CHAPTER 11

Network Analysis and Forensics

INFORMATION IN THIS CHAPTER

- OSI model
- Analysis for managers
- Flow-level analysis
- Metadata analysis
- Application-level analysis
- Signature analysis
- Full-packet capture
- Network forensics
- Sensor network architecture

INTRODUCTION

This chapter begins the first of four on strategies for recognizing when an attack has penetrated the defender's resistance. This chapter focuses on human-driven network-based analysis. Chapter 12 considers automated detection in the form of network-based intrusion detection and prevention systems (NIDPSs). Chapter 13 changes focus from the network to the host and discusses both recognition and forensics after an attack has been recognized. Finally, Chapter 14 discusses data integrity and recognition based on various properties of the data that can be location-independent.

There are several benefits to network-centric recognition strategies. This is true of both manual and automated analysis; this chapter focuses on manual analysis, because every automated rule was at some point discovered and analyzed by a human, probably in response to an incident. The techniques and procedures described in this chapter are likely to be performed due to some suspicion that something is wrong, either as part of the incident response process (see Chapter 15) or due to public reporting or news coverage of a

Introduction to Information Security. DOI: http://dx.doi.org/10.1016/B978-1-59749-969-9.00011-0

common problem. This chapter describes some tools and policies necessary to enable such retrospective analysis.

Much of the preparation process for establishing a network traffic analysis and forensics capability is a set of practical trade-offs about what the defender can reasonably expect to detect with a particular detail level in their sensor architecture and inspection. Before discussing a variety of inspection and analysis options in more depth, the chapter begins with an overview of OSI-model protocol layers and what information is available at each layer to frame the discussion.

The remainder of the chapter discusses these levels of analysis, along with the practical trade-offs they include, such as processing time and storage space. This discussion echoes that in Chapter 5 regarding the resources necessary for various network-based frustration technologies, such as packet filters and proxies. Along with the details the chapter provides some example tools or uses for each specialization.

INTRODUCTION TO THE OSI MODEL

The Open Systems Interconnection (OSI) model is a way to divide up the problem of communicating between two remote computers. The abstract model has seven layers, and each layer has certain functions that should be performed by the service at that layer [1]. Further, each layer needs only know about the layer below it, and needs to only worry about providing reliable information to the layer above it. This structure makes the communication structure modular and flexible. For instance, a web application does not need to know if it is being transmitted over radio waves, fiber optics, or copper phone lines, because the application (web) is more than one layer removed from the transmission media. If the web application did need to know this, communications over the Internet would be too complicated.

In practice, the implementation of communications protocols does not strictly align with the OSI model. OSI was developed by the International Organization for Standardization (ISO) as ISO 7498, and has some support from the International Telecommunication Union (ITU) as the X.200 series. However, much of the Internet infrastructure standards have been developed by the Institute of Electrical and Electronics Engineers (IEEE) and Internet Engineering Task Force (IETF), which do not strictly abide by the OSI model in their protocol definitions. An introduction to how the Internet actually works using the IEEE and IETF protocols is covered in Chapter 3.

Nonetheless, Internet protocols are roughly arranged to follow the OSI model, and regardless, it provides a useful framework for thinking about the different layers of abstraction in Internet protocols. Once the reader has a

OSI Model -- a summary			
	Name of Data	**Layer**	**Summary**
Processed on Host	Data, Messages	*7. Application*	From network process to application process
		6. Presentation	Encryption, convert to/from host-specific representations
		5. Session	Manage sessions between applications between hosts
	Segments	*4. Transport*	End-to-end connections and reliability
Processed on Network	Packets/Datagram	*3. Network*	Logical address for end-to-end delivery, routing to logical addresses
	Frames	*2. Data link*	Physical address for one-hop delivery
	Bits	*1. Physical*	Transmission and signals on physical medium

FIGURE 11.1

The seven layers of the OSI model, the term for data units of that layer, and a short summary of the functions of each layer [1].

handle on these abstractions, understanding the trade-offs for network analysis at different network architecture layers is easier.

Figure 11.1 summarizes the seven layers of the OSI model. There are some features of network communications that are not explicitly represented in the model, such as management and security. Conceptually, these can be applied in different ways at each layer. This discussion will leave these aspects aside. Layer 1 deals with the physics of how to transmit information reliably on a medium, such as copper or radio waves. Layer 7 is transitioning data on the host computer to a format for the user or his or her application. The intermediate layers are steps along that process.

