A State Transition Model Case Study for Intrusion Detection Systems

Brian J. d'Auriol
Department of Computer Science
The University of Texas at El Paso
El Paso, Texas 79968
dauriol@acm.org

Kishore Surapaneni
Department of Computer Science
The University of Texas at El Paso
El Paso, Texas 79968
ksurapaneni@utep.edu

Abstract ***A state transition model consisting of a physical layer, a communications sequencing layer and a state transition layer, is proposed for intrusion detection systems. A case study demonstrating the applicability of this approach is described. The case study concentrates on a port scan: states and state transitions based on the underlying layers are detailed. A series of screen shot captures illustrate identification and detection of specific intrusive activities of the port scan. The case study concludes by associating these observable conditions with states and state transitions. Lastly, we outline the applicable use of the proposed model.***

Keywords: Intrusion detection systems, Computer security, Network security

1 Introduction

Many organizations around the world deploy security devices and systems to protect their private networks from attacks delivered over the Internet. An intrusion detection system is one type of security management system used to detect intrusive activity and alert a system administrator of such activity. Intrusion detection systems monitor network and computer resource access and use and may analyze such information for signs of intrusion.

Intrusion detection systems broadly can be host-based and/or network-based [1]. Host-based intrusion detection systems monitor and detect attacks on a particular host computer, typically, by monitoring event, system and security logs. Network-based intrusion detection systems, however, monitor and detect suspicious communication traffic on the network attached to the internal private computing resources. Typically, these systems include packet-sniffers that intercept TCP or UDP packets on the network. The captured packets are then analyzed for intrusive activity. Network-node intrusion detection systems combine the properties of host-based and network-based systems. These hybrids have the intrusion detection system installed on each computer that is to be protected and monitors the traffic into and out of that particular host.

Operationally, intrusion detection systems can be broadly classified into anomaly detection or misuse detection [2]. In anomaly detection, attack behaviors are assumed different from 'normal' behavior. One advantage is it can detect previously unknown or novel attacks by comparing the monitored activity with that which has been defined as normal. Disadvantages of this model include a high false positive rate as well as difficulty to deploy the system in highly dynamic environments. The misuse detection model uses attack descriptions called signatures stored in a signature database. A signature defines a malicious or unwanted behavior considered to be an intrusion. Audit data streams generated in audit logs are matched against these signatures, most often in real time, and an intrusion is indicated if a match is found. An advantage of this type of system is its low false positive rate. However, disadvantages include the inability to detect novel attacks and the necessity of updating the signature database with new signatures.

Reported security incidents have grown rapidly during the past decade: from 1334 in 1993 to 137,529 in 2003; almost 43% of all incidents between 1988 and 2003 were reported in 2003 [3]. The number of attacks may well be greater due to the fact that there may be multiple attacks classified under a single incident [4]. In addition, the Internet continues to grow rapidly, approximately doubling each year [5]. At the same time, many key information and business resources have been migrated to the Internet thereby exposing sensitive information and operational corporate content [6].

These factors, amongst others, strongly motivate research into detecting, identifying, tracking and ultimately, responding to, attacks. Intrusion detection systems, in general, provide for one way in which to accomplish this.

An overview of a proposed state transition model for intrusion detection is given in this paper. The benefit of this model is the construction of an attack profile consisting of the components involved in the physical delivery, the involved computation and communication processes and the states and state transitions inherent in the attack. This model is deployed in an experimental prototype environment to detect probe scan activity. This paper reports on the success of associating the detection of intrusive activity with states in the state model. In turn, and when combined with a suitable attack scenario profile, this would allow us to potentially determine the operative states of an attack-in-progress. To this end, we briefly detail our approach to constructing state transitions of attack scenarios. Our approach in this work is based on characterizing intrusion attacks into sequences of computation and communication processes.

This paper is organized as follows. Section 2 presents an overview of the state transition modeling that we propose to characterize intrusive network behavior. The primary contribution of this paper is the experimental deployment of this model and is described in Section 3. Conclusions are given in Section 4.

