


“T” is a software developer, manager, project manager, or SEPG member.

The CERT Coordination Center (CERT/CC) is located at the Software Engineering Institute (SEI), a federally funded research and development center at Carnegie Mellon University in Pittsburgh, Pennsylvania. Following the Morris worm incident, which brought 10 percent of Internet systems to a halt in November 1988, the Defense Advanced Research Projects Agency (DARPA) charged the SEI with setting up a center to coordinate communication among experts during security emergencies and to help prevent future incidents. Since then, the CERT/CC has helped to establish other response teams, and our incident handling practices have been adapted by more than 200 response teams around the world.

While we continue to respond to security incidents and analyze product vulnerabilities, our role has expanded over the years. Each year, commerce, government, and individuals grow increasingly dependent on networked systems. Along with the rapid increase in the size of the Internet and its use for critical functions, there have been progressive changes in intruder techniques, increased amounts of damage, increased difficulty of detecting an attack, and increased difficulty of catching the attackers. To better manage these changes, the CERT/CC is now part of the larger SEI Networked Systems Survivability Program, whose primary goals are to ensure that appropriate technology and systems management practices are used to resist attacks on networked systems and to limit damage and ensure continuity of critical services in spite of successful attacks ("survivability").

To accomplish our goals, we focus our efforts on the following areas of work: survivable enterprise management, survivable network technology, incident handling, incident and vulnerability analysis, and courses and seminars.

We are also committed to increasing awareness of security issues and helping organizations improve the security of their systems. Therefore, we disseminate information through many channels.



Networked Systems Survivability

Survivability

Focuses on sustaining the mission in the face of an ongoing attack; requires an enterprise-wide perspective

Depends on the ability of networks to provide continuity of service, albeit degraded, in the presence of attacks, failures, or accidents

Requires that only the critical assets need the highest level of protection

Complements current risk management approaches that are part of an organization's business practices

Includes (but is broader than) traditional information security

[Lipson 99]

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 2


Lipson, Howard, Fisher, David. "Survivability - A New Technical and Business Perspective on Security." Proceedings of the 1999 New Security Paradigms Workshop. Association for Computing Machinery, 1999. Available at <http://www.cert.org/archive/pdf/busperspec.pdf>.

Ultimately it is the mission that must survive, not any particular component of the system or even the system itself.

Survivability is intended to permeate all levels of an organization

- from the C*O level, where corporate direction is set
- through middle management, where requests for capital are OK'ed and the time needed to allow other levels to achieve survivability are accepted
- to programmers, where programming for survivability is practiced
- and systems administrators, where recommending and installing technology in support of survivability achieves the corporate goals

Networked Systems Survivability



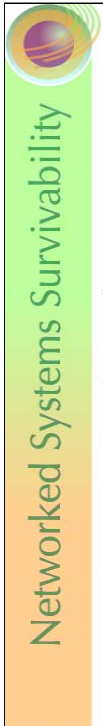
Agenda

- Motivation
- Perspectives/Questions
- Protecting Critical Assets
- Role of SEPG?

© 2003 by Carnegie Mellon University

SEPG 03 Presentation - slide 3

SEPGs have experience and skill to put enterprise-wide improvements in place. This presentation describes a problem domain that desperately needs improvement and SEPGs have the skill set to make it happen.



The Problem

“We wouldn’t have to spend so much time, money, and effort on network security if we didn’t have such **bad software security**.” [Viega, McGraw 02]

“It is **bad software** that results in [security] vulnerabilities in the first place.” [Viega, McGraw 02]

“We found **security design flaws** in 70 percent of the defects we analyzed. Nearly half (47 percent) could have been caught – and fixed inexpensively – during the design stage.” [Jaquith 2002]

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 4

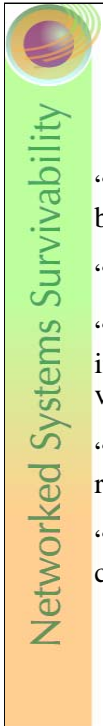
Connects to quality goals that SEPGs have first and foremost along with reputation with respect to maturity level, for example.

John Viega, Gary McGraw. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley, 2002.

Based on analysis of 45 e-business applications, @stake found **security design flaws** in 70 percent of the defects they analyzed. After they excluded flaws that were of low business impact or were not easily exploitable, nearly half (47 percent) of the remaining serious defects could have been caught – and fixed inexpensively – during the design stage. These serious defects are readily exploitable and could cause significant loss of reputation or customer revenue. Andrew Jaquith. “The Security of Applications: Not All Are Created Equal.” @stake, February 2002. Available at http://www.atstake.com/research/reports/acrobat/atstake_app_unequal.pdf.

“There is little evidence of movement toward improvement in the security of most products. We continue to see the same types of vulnerabilities in newer versions of products that we saw in earlier versions. Technology evolves so rapidly that vendors concentrate on time to market, often minimizing that time by placing a **low priority on the security of their products**.” Rich Pethia, CERT Program Director. Congressional Testimony, available at http://www.cert.org/congressional_testimony/Pethia_testimony_Sep26.html

Charles C. Mann. “Why Software Is So Bad.” *Technology Review*, July/August 2002.



Who Is Saying This?

“Security models should be easy for developers to understand and build into their applications.”

“Our products should emphasize security right out of the box.”

“As software has become ever more complex, interdependent and interconnected, our reputation as a company has in turn become more vulnerable.”

“So now, when we face a choice between adding features and resolving security issues, we need to choose security.”

“Eventually, our software should be so fundamentally secure that customers never even worry about it.”

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 5

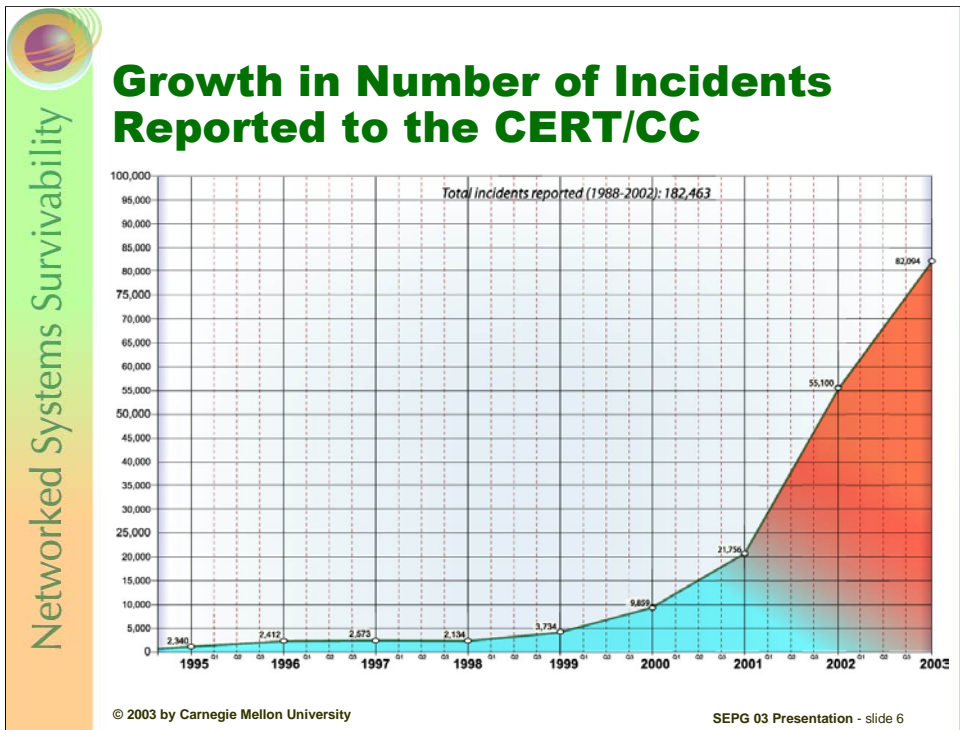
Bill Gates memo to Microsoft employees, January 15, 2002.

Business Week Online March 18, 2002 Commentary: The Best Way to Make Software Secure: Liability

“To date, there has been little incentive for Microsoft and other off-the-shelf software makers to do more. Why? Because they have insulated themselves by disclaiming all product liability. The courts have decided that buyers waive their right to sue after clicking the "I accept" button when they install software. "If Firestone produces tires with systemic vulnerabilities, they are liable," says Bruce Schneier, chief technology officer of Counterpane Internet Security Inc., a provider of network protection services. "If Microsoft produces software with systemic vulnerabilities, they're not liable."

