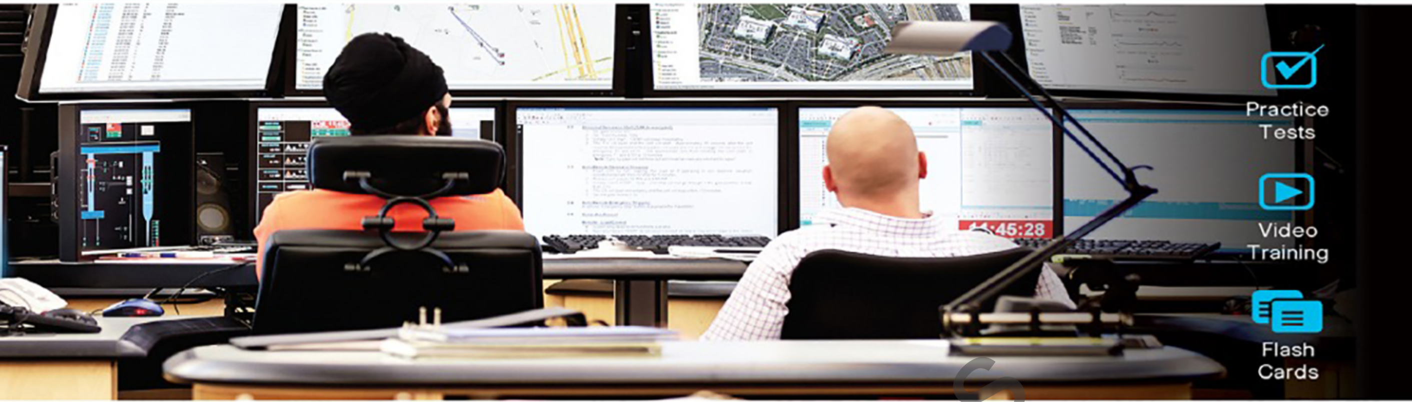

Study Planner

Review Exercises

Official Cert Guide

Advance your IT career with hands-on learning

Cisco CyberOps Associate

CBROPS 200-201

ciscopress.com

Omar Santos

Contents

CHAPTER 12

Challenges in the Security Operations Center (SOC)

This chapter covers the following topics:

Security Monitoring Challenges in the SOC

Additional Evasion and Obfuscation Techniques

There are several security monitoring operational challenges, including encryption, Network Address Translation (NAT), time synchronization, Tor, and peer-to peer communications. This chapter covers these operational challenges in detail. Attackers try to abuse system and network vulnerabilities to accomplish something; however, there is another element that can make or break the success of the attack. Attackers need to be *stealthy* and be able to evade security techniques and technologies. Attackers must consider the amount of exposure an attack may cause as well as the expected countermeasures if the attack is noticed by the target's defense measures. They need to cover their tracks.

In this chapter, you learn how attackers obtain stealth access and the tricks used to negatively impact detection and forensic technologies.

"Do I Know This Already?" Quiz

The "Do I Know This Already?" quiz allows you to assess whether you should read this entire chapter thoroughly or jump to the "Exam Preparation Tasks" section. If you are in doubt about your answers to these questions or your own assessment of your knowledge of the topics, read the entire chapter. Table 12-1 lists the major headings in this chapter and their corresponding "Do I Know This Already?" quiz questions. You can find the answers in Appendix A, "Answers to the 'Do I Know This Already?' Quizzes and Review Questions."

Table 12-1 "Do I Know This Already?" Foundation Topics Section-to-Question Mapping

Foundation Topics Section	Questions
Security Monitoring Challenges in the SOC	1–10
Additional Evasion and Obfuscation Techniques	11–20

1. Which of the following are benefits of encryption?
   a. Malware communication
   b. Privacy and confidentiality
   c. Malware mitigation
   d. Malware identification

