Evaluating Intrusion Detection and Prevention Systems Using Tomahawk and Wireshark

David Mudzingwa and Rajeev Agrawal
Department of Electronics, Computer and Information Technology
North Carolina A&T State University, Greensboro, NC, USA

Abstract - *The increase in the security breach of computer systems and computer networks has led to the increase in the number of security tools that seek to protect these asserts. Among these tools are intrusion detection and prevention systems (IDPS). An IDPS is a security system that is used to detect and prevent security violations. Evaluating the effectiveness of IDPS is complicated and there has not been much work done on ways IDPS users can follow to evaluate the IDPS. Most of the work on evaluating IDPS is focused on developing new testing methodologies. This paper seeks to offer a practical approach to evaluate both hardware and software based IDPS using publicly available open source tools Tomahawk and Wireshark.*

Keywords: Intrusion detection and prevention system (IDPS), Tomahawk, Wireshark, PCAP file

1 Introduction

Intrusion detection and prevention systems are security tools that are used to detect and prevent security threats to computer systems and computer network systems. Although these systems are widely adapted and continue to grow, they continue to be very difficult to evaluate. This is due to the lack of publicly available research and test data sets. The available test data sets are dated [1]. Test data sets have to be current and public available so that interested parties can evaluate the quality of the data sets and effectiveness of IDPS. The available work in this area mainly focus on improving the methodologies used in evaluating IDPS instead of offering a simple way to evaluate the IDPS. There are non-commercial tools that can be used to test and validate the effectiveness of an IDPS, but there are still challenges in acquiring test data that contains live exploits needed to perform credible tests. There are also other challenges in correctly evaluating IDPS such as setting up a test environment and properly conducting the evaluation. This paper presents a simple but effective way to evaluate both the appliance/hardware and software based IDPS. This paper focuses on setting up an evaluation lab using publicly available tool Tomahawk for replaying network traffic that is used in the evaluation.

2 Related Work

The first research that looked the claims of the intrusion detection systems was done in 1998. This work put forth a framework for thoroughly testing an intrusion detection system and also offered data sets for use in evaluations. This early work is evaluated and checked for accuracy in [2]. This work challenged the continual use of the data sets produced in IDPS tests. This work puts on the argument that these data sets are out dated and that the procedures used to generate the data sets were not representative of a production network [3]. These limitations in the evaluation of IDPSs were addressed by the Lincoln Adaptable Real time Information Assurance Test bed (LARIAT) [8]. This was a better testing application that used a graphical user interface instead of the command line and was easier to use but was only available to the United States government [8]. Trident evaluation was another work that tried to improve on the early works by introducing ways to add new background and attack traffic to the test data sets [9]. This work offered a way to account for evasion techniques during an evaluation of an IDPS. A comparison of two IDPS methodologies to evaluate them is given in [5]. This work describes how to set up a test bed that can be used to evaluate an IDPS using a Snort and Spade. Before an IDPS is evaluated for effectiveness, its underlying detection methodologies need to be understood. An IDPS can be based on any of the four main detection methodologies. The signature based, anomaly based, stateful protocol analysis based, and the hybrid based detection methodology [16].

3 Background

3.1 IDPS Methodologies

There are many different methodologies used by IDPS to detect changes on the systems they monitor. These changes can be external attacks or misuse by internal personnel. Among the many methodologies, four stand out and are widely used. These are the signature based, anomaly based, Stateful protocol analysis based, and hybrid based. Most current IDPS use the hybrid methodology which is a combination of other methodologies to offer better detection and prevention capabilities. All detection methodologies use the same general model and the

differences among them is mainly on how they process information they gather from the monitored environment to determine if a violation of the set policy has occurred. Figure 1 shows a broad architecture of which these systems are based on. This architecture was developed by the Intrusion Detection Working Group and has four functional blocks, the Event blocks which are the event boxes that gathers event from the monitored system and will be analyzed by other blocks, then the Database blocks which are the database boxes which stores the events from the Event blocks, then the Analysis blocks that processes the events and sends an alert, and final the Response blocks whose purpose is to respond to an intrusion and stop it [15].