“A better model for improving security may be the Y2K bug. Facing the threat of widespread computer meltdowns at the millennium, industry mobilized to change business practices and governments passed laws requiring Y2K certification for tech gear. Companies underwent massive campaigns to make certain they complied because they didn't want to be held liable for damages. The Securities & Exchange Commission required corporations to provide details of their Y2K efforts in quarterly earnings reports.”

03/10/2002 - Updated 11:19 PM ET Air Force seeks better security from Microsoft
 SEATTLE — A top U.S. Air Force official has warned Microsoft to dramatically improve the security of its software or risk losing the Air Force as a customer. In an interview, Air Force chief information officer John Gilligan revealed he has met with senior Microsoft executives to tell them the Air Force is "raising the bar on our level of expectation" for secure software.
<http://www.usatoday.com/life/cyber/tech/2002/03/11/gilligan.htm>



Incident: Any real or suspected adverse event in relation to the security of computer systems or networks; the act of violating an explicit or implied security policy.

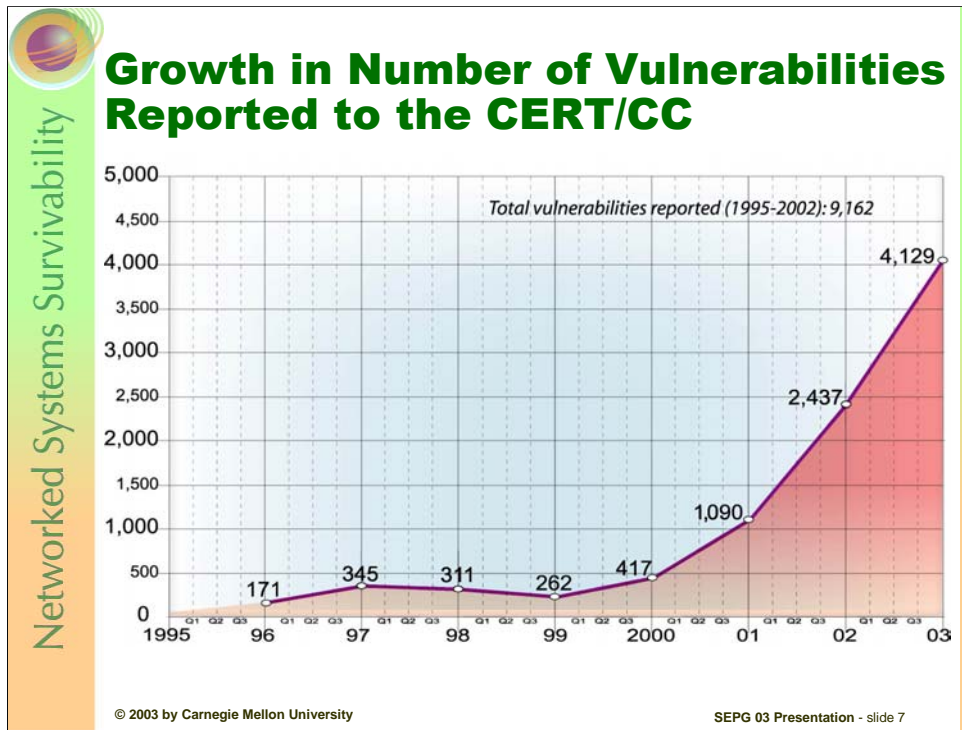
Examples include:

- failed or successful attempts to gain unauthorized access to a system or its data
- unwanted disruption or denial of service
- the unauthorized use of a system for the processing or storage of data
- changes to systems without the owner's consent
- the occurrence of computer viruses
- probes (single attempt) or scans (multiple attempts) for vulnerabilities via the network to a range of computer systems

The number of incidents reported to CERT/CC went up 164% in 1999 [3,734 to 9,859], 121% in 2000 [9,859 to 21,756], 153% in 2001 [21,756 to 55,100], and 49% in 2002 [55,100 to 82,094].

Why has this happened?

- more computers
- more at stake
- more people reporting
- CERT better known
- more incidents



Vulnerability: a set of conditions in a software system that allows an intruder to violate an implicit or explicit security policy.

Examples include:

- phf (remote command execution as user "nobody")
- rpc.ttdbserverd (remote command execution as root)
- world-writable password file (modification of system-critical data)
- default password (remote command execution or other access)
- denial of service problems that allow an attacker to cause a Blue Screen of Death
- smurf (denial of service by flooding a network)

The number of vulnerabilities reported to CERT/CC went up 161% in 2000 [417 to 1,090], 124% in 2001 [1,090 to 2,437], and 69% in 2002 [2,437 to 4,129].

See also System Administration, Networking and Security Institute. "SANS/FBI Top 20 List: The Twenty Most Critical Internet Security Vulnerabilities (Updated), The Experts' Consensus, Version 3.21." October 17, 2002. Available at <http://www.sans.org/top20.htm>.

Networked Systems Survivability

Attack Trends

- Increased automation, speed of attack tools
- Increased attack tool sophistication
- Faster discovery of vulnerabilities
- Increasing permeability of firewalls
- Increasing asymmetric threat
- Increasing threat from infrastructure attacks

© 2003 by Carnegie Mellon University
SEPG 03 Presentation - slide 8

Reference CERT paper "Overview of Attack Trends,"
http://www.isalliance.org/resources/papers/attack_trends.pdf

Slammer/Sapphire worm that hit Jan 25, 2003:

In just a few hours, the "Sapphire" worm, consisting of a minute bit of software code, shut down some Bank of America Corp. ATMs, fouled Continental Airlines' online ticketing system and essentially blacked out an emergency call center in Seattle, where computers slowed to a crawl. At the same time, it cut off access to the Internet for millions of personal computer users, including most of those in South Korea.

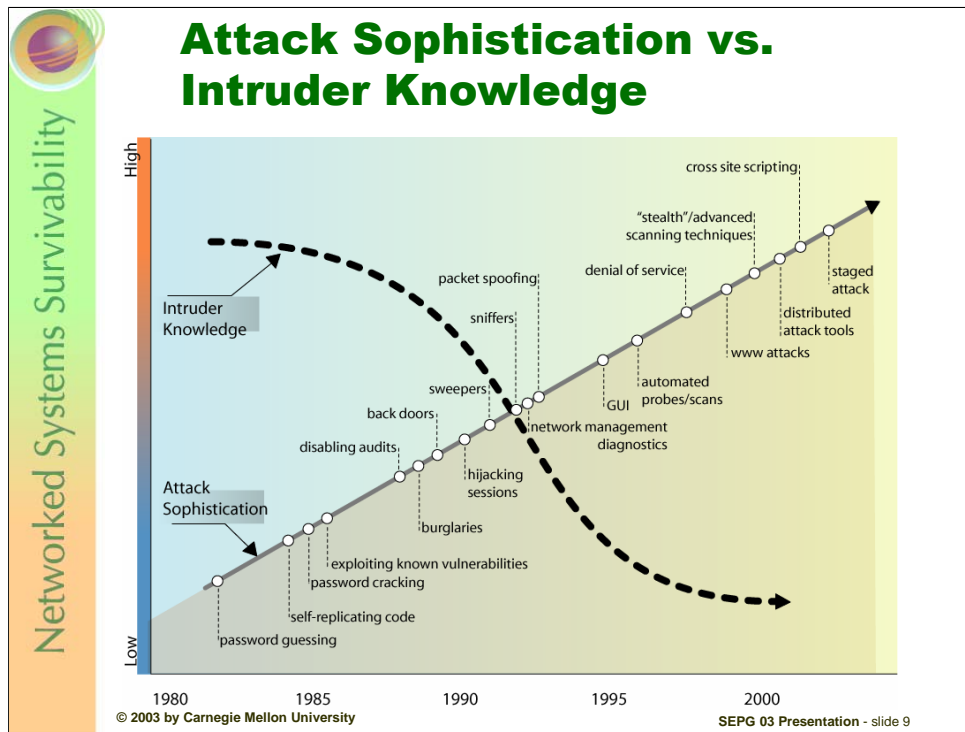
The worm, also known as "Slammer," spread quickly after it was introduced onto the Internet. Using a well-known flaw in a Microsoft Corp. database program, the worm overwhelmed computers with data. Many other systems quickly suffered ancillary effects as packets of information seeking ways around the vulnerable machines backed up in the ensuing congestion. Specialists described the impact as a sort of global traffic jam, like the ones that occur on Washington area highways when main arteries are shut down.

