Understanding Multistage Attacks by Attack-Track based Visualization of Heterogeneous Event Streams *

S. Mathew, R. Giomundo, S. Upadhyaya
Computer Science and Engineering
SUNY at Buffalo, Buffalo, NY 14260
{smathew2, giomundo, shambhu}@cse.buffalo.edu

M. Sudit, A. Stotz
Industrial Engineering
SUNY at Buffalo
Buffalo, NY 14260
{sudit, astotz}@eng.buffalo.edu

ABSTRACT

In this paper, we present a method of handling the visualization of hetereogeneous event traffic that is generated by intrusion detection sensors, log files and other event sources on a computer network from the point of view of detecting *multistage attack paths* that are of importance. We perform aggregation and correlation of these events based on their semantic content to generate *Attack Tracks* that are displayed to the analyst in real-time. Our tool, called the **Event Correlation for Cyber-Attack Recognition System (ECCARS)** enables the analyst to distinguish and separate an evolving multistage attack from the thousands of events generated on a network. We focus here on presenting the environment and framework for multistage attack detection using ECCARS along with screenshots that demonstrate its capabilities.

Categories and Subject Descriptors

H.4 [**Information Systems Applications**]: Miscellaneous

General Terms

Security

Keywords

attack tracks, intrusion detection, visualization

1. INTRODUCTION

Various techniques have been developed to efficiently display cyber event data to security analysts so that malicious actions and other threats can be detected and mitigated. Concurrently, focus in the information assurance community has changed from detecting simple attack steps to detecting compound attacks involving multiple stages. Detection of multistage attacks is non-trivial and is of importance specially in high-value networks. Most current cyber-security visualization efforts target presentation of network data [2], textual log files [3], intrusion alerts and other events [9] and netflow data [16]. We specifically target the detection and visualization of multistage attacks on computer networks.

Attack Graphs [7] have been proposed as a method to carry out vulnerability analysis of networks and to aid in the understanding and detection of multistage attacks. Some useful efforts for the visualization of attack graphs have also been presented in the literature ([13], [12]). However, the technique of attack graphs often requires extensive modeling of network resources and services which is time consuming and error prone. Attack Graphs have been shown to have quadratic complexity with exploit based representations under reasonable assumptions [1]. This exploit based representation has been used in [6], [13], [11] and others, but practical implementation is still problematic. Exhuastive enumeration of all possible exploits for different platforms is required and one needs to have *all* possible combinations of exploits represented in the graph; something that may not be feasible as this information requires experimental evaluation in many cases.

In this paper, we present a technique that provides the security analyst with real-time *multistage attack visualization*, while dealing with the problems mentioned above. We propose to detect multistage attacks solely using sensor events (IDS alerts, system log-files) as the basis for making hypotheses about multistage attacks. We utilize the attack stage oriented classification of events based on semantic content presented in [10]. Our contribution in this paper is the development of generic models of multistage attacks which are composed of the above attack stages, to guide the fusion process. These models (called Guidance Templates) are used along with IP address information to perform attack-stage based correlation of heterogeneous events. This process generates dynamic Attack Tracks which represent hypothesized multistage attacks present in the event stream. Our system uses simple interfaces augmented with effective visualization to provide real-time situation awareness of multistage attacks to the security analyst.

2. RELATED WORK

Several research efforts have recently tackled the issue of efficient visualization of computer security data. Erbacher

*Research supported in part by Alion Science and Technology subcontract F30602-03-C-0245 from ARDA and AFRL programs

VizSEC'06, November 3, 2006, Alexandria, Virginia, USA.

