Intrusion Detection and Prevention Systems

* Intrusion detection systems (IDSs) and intrusion prevention systems (IPSs) are important tools in a computer security arsenal. Often thought of as a tertiary extra after antivirus software and firewalls, an IDS is often the best way to detect a security breach.

**IDS Concepts**

* *Intrusion detection (ID)* is the process of monitoring for and identifying specific malicious traffic.
* An IDS can take the form of a software program installed on an operating system, but today’s commercial network-sniffing IDS/IPS typically takes the form of a hardware appliance because of performance requirements.
* When the IDS notices a possible malicious threat, called an *event,* it logs the transaction and takes appropriate action. The action may simply be to continue to log, send an alert, redirect the attack, or prevent the maliciousness. If the threat is high risk, the IDS will alert the appropriate people.
* Alerts can be sent by e-mail, Simple Network Management Protocol (SNMP), pager, SMTP to a mobile device, or console broadcast.
* An IDS supports the defense-in-depth security principle and can be used to detect a wide range of rogue events, including but not limited to the following:

• Impersonation attempts

• Password cracking

• Protocol attacks

• Buffer overflows

• Software vulnerability exploits

• Malicious code, like viruses, worms, and Trojans

• Illegal data manipulation

• Unauthorized file access

• Denial of service (DoS) attacks

**Threat Types**

To really understand an IDS, you must understand the security threats and exploits it can detect and prevent. Threats can be classified as attacks or misuse, and they can exploit network protocols or work as malicious content at the application layer.

**Attacks or Misuse**

*Attacks* are unauthorized activity with malicious intent using specially crafted code or techniques. Attacks include denial of service, virus or worm infections, buffer overflows, malformed requests, file corruption, malformed network packets, or unauthorized program execution. *Misuse* refers to unauthorized events without specially crafted code. Regardless of how an alert is detected, the administrator groups all alerts into one of four categories:

• True positives (correct escalation of important events)

• False positives (incorrect escalation of unimportant events)

• True negatives (correct ignorance of unimportant events)

• False negatives (incorrect ignorance of important events)

An easy way to remember these principles is by thinking about the concepts of “alert” and “condition.” A simple fire alarm can serve as a good illustration:

• A true positive happens when the alert is positive (the alarm sounded) and the condition it represents is true (meaning there actually is a fire). That’s a good thing—it’s what the fire alarm is supposed to do.

• A false positive happens when the alert is positive (the alarm sounded) but the condition it represents is false (meaning there is no fire). That’s not so great—it wastes time and annoys people.

• A true negative is when the alert is negative (the alarm is not sounding) and it is reporting a true condition (there is no fire). That’s a good situation, and it’s what you’d expect the majority of the time.

• A false negative is when the alert is negative (no alarm sounded), but the condition it represents is false (there is fire). This is a truly dangerous condition, whether in the case of a building fire or in an IDS.

**Fragmentation and Reassembly Attacks** Although not quite the security threat they once were, IP packets can be used in *fragmentation* attacks. TCP/IP fragmentation is allowed because all routers have a *maximum transmission unit (MTU),* which is the maximum number of bytes that they can send in a single packet. A large packet can be broken down into multiple smaller packets (known as *fragments*) and sent from source to destination. A *fragment* *offset* value located in each fragment tells the destination IP host how to reassemble the separate packets back into the larger packet.

Attacks can use fragment offset values to cause the packets to maliciously reassemble and intentionally force the reassembly of a malicious packet. If an IDS or firewall allows fragmentation and does not reassemble the packets before inspection, an exploit may slip by.

Today, most IDSs, operating systems, and firewalls have antifragmentation defenses. By default, a Windows host will drop fragmented packets.

**Data Normalization**

An IDS signature database has to consider all character encoding schemes and tricks that can end up creating the same malicious pattern. This task is usually accomplished by normalizing the data before inspection. Normalization reassembles fragments into single whole packets, converts encoded characters into plain ASCII text, fixes syntax mistakes, removes extraneous characters, converts tabs to spaces, removes common hacker tricks, and does its best to convert the data into its final intended form.

**Threats an IDS Cannot Detect**

IDSs excel at catching known, definitive malicious attacks. Although some experts will say that a properly defined IDS can catch any security threat, events involving misuse prove the most difficult to detect and prevent. For example, if an outside hacker uses social engineering tricks to get the CEO’s password, not many IDSs will notice. If the webmaster accidentally posts a confidential document to a public directory available to the world, the IDS won’t notice.

**First-Generation IDS**

IDS development as we know it today began in the early 1980s, but only started growing in the PC marketplace in the late 1990s. First-generation IDSs focused almost exclusively on the benefit of early warning resulting from accurate detection. When an IDS misses a legitimate threat, it is called a *false negative*. Most IDS are plagued with even higher false positive rates, however.

A false positive is when the IDS says there is a security threat by “alerting,” but the traffic is not malicious or was never intended to be malicious (benign condition). A common example is when an IDS flags an e-mail as infected with a particular virus because it is looking for some key text known to be in the message body of the e-mail virus (for example, the phrase “cheap pharmaceuticals”). When an e-mail intended to warn readers about the virus includes the keywords that the reader should be on the lookout for, it can also create a false positive.

In an effort to decrease false positives, some IDSs are tuned to be more sensitive. They will wait for a highly definitive attack within a narrow set of parameters before they alert the administrator. Although they deliver fewer false positives, they have a higher risk of missing a legitimate attack.

**Second-Generation IDS**

First-generation IDSs focused on accurate attack detection. *Second-generation* IDSs dothat and work to simplify the administrator’s life by offering a bountiful array of back-endoptions. They offer intuitive end-user interfaces, intrusion prevention, centralized devicemanagement, event correlation, and data analysis. Second-generation IDSs do more thanjust detect attacks—they sort them, prevent them, and attempt to add as much value as theycan beyond mere detection.

**IDS Types and Detection Models**

Depending on what assets you want to protect, an IDS can protect a host or a network.

**Host-Based IDS**

A *host-based IDS* (HIDS) is isnstalled on the host it is intended to monitor. The host can be a server, workstation, or any networked device (such as a printer, router, or gateway). A HIDS installs as a service or daemon, or it modifies the underlying operating system’s kernel or application to gain first inspection authority.

A *file-integrity* HIDS (sometimes called a *snapshot* or *checksum* HIDS) takes a cryptographic

hash of important files in a known clean state and then checks them again later for comparison. If any changes are noted, the HIDS alerts the administrator that there may be a change in integrity.

A *behavior-monitoring* HIDS performs real-time monitoring and intercepts potentially malicious behavior. For instance, a Windows HIDS reports on attempts to modify the registry, manipulate files, access the system, change passwords, escalate privileges, and otherwise directly modify the host.

**Network-Based IDS (NIDS)**

*Network-based IDSs* (NIDSs) are the most popular IDSs, and they work by capturing and analyzing network packets speeding by on the wire. Unlike a HIDS, a NIDS is designed to protect more than one host. It can protect a group of computer hosts, like a server farm, or monitor an entire network. Captured traffic is compared against protocol specifications and normal traffic trends or the packet’s payload data is examined for malicious content.

