in-hand with cryptographic transforms, as they are usually developed in tandem with one another. These algorithms are math-oriented functions and are developed by a third party—IPsec only adopts these algorithms, as they are not a part of IPsec implementation. IPsec can be implemented at the Operating System (OS) level, which yields many security advantages. This enables software developers to program applications with IPsec implementation. In addition, if the OS requires applications to have IPsec, it will significantly reduce the likelihood that malware will be able to exploit any vulnerabilities in the IPsec protocol. Implementing IPsec at OS-level also implies that any issues with IPsec can be patched through a single update to the OS, rather than having to update each IPsec application individually. However, there are some instances where IPsec applications exist independently from the OS which does not have IPsec implemented. These custom applications can cause implementation problems, since they need to be patched regularly apart from the OS (p. 41).

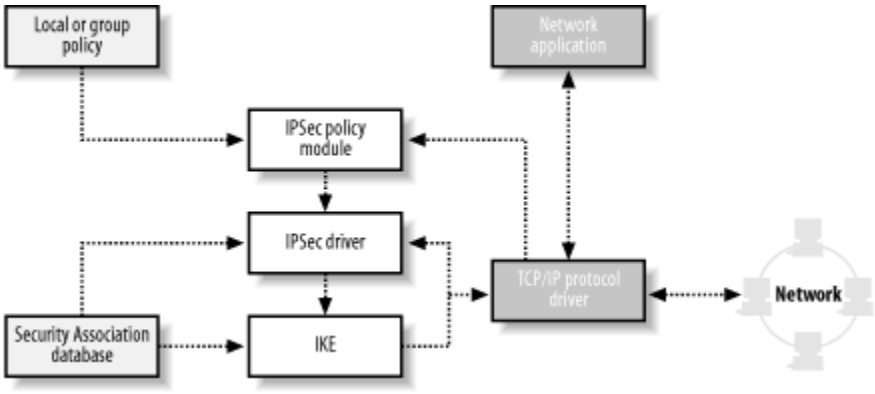

Fig. 6. IPsec component interaction by eTutorials

Security in the TCP/IP Protocol Suite

HTTPS is a secure communication protocol for web traffic. It is widely used because it offers mutual authentication and establishes a secure channel for providing end-to-end encrypted communication over the Internet (Hu et al., 2021, p. 25). It prevents Man-in-the-Middle (MitM) attacks by "verifying digital certificates issued to each HTTPS domain" (p. 25). Trusted Certificate Authorities (CAs) issue each HTTPS domain with its own valid certificate which it uses to authenticate itself to clients. HTTPS helps to prevent eavesdropping and active attacks, but it is not a bulletproof solution. Browsers and HTTPS domains must agree on the SSL/TLS version, encryption methods, and other security parameters. **SSL/TLS** is a standard protocol for authentication, data confidentiality, and message integrity (p. 28) for a TCP/IP connection, with TLS being more secure. It is used in the HTTPS protocol for adding security to HTTP. Some browsers still support HTTP 2.0, but only over TLS, while more than 50% of HTTP domains were reported (2017) to have switched to HTTPS (p. 28), and this number is only going to keep increasing as there is a much greater demand for secure communications. The SSL/TLS infrastructure has been steadily growing recently, and it has become an important component of Web communication.

Fig. 7. HTTPS Communication Data Encryption by Dr. Heron Yang

When deploying an HTTPS domain, system administrators must consider using a strong private key for preventing impersonation social engineering attacks. In addition, they should obtain a certificate from a reliable CA. Removing vulnerable cipher suites (such as SSL 3.0) and enabling the latest TLS protocol is also most advisable. Lastly, system administrators should use a comprehensive SSL/TLS assessment tool when first setting up an HTTPS domain to verify the configuration (p. 47).

CONCLUSION

Computer networks around the world handle billions of data transactions every single day. Some of these networks contain servers, which host services for other clients on the Internet. Clients connect to these servers via the three-way TCP/IP handshake before transmitting data to and from the server. The TCP/IP protocol suite has been proven to be quite effective and plays a role in every layer of the OSI model, however, in terms of security it is quite limited. This has introduced several threats and vulnerabilities that need solutions. TCP session hijacking, IP spoofing, and DDoS attacks are a few examples of how the TCP/IP protocol suite can be exploited. Security engineers have designed several systems and protocols designed to mitigate these threats, ensuring the confidentiality, integrity, and availability of information passing over these networks.