2. Why can encryption be challenging to security monitoring?
   a. Encryption introduces latency.
   b. Encryption introduces additional processing requirements by the CPU.
   c. Encryption can be used by threat actors as a method of evasion and obfuscation, and security monitoring tools might not be able to inspect encrypted traffic.
   d. Encryption can be used by attackers to monitor VPN tunnels.
3. Network Address Translation (NAT) introduces challenges in the identification and attribution of endpoints in a security victim. The identification challenge applies to both the victim and the attack source. What tools are available to be able to correlate security monitoring events in environments where NAT is deployed?
   a. NetFlow
   b. Cisco Stealthwatch System
   c. Intrusion prevention systems (IPS)
   d. Encryption protocols
4. If the date and time are not synchronized among network and security devices, logs can become almost impossible to correlate. What protocol is recommended as a best practice to deploy to mitigate this issue?
   a. Network Address Translation
   b. Port Address Translation
   c. Network Time Protocol (NTP)
   d. Native Time Protocol (NTP)
5. What is a DNS tunnel?
   a. A type of VPN tunnel that uses DNS.
   b. A type of MPLS deployment that uses DNS.
   c. DNS was not created for tunneling, but a few tools have used it to encapsulate data in the payload of DNS packets.
   d. An encryption tunneling protocol that uses DNS's UDP port 53.
6. Which of the following are examples of DNS tunneling tools? (Select all that apply.)
   a. DeNiSe
   b. dns2tcp
   c. DNScapy
   d. DNStor
7. What is Tor?
   a. A blockchain protocol
   b. A hashing protocol
   c. A VPN tunnel client
   d. A free tool that enables its users to surf the Internet anonymously

8. What is a Tor exit node?
   a. The encrypted Tor network
   b. The last Tor node or the gateways where the Tor-encrypted traffic exits to the Internet
   c. The Tor node that performs encryption
   d. The Tor browser installed in your system to exit the Internet

9. What is a SQL injection vulnerability?
   a. An input validation vulnerability where an attacker can insert or inject a SQL query via the input data from the client to the application or database
   b. A type of vulnerability where an attacker can inject a new password to a SQL server or the client
   c. A type of DoS vulnerability that can cause a SQL server to crash
   d. A type of privilege escalation vulnerability aimed at SQL servers

10. Which of the following is a distributed architecture that partitions tasks or workloads between peers?
   a. Peer-to-peer networking
   b. P2P NetFlow
   c. Equal-cost load balancing
   d. None of these answers are correct.

11. Which of the following describes when the attacker sends traffic more slowly than normal, not exceeding thresholds inside the time windows the signatures use to correlate different packets together?
   a. Traffic insertion
   b. Protocol manipulation
   c. Traffic fragmentation
   d. Timing attack

12. Which of the following would give an IPS the most trouble?
   a. Jumbo packets
   b. Encryption
   c. Throughput
   d. Updates

13. In which type of attack does an IPS receive a lot of traffic/packets?
   a. Resource exhaustion
   b. DoS (denial of service)
   c. Smoke and mirrors
   d. Timing attack

14. Which of the following is *not* an example of traffic fragmentation?
   a. Modifying routing tables
   b. Modifying the TCP/IP in a way that is unexpected by security detection devices
   c. Modifying IP headers to cause fragments to overlap
   d. Segmenting TCP packets

15. What is the best defense for traffic fragmentation attacks?

a. Deploying a passive security solution that monitors internal traffic for unusual traffic and traffic fragmentation

b. Deploying a next-generation application layer firewall

c. Configuring fragmentation limits on a security solution

d. Deploying a proxy or inline security solution

16. Which of the following is a TCP-injection attack?

a. Forging a TCP packet over an HTTPS session

b. Replacing legitimate TCP traffic with forged TCP packets

c. Adding a forged TCP packet to an existing TCP session

d. Modifying the TCP/IP in a way that is unexpected by security detection

17. A traffic substitution and insertion attack does which of the following?

a. Substitutes the traffic with data in a different format but with the same meaning

b. Substitutes the payload with data in the same format but with a different meaning, providing a new payload

c. Substitutes the payload with data in a different format but with the same meaning, not modifying the payload

d. Substitutes the traffic with data in the same format but with a different meaning

