An Isolated Network for Research

Matt Bishop, L. Todd Heberlein
Department of Computer Science
University of California at Davis
Davis, CA 95616-8562

Abstract. An isolated network is critical to the successful analysis of vulnerabilities and attack tools. Maintaining such a network introduces issues of policy and implementation which conflict with the need to transport data from the Internet to the network. This paper describes the goals of one isolated network, the policy and implementation that satisfies those goals, and other considerations to protect the confidentiality of data and programs on the isolated network.

Keywords. Isolated network, vulnerability, attack tools, design, implementation

1. Introduction

Vulnerabilities research requires the analysis of known security problems, and the development of techniques and technology to find previously unknown security problems. Exercising vulnerabilities helps researchers understand how and why those vulnerabilities occur, what interrelationships with system components exist, and the effect of proposed patches. Ideally, one could perform this analysis from a description of the vulnerability; in practice, the complexity of modern systems makes such analysis merely a starting point.

Amplifying this is the observation that many vulnerabilities come to the attention of the research community when attackers exploit those holes. This phenomenon may have many explanations. That attackers exploit vulnerabilities provides a fruitful source of information about vulnerabilities, for attackers often leave behind attack tools (binaries or scripts) automating these exploitations. Executing these scripts, and analyzing their behavior, often makes determining which vulnerabilities the scripts exploit much easier.

Research in the area of computer and network vulnerabilities entails handling information which provides detail about specific methods to compromise the security of computer systems of a specific type. Worse, some data specifies sites, hosts, and even IP addresses on which the compromise occurred. This data is therefore considered sensitive, and must not be divulged to unauthorized users or made accessible to the Internet.

This paper discusses the model used to protect the information on an isolated network, how the model and network are implemented, and plans for extending the model. Section 2 presents the model and the reason for its selection. Section 3 discusses the implementation of the model in the setup and maintenance procedures for the isolated network, and other principles used to assure a reasonable level of security. Section 4 explains how data is moved to and from the isolated network, as such motion contradicts the notion of "isolation" and is a potential point of vulnerability. Section 5 explains the larger goals of the isolated network, and how its configuration serves to advance those goals.

2. Model

The isolated network (called "isonet") currently consists of several hosts running different versions of the UNIX operating system (SunOS, Solaris, IRIX, and HP/UX). This constrained our choice of model, because the model had to be simple enough to be implemented using native UNIX protection mechanisms and yet powerful enough to provide adequate security.

"Adequate security" is a function of the isonet's requirements, which are driven by the four types of data and programs that the isonet stores. The vulnerabilities database contains descriptions of the vulnerabilities, including system types on which they were found, environmental conditions needed to exploit them, and at least one attack script demonstrating how to exploit the problem. Attack tools recovered from many sources often show previously unknown vulnerabilities, and provide a basis for studying techniques for attack script analysis. Other researchers use the isonet as a testbed for security-related tools and protocols, such as comparing the effectiveness of different intrusion detection systems. Finally, the testbed provides a development environment for tools deemed too sensitive to be placed on the Internet, such as network connection altering tools.

The isonet must protect the confidentiality of data and programs stored on the isonet (called "isoinfo") by providing the following:

1. a mechanism to keep the isoinfo inaccessible to users on the Internet; and
2. a mechanism to ensure authorized users can access only the type of information they are authorized to access.

These requirements immediately suggest a multi-level security model. That the systems are UNIX-based suggests one security level for the isonet. The hosts on the isonet all run at the High level and the Internet is considered Low; this division is enforced by physical means. The four categories are proprietary programs (Prop), development tools (Dev), vulnerabilities data, and attack tools; but as attack tools are part of the vulnerabilities studies, those two categories are merged (Vuln).

The Bell-LaPadula model [1] allows subjects in compartments to write to objects in the same compartment. This presents a problem: when attack tools are executed and vulnerability exploits are recreated, the exploitation could alter information under study or programs under development. So the model must be modified

subjects	objects			
	Low	(High, Vuln)	(High, Prop)	(High, Dev)
Low	rw			
(High, Vuln)		r		
(High, Prop)			r	
(High, Dev)				r
(High, AVuln)		rw		
(High, AProp)			rw	
(High, ADev)				rw

Figure 1. Access Matrix for the Isonet Model.

to provide:

3. a mechanism to ensure that the execution of a process does not affect isoinfo unless it is authorized to do so.

Lipner [2] showed how to extend the traditional MLS model to provide such a mechanism. Three new categories, AVuln, AProp, and ADev, allow member subjects to alter data or programs in the categories Vuln, Prop, and Dev, respectively. No objects reside in the new categories. The resulting access matrix is shown in Figure 1.