Table 1: Attack Categories of intrusion alerts and other events

Alert Class	Description
Recon_Sniffing	Reconnaisance step. Indicates that the attacker may be sniffing the channel. Motivation may be to tamper with network communications.
Recon_Footprinting	Reconnaisance step. Attacker gains knowledge of the target network or organization's security posture [14], e.g., identifying the organization's domain names.
Recon_Scanning	Reconnaisance step. Attacker tries to refine and verify the knowledge gained during the Footprinting phase [14]. E.g., Ping attacks. Sometimes reveals specific software details like versions. Can be used along with Footprinting to constitute a *Fingerprinting* attack. Non-intrusive.
Recon_Enumeration	Enumeration [14] is another Reconnaisance step. Usually employed after previous steps. Attacker tries to identify user accounts to exploit, poorly protected resources etc. Different from previous stages as it involves active connections to targets. Information gathering stage.
Intrusion_Root	Intrusive step into a target machine with the privileges of administrator. Attacker may have access to a command shell with the same privileges. May overlap with buffer overflow attacks classified here as *Escalation*, but includes attacks that may exploit configuration flaws.
Intrusion_User	Intrusion with privileges of non administrative user.
Escalation_OS	Privilege escalation step (usually buffer overflow attacks). Escalation exploites vulnerability in a specific operating system or in software usually bundled with a certain OS.
Escalation_Service	Privilege escalation step exploiting a vulnerability (usually a buffer overflow vulnerability) in a specific service or software package, rather than a specific OS vulnerability.
Goal_DoS	Alerts that indicate the possibility of Denial of Service attacks.
Goal_Ethical	Attacker's goal is purely *Ethical*. Indicated by observing that an attacker penetrates a system to the point where he can carry out malicious attacks at will, yet refrains from doing so.
Goal_Corruption	Attacker tries to corrupt a target machine, its configuration, and/or data. Clearly a hostile action and indicates an active goal-oriented adversary with malicious intent.
Goal_Espionage	Goal of the adversary is *Espionage*. This involves steps like trying to obtain password files, access keys and so on.
Goal_Backdoor	Attempt to install a backdoor on the target machine to facilitate future attacks.
Goal_Pilfering	Attacker's goal is to pilfer, steal, and/or exfiltrate data from the target machine.

et al. [2] develops techniques to handle *scalability*, *high dimensionality*, *complexity* and *temporality* of *simpcap* data by providing the analyst with multidimensional views displaying different aspects of the data. In [3] and [9], he presents techniques to display textual log-file and alert (event) data to the analyst. In [16], Yin et al. target the display of netflow data for providing situational awareness. Efforts presented in [15] and [4] are aimed at scalable visualization of large-scale network data to provide comprehension of threats (e.g., worms) and at preserving context during analysis of high-volume data. In [8], an IP matrix is used to achieve the above purpose. Techniques for analyzing complex attack graphs using recursive hierarchical aggregation and coordinated views are presented in [13] and [12].

These techniques are effective in representing high volume multidimensional data in a manner that aids analysis by security experts. However, these efforts do not explicitly display data in the context of evolving multistage attacks on a network. In sensitive operational environments like military and intelligence networks, unusually high volumes of attack activity are encountered. Most of this activity consists of ubiquitous pings and scans directed against the network. In such environments, detection of serious multistage attacks that can be the focus of analyst attention is the key requirement. This detection process requires further processing of the native outputs of IDS sensors and other event sources. In this paper, we present our approach to the realization of this goal.

3. EVENT CORRELATION, EXPERIMENTAL ENVIRONMENT AND VISUALIZATION FRAMEWORK

3.1 Guided Event Fusion

Although the events that are generated by sensors can be diverse, their number is generally finite. For sensors that perform *misuse detection*, the signature set consists of all events that the sensor can output; for anomaly detection based tools, the general types of events (e.g., *protocol anomalies*, *unusual port* etc.) are also well known from sensor documentation. Exploits can be combined in novel and unforeseen ways to carry out malicious attacks, hence exhaustive enumeration of all possible events is not feasible. Nevertheless, goal-oriented multistage attacks have well defined semantic stages - one broad outline could consist of the stages *Reconnaisance*, *Intrusion*, *Privilege Escalation*, and *Goal* [5]. We refine these steps to create a finite set of *Attack Categories* that broadly cover the semantics of any multistage attack. We map the entire signature set of every sensor and other event source to these Attack Categories. Event descriptions in the signature set for the sensor (e.g., Snort signature set available at *www.snort.org*) are used to perform the classification. An automated tool parses event descriptions looking for certain keywords indicating the Category membership of the event. As a result, every event that can be generated by a network event source belongs to one or more Attack Category. We thus claim comprehensive coverage of any multistage attack within the detection limits of the sensors deployed on the network. A description of the Attack Categories that we use is shown in Table 1. More details including examples and rationale for this classification are available in [10].

