TELEMETRY NETWORK INTRUSION DETECTION TEST BED

Graduate Student Authors: Daryl Moten and Farhad Moazzami
Advisor: Dr. Richard Dean
Department of Electrical and Computer Engineering
Morgan State University
Baltimore MD 21239

ABSTRACT

The transition of telemetry from link-based to network-based architectures opens these systems to new security risks. Tools such as intrusion detection systems and vulnerability scanners will be required for emerging telemetry networks. Intrusion detection systems protect networks against attacks that occur once the network boundary has been breached. An intrusion detection model was developed in the Wireless Networking and Security lab at Morgan State University. The model depends on network traffic being filtered into traffic streams. The streams are then reduced to vectors. The current state of the network can be determined using Viterbi analysis of the stream vectors. Viterbi uses the output of the Hidden Markov Model to find the current state of the network. The state information describes the probability of the network being in predefined normal or attack states based on training data. This output can be sent to a network administrator depending on threshold levels. In this project, a penetration-testing tool called Metasploit was used to launch attacks against systems in an isolated test bed. The network traffic generated during an attack was analyzed for use in the MSU intrusion detection model.

KEY WORDS

iNET, Telemetry, Intrusion Detection System (IDS), Hidden Markov Model, Metasploit

INTRODUCTION

Cyberspace is the electronic medium of computer networks, in which online communication takes place. Science fiction author William Gibson first coined the term in one of his novels. Cyber security deals with detecting and responding to all kinds of attacks that take place in cyberspace. Today, our financial, energy, transportation, and other infrastructure systems are controlled by computer technology. Activities in cyber space can have consequences in the physical world. This reality prompted the director of national intelligence, James Clapper, to tell a congressional committee “We all recognize [cyber attacks] as a profound threat to this country, to its future, to its economy, to its very being.” [7] Recent events illustrate the need to continuously learn and update ways of protecting computing devices and networks, including telemetry networks.

The Wireless Networking and Security lab (WiNetS) at Morgan State University is dedicated to research network implementation techniques and best practices that support networks. This project was focused on designing an isolated test bed to support system attack simulation and intrusion detection.

In order to protect systems, one must understand the basic tenants of data security and networking protocols. Knowledge of protocols allows large amounts of network traffic to be reduced to simple data vectors.

These vectors are a key component of WiNetS intrusion detection model. Previously, the model successfully detected an FTP password attack and separately a SYN attack. Subsequently, we were able to combine the analysis to detect and distinguish both types of attacks. Our ongoing goal is to design intrusion detection logic to distinguish many different attack schemes. A major part of this effort was using the isolated environment to launch attacks against target systems. Metasploit Framework, a penetration-testing tool, was used to launch and analyze intrusion mechanisms. While those attacks are in progress, a network sniffer, such as Wireshark, can be used to capture the communication stream. Then that stream can be reduced to a vector to be analyzed by the intrusion detection system.

DATA SECURITY

Data security consists of protecting data from being viewed or manipulated by persons or systems not authorized to do so. The three main aspects of data security are confidentiality, integrity, and authentication. Confidentiality prevents data from being understood by unauthorized individuals. This is achieved using cryptography, the use of mathematical algorithms and number keys to scramble data. This encrypted data is known as ciphertext. Unintended recipients may be allowed to capture data in ciphered form, but they cannot understand the data without a decryption key. An attacker can only decipher the data if they can guess the key. Strong encryption algorithms, along with long key numbers, require the use of supercomputers with many years of ongoing computation to guess the correct key. Integrity is concerned with detecting altered data. An attacker could alter data as it is being routed to a recipient, causing malicious consequences. As data packets are routed to a destination, they may be accessible to others. For example, wireless data is accessible to anyone with a sufficient receiver. And on wired networks, many users may have access to the data traveling on the network. Thus, it is difficult to restrict data access to only the intended recipient. A more obtainable goal is detecting a message has been compromised and then discarding that message. The use of hash values and checksums can be used to ensure data integrity. Authentication is a mechanism of ascertaining the identity of the message originator. Knowing the originator of a message is very important since an attacker could masquerade as a trusted data source. Authentication can be achieved by using public key cryptography. The best way to ensure data security is by deploying mechanisms that provide confidentiality, integrity, and authentication.