18. Which of the following is *not* a defense against a traffic substitution and insertion attack?

a. De-obfuscating Unicode

b. Using Unicode instead of ASCII

c. Adopting the format changes

d. Properly processing extended characters

19. Which of the following is *not* a defense against a pivot attack?

a. Content filtering

b. Proper patch management

c. Network segmentation

d. Access control

20. Which security technology would be best for detecting a pivot attack?

a. Virtual private network (VPN)

b. Host-based antivirus

c. NetFlow

d. Application layer firewalls

Foundation Topics

Security Monitoring Challenges in the SOC

Analysts in the security operations center (SOC) try to have complete visibility into what's happening in a network. However, that task is easier said than done. There are several challenges that can lead to false negatives (where you cannot detect malicious or abnormal activity in the network and systems). The following sections highlight some of these challenges.

Security Monitoring and Encryption

Encryption has great benefits for security and privacy, but the world of incident response and forensics can present several challenges. Even law enforcement agencies have been fascinated with the dual-use nature of encryption. When protecting information and communications, encryption has numerous benefits for everyone from governments and militaries to corporations and individuals.

Key Topic

On the other hand, those same mechanisms can be used by threat actors as a method of evasion and obfuscation. Historically, even governments have tried to regulate the use and exportation of encryption technologies. A good example is the Wassenaar Arrangement, which is a multinational agreement with the goal of regulating the export of technologies like encryption.

Other examples include events around law enforcement agencies such as the U.S. Federal Bureau of Investigation (FBI) trying to force vendors to leave certain investigative techniques in their software and devices. Some folks have bought into the idea of "encrypt everything." However, encrypting everything would have very serious consequences, not only for law enforcement agencies, but also for incident response professionals. Something to remember about the concept of "encrypt everything" is that the deployment of end-to-end encryption is difficult and can leave unencrypted data at risk of attack.

Many security products (including next-generation IPSs and next-generation firewalls) can intercept, decrypt, inspect, and re-encrypt or even ignore encrypted traffic payloads. Some people consider this a man-in-the-middle (MITM) matter and have many privacy concerns. On the other hand, you can still use metadata from network traffic and other security event sources to investigate and solve security issues. You can obtain a lot of good information by leveraging NetFlow, firewall logs, web proxy logs, user authentication information, and even passive DNS (pDNS) data. In some cases, the combination of these logs can make the encrypted contents of malware payloads and other traffic irrelevant. Of course, this is as long as you can detect their traffic patterns to be able to remediate an incident.

It is a fact that you need to deal with encrypted data, whether in transit or "at rest" on an endpoint or server. If you deploy web proxies, you'll need to assess the feasibility in your environment of MITM secure HTTP connections.

TIP It is important to recognize that from a security monitoring perspective, it's technically possible to monitor some encrypted communications. However, from a policy perspective, it's an especially different task depending on your geographical location and local laws around privacy. Cisco has a technology that allows you to detect malicious activity even if the communication is being encrypted. That technology is called Encrypted Traffic Analytics (ETA), and it is integrated into the Stealthwatch and Cognitive Security solution, as shown in Figure 12-1.

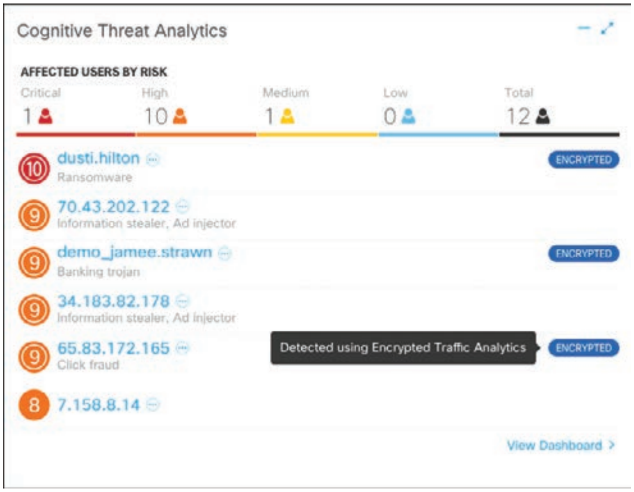

Figure 12-1 *Encrypted Traffic Analytics*

Security Monitoring and Network Address Translation

In Chapter 10, "Network Infrastructure Device Telemetry and Analysis," you learned that Layer 3 devices, such as routers and firewalls, can perform Network Address Translation (NAT). The router or firewall "translates" the "internal" host's private (or real) IP addresses to a publicly routable (or mapped) address. By using NAT, the firewall hides the internal private addresses from the unprotected network and exposes only its own address or public range. This enables a network professional to use any IP address space as the internal network. A best practice is to use the address spaces that are reserved for private use (see RFC 1918, "Address Allocation for Private Internets").

