Augmented Analysis for Network Attack Discovery

S.Vishnuvardhan[*], S.Venkatramulu[**], G.Ranjith[***]

[*] Assistant Professor, Warangal Institute of Technology and Science, Warangal, AndhraPradesh,India
[**] Associate Professor, Kakatiya Institute of Technology and Science, Warangal, AndhraPradesh,India
[***] Assistant Professor, Warangal Institute of Technology and Science, Warangal, AndhraPradesh,India

Abstract- There is an increasing awareness of the growing influence of organized entities involved in today's Internet attacks. However, there is no easy way to discriminate between the observed malicious activities of script kiddies and professional organizations, for example. For some time the project has collected data on a worldwide scale amenable to such analysis. Previous publications have highlighted the usefulness of so called attack clusters to provide some insight into the different tools used to attack Internet sites. In this paper, we introduce a new notion, namely cliques of clusters, as an automated knowledge discovery method. Clusters provide analysts with some refined information about how, and potentially by whom, attack tools are used. We provide some examples of the kind of information that they can provide. Our approach to the network attack pattern discovery problem is sketched in this section, and then presented in further detail in the later sections. It is assumed that on any target system, a packet sniffer tool captures all the raw network packets, and the TCP/UDP headers and other relevant information from the data payloads are recorded in a log file stored locally.

Index Terms- Network Security, Attack Patterns, Feedback Mechanism

I. INTRODUCTION

The Scenario we consider in this paper is the investigation that follows the occurrence of disruptive or suspected network attacks on one or more connected systems. It is well-known that network attacks typically do not occur in isolation: the activities that cause damage or detection are not stand-alone, because these attacks are impossible to carry out without some detailed required information of the target systems; therefore, they are always, by necessity, preceded by a few stages of exploratory probing by the attackers in order to obtain as much information as possible on the target system and thereby find vulnerabilities.

Consider the following simple scenario. A potential attacker A is trying to target an organization with a block of IP addresses. First, A will gather sufficient information to find vulnerabilities, by doing a ping sweep of the organization's network followed by port scans. A ping sweep of the IP address block is simply probing each IP address to see which systems are alive. Then a port scan can be carried out on each live system, which involves connecting to a large range of port numbers to find TCP and UDP ports that are listening and the corresponding services they are used for. Suppose A is able to determine that one of the IP addresses X runs a flavor of UNIX and has a Database (TTDB) server owned by root listening at port 32775, which is a remote procedure call (RPC) service that has a known vulnerability. Now A can simply execute a buffer overflow attack on port 32775 to get this program to execute and display it back on A's screen, which means A has gained root access to the system X. The awareness of the attack usually occurs when A is doing something damaging as the root, and this is the occurrence that gets investigated afterwards. Investigation into these visible breaches of security (occurrences of malicious attacks or attempts) is essential, so that there is more information learned than simply that the systems crashed or the network went down. For instance, in the above imaginary scenario, X could be used by A as an entry point into the entire organization's network, and after the unknown compromise of X, A proceeded to bring down the entire network. If no investigation is conducted afterwards, it would not have been known that X was the vulnerable point. It is critical to gather this sort of evidence. The sequence of network events leading up to the final breach as well as any others that cooperate with or contribute to these attack activities should be discovered. The information obtained is useful beyond the current investigation it can be archived and used to prevent similar future intrusions, and it can also be used to identify previously unknown security weaknesses in the system. It is becoming increasingly important to the IT industry and organizations at large to preserve and use network traffic as a form of digital evidence. Finally, our focus is on the digital forensic task of discovering attack patterns, rather than a real-time monitoring system that analyzes live network traffic data and triggers alerts. Efficiency is crucial for real-time monitoring systems; in contrast, our work is mainly concerned with the accuracy of attack pattern discovery.

II. PREVIOUS WORK

Effective network security administration depends to a great extent on having accurate, concise, high-quality information about malicious activity in one's network. However, attaining good information has become increasingly difficult because the profile of malicious traffic evolves quickly and varies widely from network to network, and because security analysts must discern the presence of new threats potentially hidden in an immense volume of "background radiation". In addition, much of the information available to security analysts from sources such as intrusion detection systems comes in the form of pinpoint descriptions of low-level activities, such as "source *A* launched attack *CVE-XXX* against destination *B*". Standard best practices rarely include automatically acting on such information due to the prevalence of false and redundant alarms. In addition, the

information often lacks sufficient breadth for forensic or root cause analysis.

