Developing an Academic Security Laboratory

Alec Yasinsac, Jennifer Frazier and Marion Bogdanov

Abstract– Information Security college-level education efforts received a financial shot in the arm late last year with the announcement of a federal funding program to train an information security workforce. In this paper, we address issues surrounding development of a viable Computer Science, Information Security laboratory that meets the three-pronged needs of research, education and outreach in a research university setting. We show how configuration of the computers can be controlled in the shared laboratory environment and discuss the software resources necessary to support the laboratory goals.

***Index Terms*- Security, NSTISSC, INFOSEC Training and Education, Security Curriculum, E-commerce, Security Laboratory.**

I. INTRODUCTION

There has been significant growth in the emphasis on educating professionals in Information Assurance in the past three years. This is in natural response to the growth of the Internet and the corresponding increase in associated risk, as well as a direct result of targeted funding from the National Science Foundation and the Department of Defense for such training. With this growth comes the need for laboratories to practice and refine Information Assurance technology and research.

Last year, we laid out a philosophy for building a academic security program in the Computer Science Department of a research university [1]. In this paper, we identify some requirements for a security laboratory in supporting such a program and detail how those requirements are being met in a state of the art laboratory at Florida State University.

A. BACKGROUND

In 1999, the Computer Science Department of Florida State University decided to expand in the area of trusted systems. Three professors specializing in information security were hired and a curriculum in Information Security was developed. These courses were certified by the National Security Telecommunications and Information Systems Security (NISTISSC) curriculum for Information Security Professionals and FSU became an institution in which certification for information security can be earned. Subsequently, the National Security Agency (NSA) designated FSU as a Center of Academic Excellence in Information Security Education.

Recognizing the need for hands-on work in Information Assurance, the FSU College of Arts and Sciences invested space and an initial purchase of equipment to start a security laboratory in the computer science department. This laboratory is the FSU Security and Assurance in Information Technology (SAIT) Laboratory.

The mission of SAIT Laboratory is to support varied demands of research, education, and outreach in information security. More specifically, the laboratory must be able to be used by professors for conducting research activities; individual student thesis and project work, class projects on information security topics, and to support faculty and students partnering with industry on issues of practical security. The next three sections describe how we enhanced SAIT Laboratory support in these three areas.

B. RESEARCH

Two essential elements for conducting many aspects of information security research are equipment and space. These two requirements are met at FSU in SAIT Laboratory. SAIT laboratory is a dedicated facility where practical security techniques can be exercised and experiments conducted in order to test theoretical ideas. There are already a wide variety of research projects ongoing in SAIT Laboratory.

One example is a current research project, supported by the US Army, that is part of an eleven-investigator team formed to study problems of critical infrastructure protection. One of the teams is comprised of two FSU professors who are developing a case-based reasoning (CBR) system for network intrusion detection. Currently, a CBR system that replicates the functionality of an Intrusion Detection System (IDS) is being devised. In addition, add-on modules that interpret and summarize the IDS output data are being constructed. Once this is accomplished, the object will be to seek ways to build-in higher levels of intelligence, enabling more sophisticated and accurate alerting capabilities.

Another SAIT Laboratory project has a goal to extend the network monitoring and IDS described in [6]. Extensive research, programming and implementing the system has been conducted. Specifically, a knowledge-based intrusion detection system that identifies intrusions into security protocols has being implemented. That system is presently being extended to include a behavior-based capability that can detect even previously unknown attacks on executing protocols. It detects malicious and questionable activity by tracking and analyzing user and network behavior with respect to security protocols.

With more existing projects in Information Hiding, Tracing and Watermarking, Intrusion Detection, Key Distribution, Key Escrow, Security Protocols, Survivable

Computation, and Threshold Cryptography, the laboratory is on sound research footing. As the examples above demonstrate, the SAIT laboratory is an ideal place for the final product to be tested. The researchers will not have to worry about contaminating the configuration of the computers in such a laboratory because it is an environment that maintains the configuration of the original installation; one of the goals is to provide a facility in which security research projects can be safely conducted and advancement in the field of information security is enhanced.

These research projects use the SAIT laboratory as a testing environment. A goal of this project is to allow research projects to be installed, analyzed and documented without having to put the Computer Science Department network in danger. Also, this process, will not violate network and system administration policies that condemn modification of baseline computers.

