Network Security

Carl B. Wade

Southern New Hampshire University

IT-640

July 29, 2016

Abstract

Network Security is one of the most important topics in networking. Everyday people send packets of information over networks both in secure and insecure ways. As integration of the internet in our everyday lives increases; so does the rise in cybercrimes. Hackers are finding more and more ways to steal personal information as it is being transmitted over a network. This information may include private emails, banking information, credit card numbers, social security numbers, and private documents. If the network is not properly secured, then the hacker can access the information freely. However, if the user uses one or more methods of security the likelihood of the information getting stolen is greatly decreased. Some of these methods include End-Point Authentication, Secure TCP protocols (like SSL), and Encryption of data packets. Even though there is no 100% guarantee way to secure a network to prevent attacks, some security is better than nothing. However, in most cases, modern use of network security is enough to stop a majority of cyber-attacks.

Brief History of Network Security

There were several key events that contributed to the evolution of the computer and creation of network security. During World War II communications was secured by encrypting text. Secured communications were essential to winning the war. Enigma machines were used during this time to encrypt the text. During the late 1960's ARPANet was created allowing scientists to share data and access remote computers. During this time e-mail was the most popular application. In the 1970's Telnet protocol was created to allow networks to be used by the public. Then in the 1980's cybercrimes began with many groups stealing top-secret information from military computers. To help combat these cybercrimes The Computer Fraud and Abuse Act of 1986 was created. Another concern that was made known in the 1980's was the creation of computer worms. Robert Morris was responsible for creating and unleashing the Morris Worm to over 6,000 vulnerable computers via the Internet. As a result, the Computer Emergency Response Team (CERT) was created to alert users of network security issues. The need for modern network security was the result of cybercrimes committed by Kevin Mitnick in the 1990's. Kevin Mitnick is responsible for committing the largest cybercrime in U.S. history. He was responsible for losses totaling up to eighty million dollars in U.S. intellectual property and source code from a variety of companies. As a result, information security became very important to networking. "Internet protocols in the past were not developed to secure themselves. Within the TCP/IP communication stack, security protocols are not implemented. This leaves the internet open to attacks. Modern developments in the internet architecture have made communication more secure." (Daya, n.d.) However, since the time of Kevin Mitnick, cybercrimes have been on the rise. Since the internet is evolving and being used to transmit sensitive data, network security is needed now more than ever.

What is Network Security?

Before we start discussing various security methods used in networks, we must know what network security entails and what it protects. In general terms, security is a process of protecting something from an attack. In terms of networking, this would be protecting a server, or the network itself, from an attack made by a hacker. The hacker may have intentions to either steal information or cause damage to the computer/network. Kizza explains security involving "the security of all its resources such as its physical hardware components such as readers, printers, the CPU, the monitors, and others. In addition to its physical resources, it also stores non-physical resources such as data and information that need to be protected. In a distributed computer system such as a network, the protection covers physical and non-physical resources that make up the network including communication channels and connectors like modems, bridges, switches, and servers, as well as the files stored on those servers." (Kizza, 2005, p. 49) The common theme in all of these examples is that security means preventing any unauthorized access to any system which would result in theft, alteration, or physical damage to the resources or computers. According to Kizza there are three elements involved in network security. These elements include: confidentiality, integrity, and availability. Kizza defines confidentiality as a way “to prevent unauthorized disclosure of information to third parties. This includes the disclosure of information about resources.” (Kizza, 2005, p. 49) Integrity is “to prevent unauthorized modification of resources and maintain the status quo. It includes the integrity of system resources, information, and personnel.” (Kizza, 2005, p. 49) Availability is “to prevent unauthorized withholding of system resources from those who need them when they need them.” (Kizza, 2005, p. 49) Keeping this in mind we can now look into ways networks can be secured to achieve the goal of confidentiality, integrity, and availability to prevent unauthorized access.