BASIC NETWORKING

System networking involves the connection of computers and computer networks such that users in distant locations may share information. Protocols were established to connect different systems with predictable results. As systems networking evolved and applications became more complex, developing networking software became more difficult. The Open Systems Interconnect (OSI) model was created to segment communication tasks into logical layers. Each layer handles a specific set of services or data requests from adjacent layers. A sending system and a receiving system would process complimentary tasks at the same layer of the model. OSI defines 7 layers. Layer 1 defines the physical layer in which networking cables connect with systems. Layer 1 defines the application layer, the layer at which networking software interacts with other host software. A sending system creates data by sending it down the 7 layers of the OSI model, appending data, called headers, at each layer. On the receiving system, the data moves up the 7 layers, removing the headers and processing the data at each layer. However, OSI is simply a suggested model. Though the tasks that are needed for systems to connect for a certain application are similar, the design of the protocol to accomplish that connection may not map exactly to the OSI model. TCP/IP, The Transmission Control Protocol/Internet Protocol is the main suite of protocols used for system communication on the Internet. TCP/IP loosely follows the OSI model, as Figure 1 shows. The application layer maps to OSI layers 5 to 7. The host layers are application, presentation, session, and transport. The interface between that application running on the host requesting network connectivity and the host networking software occurs at the application layer. The presentation layer handles data representation and encryption (if handled by the networking software). Interhost communication handling is done at the session layer, where port numbers are used to identify complimentary connection with the remote system. End-to-end connections and reliability are handled at the transport layer. A request for retransmission of lost data is done at this layer. This is also the layer in which data streams are broken into segments, called packets. The media layers are Network, data link, and physical layers. Routers operate at the network layer by making decisions on how to forward data toward the destination address. Data link has 2 sublayers – MAC (media access control) and LLC (logical link control). Switches and Ethernet operate in the datalink layer. The MAC sublayer manages network interface card addressing and LLC handles physical cable interface. The use of protocol layers allows networking designers to manage the complexity of remote system communication.