C. EDUCATION

The educational functions of SAIT Laboratory are focused on supporting undergraduate and graduate education. The Florida State University information assurance curriculum spans theory and practice. The curriculum provides a strong foundation in security principles, including the mathematical foundations of protection mechanisms and the models that represent the salient security interests.

Additionally, the curriculum in information security in Computer Science at Florida State University satisfy the National Security Telecommunications and Information Systems Security (NSTISSC) training standard for Information Security Professionals. Courses such as Network Security and Network and System Administration are excellent candidates to use SAIT laboratory's capabilities.

A goal of the laboratory is to support class projects. Two challenges to this objective are to (1) Provide an environment with sufficient freedom that students can conduct meaningful experiments, while ensuring that a baseline configuration can be easily regenerated and (2) Providing a sufficiently rich set of security tools to support a wide array of projects.

The first objective is obtained by creating a user environment which will allow users to install and execute security tools. This environment should also protect the system from being harmed by the users from the escalated permissions.

The latter of the above two goals may be met by establishing a library of security tools in SAIT Laboratory, that would be a resource to the instructors and to the students. In the Network Security course, projects can be devised from the tools in the library. For example, a project's goal for a pair of students working as a team, could be to successfully install an IDS. Checking for vulnerabilities on the machine with a scanner tool will follow this. If an array of tools is available in the laboratory, the team can easily practice the techniques that they learn in the classroom and document details of their work.

Another application for the library of tools is the Network and System Administration course, a course designed with hands-on experience in mind. This course offers a hands-on component where teams of two or three people are formed. Each team has to install and maintain three computers with three different operating systems. Part of the course curriculum is to focus on security of the computer and the network A practical attack and defend session is run where students must protect their own machines while attempting to intrude the machines of others.

With information security on the mind of everyone connected to the Internet, an educational institution that provides services that will educate individuals, government and corporations about information security is essential. The skills learned will provide better information security practices. Also, the education will create leaders who are capable of serving our governments and businesses for betterment of the nation and the economy. From the above examples a conclusion can be drawn that the SAIT laboratory type environment is an ideal facility for serving education to the public interested in applied research in information security technology.

D. OUTREACH

Along with the capabilities as a research and educational facility, the SAIT laboratory environment must also have the capabilities to support an outreach program, which will generate benefits not only for the SAIT laboratory, but also for the corporate sponsors. Creating and maintaining a mutually beneficial relationship with corporate sponsors will enhance researcher access to modern equipment and professional resources in tune with marketplace needs. However, this laboratory is not complete without a mechanism for creating a baseline of laboratory computer configurations.

The SAIT Laboratory outreach program fosters communication and promotes mutually beneficial relationships among members of the government, industry, and academic communities. Academic communities can specially benefit from collaboration with industry or government sponsors. Sponsors leverage the SAIT Laboratory resources, experiences, and relationships with only a minimum investment. By participating, the sponsors gain access to the research and education laboratory. Participation provides partner access to security experts, such as the professors or their research assistants, interaction with other center sponsors and early access to research findings. Sponsor partnerships enhance researcher access to modern equipment and professional resources in tune with marketplace needs.

An example where the current state of the SAIT laboratory can provide service is a corporate sponsor who desires to have their network scanned for vulnerabilities. An ideal solution to meet the goal would be to use a library of

tools to perform the tasks. A team of student-experts may be sent as part of a class project or other experiment to perform the vulnerability assessment.

Such outreach agreements walk a fine line in an any public university; quite a more narrow line in a research university where the emphasis is on educating scholars, not on creating trade school technicians. Nonetheless, if the tasks can be efficiently accomplished within the educational framework, there can be significant advantage to the students, the university, and the industrial partner. The tools library is one means of improving the efficiency of such agreements.

E. SAIT Laboratory Equipment

A varied suite of equipment is required to provide the necessary functionality in SAIT laboratory. There must be sufficient end-user workstations to support class projects, along with bandwidth. To be reasonably comprehensive, the laboratory must support a reasonable number of each Unix and Windows-based systems.

Additionally, the laboratory requires dedicated networking capabilities sufficient to develop and benchmark network security concepts.