End-Point Authentication

One line of defense against cyber criminals is end-point authentication. End-point authentication is the process of one node, or user, proving its identity to another node, or user, over a computer network. For example, people using a username and password to sign in to a website. Unlike the way humans authenticate each other, network authentication must be done solely on the basis of messages and data exchanged as part of an authentication protocol. "Typically, an authentication protocol would run before the two communicating parties run some other protocol (for example, a reliable data transfer protocol, a routing information exchange protocol, or an e-mail protocol). The authentication protocol first establishes the identities of the parties to each other's satisfaction; only after authentication do the parties get down to the work at hand." (Kurose, 2013, p. 700) With this in mind, computer networks cannot just simply send messages to each other as a way of authentication. Kurose points out that with such a protocol the receiving node cannot truly know if the message is really being sent by the sender or an intruder, even if the message contains the IP address of the sender or even a password. Figure 1 shows us how such a protocol will result in no real security added. The next step in trying to secure the network is to encrypt the message sent for authentication. However, simply encrypting the message does not solve the problem and opens the receiving node to what is known as a playback attack. As Kurose explains by using the Alice, Bob, and Trudy example, "Trudy need only eavesdrop on Alice's communication, record the encrypted version of the password, and play back the encrypted version of the password to Bob to pretend that she is Alice. The use of an encrypted password in ap3.1 doesn't make the situation manifestly different from that of protocol ap3.0." (Kurose, 2013, p. 703) Now if the two nodes use a combination and slight variation of sending messages, passwords, and encryption to each other then we can have secure

authentication. The sending node will send a message identifying itself to the receiver. The receiver will in turn send a nonce, or a once in a lifetime number, to the sender. The sender will then encrypt the nonce using a predetermined symmetric secret key and send the encrypted nonce back to the receiver. The receiver will decrypt the massage and if the correct nonce is received then authentication is achieved. Using such a protocol ensures that the receiver is receiving packets from the sender and that the connection is live. In other words, “By using the once-in-a-lifetime value, R, and then checking the returned value, KA-B(R), Bob can be sure that Alice is both who she says she is (since she knows the secret key value needed to encrypt R) and live (since she has encrypted the nonce, R, that Bob just created).” (Kurose, 2013, p. 704)

Secure TCP Protocols

By default, TCP does not offer much, if any, security. This creates problems when trying to create online stores, where securing credit card information is top priority. In the early days of internet commerce, Netscape designed a protocol to resolve this issue called Secure Sockets Layer (SSL). “Since its inception, SSL has enjoyed broad deployment. SSL is supported by all popular Web browsers and Web servers, and it is used by essentially all Internet commerce sites (including Amazon, eBay, Yahoo!, MSN, and so on). Tens of billions of dollars are spent over SSL every year.” (Kurose, 2013, p. 712) The basic concept of SSL has three phases: handshake, key derivation, and data transfer. During the handshake phase, the sending node must do three things when following the SSL protocol. First, the sending node must establish a connection to the receiving node. Second, the receiving node must verify its identity to the sending node by use of a certificate. Third, the sending node must send a master secret key to the receiving node; which will be used to generate all symmetric keys during the session. By looking at Figure 2 we see the SSL handshake. The validity of the certificate used in an SSL session is verified by a Certification Authority. The downside to SSL is that it only encrypts data sent by a web browser.

Another way to secure a TCP connection is to use a Virtual Private Network (VPN). Instead of creating a totally separate network from the Internet, many companies use IPsec to create a VPN. This cheaper alternative allows companies to use the Internet without sacrificing security. The process of a VPN is fairly simple. First the data is sent to the Router to encrypt all the data before it is sent through the internet. The encrypted data is then sent through the internet to the destination, either another office or a remote user. Finally, the encrypted data is decrypted and sent to the user. As we see in Figure 3, unlike SSL, all data is encrypted. The way the VPN encrypts the data is by using IPsec. The encryption process comes in three basic parts: the