The long-term objective of our work is to elevate the quality and timeliness of information provided to network security analysts. We appeal to the notion of network *situational awareness* as a means for defining information quality. Situational awareness is a military term referring to "the degree of consistency between one's perception of their situation and the reality" or to having "an accurate set of information about one's environment scaled to specific level of interest". We envision Network Situational Awareness (NetSA) as analysts with accurate, terse summaries of attack traffic, organized to highlight the most prominent facets. NetSA should also supplement these reports with drill-down analysis to facilitate countermeasure deployment and forensic study. For example, a NetSA environment should enable an analyst to quickly assess high-level information such as the cause of an attack (*e.g.*, a new worm, a botnet, or a misconfiguration), whether the attacker specifically targeted the victim network, and if this attacker matches one seen in the past.

III. PROPOSED SYSTEM

a. Initial Suspicion Score

In this module given a sequence of events, and given an attack definition, it is easy to partition E into groups each of which has the same ID and select only those events that satisfy the where condition. So without loss of generality, we assume for now that all events in E have the same ID and satisfy the where condition. We can compute the likelihood of E constituting an instance of attack type a using its aggregation and objective function; this is the initial suspicion score for E. To do this, we express the objective function ω as a probability distribution. The specific probability distribution chosen depends on the types of objective functions and the fields involved. When the objective function is to maximize the count of some field in the network events, we use a Gaussian distribution to express it. The time-span of a set of events is divided into a sequence of small subintervals, such that in each subinterval at most one event occurs which either does not increase or increases by one the count of the field of interest.

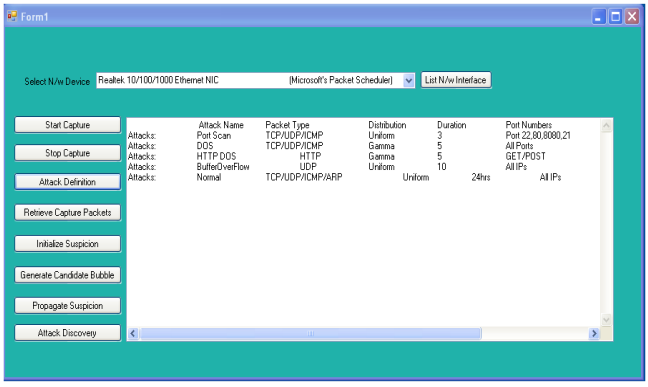

Fig.1 Attack definitions

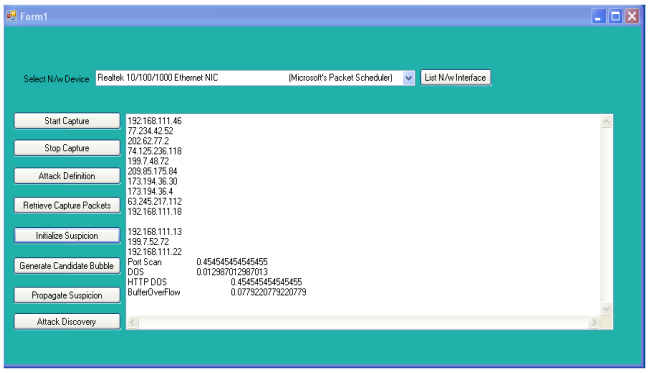

Fig. 2 Initial suspicion scores

b. Iterative Attack Pattern Discovery

In this module we discover bubbles with high suspicion values. Attack bubble is defined as a collection of network events originating from a common source that represents a likely attack. Once some bubbles of high suspicion have been identified, an iterative feedback mechanism allows the algorithm to iteratively search for more hidden but related attack bubbles, thus allowing the reconstruction of the complete attack sequence in its entirety. This module maintains a dynamic table susp feedback of feedback suspicion scores between 0 and 1 for each combination of ID, and attack type, a. Each bubble represents a potential attack, with the suspicion score being the degree of belief. So, a low suspicion score implies that the bubble x is probably not an attack, whereas a high suspicion score implies that the bubble is strongly believed to be an attack. The events associated with a bubble are the evidence that supports the belief of attack.

c. Generating Candidate Bubbles

After Every candidate bubble consists of one or more events that are usually close in time, and can be thought of as events that form a coherent unit that corresponds to some single activity. For example, an event that logs a remote procedure call (RPC) to an RPC service running on the target system is a candidate bubble, or several events that record a series of remote login attempts from the same source IP constitutes a candidate bubble. Because the log is comprised of network packet-level entries, with each entry corresponding to a single TCP or UDP packet, this initial step of generating candidate bubbles is necessary for delineating events related to the same unit of activity from the raw log. A candidate bubble defines some coherent set of events that could potentially match an attack definition, even if the probability is remote.