We show this configuration meets the above requirements.

Consider requirement 1. If the isonet is at level *High*, and the Internet at level *Low*, this bars writing between the isonet and the Internet as no subject at level High can write to an object at level Low.

Now consider requirement 2. Isonet users (subjects) are assigned to a set of categories corresponding to their needs. Subjects with a particular set of categories can only read objects within those categories to which they are assigned. Further, they can only alter information if they are in the appropriate "A" category; this restricts the ability of a user to damage data or programs when experimenting.

Clearly the security levels form a linear hierarchy. A subject in category ADev can read or write an object in Dev, but a subject in Dev can only read an object in Dev. This induces a relationship based on the number of rights a subject has over an object; defining the relationship in the obvious way, Dev < ADev, Prop < AProp, and Vuln < AVuln. This gives a lattice model of security, and meets requirement 3.

3. Implementation

Our implementation combines both procedural and technical mechanisms to achieve a level of security that prevents the accidental release of, and damage to,

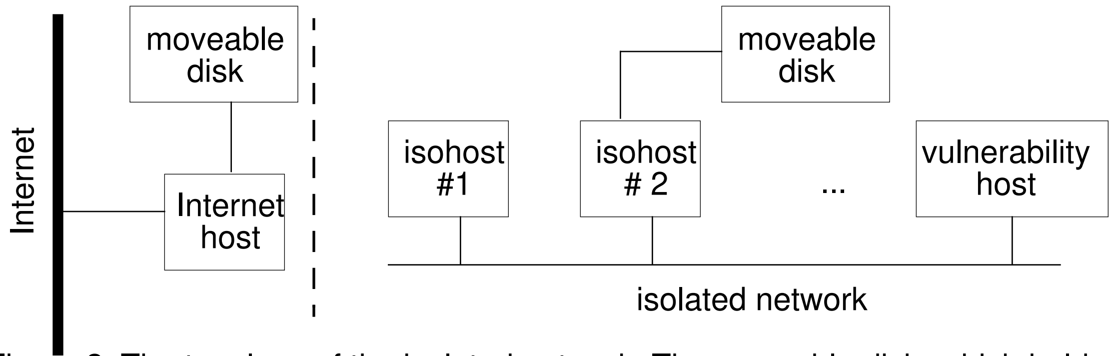

Figure 2. The topology of the isolated network. The moveable disk, which holds only data, is physically moved from the Internet host to an isohost to upload files, and from the isohost to the Internet host to download files.

the isoinfo, as well as to prevent external attacks. As is any site, the isonet is vulnerable to insider attacks; however, we have taken some steps to limit their damage and ensure they are quickly detected. This results from the systems not implementing multi-level security, and hence the isonet can only emulate the model, not implement it fully.

3.1 Configuration

Figure 2 shows the isonet topology. As the isonet grows, and the topology becomes more complex, routers, bridges, and other network infrastructure components will be added. However, the part to the left of the dashed line will remain unchanged. The configuration to the left is the interface to the Internet; the part to the right is the fully isolated net. The moveable disk is used to move data from the Internet to the isolated network, and vice versa, and will be discussed later.

The hosts labelled *isohost* are the hosts on the isonet. The isohosts contain the user home directories and non-sensitive programs (development tools that could be made available to the Internet, such as compilers, mailers, word processors, and so forth). The isohosts trust one another, and may use DNS and NIS or static host and user information tables. If the former, the isohosts are configured so that only the isonet DNS and NIS servers will be queried; if they fail, static information is used. All isohosts run remote login and file transfer servers; they may run other servers as well. Of course, these servers are inaccessible to the Internet.

One of the isohosts, the *vulnerability host* , stores vulnerability and attack information, as well as tools and programs deemed sensitive (such as proprietary programs or vulnerability detection or exploitation tools). If a user wishes to add a vulnerability or alter a program, that user must do so from this host. The vulnerability host is *never* used for experiments, and may be taken off the isolated network during experiments (should the experimenter deem it necessary).

3.2 Implementation of the Model

The separation between security levels is enforced physically; the isonet is disconnected from the Internet. This disallows High subjects from accessing Low objects, and vice versa, as the model requires.

Implementing categories is a bifurcated procedure. Users in categories Vuln or AVuln are in the group *vuln*; categories Dev and ADev correspond to *dev*, and Prop and AProp to *prop*. For example, user *bishop* is a member of the groups *vuln* and *dev*, and so can read source code for tools under development, and data in the vulnerabilities database; but this user cannot access proprietary programs.