Each layer has some control information to accomplish its task. For example, layer 3 is responsible for logical addresses of endpoints; IP addresses can be considered the layer 3 header. Layer 2 is responsible for transmitting data between each computer along the way. So to make a layer-2 frame, the machine adds a media access control (MAC) address and some other information to the IP datagram. IP and MAC addresses are not defined in or related to the OSI model, but conceptually this is where they fit.

At each network hop, the router strips off this layer-2 information, reads the layer-3 information, decides where the next destination is to move the data toward the desired endpoint, prepends new layer-2 information, and resends it. This process of adding addressing and other information around the data is called *encapsulation*. A similar encapsulation process, with different details, happens every time data is passed from one layer to the layer above or below it. Figure 11.2 displays this process.

Through several layers of encapsulation, decoding, and reencapsulating, data is transmitted across the Internet. Different network devices and network security devices strip off different numbers of layers to do their job. A rough schematic of this process is displayed in Figure 11.3. As a general rule, the more layers a device has to read and process, the more computationally

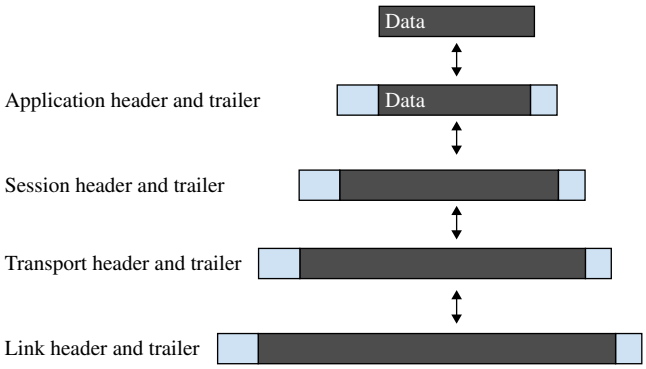

FIGURE 11.2

The general idea behind encapsulation. When each header and trailer is added or removed, the layer treats the data in the darkened area as mere data and passes it along, even though it may have header and trailer information for other layers.

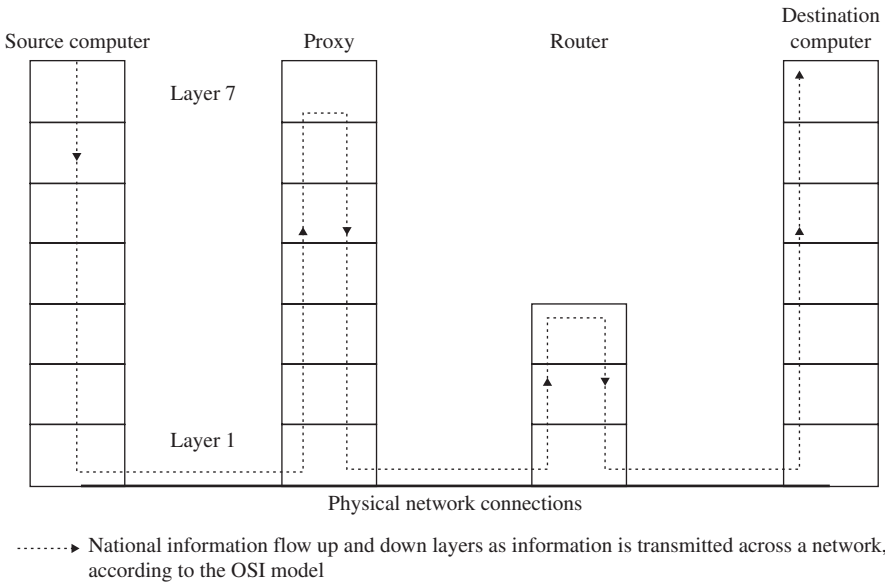

FIGURE 11.3

Sample transmission of data from one host to its destination across the network, as viewed by what OSI model layers are involved in each step. Note repeated encapsulation and processing of data units as the data traverses different layers.

expensive that process is. Switches only understand up to layer 2. Routers only need to understand up to layer 3. Application proxies need to understand all seven layers. This is also true of human analysts. If the analyst is just looking for communication between IP addresses, the task will be simpler,

both cognitively for the analyst and computationally for the computer assisting, than piecing back together a whole application from what was seen on the network.

The next discussion provides starting points for those who might manage analysts. The remainder of the chapter will start the discussion of network analysis with information available at lower layers of the OSI model. This focus is perhaps unorthodox. On smaller networks, analysts can often start analyzing whole packets without too much trouble. However, the discussion will focus on structures that can help defenders find their way through a complex problem, from general (lower layers) to more specific (higher layers). This process of starting in quick broad strokes, characterizing, and drilling down when indications of specific problems are found helps analysts be successful. The appropriate technical and policy frameworks also need to be in place to support this analysis architecture. There is a cost to supporting analysis architecture, and to extract value from this expenditure it is important the rest of the organization understands the analysts' capabilities.