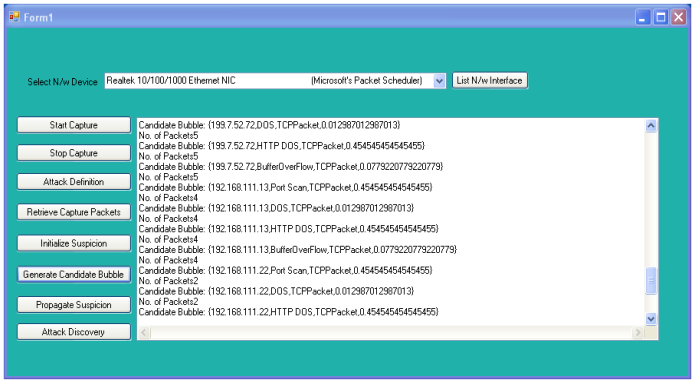

Fig.3 Candidate bubbles

d. Filtering and Propagate Suspicion

The set of candidate bubbles delineate units of coherent network activities that have an extremely wide range of suspicion, even those that are only tenuously probable to be attack activities are included. Every candidate bubble also has an associated suspicion score, which is updated at each iteration of the attack pattern discovery algorithm and represents the current suspicion that this candidate bubble is part of the attack pattern. The procedure for updating the current suspicion score of the candidates is discussed in detail later. At each iteration, we classify the candidate bubbles according to their suspicion scores into two classes, filtering the suspicious bubbles from rest. To do so, we apply the 2-means clustering algorithm to the set of candidate bubbles and return the top cluster as the bubbles. Given a set of bubbles, B, we wish to use the attack graph to infer other likely types of attacks that may not have been detected in B. This information will be the feedback to the next iteration of bubble generation. The suspicion is propagated from the set of bubbles B to a subset of the nodes of the attack graph G, and then further propagated throughout to the rest of G.

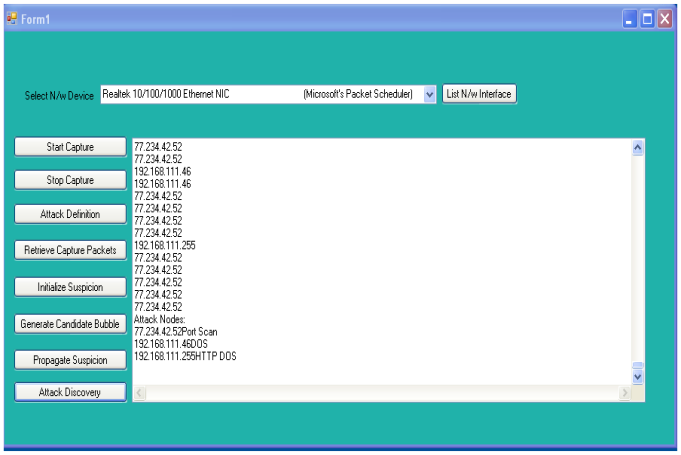

Fig.4 Attack nodes.

IV. RESULTS

The concept of this paper is implemented and different results are shown below, The proposed paper is implemented in Java technology on a Pentium-IV PC with 20 GB hard-disk and 256 MB RAM with apache web server. The propose paper's concepts shows efficient results and has been efficiently tested on different Datasets. The Fig 1, Fig 2, and Fig 3 shows the real time results compared.

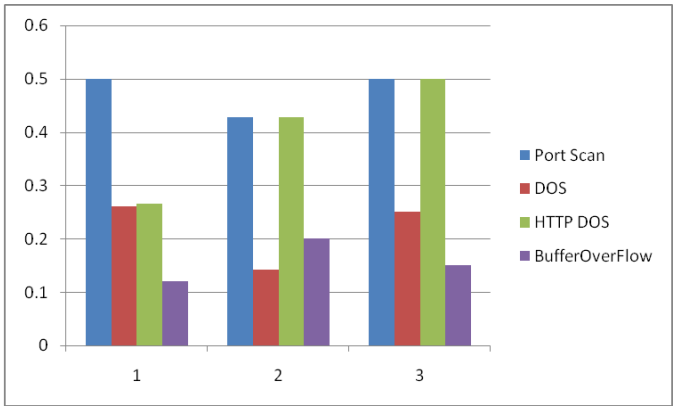

Fig. 5 Proposed system performing of attacks at different time.

