Interested in learning more about security?

SANS Institute
InfoSec Reading Room

This paper is from the SANS Institute Reading Room site. Reposting is not permitted without express written permission.

Applying the OSI Seven Layer Network Model To Information Security

This paper focuses on reviewing a key area of data networking theory - The Open Systems Interconnect (OSI) Seven Layer Network Model. This paper demonstrates the application of the model's concepts into the context of information security. This paper presents the perspective that common information security problems map directly to the logical constructs presented in the OSI Seven Layer Network Model, and seeks to demonstrate the Seven Layer Model's usefulness in evaluating information security problems and solutions. ...

Copyright SANS Institute
Author Retains Full Rights

AD

Entrust
< Let's Talk

Applying the OSI Seven Layer Network Model To Information Security

© SANS Institute 2004, Author retains full rights.

By Damon Reed
November 21, 2003

SANS GIAC GSEC Practical Assignment version 1.4b Option One

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Abstract

Data networking is a critical area of focus in the study of information security. This paper focuses on reviewing a key area of data networking theory - The Open Systems Interconnect (OSI) Seven Layer Network Model. This paper demonstrates the application of the model's concepts into the context of information security. This paper overall presents the perspective that common information security problems map directly to the logical constructs presented in the OSI Seven Layer Network Model, and seeks to demonstrate the Seven Layer Model's usefulness in evaluating information security problems and solutions. The OSI Model is presented by way of both formal definition and practical terms that affect information security on a layer-by-layer basis. For each layer, examples of common information security threats and controls are evaluated by how they fit into the OSI Seven Layer Model's layers of classification, with notes on exceptions and special cases. Once the seven layers have been covered as a basis for the discussion, it is presented that the Seven Layer Model's scheme for interaction between the layers gives insight to some of the problems faced by focused, "single-layer" security solutions. To answer these problems, a multi-layer "defense-in-depth" approach is examined by example, taken from the viewpoint of network model layers rather than discrete solutions and logical or physical hardware layers. This paper concludes with some proposed extensions to the model that complete the model's application to information security problems.

Introduction to the OSI Seven Layer Model

Networking is a prime concern for information security. The ubiquitous nature of network connectivity may let us access the world from our computer, but it also lets that same world gain access back to us in ways we may not desire. No matter how well we secure our own hosts, we are still vulnerable if the parts of the infrastructure between our distant destinations and ourselves fall victim to intentional exploitation or unwitting mishap. Information security and data networking are inextricably linked topics. Today's network engineer has no choice but to be security-conscious, and the security engineer has no choice but to understand the network he is tasked to secure. [1]

A great deal of formalized study has been devoted to the science and methodology of designing and maintaining networks. One formal system that network engineers discuss and apply frequently is the OSI Seven Layer Model for Networking, developed by the ISO (International Standards Organization) to define a standardized method for designing networks and the functions that support them. This model describes seven layers of interaction for an information system communicating over a network, presenting a stack of layers representing major function areas that are generally required or useful for data communication between nodes in a distributed environment. Starting from a high-level application perspective, data is sent down the stack layer by layer, each layer adding information around the originally presented data until that original data plus its layers of added content are represented at the bottom-most layer as a physical medium such as bursts of colored light or voltage across a wire in order for that data to physically travel from one point to the other in the real world.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Once the data takes this real-world journey, the true power of the model comes into play, as the protocols at each layer are mandated by the design model to strip cleanly away the information and formatting added by their corresponding layer at the sending end of the conversation as the data rises back up through the seven layers at the receiving side, acting on the transmitted content at their layer and pushing back up the stack what was originally pushed down at the other end. What was presented to layer three at the sending side should be exactly what layer three on the receiving side passes back up to the layer above. This can be described as the layers "communicating" between one another on the sending and receiving side, all the way up to the application layer at the top, where pure data is sent from one side and received intact and unchanged on the other. There are exceptions to this concept such as application-aware NAT, where lower layer protocols may alter the data passed to them from above, but this is an exceptional case and a technical violation of the model. The isolation of layers also allows abstraction such that lower layers are not dependant on upper layers beyond what is needed to exchange data between the layers. This is especially important at the lower levels where the same data may have to travel across different media or link-layer protocols to get where it is going. This delivers a key goal of the model - interchangeability of layers such that different environments can use the stack to standardize communications and interconnect on a common basis. [2,3,4,5,6]

Like many ISO standards, much of its formal theory does not make it into the real world of actual implementation, but the powerful concepts that the OSI model present are a key element in most modern network system designs. Anyone who has worked with data networking or security has likely heard the terms "layer three" or "layer two" or "application layer." This terminology stems directly from the ISO model and how it is applied to practical solutions. The model concepts are conventionally used to design and troubleshoot networks, and the seven-layer model is standard fare on any network engineering certification exam or interview. Careful study of the model can show us support for concepts we have learned from more conventional forms of information security theory, and understanding and applying the model to information security scenarios can also help us assess and address information security threats in a network environment, allowing us to organize efforts to make security assessments and perform forensic analysis of compromised systems and threats presented in theory and found in the wild.

Take for example the bottom-most physical layer of the network. Reviewing the flow of information through the model, we see that all layers above depend upon the physical layer to deliver the data. We can draw a parallel between this and the concept that physical security is critical for all information security assets. From a networking perspective, if one can unplug a device from the network or otherwise physically alter it, communication stops. If there are errors at the physical layer, the layers above cannot typically recover, and must either retransmit or fail. If one can physically access a device, it is near impossible to prevent some amount of data loss or disclosure. All of the above layers depend upon the integrity of the physical layer. [7]

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Another example would be application security at layer seven. Suppose that we apply good security through the underlying layers, with physical isolation (layer one), private VLANs (layer two), and firewalls with tight packet filter policies (layers three and four). But then we are deficient on our application layer security (layer seven, and often layers six and five), using unpatched server software and poorly written application and script code. Since the vulnerabilities lie within the application, in a pure seven-layer model we would be hard pressed to defend against this at the lower levels, as the controls at lower layers would only be able to address their respective layer of protocol, and not issues that occur above. This illustrates the conventional approach of defense in depth - a firewall and DMZ are not sufficient to protect a host from outside attack if the ports that the firewall allows connect to vulnerable services (WWW, SMTP, Netbios, SQL). The services themselves need to be secure. [8]

Using the model as an objective measure for security is closely related to this concept of defense-in-depth, and by way of deconstructing the layers and then examining how they interact, we can see supporting evidence and clear rationale for the need of that blended, defense-in-depth approach in securing networks, systems, applications, and data. The following sections will take each layer and examine them on the basis of their formal definition and their practical place in the network, show example security threats, and present possible controls of those risks that apply to the layer in question.

© SANS Institute 2004, Author retains full rights.

Layer One - the Physical Layer

The physical layer is responsible for the physical communication between end stations. It is concerned with the actual encoding and transmission of data in electro-mechanical terms of voltage and wavelength [2,3,4,5,6]. For purposes of information security we can widen this definition to apply to all physical world factors, such as physical media and input device access, power supply, and any other issue bounded by physical terms.