2 State Transition Model Overview

The state transition model proposed in this paper is composed of three layers: the physical layer, the communication sequencing layer and the state transition layer. The physical layer defines the components and devices necessary for an attack and includes such things as source/victim computers, software, ports, routers and in general, the Internet. The physical layer appears in Figure 1: the source and victim computers are represented, respectively, by the left-hand side and the right-hand side; the communication network is implied, never-the-less, evidence of communications is shown by the edges between the source and victim; and the double, inverted triangles represent network sensors (packet sniffers) connected to the network. The communication sequencing layer provides abstraction of all the computation and communication processes that are executed through an attack scenario. In the figure, computation processes are graphically represented as dashed boxes labeled CP1–CP4 and communication processes are shown the middle and are labeled CM1–CM4. The state transition layer describes the attack profile in terms of states and transitions between the states. In the figure, states are shown by the circles and state transitions by the arrows. The start state is the inactive state on the source machine. There are three final states. In the following, we detail how a port scan can be profiled.

A port scan has two steps. In the first, the port scanning software, executing on the source machine, tries to establish a connection to a specified port on the victim's machine by sending a TCP SYN signal. In the second step, if the port is active, the victim's machine responds with a SYN/ACK signal, otherwise, an RST signal. Since ports 0 to 1024 are associated with pre-assigned processes or services, probing all ports on a victim's machine in this manner will give the source machine information about available services and systems. This vulnerability information is used as inputs to a decision process as to whether to launch an attack on victim's machine or not.

In the first step, the computation process CP1 represents a probing process running on the source machine while CP2 represents a service associated with a TCP port on the victim's machine. CP1 initiates the communication process CM1 that sends and carries a TCP SYN message to CP2. In the second step, if CP2 is active, then a SYN/ACK message is returned to CP1; however, if not active, then a RST message is sent to CP1. The victim's machine initiates the communication process CM2 whose task is to carry the response message. Based upon CM2, CP1 extracts the victim's machine vulnerability information.

In Figure 1, the lower portion that includes CP3 and CP4 together with CM3 and CM4 models an actual attack subsequent to the decision (see state 'Decision' in the figure) to proceed with an attack. The objective of this paper does not include further discussion concerning actual attack profiling, however, we include this part of the figure to show the 'big' picture. In [7], similar profiles for a UDP port scan, Denial of Service attack and a buffer overflow attack have been constructed.

3 Experimental Implementation

The goals of this experiment are two-fold: first, to simulate an attack environment primarily consisting of a port scan on TCP port 445, and second,

to verify the observable conditions that relate to certain aspects of the state transition model described earlier. As part of the second goal, an implementation of the state transition model specific to a port scan attack is developed. (We refer the reader to www.cert.org and www.microsoft.com: e.g. see CERT Advisory CA-2003-23, CERT Vulnerability Note VU#693099 and Microsoft Security Bulletin MS03-049 for further information regarding port 445 sucurity issues.)

The configuration of both the source and victim machines is an Intel Pentium 4 processor, 2.4 GHz, 512 MB of RAM and running the Windows XP operating system. The source machine is installed with the remote port scan software LANguard [8] to scan the victim's ports. The victim's machine is installed with network sniffing software Ethereal [9] which scans packets that are entering and exiting the victim's machine on different ports. The victim's machine is also installed with Active Ports [10] which is an internal port scanner.

The methodology is as follows. A port scan attack from a source machine to a victim's machine is constructed. We identify particular activities, communications, or other detectable information that indicate the presence of such an attack. To fully accomplish the second goal, we include both the source and victim in our study. Next, we associate the gathered information with specific states in the state transition model. The results of this experiment, therefore, are the observations that are made during the intrusive activities. We conclude the experiment by describing the state changes that lead to various parts of the gathered information.