ANALYSIS FOR MANAGERS

Not everyone in an organization is an analyst, and more people than just the analysts need to benefit from the results of analysis. For those who will not be performing analysis but managing those who are, an important item to know is what questions one can reasonably ask an analyst and expect a useful answer. There is not one single set of reasonable questions; different questions are more or less reasonable with different classes of tools. This chapter discusses tools and what information they provide to help guide these expectations.

The following sections are roughly organized such that later sections describe techniques that use information from higher layers in the OSI model. This organization is necessarily rough because the Internet protocols do not strictly adhere to the OSI model. However, the general idea of trade-off holds: as one inspects traffic in more detail, the analyst trades precision for breadth. More detail equates to analyzing more layers of the model. On small networks, an organization might afford equipment to provide both. On larger networks, the analysts and managers must jointly decide if precision or breadth is the priority; different preferences may be possible at different levels of the organization. If these choices are made cohesively, subunits of the organization may be able to prioritize precision while an organization-wide team can prioritize breadth—both organizationally and temporally—storing multiple years of records. Such specialization provides better organizational defense.

Each analysis focus allows different questions, as the following sections describe. However, not every question can be answered, no matter what technology and sensors are deployed in the organization. Management cannot expect analysts to perform feats from Hollywood any more than Indiana Jones represents the average life of an archaeologist. A more subtle point is that Google has not only spent an unbelievable amount of money, but a decade of work from most of the leading experts and textbook authors on the topic to develop its search capability and infrastructure. Security analysts cannot be expected to answer questions or provide network situational awareness with the ease of an Internet search engine unless the organization has made a similarly intensive decade-long investment.

There are questions that no amount of engineering support can provide an answer to. The most frustrating questions are often the most simple. Perhaps the top of the list is "Who sent this?" or, relatedly, "What country did this come from?" The key to understanding why these questions are not sensible to ask is to understand that the Internet is a logical addressing scheme, not a physical one. An IP address is not bound to a computer or a country any more than a person named John must have short red hair. The method that is used to distribute IPs gives this illusion, and there are some bureaucratic structures that try to pin a certain IP to a certain locality, but none of that is built into the logical addressing structure of the Internet. As such, the Internet protocols are happy to oblige any attacker who wants to work around these bureaucratic assumptions. Furthermore, as discussed in Chapter 5, the attacker can use a previously compromised host as a proxy to anonymize his or her actions, so there is no reason to believe that the external IP in a communication is actually the final endpoint or the attacker's IP, even if we could know where this IP was located.

The people and computers who are playing by the rules and being good citizens may not evade the IP location structures, providing an illusion that Internet protocols are accurately geolocated. However, network security analysis is primarily concerned with those who are not playing by the rules. Therefore, questions about where something came from or where it went are less productive than questions about which internal resources were affected and what behavior or attacks were observed. The precision of the information about a particular attack and the breadth of information about all attacks will vary based on the analytic capability.

FLOW-LEVEL ANALYSIS

Network flow is a relatively abstract method of working with network data. It is mostly concerned with information abstracted from layer-3 and layer-4

headers (e.g., TCP/IP suite headers). Cisco created flow as a data format for routers to report high-level, condensed status information about the traffic traversing a network. The IETF eventually adopted this data format as a standard router reporting format [2].

A flow is a summary of all the packets identified by a set of properties. For example, a common method is to group all the packets in one direction from the same source IP and port to a particular destination IP and port, on an IP protocol—for example, Transmission Control Protocol (TCP) or User Datagram Protocol (UDP)—together into one flow. When the conversation is terminated, such as with a TCP FIN packet (a packet with the *finished* flag set), or after some configured amount of time, the flow is closed and the router exports it. The flow includes many statistics about the related set of packets—for example, number, total bytes, duration, and sometimes such items as a guess about what application protocol was used in the communication—but not the packets themselves.

One feature of network flow is that it can quickly and concisely tell the analyst that IP address 10.0.0.1 communicated with 10.0.0.2, sent 1 megabyte, used the port usually associated with web traffic, and did so in less than a second. If the analyst does not have any context for what IP 1 and 2 are, this may not be helpful. So the human analyst must be flexible about incorporating context from various sources to make flow data meaningful.