As already mentioned, the physical layer is critical to data communications. It is also the most vulnerable and changeable, not depending upon the logic and organization of the electronic world, but on the vagaries of physics. Denial of Service is a mere circuit breaker or lead pipe away when dealing with the physical layer. Something as simple as unplugging the power or removing a network cable can cause untraceable havoc on a network. It should be noted that this is the most likely realm for accidental violation - who hasn't heard the classic story of a cleaning crew or intern pulling the power cord from a critical piece of production hardware? The physical realm is also the hardest to maintain an audit log or monitor. No level of logical or programmatic controls can easily detect that a host has been detached from its normal network connection and is now connecting through an Ethernet tap, which may be silently duplicating any inbound or outbound communications for eavesdropping purposes. As far as eavesdropping is concerned, physical contact may not even be necessary. In what is regarded as a seminal paper on non-intrusive electronic eavesdropping published in 1985, Wim Van Eck states the following - (emphasis added)

> "It is possible in some cases to obtain information on the signals used inside the equipment when the radiation is picked up and the received signals are decoded. Especially in the case of digital equipment this possibility constitutes a problem, because *remote reconstruction of signals inside the equipment may enable reconstruction of the data the equipment is processing*."[9]

What was groundbreaking about Van Eck's paper was not the possibility of such eavesdropping, as he states that this possibility has been well known for decades but dismissed as demanding a high degree of sophistication, specialization, and expensive resources. The noteworthy part was instead that techniques had been identified by him to allow such eavesdropping that were inexpensive, used common materials, and only required a moderate level of technical sophistication. Based on this paper, the term Van Eck Phreaking was coined to describe remote eavesdropping on the signals in a CRT or VDT display[10]. This term is referenced in the U.S. Government's classified Tempest project, which many believe was used to develop application for use of electromagnetic eavesdropping as well as protections against such intrusions.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Fortunately, physical security for information technology can benefit from the more general discipline of physical security in the general world. As the somatic components of information technology are subject to the same threats as other "real" assets, they are also able to benefit from the same protections that the more mundane security disciplines have implemented from the beginning of modern civilization. This means that critical assets must be behind strong locks, with strict controls on who may pass those locks, and constant monitoring, logging, and review of that access. Such monitoring may include video surveillance, card-lock logging of entry and exit with PIN-based passwords, and even biometric validation to augment password and hardware based credentials to validate actual physical identity. On the information technology side, data storage cryptography is an additional security control at the physical layer, allowing control of access to data even when the physical media or resource may be wholly in the control of unauthorized elements. The aforementioned Tempest project developed standards for electromagnetic shielding to prevent monitoring of highly sensitive systems such as PKI Certificate Authorities. Techniques have also been developed to modify screen fonts in ways that attenuate the signal emanated by a CRT displaying them, reducing the RF emitted in the critical ranges that Van Eck phreaking devices use to pick up their information [11].

Physical Layer Vulnerabilities

Loss of Power
Loss of Environmental Control
Physical Theft of Data and Hardware
Physical Damage or Destruction of Data And Hardware
Unauthorized changes to the functional environment (data connections, removable media, adding/removing resources)
Disconnection of Physical Data Links
Undetectable Interception of Data
Keystroke & Other Input Logging

Physical Layer Controls

Locked perimeters and enclosures
Electronic lock mechanisms for logging & detailed authorization
Video & Audio Surveillance
PIN & password secured locks
Biometric authentication systems
Data Storage Cryptography
Electromagnetic Shielding

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Layer Two - Data Link Layer

The Data Link Layer is concerned with the logical elements of transmissions between two directly connected stations. It deals with issues of local topology where many stations may share a common local media. This is the layer where data packets are prepared for transmission by the physical layer. The data link layer is the realm of MAC addresses and VLANs as well as WAN protocols such as Frame Relay and ATM. Switch issues such as broadcast and collision domains are a layer two concern. It is also the realm of wireless protocols such as the various flavors of 802.11 wireless networking [2,3,4,5,6]. For discussion purposes we will consider layer two to pertain to any direct data transmission issue, including modems, wireless and WAN circuits.

The Data Link Layer has been a long-neglected area of study for information security, lost between the physical issues of layer one and the dominating realm of the firewall in layers three and four. This lack of attention made it an area ripe for exploitation, and some of the hottest new issues in information security have heavy involvement in layer two. Wardriving, the act of traveling around public areas and randomly accessing 802.11 wireless access points with lax or default security settings is a prime example of a vulnerability with both layer one and two elements. The wireless hardware solution may have an initial goal of ease of deployment and use, but this goal is used as a weakness to exploit the solution for unanticipated purposes on the basis of the solution exceeding its anticipated physical boundaries (The wireless signal extends from the wireless access point's inside location out to the outside public street.) and lacking sufficient use of control at layer two by letting anyone with a signal at layer one to freely connect.

Due to its interaction with a variety of media and flavors of hardware, this layer is critical to network compatibility and as such is heavily dependant on rigid protocol standards for interoperability. This dependency can allow poorly designed standards to impede security, and make the correction of issues a ponderous and drawn-out process. In the aforementioned 802.11 scenario, there are tools available to secure the layer two issues, using encryption protocols to authenticate valid users and protect their traffic from unauthorized access. Unfortunately, weaknesses were found in this encryption scheme that have to date only been partially addressed.

Weaknesses have also been found in the much-touted Ethernet switch. Originally thought to be the answer to the problem of promiscuous mode sniffing of network traffic because of their learning and selective forwarding, switches have fallen victim to the efforts of creative hackers, who have been hard at work finding the means to circumvent this protection. Some of the key issues lie in the ARP protocol. This protocol establishes the relationship between local stations that can communicate over the layer two channel, and their corresponding layer three IP addressing. The ARP process is very basic, and has no means for authentication or validation. Any station in the local layer two environment can claim any IP address. ARP typically operates on a broadcast basis, but attacks against ARP have been developed using unicast transmissions to specific targets. Known as ARP spoofing, these attacks create an

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

artificial view of what the layer two environment looks like to specific targets, and allows man-in-the-middle attacks where an attacking machine intercepts the data communication between two hosts by intercepting their traffic and forcing it to bypass through the attacking machine. [13,14]

Layer two switches are also vulnerable to attacks on their virtual separation of segments known as VLANs. Recent vulnerabilities have been found in Cisco's automatic configuration of VLAN trunks, allowing hosts that can send 802.1Q trunking protocol signaling (an ability that is becoming more and more common in modern operating systems and NIC drivers) to negotiate access to multiple VLANs. Cisco provides configurations to disable this behavior, but the default behavior is to allow automatic VLAN configuration. [12]

As a newly emergent battleground, the threats tend to outweigh the controls on the link-layer, with the only strong tools being manual MAC filtering to enforce an explicit layer two policy, and strong network design to minimize exposure from the outset. The inherent design of most layer two communication imposes a layer of involuntary trust.

Link Layer Vulnerability Examples

MAC Address Spoofing (station claims the identity of another)

VLAN circumvention (station may force direct communication with other stations, bypassing logical controls such as subnets and firewalls.)

Spanning Tree errors may be accidentally or purposefully introduced, causing the layer two environment to transmit packets in infinite loops.

In wireless media situations, layer two protocols may allow free connection to the network by unauthorized entities, or weak authentication and encryption may allow a false sense of security.

Switches may be forced to flood traffic to all VLAN ports rather than selectively forwarding to the appropriate ports, allowing interception of data by any device connected to a VLAN.

Link Layer Controls

MAC Address Filtering- Identifying stations by address and cross-referencing physical port or logical access

Do not use VLANs to enforce secure designs. Layers of trust should be physically isolated from one another, with policy engines such as firewalls between.

