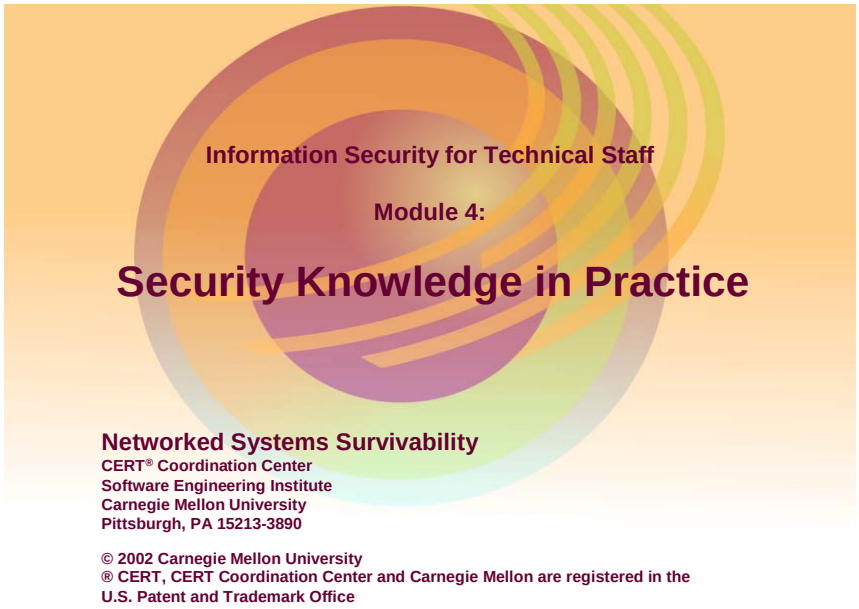
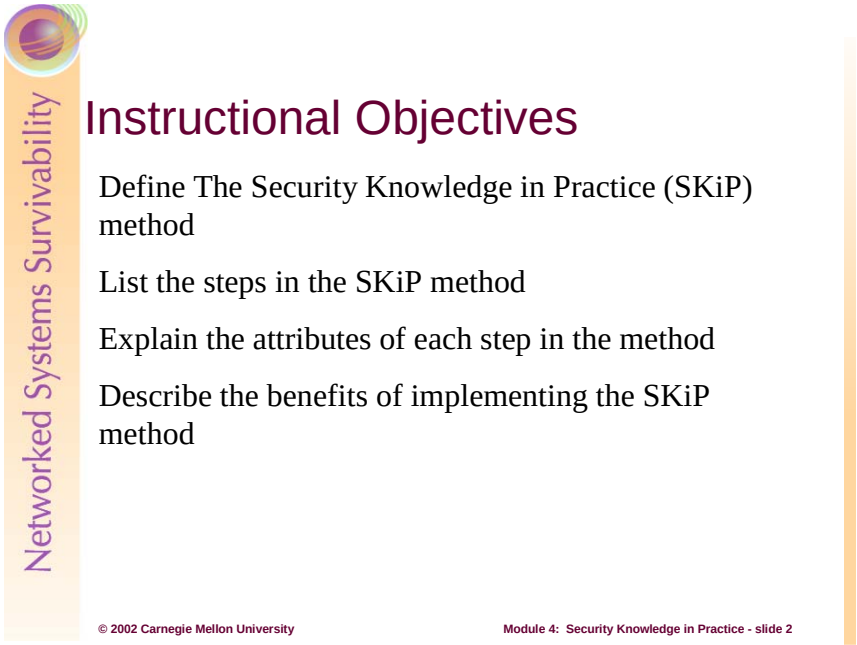
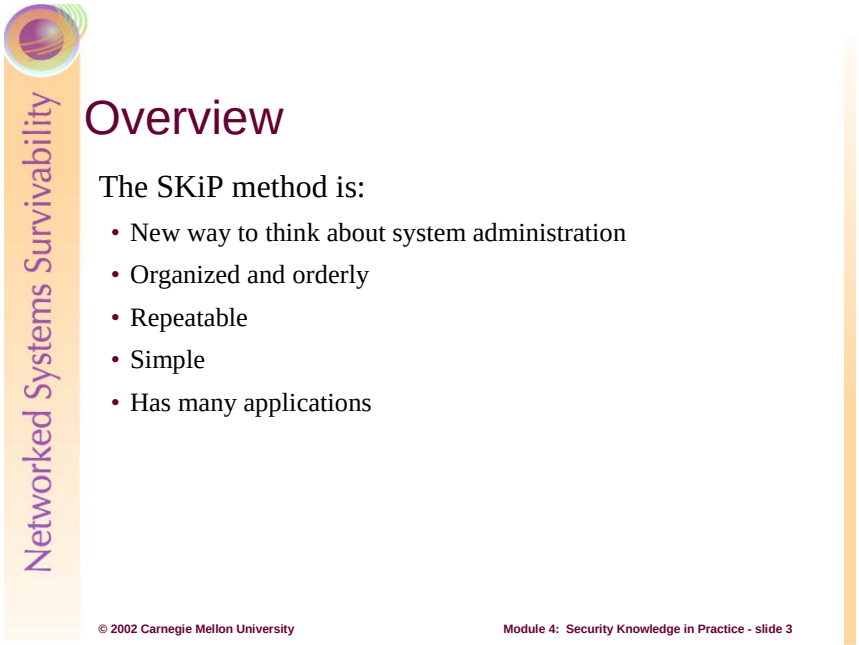


The Security Knowledge in Practice (SKiP) Method defines a continuous process for establishing and sustaining the security of a critical information asset. One example is the traditional computer system running mission critical applications. In addition, the SKiP Method can be applied to the architecture of a network, the systems that comprise a sub-infrastructure (e.g., the collection of computer systems that provide electronic mail or remote access services), and computers installed at home.



In this module, we will:

- Define the SKiP Method – you will learn what it is, where it came from, and the problems it attempts to solve.
- List the steps in the SKiP Method – you will learn the seven steps in the SKiP Method.
- Explain the concepts behind each step and what practices are conducted when applying each step to a system.
- Describe the benefits of implementing the SKiP Method – You will learn how to apply it and why.