For example, the Swiss CSIRT (computer security incident response team) publishes IP addresses of known command-and-control (C2) servers [3,4]. Hosts in the defender's network should not communicate with these IP addresses. If complete conversations are observed, then the defender's machines are almost certainly infected; botnet C2 servers usually do not have benign uses. Network flow is a convenient resource to investigate such queries. The query is relatively fast, and the content of the communication is less important than that the communication occurred at all. Additionally, since flow condenses years of traffic, data can be stored and investigated once such malicious addresses are known. This historical perspective allows the analyst to trace back to evidence of the initial problem; it is not uncommon for compromises to go undetected for months or even years, and so such historical records are often necessary to piece together what happened [5–8].

SiLK (System for Internet-Level Knowledge) and Argus are two common open-source network flow analysis tool suites. Both are under active development and have active communities of users. Sincer 2004, there is even an annual conference focusing on open-source network flow analysis, FloCon. The interested reader can find a variety of good network flow–related material in the past proceedings at *www.flocon.org*.

METADATA ANALYSIS

Metadata is data about data [9]. This can be used to contextualize network data, or some metadata is useful to analyze in its own right. Metadata comes from a variety of sources, and one can expect to learn very different things from different metadata. Some examples of metadata include the following:

- Application label of what application seemed to be running in a flow [10].
- Domain names that were hosted on an IP address at the time traffic was captured.
- ID of intrusion detection and prevention system (IDPS) alerts triggered by network traffic or flow.
- Users logged in to a system at the time traffic was captured.
- Uniform resource locators (URLs) extracted from an email message.
- Autonomous system numbers (ASNs) offering routes to an IP address at the time traffic was captured.
- User-agent strings in an HTTP transaction.

In some ways, metadata is the result of application-level analysis applied to another context than that in which it was derived. Domain name system (DNS) information is a good example of this. To extract the name–address mapping from DNS there has to be the capability to monitor it at the application level. However, as far as metadata about IPs generally is concerned, it does not matter who sent the DNS query, just what the contents are. In this way DNS analysis can provide metadata about what domains are hosted on what IPs. DNS analysis as application-level analysis would be about what computers asked what questions, which is a different analysis and provides different insights.

One example of a rather complex analysis that provides metadata for the whole network is to do "network profiling using flow" [11]. If the organization has a flow-monitoring capability, one can do this analysis to construct a profile about how IP addresses have behaved in the past. This provides metadata for other analysis, since IPs can be tagged with labels like "DNS client" and "NAT/Gateway."

Sometimes, abstracted data is not the analyst's choice but rather the lawyer's. Some amounts of data are considered too sensitive to be collected without cause. The extent to which this is true varies by jurisdiction. For example, the U.S. Department of Homeland Security issues privacy impact assessments (PIAs) about what is acceptable for security analysts to capture on U.S. government networks. This includes network flow and passive DNS, but not full-packet capture unless there is a reason. However, a vetted intrusion detection system (IDS) signature alert is considered sufficient cause to capture a small amount of the subsequent related packets [12].

If enough metadata has been collected it can be analyzed. This is usually a task for interorganizational groups like coordination centers and information sharing and analysis centers (ISACs) since these groups are collection points for information sharing. In general, when information is collected from multiple organizations patterns can be discerned that the individual organizations would not be able to discover on their own, since they lack the context of the other organizations. Information sharing is also the only way an organization can determine if an attack was a specially targeted attack, because no one else will have seen the attack. This information is itself further metadata. It is important to determine what attacks are targeted because these represent attacks from more determined adversaries who are not merely opportunistic attackers.

Some organizations share their collected intelligence back out to the public. These resources are usually more casual blogs and websites rather than formal scholarly articles. The following are a sample of some useful (network) security context–related blogs, as of 2013, which the reader will likely find helpful to read occasionally:

- Krebs on Security (*http://krebsonsecurity.com/*)
- CERT Coordination Center blog (*http://www.cert.org/blogs/certcc/*)
- Swiss CSIRT blog (*http://www.abuse.ch/*)
- Securelist, a collaborative blog with multiple respected authors (*https://www.securelist.com/en/blog*)
- Dark Reading, cyber-security news (*http://www.darkreading.com/*)
- For more, there is a list of 20 security-related blogs at *http://www.veracode.com/blog/2012/02/top-20-security-blogs/*

In general, analysts should be encouraged to spend some time reading about current threats and security trends. The point of metadata is to help analysts contextualize the data. There is some context that cannot be fit neatly into a tag on a flow, such as what the particular exploit-of-choice of a certain organized crime group is this week. Such information does often find its way to publication through various blogs and casual postings.