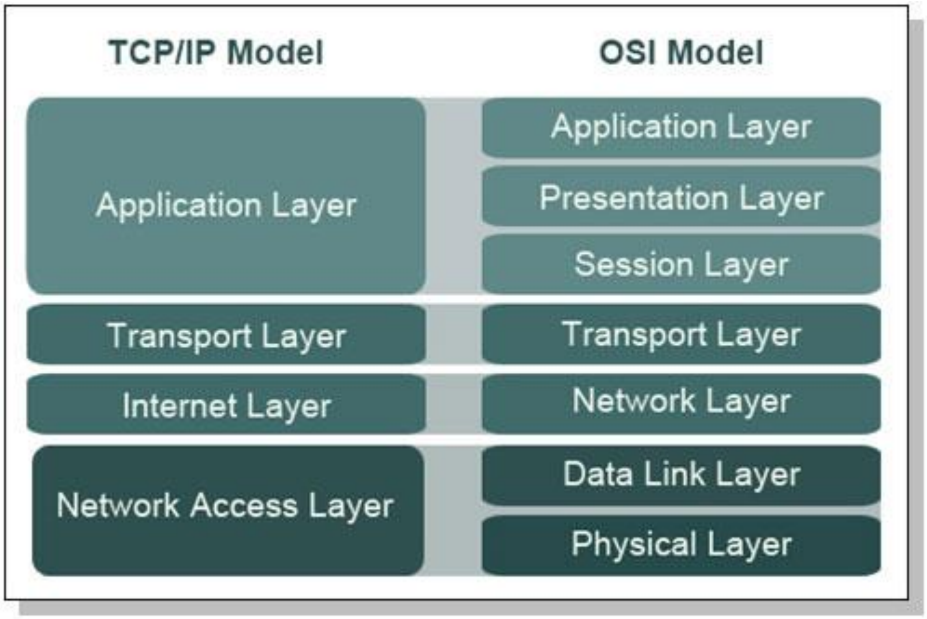

Figure 1 - How TCP/IP maps to the OSI Reference Model

INTRUSION DETECTION MODEL

Intrusion detection systems (IDS) protect networks against inside or outside attacks. These attacks occur after the network boundary has been breached. Attacks can also occur if a person with limited network access attempts unauthorized access to network resources. Once an intruder is inside a network, individual systems can be attacked and the data in those systems can be viewed or destroyed. Intrusion detection systems collect data using sensors placed at various network locations. The sensors collect data on network and individual system activity and forward this data to IDS servers. The IDS servers analyze this data in search for patterns that are consistent with network intrusion. If a log pattern is consistent with a possible break-in, an alarm is triggered and alerts are sent to network administrators. One example of an IDS trigger is successive login failures of a network host in a short time frame. A key feature of IDS is notifying network administrators soon after a compromise occurs. IDS sensors can be placed on workstations, servers, switches, routers, or other network devices. A crucial aspect of IDS is uploading log data to servers at regular intervals. Often, an attacker is able to break into a network asset and erase log file entries that contain evidence of their presence. In such a case, a network can be attacked multiple times without detection. To ensure log files reach the IDS server, network components with IDS sensors often have additional network cards and/or redundant routes to the IDS server.

The WiNetS intrusion detection model analyzes traffic flowing across a network. Statistical modeling of flowing traffic is performed to find states of the network based on patterns in the data. A network can be in a normal state, one of many attack states, or some intermediate state. This model does not employ IDS sensors or servers used in traditional intrusion detection systems. There are two (2) phases of the model – the training phase and the operational phase. Figure 2 shows a diagram of the model. In the training phase, normal and attack traffic are used to find all possible states of the network. The hidden Markov model (HMM) looks for patterns in data vectors. The vectors are constructed by using key fields of network traffic streams. Figure 3 shows an example of forming a vector from a stream of traffic. Each network stream type would have an associated vector structure. HMM uses these vectors to search for patterns. Those patterns are given state identification. HMM also uses statistical analysis to find relationships between those states.