To support these requirements, the following equipment was purchased for the laboratory:

Description	Processor	Memory	Storage
1 Sun Microsystems Enterprise™ 220R Server	450 Mhz	2048 MB	160 GB
1 Cisco Catalyst 3500 Series XL Switch	80 Mhz	16 MB	32MB
1 Sun Microsystems StorEdge A1000	50 Mhz	32 MB	16MB
9 Sun Microsystems Ultra 5 Workstations with PCI Cards	350 Mhz	128 MB	8 GB
5 Tri-C Systems Custom Build PC	600 Mhz	128 MB	20 GB
4 KVM Switches	---	---	---
1 HP LaserJet 4100N Printer	---	---	---

Figure 1

The physical layout of the devices is conducive to individual and team projects and all network devices are easily accessed. The switch allows management of all switched ports from a single IP address. Accordingly, the lab can be segmented into several networks, which can accommodate PCs serving as routers to other PCs. This environment provides multiple simulations for the mentioned goals of the laboratory. The availability of PCs with Windows 2000 allowed focus on security tools for the Windows operating system.

II. Information Assurance Laboratory Requirements

Effective utilization of SAIT Laboratory resources demands an operating environment and associated tools geared to Information Assurance. One specific requirement is that the environment must facilitate student projects that require access not normally given to students. There are many well-known risks with allowing such access privileges, not the least of which is that the configuration of the computer cannot be effectively controlled if such access is granted.

A security laboratory provides an additional challenge because many tools and techniques that security education and research demand require extended user privilege to install and remove software with powerful capabilities. Many security software tools require administrative permission to carry out the installation process and to use the software. The installation process modifies the configuration of the host computer that may create security vulnerabilities.

Controlling computer configuration is essential to maintaining the integrity of a computer. Once the baseline configuration is compromised the computer might be vulnerable to security attacks. Compromised computers are invalid research tools because the alterations may cause invalid or inaccurate data.

Additionally, introducing new software can have a negative effect on the computers performance and configuration. Once a machine has been setup properly, the ideal maintenance and operational procedure is to restrict non-administrative users from changing the configuration of that machine.

Because all SAIT Lab computers are shared, a baseline configuration must be established and it must be reasonably easy to return any/all machines to this baseline. Thus, the first objective of this project was to provide a baseline mechanism that will maintain the integrity of the original computer configurations for a general-purpose laboratory.

Secondly, any effective laboratory must provide functionally specific resources. The combined space and computing resources that were installed in SAIT Laboratory were functionally equivalent to several other computer laboratories on this campus, and those on many other campuses. The only distinguishing characteristic relative to Information Assurance was that priority access to the resources were granted to faculty and students involved in security courses and projects.

The second objective of this project is to equip SAIT Laboratory with software resources targeted to support Information Assurance. At the heart of this effort was to build a library of tools and materials that facilitate work in Information Security.

Each of these tasks is significant in their own right. Further complicating the effort is the requirement to support at least two operating environments with both a protected

configuration for the workstations, and a complete set of tools and software resources.

III. Protecting Security Laboratory Computer Configuration

Protecting the configuration of end-user computers is not a new concept. While laboratories of shared computers struggle with this problem continually, industry, government, etc. have recognized that baseline configurations must be guarded to prevent overwhelming maintenance costs.

There are three primary techniques for accomplishing the configuration baseline of a workstation:

1. Configuration locking
2. Cloning
3. Configuration blocking

We considered each of these options for their utility in a security laboratory.

A. Configuration Locking

A common technique for protecting shared or public computers is to lock the configuration so that special permission is required to make any non-trivial changes to the computing environment. In a security laboratory, this option is not viable if used in isolation, else the laboratory would be virtually useless. While we heard of schemes and mechanisms for finding middle ground between an open or a locked environment, our search for a suitable locking configuration technology was fruitless.

Configuration locking also removes an important step in the learning process, installation. With this method, security tools have already been installed onto the machine and are available only for use. Also the configurations for the tools would be set which restricts the usage level for users and during experimentation.

Thus, we discounted configuration locking relatively early in our analysis.

B. The Cloning Solution

The idea of cloning is a process where an image of the disk from one [protected] computer is regularly copied to all computers that share the common configuration. Cloning is a common solution to configuration baselining, though there are numerous drawbacks with this concept.

A fundamental flaw with the cloning approach is the time and resources necessary to copy the baseline disk image many times. Communications capacity has not kept pace with disk space growth, so copying a disk image to one computer can take hours. In a laboratory with twenty-five computers connected by a single, shared medium network, the communication bottleneck is prohibitive.

As an additional drawback, network cloning can be unpredictable because many variables are involved in correct function of the network. Network downtime creates an unsuitable risk because of the time-sensitive nature of research and educational projects.

Non-network cloning allows only one machine to copy an image from another machine. This process takes a significant amount time and does not scale, again untenable in our environment. There are other mutations of the cloning approach that we considered, but all were hardware intensive or otherwise cost prohibitive.