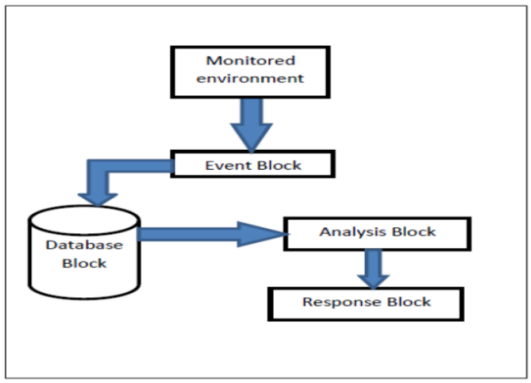

Figure 1. General architecture for IDPS systems

3.1.1 Anomaly Based Methodology

Anomaly based methodology works by comparing observed activity against a baseline profile. The baseline profile is the learned normal behavior of the monitored system and is developed during the learning period were the IDPS learns the environment and develops a normal profile of the monitored system. This environment can be networks, users, systems and so on.

3.1.2 Signature Based Methodology

Signature based methodology works by comparing observed signatures to the signatures on file. This file can be database or a list of known attack signatures. Any signature observed on the monitored environment that matches the signatures on file is flagged as a violation of the security policy or as an attack. The signature based IDPS has little overhead since it does not inspect every activity or network traffic on the monitored environment. Instead it only searches for known signatures in the database or file.

3.1.3 Stateful Protocol Analysis Based Methodology

The Stateful protocol analysis methodology works by comparing established profiles of how protocols should behave against the observed behavior. The established protocol profiles are designed and established by vendors. Unlike the signature based methodology which only compares observed behavior against a list, Stateful protocol analysis has a deep understanding of how the protocols and applications should interact/work.

3.1.4 Hybrid Based Methodology

The hybrid based methodology works by combining two or more of the other methodologies. The result is a better methodology that takes advantage of the strengths of the combined methodologies.

4 Setting Up IDPS Testing Environment Using Tomahawk

Tomahawk is an open source network tool that can be used to generate background network traffic, replay network traffic, and manipulate network traffic using captured network traffic files. The captured traffic can then be replayed during the evaluation of an IDPS. For the evaluation to be effective the traffic capture should come from the environment where the IDPS will reside. This produces a more accurate test. Once the traffic is captured, it can be replayed in a controlled environment where more experimentation can be done without negatively affecting the production environment. Within the controlled environment the captured traffic can be used as background traffic while known exploits are introduced to the monitored workstation/server.

The following are the minimum hardware and software requirements for using Tomahawk to test an IDPS:

Hardware and Software Requirements

- Workstation or server with a minimum of two network cards (running Linux/Unix flavor operating system)
- Second workstation or server with a minimum of two network cards
- Captured network traffic file in the libpcap format (cleaned)
- Network traffic capture tool (Wireshark)
- A minimum of a 2.0 GHz Pentium equivalent processor
- A minimum of 1GB of RAM (2 or more is recommended)
- Network switch (optional)
- Management pc (optional)
- Third network card on the other workstation/server (optional)

- Network cables (crossover optional)

4.1 Hardware Setup

There are three ways of configuring the hardware for testing an IPDS using Tomahawk. A basic way with just two computers with a minimum of two network cards each, a medium setup which is a basic setup with a hardware IDPS, and an advanced way which is a medium setup with an addition of a network switch and a management computer. These setups do not include an internet connection as a way to control the network traffic during the evaluation of an IDPS using tomahawk. If desired an internet connection can be added to either the attacker or the attacked computer.

4.1.1 Basic Set Up

A basic setup is the simple way to test a software based IDPS. As shown in figure 2 only two machines with two network cards and two crossed-over network cables are required. One of the machines is configured as the attack machine and Tomahawk is installed on it. The other machine is configured as the attacked machine and the software based IDPS is installed on it. The two machines are connected to one another with the crossover cables. Tomahawk only runs on Unix or Linux based operating systems and as a result the attack machine will require a Linux or Unix based operation system. The attacked machine can run any current operating system.