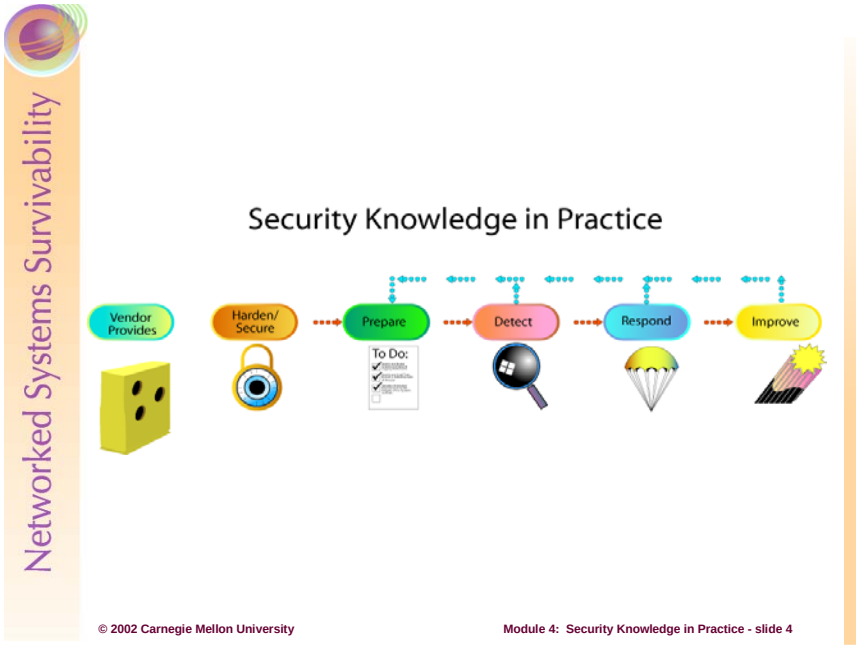


The SKiP Method is a new way to think about systems administration. Unlike previous system administration methods, SKiP defines seven specific and ordered steps, and the actions to be taken at each step. Once the system administrator (SA) is comfortable with SKiP and its steps, these steps are repeated over and over for the life of the system to which they are applied.

SKiP is a method for initially securing and subsequently sustaining the security state of an information asset. Example assets are:

- Systems running mission critical applicationsNetwork infrastructure including routers, hubs, switches, etc.
- Subsystems or sub-networks such as those providing email services, web content production and delivery services, perimeter protection services, etc.
- A network architecture and topology
- Sensitive or proprietary information such as customer data or financial projections
- Computer systems installed at home

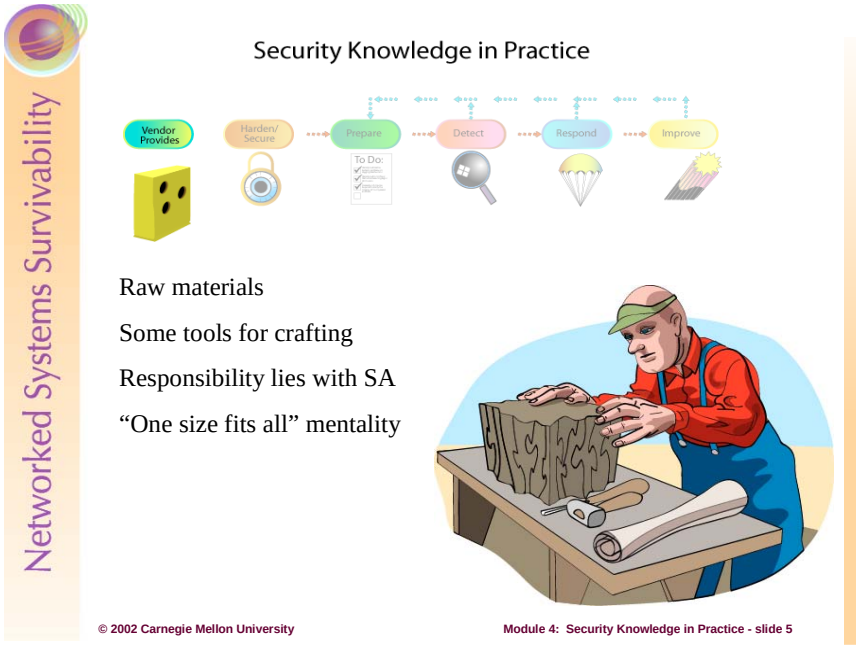
This module explains the SKiP steps and how they are most often applied to a computer system.



The SKiP Method was designed to organize information security practices published on the CERT/CC Web Site[1] into a more contiguous process-based approach, departing from the more common problem-based approach. For the less experienced system administrator, SKiP gives a road map of the major tasks to build and sustain the security of a system. For the more experienced system administrator, SKiP provides an ordered arrangement and a "checklist" of practices, allowing an administrator to identify gaps in the tasks they are already performing.

The process diagram shows the sequence of activities that comprise SKiP. The explanation that follows applies to securing a system that runs a mission critical application suite in a production environment, commonly referred to as a host computer system.

[1] http://www.cert.org/security-improvement/



Vendors sell systems that their customers will buy. In most cases, these systems are general purpose, meaning that they are fully featured with all or 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, unwanted, and most often insecurely configured;
- little to no protections on access to data objects such as files and directories;
- ease-of-use features often provided at the expense of security; and
- vulnerabilities that intruders can use to break into systems

In today's marketplace, a vendor provides an administrator with a "chunk of wood" containing the operating system and an assortment of software tools. While some of these tools are needed for the system to function, many others are provided to accommodate the demand for an *any-purpose* box. This "one size fits all" mentality is precisely what we are asking vendors for by purchasing their products and by not asking for something different, i.e., a product that operates securely. Once administrators and other customers start demanding a more secure system product – that is *voting with their dollars* – the responsibility to adjusting the "shape" of the system will shift from an administrator to the vendor.