The programs and data which this scheme protects reside on a disk exported from the vulnerability host. Each file on this disk is in one of those three classes, and all are group readable and writable; however, the disk is exported as read-only. So remote clients cannot alter the files, despite those files being group writable. The programs and data are kept unreadable and unwriteable by all other users (except the owner) to prevent any unauthorized access.

Subject membership in the categories AVuln, ADev, and AProp require that the user have an account on the vulnerability host. Then the users can alter the files, as the disk is local to the vulnerability host and is readable and writable there.

One advantage of these procedures is that no locally-developed software is required. Exporting is controlled using vendor-implemented protocols, currently the Network File System [4], and each vendor supplies its own account management facility.

While procedures clearly dictate that the isonet is never to be connected to the Internet, not all mistakes can be prevented. Hence the vulnerability host does not run any network servers other than the file exporting servers, and these are configured to export only to other isohosts. For example, no remote logins or file transfers are allowed. Second, the IP address of the vulnerability host is the same as the IP address of one of the Department's busiest servers. The vulnerability host is also a much faster system. Thus, should the isonet ever be connected to the Internet, it will receive messages intended for the Department server, and refuse them as it runs none of the daemons that the Department server makes available. Experience shows that users will quickly report these problems to the system staff, who can take immediate corrective action.

3.3 Other Aspects

Because systems are often altered for testing, the isonet hosts are maintained as close to the vendor-distributed configuration as possible. This makes reinitializing hosts very simple, and eliminates the problem of porting locally developed configuration management code to new types of systems as they are hooked up to the network. Two aspects of this are worth some elaboration.

Identification and authentication arise twice: first, in the issuing of new accounts, and second, in the access procedures. Accounts are under the control of the Computer Security Laboratory, so only those who have a legitimate need to access the isonet will receive accounts on it. To log in, a user must first enter a locked room (authorized users and graduate students have keys), and then go through the system authentication mechanism. Additional mechanisms have not proven necessary.

While auditing would allow tracking of uses of the isonet hosts, it would also provide no barrier to a determined insider. Further, providing a robust audit mechanism in this environment would require extensive system modifications. Given these, relying on the standard system audit mechanisms seemed appropriate.

3.4 Analysis

Saltzer and Schroeder's principles of secure design [3] are a useful metric against which to evaluate this design. The principles they enunciate are:

1. Users (processes) should have the minimum set of access rights necessary (least privilege).
   Procedurally, users are assigned to the least upper bound of the compartments they must access to complete their tasks. The security mechanisms ensure that access to isoinfo is limited to those authorized by the class membership.
2. Access requires explicit permission (fail-safe defaults).
   By default, users are members of the generic *other* group. Membership in a group must be given explicitly.
3. Design of the security mechanisms should be simple and small enough to be verified (economy of mechanism).
   The model presented above cannot be simplified any further; the analysis in the previous section shows the model meets the stated requirements. The analysis in this section demonstrates that the model is implemented correctly.
4. Every access should be checked for authorization (complete mediation).
   The UNIX operating system does not fully enforce this principle, checking authorization only when files are opened. But the above implementation meets this criterion to the extent possible.
5. Security should not depend on the secrecy of the design (open design).
   The model and its implementation are available to all users.
6. Access to objects should depend on more than one condition being satisfied (separation of mechanism).
   Access to the isonet itself requires physical access to a graduate student laboratory that is kept locked. Access to the data or to programs requires both an account and group membership; if the access enables the user to alter either data or programs, an account on one specific host is also required. Thus, several conditions must be met before access to objects is allowed.

7. Mechanisms shared by multiple users should be minimized (least common mechanism).
   This principle cannot be enforced adequately because the isonet provides common hardware and softwared. In particular, if the owner of data in a particular compartment desires to make that data available to all others, a simple change of protection mode accomplishes this. Having trusted users own as many files ameliorates this considerably, but the threat is not eliminated.
8. Mechanisms must be easy to use (psychological acceptability).
   The implementation mechanisms are standard UNIX security and system administration mechanisms and so are familiar to all our users. They require no extra software or hardware and thus are likely to be applied correctly and entail little to no extra burden on the users.

Thus, both the model and the implementation of the model meets the principles of secure design as much as possible in the UNIX environment

Because data from the Internet (and other sources) resides on the isonet, and data and programs are added as they become available, the above model must be modified to include the motion of data to and from the isolated network. The next section describes the modifications to the model and the implementation of a solution to this problem.

4. Uploading and Downloading

As research into vulnerabilities began, many helpful users and system administrators offered copies of various attack tools found at their sites. They enciphered these scripts using PGP [5] and mailed them to the first author, who moved the letters to a Macintosh, deciphered them, and put the cleartext onto floppies. The cleartext attack Tools could then be placed on the isonet.