NOTE Cisco uses the terminology of *real* and *mapped* IP addresses when describing NAT. The real IP address is the address that is configured on the host before it is translated. The mapped IP address is the address that the real address is translated to.

Static NAT allows connections to be initiated bidirectionally, meaning both to the host and from the host.

Key Topic

NAT can present a challenge when you're performing security monitoring and analyzing logs, NetFlow, and other data, because device IP addresses can be seen in the logs as the "translated" IP address versus the "real" IP address. In the case of Port Address Translation (PAT), this could become even more problematic because many different hosts can be translated to a single address, making the correlation almost impossible to achieve.

Security products, such as the Cisco Stealthwatch system, provide features that can be used to correlate and "map" translated IP addresses with NetFlow. This feature in the Cisco Stealthwatch system is called *NAT stitching*. This accelerates incident response tasks and eases continuous security monitoring operations.

Security Monitoring and Event Correlation Time Synchronization

Server and endpoint logs, NetFlow, syslog data, and any other security monitoring data are useless if they show the wrong date and time. This is why as a best practice you should configure all network devices to use Network Time Protocol (NTP). Using NTP ensures that the correct time is set and all devices within the network are synchronized. Also, another best practice is to try to reduce the number of duplicate logs. This is why you have to think and plan ahead as to where exactly you will deploy NetFlow, how you will correlate it with other events (like syslog), and so on.

DNS Tunneling and Other Exfiltration Methods

Threat actors have been using many different nontraditional techniques to steal data from corporate networks without being detected. For example, they have been sending stolen credit card data, intellectual property, and confidential documents over DNS using tunneling. As you probably know, DNS is a protocol that enables systems to resolve domain names (for example, cisco.com) into IP addresses (for example, 72.163.4.161). DNS is not intended for a command channel or even tunneling. However, attackers have developed software that enables tunneling over DNS. These threat actors like to use protocols that traditionally are not designed for data transfer because they are less inspected in terms of security monitoring. Undetected DNS tunneling (otherwise known as *DNS exfiltration*) represents a significant risk to any organization.

In many cases, malware can use Base64 encoding to put sensitive data (such as credit card numbers, personal identifiable information [PII], and so on) in the payload of DNS packets to cyber criminals. The following are some examples of encoding methods that could be used by attackers:

- Base64 encoding
- Binary (8-bit) encoding
- NetBIOS encoding
- Hex encoding

Several utilities have been created to perform DNS tunneling (for the good and also for the bad). The following are a few examples:

- **DeNiSe:** This Python tool is used for tunneling TCP over DNS.
- **dns2tcp:** Written by Olivier Dembour and Nicolas Collignon in C, this tool supports KEY and TXT request types.
- **DNScapy:** Created by Pierre Bienaimé, this Python-based Scapy tool for packet generation even supports SSH tunneling over DNS, including a SOCKS proxy.
- **DNScat or DNScat-P:** This Java-based tool created by Tadeusz Pietraszek supports bidirectional communication through DNS.
- **DNScat (DNScat-B):** Written by Ron Bowes, this tool runs on Linux, Mac OS X, and Windows. DNScat encodes DNS requests in NetBIOS encoding or hex encoding.
- **Heyoka:** This tool, written in C, supports bidirectional tunneling for data exfiltration.