Figure 1 represents the simulation states of the state transmission model. In particular, it shows the particular states that we select for monitoring. In this figure, the selected states of the source machine labeled with letters S1, S2, ..., S6. Similarly, the selected states of the victim's machine are labeled with the letters V1, V2, ...,V5. Screen shots of selected observable behavior and information follow; each is annotated with the label of the associated state. The following subsections detail an analysis of the association of observable information to states in the state transition model, respectively from the point of view of the source and the victim. In all screen shots, the source IP address has been replaced by SSS.SSS.SSS.SSS and the victim's IP address has been replaced by VVV.VVV.VVV.VVV for both clarity of presentation and IP address privacy.

3.1 Source Machine

Initially, the source machine is switched off and as a result, it is in the start state, 'Inactive'. When the source machine is switched on it goes to the 'Active' state. When the LANguard program on the source machine is initiated, the source goes to the 'Probe Active' state represented by Figure 2 which is related to state label S1 in Figure 1. The attacker, then, inputs the victim's IP address of to LANguard along with the port type and port number (TCP port 445) to be scanned; as a result, the source enters the 'Probing' state represented by Figure 3. Initially, the victim's machine is switched off in this part of the experiment before this probe attack is launched to realize the 'Probe Fail' state on source. Figure 4 represents the 'Probe Rcvd' and 'Probe Fail' states which indicate that the victim's machine is turned off and is not active. This is related to state labels S3, S4 in Figure 1. Now, the victim's machine is turned on and the above TCP port 445 scan actions are repeated. Figure 5 represents the 'Probe Rcvd' and 'Probe Successful' states labeled as S3 and S5 in Figure 1; this indicates that the victim has been successfully probed and the probe information has been received. The source machine then goes to the 'Decision' state based upon the probe information that has been received from victim's machine. The LANguard window in Figure 6 shows that TCP port 445 on the victim's machine is active and hence could be abused; it is labeled as S6 in Figure 1.

The purpose of this part of the analysis is to confirm that the attack scenario that we have developed for a port scan attack indeed matches realistic implementation observations.

3.2 Victim's Machine

Initially, the victim's machine is either in the switched off state 'Inactive' or the switched on state 'Active'. Figure 7 shows the captured network packets when the victim's machine is in the 'Active' state and a probe attack occurs. Here, the figure represents the 'Probing' State and shows the arrival of a TCP SYN scan signal from the source to the victim in the Ethereal window (executing on the victim's machine). It is related to state label V2 in Figure 1. The Active Ports window in Figure 8 shows that the TCP port 445 has been probed and hence represents the 'Probed' state and is related to state label V3 of Figure 1. After the TCP port 445 has been probed, the results are sent back to source machine leading to the 'Probe Snt' state; this is verified in Figure 9 in which the Ethe-

real Window shows the transmission of the probe results from the victim to the source. Since TCP port 445 on the victim's machine is open, it could be abused and is vulnerable which leads the victim to the 'Insecured' state. This state is verified in Figure 10 in which the Active Ports window shows TCP port 445 on the victim's machine open.

The purpose of this part of the analysis is twofold: first, as with the previous, to confirm that the attack scenario for the port scan attack can be reasonably observed, second, to support our approach in detection and tracking of states associated with the attack profile.

4 Conclusion

This paper demonstrates through a case study the applicability of a state transition model approach for modeling both network and host information pertaining to intrusive activities. The proposed approach requires profiling attack activities as states and state transitions as well as profiling victim responses also as states and state transitions. As such, the proposed approach is suitable for deployment in an intrusion detection system. We have demonstrated this approach for a port scan.

The broader impact of our approach is that with an integrated state transition model for both the attack and victim activity sequences, both forward and reverse prediction is enabled. Consider: in a realistic deployment, the intrusion detection system could detect suspicious activity at some point in the attack sequence, and more likely, at some point after an attack has already been launched. Our approach allows for matching the detected suspicious activity to victim states; thereby allowing determination of the state transitions both forward (in the future) and reverse (into the past). The intrusion detection system could then scan past network or host logs for evidence of specific activities that would match with past states; and heighten scanning for expected attack vectors.