In the mean time, 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 to the novice system administrator. As a result, it is often dismissed as being unnecessary, low priority, or virtually impossible. The SKiP Method aids an administrator in making this security task more orderly and more manageable. As a result, it is often considered unnecessary, low priority, or virtually impossible. The SKiP Method aids an administrator in making this security task more orderly and more manageable.

The following is an excerpt from Rich Pethia's September 26, 2001 Congressional Testimony[2]. Rich is the Director of the Networked Systems Survivability Program at the SEI.

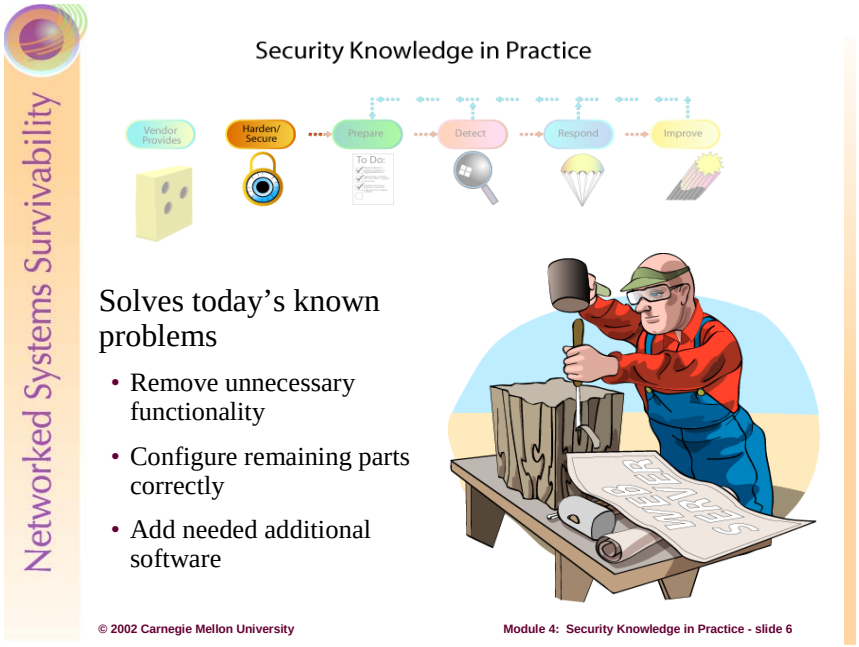
[2] http://www.cert.org/congressional_testimony/Pethia_testimony_Sep26.html

Vulnerability of Technology

Last year, the CERT/CC received 1,090 vulnerability reports, more than double the number of the previous year. In the first half of 2001, we have already received 1,151 reports and expect well over 2,000 reports by the end of this year. 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 problems.

There is little evidence of movement toward improvement in the security of most products; software developers do not devote enough effort to applying lessons learned about the sources of vulnerabilities. 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. Until customers demand products that are more secure or there are changes in the way legal and liability issues are handled, the situation is unlikely to change.

Additional vulnerabilities come from the difficulty of securely configuring operating systems and applications software packages. These products are often shipped to customers with security features disabled, forcing the technology user to go through the difficult and error-prone process of properly enabling the security features they need. While the current practices allow the user to more quickly use the product and reduces the number of calls the product vendor's service center might receive when a product is released, it results in many Internet-connected systems that are mis-configured from a security standpoint.



Harden and Secure seeks to shape a system to meet an organization's security requirements. Given the vendors' propensity to provide fully featured and general-purpose systems, an administrator must remove and constrain the excess software that is part of today's operating systems and applications. Much like the master crafter who transforms a block of wood into a smooth carving, an administrator must carve out of the general-purpose services and features provided by the vendor those that are unneeded, retaining only what is needed to address a specific business need. By removing unnecessary functionality, an administrator begins to harden the system. But hardening also means taking what is known today and applying it to best implement the tools in use today. In other words, simply sculpting the wood to limit functionality is not enough; the wood carver must also realize that wood is susceptible to the environment and needs to be stained and treated. Similarly, administrators need to configure functions and tools correctly to sustain a stable and secure configuration.

This step strengthens a system against *known* attacks by eliminating (where possible) vulnerabilities and other weaknesses that are commonly used in known attacks. The practices performed during this step may change over time to address new attacks and vulnerabilities.

While Module 9 on Host System Hardening goes into greater detail about the hardening process, here are some examples of harden and secure practices:

- Install only the minimum essential operating system features. Disable and remove unneeded vendor software. The fewer software services on a system, the harder it is to access that system through whatever means are available. For example, remove the FTP server and client if the system is not expected to need or provide FTP service. Note also that kernel-based services should also be considered. There are many documents on the web that describe how to do this task for specific versions of various operating systems.
- Install all known and applicable patches that correct known deficiencies and vulnerabilities.
- Install the most secure and current versions of system applications.
- Replace applications that contain known vulnerabilities. For example, on a LINUX/UNIX system, remove **telnet** and the Berkeley **r**-command and replace them with SSH, the Secure Shell.

- Install tools needed to operate a system securely during its production life. Examples are tools that scan for viruses, characterize a system's behavior (Tripwire[3]), detect some types of intrusions through anomalies (also Tripwire) and signature recognition (SNORT[4]), and perform secure administration (SSH[5]).
- Remove all privileged and lenient (too weak or open) object access. This follows the principle of "deny first, then allow." Grant privileges and access only as needed.
- Enable as much system logging as is possible to provide an audit trail of the activities on a system. This information aids an administrator in understanding what happened when an incident occurs.
- Install and configure tools used in other phases of the administration process, for example, The Coroner's Toolkit (TCT[6]), used in performing forensic analysis of a compromised system.