Wireless applications must be carefully evaluated for unauthorized access exposure. Built-in encryption, authentication, and MAC filtering may be applied to secure networks.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Layer Three - Network Layer

The Network layer is concerned with the global topology of the internet work - it is used to determine what path a packet would need to take to reach a final destination over multiple possible data links and paths over numerous intermediate hosts. This layer typically uses constructs such as IP addresses to identify nodes, and routing tables to identify overall paths through the network and the more immediate next-hop that a packet may be forwarded to. Protocols such as ARP facilitate that process, giving layer two mapping to layer three addresses, and telling layer three what link-layer path should be taken to follow its routing table's indication of the appropriate path. In the opposite direction, protocols such as IP will identify their higher-level layer four transmission protocol such as TCP or UDP in order to direct layer four as to how the incoming data should be handled [2,3,4,5,6].

Layer three is the last layer that has a rough physical correspondence to the real world. A given host will typically have a single layer three address or single layer three address per interface. This tends to make layer three addressing critical not only to network topology but also to node identity. In a traditional firewall, the layer three address is the primary qualifying value in a filtering rule, with some rules using them as a sole identifier (examples - denying common RFC1918 "private" addresses or other address ranges designated as invalid. Denying inbound packets from the outside that claim a source address from an inside network - so called "packet spoofing") layer three addressing is also used by applications to identify resources, using DNS resolution to map a hostname to an address or group of addresses. Layer three protocols often have mechanisms for broadcast or multicast of data to multiple machines in finite or arbitrary scopes.

In filling these many roles, a variety of means for attack at layer three become exposed. In the realm of routing, especially public routing situations such as over the Internet, most routing protocols have only an elementary level of security. Two peers may exchange routing information securely, but they have no means to validate routes that may have propagated from untrusted parts of the network. Attackers can steal entire network ranges with the right resources, allowing further attacks at layer three and above [15]. Identity is always a classic vector for attack - most layer three protocols have no built-in means to authenticate source addresses or other protocol data which may be used to attempt to establish identity, so when we rely upon what a packet claims to be a source address, we have little reason to actually expect that address to be correct. Resource identification falls victim to the same lack of authentication - DNS servers can be forced to present incorrect addresses, or by routing or the earlier ARP spoofing techniques, an illicit host can take a given address and claim to be the resource which is located therein. Techniques have also been developed to abuse broadcast mechanisms, amplifying data into crushing streams of packets that can paralyze a host, often using untraceable spoofed addressing against unsecured third party machines which are turned into unwitting tools for abuse.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

The ubiquitous control for layer three is the firewall - when correctly configured it will let only the necessary traffic pass through its boundaries. However, well-thought out policies that take into consideration the problems of identity must be part of the firewall deployment. Encryption and authentication technologies such as IPSEC can be used to more reliably identify the source of IP communications. Routers must have strict policies regarding their exchange of routes, and use reliable means of authentication and communication with their peers. Route filters should be applied to prevent the accidental or intentional introduction of spurious network routes. On the Internet, Route Registries and the Routing Arbiter Database (RADB) offer the means to register route announcements. The RADB also provides filter information that allows building of local policies to validate foreign route announcement.

Network Layer Vulnerabilities

Route spoofing - propagation of false network topology
IP Address Spoofing- false source addressing on malicious packets
Identity & Resource ID Vulnerability - Reliance on addressing to identify resources and peers can be brittle and vulnerable

Network Layer Controls

Route policy controls - Use strict anti-spoofing and route filters at network edges
Firewalls with strong filter & anti-spoof policy
ARP/Broadcast monitoring software
Implementations that minimize the ability to abuse protocol features such as broadcast

© SANS Institute 2004, Author retains full rights.

Layer Four - Transport Layer

The Transport Layer is concerned with the transmission of data streams into the lower layers of the model, taking data streams from above and packaging them for transport, and with the reassembly and passing of incoming data packets back into a coherent stream for the upper layers of the model. Transport protocols may be designed for high reliability and use mechanisms to ensure data arrives complete at its destination, such as the TCP protocol, or protocols may choose to reduce overhead and simply depend upon the best efforts of the lower layers to deliver the data, and the protocols of the upper layers to ensure success to the levels they require, such as with the UDP protocol. Transport protocols may implement flow control, quality of service, and other data stream controls to meet their transmission needs [2,3,4,5,6].

The Transport Layer is the first purely logical layer in the model. It is the primary point where multiple data conversations from or to a single host are multiplexed. Some transport protocols such as TCP and UDP use the concept of port numbers to allow multiple simultaneous conversations between numerous destinations to individual local protocols or applications. Other protocols such as ICMP might rely on higher-layer data to sort out multiplexing[*]. Because the transport layer is where data conversations to a given host are multiplexed and sorted, it is often used as the primary means of service identification within a given host, much as how layer three addresses are used to identify service locations within the context of the entire network.

Some of the key vulnerabilities found at the transport layer come from poor handling of undefined conditions. Many transport protocols seem to have been implemented under the belief that they would be dealing with well-behaved communication from both the upper and lower levels - a false assumption in the hostile world of the global public Internet. This means that protocols are subjected to unexpected or deliberately perverse input or handling exploiting the more obscure protocol details and so-called impossible conditions, and as a result often have unexpected behavior. Attacks such as Winnuke used an obscure and out-of-specification TCP flag when connecting to an open TCP port on a Windows machine, and the result was an operating system crash [15]. The behavior of a given host when presented with TCP and UDP packets with varying arbitrary contents can be used to "fingerprint" an OS and select more focused attacks due to differences in response between different operating systems and network stacks.

Another vulnerability lies in the use and re-use of ports for multiple functions. This is found quite often in the Windows arena, where differing functions such as file and print sharing, remote administration, LAN messaging, RPC functions, and a myriad

---

[*] *Some interpretations of the OSI model put protocols such as ICMP at layer three, as their use is primarily geared toward layer three issues.[6] In this paper, I layer all protocols at where the function of their typical implementation puts them in the stack. The actual protocol details of ICMP operate at layer four; it is a transport protocol identified in the IP header protocol field (IP protocol 1), and the ICMP header in general describes a modest transport function. If you take the viewpoint that a packet is a series of wrappers that the various layers apply, the ICMP header clearly occupies the layer four wrapper position.*

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

other applications all use a handful of UDP and TCP ports. This overuse of ports makes restriction of access at layer four by a firewall difficult. If any of the functions are needed, then the firewall ports are opened and in theory most if not all functions that use those ports could flow through unchecked. Imagine the surprise of a firewall administrator to open a port on a perimeter firewall supposedly for the purpose of authentication or drive sharing, only to have messenger-based spam advertising or remote vulnerabilities let in by the same rule. This overloading limits the effectiveness of network-based controls such as firewalls, and forces reliance on individual host level security controls, which are often not a practical proposition in large enterprise environments with a large amount of machines operated in many different administrative environments and functional roles.

Most transmission protocols were built with an emphasis on utility and performance. As such, they usually do not implement strong controls to validate the source of a transmission, or that a packet is a legitimate part of a data conversation. This leads to the ability to forge packets that can interrupt or redirect the flow of a transmission. Some protocols such as UDP can be trivially spoofed and fooled due to a complete lack of sequencing or state at layer four. Other protocols such as TCP are more difficult due to their more extensive flow control and integrity checking. However, with most such protocols, integrity pertains more to the accidental loss of data due to errors or packet loss rather than the deliberate attempt to attack the protocol. Thus such protocols can also fall to more sophisticated attack. The practice of TCP session hijacking is one such sophisticated attack, where the attacker must guess factors such as initial and TCP sequence numbers, and then inject fake packets to manipulate the data flow by interrupting then falsifying the flow of higher-level data. Such an attack is one-way-control may be gained but information does not return to the attacker unless he uses the control channel to open additional covert channels of attack.

