Development of a Honeynet Laboratory: a Case Study

Michael O'Leary, Shiva Azadegan, Jay Lakhani
Towson University

Abstract

Honeynets, which are designed to be digital decoys, have proven to be valuable research and teaching tool in the area of computer security and information assurance. In this paper, we discuss the development and implementation of a honeynet laboratory at Towson University. We present some background information, detail the development and implementation of the laboratory and share our challenges, experiences and learning.

keywords:
Honeynets, computer security, security education

Introduction

In this paper we describe our experiences developing a honeynet as part of our graduate education in security. Current research [1,2,3] indicates that honeynets can be safely deployed in academic environments and universities and used as a valuable teaching and research tool. Our main purpose in deploying the honeynet is to identify and detect security vulnerabilities, worms, malicious code, attack patterns, etc. We also use the honeynet to enable university security officials to enhance and better secure the campus infrastructure and raise awareness. Also, we plan to use the honeynet as a test bed for research projects in the area of information assurance and to incorporate the results into our computer security curriculum [7,8,9,10].

Background and related work

A honeynet [4,5,6] is a network of computers running honeypots. A honeypot is a closely monitored system that is intended to be attacked and or compromised, which can then be used to study the attack methods and patterns. Normally a honeynet has no production value and therefore any traffic entering or leaving the honeynet is an unauthorized activity. An outbound connection from a honeynet machine is an indication of a compromised honeypot system. Honeynets play a key role in a defensive strategy.

Honeypots and honeynets have been in existence for almost a decade. The honeynet project [4] was founded in 1999 with the main purpose of improving the security of the Internet. The honeynet project site provides an excellent set of honeynet-related publications, tools and resources. In this section, we briefly present the background information relevant to our project.

One way to categorize honeynets is based on their level of interaction. Using this criterion, we can divide honeynets into three major categories: low-interaction, medium-interaction and high-interaction. The interaction level of a honeypot is directly related to the amount of data that can be collected from intrusions. High-interaction honeypots, as the name suggests, can collect a great amount of data since the intruder has a great deal of interaction with the honeypot. However, the more flexibility the intruder has, the more risk that is involved with having that system operational. Low-interaction honeypots limit what the intruder is able to do, and therefore are less of a risk. However, these honeypots also tend to generate less data, as an intruder who cannot accomplish anything is likely to leave. Medium-interaction honeypots offers more ability to interact than do low-interaction honeypots but less functionality than high-interaction honeypots. A virtual honeypot is a program that emulates operating systems and services. Honeyd [11] is a low interaction virtual honeypot that emulates operating systems and services that can responds to our target IP addresses.

In this project, we decided to start with generation I honeynets, which are low interation and much simpler to deploy.

Development

The honeynet project at Towson University (TU) started in Summer 2005, with an initial meeting with representatives from our Office of Technology Services (OTS) which is the organization responsible for maintaining the campus computer network.

Our honeynet is designed to capture potential attacks originating from any point on the Internet; as a consequence it needed to be installed outside the campus firewall. Further, students and faculty need to work on the honeynet machines on a regular basis, both to examine the collected data as well as to modify and update their configuration. Thus the honeynet laboratory also needed to be located within the building housing the Computer and Information Sciences Department. This presented some difficulties to OTS, who wanted to ensure that unfiltered traffic to the honeynet laboratory was not accidentally bridged back into the campus network beyond the firewall. As a solution, we used existing but unused fiber from the campus data center to create a dedicated unfiltered line for the honeynet project. Thus all traffic to and from the honeynet laboratory is physically segregated from the filtered traffic that is part of the regular campus network. The expense in doing so was significant, and borne by the Computer and Information Sciences Department.

Much of the actual work needed by OTS to establish the connection to the honeynet laboratory was done by one of the students on the honeynet project who also worked for OTS.

Concurrently with the work on the technological problems, we also met with the University Counsel to examine the legal implications of running the honeynet laboratory. There are a number legal issues that need to be considered before running a honeynet; some of these are contained in [12, Chp. 8]. In this paper we describe our experiences

with out legal counsel; however we are not lawyers, and this should not be considered to be legal advice.

The first major issue we faced were federal and state wiretap laws. These forbid the interception of the content of any form of electronic communication, except in special instances. The first case is the “consent of party” exception. Federal law allows monitoring if one of the parties to the communication explicitly allows it. In a honeynet, this can be done by appropriately bannering provided services for example. However, state law in Maryland is different, as it requires all parties to a communication to consent. In particular, if someone on the honeynet contacted Google for example, we would need to have Google's consent before being allowed to monitor the text of the communication. In general, this protection is insufficient in Maryland to allow us to run a honeynet.