protocol, the connection, and a key. IPsec is a rather complicated protocol which can be divided into two principle protocols; the Authentication Header (AH) protocol and the Encapsulation Security Payload (ESP) protocol. To simplify things, the AH protocol offers source authentication and data integrity. However, the AH protocols do not offer confidentiality. The ESP protocol on the other hand, offers source authentication, data integrity, and confidentiality. Due to VPN and IPsec needing confidentially, the ESP protocol is more widely used. The encryption is done by way of logical connection called security association (SA). Before any IPsec datagrams are sent, both the sending and receiving routers must agree on how to authenticate and encrypt the datagram. This is seen in Figure 4. According to Kurose, SA is "unidirectional from source to destination. If both entities want to send secure datagrams to each other, then two SAs (that is, two logical connections) need to be established, one in each direction." (Kurose, 2013, p. 720) Much like an SSL handshake, IPsec uses Internet Key Exchange (IKE) to negotiate authentication and encryption algorithms. A series of messages are sent from sending to receiving nodes to authenticate identities and to create a secure connection.

Encryption of Other Data Packets

Web browser are not the only data packs that can be encrypted. Application like e-mail can also be encrypted to ensure data security. According to the Internet Engineering Task Force, Transport Layer Security (TLS) is widely used for e-mail protocols such as IMAP and POP. Much like SSL and VPN, a handshake deal is made between the sender and receiver. Unlike SSL and VPN, TLS is an Application Layer Protocol and needs a reliable transport channel (like TCP) to send the data. The TLS handshake uses a lockstep cryptographic handshake. This means the message must be transmitted and received in a particular order. If the messages are in any other order, an error occurs. Reordering and message loss creates problems for TLS handshakes. Also, TLS handshake messages are fairly larger than other datagram and creates problems with IP fragmentation. However, these problems are solved by the TLS protocol giving the handshake messages sequence numbers and allowing for data loss. According to the Internet Engineering Task Force, the TLS handshake contains three parts:

1. A stateless cookie exchange has been added to prevent denial-of-service attacks.
2. Modifications to the handshake header to handle message loss, reordering, and DTLS message fragmentation (in order to avoid IP fragmentation).
3. Retransmission timers to handle message loss.

   (Rescorla & Modadugu, 2012)

Like SSL, TLS must authenticate identities, agree on encryption, and exchange keys. A full picture of the TLS handshake can be seen in Figure 5.

Datagram security protocols are extremely susceptible to a variety of Denial of Service (DoS) attacks. In order to overcome this problem, the ClientHello and Server response contains a unique cookie during the TLS handshake process. This technique is borrowed from Photuris and IKE. When the Client sends a ClientHello to the server, it may respond with a HelloVerifyRequest message. This message contains a cookie that the client must retransmit with a ClientHello message. The server then continues the rest of the handshake only if the cookie is valid. This can be seen in detail in Figure 6. By adding the cookie technique to a TLS handshake, it makes it more difficult to perform a DoS attack because it forces the attacker/client to be able to receive a cookie and send it back. Since most DoS attackers use a spoofed IP, sending back a cookie with a spoofed IP will make the cookie invalid and end the handshake early. However, if the attacker manages to get a valid IP, then they could perform a DoS. This is why it is important to make sure the network's IP addresses are audited regularly to ensure that only authorized users have access to them.

Firewalls and Intrusion Detection Systems

Encryption is not the only way to secure a network. Hardware and software can be added to enhance the security of any network. Two of the most common hardware/software devices are firewalls and intrusion detection systems (IDS). Tools like firewalls and IDS are used in order to prevent attackers access to the network.

When looking at firewalls and IDS, both tools share some commonalities. Both systems are devices that are a combination of software and hardware. They both use a set of rules within the software to examine the network traffic and will use the appropriate hardware to block, alert, or allow the traffic. In traditional packet filters, "a network administrator configures the firewall based on the policy of the organization. The policy may take user productivity and bandwidth usage into account as well as the security concerns of an organization." (Kurose, 2013) In stateful packet filters "the firewall can observe the beginning of a new connection by observing a three-way handshake (SYN, SYNACK, and ACK); and it can observe the end of a connection when it sees a FIN packet for the connection. The firewall can also (conservatively) assume that the connection is over when it hasn't seen any activity over the connection for, say, 60 seconds." (Kurose, 2013) An application gateway, "is an application-specific server through which all application data (inbound and outbound) must pass. Multiple application gateways can run on the same host, but each gateway is a separate server with its own processes." (Kurose, 2013) A signature IDS, "sniffs every packet passing by it, comparing each sniffed packet with the signatures in its database. If a packet (or series of packets) matches a signature in the database, the IDS generates an alert." (Kurose, 2013) An abnormality-based IDS "creates a traffic profile as it observes traffic in normal operation. It then looks for packet streams that are statistically unusual, for example, an inordinate percentage of ICMP packets or a sudden exponential growth