Conventional firewalls are the most common control at layer four as well as layer three. Firewall rules should be written to be as strict as possible regarding transport layer identity. This means that transport layer protocols should be specified individually in rules where possible rather than permitting any communication between two layer three nodes. In terms of TCP/IP communication, this means that rules should be written applying matches for layer four protocols such as UDP/TCP/ICMP as well as sub-protocol details such as UDP/TCP port numbers or ICMP types. Modern firewall technology allows for “stateful inspection”, which allows firewalls to inspect the layer four details of a packet and determine the state of a transmission at the transport layer. This allows the firewall to determine if a packet is likely to be in response to an existing flow of data rather than a random packet trying to “sneak by” based on all aspects that govern flow in a given protocol, rather than a more arbitrary packet filter that may only check port number or simple flags which may be easily determined and set in a arbitrarily assembled packet.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Stronger mechanisms are possible in layer four implementations to make session hijacking more difficult as well. Recent improvements in TCP sequence number assignment based on random number generation rather than arbitrary and predictable sequences have made the blind takeover of TCP sessions much more difficult. The Cisco PIX firewall provides a randomized TCP sequence number to traffic it passes as part of its NAT-based Adaptive Security Algorithm (ASA) [16], fixing the problem for TCP implementations which are still non-random and predictable.

Transport Layer Vulnerabilities

Mishandling of undefined, poorly defined, or "illegal" conditions

Differences in transport protocol implementation allow "fingerprinting' and other enumeration of host information

Overloading of transport-layer mechanisms such as port numbers limit the ability to effectively filter and qualify traffic.

Transmission mechanisms can be subject to spoofing and attack based on crafted packets and the educated guessing of flow and transmission values, allowing the disruption or seizure of control of communications.

Transport Layer Controls

Strict firewall rules limiting access to specific transmission protocols and sub-protocol information such as TCP/UDP port number or ICMP type

Stateful inspection at firewall layer, preventing out-of-state packets, "illegal" flags, and other phony packet profiles from entering the perimeter

Stronger transmission and layer session identification mechanisms to prevent the attack and takeover of communications

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Layer Five- Session Layer

The Session Layer is concerned with the organization of data communications into logical flows. It takes the higher layer requests to send data and organizes the initiation and cessation of communication with the far end host. The session layer then presents its data flows to the transport layer below where actual transmission begins. Session protocols will often deal with issues of access and accessibility, allowing local applications to identify and connect to remote services, and advertising services to remote clients and dealing with subsequent requests to connect. The session layer also deals with higher-order flow control from an application perspective; just as the transport layer may control transmission from a network-oriented perspective and limit the flow to match the available network capacity, the session layer may control the flow up through to the application layer and limit the rate that data enters or leaves that realm based on arbitrary or dynamic limits [2,3,4,5,6].

The Session Layer in networking is a more obscure topic because it is fairly neglected in the TCP/IP communications model that dominates modern data communications. The Department of Defense model for TCP/IP essentially compresses the ISO Session (layer five), Presentation (layer six), and Application (layer seven) layers into a process/application layer. As both models are frameworks for design rather than unbendable standards, in implementation many TCP/IP based protocols break out into what can be classified as Session Layer behavior. Common examples include network utility protocols such as RPC, Microsoft's .NET system, and CORBA, which create frameworks for higher-level applications to sort out the availability and use of resources distributed over a network. Secure authentication protocols such as SSL and Kerberos have specific function in the session area, negotiating and controlling the flow of information for higher-level applications. On the other hand, many applications encapsulate the session functions in their application protocols. Basic network tools such as FTP and Telnet negotiate sessions within their own protocol boundaries. Also included in the Session Layer are multimedia protocols such as H.323, VoiceOverIP protocols such as SIP, and other media-streaming and communication protocols. These protocols often are required to negotiate both session creation and session-path parameters such as quality-of-service and bandwidth. [6]

As the Session Layer deals with the creation and control of access to the higher-level applications, the issue of authorization and access is a natural weakness in this layer. Similar to problems we've seen already in lower-layer protocols, multiplexing services such as RPC, .NET and CORBA which provide a wide range of services through a single channel narrow the ability of lower layers of the network to control access to resources. If these protocols themselves do not provide robust security internally they become a prime target for abuse. Even if they meet this challenge at best they still remain a single layer of protection. Many session layer protocols lack strong protection for their authorization facilities. Protocols such as standard telnet and FTP pass usernames and passwords in the clear, allowing any layer beneath them to intercept their credentials. Protocols with "stronger" protection of passwords such as the Microsoft implementation of CIFS (Common Internet File System, used by MS for

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

file and printer sharing) often fall prey to cryptographic or implementation weaknesses in the handling of passwords and authentication [18].

Even assuming perfect protection of a user's credentials in transit over the network, it is all too common that the passwords themselves are weak and subject to attack. The session layer can exacerbate this problem by poor or non-existent logging of failed access attempts, allowing an attacker unlimited and undetected attempts to guess likely passwords, or use an even cruder technique of brute-force exhaustion of all possible or probable password strings. Mechanisms within the session layer can also be used to enumerate possible usernames to pair with such guessing and brute-force attacks. On traditional Unix-based systems there were often services such as finger or rwho enabled to show logged-in users and associated information such as where a remote user may be logged in from or how long a user may have been idle. On Windows machines it is common for the Messenger network service to be activated as a default option. This service is intended to receive "net send USER MESSAGE" messages sent remotely to the local users. In order to make message services available, this service advertises a logged-in user name to any node status request for service names on the machine, allowing the use of such queries to enumerate user names remotely.

Some services such as SIP and other Voice-over-IP protocols share similar problems to those discussed at layer four with the identification of valid traffic within a session. Mostly based on UDP for performance and overhead purposes, these protocols often still must implement transport session identification such as packet sequence in layers five and above. These sequencing and flow identifiers are just as subject to guessing and attack, in many instances more so due to the open nature of UDP and the emphasis on low overhead and functionality common for real-time protocols.

As discussion has shown so far, identity is an issue with all layers in the model, with the attributes of each layer often applied as a standard for identification and authorization. In the Session Layer, identity is the key factor, and the main controls at this layer focus on the establishment of identity. Secure channels of user and session authentication are essential to private communications. Cryptography technology allows for both the reliable identification of remote parties and the means for protecting the exchange of data from prying eyes. Passwords and other user credentials should be passed and stored in encrypted form to prevent interception or theft. User accounts should have expiration dates based on both usage and fixed time, requiring the update of credentials and reauthorization of access. Session identification may need to be based on a cryptography technology in order to protect sensitive communications in real-time environments.

To prevent brute-force or focused guessing of session credentials, failed attempts can be properly logged and limited to a fixed amount of failures before an account or service is locked out. This approach is a two-edged sword in that legitimate users may be locked out by illicit access attempts either inadvertently or as the basis for

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

a denial-of-service attack. A safer possible approach is to limit connection attempts on a time basis such as only once every 30 seconds, or temporary lockout on failure with a brief enough duration that legitimate user access will recover in a practical amount of time, but a brute force attack would be rendered impractical.