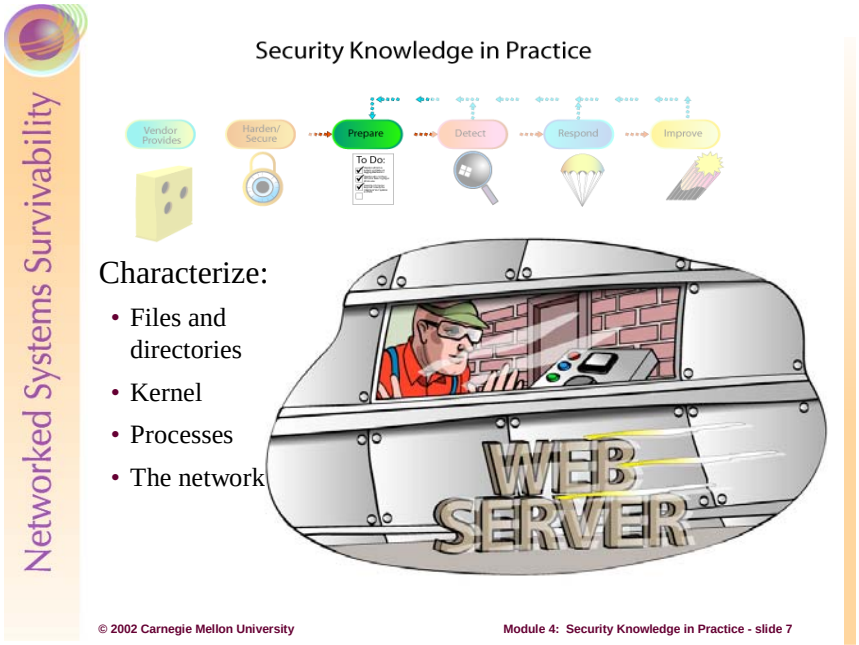
If at all possible, we recommend performing Harden/Secure on a system that is not attached to any network. This minimizes opportunities to compromise the system while it is being built. To do this, the administrator likely must install the operating system, patches, and tools from removable media such as zip disks, CDROMs, etc.

Building the system on an air-gapped test network also helps ensure that the system is not compromised while being built. However, recognize that to be efficient in the process, a network connection may make the task easier.

Applying patches and installing tools serve to harden many systems attached to the Internet. The CERT/CC provides a checklist on securing LINUX/UNIX-based systems[7] (and SANS provides guides for LINUX, Windows 2000, Windows NT, and SOLARIS[8]). There also Host Hardening and Securing documents produced by the NSA[9]. These guides assist an administrator in thinking about the issues to be considered when hardening and securing a system. An administrator needs to keep up-to-date with these and other resources, given that the Harden/Secure tasks change over time.

The CERT Guide to System and Network Security Practices[10] book 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 (Chapter 4).

[3] http://www.tripwiresecurity.com
[4] http://www.snort.org
[5] http://www.openssh.org
[6] http://www.porcupine.org/forensics/tct.html
[7] http://www.cert.org/tech_tips/AUSCERT_checklist2.0.html
[8] http://www.sansstore.org/Templates/frmTemplateK.asp?SubFolderID=22&SearchYN=N
[9] http://nsa1.www.conxion.com
[10] http://www.aw.com/product/0,2627,020173723X,00.html



One of the essential concepts behind *Prepare* is the recognition that a collection of vulnerabilities exists that is yet to be identified (in other words, *unknown*), requiring an administrator to be in a position to recognize when these vulnerabilities have been or are being exploited. To support such recognition, it is vitally important to characterize a system so that an administrator can understand how it works in an operational setting, and be able to determine departures from normal.

Our master crafter prepares his Web Server by inspecting it so that he can learn all about it. By X-raying his Web Server, he can see the knots in the wood, the depth of the hardware he's selected to attach his Web Server to its holder outside of his shop, and all other defects originally in the wood and those introduced by his chiseling. He knows all about his Web Server and is now able to recognize any changes that occur over time.

On a computer system, 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 also contains information (also called attributes) about expected changes at the network, system (including kernel), process, user, file, directory, and hardware levels. Once a trusted baseline state is captured, an administrator subsequently compares attributes of an executing system to the baseline to learn if something has changed and then makes an informed judgment as to whether or not the change is acceptable and expected.

One way to think about the distinction between Harden/Secure and the characterization part 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.

When performing Prepare the first time, a system should be connected to a network that is topologically and architecturally similar to the operational network where the system will ultimately reside. The key is to understand how the system operates in an environment that is as close to normal production as possible.

The SKiP Method is iterative. Subsequent executions of Prepare will produce an updated system characterization. This is highly recommended, as you always want to have an accurate, up-to-date characterization baseline with which to compare.

The system attributes to be characterized in Prepare are:

- Changes to files and directories – Characterization practices seek to identify the types of changes that are made in the file system, that is what files are routinely changed, added, deleted, and what directories are routinely changed, added, and deleted. For example, if the */etc/passwd* file on a UNIX System or *C:\autoexec.bat* changes on a Windows 2000 machine, are those changes routine?

  The previously mentioned Tripwire is a useful tool for discovering how files and directories change. On Windows-based platforms, Tripwire also identifies changes to the registry. The SA can define a database of file attributes and the acceptable changes to them. Once configured, Tripwire's output shows all of the anomalous file, directory, and registry behavior in one report.

- Changes to the operating system – The operating system's foundation as loaded into the system's memory – called the kernel – may change, usually through the addition of device drivers. Knowing the specifics of any changes and whether or not they are acceptable is the key. For example, if a different type of Ethernet controller driver is loaded into your kernel, is that considered routine?

  While there are no products – commercial or otherwise – that give you a complete solution – a "Tripwire for the Kernel" – there are some strategies that can be used to reduce the likelihood that the kernel will unexpectedly change. For example, Windows 2000 and beyond use a technique called driver signing. This gives the administrator more confidence that the driver being loaded into the kernel came from a known and perhaps reputable source. The chances are that the driver will work as advertised and not perturb the kernel in unexpected says.

  Similarly, on some Linux systems, adding drivers and other modules to the kernel can be prohibited as can changes to the special files that reference kernel memory. This means that the kernel can be made unchangeable beyond a specified point in the system boot process.