If a security threat is noted, the event is logged and an alert is generated. With a HIDS, you install the software on the host you want monitored and the software does all the work. Because a NIDS works by examining network packet traffic, including traffic not intended for the NIDS host on the network, it has a few extra deployment considerations.

**Anomaly-Detection (AD) Model**

*Anomaly detection* (AD) was proposed in 1985 by noted security laureate Dr. Dorothy E. The goal of AD is to be able to detect a wide range of malicious intrusions. A simple example is someone logging in with the incorrect password too many times, causing an account to be locked out and generating a message to the security log. Anomaly detection IDS expands the same concept to cover network traffic patterns, application events, and system utilization.

Here are some other events AD systems can monitor and trigger alerts from:

• Unusual user account activity

• Excessive file and object access

• High CPU utilization

• Inappropriate protocol use

• Unusual workstation login location

• Unusual login frequency

• High number of concurrent logins

• High number of sessions

• Any code manipulation

• Unexpected privileged use or escalation attempts

• Unusual content

**AD Advantages**

AD systems are great at detecting a sudden high value for some metric. Eg : if your AD system defines a buffer overflow as any traffic with over a thousand repeating characters, it will catch any buffer overflow, known or unknown, that exceeds that definition.

**AD Disadvantages**

Because AD systems base their detection on deviation from what’s normal, they tend to work well in static environments, such as on servers that do the same thing day in and day out, or on networks where traffic patterns are consistent throughout the day.

**Signature-Detection Model**

*Signature-detection* or *misuse* IDSs are the most popular type of IDS, and they work by using databases of known bad behaviors and patterns. This is nearly the exact opposite of AD systems. When you think of a signature-detection IDS, think of it as an antivirus scanner for network traffic. Signature-detection engines can query any portion of a network packet or look for a specific series of data bytes. The defined patterns of code are called signatures and often they are included as part of a governing rule when used within IDS .

**Signature-Detection Rules**

Rules are the heart of any signature-detection engine. A rule usually contains the following information as a bare minimum:

• Unique signature byte sequence

• Protocol to examine (such as TCP, UDP, ICMP)

• IP port requested

• IP addresses to inspect (destination and source)

• Action to take if a threat is detected (such as allow, deny, alert, log, disconnect)

**Advantages of Signature Detection**

Signature-detection IDSs are proficient at recognizing known threats. Once a good signature is created, signature detection IDSs are great at finding patterns, and because they are popular, a signature to catch a new popular attack usually exists within hours of it first being reported.

Another advantage of a signature-detection IDS is that it will specifically identify the threat, whereas an AD engine can only point out a generality. An AD IDS might alert you that a new TCP port opened on your file server, but a signature-detection IDS will tell you what exploit was used. Because a signature-detection engine can better identify specific threats, it has a better chance at providing the correct countermeasure for intrusion prevention.

**Disadvantages of Signature Detection**

Although signature-detection IDS are the most popular type of IDS, they have several disadvantages as compared to an AD IDS.

**Cannot Recognize Unknown Attacks** Just like antivirus scanners, signature-detection IDSs are not able to recognize previously unknown attacks. Attackers can change one byte in the malware program (creating a variant) to invalidate an entire signature. Hundreds of new malware threats are created every year, and signature-based IDSs are always playing catch up.

**Performance Suffers as Signatures or Rules Grow** Because each network packet or event is compared against the signature database, or at least a subset of the signature database, performance suffers as rules increase. Most IDS administrators using signature detection usually end up only using the most common signatures and not the less common rules.

**IDS Features**

**IDS End-User Interfaces**

IDS end-user interfaces let you configure the product and see ongoing detection activities. You should be able to configure operational parameters, rules, alert events, actions, log files, and update mechanisms. IDS interfaces come in two flavors: syntactically difficult command prompts or less-functional GUIs.

Historically, IDSs are command-line beasts with user-configurable text files. Commandline consoles are available on the host computer or can be obtained by a Telnet session or proprietary administrative software.

**Intrusion-Prevention Systems (IPS)**

Since the beginning, IDS developers have wanted the IDS to do more than just monitor and report maliciousness nistrative software.

A few of the command-line IDS programs have spawned GUI consoles that hide the command-line complexities.

Going far beyond mere monitoring and alerting, second-generation IDSs are being called *intrusion-prevention systems* (IPSs)*.* They either stop the attack or interact with an external system to put down the threat.

**IPS Disadvantages**

A well-known consequence of IPSs is their ability to exacerbate the effects of a false positive.

With an IDS, a false positive leads to wasted log space and time, as the administrator researches the threat’s legitimacy. IPSs are proactive, and a false positive means a legitimate service or host is being denied.

**IDS Management**

Central to the IDS field are the definitions of *management console* and *agent*. An IDS agent (which can be a *probe, sensor,* or *tap*) is the software process or device that does the actual data collection and inspection.

IDS management consoles usually fulfill two central roles: configuration and reporting. If you have multiple agents, a central console can configure and update multiple distributed agents at once. For example, if you discover a new type of attack, you can use the central console to update the attack definitions for all sensors at the same time. A central console also aids in determining agent status—active and online or otherwise.

**IDS Logging and Alerting**

When security events are detected by an IDS, they generate alerts and log files.

**Alerts**

Alerts are high-priority events communicated to administrators in real time. The IDS’s policy determines what security threats are considered high risk, and the priority level is set accordingly. Typically, you would not want an IDS administrator to respond as quickly to a NetBIOS scan against your appropriately firewalled network as you would to a successful DoS attack against your company’s primary web server. When an event is considered high risk against a valuable asset, it should be communicated immediately.

**Logs**

IDS log files record all detected events regardless of priority and, after its detection engine, have the greatest influence on the speed and use of an IDS. IDS logs are used for data analysis and reporting. They can include just a barebones summary of events or a complete network packet decode. Although complete network traces are preferable for forensics, they can quickly take up a lot of hard drive space. A small network can generate hundreds of events a minute, and a mid-sized network can generate tens of thousands. If you plan to store multiple days’ worth of logs with full packet decoding, make sure your IDS’s hard drive is large enough.

**IDS Deployment Considerations**

IDSs are beneficial tools, but they have weaknesses. They need to be fine-tuned if you want to maximize their usefulness, and if you intend to deploy one, you’ll need to come up with a deployment plan to do so successfully.

This section summarizes these deployment issues.

**IDS Fine-Tuning**

Fine-tuning an IDS means doing three things: increasing inspection speed, decreasing false positives, and using efficient logging and alerting.

**Increasing Inspection Speed**

Most IDS administrators start off monitoring all packets and capturing full packet decodes. You can narrow down what packets an IDS inspects by telling it to include or ignore packets based on source and destination addresses. For example, if you are most concerned with protecting your servers, modify the IDS’s packet inspection engine so it only captures packets with server destination addresses. The more packets the IDS can safely ignore, the faster it will be.

**Decreasing False Positives**

Because IDS have so many false positives, the number one job of any IDS administrator is to track down and troubleshoot false positives. In most instances, false positives will outweigh all other events. Track them all down, rule out maliciousness, and then appropriately modify the source or IDS to prevent them.

**Using Efficient Logging and Alerting**

Most vendor products come with their own preset levels of event criticalities, but when setting up the IDS, take the time to customize the criticalities for your environment. For instance, if you don’t have any Apache web servers, set Apache exploit notices with a low level of prioritization. Better yet, don’t track or log them at all.