C. Configuration Containment

Having decided that the characteristics of cloning and configuration locking were less than optimal for our purposes, we sought a somewhat more sophisticated, software solution. During our analysis, we considered proprietary and non-proprietary software to accomplish the requirements for the SAIT Laboratory environment. The applications that we considered could generally be categorized as either sandbox or wrapper[1] applications.

Wrappers are programs that separate the outer layer client from the lower layer software resources. A user operating outside a wrapper will not be able to tell the difference between the wrapper and the actual lower layer resource, except for the added functionality that the wrapper provides.

A sandbox application, on the other hand, only protects a subset of the lower layer resources. In a sandbox operating system, for example, the user may be granted superuser privileges to the machine that will allow them to install, configure and execute software, but only within a constricting file system that will serve as a playpen, or sandbox. The sandbox keeps them from making changes to the protected part of the operating system by restricting action on the file system.

Wrappers and sandboxes are similar concepts that allow extended user privileges, while protecting important aspects of the workstation. The main difference between a wrapper and a sandbox is that a wrapper appears as a normal computer to the user and will allow all the same functionalities. The sandbox, on the other hand, will only contain the necessary parts of the file system and libraries needed for installing and executing programs.

As we noted earlier, SAIT Laboratory sports workstations in two different operating systems. As one may expect, there are few parallel software solutions in these two environments. In the next two sub-sections, we discuss the approaches we

---

We consider a wrapper application to protect the user from any view other than that of the expected environment, while a sandbox may encumber the user with some application dependent requirements.

took in finding software solutions to control the configurations for workstations for these two operating systems. For Windows systems, we describe VMWare [2], a wrapper, and the Unix architecture to be used, JSS (J's Secure Sandbox), is referred to as a sandbox.

IV. A Windows 2000 Wrapper Solution

The first program we considered was a proprietary product named Deep Freeze. As its name suggests, Deep Freeze attempts to freeze the configuration files [1]. Appropriate users are allowed to make changes to the machine while the software is in freeze mode.

This is an excellent software product that contains sufficient functionality for a security laboratory environment. Unfortunately, Deep Freeze is only available for Windows 95, 98 and Me, which have file allocation table (FAT32) type structure. The laboratory computers are installed with Windows 2000 professional operating system, which supports the new technology file system (NTFS); therefore, Deep Freeze software is incompatible with our requirements.

Our second attempt at finding a software solution led us to the proprietary product, VMWare [2]. VMWare protects the computer configuration, yet allows the user to accomplish tasks that require computer configuration changes. VMWare accomplishes this by establishing a virtualization layer that turns the physical computer into a logical pool of resources. These resources can be allocated to any application or to another operating system.

VMWare is a software wrapper that is installed as a guest operating system that sits logically on top of the host Windows 2000 operating system. The result is an environment that grants administrative privileges on the guest operating system. VMWare users may have permissions to install any software on the machine. When operating within VMWare, users can change the host configuration, install software, and execute privileged processes. VMWare also allows users to build and configure virtual networks between virtual machines. These changes are contained to the VMWare environment, so when the user closes their VMWare session, the configuration changes disappear.

Details of VMWare installation and use are provided in Appendices A and B.

V. Programming a Solution for Solaris

Although VMWare was a suitable solution for Windows, there is no Solaris version of VMWare. Our in depth search was not able to locate any proprietary solution for a wrapper or other suitable product controlling workstation configuration in the Solaris environment. This created the need to write a program to provide the needed capabilities for the Unix environment of SAIT.

As we mentioned earlier, programs written in the JSS architecture create a file system inside of the operating file system so it appears to the user as if they are able to access the actual file system residing on the computer. However, they are logically inside the secondary file system and while inside this jail will have ownership of all the files on what looks like the actual file system, but will not be able to access or damage the actual file system.

A script written in BASH, which resides on the server (startup.sh) will create the default sandbox environment. If any updates take place on the server after the JSS environment has been created, the script only needs to be rerun to update the environment to the most current settings. Updates will normally include any type of software upgrade made to a program or library on the server which is used in the JSS environment. Only the lists of necessary commands, directories and libraries are placed into the sandbox environment.

The main program executable will be placed on each of the client machines. The main program is written in C++. Each of the sandboxes created on client machines will rsync with the server to ensure they have an exact replica sandbox to maintain the same environment on all the clients. The client program will update in the beginning in the execution of the program. This ensures if any updates/changes were made on the server, they will be reflected on the clients before the user is launched into the program environment.