  In both of these cases, the kernel's integrity is not checked, but controls are used to limit what can be done. This gives the administrator more confidence that the system is running as expected and that an intruder has not altered it.

- Processes and their attributes – When a system operates, users run programs at specific times or under certain circumstances. To fully characterize that behavior, it is necessary to know who runs what programs, when they routinely run, and some notion of the resources that they consume. For example, if a program claiming to be the disk backup program runs at 10:00AM on a weekday and consumes 28Mb of virtual memory, is this normal behavior?

  This functionality can be thought of as "Tripwire for Processes," and some products are emerging that begin to provide this information. Examples include Emerald[11] from SRI, Nabou[12], and Panoptis. These tools can help the system administrator understand more about processes behavior.

- The network – When a system operates, it consumes and produces network traffic. To fully characterize that behavior, it is necessary to know the volume of traffic consumed and produced vs. the time of day vs. the network identity of the consumer/producer. For example, is a large volume of WWW traffic produced for a network address in a foreign country at 2:00AM normal behavior?

  This functionality can be thought of as "Tripwire for the Network." Tools such as Argus[13] describe the connections made vs. time of day vs. network identity. Other useful tools are NetScout[14] and TrafficShaper[15].

---

[11] http://www.sdl.sri.com/projects/emerald/
[12] http://www.nabou.org/
[13] ftp://ftp.andrew.cmu.edu/pub/argus/current/README and http://www.cert.org/security-improvement/implementations/i042.09.html
[14] http://www.netscout.com/
[15] http://www.trafficshaper.com/

- The hardware – When a system operates, the hardware should remain unchanged, that is, adding new hardware such as a modem or another network interface card is likely unexpected. For example, is the addition of a second Ethernet controller on a desktop workstation normal behavior?

  There are many hardware inventory programs, but they are not packaged in such a way that they report the set of changes from a known configuration. Such functionality can be thought of as "Tripwire for the hardware." The system administrator will again have to rely on site-specific tools and procedures to identify changes in a system's hardware inventory and whether or not they are expected.

After completing this task, an administrator has generated a system characterization baseline and knows:

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

An administrator subsequently compares attributes of an executing system to the characterization baseline to learn if something has changed and then make an informed judgment as to whether or not the change is acceptable.

Characterization information must be recorded and stored in a secure manner so that it is trusted and can serve as a reliable basis for comparison. Use WORM (Write Once, Read Many) media such as CDROM. We also recommend the use of cryptographic checksums and digital signatures like those provided with PGP[16] to help ensure characterization baseline integrity.

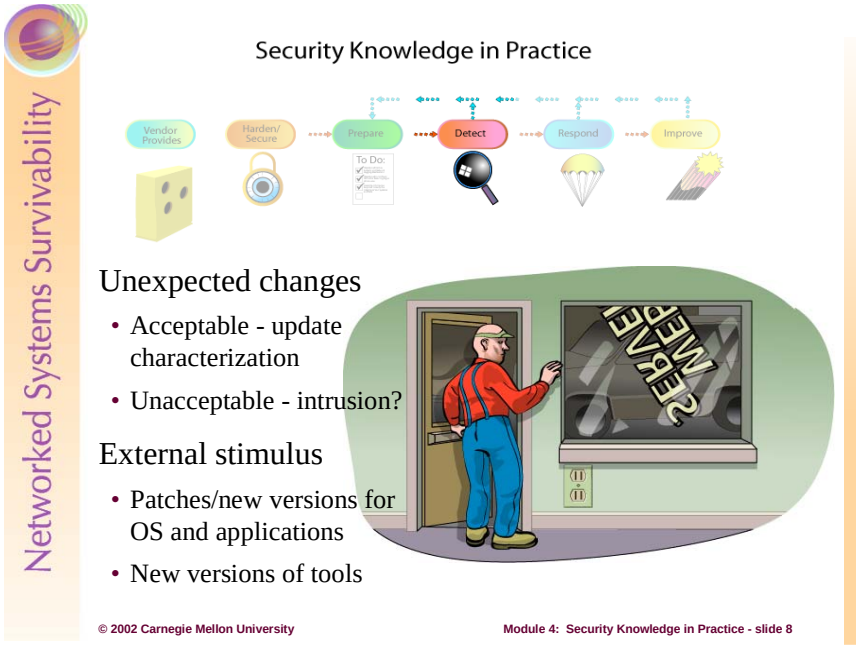
Additional practices in the Prepare step include:

- 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.

After the initial completion of Prepare, the system can be put into operation. Each subsequent execution of this step is done in an operational environment.

[16] http://www.pgp.com

While characterization is the bulk of the tasks done in Prepare, there are more practices to perform. *The CERT Guide to System and Network Security Practices* book provides a detailed description of these practices necessary to prepare a system to detect signs of intrusion and respond to intrusions (Chapter 5).



An administrator needs to regularly monitor the hardened and prepared "chunk of wood" to detect changes. While some of these changes are predictable and constitute normal behavior, what an administrator really wants is to concentrate on detecting signs of anomalous, unexpected behavior and more specifically those anomalies that indicate possible intrusions and system compromise. A caveat is that an administrator can also listen for things that could be detrimental to the system. When the administrator hears about one of them, he can take an action before it can be exploited. In the same manner that the sculptor finds out the wood stain has been recalled due to a chemical anomaly that may be present in the original stain.

Detect occurs while monitoring system running in a production mode (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, suspicious behavior that may be the result of an intrusion and drawing some initial conclusions, which are further refined during the next step, Respond.

An administrator uses many of the same tools and procedures that generated the system characterization baseline to detect signs of intrusion on an operational system. The difference is that these results are compared against the trusted baseline. An administrator's task is to reconcile the differences between that baseline and what has now been found.