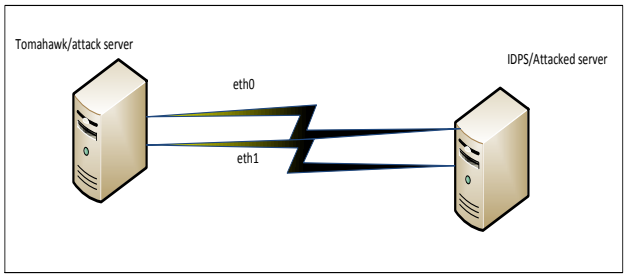

Figure 2. Basic Setup

4.1.2 Medium Setup

The medium set up is also simple but adds a hardware/appliance based IDPS. It requires two machines with two network cards each and three network cables. The machines are connected through the IDPS as shown in figure 3. In this setup the IPDS saves as a network switch connecting the two computers. The computer that tomahawk will be installed on has to have a Linux or Unix based operating systems. Also the computer that has tomahawk running on it must have two network cards that will used by tomahawk. The other computer that saves as the attacked one can have any of the current operating systems.

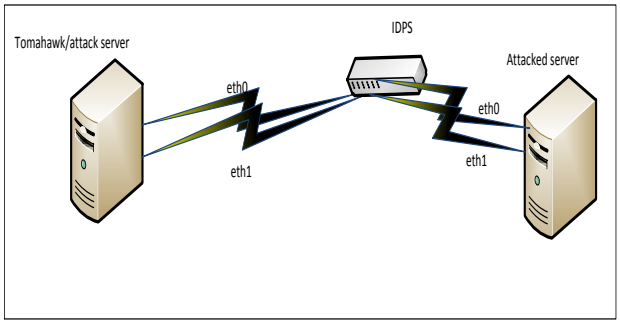

Figure 3. Medium Setup

4.1.3 Advanced Setup

The advanced setup is more involved than the other two as it adds a third network card, a switch, five network cables, a management machine, and the internet. The machines are connected through the switch and the IDPS as shown in figure 4. One of the machines with three network cards is configured as the attack machine and Tomahawk is installed on it. The other machine is configured as the attacked machine and sets on behind the IDPS and a third management machine is connected to the switch. This the ideal setup for testing IDPSs as it allows for different configuration changes to be made. For example more computers can be added to the test by adding another switch between the IDPS and the attacked computers or by adding more computers on both sides. This would allow for evaluating the IDPS behavior under high network traffic. .

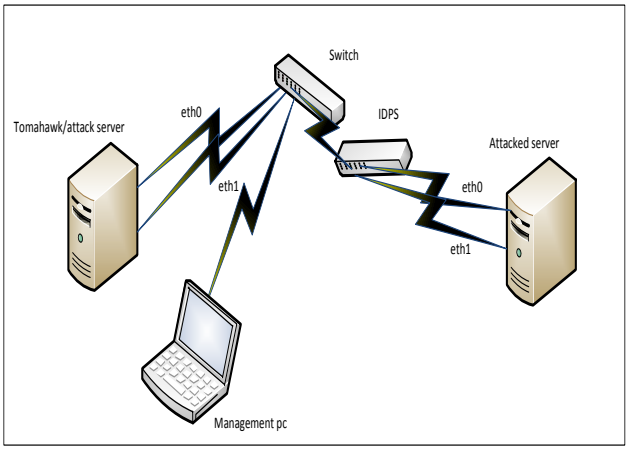

Figure 4. Advanced Setup

5 Advantages of using Tomahawk

Tomahawk was chosen for this setup due to the advantages it offers over other tools that replays captured network packets. Tomahawk uses simple commands and flags that can be teamed together to easily manipulate the traffic going to the attacked computer. It can take a small packet and manipulate it to produce the desired traffic flow.

Some advantages of using tomahawk include:

- Tomahawk is free and publicly available
- It is simple to use
- It is very stable and mature
- Does not require a lot of resources to run it
- Can evaluate both software and hardware based IDPS

5.1 PCAP File

A PCAP file is a file that contains captured network activity and saved in the libpcap format with a PCAP extension. This format and extension allows the file to be used by multiple network related tools on most current operating systems. Tomahawk works by replaying and manipulating PCAP files. Tomahawk does not create its own PCAP files but they can be created by other network monitoring tools such as Wireshark. Wireshark is an open source network monitoring tool that has multiple functions and it runs on most current operating systems. PCAP files can also be downloaded from the Internet from trusted sources. There are advantages to creating own PCAP file for use with Tomahawk. Using own PCAP files allows to use traffic that is representative of the environment where the IDPS will reside and protect. It also allows the capture of traffic at different times with diverse load situations. This facilitates different mixes of traffic volumes and applications on the network. Using a downloaded pcap may not present a true picture of the environment being tested which can led to a wrong IDPS been chosen.

