# Global Information Assurance Certification Paper

Dale Wutz
December 27, 2001

**Application of the Survivable Network Analysis Method to Secure My Office System**

**Abstract**

I will present the results of applying the Survivable Network Analysis method to my office system. A brief overview of the method will be presented followed by a detailed description of the method. The method consists of four basic steps which will be explained. I will show how to implement these steps and the results obtained in the application of this method to securing my system. The final results of the analysis will be presented which show that this method can produce a survivable Windows 98 machine, Sun machine and a disk array.

**Introduction**

I wanted to perform a risk assessment on my office environment. I also wanted to plan and implement new features to my existing configuration. The Survivable Network Analysis (SNA) method was chosen. This method was used because it is capable of doing both assessments. I also chose this method because security is an integral part of each step. It will become clear that this is a concept paper because I do not have the resources for a full analysis but the key concepts will be covered and the results used in the assessment. I will present a brief background of the method, an overview of the key concepts and the application to my system.

**Background**

SNA is an engineering process. The methodologies developed for software engineering were modified for use in system planning and implementation with security assessment included in all phases (1). Software engineering now includes security issues as integral parts of the process (2). The SNA method is a holistic method with security as an integral part from the initial planning stage to the implementation stage. The method builds on the normal security assessment, which is considered too focused, one threat and one defense, by taking the approach of analyzing all threats and all defenses necessary to produce a survivable system. The method starts with a high level view of the organizational requirements of the system to be analyzed. All of the components, processes and architecture are analyzed and combined to find the weakest link(s). Whatever failure points are found that affect the organization's operation are analyzed and steps to produce a survivable system are presented. It can also provide a roadmap for future upgrades. It is an evolving process. The starting point is the fact that your system will be attacked. The following describes the process used to build a survivable system.

**Definition**

The definition of survivability is the capability of a system to fulfill its mission, in a timely manner, in the presence of attacks, failures or accidents (3). The term system is used in a broad sense. It can be a single machine all the way up to a large network. Systems are also categorized as unbounded or bounded. An unbounded system is characterized as one that has no central administration, an unknown topology and unknown components. The classic example is the Internet. A bounded system is one that has a central administrator, a known topology and known components. An example would be a corporate intranet. Unless there is a physical vulnerability, only unbounded portion of the system is analyzed. The mission is a high-level definition of what this system will deliver for the organization. The important point is that the mission fulfillment must survive, not individual components of the system. The time component is usually in the mission statement. It is put in the definition for added visibility. Attacks are usually external events that are initiated by an intelligent source, such as probes and intrusions. Failures are normally internal events like disk failures. Accidents are a broad range of events that are random in nature, such as storms. The three words were chosen to encompass all events and not to imply any categorization since sometimes you do not know how to determine what the event was until after the event has occurred. It is the response to the event that is important.

**Properties of Survivable Systems**

A survivable system is characterized as a system capable of delivering essential services in the face of attack, failure or accident. These essential services must be identified. For this to be a survivable system, four key properties must be exhibited. They are resistance, recognition, recovery (the three Rs) and adaptation to attacks. Resistance to attacks is the ability to defend against intrusion. Recognition is the ability to detect attacks and determine the damage done. Recovery is the ability to resume normal functions after the attack. Adaptation is the ability to reduce the effectiveness of future attacks based on the knowledge gained from analyzing a past attack. This last property is not part of the analysis at this time. It is a future goal of SNA.

**The Method**

The method was developed by the Carnegie Mellon Software Engineering Institute CERT Coordination Center (4). This is an engineering process that delivers an assessment of the survivability of current systems, proposed systems and modifications of existing systems. This is a four-step process. Step1, mission objectives and usage requirements for the system are examined and the architecture is determined. Step2, based on the mission objectives and failure consequences, the essential services (those services which must be survivable) and essential assets (those assets that must be maintained during an attack) are identified. Then usage scenarios are determined for the above based on how the business functions. The above are then combined and associated with the architecture of the system to define essential components (ones that must be able to deliver the essential services and protect the essential assets during an attack). Step 3, intrusion scenarios are selected to determine the compromisable components (the ones that can be penetrated).

The final step is to determine the softspot components of the architecture (the essential components that are compromisable). Step 4, the components are analyzed for the three key survivability properties of resistance, recognition and recovery. The deliverable is a Survivability Map, which is a chart associating all attack scenarios with the corresponding softspots to associate the current and recommended architecture strategies for resistance, recognition and recovery.

The above process is carried out by two teams, the company team (CT) and the outside security team (ST). The two teams interact through a series of meetings. The CT delivers the mission statement, business processes and system architecture to the ST. The ST then uses the information to determine the essential components and reports it back to the CT. The ST then does the attack analysis and reports back the compromisable components to the CT. Then the Survivability Map is determined by the ST and given to the CT.