Another exception is the “provider protection” exception. Provided the monitoring by the honeynet project is used to protect our campus network and servers here at Towson, then we could legitimately use this exception. As a consequence, we are working closely with OTS and both the acting Campus Chief Information Officer and our Information Security and Compliance Manager. We have codified the relationship between the honeynet project and OTS with a formal memorandum that describes how the data collected from the honeynet will be provided to OTS and used. We have also jointly developed policies that determine how the data is shared, that coordinate intrusion response policies, and that describe how to react to potentially criminal activity that is detected. To keep the lines of communication open, another of our graduate students on the honeynet project is working with the campus Information Security and Compliance Manager to fine tune the campus intrusion detection system. The Campus security officer also regularly attends our honeynet / computer security lecture series.

There are other legal issues that were considered as well. These include Pen register, trap and trace device statutes, which have exceptions similar to the wiretap regulations. We also have constructed the honeynet so that connections outbound from the honeynet are closely monitored and can be shut off to prevent attacks on third parties originating from honeynet machines.

Implementation

The equipment for the laboratory consists primarily of surplus computers and networking equipment, all of which was provided by the Computer Science Department. This includes

- Cisco Pix 501
- Cisco Catalyst 2950
- 6 PC's
- One-way network cable
- Crossover network cable

Figure 1 depicts the overall design of our honeynet. The firewall is used for egress traffic filtering, and provides a gateway for the honeynet environment. It is designed to block any unestablished outbound connections. Inbound traffic is allowed to any of the public IP addresses we have reserved for the targets. This traffic is then translated to our local

network (via NAT) and then forwarded to the appropriate target. Return traffic is handled in the same fashion. At the same time, copies of all traffic that pass through the switch is forwarded through the monitor port through a read-only cable to our intrusion detection system. This lets our IDS monitor all the traffic passing through the network, while still being relatively well-protected from outside interest.

The targets themselves are VMWare guests running on a Windows host. We are using VMWare hosts for the targets because it is easy to maintain, back-up, and store the state of a VMWare guest. Each physical host runs two guests, and we have two hosts, giving us four targets in total. On each of the VMWare guest targets, we are running honeyd, which is a low interaction honeypot.

The intrusion detection system in a SuSE linux system running snort, with a customized ruleset. The collected data is sent via a crossover cable to a second database server running MySQL under SuSE linux.

Software:

- ❖Pix IOS 6.3(4)
- ❖Cisco IOS 12.1(12c)EA1
- ❖VMware workstation version 5
- ❖honeyd 1.5
- ❖snort 2.4.3
- ❖mysql 4.1.10