**IPS Deployment Plan**

So you want to deploy your first IPS. You’ve mapped your network, surveyed your needs, decided what to protect, and picked an IPS solution. Here are the steps to a successful

IPS deployment:

1. Document your environment’s security policy.

2. Define human roles.

3. Decide the physical location of the IPS and sensors.

4. Configure the IPS sensors and management console to support your security policy.

5. Plan and configure device management (including the update policy).

6. Review and customize your detection mechanisms.

7. Plan and configure any prevention mechanisms.

8. Plan and configure your logging, alerting, and reporting.

9. Deploy the sensors and console (do not encrypt communication between sensors and links to lessen troubleshooting).

10. Test the deployment using IPS testing tools (initially use very broad rules to makesure the sensors are working).

11. Encrypt communications between the sensors and console.

12. Test the IPS setup with actual rules.

13. Analyze the results and troubleshoot any deficiencies.

14. Fine-tune the sensors, console, logging, alerting, and reporting.

15. Implement the IPS system in the live environment in monitor-only mode.

16. Validate alerts generated from the IPS.

17. One at a time, set blocking rules for known reliable alerts that are important in your environment.

18. Continue adding blocking rules over time as your confidence in each rule increases.

19. Define continuing education plans for the IPS administrator.

20. Repeat these steps as necessary over the life of the IPS.

**Security Information and Event Management (SIEM)**

security systems can report to a centralized *Security Information and Event Management*

*(SIEM) system,* bringing together logs and alerts from several disparate sources.

SIEM platforms take the log files, find commonalities (such as attack types and threat origination), and summarize the results for a particular time period. For example, all logs and alerts from all IDSs, perimeter firewalls, personal firewalls, antivirus scanners, and operating systems can be tied together. Events from all logs are then gathered, analyzed, and reported on from one location.

SIEMs have a huge advantage over individual IDS systems because they have the capability to collect and analyze many different sources of information to determine what’s really happening.

A SIEM is one of the most important tools used by security operations and monitoring staff, because it provides one-stop visibility into many different areas of the information processing environment and attacks against those areas. Let’s take a look at what a SIEM can do.

**Data Aggregation**

SIEMs collect information from every available source that is relevant to a security event.

These sources take the form of alerts, real-time data, logs, and supporting data. Together, these provide the correlation engine of the SIEM with information it can use to make decisions about what to bring to the security administrator’s attention.

**Alerts**

When is an alert real, and when is it a false positive? This is the key question associated with an IDS, and a source of frustration for security administrators in charge of tuning IDSs. This is where a SIEM enters the picture. The SIEM’s key function is to validate security alerts using many different sources of data to reduce false positives, so only the most reliable alerts get sent on to the security administrator. Thus, the alerts from all IDS sources as well as all other security monitoring systems should be given only to the SIEM, so it can decide which ones to pass along.

**Real-Time Data**

Real-time data such as network flow data gives the SIEM additional information to correlate. Streaming this data into the SIEM provides important information about normal and abnormal traffic patterns that can be used in conjunction with alerts to determine whether an attack is in progress

**Logs**

Logs are different from events, in that they are a normal part of system activity and usually meant for debugging purposes. Logs can be an important additional data source for a SIEM, however. Logs contain valuable information about what’s happening on a system, and they can give the SIEM a deeper view into what’s happening.

Ideal log sources for any SIEM include the following:

• End-user computers

• Windows and Unix servers

• DNS and DHCP servers

• Mail servers

• Databases

• Web servers

• Applications

• Switches and routers

• Firewalls

• Web filters and proxies

• Antivirus

**Supporting Data**

You can enhance the quality of a SIEM’s correlation even more by providing the SIEM with supporting data that has been previously collected. Data can be imported into the SIEM, and it will use that data to make comparative determinations.

**Analysis**

A SIEM takes all the data given to it and makes decisions, so the security administrator can focus on the most important alerts. For this reason, event correlation is a SIEM’s most important feature. Real-time analysis of security events is only made possible with a SIEM. Thousands, or even millions, of events occur every second across most networks. No human can hope to see, absorb, and understand all of them at once.

**Operational Interface**

For all the data collected by the SIEM and its resulting alerts to be human-readable, it must present the information in a way that an administrator can understand at a glance. SIEMs do this with a dashboard. A dashboard is a graphical and organized representation of alerts, event data, and statistical information that allows the administrator to see patterns, understand trends, identify unusual activity, and perceive the current threat landscape quickly at any point in time.

New chapter : Voice over IP (VoIP) and PBX Security

**Background**

Today businesses of all sizes are compromised in a variety of ways through their voice systems. When you layer a VoIP system on top of an IP network, you combine the risks associated with both, creating a superset of new risks as of result. Here are two examples:

• Many VoIP systems are server-based and rely on common operating systems (mainly

Windows and Linux) to run their hardware interface. Therefore, they are susceptible to a class of problems that from a voice systems perspective were not previously a threat.

• IP-based voice protocols, while providing low-cost, advanced end-user features and reliable transport mechanisms for voice traffic, also give attackers a new method for exploiting voice systems and additional avenues for compromising data networks in general. the components of a modern enterprise IP-based phone or video system:

1. Call control elements (call agents)

• Appliance or server-based call control—Internet protocol private branch exchange (IPPBX)

• Soft switches

• Session border controllers (SBCs)

• Proxies

1. Gateways and gatekeepers

• Dial peers

1. Multi-conference units (MCUs) and specialized conference bridges
2. Hardware endpoints

• Phones

• Video codecs

• Other devices and specialized endpoints

1. Soft clients and software endpoints

• IP phones

• Unified messaging (UM) integrated chat and voice clients

• Desktop video clients.

• IP-based smartphone clients .

1. Contact center components

• Automated call distribution (ACD) and interactive voice response (IVR) systems

• Call center integrations and outbound dialers

• Call recording systems

• Call center workflow solutions

1. Voicemail systems.

**VoIP Components**

**Call Control**

The call control element (the “brains” of the operation) of a VoIP system can be either a purposed appliance, a piece of software that runs on a common or specialized server operating system, or a piece of network hardware embedded or integrated into another networking component such as a switch blade or software module (soft switch). Primarily responsible for call setup and teardown, signaling, device software serving, and feature configuration, call control is one of the easier pieces of the voice infrastructure to protect.

**Voice and Media Gateways and Gatekeepers**

The voice (or media) *gateway* is the pathway to the outside world. This component is what allows termination to a PSTN, transcoding between TDM and IP networks, media termination, and other types of analog/digital/IP interface required in today’s multimediarich

IP infrastructures. Gateways are configured to use *dial peers* (defined as “addressable endpoints”) to originate and receive calls.

**MCUs**

Conferencing and collaboration is used extensively within and across all enterprises as part of the fundamental communications capability that connects all users to each other. At the heart of this technology is the conference bridge, or multi-conference unit (MCU), a multiport bridging system for audio, video, and multimedia collaboration.

**Hardware Endpoints**

The hardware phone or video codec, sitting quietly idle in the office but running 24/7, may, however, become an important tool for advanced corporate espionage, eavesdropping, or denial of service attacks. Modern VoIP phones have a fair bit of intelligence built into them and offer a previously unavailable avenue—some phones have a built-in layer two switch and are capable of executing XML scripts or Java code locally.