There are two possibilities when differences occur:

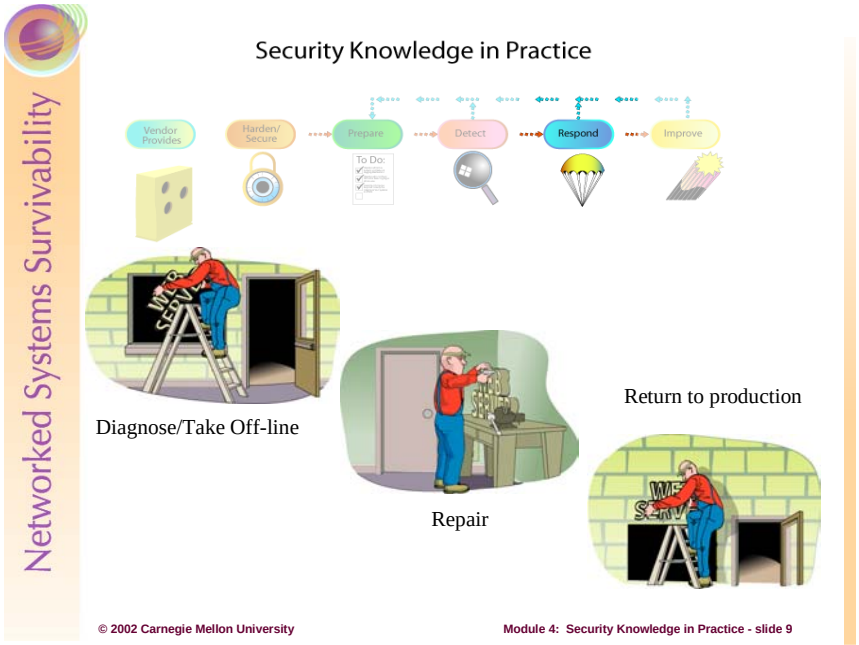
1. An administrator did not or was not able to accurately characterize the system and the discrepancies represent a previously unknown but ultimately acceptable behavior. An administrator need only update characterization information, creating a new CDROM, and updating checksums and digital signatures 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.

In addition, some external stimulus can also cause a transition from Detect to Respond. This includes the release of a relevant patch or a new tool. This transition is normally under administrator control and can be scheduled based on priority and available time.

Detect always occurs when the system is running in a production setting. An administrator continuously looks for signs of an intrusion, reviews operating system and applications vendor information, and reviews tools to add to the arsenal.

The CERT Guide to System and Network Security Practices book provides a detailed description of the practices necessary to detect signs of intrusion (Chapter 6).



Once a problem is discovered, our master crafter needs to do something – Respond – to the problem. In this case, the crafter selected hardware that wasn't up to the task; it wasn't strong enough to hold the Web Server so it crashed to the ground. The crafter needs to repair the wood and then select hardware better suited to the weight of the Web Server.

On a computer systems, and again once a problem is discovered, an administrator performs several practices to Respond to and contain the problem. The response can be as simple as taking minimal to no action by accepting the risk or taking significant steps to contain and recover from the problem. Successful response means that the system is recovered to a configuration and 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, the scope of, and damage caused by the intrusion. An administrator needs to understand what happened, the effects on the business, and any collateral effects to appropriately respond to the intrusion. Dealing with the effects of an intrusion may result in the insertion of new technology, practices, procedures, and personnel. Overreacting by unjustifiably introducing security measures may hamper the organization's ability to conduct business, ultimately leading to a loss or the end of that business.
- Contains these effects as far as possible. One of the many challenges administrators faces in responding to an intrusion 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 so as 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 an extended period of time.
- Works to eliminate future intruder access. Part of the analysis activity 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, as 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 of that decision.

- Returns the system to a known, operational state while continuing to monitor and analyze. One of the goals of responding to an intrusion is to return the system to a usable state. That state should be 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, accurate, and appropriately directed reports of the intruder's activities and the response actions taken. 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 against the intruder, 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.

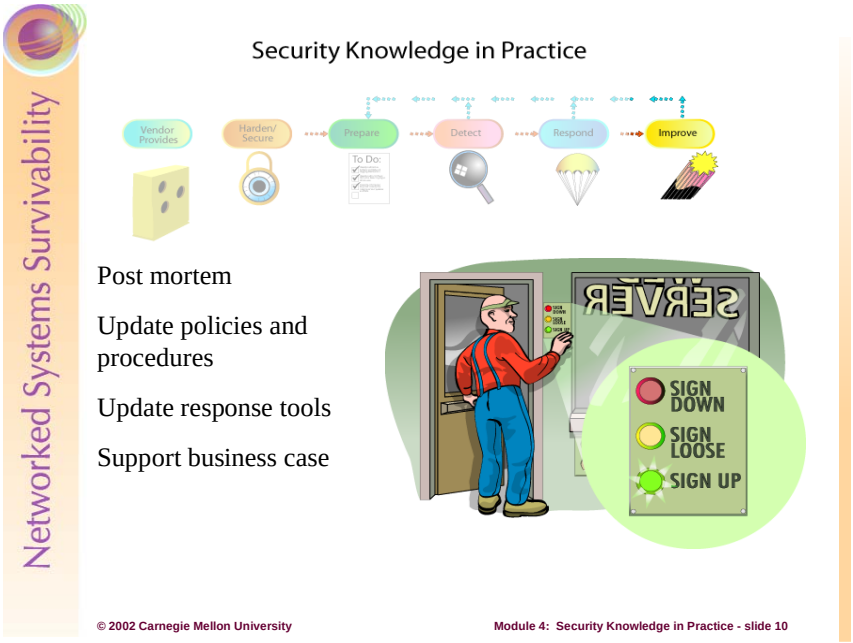
The collection and analysis of intrusion-related information is called *computer forensics*. It is a growing information security discipline characterized by new tools, practices, and procedures. An administrator should track developments in this area including the availability of new analysis approaches and tools.

If the transition to Respond is the result of some other external stimulus, that stimulus is addressed and then the process transitions to Improve. For example, if the stimulus is the release of a patch from a vendor, the administrator applies the patch that constitutes completion of Respond.

Respond activities take place when the system is operating in a production mode. Systems may be unavailable for a period of time while repairs are made.