IPsec was designed to incorporate multiple security services, such as authentication, integrity, confidentiality, encryption, and nonrepudiation. It has become one of the best security systems available for securing TCP/IP connections. It still has much room for improvement, but its current implementation has produced significant results in preventing cyber attacks. SSL/TLS over HTTP, or HTTPS, was designed to address the vulnerabilities of HTTP, which is used by many web browsers. By using digital certificates issued to domains by trusted CAs, HTTPS allows for encrypted communication between mutually authenticated parties from end-to-end over the Internet. This protocol uses SSL/TLS as the standard protocol for authentication, data confidentiality, and message integrity for a TCP/IP connection. The implementation of HTTPS continues to increase, as more people are made aware of the need for securing Web communications. It is critical that developers employ these security protocols when designing systems or applications, being mindful of the TCP/IP protocol suite's security shortcomings—doing so helps in establishing a strong resilience against malicious hackers.

Security in the TCP/IP Protocol Suite

REFERENCES

Alotaibi, A. M., Alrashidi, B. F., Naz, S., & Parveen, Z. (2017). Security issues in Protocols of TCP/IP Model at Layers Level. *International Journal of Computer Networks and Communications Security*, *5*(5), 96-104. https://www.ijcncs.org/published/volume5/issue5/p2_5-5.pdf

Anomaly Detection of IP Header Threats - Scientific Figure on ResearchGate. Available from: https://www.researchgate.net/figure/TCP-SYN-Flood-Attack_fig2_49966079

Bae, Y., Kim, I., & Hwang, S. O. (2018). An efficient detection of TCP Syn flood attacks with spoofed IP addresses. *Journal of Intelligent & Fuzzy Systems*, *35*(6), 5983–5991. https://doi-org.proxy.lib.odu.edu/10.3233/JIFS-169839

Barker, E., Dang, Q., Frankel, S., Scarfone, K. and Wouters, P. (2020). *Guide to IPsec VPNs*. National Institute of Standards and Technology. https://doi.org/10.6028/NIST.SP.800-77r1

Bouyeddou, B., Harrou, F., Kadri, B., & Sun, Y. (2021). Detecting network cyber-attacks using an integrated statistical approach. *Cluster Computing*, *24*(2), 1435–1453. https://doi-org.proxy.lib.odu.edu/10.1007/s10586-020-03203-1

Basta, A., Basta, N., & Mary Brown, C. (2013). *Computer security and penetration testing* (2nd ed.). Cengage Learning.

Chapple, M., & Solomon, M. G. (2004). Chapter 6 securing TCP/IP. In *Information security illuminated*. Jones & Bartlett Learning. https://samples.jblearning.com/076372677X/chapple06.pdf

Choudhary, B. Session hijacking. http://vednam.com/what-is-session-hijacking-what-are-the-methods/

D. V. Bhatt, S. Schulze and G. P. Hancke, "Secure Internet access to gateway using secure socket layer," in *IEEE Transactions on Instrumentation and Measurement*, vol. 55, no. 3, pp. 793-800, June 2006, doi: 10.1109/TIM.2005.862009.

Hu, Q., Asghar, M. R., & Brownlee, N. (2021). A large-scale analysis of HTTPS deployments: Challenges, solutions, and recommendations. *Journal of Computer Security*, *29*(1), 25–50. https://doi-org.proxy.lib.odu.edu/10.3233/JCS-200070

Yang, H. HTTPS Communication Data Encryption. http://www.herongyang.com/PKI/HTTPS-Communication-Data-Encryption.html

IPsec component interaction. eTutorials.org. http://etutorials.org/Server+Administration/securing+windows+server+2003/Chapter+8.+IP+Security/8.2+How+Does+IPSec+Work/

Kessler, G. Smurf DoS Attack. http://www.webtorials.com/main/vici/briefing/denial-of-service/slides/sld007.htm

Oppliger, R. (1997). Internet Security: FIREWALLS and BEYOND. *Communications of the ACM*, *40*(5), 92–102. https://doi-org.proxy.lib.odu.edu/10.1145/253769.253802

The OSI Model Layers. Geekinfo. https://geekinfozangi.blogspot.com/2016/05/the-osi-model-layers.html

Thomas, J., & Elbirt, A. J. (2004). *Understanding Internet Protocol Security*. Information Systems Security, *13*(4), 39–43. https://doi-org.proxy.lib.odu.edu/10.1201/1086/44640.13.4.20040901/83731.6