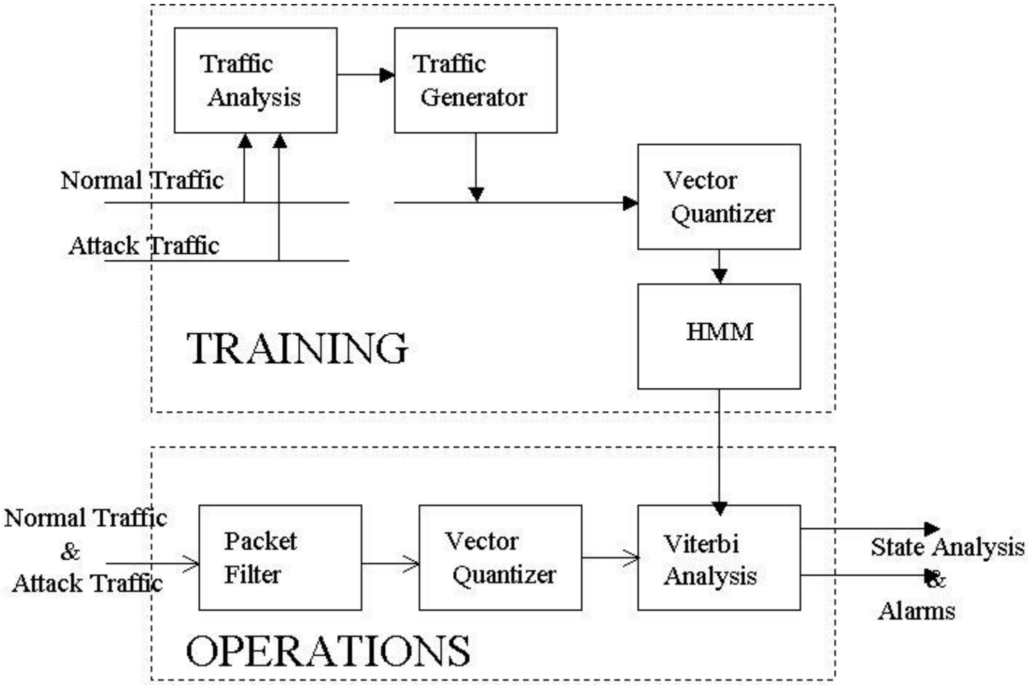

Figure 2 - MSU WiNetS Intrusion Detection Model

In the operations phase, live network traffic is analyzed to find the current state of the network. Existing network traffic is matched against the state information found during the training phase. Network traffic is filtered into predefined streams. These streams of data are reduced to vectors. The current state of the network can be determined using Viterbi analysis of those stream vectors. Viterbi uses the output of the HMM to find the current state of the network. The state information describes the probability of the network being in a predefined state based on the training data. This output can be sent to a network administrator based on defined thresholds. For example, if Viterbi analysis finds there is a 70% probability the network is in a certain attack state and the threshold for that state has been set to less than 70%, an alarm will be triggered. These alarm thresholds can be set for each network stream type, giving a network administrator control of the level of notification and false alarms.

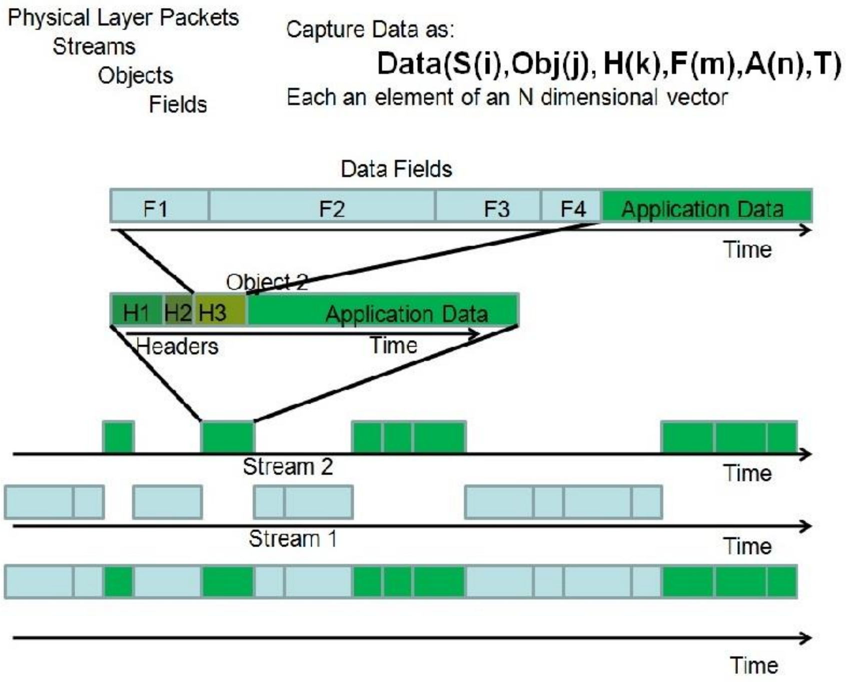

Figure 3 – A Vector formed from Network Streams and Associated Fields

Synthesis of the model components will eventually result in a complete intrusion detection system. The protocol analysis performed in this project will be used for vector quantization. Ultimately, the developed cyber security test bed will be used to test this intrusion detection model.