Metadata is useful, especially if the analyst knows where it came from and how it was generated. In many cases, this is derived from application-level analysis, both within the organization and data shared from other organizations.

APPLICATION-LEVEL ANALYSIS

Application-level analysis is about analyzing the data transmitted by an application as the application would have interpreted it. This is a resource-intensive type of analysis in several regards. To capture it, the device has to traverse all seven layers of the OSI model, including possibly decrypting data, which requires computing time. Further, there needs to be a different application

parser for each application to be analyzed. One tool, such as Wireshark, can utilize these parsers together, but this is additional overhead. The payoff for this effort depends on the application, but as mentioned in other chapters, the most popular applications are those likely to be the target of application-level analysis. The following are good candidates:

- HTTP
- DNS
- Email
- Kerberos (perhaps within Microsoft Active Directory)
- BGP (Border Gateway Protocol)[1]
- TLS (Transport Layer Security)[2] connection information and certificates

Analyzing each of these applications in detail allows the analyst to extract information that is otherwise unavailable. This approach requires appropriate monitoring at the relevant servers, network edge, or both, depending on the application and whether the goal is to collect information about internal or external computers. Selective application-level analysis can provide benefits over blanket full-packet capture since it potentially eliminates a lot of noise from the data by extracting just the features of the application that an analyst finds interesting. The other side of this coin is that if the analyst is not consulted in the development of the system, or as analytic goals change, the system may not collect the correct information.

A mature analytic group within an organization will want to ask questions of all of these applications for various reasons. Each provides some information that only it can provide. However, that does not mean that throwing all of these application data sources at a novice analyst capability will instantly provide situational awareness and actionable intelligence. The opposite is more likely—novice analysts will be overwhelmed by the avalanche of different information types. The analysis will have to be tailored to the organization itself, and the human analytic capability will also need to grow up with the organization to some extent. For this reason, organizational memory and low staff turnover are important goals; a reasonable expectation might be that it would take an analyst two to five months to understand how to fluently analyze each particular application. Each of flow, packet capture, IDS logs, and so on has a similar learning curve. Therefore, supporting people is still one of the most important steps in ensuring quality application-level analysis, and analysis in general.

---

[1] Although BGP is a method of determining how IP addresses are routed, it is also an application that runs on TCP/IP; by default it listens on TCP port 179 [13].

[2] TLS is not technically an application, but runs at what could be called layers 5 or 6 of the OSI model.

PROFILE: SURESH LAKSHMAN KONDA

Network Analysis on an Internet Scale

Suresh Konda was born in 1950 in India. Konda earned his M.S. in public policy and management from the Heinz College (then School) at Carnegie Mellon in 1975 and went on to earn his Ph.D. in 1980. He published in artificial intelligence (data mining), information portals, and vulnerability trends. Konda was one of the world's pioneering researchers in information security, and worked at Carnegie Mellon's Software Engineering Institute (SEI) for many years, often collaborating across the university.

He was recognized widely for his work on the security of large networks. To provide a quantitative basis for security decision making, he pioneered the development of large-scale collection infrastructures for network traffic summaries, ultimately resulting in the System for Internet-Level Knowledge, a network flow data collection and analysis tool suite. He received the SEI's first Angel Jordan Award for Innovation in 2002. Konda was an extremely passionate individual, eager to promote quality work in himself and in others. In turn, he provoked intense loyalty in his colleagues.

Suresh died suddenly in 2003, while living in Washington, DC. Subsequent to his death, the tool suite he pioneered was renamed SiLK, with the capitalization taken from his initials. The Heinz College also established the Konda Memorial Lecture Program and the Konda Award for outstanding Ph.D. student publication.

SIGNATURE ANALYSIS

Signature analysis is, most simply, looking for something that is already known to be suspicious or malicious. Usually the term applies to an IDS inspecting full-packet and application data and comparing it to known signatures, producing alerts. Network IDSs are the topic of Chapter 12, so we will leave that description aside for now. Human analysts can use signatures in a few ways: testing, as metadata, hunting, and campaign detection.

Before signatures are installed in an IDS, they should be tested. This process is a special case of change management, as described in Chapter 10. The producers of a signature are probably good at determining that the signature has a high true-positive rate: that it detects what it is supposed to detect. Signatures should be tested before deployment because the false-positive rate—that is, erroneous alerts—will be different for each environment. As discussed in Chapter 12, too many false positives can render a system useless.