- **Iodine:** Written by Bjorn Andersson and Erik Ekman in C, this tool runs on Linux, Mac OS X, and Windows, and can even be ported to Android.
- **Nameserver Transfer Protocol (NSTX):** This tool creates IP tunnels using DNS.
- **OzymanDNS:** Written in Perl by Dan Kaminsky, this tool is used to set up an SSH tunnel over DNS or for file transfer. The requests are Base32 encoded, and responses are Base64-encoded TXT records.
- **psudp:** Developed by Kenton Born, this tool injects data into existing DNS requests by modifying the IP/UDP lengths.
- **Feederbot and Moto:** Attackers have used this malware using DNS to steal sensitive information from many organizations.

Some of these tools were not created with the intent of stealing data, but cyber criminals have used them for their own purposes.

The examples in Figure 12-2 and Figure 12-3 demonstrate how DNS tunneling can be achieved with the Iodine tool. Figure 12-2 shows the Iodine server listening for any connections from clients using DNS resolution for the domain h4cker.org.

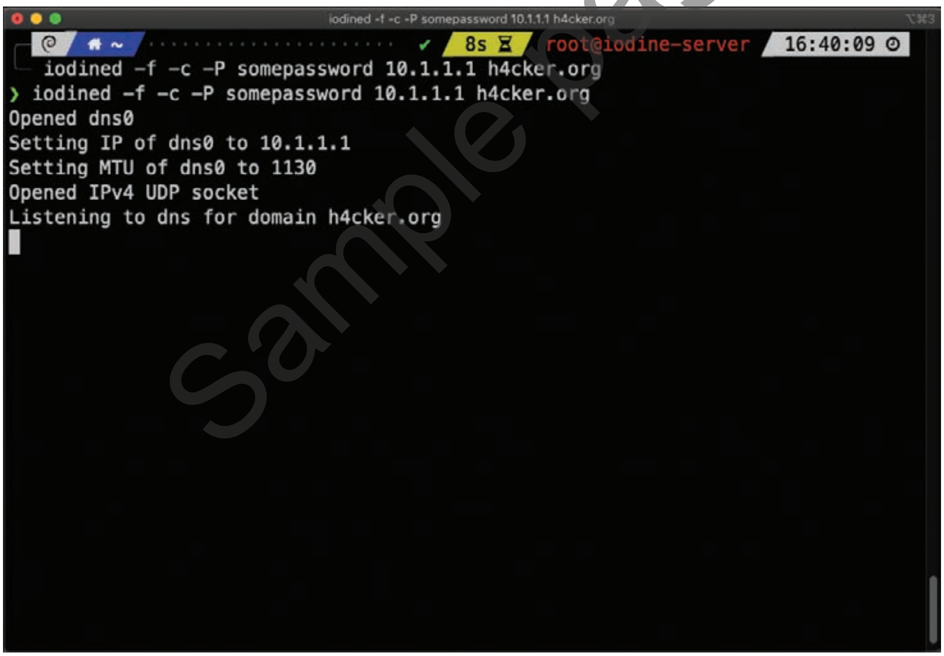

Figure 12-2 *Iodine DNS Tunneling Server*

Figure 12-3 shows the Iodine client (assume that this is a compromised system). The client successfully established a connection to the Iodine server. The 192.168.88.207 IP address is the address configured in the network interface card (NIC) of the server. The 10.1.1.1 is the IP address used by Iodine to communicate with the clients over the tunnel. In this example, the client IP address is 10.1.1.2, and the server tunnel IP address is 10.1.1.1. All data is now sent over the DNS tunnel, and the domain h4cker.org is used for DNS resolution.