in port scans and ping sweeps." (Kurose, 2013) Another similarity in both systems is both require a system administrator to either update the software or implement the system. This is the most important part of the system. If the firewall or IDS is not installed corrected or is not regularly updated; the system will not properly prevent attacks on the network.

Taking a closer look at firewalls and IDS, there are some things that both systems differ in. First, firewalls are the boundary between the internal network and the outside world. They are placed at each access point to the network. As seen in Figure 7. IDS on the other hand, are implemented in various locations on the network. This allows the IDS to focus on the fraction of the network traffic involved in each section and as a result making the tables more manageable. This is seen in Figure 8. Second, firewalls once installed do not require the network administrator's attention; with the exception of updates. The firewall sorts through all the incoming and outgoing traffic and automatically either allows the traffic or blocks the traffic based on the rules set. IDS on the other hand, requires the network administrator's attention once the traffic is flag as suspicious. The IDS runs through its rules and if the traffic is marked as suspicious, the network administrator is alerted to take further action.

Even though both firewalls and IDS have some overlap in their similarities, it is important to have both systems in place to offer full protection from attackers. Attacks are getting more and more sophisticated every day. Firewalls inspects only the header fields when determining if the traffic is to be allowed or blocked. However, as Kurose states it, "to detect many attack types, we need to perform deep packet inspection, that is, look beyond the header fields and into the actual application data that the packets carry… Clearly, there is a niche for yet another device—a device that not only examines the headers of all packets passing through it (like a packet filter), but also performs deep packet inspection (unlike a packet filter)." (Kurose,

2013) This is where a combination of firewalls (packet filters) and IDS (deep packet inspection) comes into play.

When using a firewall and VPN combination it is really important to ensure that the firewall and VPN are configure correctly. There are some cases where firewalls will block traffic that is needed by the VPN. For example, if the VPN is using port 80 to deliver its datagrams and the firewall is setup to block port 80; the VPN traffic is not going to get through the firewall. The VPN is needed to encrypt that data and the firewall is needed to keep potential attackers out. So both are needed and both cannot be used without the proper configuration. The same goes for TLS and any other security tool. If the firewall and the tools are not working together, then nothing is getting done. It's important to configure firewalls to block all (or as much as possible) bad traffic while allowing all good traffic through.

Conclusion

Network security is the most important aspect to any network. Every day there are many attacks on networks. Some are small while others are large. Without any security in place, hackers will have free range to any personal data; like SS numbers, credit card numbers, and so on. To fully secure any network multiple levels of security is needed. Not only does data need to be secured when transmitting but also when stored. This requires knowledge in order to properly design and setup such a security infrastructure. Unfortunately, network security is not a type of thing that can be setup and forgotten. It must be properly monitored and updated to ensure top security at all times. In a lot of cases multiple security tools are used to fully secure a network. Each tool is created for a particular security feature. Just using one tool would only secure part of the network and leaves other parts open for attacks. The network is only as strong as it's security tools. If the tools are weak; the security will be weak. The end goal is to make the network as strong as possible and keeping all of the data secure.

References

Daya, B. (n.d.). *Network Security: History, Importance, and Future.* Retrieved from http://web.mit.edu/~bdaya/www/Network%20Security.pdf

Kizza, J. M. (2005). *Computer Network Security.* Springer Science+Business Media, Inc.

Kurose, J. F. (2013). *Computer Networking: A Top-Down Approach.* New York: Pearson.