More interesting is to use the fact that some particular trace of traffic matched a known-malicious signature as metadata for other analysis. If this capability is designed well, the analyst can ask for details such as "Show me all the traffic flows that matched the most recent flash zero-day exploit signature." Such questions help the analyst orient him- or herself faster to recognize which attacks succeeded and which the host managed to resist.

Skilled analysts will be able to go hunting for evidence of attacks that have evaded all the automated detection and recognition systems. Part of this

process is the rather fuzzy process of looking for behavior that is outside the accepted normal behavior of machines on a network. This knowledge of what is unacceptable behavior is more complex than what a machine can usually bring to bear, because it depends on context and inferring intent to an extent that machines have not been able to match human capabilities. It also requires an analyst with detailed knowledge of both how the Internet protocols are intended to be used and how they are used in practice. Any one tool or monitoring capability is probably insufficient to allow an analyst to hunt down intrusions. A hunting capability within an organization is expensive, but it is an important investment if the organization is being specifically targeted by attackers. So far, persistent human attackers can always outsmart machines, so the only way to find them is with other persistent humans looking to defend the network.

Detecting campaigns against the organization is related to the task of finding persistent human attackers. If an analyst detects 50 distinct attack events, it is important to know if those attackers were by 15 different entities or 1. The organizational response to attacks by 15 distinct entities should be different than if there is 1 entity that is so tenacious and successful. The best public example of campaign identification is Mandiant's tracking of APT1 (advanced persistent threat) as an element of the People's Republic of China armed forces [5]. Campaign detection relies on certain attackers using certain signature tactics, techniques, and procedures (TTPs). Attackers cannot really avoid this; having no pattern is itself a pattern. For example, the TDSS (aka Alureon) botnet randomizes certain characteristic startup communications to avoid signature detection, which itself can be used as a signature to detect its startup communications [14].

FULL-PACKET CAPTURE

Full-packet capture is any technology that records every bit that goes over the wire for later inspection. The results are often colloquially referred to by their file extension: pcap.[3] This is the most voluminous network analysis option; even a rather modest 100-megabit link—a good and common residential connection in 2013—could fill up a terabyte of hard drive space in just 22 hours. A common corporate link of 10 gigabits could fill the same space in less than 14 minutes. As discussed before, many if not most intrusions are not detected for weeks [8]. Retention of sufficient packet capture data so that it can be analytically useful is a challenge. On the other hand, packet capture is the only way to reconstruct the exact details of an attack, which makes it an important tool.

[3] Various pcap files are available for free for training or research. For an index, see *http://www.netresec.com/?page=PcapFiles*.

Unfortunately, network payload traffic is often obscured. Defenders advocate network encryption, such as discussed in Chapter 8. Attackers have taken this lesson as well, and often enough it is precisely the data that would make pcap valuable that is encrypted. In some cases this encryption can be avoided, such as by forcing the hosts on the network to use a proxy and simply having the proxy record the traffic in its unencrypted state. The proxy breaks the connection into two distinct connections, and since network encryption only protects data in transit and not at the endpoints, the proxy can observe it. However, attackers can still obscure data from proxies, such as by encrypting or encoding the data twice.

Once packet capture data is stored, and assuming it has the desired data in an unencrypted format, it can still be challenging to find the important data. Searching through a large collection of pcap files has been likened to finding a needle in a needle-stack. Most of the solutions to this problem involve using one of the OSI analysis levels as an index into the pcap itself. Network flow particularly lends itself to this task, since it contains a condensed form of all the same data as full-packet capture. Broad queries can be done using flow tools, and then when more detail is needed all the packets that were part of the flow in question can be pulled out. Some flow meters provide the option to retain and index the pcap files when it processes packet capture to produce network flow [10].

Full-packet capture remains an important tool for analysts, but it needs to be filtered before it is stored. Judicious full-packet capture, rather than blanket capture, provides useful information. Do you really need a complete copy of every streaming video that the employees have watched for the past three weeks? No. Realistically, even if it becomes important to watch for some odd reason, keeping the URL should be enough to fetch the video again from the content provider. When planning a packet capture capability, it is important to allow time for optimizing what is captured and what is allowed to be summarized by other technologies. That process will be idiosyncratic to each network, but without adaptation and a focused goal, packet capture can be hard to work with.

NETWORK FORENSICS

The difference between what might be called human-driven network analysis and network forensics is not sharply drawn. One might say that network forensics is network analysis in some relation to a reactive investigation, although not necessarily a legal one. Kessler and Fasulo [15] provide a more detailed discussion of the attributes of network forensics, and its importance within digital forensics. Network forensics is a subset of digital forensics, which is a topic covered more thoroughly in Chapter 13. This section introduces some features specific to networks, but there are many aspects shared with forensics more generally as discussed in Chapter 13.