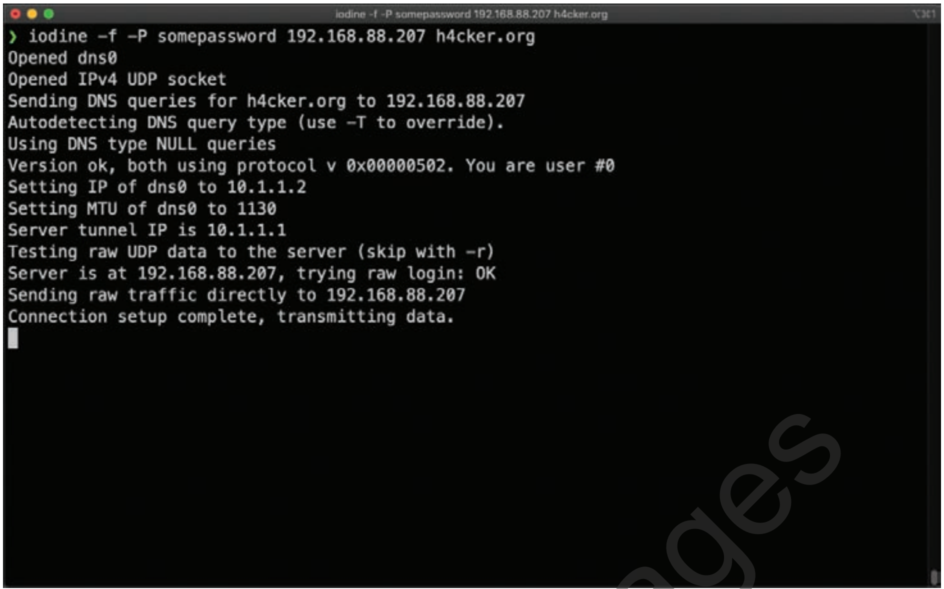

Figure 12-3 *Iodine DNS Tunneling Client*

Key Topic

Security Monitoring and Tor

Many people use tools such as Tor for privacy. Tor is a free tool that enables its users to surf the web anonymously. Tor works by routing IP traffic through a free, worldwide network consisting of thousands of Tor relays. Then it constantly changes the way it routes traffic to obscure a user's location from anyone monitoring the network.

NOTE Tor is an acronym of the software project's original name, "The Onion Router."

The use of Tor also makes security monitoring and incident response more difficult because it's hard to attribute and trace back the traffic to the user. Different types of malware are known to use Tor to cover their tracks.

This "onion routing" is accomplished by encrypting the application layer of a communication protocol stack that's nested just like the layers of an onion. The Tor client encrypts the data multiple times and sends it through a network or circuit that includes randomly selected Tor relays. Each of the relays decrypts a layer of the onion to reveal only the next relay so that the remaining encrypted data can be routed on to it.

Figure 12-4 shows the Tor browser. You can see the Tor circuit when the user accessed h4cker.org from the Tor browser. The packets first went to a host in the Netherlands, then to hosts in Norway and Germany, and finally to h4cker.org.

A Tor exit node is basically the last Tor node or the gateway where the Tor encrypted traffic exits to the Internet. A Tor exit node can be targeted to monitor Tor traffic. Many organizations block Tor exit nodes in their environment. The Tor project has a dynamic list of Tor exit nodes that makes this task a bit easier. This Tor exit node list can be downloaded from https://check.torproject.org/exit-addresses.