The above process is not necessarily linear. Information can be revised at any joint meeting and the revisions used to update the results of any step. This is called a "spiral process" to point out that overall process can turn back on itself. Any step can be repeated and even at the end, the first step could be done again if new information is presented.

**The Analysis**

Obviously, I can't perform a full-blown SNA since I don't have access to an outside security company. This organization would not pay for it. I can do the analysis by playing both parts and using the framework for the process. Since the system configuration is so simple, the analysis will be done on broad categories. I am going to use two scenarios. The first one is evaluating the current system setup and the other one is upgrading the Sun OS to Solaris 8 and adding some type of routing capability to that machine so that the other machines connected to the switch can have network access. I will follow the model presented in another SNA study (5).

My job functions are in support of the mission of IT in the area of system and software support for the campus. We have a number of centralized UNIX clusters for faculty, staff and student computing needs. I am responsible for documentation regarding access, available software and OS commands. I also answer questions regarding these clusters via e-mail and telephone. I am the system administrator for our group's Sun machines. I also do software and OS evaluations for various flavors of UNIX. This is the reason for the switch. I only have two ports in my office. To evaluate another workstation would mean taking one of my primary machines off the network for a period of time, which is not acceptable. I can simply hook the evaluation machine to the switch, configure the machine with a non-routable IP address and transfer any needed software to the machine using the Ultra 5 as a front-end. This also minimizes network access problems

The current configuration looks like this. The machines are all physically accessible by anyone who can enter my floor. The Windows and Solaris machines are connected to the

campus network. Any other machine in the office is connected to the switch. The Windows machine is used for maintaining documentation and e-mail support for consulting. The Solaris machine is used for software distribution to our group and to any machine on the switch. Any machine connected to the switch is for OS evaluation. I am the only user of the Windows machine and any machine connected to the switch. The Solaris machine has some user accounts, which I maintain.

My office is located in a four story, multiple use building. The lower two floors are a combination of classrooms and computing site. The machines are used by anyone who has a valid userid. The upper two floors are for IT staff. They contain a combination of offices and cubicles. I am located in a cubicle. These floors are accessible to anyone by elevators during normal working hours. All outside access to these floors is restricted to access card holders as is elevator access after working hours. All entrances are monitored by security cameras, which are activated by opening the door without using the access card.

My cubicle contains a number of workstations. I have a pc running Windows 98 SE that is connected to the campus network. This machine is used for normal office operations such as e-mail, word processing and Internet access. It is only on when I am in the office. Another machine is a Sun Ultra 5 with Solaris 7 as the operating system with four external SCSI disk drives. This box is always on except for system maintenance. This box has two Ethernet cards. One is for connecting to the campus network and the other connects to a SMC TigerSwitch 100. This machine has multiple duties. One function is as a software repository for our group, another function is as a development platform and another function is for file transfer to any machine that is connected to the switch. Currently, the only machine connected to the switch is a pc running Linux. The Sun has been hardened using Lance Spitzner's outline (6). There are automated ways to do this (7) but I like to know what is being done. The only suggestion I did not follow was removing rpcbind since I use the CDE interface from the console. I do not allow X applications to run on my machine from external sources. I have also added libsafe (8), which detects and handles buffer overflows.

The proposed configuration is an upgrade to Solaris 8 and configuring the machine so that the machines attached to the switch have network access. The machines on the switch will not be accessible from outside of the Sun.

Step One: System definition

I'll layout the normal usage scenarios for the pc (NUSW), the currently configured Sun (NUSS). For the proposed configuration, they will be for the Sun (NUSP). Machines connected to the switch form a bounded network and will not be considered further.

NUSW1: Access the central UNIX servers. I use ssh to connect to our central servers for system support and compiler use.

NUSW2: Access Internet resources. I use Internet Explorer and Netscape for accessing information and downloading evaluation software.

NUSW3: Create and update documentation. This is for maintaining our printed documentation.

NUSW4: E-mail access. This is to support the consulting responsibilities.

NUSS1: Access to the central UNIX servers. The same as NUSW1.

NUSS2: Access Internet resources. I use Netscape for accessing information and downloading software.

NUSS3: Allow staff access for development and file transfer. This is a responsibility for support of our staff and for file transfers to the machines on the switch.

NUSS4: Develop software for projects. I use the compiling systems for project support. Normally done through console access but occasionally use telnet for connectivity.

NUSP1: Access to central UNIX servers. The same as NUSS1.

NUSP2: Access Internet resources. The same as NUSS2.

NUSP3: Allow staff access for development and file transfer. Same as NUSS3.

NUSP4: Allow Internet access to the machines on the switch. The machines on the switch use this machine to access Internet resources.

The architecture components are identified as:

User interface: resides on the machines as their associated operating systems, for the pc there is only one user while there could be multiple users on the Sun.