<http://www.washingtonpost.com/ac2/wp-dyn/A57550-2003Jan28>

Slammer is Fastest Spreading Worm (3 February 2003)

The Slammer worm infected 90% of vulnerable computers within ten minutes, according to the Cooperative Association for Internet Data Analysis (CAIDA). The number of infections doubled in size every 8.5 seconds; after three minutes, Slammer was generating 55 million scans for vulnerable computers every second.


<http://news.zdnet.co.uk/story/0,,t269-s2129785,00.html>



Trends indicate that the sophistication of the attack tools is increasing while the amount of technical knowledge required to launch an attack is decreasing. Sum of the curves is the key. Either it's in the brain or in the tool. If it's in the tool, then more can take advantage and it also gives way to script kiddies. It takes less than 2 minutes for an unsophisticated attacker to download an automated tool from the Internet and attack a site.

[More detail: In the 1980s intruders were the system experts. They had a high level of expertise and personally constructed the methods for breaking into systems. Using automated tools and exploit scripts was the exception rather than the rule. Today, absolutely anyone can attack a network. This is due to the widespread and easy availability of intrusion tools and exploit scripts that can easily duplicate known methods of attack. While experienced intruders are getting smarter, as demonstrated by the increased sophistication in the types of attacks, the knowledge required on the part of novice intruders to copy and launch known methods of attack is decreasing. Meanwhile, as evidenced by distributed denial-of-service attacks and variants of the Love Letter Worm, the severity and scope of attack methods is increasing.

In the early/mid 1980s, intruders manually entering commands on their personal computer could access tens to hundreds of systems; today, intruders use automated tools to access thousands to tens of thousands of systems. In 1980s, it was relatively straightforward to determine if an intruder had penetrated your systems and discover what they did. Today, intruders are able to totally hide their presence by, for example, disabling commonly used services and reinstalling their own versions, and erasing their tracks in audit and log files. In the 1980s and early 1990s, denial-of-service attacks were infrequent and not considered serious. Today, for organizations such as Internet service providers that conduct business electronically, a successful denial-of-service attack can put them out of business. Unfortunately, these types of attacks occur more frequently each year.]



Networked Systems Survivability

Attack Impacts

- Loss/compromise of sensitive data
- System downtime; lost productivity
- System damage
- Financial loss
- Loss of reputation, customer confidence
- Other organizations' systems affected

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 10

CERT/CC Overview: Incident and Vulnerability Trends. Available at <http://www.cert.org/present/cert-overview-trends/>.

One of the ways to think about this is **quantifying/qualifying impacts in SPI-meaningful terms:**

- staff work hours are increased by 40% for two days as a result of losses and re-work due to a security incident
- operating costs for the quarter exceed budget by 20% due to eradicating a virus and repairing the damage it caused
- lost network availability [essential to working in a distributed development environment and with partners/collaborators] resulting in lost productivity, cost overruns, schedule slips
- rework, rework, rework
- overtime; sleep deprivation
- development systems unavailable; can't do my job
- late deliverables; unhappy customers
- loss of project/customer; layoffs
- loss of market position



Agenda

Motivation

Perspectives/Questions

Protecting Critical Assets

Role of SEPG?





SPI Perspective

Dealing primarily with software developers and their management chain

End objective is to produce quality systems and products, on schedule and on budget

Security typically addressed

- during the software development life cycle
- during the O&M phase as an add-on/after the fact consideration
- for COTS software, as a provider responsibility

O&M: Operations and Maintenance

COTS: Commercial Off-The-Shelf



Security Improvement Perspective

Typically dealing with an organization's infrastructure provider, their management chain, and the CIO


End objective of providing a functional and secure operational infrastructure for all users, within tight budget constraints (competing for internal dollars)

Questions to Consider

As a software developer, am I responsible for:

- Following secure programming practices?
- Eliminating known vulnerabilities during design?
- Protecting my work from viruses and other compromises?
- Identifying suspicious behavior on my system and network?
- Minimizing rework and downtime?
- Backup and recovery of my critical data?
- Ensuring that the software I rely on (such as the operating system, applications packages, tools, other COTS) is secure?





Networked Systems Survivability

Questions to Consider (cont.)

As a SEPG member

- Do I consider security improvement as within my area of interest/responsibility? If not, why not?
- What have I learned about making SPI work that could aid in bringing about a continuous security improvement process?
- Am I not in one of the best possible positions to help make this happen?


© 2003 by Carnegie Mellon University

SEPG 03 Presentation - slide 15

SEPG members and those actively involved in SPI are typically boundary spanners, change agents, and excellent communicators. These are unique skills in an organization and can be applied more broadly than just SPI. The very skills SEPG members use every day can aid in mitigating the risks of an incident and bringing about security improvement.

Security concerns are also important to SEPGs. Not paying attention to security could adversely affect an SEPG's ability to be successful:

- quality of life for employees; poor security and security breaches can affect morale
- product quality
- reputation
- the organization's ability to make and keep commitments
- sense of urgency/heightened awareness and visibility since Sept 11
- parallels with physical security
- confidentiality and protection of proprietary data (e.g., assessment data) as a critical asset



Why Is Security Improvement So Hard?

- Abstract, concerned with hypothetical events
- A holistic, enterprise-wide problem; not just technical
- No widely accepted metrics
- Disaster-preventing rather than payoff-producing (like insurance)
- Installing security safeguards can have negative aspects (added cost, diminished performance, inconvenience)

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 16

Why is security improvement so hard to sell? Implement?

- Security, for most, is abstract, concerned with hypothetical events
- Security is a holistic, not just a technical, problem:
- Technological, organizational, regulatory, economic, and social aspects interact
- No widely accepted metrics for characterizing security or (de facto) standards of best practice
- Cybersecurity today is far worse than what known best practices can provide
- Security measures are disaster-preventing rather than payoff-producing (like insurance)
- Benefits can be seen only in events that do not happen (impossible to prove a negative); same difficulty has stalked efforts to improve software quality, conduct proper testing, keep documentation up-to-date, maintain current configuration and hardware/software inventory records, etc. [Braithwaite 01]
- Installing security safeguards has negative aspects (added cost, diminished performance, inconvenience, etc.)

Some, possibly all, of these characteristics currently apply or at one time did apply to SPI.

“Cybersecurity Today and Tomorrow: Pay Now or Pay Later,” Computer Science and Telecommunications Board, National Research Council. National Academy Press, Washington, DC. Prepublication edition. <http://books.nap.edu/html/cybersecurity/>

With respect to 6th and 7th bullets above: “The principal accusation was that Y2K was a relatively minor problem that had been created by consultants to obtain work and that the whole thing had been greatly over-hyped. Accusers held that the accusation was true simply because nothing much happened at the rollover. It is important to acknowledge this type of thinking because of the distinct possibility that architects of a successful cyber security program may find themselves under attacks similar to those leveled at the solvers of Y2K.” Timothy Braithwaite. ““Executives Need to Know: The Arguments to Include in a Benefits Justification for Increased Cyber Security Spending.” *Information Systems Security*, Auerbach Publications, September/October 2001.



Agenda

Motivation


Perspectives/Questions

Protecting Critical Assets

Security Knowledge in Practice

Role of SEPG?





Networked Systems Survivability

Security Improvement

Security

- preserving confidentiality, integrity, availability
- avoiding critical asset disclosure, modification, loss/destruction, interruption

Improvement

- assessment
- action planning
- taking action
- feedback

Risk management (enterprise-wide, not at KPA level)

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 18

Confidentiality: the need to keep proprietary, sensitive, or personal information private and inaccessible to anyone who is not authorized to see it

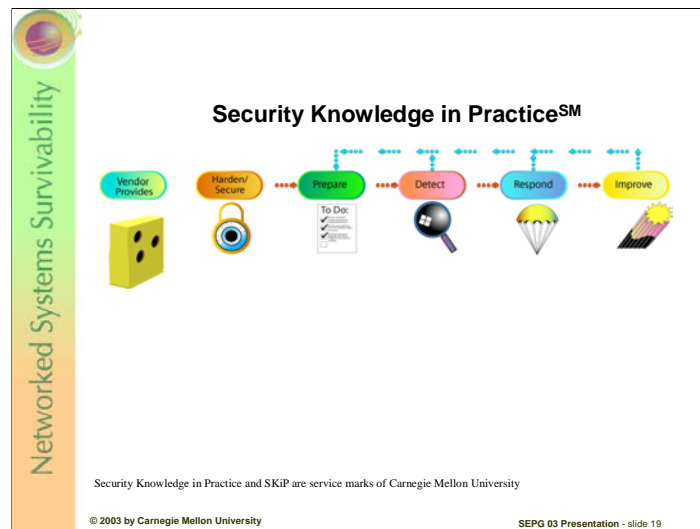
Integrity: the authenticity, accuracy, and completeness of an asset; assuring that information and programs are changed only in a specified and authorized manner

Availability: when or how often an asset must be present or ready for use

Risk: the possibility of suffering harm or loss. It is the potential for realizing unwanted negative consequences of an event. Risk refers to a situation where a person could do something undesirable or a natural occurrence could cause an undesirable outcome, resulting in a negative impact or consequence.