An administrator needs a test environment to more fully understand the nature of the intrusion. In this environment, an administrator may be able to run a quarantined version of any captured attack tools to learn which vulnerabilities the intruder has used to gain access, thus ensuring that these means of access have been removed before the systems are returned to operation.

The CERT Guide to System and Network Security Practices book provides a detailed description of the practices necessary to respond to intrusions (Chapter 7).



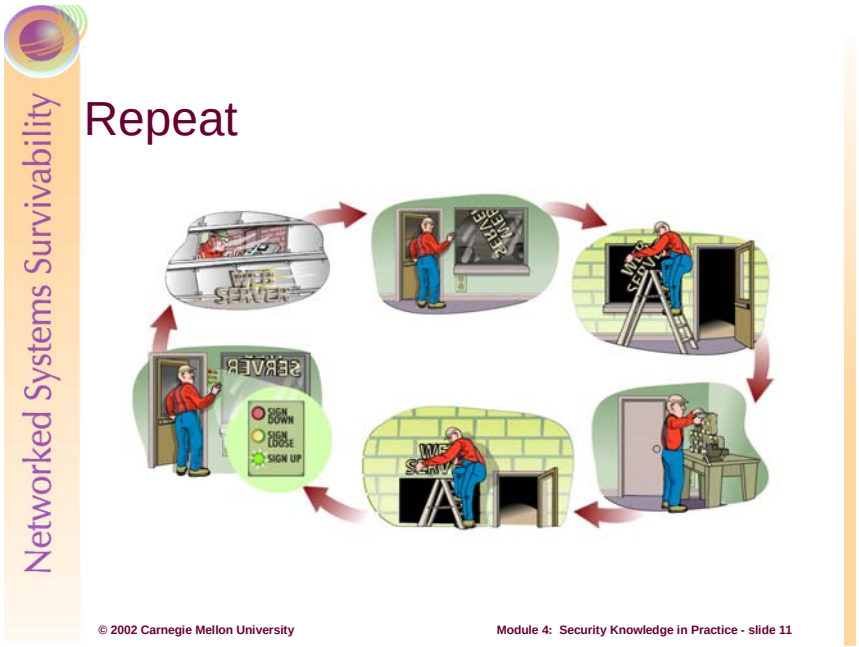
Our master crafter has Responded to the bolts breaking on the Web Server by fixing the wood and selecting and installing higher weight-rated bolts and installing a sensor that tells him about the state of the sign. He then returned the Web Server to production – hanging it on the post outside of his shop – after X-raying it again. The Improve step consists of changing the practices and procedures that surround his Web Server. For example, he could look more often at the server to see if it was still attached, or he could hire somebody to do this for him.

Back in the realm of computer systems, improvement actions typically occur following Detect or Respond. Improvement actions may include:

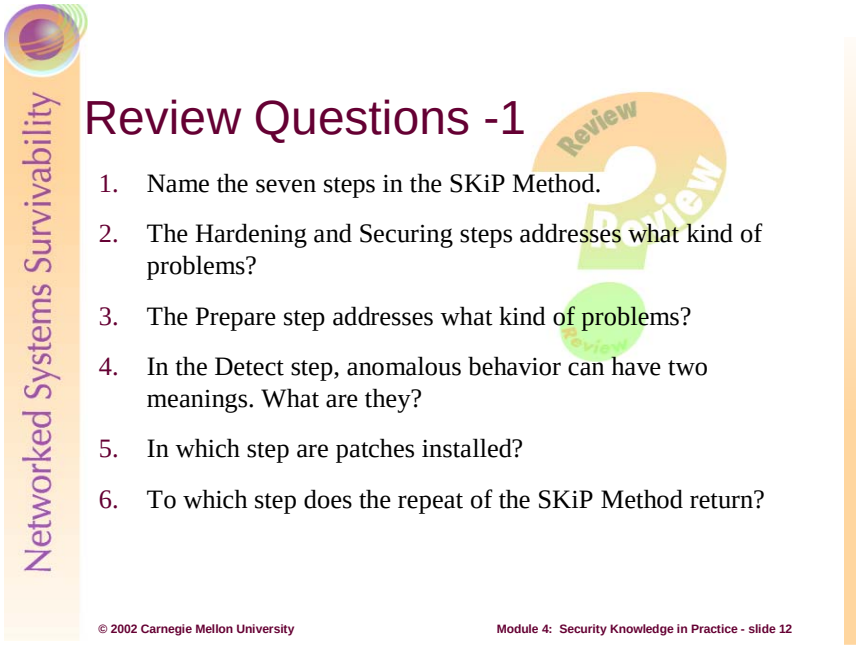
- Holding a post-mortem review meeting to discuss lessons learned. During Prepare, many decisions and role assignments are made. This is the time when those decisions and assignments should be reviewed and updated where appropriate.
- Updating policies and procedures. An organization typically has many policies and procedures dedicated to securing information assets. During Improve, these policies and procedures should be reviewed and updated based on what has been learned in handling the intrusion.
- Updating tool configurations and selecting new tools. An administrator uses a collection of tools in various configurations to respond to an intrusion. Were they the most effective or should they be replaced by a newer or different tool? Now is the time to review the tools used to respond and acquire new tools as needed.
- Collecting business case measures including the resources required when dealing with the intrusion and impacts resulting from the intrusion such as loss of user productivity. It is important to quantify the cost of the intrusion just experienced, so as to effectively reallocate resources and to 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 installing and executing any new tools in a test environment before deploying them.