The state transition model proposed in this paper is limited both by the simplicity of states as well as its applicability to a few relatively simple intrusive activities. A much greater effort is required in profiling so as to determine greater number of states, and thereby, increase the likelihood of successful detection. Also, work in modeling more complex and realistic attacks is needed.

References

[1] A. Sarmah, "Intrusion detection systems; definition, need and challenges," October 3, 2001. SANS Institute, http://www.sans.org.

[2] R. Anderson, *Security Engineering, A Guide to Building Dependable Distributed Systems*. John Wiley & Sons, 2001.

[3] Computer Emergency Response Team Coordination Center, "CERT/CC Statistics 1988-2003," January 22 2004. http://www.cert.org/stats/.

[4] J. D. Howard, *An Analysis Of Security Incidents On The Internet*. PhD thesis, Carnegie Mellon University, Pittsburgh, Pennsylvania 15213, USA, April 1997.

[5] A. M. Odlyzko, "Internet traffic growth: Sources and implications,," in *Optical Transmission Systems and Equipment for WDM Networking II* (B. B. Dingel, W. Weiershausen, A. K. Dutta, , and K.-I. Sato, eds.), pp. 1–15, 2003. Proc. SPIE, vol. 5247.

[6] J. S. Sherif and T. G. Dearmond, "Intrusion detection: Systems and models," in *proc. of the Eleventh IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE'02)*, pp. 1–19, 2002.

[7] K. Surapaneni, "Intrusion detection: Computation communication characterization of probing and network attacks," Master's thesis, Department of Computer Science, The University of Texas at El Paso, May 2004.

[8] GFI Security And Messaging Software, "GFI LANguard Network Security Scanner Overview," 2003. http://www.gfi.com/lannetscan.

[9] Ethereal, "Ethereal description," December 28, 2003. http://www.ethereal.com.

[10] SmartLine, "Freeware for windows: Active ports," 2003. http://www.ntutility.com/freeware.html.