**Software Endpoints**

Enterprise desktop strategy focuses on convergence and extending simple, useful technologies to end users. This focus is intended to increase overall productivity and collaboration. One component of this strategy is the soft phone or voice and video-enabled chat client. This is a piece of software that runs on a PC or mobile device and acts like a hardware endpoint by registering to the call control element(s) as a device.

Why would you install a soft client on a mobile device, which already has mobile capability?

Two reasons: Cost is, of course, the first one. In many places, data usage on a cell phone is less costly than calling minutes, and by running a soft client, you convert what would otherwise be cellular usage minutes into an IP data stream (thank the “unlimited data plan” for this being a viable option). Second, by running the soft client, you can extend your enterprise features to the mobile user, including functionality not typically available on mobile devices such as consolidated extension-based or URI dialing.

**Call and Contact Center Components**

Call centers have made a remarkable evolutionary leap, from initially being used as a place to take orders and field complaints, to being a strategic asset that most enterprises cannot survive without. Within the last decade, call centers have morphed into “contact centers” and “centers of excellence.” Trusted to sustain 24/7/forever operation and provide all levels of support to customers across every industry imaginable, these highly complex distributed systems, which now support millions of agents worldwide, have taken advantage of VoIP technologies in new and exciting ways—or, for the security administrator, in completely frightening ways. Their complexity has increased exponentially as the expectations of agents and customers alike have increased in sophistication.

The two core components of any call center are automatic call detection (ACD) and interactive voice response (IVR). Simply put, the ACD moves calls around, and the IVR collects information from the caller and queues those calls in the appropriate places, based on defined variables such as agent skills. Whereas some systems simply queue calls and route them when an agent is available, others have advanced speech recognition capability and complicated algorithms predicting variables such as wait time for the next agent. Because of the complexity of these systems, it is especially important to ensure that they are patched and updated on a regular basis. A compromise of ACD or IVR could spell disaster for the victim, up to and including unrecoverable brand damage.

**Voicemail Systems**

The last, but certainly not least, major component of a VoIP-based telephony system is the voicemail system. Auto attendants, direct inward system access (DISA) features used for manual call forwarding, automatic call forwarding, and other voicemail features are a “standard” component of enterprise life, which nearly everyone has come to expect and rely on. Unfortunately, they have historically been one of the easiest systems to abuse for three main reasons:

• Access to mailboxes is typically numeric-only, and people find long strings of numbers

difficult to remember. Easy (and often default) passwords are commonplace. War

dialers can be set up to target these systems and record successful logins for attackers to return to later. Anyone who has ever built a voicemail system knows the practice of initially setting everyone’s default password to their extension, or perhaps the last four digits of their direct inward dialing (DID) phone number, or some other easy-tofigure- out formula. This is a good opportunity to stretch your creative brain muscle and come up with something better.

• Since voicemail systems have never really been considered a “key” component of an enterprise infrastructure, much less attention has been paid to securing these systems than to, say, the enterprise ERP or financial systems. Keep in mind, access to this type of functionality in the wrong hands can cause permanent damage to an organization in financial (and worse) ways.

• More often than not system-level access to and from the outside world is not carefully controlled or audited, as some of a voicemail system’s convenience “features” need outside access in order to work properly.

**VoIP Vulnerabilities and Countermeasures**

• The original hacks—how to protect yourself from the oldest tricks in the book

• Adding insult to injury: consider who tries to exploit *voice* services vs. *VoIP* services?

• Vulnerabilities and exploits

• The network

* The servers
* The appliances

• The “other stuff”

• The protocols—examining specific areas of concern

• System integrators, hosted systems, and TEM as part of an enterprise security posture

**Steps**

1. Create a scorecard from the information you’ve gathered from your audit in order to identify your most significant risks and areas in need of attention; prioritize high risk items with a standard likelihood and severity graph or matrix.

2. Know your dial-in numbers; only publish them for those who may need to use them, and ensure the executive team is aware of the risk of offering this service.

3. Enforce password requirements for system access.

4. Delete old and unused mailboxes as soon as possible.

5. Use restrictions (like secondary authorization codes) to prevent DISA from being used for long distance and international calling; if not possible or if the feature is needed, ensure that all calls made via DISA are logged and auditable and users with access to the service are educated on the risks.

6. Limit exposure where possible by using fewer external dial-in numbers; enforce a business process that requires security team review and approval prior to enabling new services.

7. Do not offer all user features to all users by default, unless your security program can support the ongoing use, auditing, and management of these features for the full user population.

8. Pay attention to call forwarding and who is allowed to use the feature to send calls outside of your perimeter.

9. Determine how your TEM program can flag abnormal patterns or utilization in order to give you visibility into when you may have a problem.

**Vulnerabilities and Exploits**

For our purposes in this section, *vulnerability* means a weakness that has not yet been used to compromise a perimeter, whereas *exploit* is a compromised vulnerability.

**Network**

Security administrators need to understand how to strike a balance between functionality and security, particularly when their peers (network and systems administrators) have the job of trying to move traffic in an unobstructed fashion across common multiaccess networks as fast as possible.

Basic documentation you may want to have in place and keep updated on a regular change-driven or scheduled basis should include layer one, layer two, and layer three diagrams indicating the location of all voice system components in the network, and both physical and logical topology.

**Servers**

For any server-based system that runs on a commodity OS (typically Windows or Unix), ensure that your network or server teams are prepared to follow patch management procedures for these resources along with the rest of the environment. With companies like Microsoft enabling features like enterprise voice services and voicemail, system administrators have the added responsibility of ensuring that Windows servers are patched for these in addition to the rest of their KB patches.

**Appliances**

The real relevance for security administrators is in the amount of customization the provider does in order to offer their features. In one sense, a certain “security by obscurity” is achieved with highly customized platforms because there are generally fewer of them in the field and they present a less attractive target than something more widely deployed (which is additionally true for proprietary protocols). Inversely, an exploit specific to a unique platform may remain undiscovered for a longer period of time, as you are dependent on the manufacturer or specific product community to identify such vulnerabilities.

**Everything Else**

For the voicemail system:

• Use a least-privilege model in which administrators do not have mailboxes accessible via external means; require a VPN and strong authentication.

• Delete unused mailboxes.

• Force complexity requirements for voicemail passwords and access codes.

• Carefully consider the risks of allowing remote call forwarding or other call forwarding features, particularly those that can be enabled remotely; if a feature is not absolutely necessary for your users, do not allow it.

• Use strong authentication for “remote destination” calling or calling-card type features.

**The Protocols**

At the heart of the family of VoIP technologies are the specific protocols that enable the transit and real-time conversations that IP networks were not originally designed to handle. the following section lists the mechanics of the protocols you’ll encounter on an enterprise network, some associated risks, and practical suggestions for protecting them.