Augmenting the model to handle the reclassification of data from Low to High would imply that users in the compartments would be able to read and write to the Internet. This is undesirable for several reasons. First, the users may download sensitive information without meaning to, for example by mistyping a file name. Second, users may upload programs with Trojan horses or other malicious logic. Third, if a user of the isonet wishes to pass information to someone on the Internet, or to use an attack script against an Internet host, the user may download the information or the attack script. Thus, access to the Internet is an exception to the rules of the model. This emphasizes the trust in those granted such access.

The moveable disk is a disk that can be connected to either a system on the Internet or a system on the isonet, but not both simultaneously. Because this is the *only* hardware that can be on both the Internet and the isonet, its management is central to the security of the vulnerability and attack data. Its requirements are:

1. The structure of the isonet must not be visible from the Internet. This allows the isonet to be reconfigured, and network infrastructure added, without affecting any other hosts or databases.
2. The moveable disk cannot contain any programs other than those being transferred. In particular, no system or user binaries may reside on it.
3. Themoveable disk must not be a networked disk. Uploading data to, or downloading data from, the Internet host requires the user to be physically at the Internet host, to which the moveable disk hardware is attacked.
4. The moveable disk must hold no sensitive data (except, possibly, for the data being uploaded or downloaded). This includes cryptographic keys.

The moveable disk requires special-purpose hardware (basically, a mounting bay) and so can only be used on systems with that hardware installed. This allows tight control over which hosts can be accessed, but requires a two-step process to upload data (the data must be placed on the moveable disk attached to the Internet host, and then that disk physically moved to an isohost and the data transferred).

Figure 2 shows the relationship of the moveable disk to the isonet and the Internet. The mamagement procedures and hardware set-up implement the above requirements directly. Further, only those users trusted to upload or download data or programs have accounts on the Internet host; in other words, accounts on this host are completely independent of the accounts on the other isonet hosts. If a user without an account on the Internet host wishes to move data between the isonet and the Internet, a trusted user must perform the transfer; as trusted users will question the need for such a transfer, this performs the function of a trusted certification that the data may indeed be transferred without endangering confidentiality.

Remote users and system administrators who wish to contribute data, programs, or attack tools are rarely willing to send the information over the Internet in the clear. To allow the information to be sent in encrypted form, a PGP key pair is associated with the isonet. The secret key resides on the vulnerability host. The public key is available on a number of public servers, and may be distributed freely. Contributors of information may use this key to encipher the data before they send it; as the secret key resides on the isonet, the recipients can only decipher the contribution on that network.

5. Conclusion

The isolated network has been in use for two years, and the current model evolved from our experiences and the needs of our laboratory and contributors. It appears to work quite well, as we have not yet had a leak of information from the isonet. This also speaks of the character of the users of the isonet, and emphasizes the need for non-technical controls. Undoubtedly, as the uses and needs of the project change, the model will evolve; in particular, network infrastructure will be added to enable us to test network-based vulnerabilities.

The isonet is a component of the Information Warfare Forensic Center. The IWFC's mission is to study the nature and types of vulnerabilities in complex systems including operating systems, network applications, and the network infrastructure (such as DNS, routers and their protocols). Among its goals are an understanding of why vulnerabilities occur, how to prevent and detect them, how to detect exploitation of vulnerabilities, and how to classify vulnerabilities. The development of vulnerability models is central to meeting these goals. Another primary objective is to develop forensic tools and methodologies to detect, analyze, and counter attacks. These tools will provide the foundation with which we can observe and analyze vulnerability exploitation and their effects in progress. The isolated network provides the foundation for experiments in support of these goals.

Acknowledgements: This work was supported under a contractual arrangement with the United States Air Force. Our thanks to our sponsors, especially Kevin Ziese and Scott Waddell, and to those who have helped build, and rebuild the isolated network, especially David O'Brien, Michael Dilger, and Scot Templeton.

6. References

[1] D. Bell and L. LaPadula, "Secure Computer Systems: Mathematical Foundations and Model," Technical Report M74-244, MITRE, Bedford, MA (Oct. 1974).

[2] S. Lipner, "Non-Discretionary Controls for Commercial Applications," *Proceedings of the 1982 Symposium on Security and Privacy* pp. 2-10 (Apr. 1982).

[3] J. Saltzer and M. Schroeder, "The Protection of Information in Computer Systems," *Proceedings of the IEEE* **63**(9) pp. 1278–1308 (Sep. 1975)

[4] Sun Microsystems, Inc., "NFS: Network File System Protocol," RFC 1094 (March 1989).

[5] P. Zimmerman, *PGP User's Guide* (Sep. 1992).