Session Layer Vulnerabilities

Weak or non-existent authentication mechanisms
Passing of session credentials such as user ID and password in the clear, allowing intercept and unauthorized use
Session identification may be subject to spoofing and hijack
Leakage of information based on failed authentication attempts
Unlimited failed sessions allow brute-force attacks on access credentials

Session Layer Controls

Encrypted password exchange and storage
Accounts have specific expirations for credentials and authorization
Protect session identification information via random/cryptographic means
Limit failed session attempts via timing mechanism, not lockout

© SANS Institute 2004, Author retains full rights.

Layer Six- Presentation Layer

The Presentation Layer deals with the organization of data passed from the application layer into the network. This layer allows for the standardization of data and the communication of data between dissimilar hosts, such as platforms with different binary number representation schemes or character sets (ASCII vs. UNICODE, for example.) Presentation Layer protocols typically rely upon a standardized data format for use on the network, and various conversion schemes to convert from the standardized format into and out of specific local formats. The Presentation Layer can also control network-layer enhancements such as compression or encryption [2,3,4,5,6].

The Presentation Layer is another obscure layer, usually hidden deep in the implementation of applications and operating systems. End users may need to be passingly familiar with the outermost details such as JPEG picture formats or SSL encryption, but leave the details to their applications to sort out. For discussion purposes the Presentation Layer can be considered to be all interfaces implemented by or called from an application to prepare and present data to the network for transmission, such as program library functions that take data and re-order into "network-byte" order. Taken more broadly for an information security viewpoint, we can evaluate at this level any system or library functions that take data input and process it into standard formatting or condition for all purposes. Many applications make use of SSL and TLS libraries to secure communications, where secure in this context means the use of strong authentication, data encryption, and other assurance functions based on cryptography. These features are nearly always presented to the application layer as an API into a standard library of functions. Some presentation services may use separate channels to control passage as well as directly pass data, allowing control at from higher levels of the presentation of the data stream.

Vulnerabilities at this layer often originate from weaknesses or shortcomings in the implementation of the presentation layer functions. Continuing on the theme of taking advantage of the original atmosphere of implicit trust and simple functionality that systems were (and continue to be) built in, attackers feed unexpected or illegal input into presentation-layer facilities, gaining results that are undesired or contrary to what the original designers intended. Buffer overflows, where program execution can be redirected into completely unintended areas, can be classified at this level as a problem with the presentation of data by an application into the execution environment of the machine. When the presentation from the application exceeds or mismatches the required convention at the presentation layer, unexpected events can happen.

A recently recognized weakness known as format string vulnerability can also be classified in the Presentation Layer. Format string vulnerabilities take advantage of applications that use user-supplied information for the basis of input into I/O libraries in such a way that the user-supplied data stream could control how that data is transmitted, formatted, or stored in the process of transmission. This occurs due to either the direct or indirect use of the user input in the format portion of routines used to process the data. Many routines allow this type of use for unformatted, simple output.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

The assumption is that the user input is passed unformatted and verbatim through the routine. The actual result however, is that the user has access to pass control information through his data channel, and can use this access to crash the program, control its execution, or display arbitrary information on the other end of the output stream.[19,20]

Cryptographic presentation services can fall prey to weaknesses in their implementation or fundamental design. Many secure web servers using SSL have had subtle bugs in the underlying cryptography of the SSL implementation turn into either theoretical or practical security exploits.

Controls at the Presentation Layer will typically take the form of cautious and untrusting coding practices when using routines and facilities for network and other inter-process communications. Checking and rechecking input from both the network and the user/application for proper form, and not relying upon lower/upper layers to present properly formed data is a must in an environment thick with arbitrary and deliberately perverse manipulations of communications. Assumptions that ease implementation or time-to-market for systems that shortcut this process may end up being disastrous from a security perspective. User and peer input should always be highly suspect, whether the input is received from a remote station or a local user. Careful specification is needed to determine what is expected from input, and code that carefully checks that input is needed to enforce that specification.

Cryptography is a fast-moving target, and technology and hardware capabilities advance constantly. The cryptographic strength of data protection services in the presentation layer should be selected carefully and reviewed periodically. Many cryptography protocols have been found to have subtle flaws well after being declared secure, so a process of periodic re-evaluation of crypto solutions is also vital.

Presentation Layer Vulnerabilities

Poor handling of unexpected input can lead to application crashes or surrender of control to execute arbitrary instructions.

Unintentional or ill-advised use of externally supplied input in control contexts may allow remote manipulation or information leakage.

Cryptographic flaws may be exploited to circumvent privacy protections

Presentation Layer Controls

Careful specification and checking of received input incoming into applications or library functions

Separation of user input and program control functions- input should be sanitized and sanity checked before being passed into functions that use the input to control operation

Careful and continuous review of cryptography solutions to ensure current security versus know and emerging threats

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Layer Seven- Application Layer

The Application Layer deals with the high-level functions of programs that may utilize the network. User interface and primary function live at this layer. All functions not pertaining directly to network operation occur at this layer [2,3,4,5,6].

Occupying the top-end of the stack, the Application Layer is the most open-ended of all of the layers, and can be considered the catchall for any issues not addressed within the other six layers. Taking a more narrow view from a protocol-perspective, user-oriented protocols such as naming (DNS, WINS), file-transfer (HTTP, FTP), messaging (SMTP, TOC/OSCAR[used by AIM]), and access (Telnet, RDP) all fall within the Application Layer in a more strict interpretation that views even higher level functions as outside the model completely. For the purposes of information security, the Application Layer can be considered the realm where user interaction is obtained and high-level functions operate above the network layer. These high level functions access the network from either a client or server perspective, with peer-based systems filling both functions simultaneously.

Similar to the physical-layer, the open-ended nature of the Application Layer groups many threats together at its end of the stack. One of the prime threats at the Application Layer is poor or nonexistent security design of the basic function of an application. Some applications may insecurely handle sensitive information by placing it in publicly accessible files or encoding it in “hidden” areas which are trivially displayed, such as in the HTML code of a web form. Programs may have well-known backdoors or shortcuts that bypass otherwise secure controls and provide unauthorized access. Applications with weak or no authentication are prime targets for unauthorized use and abuse over the network. The TFTP protocol is extensively used for booting of diskless workstations and network device management, but does not require any sort of username or password authentication to use its file access ability, giving an intruder possible access to configuration and access information without challenge other than the need to guess filenames. (This could equally be considered a session-layer vulnerability, or the failure to use session-layer controls.) Applications may rely upon untrustworthy channels to establish identity or set privilege. The Unix rlogin, rsh, and other “r-functions” typically use the implied trust of a local list of trusted remote hosts and remote users without a strong means of verifying the identity this trust is based upon. DNS names can be spoofed or DNS servers can be compromised, and compromised or maliciously controlled remote hosts can report whatever user identity is desired. Even worse, universal trust could be established to any user from any machine by placing the string “+ +” or “+” in specific system configuration files. This “trust the world” profile was shipped as the default on some commercial Unix systems.

Applications often grant excessive access to resources, allowing unprivileged users excessive access or imposing inadequate control to prevent the corruption or loss of data. The lack of detailed controls lead many data systems to have access granted on an “all or nothing” basis, forcing administrators to either give unlimited access or

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

none at all. Overly complex access controls may seem to protect access but fail to prevent unauthorized activity due to poorly understood or written access rules.

From the higher levels outside of the model, user input is a significant threat from both deliberate and accidental standpoints. Users may provide unexpected input into the application environment, which if not handled properly could lead to crashes or other unexpected behavior. The unsuspecting hapless user may cause his application to crash or otherwise fail. A malicious user may be able to use bugs and program flaws to attack and gain access to resources or data.