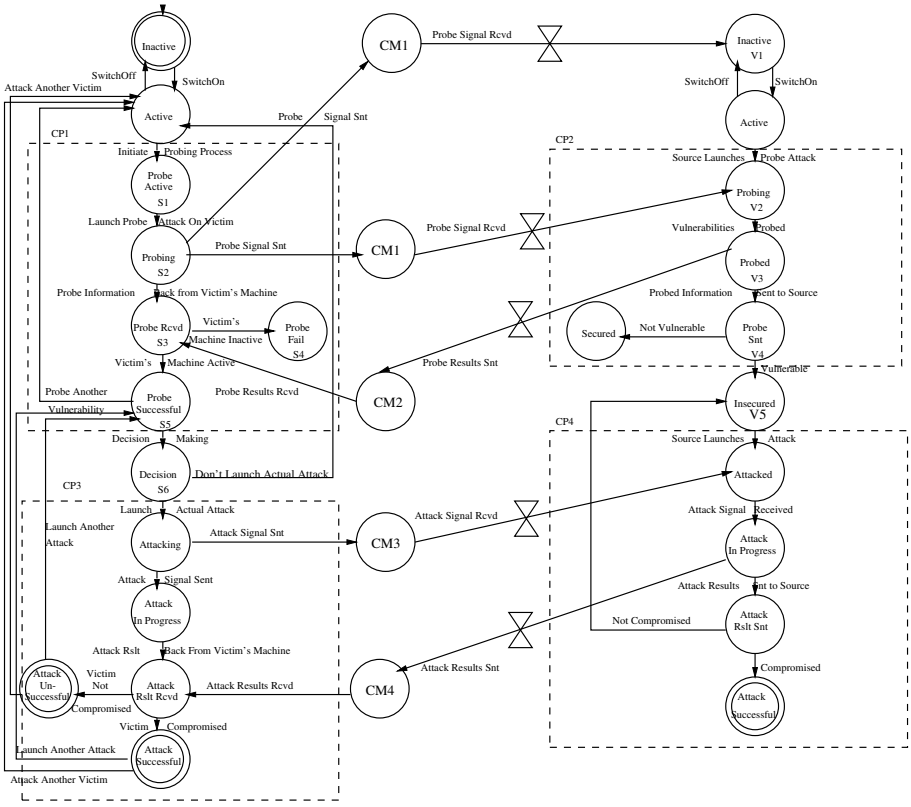

Figure 1: State Transmission Model

Figure 2: Probe Active State: S1

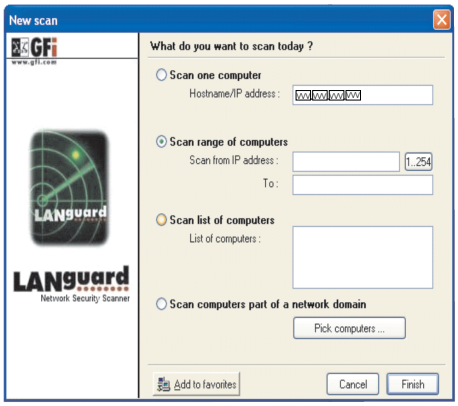

Figure 3: Probing State: S2

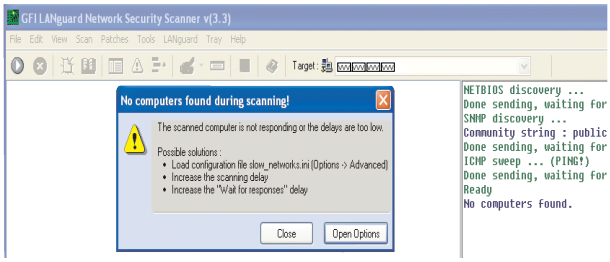

Figure 4: Probe Rcvd and Probe Fail States: S3, S4

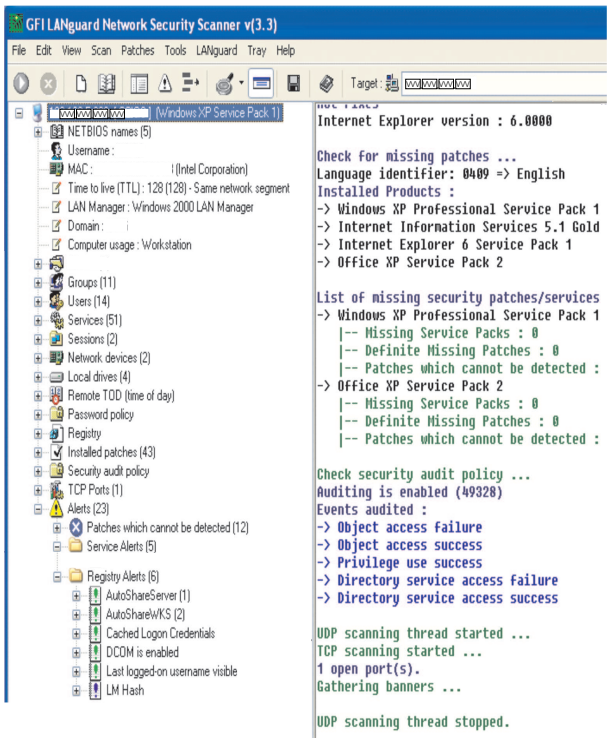

Figure 5: Probe Rcvd and Probe Successful States: S3, S5

