CAMNEP: Agent-Based Network Intrusion Detection System[*]

(Short Paper)

Martin Rehak
Center for Applied Cybernetics
FEE, Czech Technical
University in Prague
mrehak@labe.felk.cvut.cz

Michal Pechoucek
Department of Cybernetics
FEE, Czech Technical
University in Prague
pechouc@labe.felk.cvut.cz

Pavel Celeda
ICS, Masaryk University
INVEA-TECH, Inc.
celeda@invea-tech.com

Jiri Novotny
Institute of Computer Science
Masaryk University
novotny@ics.muni.cz

Pavel Minarik
ICS, Masaryk University
Mycroft Mind, Inc.
min@mycroftmind.com

ABSTRACT

We present a prototype of agent-based intrusion detection system designed for deployment on high-speed backbone networks. The main contribution of the system is the integration of several anomaly detection techniques by means of collective trust modeling within a group of collaborative detection agents, each featuring a specific detection algorithm. The anomalies are used as an input for the trust modeling. In this stage, each agent determines the flow trustfulness from aggregated anomalies. The aggregation is performed by extended trust models that model the trustfulness of generalized situated identities, represented by a set of observable features. The system is based on traffic statistics in NetFlow format acquired by dedicated hardware-accelerated network cards, and is able to perform a real-time surveillance of the gigabit networks.

Categories and Subject Descriptors

I.2.11 [**ARTIFICIAL INTELLIGENCE**]: Distributed Artificial Intelligence—*Intelligent agents*

General Terms

Security

Keywords

trust, intrusion detection, network behavior analysis

1. INTRODUCTION

Current generation of network devices allows real-time scraping of structured snapshots of the traffic on the networks. This information is provided in the NetFlow [2] format introduced by CISCO, and allows us to observe individual flows on the network. **Flow** is an unidirectional component of TCP connection (or UDP/ICMP equivalent), defined as a set of packets with identical source and destination IP addresses, ports and protocol. The availability of this information allows the deployment of **Network Behavior Analysis** (NBA) [11] systems that process this information and infer the conclusions about the maliciousness of specific groups of flows. The NBA systems are not designed to detect stealth attacks against single hosts, but provide a detection capability against the attacks that are *significant from network perspective*, such as horizontal scanning (used to map the network for on-line hosts, typical for malware propagation), vertical scanning (used to determine the services offered by a host), denial of service attacks and other relevant events. Furthermore, the methods outlined in our system also aim to detect the activity of the hosts that were taken over by an attacker (typically using zombie networks) and are used to stage further zombie recruiting or exploitation.

The CAMNEP system presented in this paper uses a set of *anomaly detection* [3] techniques[12, 6, 5, 4]. These algorithms maintain a model of expected traffic on the network and compare it with real traffic to identify the discrepancies that are identified as possible attacks. These methods are effective against zero-day attacks and previously unknown threats, but suffer from comparatively higher error rate [7], frequently classifying legitimate traffic as malicious (false positives), or failing to spot the malicious flows (false negatives). CAMNEP addresses this problem by the use of classic agent techniques: **trust** and **reputation** [10] to improve the quality of individual agent's classifications (see Section 2.2). It integrates several methods using a collective, trust-based detection process. This combination allows us to correlate the results of the used methods and to combine them to improve their effectiveness.

[*] This material is based upon work supported by the European Research Office of the US Army under Contract No. N62558-07-C-0001. Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not necessarily reflect the views of the ERO, US Army. Also supported by Czech Ministry of Education grants 1M0567 and 6840770038

Cite as: CAMNEP: Agent-Based Network Intrusion Detection System (Short Paper), M. Rehak, M. Pechoucek and P. Celeda, *Proc. of 7th Int. Conf. on Autonomous Agents and Multiagent Systems (AAMAS 2008)- Industry and Applications Track*, Berger, Burg, Nishiyama(eds.),May,12-16.,2008,Estoril,Portugal,pp.133-136.