Our correlation and fusion framework is based on using generic models of multistage attacks called Guidance Templates. A Guidance Template is a model of successive stages each of which consists of one or more Attack Categories. Each stage has a weight which represents the *importance* of this stage in the context of the multistage attack model that is defined by the Guidance Template. The advantage of this approach is the flexibility that the analyst has in defining attack models that are characteristic of various kinds of threat - for example, an attack model geared towards detecting outside threats would weigh reconnaisance attacks more highly than a model targeting insider attacks, where presumably, the malicious insider already has knowledge of the location of targets. A stream consisting of events is fused to generate *Attack Tracks* which are dynamic representations of multistage attacks. An Attack Track is a sequential list of Attack Stages consisting of Attack Categories that is dynamically generated such that events in successive Stages are correlated on the basis of IP address (the events in Attack Categories in one Stage have source IP addresses which are the target addresses of events in the preceeding Stage). A dynamic *Criticality Score* is calculated for the tracks which reflects both the importance of the Attack Stages represented in the track and the degree of match with the Guidance Template. A set of Active Tracks is maintained and tracks become inactive after a period of time during which there are no correlations with new events in the stream. A new event is correlated as follows:

- The system tries to correlate the new event with existing active tracks; if possible, the event is grouped into the corresponding Attack Category in the Attack Track (a new Attack Category is created if necessary)
- If the event cannot be correlated with existing active tracks, and if the Attack Category represented by the event is sufficiently important, then a new Attack Track is created for the event and added to the list of active tracks

At any instant of time, the set of active Tracks together with the Criticality values which denote their importance constitute Situation Awareness indicators to the security analyst.

3.2 Experimental Environment

The system was set up to operate on data generated by sensors and log files on a network designed to simulate an unclassified military network called the Open Source Information System (OSIS). The configuration consists both of physical hosts and machines that simulate a number of virtual hosts. The OSIS network consists of a number of enclaves which host privileged users in addition to two external enclaves with limited privileges. Both Windows and Linux machines, web, mail and other servers are part of the network. The event traffic available to us consisted of both host and network alert traffic generated by normal activity and so-called background attacks (high-value networks, especially military ones, experience a large number of pings, portscans and other reconnaisance type attacks which add 'noise' to the event traffic) as well as by specially developed multistage attacks. The data available to us also documented 'ground truth' with respect to these attacks so that system performance in multistage attack detection could be quantified. This comprehensive dataset was provided to us by a federal research sponsor.

The event sources that provided inputs to the system are *Snort* and *Dragon* (a network sensor) intrusion detection sensors. The available Snort versions were 2.1.2, 2.3.2 and 2.4.1 each of which generates alerts consistent with its signature set. Log-files available comprised of IIS and Apache web logs, Windows *AppEvent*, *SecEvent* and *SysEvent* files, Unix log-files in addition to FTP and database log-files from servers on the network.

Several sophisticated multistage attacks were part of the test dataset. An example scenario (called *PNP with Phishing Exploit*) consisted of the stages - *1. An attacker sets up a 'porn' site with users providing their own username and password in a form*, *2. A user fills in the form with login information that could compromise his workstation*, *3. Attacker logs in via ssh to the victim's machine using the obtained username and password*, *4. Attacker downloads a PNP exploit executable*, *5. Attacker exploits a Windows host gaining a command shell*, *6. Attacker uploads all the files in the victim's home directory to a remote FTP site.*

An event stream consisting of events representing both normal activity and malicious attacks is provided as input to ECCARS. The system outputs a dynamically varying list of Attack Tracks along with indications as to which tracks it considers as malicious. The ground truth data (part of the dataset) identifies events corresponding to malicious attacks and allows us to analyze system performance. Fragmented attack tracks (due to events missed by sensors) can be handled by comparing them with the guidance template, although perfect attack comprehension may be difficult.

We use several metrics to evaluate ECCARS performance, most importantly,
$Precision = \frac{Number\ of\ True\ Positives}{Number\ of\ True\ Positives\ +\ No\ of\ False\ Positives}$
and
$Recall = \frac{No\ of\ True\ Positives}{No\ of\ True\ Positives\ +\ No\ of\ False\ Negatives}$.