Risk Management: the ongoing process of identifying risks and implementing plans to address them

Alberts, Christopher and Dorofee, Audrey. "OCTAVE Method Implementation Guide, Version 2.0." Pittsburgh, PA: Software Engineering Institute, Carnegie Mellon University, 2001. Information available at <http://www.cert.org/octave>.



[Software developer correlates to “Vendor Provides”]

[Consider the application of this method to the protection of assessment data as a critical asset - residing on a system and accessible via an internal network.]

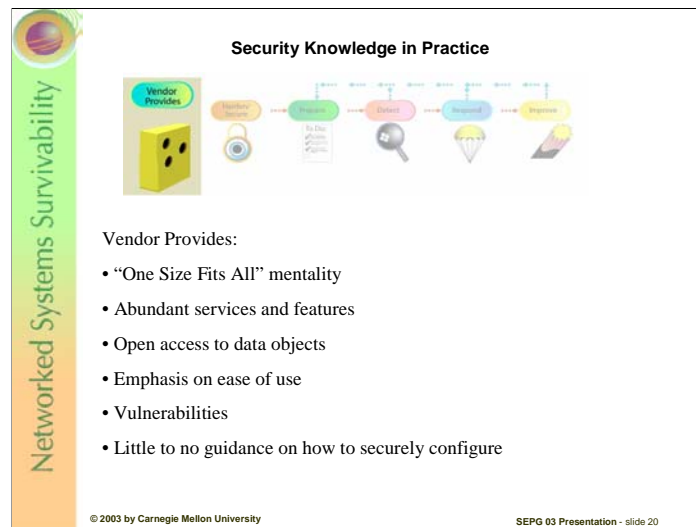
All SKiP descriptions are taken from Rogers, Larry and Allen, Julia. “Securing Information Assets – Security Knowledge in Practice.” Crosstalk: The Journal of Defense Software Engineering, Nov 2002. Available at <http://www.stsc.hill.af.mil/Crosstalk/crostalk.html>.

CERT security practices are for mid-level system administrators who operate networks of 10-100 computers for Fortune 500 companies, critical infrastructures, the Department of Defense, and civil agencies. The practices contain guidance that enables experienced administrators to protect systems and information against both malicious and inadvertent compromises. They can protect systems against 75-80 percent of intruder attacks (as reported to the CERT/CC). The practices are operating-system-neutral for broad application.

The Security Knowledge in Practice (SKiPSM) Method was developed to organize security practices published on the CERT/CC web site into a more process-based approach, departing from the more common problem-based approach. SKiP defines a cyclical process for establishing and sustaining the security of critical information assets such as

- systems running mission critical applications
- network infrastructure, including routers, hubs, and switches
- subsystems or sub-networks such as those providing email services, web content production and delivery services, and perimeter protection services
- a network architecture and topology
- sensitive or proprietary information such as mission logistics, customer data, or financial projections
- computers installed at home

SKiP defines seven steps, actions for each, and a well-defined order for executing the steps. For a less experienced administrator, SKiP provides a road map of the practices necessary to build and sustain the security of an information asset. For the more experienced administrator, SKiP provides an ordered arrangement and a checklist of practices, allowing an administrator to identify gaps in the tasks they are already performing. Once an administrator understands SKiP steps, they are repeated for the life of the asset. See also <http://www.cert.org/security-improvement/skip.html>



Vendors sell systems configured so their customers will be eager to buy. Often, systems are general purpose, that is, fully featured with most of the software enabled for ease of use. They are meant to satisfy everyone's needs and, perhaps, some they didn't realize they had. Such systems frequently contain

- services that are unneeded and often insecurely configured
- little to no protection on access to data objects such as files and directories
- ease-of-use features often provided at the expense of security
- vulnerabilities that intruders can use to break into systems

In today's marketplace, a vendor provides an administrator with a product containing the operating system and an assortment of software applications and tools. While some are needed for the system to function, others are provided to accommodate the demand for an *any-purpose* box. Unfortunately, this "one size fits all" mentality will continue as long as we continue to purchase a vendor's products instead of demanding they produce a more secure product. Once customers start "voting with their dollars," the responsibility for securely configuring systems will shift from administrators to vendors.

Until then, the system administrator's role is to identify the tasks that will be performed by the system and determine the necessary (minimum essential) functions to meet the organization's goals, eliminating those that are unnecessary. Securing a system is challenging, especially for a novice administrator. As a result, it is often considered unnecessary, low priority, or virtually impossible. The SKiP method helps an administrator make this security task more orderly and manageable.

In 2001, the CERT/CC received 2,437 vulnerability reports, more than double the number reported in 2000. In 2002, we received 4,129 reports, a 70% increase. These vulnerabilities are caused by software designs that do not adequately protect Internet-connected systems and by development practices that do not focus sufficiently on eliminating implementation flaws that result in security vulnerabilities.

Networked Systems Survivability

Security Knowledge in Practice

Harden/Secure:

- Configure operating system as the minimum essential (disable/remove unneeded software/services)
- Install applicable patches
- Use secure applications where available
- Install tools such as virus scanners
- Close lenient access controls (deny first, then allow)
- Enable logging

© 2003 by Carnegie Mellon University SEPO 03 Presentation - slide 21

Harden/Secure configures a system to meet an organization's security requirements, retaining only those services and features needed to address specific business needs. By removing unnecessary functionality, an administrator begins to harden the system. However, simply limiting functionality is not enough; administrators need to configure remaining functions correctly (such as installing necessary patches) to sustain a stable and secure configuration.

This step strengthens a system against *known* attacks by eliminating vulnerabilities and other weaknesses commonly used by intruders. The practices performed during this step may change over time to address new attacks and vulnerabilities.

Here are some examples of Harden/Secure practices:

- Install only the minimum essential operating system features. Disable and remove unneeded software. The fewer the services and software on a system, the harder it is to access that system through available means. For example, remove the FTP server and client if the system is not expected to need or provide FTP service.
- Install all applicable patches that correct known deficiencies and vulnerabilities.
- Install the most secure, current versions of system applications.
- Replace applications that contain known vulnerabilities. For example, on a UNIX system, remove telnet and the Berkeley r-commands [rpc, rdate, rdist, remsch, rlogin, rpcinfo, rsh, rksh, rup, ruptime, rusers, rwho] and replace them with SSH, the Secure Shell.
- Install tools needed to operate a system securely. Examples are tools that scan for viruses, characterize a system's behavior (Tripwire; <http://www.tripwiresecurity.com>), and perform secure administration (SSH; <http://www.openssh.org>).
- Remove all privileged and lenient (too weak or open) object access controls. This follows the principle of "deny first, then allow." Grant privileges and access only as needed. Remove all default accounts.

If at all possible, we recommend performing Harden/Secure on a system that is not attached to any network. This minimizes opportunities to compromise the system while it is being built. And install the operating system, patches, and tools from removable media such as zip disks or CD-ROMs.

Networked Systems Survivability

Security Knowledge in Practice

Prepare:

- Characterize files and directories, the operating system, processes, network traffic and performance, and inventory all hardware
- Develop intrusion detection and response (IDR) policies/procedures
- Manage data collection mechanisms
- Select, configure, and install IDR tools

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 22

One of the essential concepts behind the Prepare step is that undiscovered vulnerabilities exist. This requires an administrator to be able to recognize when previously *unknown* vulnerabilities start to be exploited. To support such recognition, it is important to characterize a system, understand its normal behavior in an operational setting, and be able to determine departures from normal.

Characterization entails examining a system's operation and performance under normal conditions and recording expected behavior as the system's known, baseline state. This baseline state contains information (also called attributes) about expected changes at the network, system, process, user, file, directory, and hardware levels. Once a trusted baseline state of system attributes is captured, an administrator compares attributes of an executing system to the baseline to learn if something has changed and then judges whether or not the change is acceptable and expected.

One way to think about the distinction between Harden/Secure practices and the characterization practices in Prepare is that hardening attempts to solve known problems by applying known solutions, whereas characterization helps to identify new problems and formulate new solutions. In using a characterization baseline for comparison, problems can be identified through anomaly-based detection techniques, that is, departures from normal behavior, so that new solutions can be formulated and applied.