2. SYSTEM ARCHITECTURE

System architecture is split into several layers, distinguished by their functionality, physical distribution and processing speed requirements. *Traffic acquisition layer* uses hardware-accelerated NetFlow probes [1] to acquire traffic from gigabit-speed networks and to extract the meaningful features for attack detection. The *detection layer* then classifies the preprocessed traffic and detects the attacks, that are presented to operators by visualization agents from the *operator interface* layer.

The lower-layers are based on hardware network probes and modified open-source software, the intelligent core of the system is developed within the AGLOBE multi-agent platform that facilitates agent cooperation and runtime system integration.

2.1 Traffic Acquisition

The traffic acquisition and preprocessing layer is responsible for network traffic acquisition, data preprocessing and distribution to upper system layers. It only uses the flow characteristics based on information from packet's headers, and can therefore analyze even *ciphered traffic*.

Therefore we use hardware accelerated NetFlow probes called FlowMon. The FlowMon probe is a passive network monitoring device based on the COMBO hardware [1], which provides high performance and accuracy. The probe handles 1 Gb/s traffic at line rate in both directions and exports acquired NetFlow data to different collectors. The high performance of the card guarantees robust performance even under attack, when the traffic characteristics make the processing more computationally intensive.

The collector servers store incoming packets with NetFlow data from FlowMon probes into its internal database. Each collector server provides interface to graphical and text representation of raw network traffic, simple flow filtration, aggregation and statistics evaluation, using source and destination IP addresses, ports and protocol. Detection agents connect to collector to obtain the data and use them for the detection.

Even after probes deployment in monitored network, the probes can be reprogrammed to acquire new traffic characteristics. The system is fully reconfigurable and the probes can adapt their features and behavior to reflect the changes in the agent layer.

2.2 Attack Detection by Trusting Agents

The goal of the cooperative threat detection layer is to provide the assessment of maliciousness of the individual flows in each flow set observed by the system. To achieve this goal, we use the trust modeling techniques, and extend them to cover the domain-specific needs.

Each detection agent contains one of the anomaly detection methods (detailed in [9]), coupled with an extended trust model defined in [8]: (*i*) MINDS agent [4], which reasons about the number of flows from and towards the hosts in the network, and detects the discrepancies between the past and current traffic, (*ii*) Xu agent [12], which reasons about the traffic from individual hosts using the normalized entropies and rules, (*iii*) Lakhina Entropy [6] agent, which builds a model that predicts the entropy of traffic features from individual hosts and identifies anomalies as differences between predicted and real value, and finally (*iv*) Lakhina Volume agent [5], which applies the same method to traffic volumes.

All agents, regardless of their type, process the data received from the acquisition layer in three distinct stages (see Figure 1): (*i*) anomaly detection, (*ii*) trust update and (*iii*) collective trust conclusion.

In the network security domain, low trustfulness of the flow means that the flow is considered as a part of an attack. Trustfulness is determined in the $[0, 1]$ interval, where 0 corresponds to complete distrust and 1 to complete trust. The *identity* of each flow is defined by the features we can observe directly on the flow: *srcIP, dstIP, srcPrt, dstPrt, protocol*, number of *bytes* and *packets*. If two flows in a data set share the same values of these parameters, they are assumed to be identical. The *context* of each flow is defined by the features that are observed on the other flows in the same data set, such as the number of similar flows from the same *srcIP*, or entropy of the *dstPrt* of all requests from the same host as the evaluated flow. While the agents in our system use the same representation of the identity, the context is defined by the features used by their respective anomaly detection methods to draw the conclusions regarding the anomaly of the flow. Identity and context are used to define the *feature space*, a metric space on which the trust model of each agent operates [8]. The metrics of the space describes the similarity between the identities and contexts of the flows, and is specific to each agent.

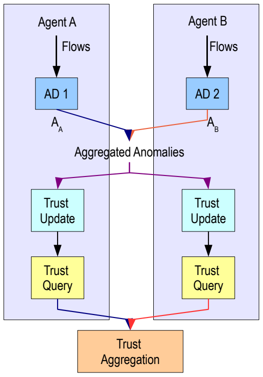


Figure 1: Agent layer operation.