Some of the most prevalent controls at the application layer relate to strong design practices in application design and implementation. Applications should make use of the secure facilities available to them in the lower network layers, carefully check incoming and outgoing data, and assume that communications can and will be subject to attack, requiring the use of strong authentication and encryption to validate and protect data as it travels across the network. Applications should also implement their own security controls, allowing for fine-grained control of privilege to access resources and data, ideally using a mechanism that is straightforward and strikes a balance between usability and effectiveness. Detailed logging and audit capability should be a standard feature of any application that handles sensitive or valuable data.

Testing and review is also critical as a control for the application layer. Given the wide variety of both problems and solutions, standards and practices will not be able to capture all possible twists and turns in the application environment. Developers will often have conflicting motivations and agendas regarding their applications, and in a structured programming environment, mandated code security review and application security testing are critical parts of a secure Software Development Life Cycle (SDLC).

On the hardware front, Intrusion Detection Systems (IDS) can observe data traffic for known profiles of network activity that can indicate probes for vulnerable applications or an imminent or ongoing attack, as well as detecting the presence of undesirable application traffic.

Many current host-based firewall systems also include the means to control the access of applications to the network. This control is useful in preventing the unauthorized or covert use of network resources by local programs, as well as providing the conventional layer three and four control functions of a firewall. Many also include basic IDS functionality as well.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Application Layer Vulnerabilities

Open design issues allow free use of application resources by unintended parties

Backdoors and application design flaws bypass standard security controls

Inadequate security controls force "all-or-nothing" approach, resulting in either excessive or insufficient access.

Overly complex application security controls tend to be bypassed or poorly understood and implemented.

Program logic flaws may be accidentally or purposely used to crash programs or cause undesired behavior

Application Layer Controls

Application level access controls to define and enforce access to application resources. Controls must be detailed and flexible, but also straightforward to prevent complexity issues from masking policy and implementation weakness

Standards, testing, and review of application code and functionality-A baseline is used to measure application implementation and recommend improvements

IDS systems to monitor application inquiries and activity

Some host-based firewall systems can regulate traffic by application, preventing unauthorized or covert use of the network.

© SANS Institute 2004, Author retains full rights.

Interlayer Communication & Extensions to the Model for Information Security

The presentation of the layer-by-layer issues has focused on elements within each layer. However, the purpose of the seven-layer model is communication, and communication between layers is where many key issues lie. Understanding the layers individually now gives us a framework to examine how they communicate between one another, between matching layers on remote hosts, and how they contribute to the communication of data over the network.

An alternate way of organizing the seven-layer model is by concentrating on the borders of the layers rather than by examining what each layer encompasses. This approach focuses on the communication between the layer above and the layer below. Even the formal definitions presented earlier for each layer sometimes touch on these issues, and the actual implementations of network protocols are rarely as clean as the theoretical model. An example of such an issue is the process of ARP where a layer three protocol (IP) communicates into layer two to associate a remote MAC address with an IP address the local host is seeking to transmit to. There is a definite communication between the two layers to negotiate the issues required to match up a MAC address to its corresponding IP address and transmit the outbound data. The IP layer maintains a MAC to IP table based on this gathered information, essentially tracking and following layer two details at layer three.

To draw a parallel in the information security realm, multiple hosts secured behind firewalls on a single subnet present some interesting layer two vs. layer three problems. The firewall may be able to prevent outside subnets from communicating into the local subnet, but will not be able to prevent communication between the hosts on the local subnet. The hosts will be able to use ARP and discover one another, and packets will be forwarded across the layer two environment directly between hosts. One solution to this problem is to segregate hosts on to their own dedicated subnets. This approach is highly secure but is inefficient in that it wastes IP address space; each subnet requires reserved addresses for broadcast and network identity, as well as an address on each subnet for the firewall itself. This approach would also waste physical interfaces on the firewall and switch environment. The option to use VLAN trunking is sometimes available, but is a poor choice in a secure environment based on potential vulnerabilities such as those mentioned in the layer two analyses.

There are options to secure the hosts on the shared subnet. Static ARP entries can prevent traffic interception, and many Ethernet switches offer options to secure their ports in order to limit traffic or MAC addresses to pass only into authorized ports. These solutions must be carefully deployed, however, as the layer two environments may unexpectedly pass traffic or create unobvious relationships. Take as an example a scenario the author has encountered in an active production network; An Internet-facing DMZ is secured via the private VLAN function provided by switches made by a particular hardware vendor. This feature allows the designation of a port as “secure.” This designation prevents communication from any other port also designated as “secure”. At least it did in theory. In practice it was found that “secured” hosts were still

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

communicating in some instances. The problem was found in the fact that multiple switches were deployed in the environment to support redundancy and reduce the risk of outage should a single switch fail. The trunk between the switches could not be designated as private in either direction as the high-availability firewall pair serving as a gateway onto the DMZ subnet had a firewall attached one each to both switches, and hosts on one switch may need to traverse the connection between the switches to reach the firewall attached to the adjacent switch should that particular firewall be primary in the HA pair at that moment. Of course, once the packet passed through the switch trunk, it was no longer on a private port, and it was eligible to be forwarded to any port, "secure" or not. In essence, this can be considered a physical layer issue in that the policy control of a specific port's privilege does not extend past the boundaries of a single switch. A more robust control perhaps would account for the fact that multiple switches may require a uniform access policy across their shared environment.

Network Address Translation (NAT) is a special case in the seven-layer model, and is an excellent way to introduce how a clean theoretical model starts to get messy in actual implementation. On its surface, it is a straightforward function. NAT devices divide the network into two areas: those on the "outside" and those on the "inside". When packets enter a NAT-enabled device, rules are applied based on source and destination IP addresses of the packets as well as direction of traffic (heading "outside" vs. heading "inside"). Rules may dictate that individual given addresses appearing as either sources or destinations translate to different addresses on a one-to-one basis. This is often referred to as static NAT, and is often used to reference servers or other resources that must be consistently accessed by a consistent IP address. Another option is for NAT rules to define ranges or conditions (such as "all inside traffic") and apply a pool or single address to all traffic that matches the rule. This is referred to as dynamic NAT, and is usually used to translate traffic that does not require a consistent address or where details of individual hosts in the range are not relevant or should not be divulged.

Static NAT is a simple matter of finding data packets that match the rule, and rewriting the packet to the parameters of the NAT rule. This typically means replacing the layer three address with a translated one. Dynamic NAT is trickier in that multiple inside hosts may have the same outside address. The NAT device must have some way to identify traffic, and so must maintain a table of layer three source and destination addresses along with layer four information, such as UDP/TCP source and destination port numbers. The layer four information is critical for sorting out the condition where two or more inside hosts connect to the same destination address/port with the same ephemeral source port.[*] This means on the outside area of the NAT translation where the two inside hosts appear as one source address, there may be two identical connections sent to a single destination. The server side would not be able to multiplex these connections properly, and the NAT device would not have a deterministic way to

[*] *Ephemeral port* is a formal term for the arbitrary port assigned automatically as a source port from a range designated by the operating system and TCP/IP stack for use by clients requesting a connection when a source port is not specifically specified by the application. This is typical behavior for a client application when initiating a remote connection.[21]

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

separate return traffic for one host versus another when ephemeral ports. Thus, with dynamic NAT, layer four ports must also be tracked and occasionally translated in order to avoid collisions.