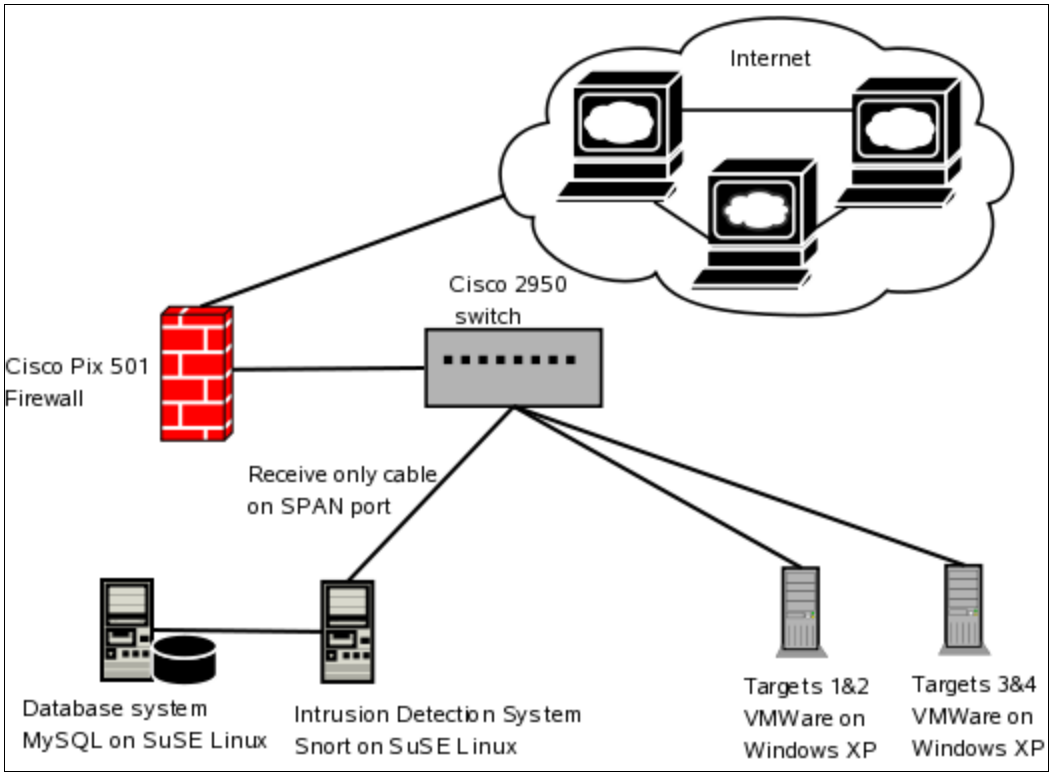

Figure 1

Jay Lakhani, one of the graduate students working on this project, was the primary person responsible for the setting up the lab.

Experiences & Learning

Though the project is less than a year old, we have learned several lessons that we would like to share with you.

- ❖We underestimated the amount of time required to set up the laboratory. We initially budgeted two months for one graduate student to set up the laboratory; in fact it took three students almost three months to complete the initial setup.

- ❖The close cooperation between us and the security officers at the OTS was crucial to the success of the project.

- ❖We decided to use an isolated stand-alone blog server in the laboratory to facilitate documentation and communication among the team members. We also have weekly meetings to review and share progress reports.

- ❖We found out that some of the widely published scripts, such as honeyd script, still contain problems with running on the windows environment.

- ❖Our students first installed and upgraded Analysis Console for Intrusion Databases (ACID). They then found it very challenging to implement Basic Analysis and Security Engine (BASE) which is another front end for snort IDS system.

Conclusion

Honeynets provide a valuable teaching and research tool and provide students with most up-to-date security challenges and threats. Moreover, honeynets can be a valuble addition to an institution security system. At Towson University to better prepare and educate our students in the area of information assurance we have developed a honeynet laboratory. The project is a close collaboration with the University Office of Technology Services; legal agreements have been established with the University's legal department; and memorandum of understanding was made between the University and the Department of Computer and Information Sciences.

References:

[1] Jones, Romney, "Honeynets: An Educational Resource for IT Security", Proceedings of ACM SIGITE 2004, Salt Lake City, Utah.

[2] "Know Your Enemy: Honeynets in Universities", Honeynet Project, www.honynet.org.

[3] Levine, LaBella, Owen, Contis, Culver, “The use of Honeynets to Detect Exploited Systems Across Large Enterprise Networks”, Proceedings of IEEE Workshop on Information Assurnance, West Point New York, June 2003.

[4] **http://Honeynet.org**

[5] Lance Spitzner, Honeynets tracking Hackers, Addison Wesley, 2003.

[6] Know Your Enemy: Honeynets, www.honeynet.org/papers/honeynet/ index.html

[7] M. O'Leary, *A Laboratory Based Capstone Course in Computer Security for Undergraduates*, Proceedings of SIGCSE 2006.

[8] S. Azadegan, M. O'Leary, A Wijensinha, and M. Zimand, *Undergraduate Computer Security Education: A Report on our Experiences & Learning*, Proceedings of WECS 7, January 2006.

[9] S. Azadegan, M. Lavine, M. O'Leary, A. Wijesinha, and M. Zimand, *A Dedicated Undergraduate Track in Computer Security Education*, in *Security Education and Critical Infrastructures*, Cynthia E. Irvine, Helen Armstrong (Eds.), IFIP Conference Proceedings **253**, Kluwer, 2003, pp. 319-332.

[10] S. Azadegan, M. Lavine, M. O'Leary, A. Wijesinha, and M. Zimand, *An Undergraduate Track in Computer Security*, in Proceedings of the 8th Annual Conference on Innovation and Technology in Computer Science Education (ITiCSE-03), Thessaloniki, Greece, June 30 - July 2 2003. pp. 207-210.

[11] **http://Honeyd.org**

[12] L. Spitzner “Know your Enemy”, 2nd Edition, Addison-Wesley.