Rescorla, E., & Modadugu, N. (2012, January). *Datagram Transport Layer Security Version 1.2*. Retrieved from Internet Engineering Task Force (IETF): https://tools.ietf.org/html/rfc6347

Figures

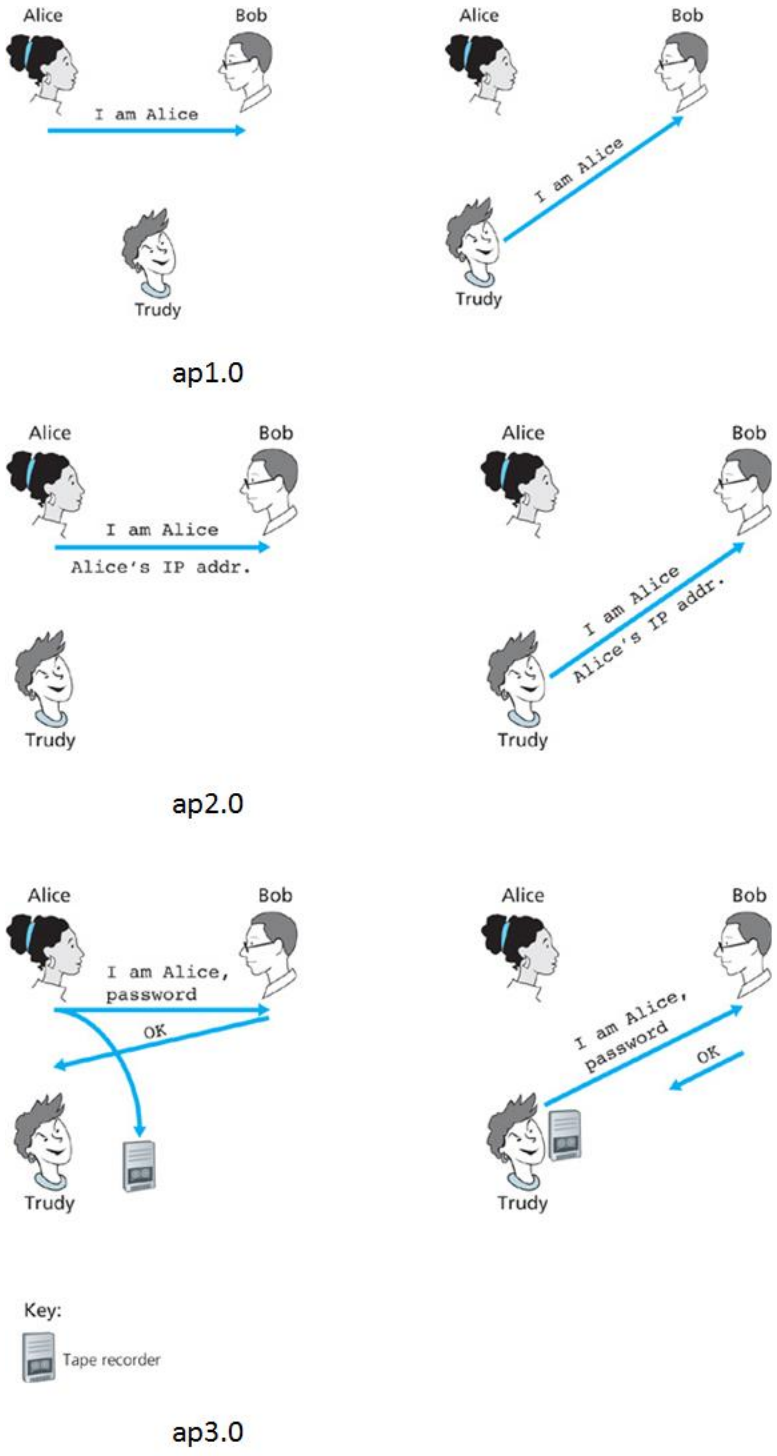

Figure 1. Protocol ap1.0, ap2.0, and ap3.0 with a failure scenario. Reprinted from Computer Networking: A Top Down Approach (p. 700-702), by Kurose, J. F., 2013, New York: Pearson.