Once the client program has created the environment and set the permissions to the program user, jss_user, it will launch a bash shell for the user to work in. Due to the fact that jss_user is the owner of this sandbox, once the program is closed, no one except root may access the JSS filesystem except for root until the program is restarted by a user. This shell will allow users to install the database of security tools. While inside this environment, the user is not able to access any part of the actual filesystem until they log out of the JSS program.

Once the user types the word "exit" to close their JSS session, the program closes the user shell, resets any local changes made to the host file-system and exits. The security tools will be a part of the default JSS environment, each individually compressed in a tool directory along with any programs or libraries necessary for the security tools.

This secondary file system appears to the user as the original file system but they have the flexibility and virtual super user privileges to install the collected security tools without affecting any part of the main file system on the computer due to being confined in the sandbox.

VI. Security Tools for SAIT Laboratory

The second area that we sought to improve in SAIT Laboratory was to establish a library of security tools for use in research, education, and outreach. Our goal was to find

existing tools with a broad variety of security-related functionality. There are many sources for analysis of tools in trade magazines [4]. Unfortunately, we were not able to find any source for analyzing tools to support security laboratories.

The tools collected for the SAIT laboratory are all non-offensive tools. Users are limited to the tools in the security database during sessions.

A. Categorizing Tools

We began this effort by establishing a categorization of tools that we desired to accumulate. While many organizations have their own, de facto, standard tool categories, our process of creating categories of tools was based on software application, functionality, description and previous research. Many security-oriented software tools overlap in their functionality. As an example, intrusion detection systems, network monitors, and sniffers share some functionality. After evaluating tools from these categories, we decided that these tools could be included in a super-category, somewhat inexactly called Intrusion Detection Systems. Figure 1 shows our tools categorization and summarizes our collection efforts. A glossary of terms describing our categories is provided as Appendix A.

(S)hareware (F)reeware	Windows		UNIX	
	S	F	S	F
Firewalls	15		4	8
Antivirus	12	5	1	
Vulnerability Assessment	6			11
Intrusion Detection	8	4	7	28
Authentication/Encryption			4	3
Miscellaneous	3	5		12
Total:	44	14	16	62

Figure 2

B. Collecting and Testing a Library of Security Tools

With our initial categorization complete, we set out to populate the library of the eight different security categories. Many hours were spent searching the World Wide Web, reviewing trade magazines, and talking with security practitioners looking for tool sources.

The maintenance of these security tool libraries will be passed onto the next maintainer of the software for VMWare and JSS. This will be either the system administrator or the next graduate who expands upon these project ideas. Also, in the information readme files created for each of the tools, the web location is included and informs the student where they may download the newest version of the program.

The only security tools available on the machines are the ones that have been made available in the library provided by the administrator.

One of the hot issues in computing is licensing issues in software. Software vendors claim that piracy costs them millions of dollars per year. Copywrite considerations were given high priority in our project and all property rights of software that we analyzed and stored in our library were carefully protected[2]. Many of the tools that we acquired grant full use privileges for personal and educational purposes. Others only allowed downloading for evaluation purposes. In case of the former, we provide documentation and source code for students and researchers that may need the tool for use in the laboratory. In the latter case, we identify the product that we evaluated along with the evaluation and reference information in case the product is needed at a later time.

The proprietary software that we include contains only a brief evaluation of the product, its capabilities and requirements. Again, this information may be helpful for outreach projects where a specific functionality is needed in a limited period of time.

B.1 Freeware

Freeware allows you to acquire software freely either through existing libraries at other universities centers and other hosting sites such as the "Computer Incident Advisory Capability, U.S. Department of Energy (http://ciac.llnl.gov/ciac/). We also found industrial web sites for freeware tools where we accumulated significant software for evaluation.

Freeware is very attractive in the academic environments. It eliminates concerns about copy write and licensing issues for the software. Effectively, the software can be used by anyone for almost any purpose.

Freeware can either be closed or open source. Closed source programs do not give the user access to any of the underlying code while open source makes it sources open to the public. All the UNIX security tools researched for this product are open source freeware tools while the Windows side has closed source freeware tools.

There are several drawbacks when using freeware. Firstly, there is no guarantee that the program will be able to execute correctly on your system. In these cases there is usually little or no user support available and a lack of documented quality control. Also, the contents of the program (unless the code is carefully examined) could contain malicious code or may be infested with security holes that render the user system vulnerable to malicious activities. Although this is normally not the case with freeware, it is a situation that can occur and therefore shows the importance

[2] It is for copywrite reasons that we do not list tool names in this paper.

of obtaining security programs from trusted companies and web sites.