**Protocol: SIP**

**Governing Standard**

The Session Initiation Protocol (SIP)

**Purpose** Application layer control (signaling) protocol for creating, modifying, and terminating sessions with one or more participants. Sessions include Internet telephone calls, multimedia calls and distribution, and multimedia conferencing. In plain English: SIP is used for all kinds of voice and multimedia applications and is prolific both on corporate networks and the Internet, sometimes appearing unintentionally in enterprise environments via voice-enable chat clients that are both sponsored (e.g. Lync, Connect, Jabber, etc.) and unauthorized (Yahoo messenger, AIM, etc.).

**Function** SIP is a session-based protocol, using SIP invitations that are used to create sessions. These carry session descriptions that allow participants to negotiate a set of compatible media types (in the event that different endpoints or devices have different capabilities). SIP makes use of proxy servers to route requests to a user’s registered location (“current” location), authenticate and authorize services, implement provider call-routing policies, and provide features. SIP also provides a registration function that allows users to upload their current locations for use by SIP proxies. SIP runs on top of several different transport protocols and relies on a variety of different mechanisms for security.

**Known Compromises and Vulnerabilities** Because there are so many SIP-related vulnerabilities that exist based on the different implementations of the protocol and extensions, it is worth classifying them into the following categories:

• DoS, DDoS, flooding

• SPAM over Internet Telephony (SPIT)

• Vishing (the criminal practice of using social engineering over a telephony system, widely facilitated by VoIP and SIP-based systems)

• Spoofing, barging, and redirection

• Replay and interception

**Recommendations** If you’re going to allow SIP on the network or enable SIP-based enterprise applications, either for voice and video (or other converged services) or for less specific uses (third-party IM clients, etc.), seriously consider the minimum level users *need* in order to function. Discuss this with whoever in your organization is responsible for the services that use SIP and ensure that they understand the risks of this highly dynamic protocol.

**Protocol: SRTP**

**Governing Standard** RFC 3711 .

**Purpose** Secure Real-Time Transport Protocol (SRTP) is a profile of RTP, which can provide authentication, confidentiality, replay protection, and protection to the RTCP traffic.

**Function** SRTP provides a framework for authenticating and encrypting RTP and RTCP streams, including definition of a default set of transforms and extensibility for inclusions of future transform sets. SRTP offers high throughput and low packet expansion, both critical considerations for any protection mechanism of a real-time media capability.

**Known Compromises and Vulnerabilities** Although using SRTP is significantly better than not using anything, it is not by itself a catch-all or complete security mechanism for protecting voice or multimedia traffic. The default settings are susceptible to brute-force attacks, as in many implementations, SRTP only requires DES encryption, which is relatively easy to crack by modern computing standards. On top of this, key management is critical, as a compromised key negates the relevance of even strong encryption.

**Recommendations** Following security best practices ensures that the default encryption requirements that SRTP negotiates are suitably strong to prevent brute-force attacks, and a key management program helps guarantee that keys are changed frequently to preserve the integrity of the encryption in place.

**Protocol: IAX and IAX2**

**Purpose** Inter-Asterisk eXchange Protocol (IAX) was developed to minimize bandwidth utilization over slower network links, with support for trunking and multiplexing, and ability to traverse firewalls and NAT.

**Function** IAX is an “all-in-one” application layer control protocol for creating, modifying, and terminating multimedia sessions over IP networks from server-to-server and server-toclient.

Although primarily targeted at VoIP, IAX can be used for other multimedia applications including streaming video. IAX is somewhat unique in its “in-band” approach, delivering both control and media services together. IAX uses a single static-port UDP data stream that simplifies NAT traversal, a problem for some other voice control protocols. The intent is to simplify firewall and network management. IAX is also compact and efficient, and as an open protocol, supports future additional payload types and services, although to be incorporated, features have to be added to the protocol.

**Known Compromises and Vulnerabilities** As with all real-time systems, risks of resource exhaustion or DoS-type attacks are ever present. For IAX, because of the well-known single static port and risk of added processing time to the nonlatency-tolerant media streams, this risk should not be taken lightly. Additionally, some known vulnerabilities for the IAX2 libraries allow remote code execution via a truncated frame exploit. However, the most significant risk from IAX, in particular, is also one of the protocol’s main benefits—its efficiency and ability to support many different traffic streams in a multiplexed fashion over a firewall. While most IAX issues will be a result of the implementation versus the capability of the protocol, organizations with sensitive data or intellectual property that may be subject to corporate espionage or other commercial for-profit exploitation should carefully evaluate whether they want to support a protocol that makes it easier for someone to smuggle data outside the walls in an almost steganography-derived way.

**Protocol: ISDN**

**Purpose** Integrated Systems Digital Networks (ISDN) are the foundation of many of the modern TDM networks that support the PTN and PSTN.

**Function** As related to VoIP, ISDN networks are either used for IP-based transport or are linked via gateways to VoIP networks for PSTN access.

**Known Compromises and Vulnerabilities** ISDN has been around for some time and is a cornerstone of today’s global voice transport capabilities; consider how ISDN might play into your overall VoIP and multimedia systems. It is common to see exploits tried from IP-networks attempting to bridge the PSTN network; but it is also possible to compromise a gateway from the PSTN and create a hairpin, which is just as damaging to long-distance bills (and can be worse if you pay for inbound minutes as well).

**Recommendations** Audit all gateways on a regular basis that have both VoIP networks and PSTN networks connected to them. If using ISDN for videoconferencing, utilize the stronger authentication methods built in to the PPP protocol (CHAP), and preferably control who is allowed to dial in via ISDN. You can also use well-documented features like call back in order to prevent spoofing.

**Protocol: SMS**

**Purpose** Short Message Service (SMS) is a methodology for sending text messages via cellular or other mobile technologies, but is now being adopted and integrated into other multimedia applications.

**Function** Everyone today uses SMS with or without realizing it, but adoption in enterprise environments is increasing at an incredible rate for business communications.

**Known Compromises and Vulnerabilities** While SMS is not strictly related to enterprise

VoIP, understanding the trend toward owning and operating corporate SMS gateways is relevant. *Direct text marketing* and other methods of text SPAM/unsolicited/unregulated

SMS messaging will become a tool in the black hat’s toolkit in the near future (if it has not already come to pass). The same sophisticated social engineering tricks that can leverage

SIP so easily can also use SMS as another convenient launch medium.

**Recommendations** Specifics related to securing the operation of SMS are unique and need special consideration. The IP multimedia subsystem (IMS), part of the next-generation network (NGN) developed as a replacement for GSM by 3GPP, added support for SMS in release 11, and both this and other cellular network technologies (4G LTE for example) either support today or will support SMS. If interacting with or supporting cellular networks, ensure that the considerations for SMS make it into the overall risk assessment, and the specifics of the installation are defined, measured for risk, and evaluated on an ongoing basis as the services and uses evolve.

**Security Posture: System Integrators and Hosted VoIP**

How much does the system integrator or vendor that’s chosen really know about the selected VoIP or multimedia platform? Are they experts on security or on securing this specific system? How many times have they implemented a similar system, and have any of those systems been compromised? If deploying an off-premise solution, how will we guarantee the integrity of sensitive corporate conversations? What capabilities do we have to ensure that our phone bills are actually correct? These questions—and many more— need to be answered if your organization is in the process of evaluating or deploying a new

VoIP technology. The three specific areas alluded to in these questions can be outlined as detailed in the following sections.

For hosted VoIP:

• Should I consider a hosted option for enterprise use?