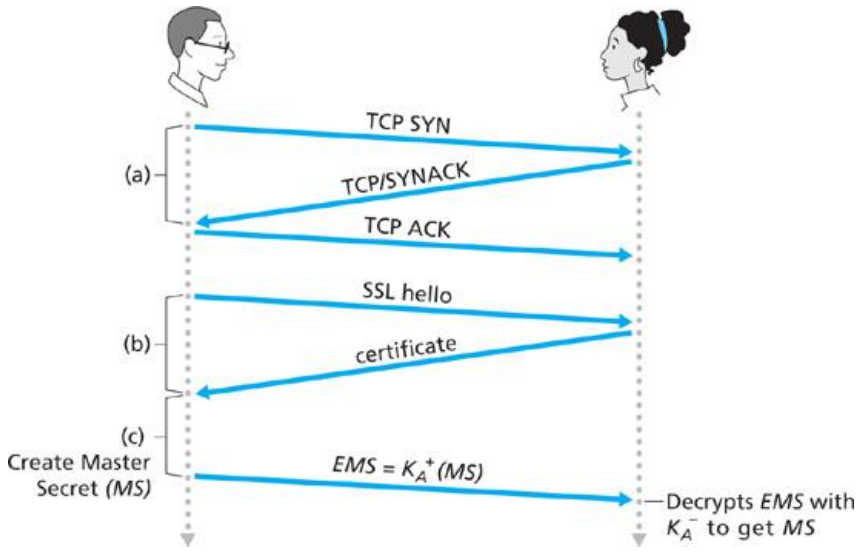

Figure 2. The almost-SSL handshake, beginning with a TCP connection. Reprinted from Computer Networking: A Top Down Approach (p. 713), by Kurose, J. F., 2013, New York: Pearson.

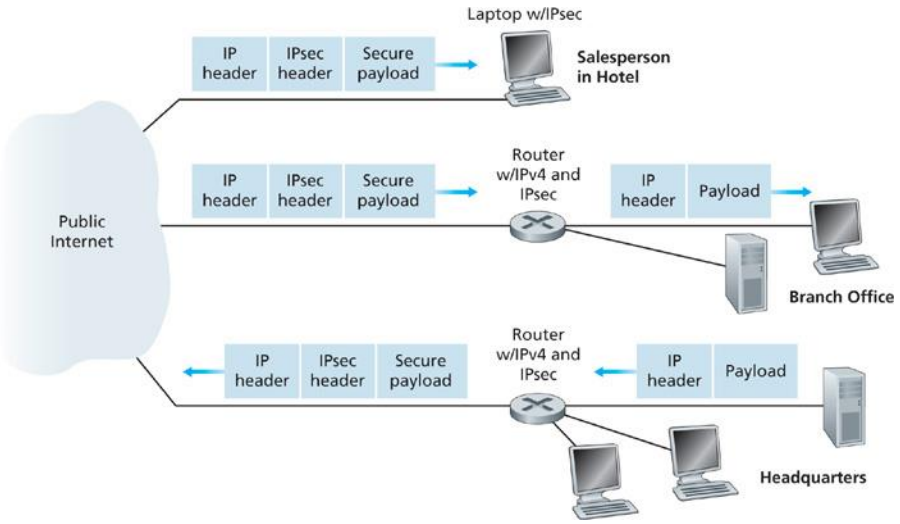

Figure 3. A Simple VPN. Reprinted from Computer Networking: A Top Down Approach (p. 719), by Kurose, J. F., 2013, New York: Pearson.