3.3 User Interfaces and Visualization

The **FUsion Model Editor**, shown in Figure 1 is the interface through which the analyst can define a fusion model. A fusion model consists of a Guidance Template, definition of Attack Stages along with their constituent Attack Categories and weight values which reflect the importance of the Attack Stages. The entire signature set of different sensors and the role of specific sensor events in the Guidance Template is available by drilling down different elements in the interface. In the figure, the top panel of the GUI shows a Guidance Template, the columns are the different Attack Stages (each having a corresponding weight) and the nodes in each column are the Attack Categories. The bottom left panel shows a listing of events in each Category and the bottom right panel depicts event and sensor details.

The **Information Fusion Engine for Real-time Detection (INFERD)**, is the central control module of the system. It enables the analyst to start and stop fusion runs, set up configuration information and displays statistics and the status of real-time event fusion.

The **Native Visualization** provided by the system is shown in Figure 2. It depicts active Attack Tracks along with their Criticality values, in additional to information about the dynamic composition of the different Attack Stages and characterization information for specific events. In the figure, the top panel diplays a list of Attack Tracks for a

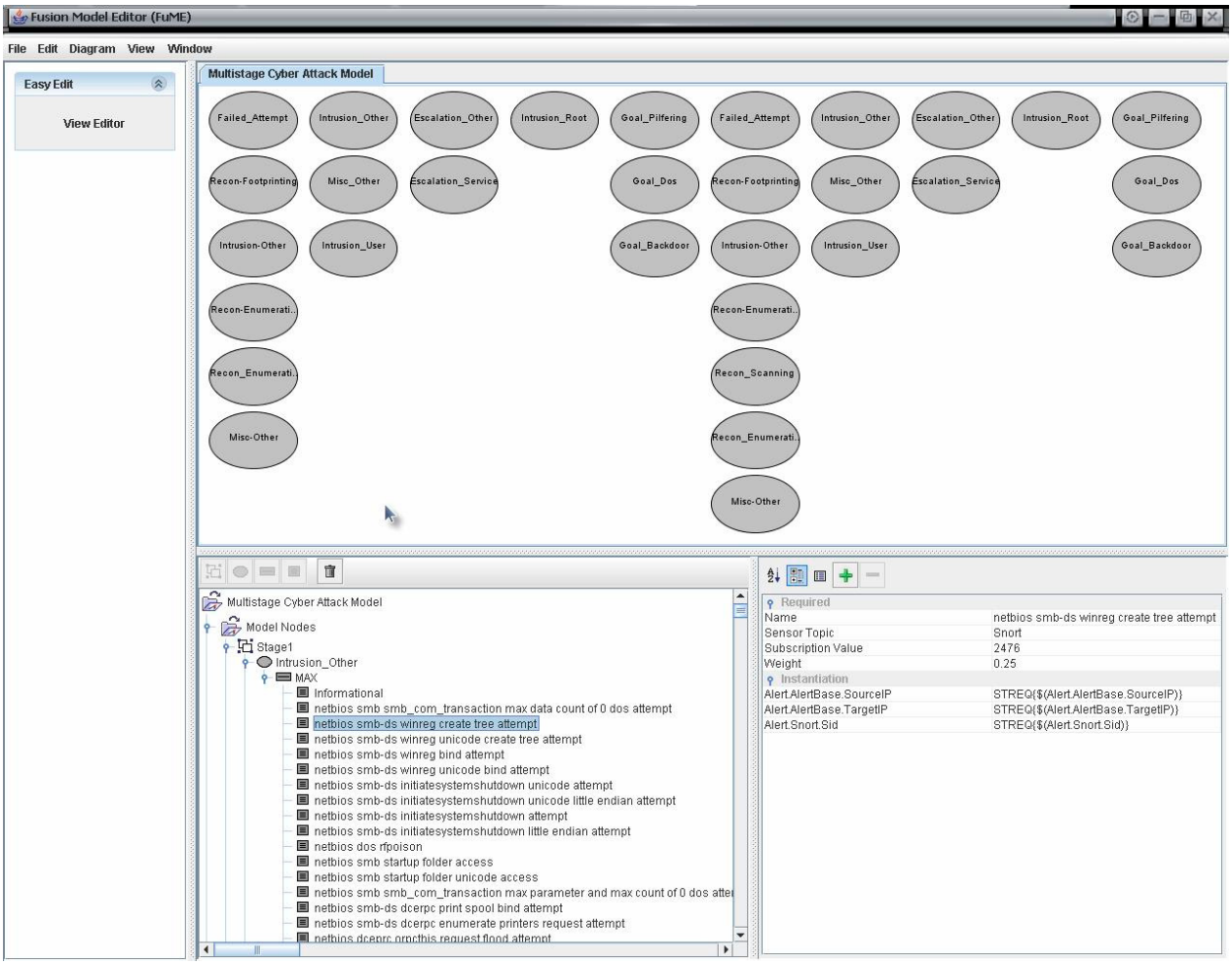

Figure 1: FUsion Model Editor (FUME) enables the analyst to define a Guidance Template and other elements of a fusion model

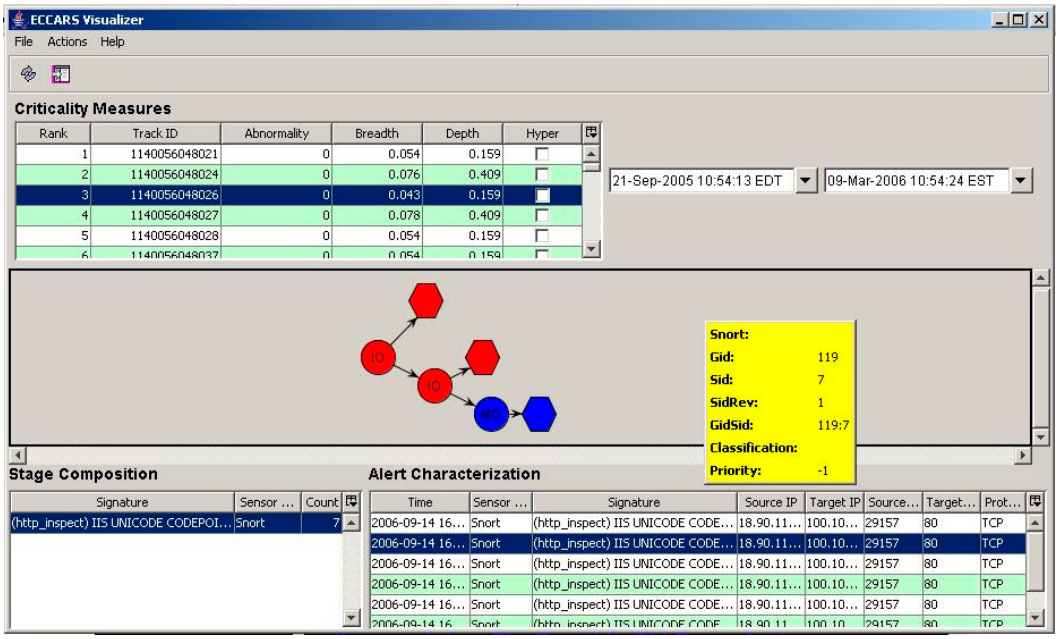

Figure 2: Native ECCARS Visualization displays color coded Attack Tracks with Criticality Values

certain event stream along with their scores with respect to certain metrics. The bottom panel shows one Attack Track, the circular nodes represent the Attack Categories, the colors/shades represent their weights and the hexagonal figures represent other tracks to which categories in this track are related (i.e., Attack Categories in one Attack Track may be part of other active tracks at the same time).

In addition to its own native visualization, ECCARS can also provide information to **Third-Party Visualization** tools that add further value to its event correlation process. One such tool that we have used for ECCARS visualization is Flexviewer, used by the Air Force Research Laboratory (AFRL).