Anomaly detection During the anomaly detection stage, the agents use the embedded anomaly detection method to determine the anomaly of each flow as a value in the $[0, 1]$ interval, where 1 represents the maximal anomaly, and zero no anomaly at all. The anomaly values are shared with other detection agents, and used as an input in the second phase of the processing.

Trust update During the trust update, the agents integrate the anomaly values determined for individual flows during the first phase into their trust models. As the reasoning about the trustfulness of each individual flow is both computationally infeasible and unpractical (the flows are single shot events by definition), the model holds the trustfulness of significant flow samples (e.g. centroids of (fuzzy) clusters) in the identity-context space, and the anomaly of each flow is used to update the trustfulness of centroids in its vicinity. The weight used for the update of the centroid's trustfulness with the anomaly values provided for the flow decreases with distance from the centroid. Therefore, as each agent uses a distinct distance function, each agent has a different insight into the problem. The flows are clustered according to the different criteria, and the cross-correlation implemented by sharing of the anomaly values used to update the trustfulness helps to eliminate random anomalies.

Collective trust estimation In the last stage of processing, each agent determines the *trustfulness* of each flow (with an optional normalization step), all agents provide their trustfulness assessment (conceptually a reputation opinion) to the aggregation agents and the visualization agents, and the aggregated values can then be used for traffic filtering.

In order to be successful, the trustfulness aggregated by the system should be as close as possible to the maliciousness of the flow. When we reason about the malicious and untrusted flows as sets (they are actually fuzzy sets), we wish them to overlap as much as possible. We can define the common misclassifications errors using the trustfulness and maliciousness of the flow. The flows that are malicious and trusted are denoted as *false negatives*, and the flows that are untrusted but legitimate are denoted *false positives*. Typically, when we tune the system to reduce one of these sets, the size of the other increases. Intuitively, it may seem that we may be ready to ignore higher rate of false positives, rather than false negatives. However, this is rarely the case in the IDS systems deployed for operational use, as the legitimate traffic vastly outnumbers the attacks and even a low rate of false positives makes the system unusable.

The importance of the trust model lies in the cross-validation of anomaly opinions in the trust models of detection agents, each of these models being based on different traffic features. In order to classify a set of flows as an attack, the flows from the set need to fall in the vicinity of centroids with low trustfulness in the models of most agents. In practice, most attack flows fall in the neighborhood of a single centroid, as we can see in Fig. 2. On the other hand, when one of the agents creates a false positive, the flows are likely to be dispersed in the trust models of the other agents and the on-average higher trustfulness of the associated centroids prevails during the final aggregation.

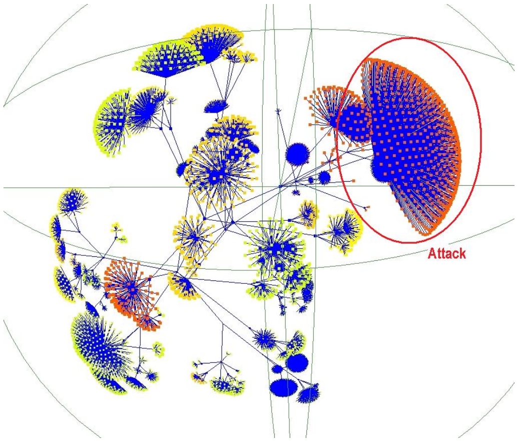
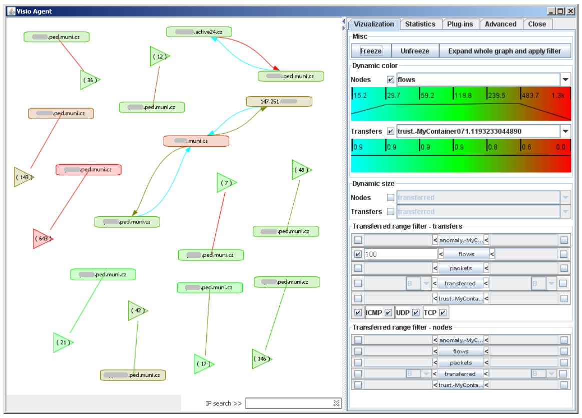


Figure 2: A peek into the trust model of a detection agent. The flows are displayed as tree extremities attached to the closest centroid of the trust model. Note that the attack flows are concentrated next to a single centroid.