To do network forensics the network traffic must be intentionally recorded. This fact contrasts sharply with host-based forensics, since hosts generally write information to stable storage media as a matter of course. Traffic can be recorded under specific goals, in response to a warrant or other cause for concern raise by the CSIRT (see Chapter 15), or traffic can be recorded as a matter of course all the time. The level of detail about traffic that can be recorded without cause varies by legal jurisdiction. On most large networks, it is impractical to keep full-packet capture of the whole network for long periods of time anyway, and so network flow is both a reasonable technical and legal compromise. This is essentially the assessment the U.S. government has reached in regard to monitoring its own civilian networks [12], for example.

The other primary purpose of network forensics is to support root-cause analysis. The goal of root-cause analysis is to discover the initial method and manner of compromise. These results can be used to harden the organization against future attacks; for more information on this feedback process, see Chapter 15. As noted earlier, most intrusions go undetected for weeks, if not months or years [5–8]. This makes network forensics particularly challenging. It is even more difficult if only reactive network capture is used, since there is no network evidence of how the compromise began. Without general historical records (as opposed to purpose-captured traffic in response to an incident), root-cause analysis is often impossible.

A network forensics capability is an important supplement to host forensics because the traffic record provides out-of-band evidence of activity. If the adversary compromises a host, he or she may be able to erase some or all the evidence of the compromise. However, the network monitoring devices are not subject to such manipulation, and so can provide a more stable, if less precise, measure of activity in the organization. There are certainly steps that an adversary can take to avoid network detection; some are discussed in Chapter 12. All the same, most common attacks arrive on the network, and so they will leave some trace.

The tools for network forensics are largely the same as those for network analysis. As alluded to before, it is mostly the motivation that differentiates the two terms. With forensics, the motivation is more about gathering evidence to assign responsibility to specific individuals. The anonymity possible on computer networks makes attribution particularly challenging. There are a variety of special considerations that need to be made to preserve the proper legal and ethical frameworks while performing work with this motivation. These considerations are not special to network forensics, but hold more generally for digital forensics. Since these handling considerations make up the bulk of the difference between human-driven network analysis and network forensics, they are left for Chapter 12.

SENSOR NETWORK ARCHITECTURE

Where the network sensors are located in the network is important. Sensor architecture is a subset of the importance of network architecture generally, which was introduced in Chapter 5. Just as some analysis techniques are better or worse at detecting certain attacks, certain sensor placements are better suited for detecting certain attacks or gathering certain information. All of the preceding types of sensors can be located in various places on the organization's network. It is also important for analysts to understand where a sensor is located to accurately analyze the information from the sensor and its impact on the organization.

There are two broad types of sensor placement: on the edge and internal. Observing traffic as it is entering or leaving the organization is more common. There are usually a small number of Internet access points for an organization, so installing sensors on the edge is somewhat easier. The other reason for sensing on the edge is the assumption that attacks on the organization come from the outside. If this were true, then if the defenders can observe and analyze all the attacks coming in on the edge, then the defenders could see all the attacks.

Unfortunately, not all attacks come from the outside. Internal sensors are necessary to detect such problems. There are two general use cases for internal attacks that could be important. One is detecting insider attacks where the attacker is actually a member of the organization with some valid credentials and authorizations. The other would be in the case where an attacker compromises an internal resource and uses that resource to compromise other internal resources. Therefore, determining the full scope and root cause of an infection likely will involve monitoring internal communications as well as communications at the edge.

SUMMARY

Human-driven network analysis is an important part of a recognition strategy because IT systems are made by humans, for humans, and are attacked by humans, therefore, humans must also be involved in their effective defense. There are several specializations of network analysis. The OSI model can be used to conceptualize the different analysis types. One benefit of this model is the idea that as an analysis moves up to process more and higher layers, the resources required for the analysis increase. Figure 11.4 provides a rough guide for which layers are involved in which analysis type. It introduces a layer 8, for human intelligence tasks like reading blogs and campaign analysis that do not easily fit into the technical OSI model.