E-mail application: resides on the pc.

Netscape application: resides on the pc and the Sun.

Internet Explorer: resides on the pc.

MS Word application: resides on the pc.

Mount points for the disk array: resides on the Sun for accessing the disk array.

Ssh application: resides on the pc and the Sun.

Telnet services: resides on the Sun for staff access.

Step Two: Essential Capability Definition.

Essential services represent the capabilities of the machines that must be maintained during attack. Since there are two separate machines in the system, any redundancy is not an essential service. For the current configuration, NUSW1 is considered an essential service for the consulting support that needs to survive and NUSS3 is also considered essential for support of our group. All other services are either duplicated (internet access) or could be moved to another group very quickly (the central UNIX server access) and not affect our mission. The essential asset is the disk array. Finally, the essential system components are the machines themselves. They must survivable for the essential services and asset to be available.

The essential services, essential asset and essential system components are the same as above for the proposed configuration. This is because the machines behind the switch only have access out and there can be no access from the outside network thus the machines still form a bounded network.

Step Three: Compromise Capability Definition.

Based on the system environment and the sophistication of the intruder, the following intrusion usage scenarios (IUS) were selected. These scenarios are the same for the current configuration and the proposed configuration.

IUS1 (Physical Attack with No Software): The intruder enters the office and sits down at the console of a running machine.

IUS2 (Physical Attack with Software): The intruder enters the office and inserts an infected disk.

IUS3 (Reboot with No System Disk): The intruder enters the office and reboots the machine.

IUS4 (Reboot with System Disk): The intruder reboots the machine with an operating system disk.

IUS5 (Virus Attack): Someone sends an infected e-mail message.

IUS6 (Data Integrity and Recovery Attack): An intruder corrupts major portions of the disk array.

The IUS1, IUS2 and IUS3 attack scenarios assume an unsophisticated attacker. The IUS4

and IUS5 attack scenarios assume a moderately sophisticated attacker. The last attack scenario assumes either a moderately sophisticated attacker or a sophisticated attacker depending on the method used for the attack.

Based on the above attack scenarios we can identify the compromisable components, which in each case is the machine's operating systems.

IUS1: This scenario only affects the pc since the Sun machine has a screen lock.

IUS2: This scenario only affects the pc.

IUS3: This scenario only affects the pc since the Sun machine requires a userid and password to complete the reboot and the assumption is an unsophisticated attacker.

IUS4: This scenario affects both machines.

IUS5: This scenario only affects the pc.

IUS6: This scenario only affects the Sun machine.

Step Four: Survivability Analysis

With this particular system configuration, the identification of compromisable and essential systems found that the machine's operating systems were it. Now we can create the Survivability Map based on analysis of resistance, recognition and recovery. My goal is to protect against a moderately sophisticated attacker. The security team combined IUS1, IUS2, IUS3 and IUS4 in the analysis for the pc and labeled it IUS1-4. Here are the results.

For the pc:

Intrusion Scenario: (IUS1-4) Intruder attempts to gain access to the machine directly

| | | |
|---|---|---|
| Resistance Strategy: | Current – none | |
| | Recommended - | implement BIOS password, screen lock, disable booting from floppy or cdrom and disable auto executing off the cdrom |
| Recognition Strategy: | Current – none | |
| | Recommended - | check file's creation dates periodically and monitor what is currently executing |
| Recovery Strategy: | Current – none | |
| | Recommended - | backup critical files to external media and have clean system disks available |

Intrusion Scenario: (IUS5) Receive infected e-mail

| | | |
|---|---|---|
| Resistance Strategy: | Current – | none |

| | | |
|---|---|---|
| | Recommended - | install virus checking software |
| Recognition Strategy: | Current – | none |
| | Recommended - | keep virus checking software up to date |
| Recovery Strategy: | Current – | none |
| | Recommended - | depends on what the virus does, worst case would be the same as above |

For the current Solaris machine:

Intrusion Scenario: (IUS4) Intruder boots off his cdrom

| | | |
|---|---|---|
| Resistance Strategy: | Current – | none |
| | Recommended - | password protect the EEPROM |
| Recognition Strategy: | Current - | check the system logs and running processes |
| | Recommended - | same |
| Recovery Strategy: | Current - | use clean system disks to reinstall |
| | Recommended - | backup critical software and use clean system disks to reinstall |

Intrusion Scenario: (IUS6) Intruder corrupts major portions of disk array

| | | |
|---|---|---|
| Resistance Strategy: | Current – | none |
| | Recommended - | install intrusion detection software |
| Recognition Strategy: | Current - | check the filesystems |
| | Recommended - | same |
| Recovery Strategy: | Current - | none |
| | Recommended - | backup the disks |

For the proposed Solaris configuration, all of the above apply after hardening the OS a firewall will be installed.