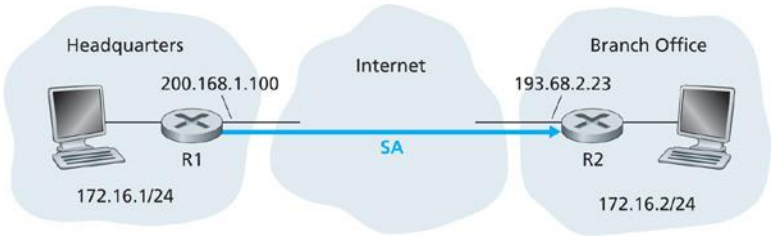

Figure 4. Security Association (SA) from R1 to R2. Reprinted from Computer Networking: A Top Down Approach (p. 720), by Kurose, J. F., 2013, New York: Pearson.

```
Client                                          Server
------                                          ------

ClientHello             -------->                           Flight 1

                        <-------    HelloVerifyRequest      Flight 2

ClientHello             -------->                           Flight 3

                                              ServerHello    \
                                             Certificate*     \
                                       ServerKeyExchange*      Flight 4
                                      CertificateRequest*     /
                        <--------        ServerHelloDone    /

Certificate*                                                \
ClientKeyExchange                                            \
CertificateVerify*                                            Flight 5
[ChangeCipherSpec]                                           /
Finished                -------->                           /

                                      [ChangeCipherSpec]    \ Flight 6
                        <--------             Finished      /
```

Figure 5. Message Flights for Full Handshake. Reprinted from https://tools.ietf.org/html/rfc6347, by Internet Engineering Task Force (IETF), 2012.

```
The exchange is shown below:

   Client                                  Server
   ------                                  ------
   ClientHello           ------>

                         <----- HelloVerifyRequest
                                (contains cookie)

   ClientHello           ------>
   (with cookie)

   [Rest of handshake]
```

Figure 6. TLS ClientHello ServerHello with cookies. Reprinted from https://tools.ietf.org/html/rfc6347, by Internet Engineering Task Force (IETF), 2012.