• Where does the responsibility lie for the security of a hosted system?

• Is it possible to integrate an off-premise solution with something internally hosted and managed, and is this a good idea?

For TEM:

• What is TEM and what does it do for the enterprise?

• How does TEM relate to security?

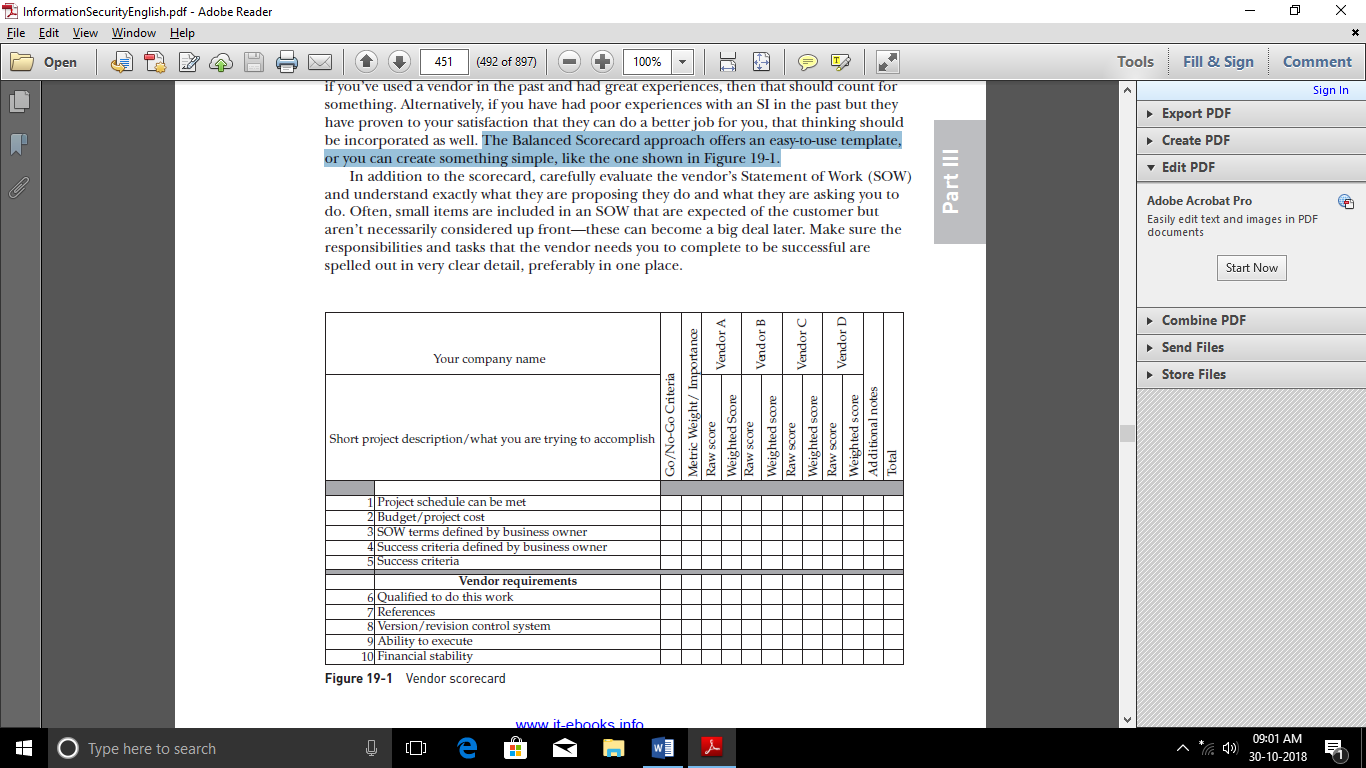
Before starting, ask yourself these questions:

• How can I choose a quality integrator and determine if the integrator has the necessary skills to implement the system?

• What questions can I ask in order to determine if one integrator is more security aware than another if they are both technically competent?

• Does the network require other attention prior to a VoIP deployment?

The Balanced Scorecard approach offers an easy-to-use template, or you can create something simple, like the one shown in Figure 19-1.



You can ask things like:

• How many deployments of the specific system have you completed?

• Are you familiar with this code revision and any security-related release notes and default setting changes for the version to be deployed?

• Have any of the systems you’ve previously installed been compromised?

• If so, why? Were you involved in the root cause analysis?

• What did you learn and what internal processes have you changed as a result of that experience?

• At what point do you change passwords during the install process?

• What are the basic ACLs or protections you put in place for every deployment, by default, without specific customer request?

• Which sets of security standard practices are you familiar with and which do you employ in your planning, installation, and deployment processes?

• A question for yourself: Since you’re going to *rely* on this SI to perform work that you will have to put your seal of approval on and possibly attach your name to … do you have a sense of confidence that the vendor will to “do the right thing” or do what you would do given a difficult choice?

**PBX**

A Private Branch Exchange (PBX) is a computer-based switch that can be thought of as a local phone company. Following are some common PBX features:

• Multiple extensions

• Voicemail

• Call forwarding

• Fax management

• Remote control (for support)

**Hacking a PBX**

Attackers hack PBXs for several reasons:

• To gain confidential information (espionage)

• To place outgoing calls that are charged to the organization’s account (and thus free to the attacker)

• To cause damages by crashing the PBX

This section briefly reviews some common attacks, without delving into details.

**Administrative Ports and Remote Access**

Administrative ports are needed to control and diagnose the PBX. In addition, vendors often require remote access via a modem to be able to support and upgrade the PBX.

This port is the number one hacker entry point. An attacker can connect to the PBX via the modem; or if the administrative port is shared with a voice port, the attacker can access the port from outside the PBX by calling and manipulating the PBX to reach the administrative port. Just as with administrative privileges for computers, when attackers have remote administrative privileges, “they own the box” and can use it to make international calls or shut down the PBX.

**Voicemail**

An attacker can gain information from voicemail or even make long-distance phone calls using a “through-dial” service. (After a user has been authenticated by the PBX, that user is allowed to make calls to numbers outside the PBX.) An attacker can discover a voicemail password by running an automated process that “guesses” easy passwords such as “1111,”

“1234,” and so on.

**Denial of Service**

A PBX can be brought down in a few ways:

• PBXs store their voicemail data on a hard drive. An attacker can leave a long message, full of random noises, in order to make compression less effective—whereby a PBX might have to store more data than it anticipated. This can result in a crash.

• An attacker can embed codes inside a message. (For example, an attacker might embed the code for message rewinding. Then, while the user listens to the message, the PBX will decode the embedded command and rewind the message in an endless loop.)

**Securing a PBX**

Here is a checklist for securing a PBX:

• Connect administrative ports only when necessary.

• Protect remote access with a third-party device or a dial-back.

• Review the password strength of your users’ passwords.

• Allow passwords to be different lengths, and require the # symbol to indicate the end of a password, rather than revealing the length of the password.

• Disable all through-dialing features.

• If you require dial through, limit it to a set of predefined needed numbers.

• Block all international calls, or limit the number of users who can initiate them.

• Block international calls to places such as the Caribbean that fraudsters tend to call.

• Train your help desk staff to identify attempted PBX hacks, such as excessive hangups, wrong number calls, and locked-out mailboxes.