TEST BED

The test bed consists of hardware and software that function to form an isolated cyber security test bed. A diagram of test bed network is show in figure 4. Several connected computers act as attack targets in this isolated environment.

A protocol analyzer, also known as a network sniffer, is software that collects all data packets flowing across a network cable. The software puts the network interface card in a mode to process all packets on the cable. As the packets are captured, the data can be filtered for presentation to users. Several computers have the WireShark network sniffer installed. This is the main form of network monitoring in the test bed.

Additionally, the test bed includes a computer that acts as the intrusion detection system processor. Once the model is complete, this PC will be the main point of data analysis and alarm generation. A traffic generator is a software system than can construct valid network packets. The traffic re-player can record network traffic to be replayed at later time. This allows a small physical network to seem greater by having traffic that was recorded from a much larger network.

Future adaptations of the test bed might employ the use of a network switch in place of the hub. A switch would allow simulation of multiple sub-networks using data routing between VLANS (virtual local area networks).

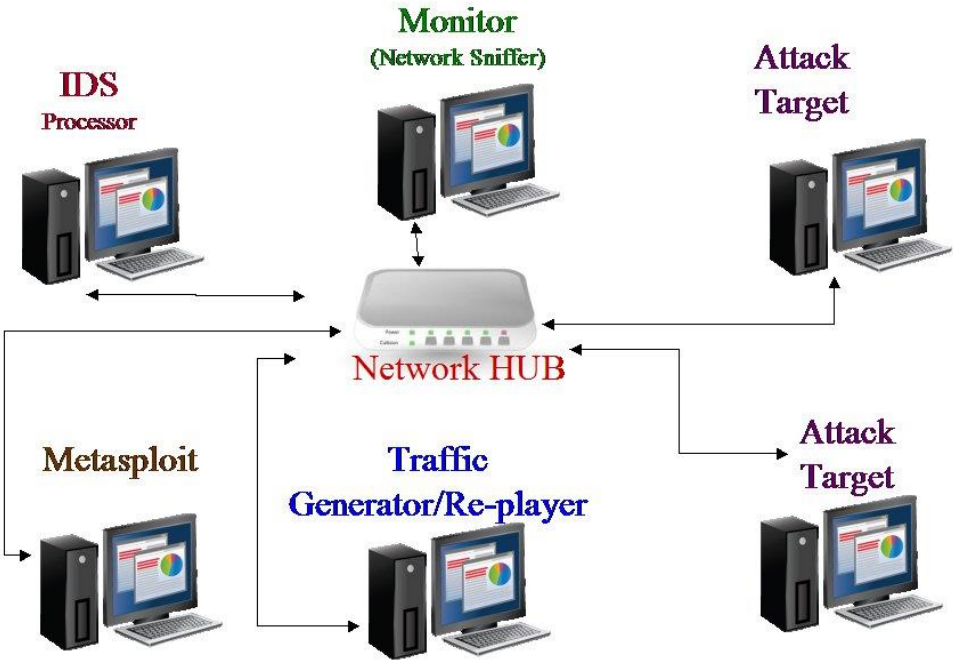

Figure 4 - Isolated Ted Bed

METASPLOIT

Metasploit Framework is an open source vulnerability-testing tool. Metasploit aids security and IT professionals in understanding the form and nature of all kinds of computer and network attacks. New information is shared with the community in order to highlight and promote best practices. Open-source means the underlying programming code of Metasploit is shared and updated through a community process.

The package provides a set of tools designed to probe systems on a network for vulnerabilities. The Framework software allows a user to configure over 1,100 included “exploits” for Windows, Unix/Linux and Mac OS X systems. Probing a target machine for susceptible entry points is often the starting point of an attack. Metasploit includes mechanisms to check a computer’s susceptibility to several known software exploits.

An exploit is an amount of data or a sequence of commands a person(s) can use for an attacker’s advantage. It usually takes advantage of software bugs or vulnerabilities that exists in computers or embedded devices. A person uses an exploit to gain unauthorized access to computers or systems.