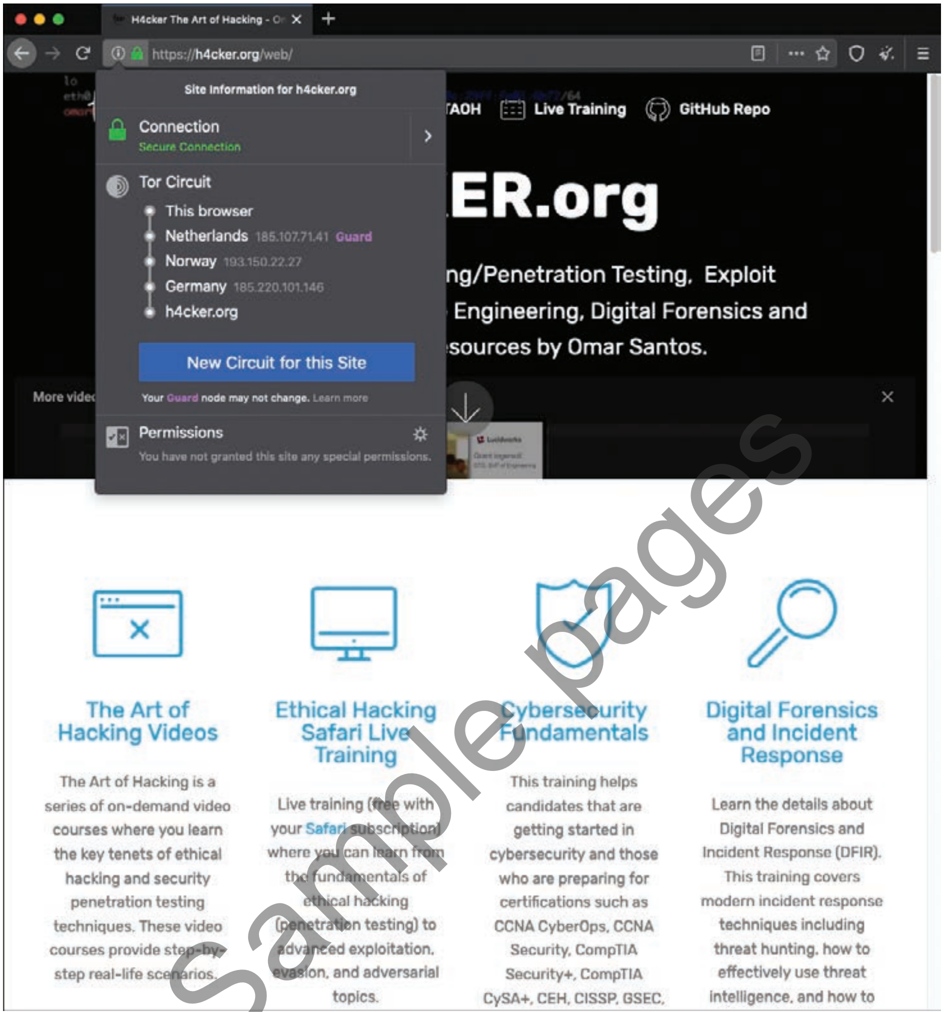

Figure 12-4 *The Tor Browser*

NOTE Security products such as the Cisco Next-Generation Firepower software provide the capability to dynamically learn and block Tor exit nodes.

Security Monitoring and Peer-to-Peer Communication

Key Topic

Peer-to-peer (P2P) communication involves a distributed architecture that divides tasks between participant computing peers. In a P2P network, the peers are equally privileged, which is why it's called a *peer-to-peer* network of nodes.

P2P participant computers or nodes reserve a chunk of their resources (such as CPU, memory, disk storage, and network bandwidth) so that other peers or participants can access those resources. This is all done without the need of a centralized server. In P2P networks, each peer can be both a supplier as well as a consumer of resources or data. A good example was the music-sharing application Napster back in the 1990s.

P2P networks have been used to share music, videos, stolen books, and other data; even legitimate multimedia applications such as Spotify use a peer-to-peer network along with streaming servers to stream audio and video to their clients. There's even an application called Peercoin (also known as PPCoin) that's a P2P crypto currency that utilizes both proof-of-stake and proof-of-work systems.

Universities such as MIT and Penn State have even created a project called LionShare, which is designed to share files among educational institutions globally.

From a security perspective, P2P systems introduce unique challenges. Malware has used P2P networks to communicate and also spread to victims. Many "free" or stolen music and movie files usually come with the surprise of malware. Additionally, like any other form of software, P2P applications are not immune to security vulnerabilities. This, of course, introduces risks for P2P software because it is more susceptible to remote exploits, due to the nature of the P2P network architecture.

Additional Evasion and Obfuscation Techniques

Attackers can use SSH to hide traffic, such as creating a reverse SSH tunnel from a breached system back to an external SSH server, hiding sensitive data as the traffic leaves the network. Figure 12-5 provides an example of how a typical SSH session functions.