After completing characterization, an administrator knows

- the expected changes in files and directories and the operating system
- the expected list of processes, when they run, by whom, and what resources they consume
- the expected network traffic consumed and produced
- the expected hardware inventory on the system

Networked Systems Survivability

Security Knowledge in Practice

Prepare:

- Characterize files and directories, the operating system, processes, network traffic and performance, and inventory all hardware
- Develop intrusion detection and response (IDR) policies/procedures
- Manage data collection mechanisms
- Select, configure, and install IDR tools

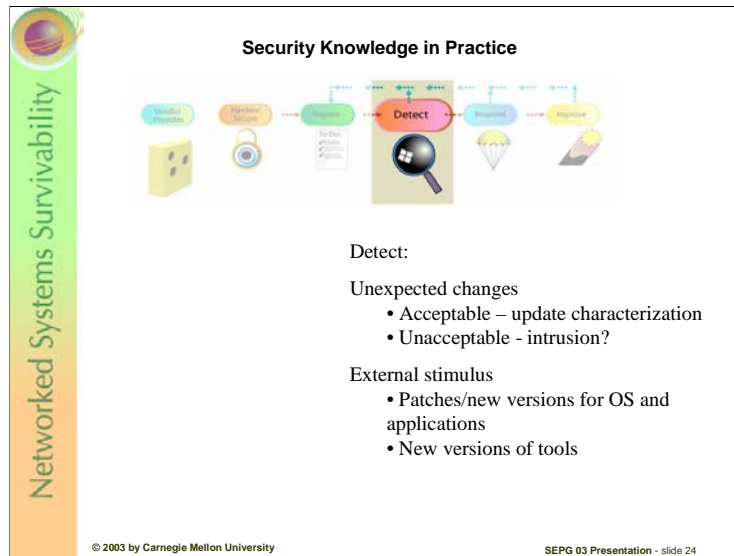
© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 23

[notes continued]

Characterization needs to be performed periodically so that an administrator always has an accurate, up-to-date characterization baseline with which to compare. Characterization information must be recorded and stored securely so it can serve as a reliable basis for comparison. Use WORM (Write Once, Read Many) media such as CD-ROM. We also recommend the use of cryptographic checksums and digital signatures like those provided with PGP [<http://www.pgp.com>] to help ensure characterization baseline integrity. See also “Is There An Intruder in My Computer” by Larry Rogers, available at <http://www.cert.org/archive/html/feature/green.html>.

Additional practices in the Prepare step include

- instigating the development of policies and procedures to
 - identify critical assets, threats to those assets, and possible response actions
 - determine the priority and sequence of detection and response actions
 - specify the authority to act when an intrusion is detected
 - form and operate a computer security incident response team or equivalent capability
 - define what data to collect, where and when to collect it, and the means for its review and protection
 - assign necessary roles and responsibility
 - ensure users are adequately trained
 - ensure your organization is legally compliant with all laws and regulations
- managing data collection mechanisms (such as logging and monitoring tools) and the outputs they produce. Enable as much system logging as possible to provide an audit trail of all system activities. This information aids an administrator in understanding what happened when an incident occurs.
- understanding, selecting, configuring, installing, and maintaining tools for intrusion detection and response. Such mechanisms must be in place well before they need to be used.



An administrator needs to regularly monitor the hardened and prepared system to detect changes. While some of these changes are predictable and constitute normal behavior, administrators must concentrate on detecting signs of anomalous, unexpected behavior, especially those anomalies that indicate possible intrusions and system compromise. An administrator can also watch for early warning signs of potential intruder action such as scanning and network mapping attempts.

Detect occurs while monitoring a system running in a production environment (such as looking at the logs produced by a firewall system or a public web server). An administrator

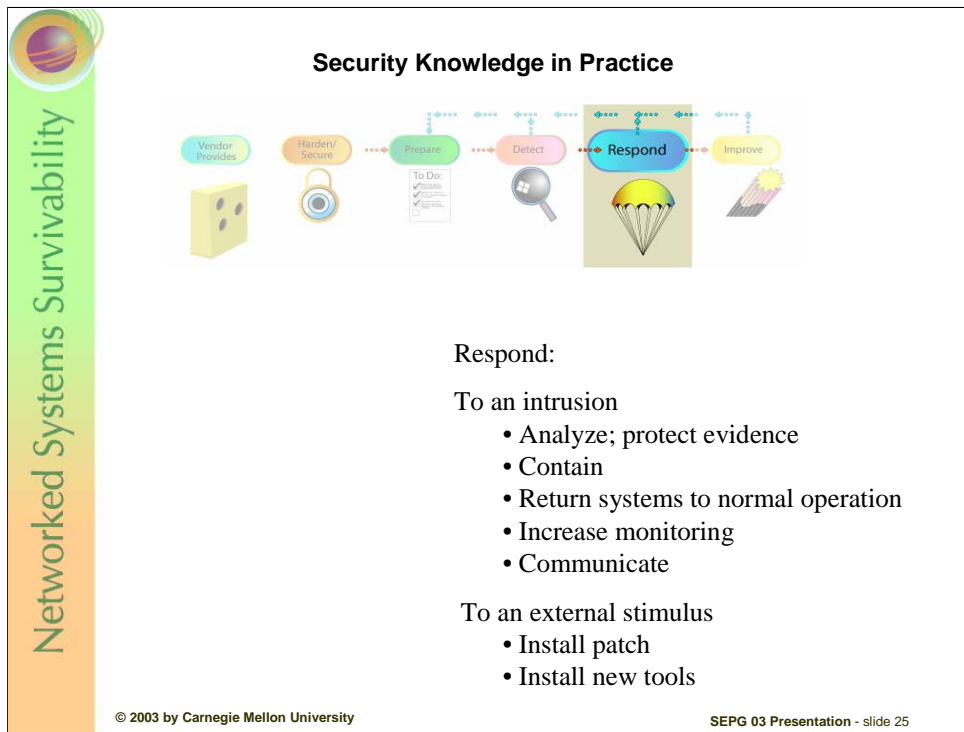
- notices some unusual, unexpected, or suspicious behavior
- learns something new about the system’s characteristics
- receives information from an external stimulus (a user report, a call from another organization, a security advisory or bulletin)

These indicate either that something needs to be analyzed further or that something on the system has changed or needs to change (a new patch needs to be applied, a new tool version needs to be installed, etc.). Analysis includes investigating unexpected behavior that may be the result of an intrusion and drawing some initial conclusions, which are further refined during the next step, Respond. In other words, sufficient analysis is performed during Detect to determine what next action to take.

An administrator uses many of the same tools and procedures that generated the system characterization baseline to detect signs of intrusion. The difference is that characterization results from the currently executing system are compared against the trusted baseline. An administrator’s task is to reconcile the differences between that baseline and the new behavior.

There are two possibilities when differences occur:

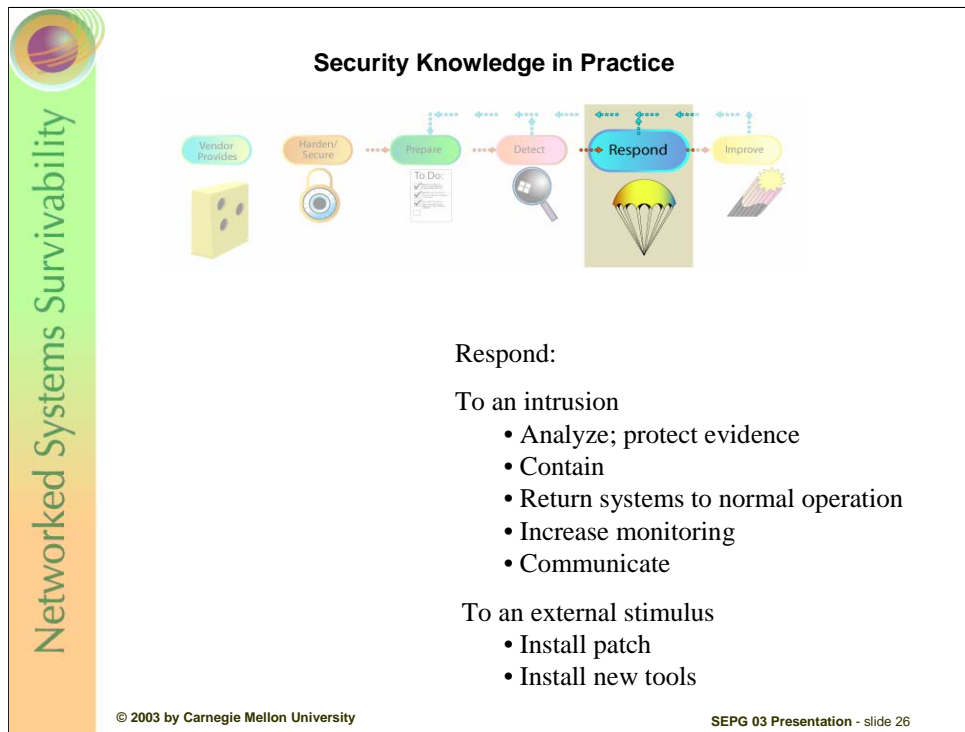
1. An administrator did not or was not able to accurately characterize the system and the discrepancies represent previously unknown but acceptable behavior. An administrator returns to Prepare to update the characterization baseline, create a new CD-ROM, and update checksums and digital signatures where appropriate.
2. The difference is truly anomalous and indicates that something has been tampered with on the system. An administrator moves to Respond and proceeds with those practices.



