

Network Vertical Intrusion Model (NetVIM)

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

Abstract—*The Network Vertical Intrusion Model (NetVIM) is proposed in this paper. The NetVIM is a vertical four layer model that provides for the identification, detection and profiling of network based attacks. The NetVIM models attack profiles consisting of the components involved in the physical delivery, the involved computation and communication processes and the states and state transitions inherent in an attack. This paper describes the NetVIM together with an example application to a TCP port scan attack.*

Keywords: Intrusion detection, Computer security.

I. INTRODUCTION

The Network Vertical Intrusion Model (NetVIM) is proposed in this paper. NetVIM is a vertical four layer model that provides for the identification, detection and profiling of network based attacks. The NetVIM models attack profiles consisting of the components involved in the physical delivery, the involved computation and communication processes and the states and state transitions inherent in an attack. Attack profiles allow for both forward and reverse prediction of attacks. Consider a system based on NetVIM: this system would detect suspicious activity at some point in an attack sequence, but likely after the attack has already been launched. The NetVIM system would match the detected suspicious activity to states in the attack profile, thereby also determining attack states that have already occurred, or likely to occur. The 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 future attacks. The NetVIM is presented in Section II. Conclusions are given in Section III.

II. NETWORK VERTICAL INTRUSION MODEL

Many network-based attacks go undetected. This raises two questions. How to detect suspicious activity? If suspicious activity is detected, how to determine the scope of the attack? Anomaly and/or signature-based detection methods used on host-based or network-based

intrusion detection systems have commonly been used to detect suspicious activity (surveys of IDS include [1], [2]). Characterization of network attacks is one area of much active research [3]–[5]. This addresses the first question. NetVIM specifically addresses the second question. The detection of a single suspicious event raises doubts, thereby suggesting closer monitoring of ‘likely events’ whether in and by themselves suspicious or not. Single events that ordinarily would not flag intrusive activity could then be included into a collection or sequence of suspicious activities. An emergent description of an intrusion is then characterized by such a sequence of events.

The primary goal of NetVIM is to model network-based intrusions so as to be able to identify emergent intrusive behavior through correlating the monitoring of single network or computer usage events with collections and sequences of events that pertain to specific intrusions. Two specific sub-goals to accomplish this are: first, the development of a model framework that consists of low-level event monitoring, middle-level event collection and sequencing, and high-level intrusion representation; second, the development of multi-layer intrusion and attack profiles for specific classes and instances of attacks, e.g., ports scans and denial of service. The following subsections describe NetVIM in the context of these two sub-goals.

A. Layered Model

The NetVIM is composed of four layers as described below and in Figure 1.

a) Physical Layer: The physical layer is the lowest layer. It defines the components and devices necessary for an attack, for example, the source/victim computers, software and routers. Let M denote a set of observations m_k about network or host monitored suspicious activity.

b) Computation Communication Sequencing Layer: This layer abstracts the computation and communication processes, including the dependencies

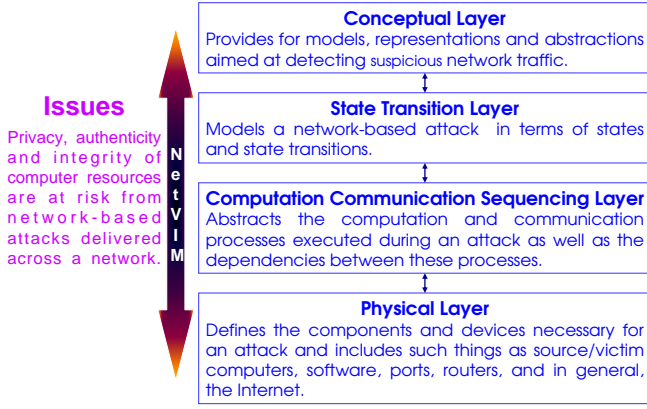


Fig. 1. NetVIM Layers

between those processes, that execute during an attack. A graph-based approach is used: $G = (V, E)$ where $V = \{CP_i, CM_j\}$ for a computation process CP and a communication process CM , $e \in E$ connects a CP_i with a CM_j . This abstraction is used in attack profiling to construct a chronological set of processes that occur during a network attack.

c) *State Translation Layer:* The state transition layer models a network-based attack in terms of states and state transitions. Let $S = (H = (A, I), B, F)$ denote a state transition model where H denotes a state transition graph with the set of states, A , and the set of state transitions, I , together with B , the set of start states and F , the set of terminal states. There is a single start state corresponding with the initiation of an attack by an attacker. There are a number of terminal states corresponding with a successful or unsuccessful attack. S is determined by attack profiling. This layer provides the hooks needed to integrate the other layers.

d) *Conceptual Representation Layer:* The conceptual representation layer is the highest layer. Currently, a superellipsoid-enhanced Conceptual Space model is used to represent, store and analyze suspicious activities [6]. This model is parameterized by spatial variables, form and scaling factors.