In addition to the logistical issues at layers three and four, there is often a challenge presented to NAT at layer seven. Some applications may embed information pertaining to network issues in the application layer of the network conversation. This is usually used functionally to identify remote hosts or set up additional communications. The problem arises when such an application attempts connection through a path subject to NAT translation. If the NAT implementation simply rewrites the layer three and four information, there will now be a divergence between the layer three and four information in the packet which are critical to get correct for proper return communications, and the layer seven data which reflects the application's viewpoint of an untranslated inside address. The classic example of this behavior is the FTP protocol. In simple terms, FTP uses two channels of communication to conduct business - A control channel which is opened by the client to initiate a session and issue instructions such as requests to send files or list directories, and one or more data channels which are opened and closed at the point where files or other bulk data need to be passed across the network. The control channel will open the data channel by specifying an address and port the FTP server is to initiate the data connection to. If this transaction were to be attempted through a NAT translation, the control channel would likely specify an unreachable or otherwise invalid IP address for the FTP server to connect to, and the data transfer would likely never occur. Applications such as the example FTP, SNMP, IRC, SIP, and others, as well as quasi-applications such as IPSEC and DNS all have issues with NAT [22]. These issues typically require intelligence to be written into the NAT implementation to inspect and translate the application-layer data to be in sync with the layer three and four translations occurring elsewhere.

The examination of NAT shows a network issue that extends over multiple model layers in both its problem set and the possible solutions examined. Information security problems and solutions often can be taken in this same multi-layer approach. In the layer description sections, many vulnerabilities and controls were covered that address single issues in a single layer. As was shown with NAT, the network often must be considered in its entirety to root out potential problems. Controls that impose policy at the lower layers may be ineffective against higher-layer problems due to the intrinsic transparency of the network layers. A firewall may operate at layers three and four when imposing a policy that says only connections to port 80 may go to a certain IP address, giving us a solid control for these layers, but if unpatched or poorly written software lurks at the other end of that connection, or if session controls depend upon a weak password, our packet-filter firewall will be helpless to impose further control on the security vulnerabilities that occur at higher levels. Stateful inspection engines and so-called intrusion prevention devices can possibly mitigate some of the risk posed by these higher-layer threats, but only to a certain extent. Meantime, other risks dwell on the bottom-end of the model. A firewall is effectively useless if network or other types of connections physically bypass it. A common network design flaw is to traverse a

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

firewall on administrative or utility networks, or to implement the different areas of trust divided by the firewall on the same physical switch using VLANs. Those VLANs may be subject to attack and circumvention. On full examination of the multi-level threats, the IT manager's common assertion that "our network is secure, we have a firewall" is beginning to sound a little suspect. A thorough security review must always consider the whole situation, and the derived security solution must address those concerns.

Another area of interest for examination of multi-layer issues is the world of wireless networking. Removing the physical barrier of cabling and connection has created an environment that seriously tests some of the assumptions that traditional networking has been built upon. There was never a concern prior to wireless that some nerd with a laptop that cost more than his car would park said laptop and car outside your house and use your Internet connection as he pleased. With the advent of 802.11b, once the problem of getting a wireless network card to easily function at distance with a minimum of configuration for the user was solved, a new problem developed; how do we keep everyone else's wireless network cards from also working so easily? The radio signals from wireless devices easily travel, especially when boosted by specialized antennas, making layer one physical separation very impractical to enforce. The layer two identities of various stations on the wireless network could be easily learned and emulated, making any sort of pure layer two control marginally effective at best. In the wired world, measures such as MAC filters often depend in part upon the mapping of physical attributes such as physical port number to the layer two attributes used as the basis for the control. There are no such physical attributes to match to in the wireless realm. The 802.11 basic standards have some controls for associating such as a Service Set Identifier (SSID), but ease of configuration concern has mostly removed this as a barrier. Concern for these security issues prompted the inclusion of encryption technology for authorization and privacy at layer two, but numerous problems have been found in the encryption technology chosen for the standard and as a result there remains no solid layer two security control. Even if there were, wireless manufacturers want to make their customers happy, and the average customer would rather a wireless device work right out of the box than be secure. Thus almost all wireless devices ship with the modest protection they have disabled [24].

Thus wireless users seeking security have been forced to look elsewhere for their security solutions. Wireless networks are defined as untrusted and connected behind firewalls, and solutions such as layer three IPSEC VPNs or layer six/seven application encryption such as SSL/TLS are used to provide more secure authentication and privacy to valid wireless user sessions.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Extending the model - The Infosec Nine-Layer Model

The seven-layer model is more than adequate for network purposes, but when used in the context of information security there are concepts that need organization that sit outside of the conventional network model. In a short web article, Steve Crutchley points to two additional aspects that are central to evaluation of an information security posture - People and Policy [25]. Crutchley proposes to add these two elements to the model as two additional layers, with people interacting with applications at layer eight, and policies controlling the actions of people (in theory) at layer nine. This proposal bears further examination.

Placing the user at layer eight may at first seem counterintuitive - the natural assumption would be that the user would sit at the top of the stack, controlling all that lies below. The experienced information security engineer knows however that a user's actions should always be guided by well-organized and carefully developed policies. We know that the user will not always conform to this policy, but in those instances the policy also exists to define the boundaries once they are crossed.

In keeping with the layer model approach, we can look at the ideal security policy as also being independent of the layers below; your policy should apply across all platforms and applications independent of the specifics of the application, and apply and fit with all classes of users, from the anonymous Internet user up to the most trusted administrators and officers of the company. Of course, in actual implementation there would always be layer-specific details, much as with the network layer model.

A problem with applying the policy layer at level nine is that it implies control over the user layer that then independently operates within the framework of the other layers. This does not address how policy should directly apply to the first seven layers. Of course the argument can be made that the users indeed must implement the policy at all layers, using firewall manager applications to set filtering rules, and plugging in devices in the correct physical scheme. This allows us to preserve the communication flow in the model, but could be considered a little too abstract. An alternative viewpoint is that the top layer of policy actually covers all layers like an umbrella, stretching around as well as above to speak to the identification of perceived threats and risk at each specific layer, and prescribing specific requirements for controls to be implemented at each layer.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Conclusion

Looking back, we have covered a lot of ground in the examination of the OSI Seven-Layer Model and its use as an information security tool. Having extended it with the inclusion of users and the policy they operate under, we have in many senses covered the entire spectrum of data assurance. In both the standard network context and in the extended context of information security, the seven-layer model is better applied as a tool for organizing concepts and scenarios rather than as a conceptual straightjacket. Examples from both worlds show that exceptions and variances to the model are sometimes desirable.

In the security context, the nine-layer model can be applied to assess both strengths as weaknesses, just as the individual layer examinations gave examples of both vulnerabilities and controls. This paper is nowhere even close to an exhaustive examination of these issues, and the discussed specific issues barely scratch the surface of the possible depth of issues. Hopefully the examples and discussion leave the reader with his own conclusions to pursue - The whole idea of a conceptual model such as this is not to describe the entire landscape of the modeled issue, but instead to serve as the foundation for further refinement and extension. The curious reader should now be prepared to address his own information security problems and solutions with the additional perspective of how those problems relate in a multi-layer perspective.

A point of interest is that the ISO group developing the OSI model also developed a security model to compliment the network interconnect model. This model is orthogonal to the seven layers of the OSI model however, specifying security services and mechanisms such as access control, authentication, data-integrity, and encryption as mapping three-dimensionally against the OSI network model [26]. This model in practice has been found to be complex and arbitrary, and lacks the conceptual elegance of the seven-layer model in how the layers interact. ISO leaves the service and mechanism interaction to the implementer, inviting a pick-and-choose approach that lacks coherence. In evaluating the topic for presentation in this paper, it was the author's feeling that a model that parallels the OSI network model was a stronger tool for organizing and evaluating the security environment, as well as being more approachable from the user perspective.