**The Implementation**

I'll start with the pc. I will implement all of the recommendations. I have been searching for a screen lock program. I have found a few and will evaluate them. I rearranged the folders to put the documentation I support in them. These folders will be regularly copied to an external zip drive for storage. The rest of the system is pretty static and can be restored from the Windows 98 distribution disk. I found instructions (9**)** on how to disable the CD-ROM Autorun feature and turn off file sharing. I followed the suggestions found on a web page (10) on tweaking the startup files. I disabled booting from either the floppy drive or the CD-ROM drive. Finally, I set a BIOS password. This last step could be circumvented by a sophisticated attacker via backdoor methods (11), but I'll live with it since my attack scenario is against a moderately sophisticated attacker.

I will concentrate on the survivability of the disk array. I'll be upgrading to Solaris 8 soon so I will live with the system configuration foe now and put my efforts into hardening the

new OS. After carefully studying the disk usage and layout, I came up with a plan. These disks contain the GNU development software, Sun compiling systems, other software for distribution and software under development. I only need to use two of them. The mount point for the GNU distribution is /usr/local (one disk), the mount point for the Sun compiling systems is /opt (another disk), the software for distribution is mounted under /disk3 (another disk) and the development software is mounted under /disk4 (the last disk). I cleaned up the software distribution disk and the development software disk. Using tar and gzip I packed up the software on all the disks and moved the files to /disk4. I erased /disk3. Then I unmounted /disk3 and /disk4, removed their entries from /etc/vsftab and removed them from the case. I now have the ability to quickly restore in case of disk corruption and I have a spare in case of disk failure. I have a clean OS CD-ROM. The last thing I did was to set the EEPROM password and security level to command, which doesn't allow any parameters to be passed to commands.

I am working on the upgrade plan. I went to see what Lance Spitzner (12) had on hardening Solaris 8 and found that Sun's site (13) contains all the information I will need. I'll be evaluating using OpenSSH server (our current licensed version of ssh only allows 2 concurrent users on a workstation, a limitation I couldn't work with), the SunScreen firewall and their security toolkit JASS.

**Conclusions**

I found applying SNA to hardening my office a demanding process. But, it was well worth the insight gained in security issues. I like the fact that the method combines individual tools to make the system survivable. I never considered securing Windows 98 because I didn't think it was possible, I was wrong. The method also helped me plan for the Solaris 8 upgrade. The big success was making my disk farm survivable. I look forward to expert systems development that will make the deliverable the "four R's" with the inclusion of adaptation.

**References**

1. Ellison, R. J., et al., "Survivable Network Systems: An Emerging Discipline", Technical Report CMU/SEI-97-TR-013 (Nov. 1997).
http://www.sei.cmu.edu/pub/documents/97.reports/pdf/97tr013.pdf

2. Viega, J. and McGraw, G., "Building Secure Software: How to Avoid Security Problems the Right Way", Addison-Wesley (2002).

3. Mead, N, R., et al., "Survivable Network Analysis Method", Technical Report CMU/SEI-2000-TR-013 (Sep. 2000).
http://www.sei.cmu.edu/pub/documents/00.reports/pdf/00tr013.pdf

4. Testimony of Richard Pethia before the U. S. Senate Judiciary Committee (May 25, 2000).

http://www.senate.gov/~judiciary/525200rp.htm

5. Ellison, R. J., et al., "A Case Study in Survivable Network System Analysis", Technical Report CMU/SEI-98-TR-014 (Sep. 1998).
http://www.sei.cmu.edu/publications/pub/documents/98.reports/98tr014.pdf

6. Spitzner, L., "Armoring Solaris" (Oct. 22, 2000).
http://www.enteract.com/~lspitz/armoring.html

7. Brumly, D. J., "Solaris Security Recommendations from SANS Step by Step Guide, Titan, and YASSP" (OCT. 2000).
http://www.theorygroup.com/Theory/matrix.pdf

8. "libsafe – Detect and handle buffer overflow attacks" (July 16, 2001).
http://www.gnu.org/directory/libsafe.html

9. McClure, S., Scambray, J. and Kurtz, G., "Hacking Exposed: Network Security Secrets and Solutions", Osborne/McGraw-Hill (1999).

10. "Securing your PC" (August 19, 2001).
http://www.lockdown.co.uk/security/your_pc.php

11. "BIOS Password and Locked Hard Disk Recovery" (2001).
http://www.pwcrack.com/BIOS/bios.html

12. Spitzner, L., "Armoring Solaris II" (November 5, 2001).
http:www.enteract.com/~lspitz/armouring2.html

13. "Security: Sun BluePrints Program and Sun BluePrints OnLine Magazine" (2002).
http://www.sun.com/security/blueprints/