Once an intrusion is discovered, an administrator performs several practices to respond to the problem and contain it. The response can be as simple as taking little or no action — accepting the risk — to taking significant steps to contain and recover from the intrusion. Successful response means that the system is returned to the normal operational capability that existed before the compromise.

If the transition to Respond results from anomalous behavior caused by an intrusion, an administrator

- further analyzes the effects of damage caused by the intrusion and its scope. Dealing with the effects of an intrusion may result in the insertion of new technology, practices, procedures, and personnel.
- contains these effects as far as possible. One of the challenges administrators face in responding is deciding when they have learned enough about that intrusion so that they can take the appropriate recovery steps vs. continuing to monitor an intruder’s actions in order to discover all access paths and entry points. It is a delicate balancing act. If an administrator does not discover and eliminate all intruder access paths, then it is likely that the intruder will return. However, if the intruder is allowed to roam through systems, then the damage caused to an organization’s assets may be fatal. Identifying and containing the full effects of an intrusion can be a very difficult task and can take a long time.
- works to eliminate future intruder access. Part of the analysis involves discovering how the intruder gained access. Frequently, there is a strong push to return a system to operation even if it means recovering to a previous but vulnerable and insecure state. In this event, the compromised system state is lost, and so are the indicators of how the intrusion happened. The key is to consciously decide that this is the desired course of action and to recognize the ramifications and risks associated with that decision.



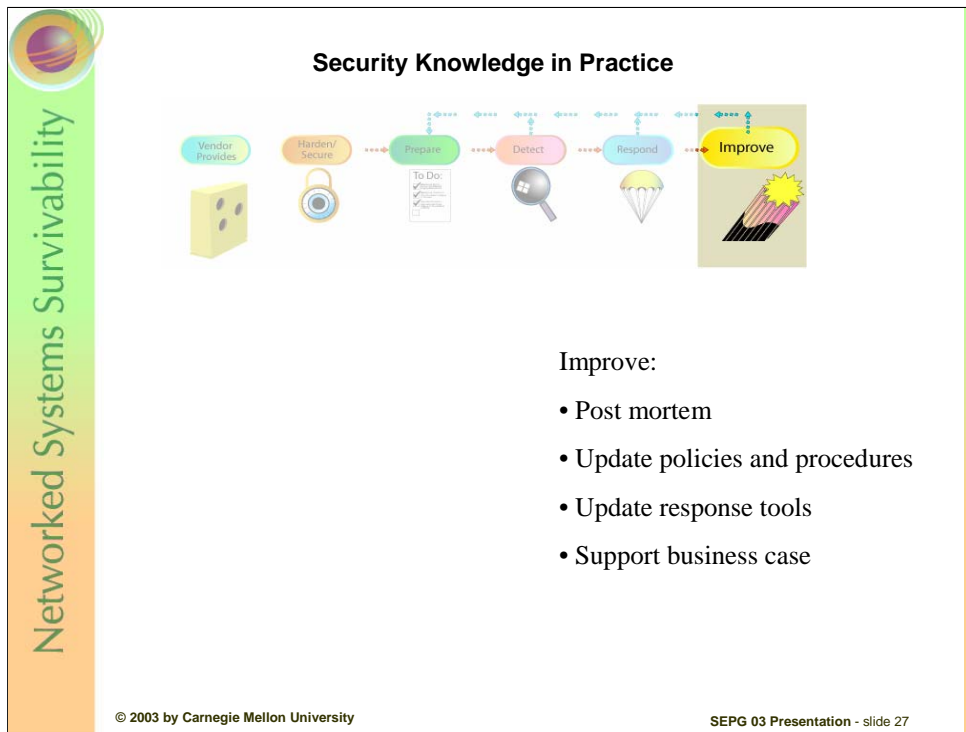
[notes, continued]

- returns the system to a known, operational state while continuing to monitor and analyze. One of the goals of responding is to return the system to a usable state that is an improvement over the previous state that resulted in the intrusion. Once the system is returned to operation, it is a heightened target for intruders, who may be planning to gain access through back doors they have installed. An administrator may learn more about the intruder's attack methods (and therefore the required defenses) through more detailed monitoring and analysis.

While these activities are going on, an administrator notifies all other parties that may be affected, conveying concise and accurate reports of the intruder's activities and the response actions taken to the proper party. This notification must be in concert with the organization's information dissemination policy.

An administrator must collect and protect information that may become evidence in possible legal proceedings, regardless of the organization's policy on prosecuting intruders. An organization must assume that other sites affected by the intrusion will request this information for use in their prosecutions, perhaps by subpoena.

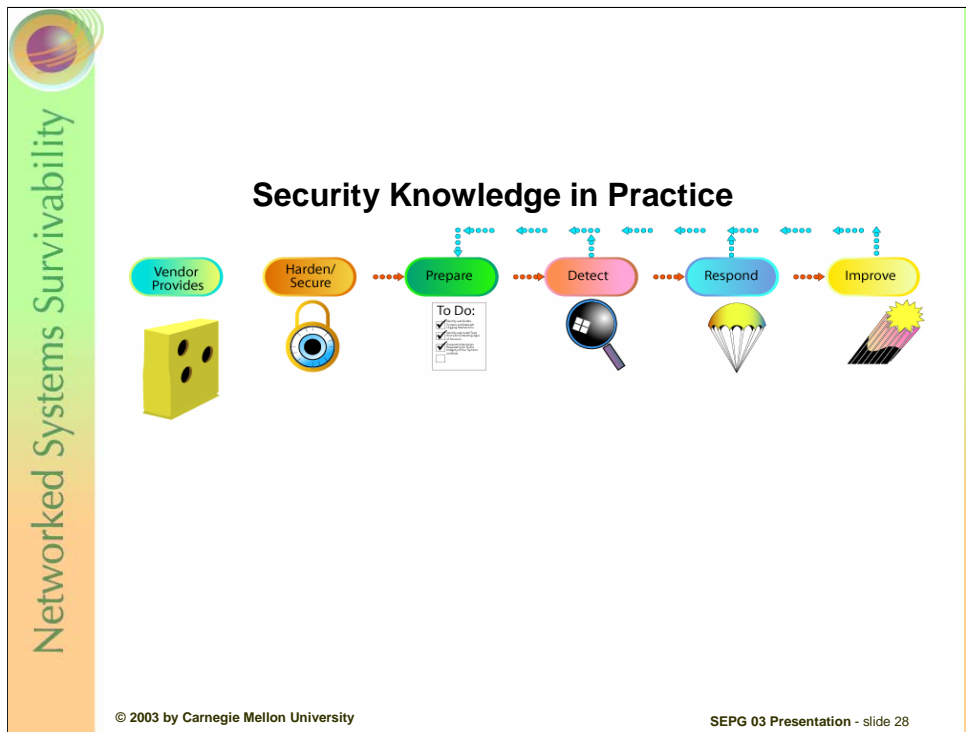
If the transition to Respond is the result of another external stimulus, it is addressed and then the SKiP process transitions to Improve. For example, if the stimulus is the release of a vendor's patch, the administrator applies the patch and this action constitutes completion of the Respond step. Respond activities take place with the system operating in a production mode. Systems may be unavailable while repairs are made.