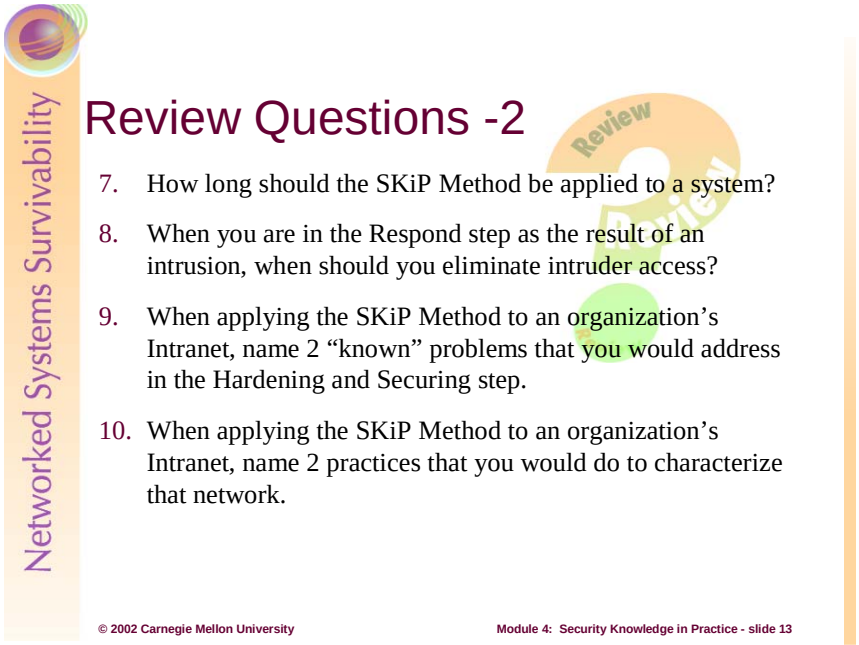
The CERT Guide to System and Network Security Practices book provides a detailed description of the practices necessary to improve a system after detecting (Section 6.9) and responding (Section 7.8) to an intrusion.



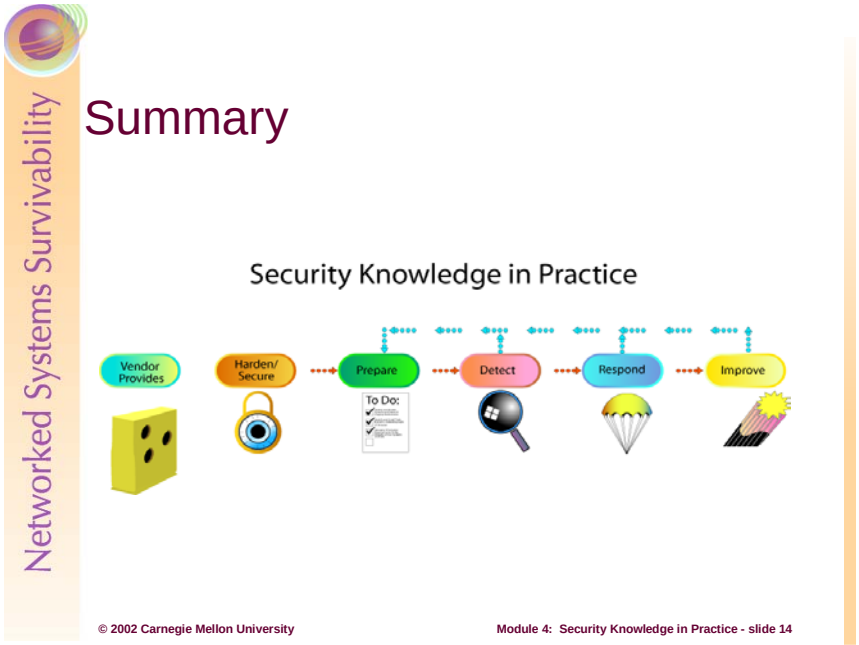
Any changes made during Detect, Respond, and Improve are factored into the system's baseline characterization by iterating back to Prepare. Note that this iteration is different from the initial one because the practices are performed when the system is operating in a production mode.



1. The seven steps are:
    1. Vendor ProvidesHarden and Secure
    3. PrepareDetect
    5. Respond
    6. Improve
    7. Repeat
2. Harden and Secure addresses known problems.
3. Prepare addresses unknown problems.The two types of anomalous behavior are
    a. Previously unrecognized but acceptable behavior.
    b. Unacceptable behavior representing an anomaly and likely an intrusion.
5. Patches are initially installed during Harden/Secure and subsequently during Respond.The SKiP Method returns to the Prepare state.



7. The SKiP Method should be applied until the system is retired.
8. Eliminate intruder access in accordance with your incident response policies and procedures. A key tradeoff is allowing the intruder to remain on your systems long enough to know how they gained access but short enough to minimize any damage (current and in the future).
9. There are many known problems, and here are two:
    a. All Intranet hosts can see all Intranet traffic sent by any host due to being on the same "blue" Ethernet cable. These days, networks are not built with a single blue Ethernet cable. They are built switches that effectively filter the traffic a host can view. This Harden/Secure action provides access controls at the network level.
    b. All Intranet hosts can see all Internet traffic. Currently, there is only a router that moves packets between the Internet and all Intranet hosts. The router does no filtering. An appropriate Harden/Secure action is to add a firewall, providing greater access control at the network level.
10. There are many aspects of a network to characterize, and here are two:
    a. Build an expected connectivity table that contains:
        1. Source host
        2. Destination host
        3. Protocol used
        4. Start time
        5. Duration of connection
        6. Quantity of data transferred
    b. Map the Intranet, that is find out all hosts on this network.



SKiP is a method for initially securing and subsequently sustaining the security state of an information asset. Example assets are:

- Systems running mission critical applicationsNettwork infrastructure including routers, hubs, switches, etc.Subsystems or sub-networks such as those providing email services, web content production and delivery services, perimeter protection services, etc.
- A network architecture and topology
- Sensitive or proprietary information such as customer data or financial projections
- Computer systems installed at home

The steps are:

- *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 that the administrator will be able to spot anomalies that may indicate the occurrence of unknown problems.
- *Detect* those anomalies and other changes, such as the release of a patch, related to the system.
- *Respond* to them when they occur.*Improve* the practices and procedures after fixing the systems.
- *Repeat* the process as long as the organization needs to protect the information assets on the system and the system itself. The SKiP Method should be applied until the system is retired.