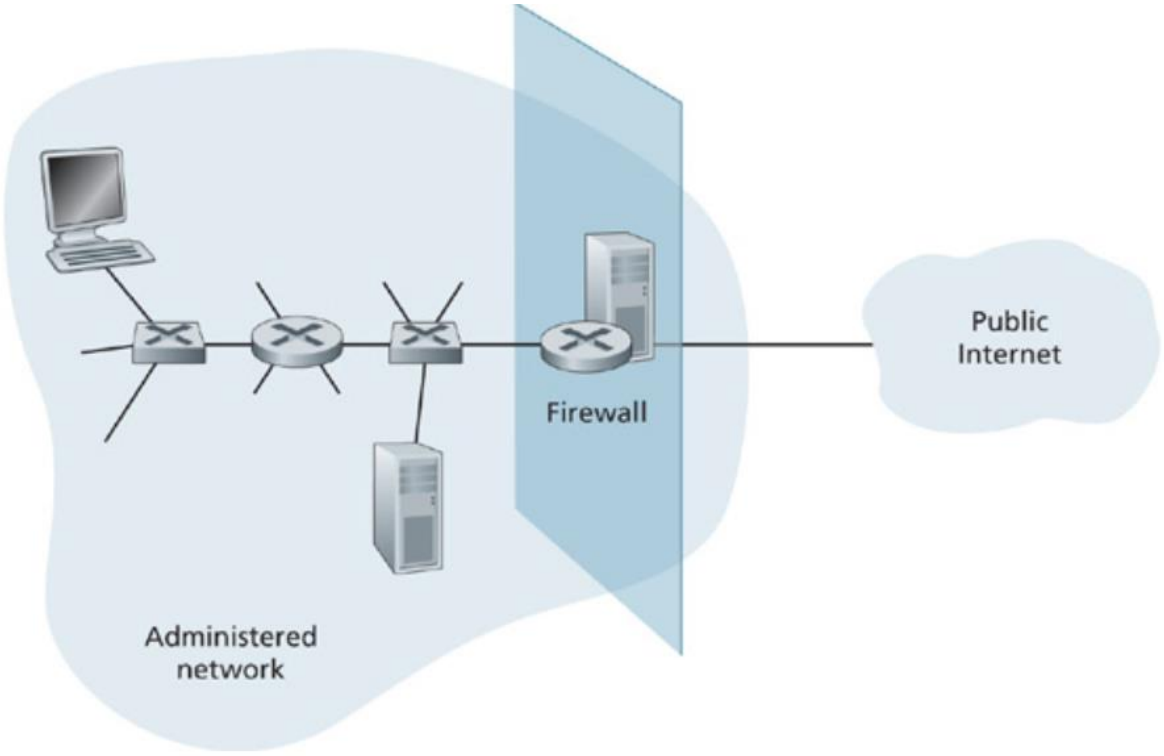

Figure 7. Firewall example. Reprinted from Computer Networking: A Top Down Approach (p. 732), by Kurose, J. F., 2013, New York: Pearson.

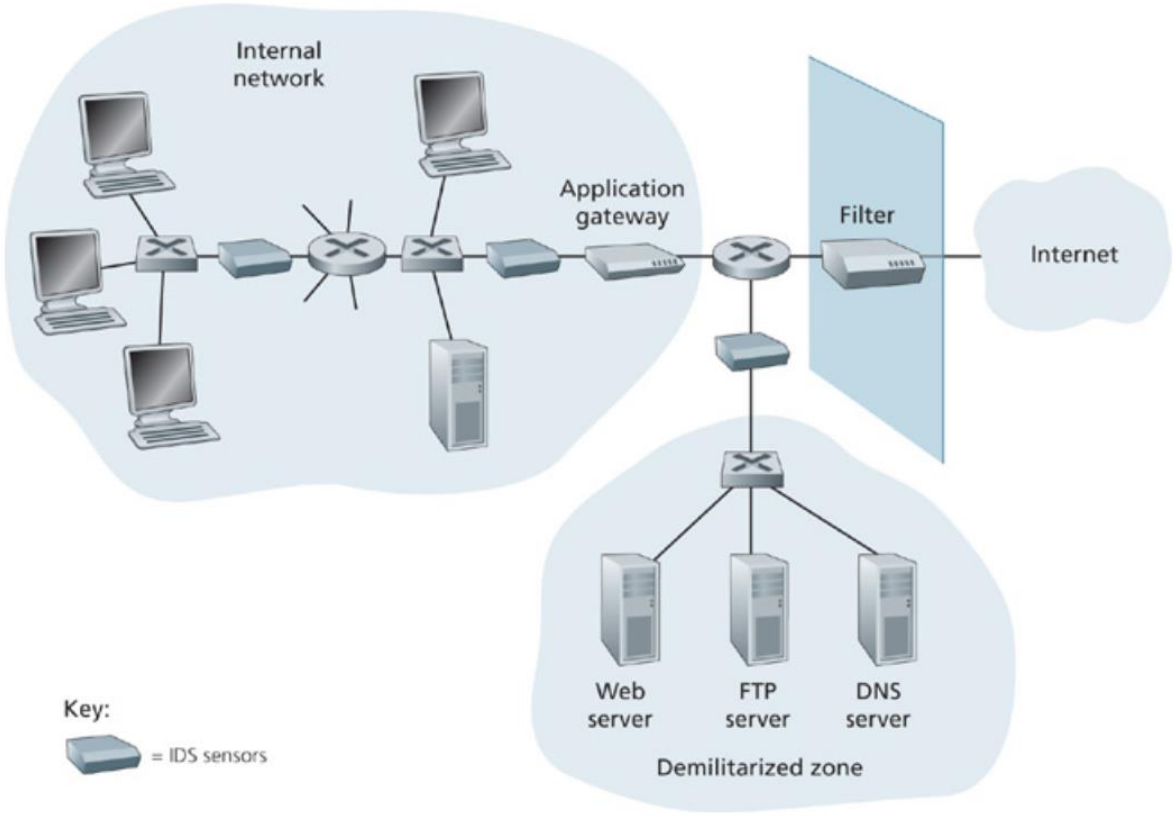

Figure 8. IDS example. Reprinted from Computer Networking: A Top Down Approach (p. 739), by Kurose, J. F., 2013, New York: Pearson.