EXPERIMENT

Six windows PCs and a network hub were networked to produce an isolated environment. Network cabling and operational procedures kept this environment from being connected to the larger Morgan State University network. Several open-source software packages were added to allow network attacks to be unleashed on target systems. Network traffic generators and sniffers were used to simulate and analyze different scenarios. Once the environment was set up, Metasploit Framework was a main software package used to learn intrusion techniques.

RESULTS

An isolated network of PCs, with the selected software, successfully supports the simulation of network and system intrusion. Metasploit was used to take advantage of the print spooler vulnerability on a Windows XP computer.

On a Windows system, each print server has an associated print spooler service. That service manages all print jobs and queues for print server. In September 2010, Microsoft issued a critical security bulletin, named MS10-061. That bulletin was entitled "Vulnerability in Print Spooler Service Could Allow Remote Code Execution." The bulletin describes how a remote attacker could use a specially coded print request to insert instructions for execution.

Using Metasploit, the print spooler vulnerability was exploited by inserting a Windows DLL. The DLL enabled a command interface between the attacking machine and the target computer. The initial connection established a unique windows process that could be viewed by the user as a long process name in Windows Task Manager. However, Metasploit session commands enabled the process to be migrated (and hidden from the user) to the explorer.exe process. This exploitation allowed total control of the PC. Additional executable files could be downloaded and launched. Complete access to the computer’s file system was granted. A user’s keystrokes could be monitored with an included key logger. Also, a GUI mirror of the target computer’s desktop with mouse control could be installed. Finally, the environment of the attacked computer was monitored by surreptitiously turning on the default microphone. The target computer packaged recordings in .wav files and sent them over the network to the attacking computer. The same process could be used with a default web cam, though this was not a part of our testing.

CONCLUSION

This work serves as a foundational effort for future cyber security and intrusion detection research to be performed in the Wireless Networking and Security Lab. Metasploit is an effective tool to aid ongoing intrusion detection research. This tool can be used to capture data streams of network and system intrusions. Analysis of those data streams will yield TCP/IP fields and parameters that define the attack. Results of that analysis can be used to create a representative vector for use in the Hidden Markov Model, which forms the basis of the WiNetS Intrusion detection model.

ACKNOWLEDGEMENTS

Research Advisor: Dr. Richard Dean
The author appreciates the support of the DoD/TRMC SRC and CRC for their support of this work

REFERENCES

[1] Stallings, William. *Data and Computer Communications,* 5th Edition. Upper Saddle River, NJ: Prentice Hall, Inc., 1997.

[2] Stallings, William. *Network Security Essentials, 2nd Edition*. Upper Saddle River, NJ. Pearson Education, Inc., 2003.

[3]] Odesanmi, Abiola & Daryl Moten. "Secure Telemetry: Attacks and Counter Measures on iNet". 7/2011.

[4] Doyle, Jeff. *Routing TCP/IP,* Volume 1. Indianapolis, IN. Cisco Press, 1998

[5] Doyle, Jeff and Jennefer DeHaven Carroll. *Routing TCP/IP,* Volume 2. Indianapolis,IN. Cisco Press, 2001

[6] Dukes, Renaea. INET Network Security Architecture. Morgan State University. December 2009..

[7] Lieberman, Joseph I. *"The Dangers of Delaying Heightened Cybersecurity."* Internet: http://www.washingtonpost.com/opinions/joseph-lieberman-the-dangers-of-delaying-heightened-cybersecurity/2012/10/09/b2c0621e-0cc6-11e2-bd1a-b868e65d57eb_story.html, Oct 9, 2012 [Nov 15, 2012].

[8] Microsoft Security Bulletin. *"Vulnerability in Print Spooler Service Could Allow Remote Code Execution."* Internet: http://technet.microsoft.com/en-us/security/bulletin/MS10-061, Sept 29, 2010 [Jun 17, 2013].