2.3 Operator and Analyst Interface Layer

The operator and analyst interface layer of the CAMNEP system is represented by *Visio Agent*. This agent provides visualization of the network traffic based on graphs where nodes represent particular hosts and oriented edges represent network flows. Graph-based visualization is complemented by a high-level visualization by histogram of trustfulness and a fast glimpse on traffic characteristics provided by the statistical analysis tool. Besides being used to present the whole data, the visualizer allows selective visualization of user-selected flow groups from the histogram and/or analysis components.

Visio Agent provides visual support for analytic reasoning with the use of the detection layer results. The network visualization based on graphs approximates the structure of network topology and communication, therefore being natural for both network administrators and non-specialists. It also autonomously gathers support data on behalf of its users.



Nodes ☑ flows

Transfers ☑ trust.-MyContainer071.1193233044890

Nodes ☐ transferred

Transfers ☐ transferred

☑ 100 < flows <

☑ ICMP ☑ UDP ☑ TCP ☑

Figure 3: Example of the overall situation in the network with filter applied.

The graph-based traffic representation is enhanced with several significant features. The user can list the flows and traffic statistics associated with each edge/host, and display associated trustfulness. The traffic can be **filtered** and **aggregated** according to many relevant features, including trustfulness and anomaly values. The visual attributes of the display (such as node/edge size and color) can also adapt to these characteristics, making the user orientation easier. The information provided by third parties (DNS, whois) is seamlessly integrated into the visualization.

As current network traffic is a scale-free network, it is particularly important to handle the visualization of supernodes, i.e. the nodes with high number of connections. These nodes are typical for many attack scenarios, as well as for high-value targets. Visualizer therefore replaces the one-shot connections to/from these hosts by a special representation of a "cloud" of traffic, and only singles out the nodes that also connect to other nodes in the observed network.

3. SYSTEM EVALUATION

Any intrusion detection system is evaluated in terms of false positives/false negatives. We have performed several series of experiments, both on known attacks with well defined characteristics, and on real-world attacks observed on protected networks. In this paper, we present a selection of results.

In the first set of experiments, we have measured the ability of the system to detect a mix of attacks, including vertical and horizontal scans (TCP SYN, TCP CONNECT and UDP), brute force attacks on SSH passwords, OS fingerprinting and others. The trustfulness assigned to attacks is shown in Fig. 4, where we highlight the attacks not-detected by the system. An attack is considered as detected if (i) its trustfulness is below 0.2, or (ii) if the trustfulness is more than one σ below the average of the trustfulness distribution. We can clearly see that the system consistently detects the attacks with more than 400 flows over a 5 minute interval – this corresponds to about 1% of traffic volume measured in number of flows.

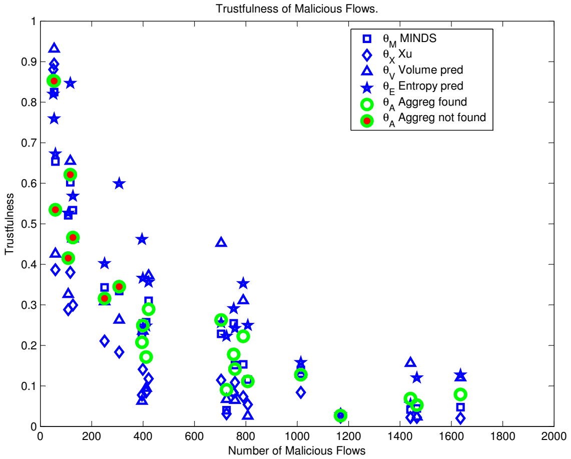


Figure 4: Average trustfulness of attack flows according to attack size.