• Make sure your PBX model is immune to common DoS attacks.

**TEM: Telecom Expense Management**

Phone bills can be more complex to read than ancient hieroglyphs, and there has been little progress made on simplifying or decoding them for the average consumer or telecom manager. Understanding what is on your phone bill so you can tell whether your voice providers are doing the right thing is important (there are alarming statistics on the error percentage in consumer and corporate phone bills). But that’s the job of your telecom group—why would a security professional care about phone bills? Your phone bill can have some clues to other problems in your environment, and a TEM program can help automate the process of getting to the goodies, the high-quality information you need to tell quickly if you have a security problem related to your phone system. There are many firms armed with specialized software ready to help you collect, organize, understand, interpret, and audit your telephone bills, all for a modest gain-share or percentage of savings fee (an interesting side note and case in point, that’s how bad telecom bills are—companies will *guarantee* that they will save you *so* much money that they will derive their compensation purely from a percentage of the money they save you or get back for. And TEM firms are doing quite well, illustrating the level of opportunity out there). While effort (hopefully someone else’s) is involved in the setup and optimization of the billing, once you’ve reached the point where a TEM firm can actually audit bills, you’re likely to have a useful tool to spot irregular or suspicious activity that may otherwise be tough to catch. Although phone bills are generally not directly related to the security group’s main role, it is the objective of every security group to protect stakeholder interests, and TEM can help a security group detect anomalous behavior and operate more quickly and effectively when they are called in to action for this type of an issue.

Next Chapter : Operating System Security Models

**Operating System Models**

The *operating system security model* (also known as the *trusted computing base,* or TCB) is simply the set of rules, or protocols, for security functionality. Security commences at the network protocol level and maps all the way up to the operations of the operating system. the trend in operating systems has been toward a microkernel architecture. In contrast to the monolithic kernel, microkernels are platform independent. Although they lack the performance of monolithic systems, they are catching up in terms of speed and optimization.

A microkernel approach is built around a small kernel with a common hardware level. The key advantage of a microkernel is that the kernel is small and easy to port to other systems.

**The Underlying Protocols Are Insecure**

The protocol’s main problems are as follows:

**• Vulnerable to spoofing** Spoofing is the term for establishing a connection with a forged sender address. Normally this involves exploiting trust relations between the source address and the destination address. The ability to spoof the source IP address assists those carrying out DoS attacks by making it difficult for victims to block the DoS traffic, and the predictability of the initial sequence number (ISN), which is a unique number that is supposed to guarantee the authenticity of the sender, contributes more to spoofing attacks by allowing an attacker to impersonate legitimate systems and take over a connection (as in a man in the middle attack).

**• Vulnerable to session hijacking** An attacker can take control of a connection by intercepting the session key and using it to insert his own traffic into an established TCP/IP communication session, usually in combination with a DoS attack against the legitimate sender so that traffic cannot get through, as in a man in the middle attack.

**• Predictable sequence guessing** The sequence number used in TCP connections is a 32-bit number, so the odds of guessing the correct ISN would seem to be exceedingly low. If the ISN for a connection is assigned in a predictable way, however, it becomes relatively easy to guess. The truth is that the ISN problem is not a protocol problem but rather an implementation problem. The protocol actually specifies pseudorandom sequence numbers, but many implementations have ignored this recommendation.

**• No authentication or encryption** The lack of authentication and encryption with TCP/IP is a major weakness.

**• Vulnerable to SYN flooding** SYN flooding takes advantage of the three-way handshake in establishing a connection. When Host B receives a SYN request from A, it must keep track of the partially opened connection in a *listen queue,* enabling successful connections even with long network delays. The problem is that many implementations can keep track of only a limited number of connections. A malicious host can exploit the small size of the listen queue by sending multiple SYN requests to a host but never replying to the SYN and ACK the other host sends back. By doing so, the malicious host quickly fills up the other host’s listen queue, and that host stops accepting new connections until a partially opened connection in the queue is completed or times out.

**Access Control Lists**

Much of the security functionality afforded by an operating system is via the ACL. Access control comes in many forms, but in whatever form it is implemented, it is the foundation of any security functionality.

An access control list is defined as a table that tells a computer operating system which access rights each user has to a particular system object, such as a file directory or an individual file. Each object has a security attribute that identifies its access control list. The list has an entry for each system user with access privileges. An object’s security descriptor can contain two ACLs:

• A *discretionary* access control list (DACL) that identifies the users and groups who are allowed or denied access

• A *system* access control list (SACL) that controls how access is audited

**Classic Security Models**

**Bell-LaPadula**

While the Bell-LaPadula model was revolutionary when it was published in 1976 . The Bell-LaPadula model was one of the first attempts to formalize an information security model. The Bell-LaPadula model was designed to prevent users and processes from reading above their security level. this model prevents objects and processes with any given classification from writing data associated with a lower classification.

**Biba**

Biba is often known as a reversed version of Bell-LaPadula, as it focuses on integrity labels, rather than sensitivity and data classification. (Bell-LaPadula was designed to keep secrets, not to protect data integrity.)

Biba covers integrity levels, which are analogous to sensitivity levels in Bell-LaPadula, and the integrity levels cover inappropriate modification of data. Biba attempts to preserve the first goal of integrity, namely to prevent unauthorized users from modifying data.

**Clark-Wilson**

Clark-Wilson attempts to define a security model based on accepted business practices for

transaction processing. Much more real-world-oriented than the other models described, it