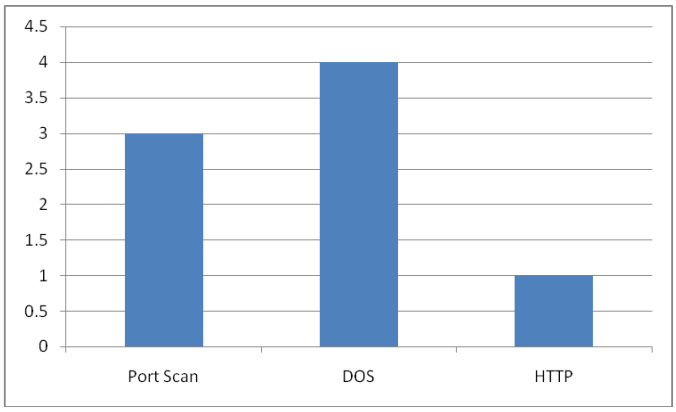

Fig. 6 Proposed system different types of attacks

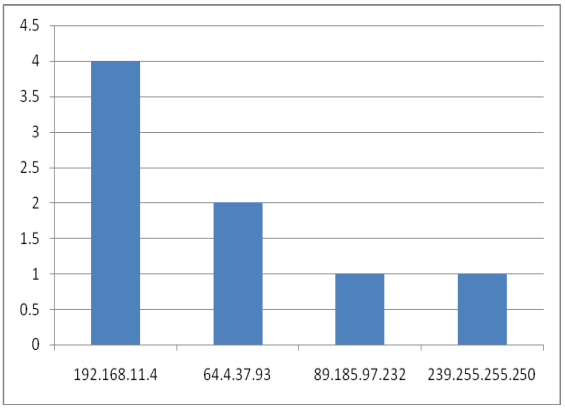

Fig. 7 Proposed Systems different sources of Attacks

V. CONCLUSIONS

In this paper, we have identified the problem of discovering the context of network events related to a security breach, by mining the logs of network traffic data. We proposed an iterative algorithm that uses a feedback mechanism to propagate likelihoods of attack events or suspicion scores to the next iteration, thereby increasingly refining the search for events or

attacks related to the ones already found. Our simulations verify the accuracy of the algorithm in discovering the attack patterns.

REFERENCES

[1] E. Casey, "Network traffic as a source of evidence: tool strengths, weaknesses, and future needs," *Elsevier Journal of Digital Investigation*, vol. 1, pp. 28–43, 2004.

[2] A. Lakhina, M. Crovella, and C. Diot, "Mining anomalies using traffic feature distributions," in *Proc. SIGCOMM'05*, August 21–26, 2005, Philadelphia, PA.

[3] S. Singh, C. Estan, G. Varghese, and S. Savage, "Automated worm fingerprinting," in *Proceedings of the 6th Symposium on Operating Systems Design and Implementation (OSDI'04)*. USENIX, 2004, San Fransisco, CA.

[4] K. Shanmugasundaram, H. Bronnimann, and N. Memon, "Payload attribution via hierarchical Bloom filters," in *Proc. ACM Conference on Computer and Communications Security*, 2004, Washington DC.

[5] C. Cho, S. Lee, C. Tan, and Y. Tan, "Network forensics on packet fingerprints," in *Proc. 21st IFIP Information Security Conference (SEC 2006)*, 2006, Karlstad, Sweden.

[6] M. Ponec, G. P., W. J., and B. H., "New payload attribution methods for network forensic investigations," *ACM Transactions on Information and System Security*, vol. 13, no. 2, pp. 15:2–15:32, February 2006.

[7] N. Provos, "A virtual honeypot framework," in *Proc. 13th USENIX security symposium*, 2004, San Diego, CA.

[8] P. Baecher, M. Koetter, T. Holz, M. Dornseif, and F. Freiling, "The nepenthes platform: an efficient approach to collect malware," in *Proc. 9th international symposium on recent advances in intrusion detection (RAID)*, September 2006, Hamburg, Germany.

[9] J. Riordan, D. Zamboni, and Y. Duponchel, "Building and deploying billy goat, a worm-detection system," in *Proc. 18th annual FIRST conference*, 2006, Baltimore, Maryland.

AUTHORS

First Author –S. Vishnuvardhan is a Post Graduate in Master of Technology from Kakatiya University, in Software Engineering. Having Teaching Experience of 03 Years and at present Working as Asst Professor, Department of Computer Science and Engineering, Warangal Institute of Technology and Science, Oorugonda (V), Gudepadu X Roads, Atmakur (M), Warangal-506342., Email: Vishnu.0992@gmail.com

Second Author – S.Venkatramulu is a Post Graduate in Master of Technology . Having Teaching Experience more than 15 Years and at present Working as Associate Professor, Department of Computer Science and Engineering, Kakatiya Institute of Technology and Science, Yerragattu gutta, Hasanparthy (M), Warangal, Email: venkatramulu10@gmail.com

Third Author – G.Ranjith is a Post Graduate in Master of Technology from J.N.T University, in Software Engineering. Having Teaching Experience of 03 Years and at present Working asAsst Professors, Department of Computer Science and Engineering, Warangal Institute of Technology and Science, Oorugonda (V), Gudepadu X Roads, Atmakur (M), Warangal-506342., Email: gyaderlaranjith@gmail.com