We have performed an evaluation of algorithm on a 30-minutes sample of real network traffic. The performance of the system is slightly superior to off-line manual analysis performed by an experienced administrator, which took more than one day to perform. Reconciliation of the the results showed that the system had slightly lower false positive rate and slightly higher true negatives, but the principal attacks (botnet nodes and buffer overflow) were consistently detected. In both cases, false positives were roughly a half of the reported incidents (from a total of 17).

When we compare the system results with those of integrated anomaly detection methods in terms of false positives/false negatives, the aggregated results outperform any of the methods in both criteria (by ratio of 10 when considering traffic sources), and also outperform the classification by averaged anomalies by a factor of 2 in FP, while identifying more true attacks.

4. CONCLUSIONS

In our work, we have presented a multi-agent framework that enables the integration of several existing network behavior analysis methods. The agent techniques are used not only as a code-level and integration framework, but also as a reasoning core of the approach which is based on extended trust modeling and simple reputation mechanism. The experimental results on the real traffic, as well as the evaluation performed by network administrators hint that this combination is significant not only from the research perspective, but also from the industrial perspective.

The fact that the system assigns the trustfulness score rather than binary label (attack/legitimate) provided by classic IDS systems is actually an advantage. The trustfulness together with the number of flows, is also a good estimate of the priority that the attack requires. The primary output of the system is a histogram of current traffic trustfulness. This form is very convenient for rapid analysis. Once the traffic is classified, the system leaves the further steps of the analysis on the operator. In the current version of the system, the anomaly detection methods are selected to address the same types of attacks, albeit with different effectiveness. This is a very strong restriction, and we intend to relax it in the future by introducing a more elaborate reputation mechanism instead of simple average.

5. REFERENCES

[1] CESNET, z. s. p. o. Family of COMBO Cards. http://www.liberouter.org/hardware.php, 2007.

[2] Cisco Systems. Cisco IOS NetFlow. http://www.cisco.com/go/netflow, 2007.

[3] D. E. Denning. An intrusion-detection model. *IEEE Trans. Softw. Eng.*, 13(2):222–232, 1987.

[4] L. Ertoz, E. Eilertson, A. Lazarevic, P.-N. Tan, V. Kumar, J. Srivastava, and P. Dokas. MINDS - Minnesota Intrusion Detection System. In *Next Generation Data Mining*. MIT Press, 2004.

[5] A. Lakhina, M. Crovella, and C. Diot. Diagnosis Network-Wide Traffic Anomalies. In *ACM SIGCOMM '04*, pages 219–230, New York, NY, USA, 2004. ACM Press.

[6] A. Lakhina, M. Crovella, and C. Diot. Mining Anomalies using Traffic Feature Distributions. In *ACM SIGCOMM, Philadelphia, PA, August 2005*, pages 217–228, New York, NY, USA, 2005. ACM Press.

[7] S. Northcutt and J. Novak. *Network Intrusion Detection: An Analyst's Handbook*. New Riders Publishing, Thousand Oaks, CA, USA, 2002.

[8] M. Rehak and M. Pechoucek. Trust modeling with context representation and generalized identities. In *Cooperative Information Agents XI*, number 4676 in LNAI/LNCS. Springer-Verlag, 2007.

[9] M. Rehak, M. Pechoucek, K. Bartos, M. Grill, and P. Celeda. Network intrusion detection by means of community of trusting agents. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2007 Main Conference Proceedings) (IAT'07)*, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[10] J. Sabater and C. Sierra. Review on computational trust and reputation models. *Artif. Intell. Rev.*, 24(1):33–60, 2005.

[11] K. Scarfone and P. Mell. Guide to intrusion detection and prevention systems (idps). Technical Report 800-94, NIST, US Dept. of Commerce, 2007.

[12] K. Xu, Z.-L. Zhang, and S. Bhattacharrya. Reducing Unwanted Traffic in a Backbone Network. In *USENIX Workshop on Steps to Reduce Unwanted Traffic in the Internet (SRUTI)*, Boston, MA, July 2005.