B. Layer Integration

The NetVIM is layered in order to connect the information modeled within each layer. At the extremes, physical components are used for delivery whereas representation and identification of the attack is performed by the Conceptual Representation Layer. In-between layers enhance the detection process as follows. First, attack profiles are constructed that describe the states involved in the attack together with the computational

and communication processes that are required to convey the attack. Consequently, a suite of attack profiles is maintained in a library and made available during the subsequent run-time operations. Second, activity sensed at the physical level is matched to specific locations in the sequencing layer of the profile. Third, the tight integration between the sequencing layer and the state layer allows for extrapolation of states that associate with the computation and/or communication processes now identified. Fourth, the state transition graph provides information regarding states that precede the monitored activity and those that follow the activity. Fifth, for preceding states, previously monitored activity data could be reviewed in the context of the knowledge made available by the attack profile that has been matched. For states that follow the monitored activity, increased vigilance can be attained, again, in the context of the attack profile. The idea therefore is that a single monitored event in and by itself is not likely to indicate an attack, however, a collection of events which correspond to an attack profile is likely to indicate an attack. A conceptual representation is constructed as the emerging set of suspicious activities are clarified.

The various layers integrate as follows.

- The Physical layer and Computation Communication Sequencing layer: Let $m \in M$ be observed by some network or host-based sensor and identify a corresponding $v \in V$ or an $e \in E$, that is, the computation process or communication process associated with the monitored activity is identified. For host-based detection, this would usually associate m with a CP whereas for network-based detection, m with a CM .
- The Computation Communication Sequencing layer and the State Transition layer: Let $\bar{A} \subseteq A$ be identified with $v \in V$ and let $\bar{I} \subseteq I$ be identified with $e \in E$, that is, a computation process or a communication process may have one or more states associated with it.
- The Physical layer and the Conceptual Representation layer: Let m be associated with a spatial dimension in a conceptual space.
- The State Transition layer and the Conceptual Representation layer: Let S be associated with the conceptual space such that the form and scaling factors are interpreted (for example, by expert human intervention) based on S and whose quality dimensions are given by M .

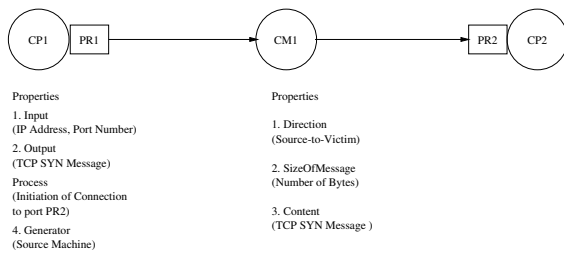


Fig. 2. TCP Probe Scan Step 1

C. Attack Profiling

The two main purposes of profiling are: (a) to understand the processes involved, and (b) to provide the basis for incorporation into the vertical intrusion detection model.

This section describes in overview an attack profile for a TCP port-scan attack as an example of the methods involved in characterizing intrusions and attacks. Details concerning the computation-communication sequencing are presented in [7] while details about the state-based modeling appear in [8].

A port scan has two steps. First, a connection to a specified port on the victim’s machine is attempted by sending a TCP SYN signal. Second, the victim’s machine responds with a SYN/ACK signal if the port is active, otherwise, an RST signal. Probing all ports on a victim’s machine in this manner informs the attacker about vulnerable services.

In the first step, the computation process CP_1 represents a probing process running on the source machine while CP_2 represents a service associated with a TCP port on the victim’s machine. CP_1 initiates the communication process CM_1 that sends and carries a TCP SYN message to CP_2 . Figure 2 illustrates this step. Next, CP_2 initiates CM_2 whose task is to carry the response message. Based upon CM_2 , CP_1 extracts the victim’s machine vulnerability information. Hence: $CP_1 \rightarrow CM_1 \rightarrow CP_2$, and $CP_2 \rightarrow CM_2 \rightarrow CP_1$.