Many times open source freeware tools require are not standalone, but require installation of additional programs and libraries to execute correctly.

Although source code adjustments are usually necessary to get programs to execute in an environment, this is also one of the best features of open source freeware. This allows users to alter code to fit their specific needs and to fix arising problems running the software on their system. Also, freeware eliminates concern dealing with copy write and licensing issues such as when using their product for SAIT laboratory

Benefits to freeware include that these programs may release updates more frequently and be more responsive to releasing bug fixes for new arising security holes while proprietary software vendors may only release updates to their software at set intervals throughout the year. Also, when user support is available for a freeware product, direct contact with the original software creator or maintainer is normally possible.

Having freeware acquired from reputable sources clearly adds to the quality content of our tools library.

B.2 Shareware

Shareware programs normally request something in return for the privilege of using the software. This can range from anything to filling out agreement forms or paying a small cash sum. More commonly, shareware is normally a demonstration version that offers only a portion of the full package functionality and that lasts only for a limited amount of time.

This limited functionality is a significant drawback to having shareware in our library. Our reasoning in including these products is that it may be that a user of the lab needs the functionality that only a certain product can provide. If we have a shareware copy in the library, the user is more likely to find the tool and can save time in acquiring proper licensing.

B.3 Proprietary Software

Proprietary software is software that must be purchased through a vendor. Not all proprietary software is as expensive as the prominent desktop tools that dominate the shelves of software retailers. However, you are ensured technical support, regular updates and fixes, and full functionality of the security tool.

VII. Conclusion

In this paper, we laid out the two requirements for a security laboratory of establishing a protected workstation configuration and a library of security tools. We describe separate solutions for configuration containment for Solaris and Windows environments, distinguished by their adherence to the sandbox paradigm for our Solaris machines and the wrapper paradigm for Windows workstations. We further describe the need for a library of tools for our laboratory and describe the process and results of our search for tools and our population of the library. It is our contention that these actions increase the utility and efficiency of SAIT Laboratory.

References

[1] "Information Security Curricula in Computer Science Departments: Theory and Practice", Alec Yasinsac, NCISSE 2001

[2] "Benefits of Using Deep Freeze", Copyright 2001 Faronics Technologies, Inc. and/or Hyper Technology Dec 2001, http://www.deepfreezeusa.com/benefits.htm#2

[3] "VMWare", Hallogram Publishing, Aurora, CO, site verified January 2002, http://www.hallogram.com/vmware/?source=goto

[4] "Index of 2000 Information Security Product Reviews", Information Security, December 2000, Volume 3, Number 12, pp 46-48

[5] "Detecting Intrusions in Security Protocols", Alec Yasinsac, Proceedings of the First Workshop on Intrusion Detection, held in conjunction with *7th ACM Conference on Computer and Communication Security*, Athens Greece, Nov 1-4 2000.

Appendix A
Tools Glossary

Firewalls

Any set of related programs that examines packets and decides whether to forward it towards its destination.

Intrusion Detection Systems.

Real time monitor system that compares ongoing activity in the target domain to signatures or profiles to detect malicious or abnormal activity.

Network Based

Intrusion detection program that monitors an entire network.

Host Based

Intrusion detection program that monitors a single host.

Sniffers

Packet Capturing Program.

Auditing Tools

Constant system monitor for sundry services (log files, resources).

Hybrid Tools

Intrusion detection program that has functionalities of host and network intrusion detection system such as monitoring an entire network or single computer.

Vulnerability Management Tools

Lockdown

Defines access controls for resources on a machine to help reduce vulnerability on a machine.

Password Assessment

Checks for weak (easy to guess) passwords.

Port Scanners

Scans the ports of a machine to see what services are running.

Security Assessment

Scans a machine or multiple machines to detect known vulnerabilities.

Anti-virus Tools

Searches media for potential or known viruses. Due to most viruses dealing with Windows, anti-virus tools are mainly used for UNIX in interoperability mechanisms.

Auditing Tools

System monitor, but does not audit the system in real-time. These Anti-virus/Auditing tools do not operate in real-time (are not constant system monitors).

Miscellaneous

This category includes miscellaneous tools which do not fall under any of the other categories, mainly front end tools for the given security tool categories.