6 Testing Methodology

Tomahawk can be configured and used in variety ways to support different test configurations. The three setups described above are examples of setting up different test environments for evaluating an IDPS using Tomahawk. Tomahawk works the same way regardless of the configuration of the setup and it supports both software and hardware based IDPS. Tomahawk works by replaying captured network packets that are saved as a pcap file in a bi-direction fashion and breaking the PCAP file into two pieces and then assigns these pieces to the client and the other to the server [13]. Using this system allows Tomahawk to keep track of the PCAP file as it is replayed. By breaking the PCAP file into packets allows Tomahawk to assign the first IP address it encounters in the PCAP file to the client and the second IP address to the server. This process is repeated until the whole pcap file is replayed entirely. Once the PCAP file is broken down into packets and the client and server IP addresses are assigned, tomahawk starts replaying the packets. The client packets are sent out on eth0 and the server packets are send out on eth1. Tomahawk has a default of 0.2 seconds for re-transmissions of lost packets and it also auto manages other network related tasks such as MAC addresses, client and server IP address. If the IDPS detects and blocks the pcap file that contains an attack, tomahawk will report a time out. If tomahawk reports that the pcap containing an attack completed without any errors, then the IDPS will have missed the attack [13].

6.1 Capturing the PCAP files

We use Wireshark to capture and create four PCAP files that we use in our test environment. Creating PCAP files with Wireshark is documented in [12]. The PCAP files we use are created using default settings in Whireshark. We started Wireshark and started recording the network traffic and then initiated the attacks/exploits that way we capture all the packets related to the attack. The following PCAP files were created:

6.1.1 PCAP1

This is a simple file that contained normal network traffic and no attack traffic. This file is a capture of traffic browsing a server. This capture will be used to test how the IDPS handles normal traffic and establish some baselines.

6.1.2 PCAP2

This file is a capture of a known OS exploit and will be used to test if the IPDS will detect and respond to the attack.

6.1.3 PCAP3

This file contains a DOS attack on the server and will be used to test how the IDPS detects and responds to the attack.

6.1.4 PCAP4

This file is a capture of an exploit and a DOS attack while there is high volume of traffic on the network. This file will be used to verify how the IDPS reacts under different situations.

6.2 Using Tomahawk

Tomahawk is a command line utility that runs on Linux based operating systems. Tomahawk commands can be used to run a basic evaluation on an IDPS. To use Tomahawk just type *tomahawk* on the command prompt followed by any of the flags. A detailed explanation of Tomahawk's every command and flag is detailed in [13]. In our test setup we used the Medium setup described above with the following hardware and software:

- The attack workstation- IBM Workstation running SUSE Linux
- A switch that has DHCP and IDPS capabilities
- The attacked server- An IBM Workstation running SUSE Linux
- Four network cables

The attack workstation and the attacked server were connected through the switch/IDPS. Care was taken to make sure that all the traffic from the attack workstation to the attacked server passed through the IDPS.

The first test involved the PCAP1 been replayed against the attacked server and these are the Tomahawk commands we used:

tomahawk -l 2 -f pcap1.pcap

This command replayed the pcap1 file twice and produced the following output:

Beginning test

Completed 1 loop of trace pcap1.pcap

Completed 1 loop of trace pcap1.pcap

Finished 2 loops of trace pcap1.pcap Completed: 2, Timed out: 0 Retrans: 0 Sent: 1686 Recv: 1686

This output shows that the both replays finished without being blocked. If the pcap1 file was blocked/dropped by the IDPS then the loop will have not completed.

tomahawk –l 2 –f pcap2.pcap

The above command replayed PCAP2 file which contained a known exploit against the server. The IDPS blocked this attack and the packets were dropped. As a result the loops did not complete.

tomahawk –l 2 –f pcap3.pcap

The above command replayed PCAP3 file which contained a DOS attack against the server. The IDPS blocked this attack and the packets were dropped. As a result the loops did not complete.

tomahawk –l 2 –f pcap4.pcap

The above command replayed PCAP4 file which contained an exploit and a DOS attack while there is high volume of traffic on the network. These replay packets were dropped by the IDPS and as result the loops did not complete.