Improvement actions typically occur following Detect or Respond.
Improvement actions may include

- holding a post-mortem review meeting to discuss lessons learned
- updating policies and procedures
- update existing tools
- collecting business case measures, including the resources required to deal with the intrusion and impacts resulting from the intrusion such as loss of user productivity and administrator time. It is important to quantify the cost of the intrusion, in order to effectively reallocate resources and better prepare for future attacks. This information often serves as the most compelling argument to convince management to allocate sufficient resources to address security issues. At a minimum, capture staff effort (hours, weeks) and capital investments.

Improve takes place when the system is operating in a production mode, though we strongly recommend updating and executing tools in a test environment before deploying them.




Repeat

Any changes made during Detect, Respond, and Improve are factored into the system's characterization baseline by returning to the Prepare step. Note that this repetition is different from the initial one because the practices are performed when the system is operating in a production mode.

Summary

This presentation described the seven steps in the SKiP method and the practices comprising each step:

- *Vendor Provides* systems that are general-purpose and need to be handcrafted to meet an organization's needs.
- *Harden/Secure* the system against known problems.
- *Prepare* the system so the administrator will be able to spot anomalies that may indicate the occurrence of unknown problems.
- *Detect* those anomalies and other changes.
- *Respond* to them when they occur.
- *Improve* the practices and procedures after updating the systems.
- *Repeat* the process as long as the organization needs to protect the information assets on the system and the system itself. SKiP should be followed until the system is retired.



Identifying Risks to Critical Assets: OCTAVESM

- Self-directed method for evaluating information security risks
- Conducted in three phases
- Elicits knowledge from multiple levels of the organization
- Identifies critical assets and threats to assets
- Identifies vulnerabilities that expose threats
- Develops a protection strategy and risk mitigation plans

Operationally Critical Threat, Asset, and Vulnerability Evaluation and OCTAVE are service marks of Carnegie Mellon University

© 2003 by Carnegie Mellon University

SEPG 03 Presentation - slide 29

Information systems are essential to most organizations today. However, many organizations form protection strategies by focusing solely on infrastructure weaknesses; they fail to establish the effect on their most important information assets. This leads to a gap between the organization's operational and information technology (IT) requirements, placing the assets at risk. Current approaches to information security risk management tend to be incomplete. They fail to include all components of risk (assets, threats, and vulnerabilities). In addition, many organizations outsource information security risk evaluations. The resulting evaluation may not be adequate or address their perspectives. Self-directed assessments provide the context to understand the risks and to make informed decisions and tradeoffs.

The first step in managing information security risk is to understand what your risks are. Once you have identified your risks, you can build mitigation plans to address those risks.

Asset: something of value to the enterprise. Information technology assets are the combination of logical and physical assets and are grouped into the specific classes (information, systems, software, hardware, people).

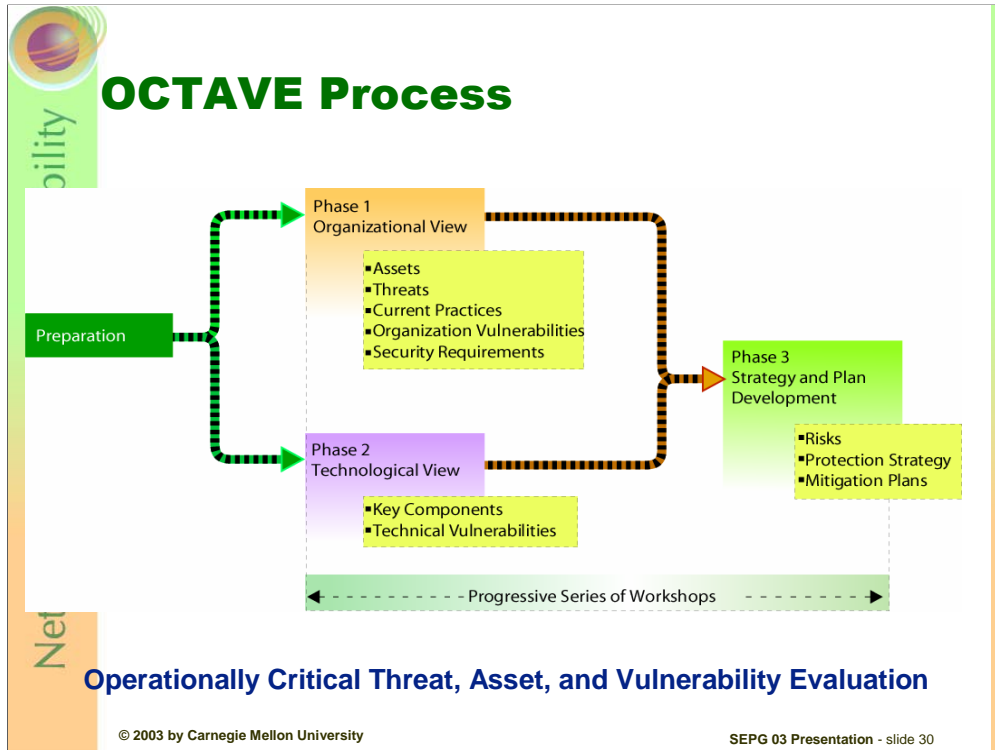
Critical assets: those that are most important such that an organization will suffer a large adverse impact if something happens to these assets.

Threat: an indication of a potential undesirable event. It refers to a situation in which a person could do something undesirable (an attacker initiating a denial-of-service attack against an organization's email server) or a natural occurrence could cause an undesirable outcome (a fire damaging an organization's information technology hardware). Threats have defined properties (asset, actor, motive, access, outcome).

Vulnerability: a weakness in an information system, system security practices and procedures, administrative controls, internal controls, implementation, or physical layout that could be exploited by a threat to gain unauthorized access to information or disrupt processing. There are two basic types of vulnerabilities (organizational and technology).

OCTAVE security requirements: confidentiality, availability, integrity with the impact being a loss or failure to meet such requirements

OCTAVE areas of concern: disclosure, modification, loss/destruction, interruption



The OCTAVE process has three phases:

- Phase 1 takes an organizational view, identifying the critical information-related assets of the organization, their security requirements, the threats to those assets, and the current organizational strengths and weaknesses relative to a set of known security practices.
- Phase 2 is a technology view, similar to the vulnerability tools and assessments usually conducted on a system. This phase uses the information from Phase 1 to focus the technology view to those key components that support the critical assets.
- Phase 3 takes all of the information from the first 2 phases and identifies the critical risks to the organization's critical assets. The organization-wide protection strategy and asset-specific mitigation plans are developed during this phase.



Agenda

Motivation

Perspectives/Questions

Protecting Critical Assets

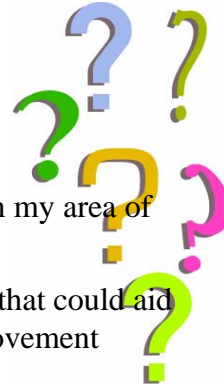
Role of SEPG?




Questions to Consider

As a SEPG member

- Do I consider security improvement as within my area of interest/responsibility? If not, why not?
- What have I learned about making SPI work that could aid in bringing about a continuous security improvement process?
- Am I not in one of the best possible positions to help make this happen?



Consider: What would it take to get the SPI community, in general, and SEPGs, in particular, engaged in security improvement across the full life cycle, as innovators and early adopters?



Networked Systems Survivability

Opportunity for SEPGs

Next big improvement push?
A legitimate technology improvement process?
Career opportunity? SEPG members are in the ideal position.

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 33

This is an emerging area of high interest and possible crisis. It is becoming an increasing risk to achieving business goals.

Security improvement demands a mix of skills that SEPG members already have; view this as a career opportunity. Consider assisting your organization in establishing a CSIRT (Computer Security Incident Response Team). See <http://www.cert.org/training>.

Think about similarities to working Y2K.

How Information Security Helps Achieve Business Objectives

Creates value by removing barriers to conducting business

Facilitates appropriate and authorized access

Enhances and protects business reputation, brand image, customer confidence, and market value

Allows new distribution channels and revenue streams

Can serve to increase profits, not just reduce risk

[Worstell 00]

Worstell, et al. "Net Present Value of Information Security." Developer.com, courtesy of Atomic Tangerine, October 2000. Available at:

Part I: <http://www.developer.com/tech/article.php/640831>

Part II: <http://www.developer.com/tech/article.php/640841>

Part III: <http://www.developer.com/tech/article.php/640851>

Part IV: <http://www.developer.com/tech/article.php/640881>

"Value that is created when barriers to business are removed through mechanisms that ensure business integrity, service availability, and customer/consumer confidence and privacy.

"Realized when appropriate access is facilitated so that businesses can run seamlessly without interruption.