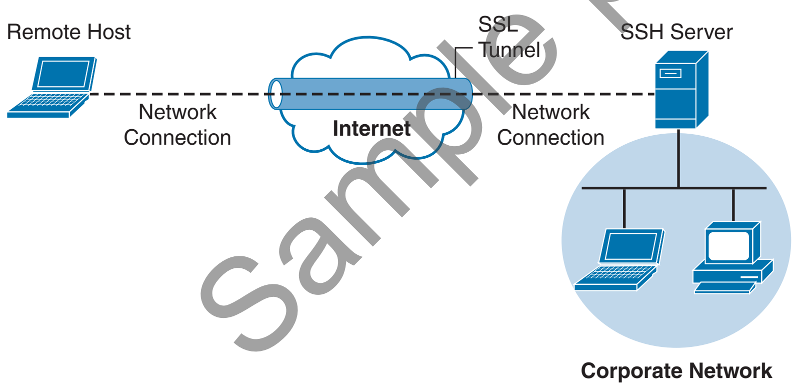

Figure 12-5 *SSH VPN Example*

You can use SSH tunnels over other tunnels such as VPNs, DNS tunnels, and so on. For instance, you can create a DNS tunnel and then have an SSH tunnel over it.

There are many use cases where an attacker breaches a network and launches some form of a VPN session. An example is using Hak5's LAN Turtle USB adapter, which can be configured to auto-launch a reverse SSH tunnel to a cloud storage server, essentially creating a cloud-accessible backdoor to a victim's network.

It is challenging for an administrator to identify the LAN Turtle because it sits on a trusted system and does not require an IP address of its own to provide the reverse-encrypted tunnel out of the network.

Figure 12-6 shows an example of a LAN Turtle plugged into a server, providing an encrypted tunnel to an attacker's remote server. This would represent a physical attack that leads to a backdoor for external malicious parties to access.

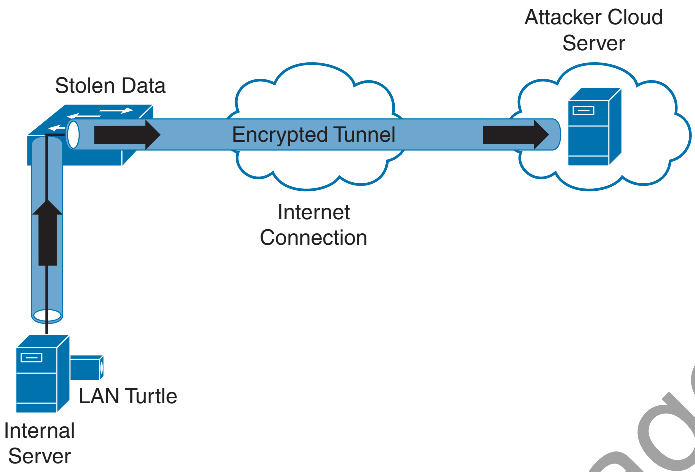

Figure 12-6 *LAN Turtle SSH Tunnel*

The LAN Turtle is just one example of the many tools available that can be planted on a network to create an unauthorized backdoor. The Packet Squirrel is another device that can be deployed to give an attacker remote access to a target network. All of these tools are available to the public on websites like hak5.org.

Another encryption concept is hiding the actual data. There are many techniques for doing this, such as enterprise file encryption technologies that encrypt files and control access to opening them. An example is having a software agent installed on a server that specifies which files should be encrypted. When a file is removed that should be encrypted, it is tagged and encrypted, with access provided only to people within a specific authentication group. People within that group can use a host-based agent that auto-logs them in to the file, or they could be sent to an online portal to authenticate to gain access to the file.

The term *data at rest* means data that is placed on a storage medium. Data-at-rest security requirements typically refer to the ability to deny all access to stored data that is deemed sensitive and at risk of being exposed. Typically, this is done by encrypting data and later removing all methods to unencrypt the data. Examples include hard disk encryption where a hard drive is encrypted, making it impossible to clone. The same concept can be applied to file encryption technology, where the data owner can expire access to the file, meaning all users won't be able to unencrypt it.

Many attackers abuse encryption concepts such as file and protocol encryption to hide malicious code. An example would be an attack happening from a web server over SSL encryption to hide the attack from network intrusion detection technologies. This works because a network intrusion detection tool uses signatures to identify a threat, which is useless if the traffic being evaluated is encrypted. Another example would be encoding a malicious file with a bunch of pointless text, with the goal of confusing an antivirus application. Antivirus applications also use signatures to detect threats, so adding additional text to malicious code