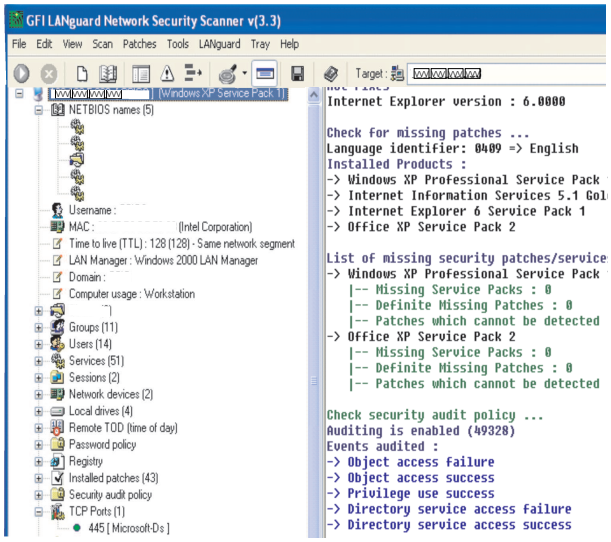

Figure 6: Decision State: S6

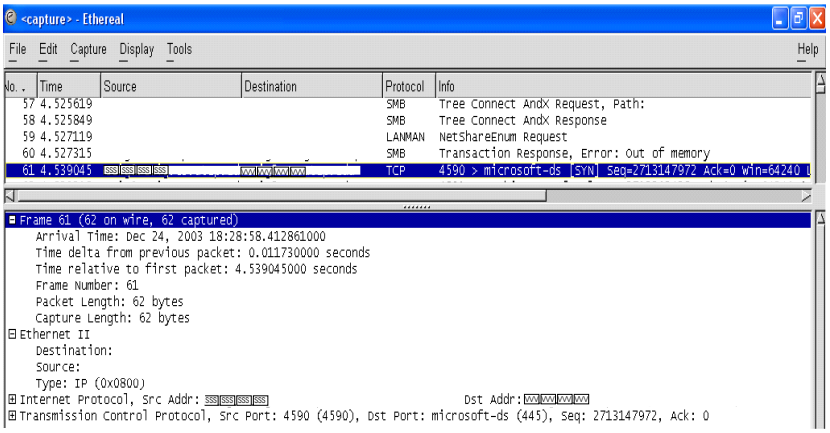

Figure 7: Probing State: V2

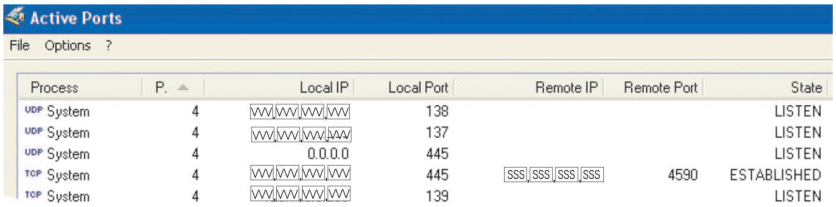

Figure 8: Probed State: V3

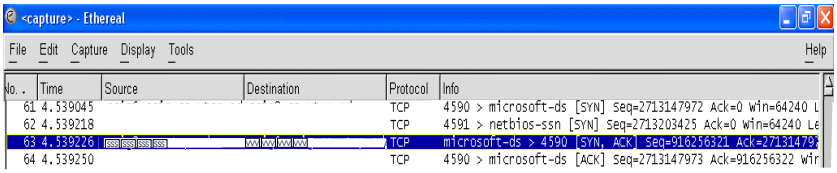

Figure 9: Probe Snt State: V4

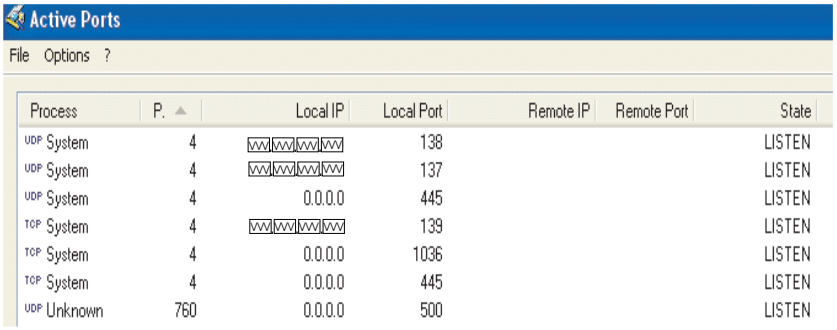

Figure 10: Insecured State: V5