In the practical world of networking and security, there is a tendency to gravitate to things that work best. The pervasiveness of the OSI model in network discussion, design, and description is a strong testament to its functionality. Most information security practitioners already apply the concepts of the model, and its terminology is embedded in the industry jargon that surrounds both network and security hardware and software. Hopefully the concepts presented in this paper contribute to a better understanding of how closely related information security and data networking are related, and how the understanding of one topic is essential to success in the other.

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

References

[1] E Cole, J Fossen, S Northcutt, H Pomeranz, "SANS Security Essentials with CISSP CBK", SANS Press, 2003, p4, p52

[2] J Patterson ("InetDaemon"), "Theory – OSI Model", 2002
URL:http://www.inetdaemon.com/tutorials/theory/osi/index.html

[3] M Egan, "ISO OSI 7 Layer Model forced with TCP/IP", 1999
URL:http://mike.passwall.com/networking/netmodels/isoosi7layermodel.html

[4] R Feig, "The OSI Reference Model", Date Unknown,
URL:http://www2.rad.com/networks/1994/osi/osi.htm

[5] H S Ha, "OSI 7 Layer", 2000
URL:http://userpages.umbc.edu/~hha1/network/protocol.html

[6] J Drabik, "The Seven Faces of the OSI Network Model: Parts 1 & 2", April 2003
URL:http://www.commsdesign.com/design_corner/OEG20030415S0027
URL:http://www.commsdesign.com/design_corner/OEG20030416S0015

[7] E Cole, J Fossen, S Northcutt, H Pomeranz, "SANS Security Essentials with CISSP CBK", SANS Press, 2003, p257-258

[8] S Northcutt, L Zeltzer, S Winters, KK Fredrick, RW Ritchey, "Inside Network Perimeter Security", New Riders, 2003, p 4

[9] W Van Eck, "Electromagnetic Radiation From Video Display Units: An Eavesdropping Risk?", in North-Holland Computers & Security 4, 1985, p269
URL:http://web.archive.org/web/20000830130750/www.shmoo.com/tempest/emr.pdf

[10] whatis.com, "Van Eck Phreaking", Date Unknown
URL http://whatis.techtarget.com/definition/0,289893,sid9_gci550525,00.html

[11] R Anderson, "Security Engineering", Wiley, 2001, p314

[12] Cisco Systems, "Cisco SAFE Enterprise Layer Two Addendum", 2003,
URL:http://www.cisco.com/en/US/netsol/ns340/ns394/ns171/ns128/networking_solutions_white_paper09186a008014870f.shtml

[13] S Sipes, "Why your switched network isn't secure", Sept 2000
URL:http://www.sans.org/resources/idfaq/switched_network.php

[14] Pascal Meunier, Purdue University- Lecture Notes, Feb 2003
http://www.cs.purdue.edu/homes/cs490s/LectureNotes/cs490sARP.pdf

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

[15] completewhois.com, "Hijacked IP Q/A", Oct 2003, URL:http://www.completewhois.com/hijacked/hijacked_qa.htm

[16] JM Stewart, "Winnuke lives on, and it's coming to a system near you", techrepublic.com, Oct 2002 URL:http://techrepublic.com.com/5100-6313-1054537.html

[17] S Malik, "Network Security Principles and Practices", Cisco Press, 2003, pp 147-148

[18] "Hobbit", "CIFS- Common Insecurities Fail Scrutiny", Jan 1997 URL:http://downloads.securityfocus.com/library/cifs.txt

[19] "Voyager", "alt.2600/#hack FAQ - Q: What is a Format String Vulnerability?" www.hackfaq.org, 1994-2003, URL:http://www.hackfaq.org/computers-30.shtml

[20] "Scut", "Exploiting Format String Vulnerabilities", Sept 2001, URL:http://teso.scene.at/releases/formatstring-1.2.tar.gz

[21] "Hyperdictionary - Definition: Ephemeral Port", hyperdicitonary.com, 2000-2003, URL:http://www.hyperdictionary.com/computing/ephemeral+port

[22] B Aboba, "IPSEC-NAT Compatibility Requirements", IETF, June 2001, URL:http://www.ietf.org/proceedings/01dec/I-D/draft-ietf-ipsec-nat-reqts-00.txt

[23] Vicomsoft, "Firewall White Paper - What different types of firewalls are there?" 2003, URL:http://www.firewall-software.com/firewall_faqs/types_of_firewall.html

[24] CW Klaus, "Wireless LAN Security FAQ", ISS, Oct 2002, URL:http://www.iss.net/wireless/WLAN_FAQ.php

[25] S Crutchley, "Information Security: Addressing the human factor", SC Infosecurity News, June 2002 http://www.infosecnews.com/opinion/2002/06/19_03.htm

[26] "NSFOCUS Information Technology- ISO 7498-2 Security Model", NSFOCUS 2003 URL: http://www.nsfocus.com/english/homepage/solutions/system.htm

© SANS Institute 2004, As part of the Information Security Reading Room Author retains full rights.

Last Updated: June 19th, 2013

Upcoming SANS Training

Click Here for a full list of all Upcoming SANS Events by Location

Digital Forensics & Incident Response Summit 2013	**Austin, TXUS**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS London Summer 2013	**London, GB**	**Jul 09, 2013 - Jul 16, 2013**	**Live Event**
SANS Rocky Mountain 2013	**Denver, COUS**	**Jul 14, 2013 - Jul 20, 2013**	**Live Event**
SEC528 SANS Training Program for the CompTIA® New Advanced Security Practitioner (CASP) Certification	**Washington, DCUS**	**Jul 22, 2013 - Jul 26, 2013**	**Live Event**
SANS Mumbai 2013	**Mumbai, IN**	**Jul 22, 2013 - Jul 27, 2013**	**Live Event**
SANS San Francisco 2013	**San Francisco, CAUS**	**Jul 29, 2013 - Aug 03, 2013**	**Live Event**
SANS Boston 2013	**Boston, MAUS**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
SANS SEC 560: Network Penetration Testing @ Bangalore 2013	**Bangalore, IN**	**Aug 05, 2013 - Aug 10, 2013**	**Live Event**
Critical Security Controls Summit	**Washington, DCUS**	**Aug 12, 2013 - Aug 18, 2013**	**Live Event**
Industrial Control Systems Security Training - DC	**Washington, DCUS**	**Aug 12, 2013 - Aug 16, 2013**	**Live Event**
SANS Virginia Beach 2013	**Virginia Beach, VAUS**	**Aug 19, 2013 - Aug 30, 2013**	**Live Event**
SANS Thailand 2013	**Bangkok, TH**	**Aug 19, 2013 - Aug 31, 2013**	**Live Event**
SANS Melbourne 2013	**Melbourne, AU**	**Sep 02, 2013 - Sep 07, 2013**	**Live Event**
SANS Capital City 2013	**Washington, DCUS**	**Sep 03, 2013 - Sep 08, 2013**	**Live Event**
Network Security 2013	**Las Vegas, NVUS**	**Sep 14, 2013 - Sep 23, 2013**	**Live Event**
SANS Canberra 2013	**OnlineAU**	**Jul 01, 2013 - Jul 13, 2013**	**Live Event**
SANS OnDemand	**Books & MP3s OnlyUS**	**Anytime**	**Self Paced**