7 Conclusion and future work

There are two main problems in evaluating the effectiveness of an IDPS. One is the lack of previous work that lays down a simple and straight forward way for setting an evaluation lab that can evaluate both a software and hardware based IDPS. The other problem is the lack of publicly available datasets that can be used in the evaluation of an IDPS. This work targeted both problems by presenting a basic setup for evaluating an IDPS using Tomahawk and Wireshark. The evaluation setup presented three ways to setup the hardware and software involved in testing an IDPS and how it can be used to evaluate both the software and hardware based IDPS.

Future work entails building out a test environment based on the setups presented here and carrying out some experiments with different IDPS products currently on the market. The work will also involve documenting how to create PCAP files for use with Tomahawk using Wireshark.

8 References

[1] J. W. Haines, R.P. Lippmann, D.J. Fried, M.A. Zissman, E. Tran, and S.B. Boswell. 1999 DARPA Intrusion Detection Evaluation: Design and Procedures. MIT Lincoln Laboratory: Lexington, MA, 2001.

[2] J. McHugh. "Testing Intrusion Detection Systems: A Critique of the 1998 and 1999 DARPA Intrusion Detection System Evaluations as Performed by Lincoln Laboratory", Proc. ACM TISSEC 3(4) 262-294, 2000.

[3] Mahoney, M. V. & Chan, P. K. (2003), An Analysis of the 1999 DARPA/Lincoln Laboratory Evaluation Data for Network Anomaly Detection, In Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection, Springer-Verlag, 220-237, 2003.

[4] Brugger, S. & Chow, J. An assessment of the DARPA IDS Evaluation Data set using Snort, Technical report, Department of Electrical Engineering and Computer Sciences, University of California, Berkeley, 2007.

[5] Cardenas, A.A., Baras, J.S., Seamon, K.: A framework for the evaluation of intrusion detection systems. In: Proc. of the IEEE Symposium on Security and Privacy (IEEE security'06), Washington, DC, USA,IEEE Computer Society, 63–77, 2006.

[6] Sommers, J., Yegneswaran, V., & Barford, P. Toward comprehensive traffic generation for online ids evaluation. University of Wisconsin: Tech Rep, 2005.

[7] Corsini, J. Analysis and Evaluation of Network Intrusion Detection Methods to Uncover Data Theft, Master's thesis, Edinburgh Napier University, Edinburgh, UK, 2009.

[8] Athanasiades, N., Abler, R., Levine, J., Owen, H. & Riley, G. Intrusion detection testing and benchmarking methodologies', Proceedings of First IEEE International Workshop on Information Assurance, IWIAS2003, 63-72, 2003.

[9] Sommers, J., Yegneswaran, V. & Barford, P. Toward Comprehensive Traffic Generation for Online IDS Evaluation, Technical report, Department of Computer Science, University of Wisconsin, Madison, 2005.

[10] Walsh, J & Koconis, D. Cleaning Packet Captures for Network IPS cleaning, 2006. http://www.icsalabs.com

[11] Sannella, M. J. Constraint Satisfaction and Debugging for Interactive User Interfaces. Doctoral Thesis. UMI Order Number: UMI Order No. GAX95-09398, University of Washington, 1994.

[12] Wireshark 2012. http://www.wireshark.org/

[13] Tomahawk 2012. http://tomahawk.sourceforge.net/

[14] T. Brugger. KDD cup'99 dataset (network intrusion) considered harmful,15Sept, 2007. http://www.kdnuggets.com/news/2007/n18/4i.html.

[15] Mudzingwa, D & Agrawal. R. A Study of Methodologies used in Intrusion Detection and Prevention Systems, Proceedings of the IEEE SoutheastCon2012, Mar 2012.