The states in Figure 3 are shown as circles and the state transitions, as arrows. The starting state of the state machine is the ‘Inactive’ state for the source and corresponds with the attacker’s computer switched off. The final states of the source machine are either the ‘Attack Successful’ or the ‘Attack Unsuccessful’ state. The attacker initiates the probing process which results in the ‘Probe Active’ state, next, the attacker enters the ‘Probing’ state. The probe results of the victim’s machine are sent back to the attacker’s process, as a result, the source changes to the ‘Probe Rcvd’ state. The

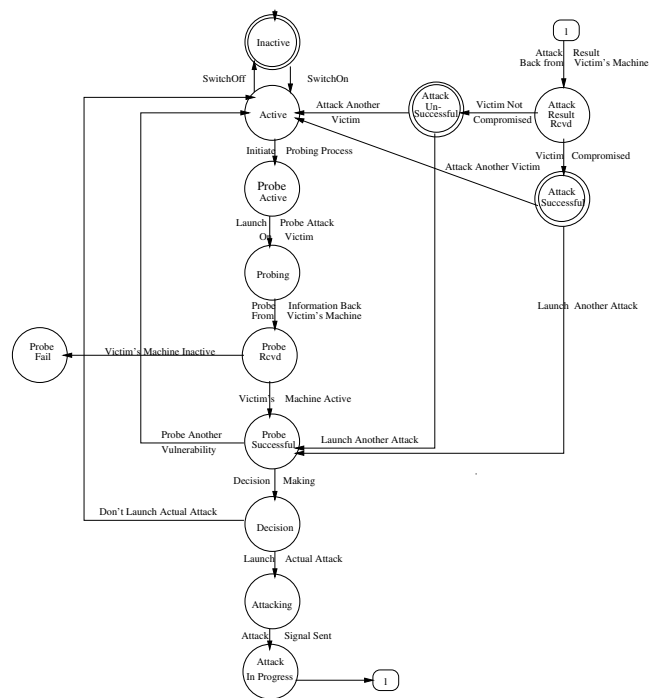


Fig. 3. State transition model for TCP Probe Scan

source goes to the ‘Probe Fail’ state if the victim’s state was ‘Inactive’, otherwise, the source goes to the ‘Probe Successful’ state. The attacker enters the ‘Decision’ state to launch the actual attack or not.

The integration of the lower three layers for the TCP port scan example is shown in Figure 4. In the figure, the source and victim computers are represented, respectively, by the left-hand side and the right-hand side; the communication network is implied, nevertheless, evidence of communications is shown by the (state transition) edges between the source and victim; and the double, inverted triangles represent network sensors (packet sniffers) connected to the network. The computation communication sequencing layer is graphically represented in three parts: first the computation processes are shown as dashed boxes labeled CP_1 – CP_2 , second, the communication processes are shown in the middle and are labeled CM_1 – CM_2 , and third, the sequencing is implied by the state changes. A complete diagram including an attack profile appears in [8]

Consider the network sensor identified with the label ‘1’ (highlighted in light gray). This monitors the network activity between the victim and the attacker during the change of state from the Victim’s ‘Probed’ state to the CM_2 state ‘probe Results snt’. Assume that this sensor detects a suspicious network packet. The model can extrapolate prior states and processes based on the se-