4. CONCLUSION

Preliminary testing of our system indicates that correlation and visualization of heterogeneous network events in the context of multistage attacks adds significant value to the practice of cyber-attack detection. For complex attacks, multiple sensor fusion is invariably necessary. Detecting and mitigating against evolving attacks is simplified when the analyst can concentrate on attack paths that deserve more attention without getting bogged down with the sheer volume of events. Preliminary results indicate *Precision* values close to 100% and *Recall* values of about 83% when tested with the kind of scenarios presented here. The authors plan to present a more detailed report on ECCARS and its performance in the near future.

5. ACKNOWLEDGEMENTS

The authors would like to thank Skaion Corporation for providing the dataset used in the experiments. We also express our gratitude to Alion Science and Technology for their contributions to this research.

6. REFERENCES

[1] P. Ammann, D. Wijesekera, and S. Kaushik. Scalable, graph-based network vulnerability analysis. In *CCS '02: Proceedings of the 9th ACM conference on Computer and Communications Security*, pages 217–224, New York, NY, USA, 2002. ACM Press.

[2] R. Erbacher, K. Christensen, and A. Sundberg. Designing Visualization Capabilities for IDS Challenges. In *Proceedings of the IEEE Workshop on Visualization for Computer Security*, 2005.

[3] R. Erbacher, K. Walker, and D. Frincke. Intrusion and Misuse Detection in Large Scale Systems. *Computer Graphics and Applications*, 22(1), 2002.

[4] J. Goodall, W. Lutters, P. Rheingans, and A. Komlodi. Preserving the Big Picture: Visual Network Traffic Analysis with TNV. In *Proceedings of the 2005 ACM Workshop on Visualization for Computer Security*, October 2005.

[5] J. Haines, D. Ryder, L. Tinnel, and S. Taylor. Validation of sensor alert correlators. IEEE Security and Privacy, May 2001.

[6] S. Jajodia, S. Noel, and B. O'Berry. Topological analysis of network attack vulnerability. Managing Cyber Threats: Issues, Approaches and Challenges, Kluwer Academic Publishers, 2003.

[7] S. Jha, O. Sheyner, and J. Wing. Two Formal Analyses of Attack Graphs. In *15th IEEE Computer Security Foundations Workshop (CSFW'02)*, pages 49–63, Cape Breton, Nova Scotia, Canada, 2002.

[8] H. Koike, K. Ohno, and K. Koizumi. Visualizing Cyber Attacks through IP Matrix. In *Proceedings of the 2005 ACM Workshop on Visualization for Computer Security*, October 2005.

[9] Y. Livnat, J. Agutter, S. Moon, R. Erbacher, and S. Foresti. A Visualization Paradigm for Network Intrusion Detection. In *Proceedings of the IEEE Information Assurance Workshop*, 2005.

[10] S. Mathew, D. Britt, R. Giomundo, S. Upadhyaya, M. Sudit, and A. Stotz. Real-time Multistage Attack Awareness through Enhanced Intrusion Alert Clustering. In *Situation Management Workshop (SIMA 2005), MILCOM 2005*, Atlantic City, NJ, October 2005.

[11] S. Mathew, C. Shah, and S. J. Upadhyaya. An Alert Fusion Framework for Situation Awareness of Coordinated Multistage Attacks. In *Proceedings of the Third IEEE International Information Assurance Workshop*, pages 95–104, College Park, MD, March 2005.

[12] S. Noel, M. Jacobs, P. Kalapa, and S. Jajodia. Multiple Coordinated Views for Network Attack Graphs. In *Proceedings of the 2005 ACM Workshop on Visualization for Computer Security*, October 2005.

[13] S. Noel, E. Robertson, and S. Jajodia. Correlating Intrusion Events and Building Attack Scenarios through Attack Graph Distances. In *Proceedings of the 20th Annual Computer Security Applications Conference*, December 2004.

[14] J. Scambray, S. McClure, and G. Kurtz. Hacking exposed: Network security secrets and solutions. Osborne/Mcgraw-Hill, 2001.

[15] A. Valdes and M. Fong. Scalable Visualization of Propogating Internet Phenomena. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, October 2004.

[16] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju. VisFlowConnect: NetFlow Visualizations of Link Relationships for Security Situational Awareness. In *Proceedings of the 2004 ACM Workshop on Visualization and Data Mining for Computer Security*, October 2004.