"IS protects business reputation, consumer confidence and market valuations, and it delivers a competitive edge by allowing new distribution channels, revenue streams and even business models in an otherwise diluted and overly compromised marketplace. In other words, instead of being viewed solely as a risk-avoidance measure (like a kind of insurance policy that never pays anything back), information security is required both to support and enable e-business.

"Effective IS can serve to increase business and profits, not merely to reduce risk."

Increasingly, security investments are considered strategic. Along with government and industry regulations and internal compliance audits, customer confidence is a key factor driving companies to invest in information security. Organizations that instill confidence in customers that their personal and business information is safe will have a competitive edge. Ware, Lorraine Cosgrove. "CSOs Prioritize Security Spending for 2003." CSO Online Magazine, January 7, 2003. Available at <http://www.csoonline.com/csoresearch/report50.html>.

"The ability of a corporation to capitalize on future business opportunities may very well depend on whether or not its computer and communications systems are secure." Braithwaite, Timothy. "Executives Need to Know: The Arguments to Include in a Benefits Justification for Increased Cyber Security Spending." *Information Systems Security*, Auerbach Publications, September/October 2001.

Can achieve improved or enhanced reputation and brand image due to increased security for services where security is important to your customer.



Networked Systems Survivability

Six Tips for Selling Security

- Establish Need Before Cost
- Hit 'Em with the Numbers
- Use Others' Loss to Your Advantage
- Put It in Legal Terms
- Keep It Simple

[Field 00]

© 2003 by Carnegie Mellon University SEPG 03 Presentation - slide 35

Field, Tom. "Protection Money." CIO Magazine. October 1, 2000. Available at <http://www.cio.com/archive/100100/money.html>

For CIOs and security experts selling security to reluctant senior managers

Establish Need Before Cost

- how critical business processes are tied to the network
- how much it would cost to secure the systems where these processes reside
- Ask "what do you want to protect?" "This is what it will cost to do it."

Hit 'Em with Numbers

- Such as number of attempted intrusions and viruses, plotted on a graph over time
- present "negative ROI" - analysis of what costs might be incurred without the necessary security protection measures

Use Others' Loss to Your Advantage

- particularly competitors and others in your market segment

Put It in Legal Terms

- Executives and boards of directors receptive to argument that they have a fiduciary responsibility to detect and protect areas where their information assets might be exposed
- business continuity is the best argument

Keep It Simple

- firewall, server protection to scan incoming e-mail, anti-virus software installed on desktops
- user training (change passwords frequently, keep an eye on laptops/floppies, don't open or forward unknown e-mail attachments)
- back up servers frequently

Genusa, Angela. "12 Keys for Locking Up Tight." CIO Magazine, March 1, 2001. Available at <http://www.cio.com/archive/030101/keys.html>.

"If I say, 'I need a million dollars to minimize the chances we will potentially lose a million dollars,' it will be tough to acquire that budget," Mudge says. "It's a lot easier to get that money if I say, 'I need \$1million to enable us to drive more revenue. With our existing architecture, we can do only 1,000 transactions per day, but with this new architecture we could do 5,000.'" Pitched as an opportunity and strategic advantage rather than a potential loss, security becomes a fortuitous byproduct, he says."



Cost Savings Justify an Early Investment in SSE

Spending a dollar to fix a software bug (including a security vulnerability) in the design process saves \$99 against fixing it during implementation. [Berinato 02]

Addressing security during design nets a 21% ROSI. Waiting until implementation reduces that to 15%. During test, the ROSI falls to 12%. [Berinato 02]

“\$1 required to resolve an issue during the design phase grows into \$60-\$100 to resolve the same issue after the application has shipped.” [Soo Hoo 01]

SSE: Security Software Engineering; ROSI: Return on Security Investment

Berinato, Scott. “Finally, a Real Return on Security Spending.” CIO Magazine, February 15, 2002. Available at http://www.cio.com/archive/021502/security_content.html.


Soo Hoo, Kevin. “Tangible ROI through Secure Software Engineering.” Secure Business Quarterly, Volume One, Issue Two, @stake, Fourth Quarter 2001. Available at <http://www.sbq.com/sbq/rosi/>.

Levine, Matthew. “The Importance of Application Security.” @stake, March 2002. Available at http://www.atstake.com/research/reports/acrobat/atstake_application_security.pdf.

Jaquith, Andrew. “The Security of Applications: Not All Are Created Equal.” @stake, February 2002. Available at http://www.atstake.com/research/reports/acrobat/atstake_app_unequal.pdf.

“In an analysis of 45 e-business applications, @stake found that nearly half of application security defects were preventable during the design stage. By making security a key part of the design stage, companies can gain a 21 percent ROI - compared to just 12 percent when security efforts are delayed until the testing phase.”

http://www.atstake.com/research/strategic_security/rosi.html



Networked Systems Survivability


Top Ten Application Security Defects

Session replay/hijacking	31%
Password controls	27%
Buffer overflows	27%
File/application enumeration	27%
Weak encryption	24%
Password sniffing	24%
Cookie manipulation	20%
Administrative channels	20%
Log storage/retrieval issues	20%
Error codes	20%

[Jacquith 02]
© 2003 by Carnegie Mellon University

SEPG 03 Presentation - slide 37

Jaquith, Andrew. "The Security of Applications: Not All Are Created Equal." @stake, February 2002. Available at http://www.atstake.com/research/reports/acrobat/atstake_app_unequal.pdf.



Networked Systems Survivability

Patterns that Differentiate Top Performers

- Early design focus on user authentication and authorization
- Mistrust of user input
- End-to-end session encryption
- Safe data handling
- Elimination of administrator backdoors, mis-configurations, and default settings
- Security quality assurance

[Jacquith 02]
© 2003 by Carnegie Mellon University

SEPG 03 Presentation - slide 38

Application security "in a grim state." Recent research done by a commercial security firm (@stake) suggests that almost half of all application security vulnerabilities are readily exploitable through entirely preventable defects. Security researchers, contrasting the performers with regards to security, say that six areas differentiate top and bottom performers:

- early design focus on user authentication and authorization
- mistrust of user input
- end-to-end session encryption
- safe data handling
- elimination of administrator backdoors and default settings
- security quality assurance.

The most common application security mistake is a lack of adequate authentication and access control. According to researchers, user session security remains the Achilles heel of most e-business applications because user input is trusted implicitly or relies on client-side validation, rather than having the server itself check for inappropriate data.

James Middleton. "Where Are You?" vnunet.com, Feb 19, 2002. Available at <http://www.vnunet.com/News/1129340>.



Designing Secure Software

- Secure the weakest link
- Practice defense in depth
- Fail securely
- Follow the principle of least privilege
- Compartmentalize
- Keep it simple
- Promote privacy
- Remember that hiding secrets is hard
- Be reluctant to trust
- Use your community resources

[Viega 02]

© 2003 by Carnegie Mellon University

SEPG 03 Presentation - slide 39

John Viega, Gary McGraw. *Building Secure Software: How to Avoid Security Problems the Right Way*. Addison-Wesley, 2002.

Gary McGraw. "Managing Software Security Risks." *Computer*, IEEE Computer Society, April, 2002.



For More Information

CERT web site

OCTAVE Method Implementation Guide

The CERT Guide to System and Network Security Practices

CERT Security Improvement Modules

Internet Security Alliance Common Sense Guides

- CERT web site: <http://www.cert.org>
- About the CERT/CC: http://www.cert.org/meet_cert/meetcertcc.html
- OCTAVE Method Implementation Guide: information available at <http://www.cert.org/octave/omig.html>
- *The CERT Guide to System and Network Security Practices* provides a detailed description of the practices necessary to harden and secure a general-purpose server (Chapter 2), a public web server (Chapter 3), and a firewall system (Chapter 4). It also has detailed descriptions of the practices involved in the steps Prepare, Detect, Respond, and Improve; information available at <http://cseng.aw.com/book/0,,020173723X,00.html>
- CERT security improvement modules: <http://www.cert.org/security-improvement>
- Guidelines for senior managers: <http://www.isalliance.org>
- Guidelines for home users: <http://www.isalliance.org>, <http://www.cert.org/homeusers/HomeComputerSecurity/>
- The CERT/CC provides a checklist on securing UNIX-based systems [http://www.cert.org/tech_tips/AUSCERT_checklist2.0.html] and SANS provides guides for Linux, Windows 2000, Windows NT, and Solaris [<http://www.sansstore.org/Templates/frmTemplateK.asp?SubFolderID=22&SearchYN=N>]. An administrator needs to frequently check these resources and others, because Harden/Secure practices change over time.