Network analysis and the OSI Model -- a summary					
Layer	Network analysis type				
	Flow-	*metadata*	*application-*	*signature*	*full*
"8. Human Intel"		-		-	
7. Application		-	#	-	#
6. Presentation			-	-	#
5. Session			-	-	#
4. Transport	#			-	#
3. Network	#	-		-	#
2. Data link					#
1. Physical					
Key:	*# :used in analysis*		*- :partially used*		

FIGURE 11.4

The seven layers of the OSI model and the rough usage of information from those layers in different network analysis specializations. The chart introduces a layer 8, not in the OSI model, to cover human intelligence tasks relevant to network defense.

The questions that one can expect an analyst to answer vary depending on what technology and analysis types are available to them. However, some questions are always difficult to answer based on certain features of the Internet, such as the fact that IP addresses are logical, not physical, which makes physically locating machines and attacks difficult. Yet, in most cases, the variety of analysis techniques available can be arranged in a complementary manner, such that the strengths of one support the weaknesses of the others.

For better or worse, effective human-driven analysis remains trade craft rather than science. A skilled analyst derives much benefit from familiarity with the network he or she is defending, and this specific knowledge is not easy for an organization to capture. This difficulty is not true for all analyst experiences. Chapter 12 discusses the aspects of analyst workflow that are easier to codify, and the tools and procedures used to capture those experiences and apply them to network defense at machine speed.

REFERENCES

[1] International Organization for Standardization and International Electrotechnical Commission. Open systems interconnection—basic reference model: the basic model. ISO/IEC, 1996. 7498-1:1994(E).

[2] Claise B. Cisco Systems NetFlow Services Export Version 9. RFC 3954, 2004.

[3] Abuse.ch blog. abuse.ch Palevo Tracker. Retrieved Apr 6, 2013, from <https://palevotracker.abuse.ch/>; 2013.

[4] Abuse.ch blog. abuse.ch Zeus Tracker. Retrieved Apr 6, 2013, from <https://zeustracker.abuse.ch/>; 2013.

[5] Mandiant. APT1: exposing one of China's cyber espionage units. Retrieved Apr 8, 2013, from <http://intelreport.mandiant.com/>; 2013.

[6] The Center for Internet Security. The CIS security metrics: consensus metric definitions, v1.0.0, 2009. Retrieved Apr 8, 2013, from <https://buildsecurityin.us-cert.gov/swa/downloads/CIS_Security_Metrics_v1.0.0.pdf>.

[7] Verizon. 2012 Data breach investigations report. Retrieved Apr 8, 2013, from <http://www.verizonenterprise.com/DBIR/2012/>; 2012.

[8] Verizon. 2013 Data breach investigations report. Retrieved Apr 8, 2013, from <http://www.verizonenterprise.com/DBIR/2013/>; 2013.

[9] Weinberger D. Everything is miscellaneous: the power of the new digital disorder. Times Books, New York City; 2007.

[10] Software Engineering Institute. CERT NetSA security suite—monitoring for large-scale networks. Retrieved Apr 15, 2013, from <http://tools.netsa.cert.org/>; 2013.

[11] Whisnant A, Faber S. Network profiling using flow. CMU/SEI-2012-TR-006, 2012. Retrieved from <www.sei.cmu.edu/library/abstracts/reports/12tr006.cfm>.

[12] U.S. Department of Homeland Security. Privacy impact assessment for the national cybersecurity protection system, 2012. Retrieved from <http://www.dhs.gov/sites/default/files/publications/privacy/privacy-pia-nppd-ncps.pdf>.

[13] Rekhter Y, Li T, Hares S. A border gateway protocol 4 (BGP-4). RFC 4271. Updated by RFCs 6286, 6608, 2006.

[14] Jerrim J. Detecting Malware P2P traffic using network flow and DNS analysis. Flocon 2013. Software Engineering Institute, CERT Directorate. Retrieved Apr 15, 2013, from <http://www.cert.org/flocon/2013/presentations/jerrim-john-detecting-malware.pdf>; 2013.

[15] Kessler GC, Fasulo M. The case for teaching network protocols to computer forensics examiners. Proceedings of the conference on digital forensics, security and law, 2007. p. 115–137.

Chapter Review Questions

1. What are the seven layers of the OSI model? Which layers are processed by network nodes and which are processed on the host?
2. What simple questions are hard to answer using only IP address information?
3. What are some common applications that might be important to do application-level analysis of?
4. What is campaign detection and analysis?
5. How is network forensics different from human-driven network analysis?

Chapter Exercises

1. Figure out which applications are most prevalent on a network you care about, either your home network or your corporate/school network if you are authorized. How would you prioritize application-level analysis of these applications? Which are most important to the security of the network?
2. Read some of the blogs listed on page 237 and summarize a couple recent relevant posts.