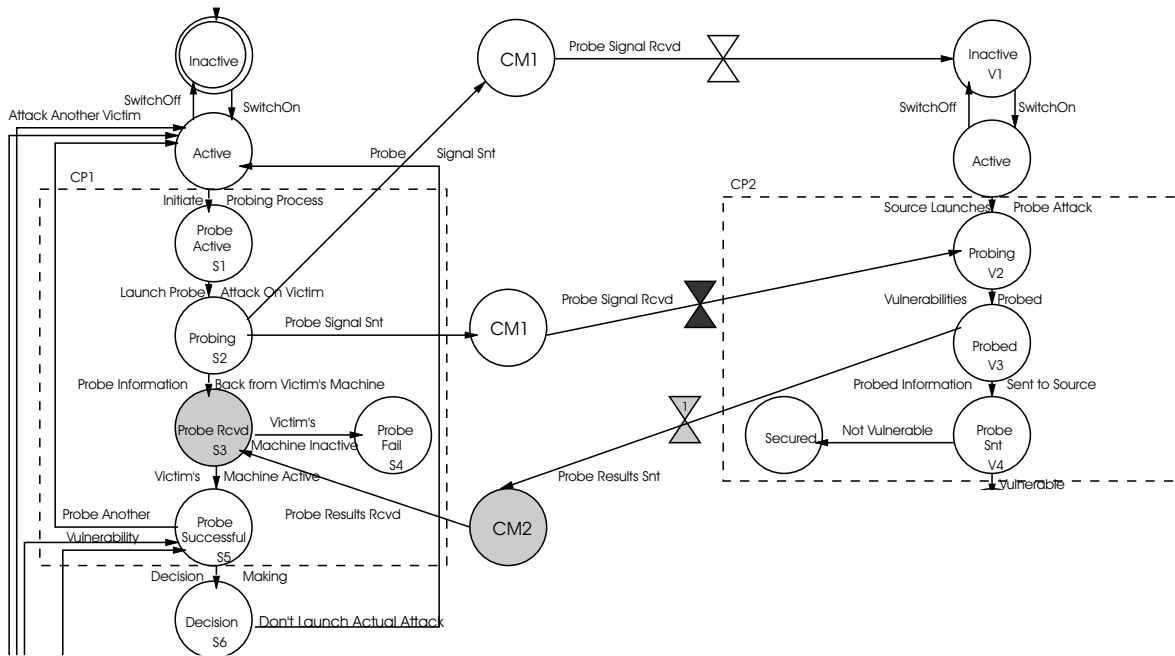


Fig. 4. Example of an integrated three-layer profile.

quencing and state transitions: for example, the ‘Probing’ attacker state has already occurred; and, CP1→CM1 and CM1→CP2, has already occurred. The network sensor, shown in dark gray, that monitors events leading to the state transition from the ‘Prob Signal Rcvd’ state to the victim’s ‘Probing’ state should have been able to detect suspicious events pertaining to this intrusion. In this example, it did not. The proposed system would either alert system administrators or, if so designed, would be able to review audit logs from this sensor for such evidence. In a similar way, the proposed model could also be able to predict suspicious activity.

III. CONCLUSIONS

The Network Vertical Intrusion Model (NetVIM) allows for the identification of emergent intrusive behavior via its prediction-based capability. The emergent behavior is based on monitoring a single possible suspicious event, determining associated states in the attack profiles, determining prior and post possible states based on the possible computation communication sequences, determining the network or host sensors needed to monitor prior and post possible events, and finally, reviewing post events, or heightening monitoring of future events. Any subsequent event so identified as suspicious is added to the growing collection of suspicious events, thereby generating an emerging presence of the attack. The NetVIM is described in brief in this paper.

REFERENCES

- [1] 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)*, 2002, pp. 1–19.
- [2] J. Allen, A. Christie, W. Fithen, J. McHugh, J. Pickel, and E. Stoner, “State of the practice of intrusion detection technologies,” Pittsburgh, PA 15213-3890, Tech. Rep. CMU/SEI-99-TR-028, Jan. 2000.
- [3] P. Barford and D. Plonka, “Characteristics of network traffic flow anomalies,” in *ACM SIGCOMM Internet Measurement Workshop 2001*, San Francisco, Nov. 2001.
- [4] C. B. Lee, C. Roedel, and E. Silenok, “Detection and characterization of port scan attacks (manuscript, believed 2002),” <http://www.cse.ucsd.edu/~esilenok/portscans.html>, University of California, San Diego.
- [5] S. Zheng, C. Peng, X. Ying, and X. Ke, “A network state based intrusion detection model,” in *Proc. of the 2001 International Conference on Computer Networks and Mobile Computing*, Los Alamitos, CA USA, Oct. 2001, pp. 481–486.
- [6] B. J. d’Auriol and A. Akinsanmi, “A conceptual space model for intrusion detection,” in *Proc. of The 2005 International Conference on Security and Management (SAM’05)*, Monte Carlo Resort, Las Vegas, Nevada, USA, June 2005, in press.
- [7] B. J. d’Auriol and K. Surapaneni, “A computation-communication sequencing model for intrusion detection systems,” in *Proc. of The 2005 International Conference on Security and Management (SAM’05)*, Monte Carlo Resort, Las Vegas, Nevada, USA, June 2005, in press.
- [8] —, “A state transition model case study for intrusion detection systems,” in *Proc. of the 2004 International Conference on Security and Management (SAM’04)*, H. R. Arabnia, S. Aissi, and Y. Mun, Eds., Monte Carlo Resort, Las Vegas, Nevada, USA, June 2004, pp. 186–192.