articulates the concept of *well-formed transactions* that

• Perform steps in order

• Perform exactly the steps listed

• Authenticate the individuals who perform the steps

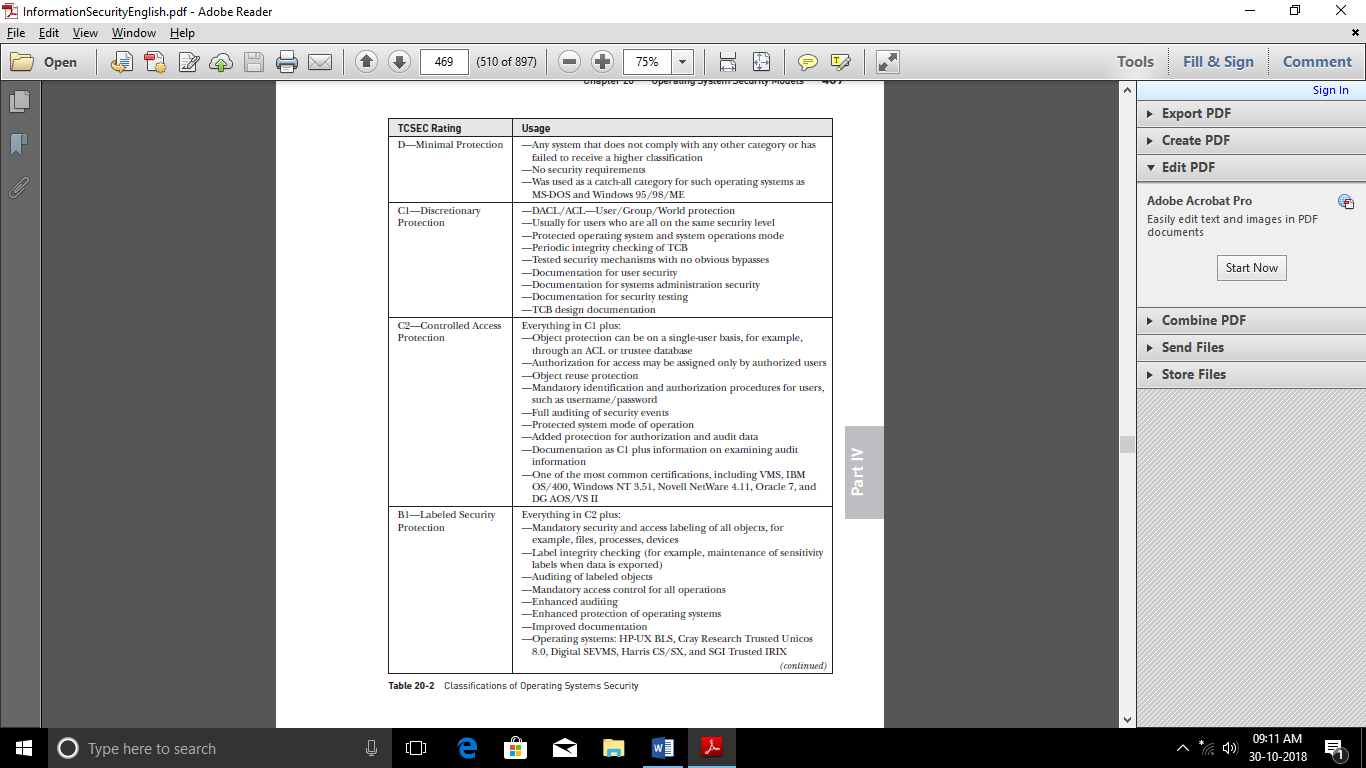
**TCSEC**

TCSEC was developed to meet three objectives:

• To give users a yardstick for assessing how much they can trust computer systems for the secure processing of classified or other sensitive information

• To guide manufacturers in what to build into their new, widely available commercial products to satisfy trust requirements for sensitive applications

• To provide a basis for specifying security requirements for software and hardware acquisitions



**Reference Monitor**

**The Reference Monitor Concept**

The National Institute of Standards and Technologies describes the reference monitor concept as an object that maintains the access control policy. It does not actually change the access control information; it only provides information about the policy.

The main elements of an effective reference monitor are that it is

**• Always on** Security must be implemented consistently and at all times for the entire system and for every file and object.

**• Not subject to preemption** Nothing should be able to preempt the reference monitor. If this were not the case, then it would be possible for an entity to bypasthe mechanism and violate the policy that must be enforced.

**• Tamperproof** It must be impossible for an attacker to attack the access mediation mechanism such that the required access checks are not performed and authorizations not enforced.

**• Lightweight** It must be small enough to be subject to analysis and tests, proving its effectiveness.

**Windows Security Reference Monitor**

The Windows Security Reference Monitor (SRM) is responsible for validating Windows process access permissions against the security descriptor for a given object. The Object

Manager then, in turn, uses the services of the SRM while validating the process’s request to access any object.

Windows is clearly not a bulletproof operating system, as is evident from the number of security advisories alone. In fact, it is full of security holes. But the fact that it is the most popular operating system in use in corporate settings and that Microsoft has been, for the most part, open with its security functionality, makes it a good case study for a real-world example of how an operating system security model should operate.

**Trustworthy Computing**

The four goals of the Trustworthy Computing initiative are

**• Security** As a customer, you can expect to withstand attack. In addition, you can expect the data is protected to prevent availability problems and corruption.

**• Privacy** You have the ability to control information about yourself and maintain privacy of data sent across the network.

**• Reliability** When you need your system or data, they are available.

**• Business integrity** The vendor of a product acts in a timely and responsible manner, releasing security updates when a vulnerability is found. *Secure by design* simply means that all vulnerabilities are resolved prior to shipping the product. Secure by design requires three steps.

1. *Build a secure architecture.* This is imperative. Software needs to be designed with security in mind first and then features.

2. *Add security features.* Feature sets need to be added to deal with new security vulnerabilities.

3. *Reduce the number of vulnerabilities in new and existing code.* The internal process at

Microsoft was revamped to make developers more conscious of security issues while designing and developing software.

**International Standards for Operating System Security**

**Common Criteria**

Common Criteria are meant to be, a global security standard ensuring that there is a common mechanism for evaluating the security of technology products and systems. By providing a common set of requirements for comparing the security functions of software and hardware products, the Common Criteria enable users to have an objective yardstick by which to evaluate a product’s security.

**Common Criteria Sections**

Common Criteria is a set of three distinct but related parts. These are the three parts of the Common Criteria:

• Part 1 is the introduction to the Common Criteria. It defines the general concepts and principles of information technology security evaluation and presents a general model of evaluation. Part 1 also presents the constructs for expressing information technology security objectives, selecting and defining information technology security requirements, and writing high-level specifications for products and systems. In addition, the usefulness of each part of the Common Criteria is described in terms of each of the target audiences.

• Part 2 details the specific security functional requirements and details a criterion for expressing the security functional requirements for Targets of Evaluation (TOE).

• Part 3 details the security assurance requirements and defines a set of assurance components as a standard way of expressing the assurance requirements for TOE.

Part 3 lists the set of assurance components, families, and classes and defines evaluation criteria for protection profiles (PPs). A protection profile is a set of security requirements for a category of TOE and security targets (STs). Security targets are the set of security requirements and specifications to be used as the basis for evaluating an identified TOE. Part 3 also presents evaluation assurance levels that define the predefined Common Criteria scale for rating assurance for TOE, namely the evaluation assurance levels (EALs).

**Protection Profiles and Security Targets**

Protection profiles (PPs) and security targets (STs) are two building blocks of the Common

Criteria.

A *protection profile* defines a standard set of security requirements for a specific type of product (for example, operating systems, databases, or firewalls). These profiles form the basis for the Common Criteria evaluation. By listing required security features for product families, the Common Criteria allow products to state conformity to a relevant protection profile. During Common Criteria evaluation, the product is tested against a specific PP, providing reliable verification of the product’s security capabilities.

The overall purpose of Common Criteria product certification is to provide end users with a significant level of trust. Before a product can be submitted for certification, the vendor must first specify a security target. The *security target* description includes an overview of the product, potential security threats, detailed information on the implementation of all security features included in the product, and any claims of conformity against a PP at a specified EAL.

**Problems with the Common Criteria**

Although there are benefits to the Common Criteria, there are also problems with this approach. The point of this section is not to detail those problems, but in a nutshell, to give you a brief summary of the issues:

**• Administrative overhead** The overhead involved with gaining certification takes a huge amount of time and resources.

**• Expense** Gaining certification is extremely expensive.

**• Labor-intensive certification** The certification process takes months.

**• Need for skilled and experienced analysts** Availability of information security professionals with the required experience is still lacking.

**• Room for various interpretations** The Common Criteria leave room for various interpretations of what the standard is attempting to achieve.

**• Paucity of Common Criteria testing laboratories** There are only seven laboratories in the United States.

**• Length of time to become a Common Criteria testing laboratory** Even for those organizations that are interested in becoming certified, the process in and of itself takes quite a while.S