DEMO–ReasONets: A Fuzzy-based Approach for Reasoning on Network Incidents

Giuseppe Petracca
Information Sciences and Technology
The Pennsylvania State University
University Park, USA
gxp18@psu.edu

Anna Squicciarini
Information Sciences and Technology
The Pennsylvania State University
University Park, USA
acs20@psu.edu

William Horne,
Hewlett-Packard Research Labs
Princeton, US
wiliam.horne@hp.com

Marco Casassa-Mont,
Hewlett-Packard Research Labs
Bristol, UK
marco.casassa@hp.com

ABSTRACT

We provide an approach for real-time analysis of ongoing events in a controlled network. We propose ReasONets, i.e. Reasoning on Networks, a distributed and lightweight system, able to process and reason about anomalies and incidents observed in closed networks. To the best of our knowledge this is the first system combining detections and classification of network events with real-time reasoning. Our demo will show a running prototype of the ReasONets, demonstrating the power and accuracy of the reasoning process in presence of incidents of various nature.

Categories and Subject Descriptors

K.6.5 [**Security and Protection**]: Unauthorized Access

Keywords

Reasoning, Situational Awareness

1. INTRODUCTION

Detection is just one step in an overall plan to handle security incidents. Once detected, security incidents are typically forwarded to a Computer Securities Incident Detection Team (CSIRT), where security analysts triage and investigate incidents and formulate a response [6]. This work is labor intensive and relies on extensive domain expertise. The objective of our work is to provide an approach for deeper and real-time understanding of ongoing events in a controlled network. We aim to detect anomalies in the behavior of machines and inappropriate use of the network, further identifying the nature and severity of the observed security incidents.

We propose ReasONets, an effective and lightweight system, able to process and reason about anomalies and incidents observed in closed networks. ReasONets combines aspects of anomaly detection with Case-Based Reasoning methodologies [2, 3], in order to provide situational awareness in case of network incidents. Underlying the Case-Based Reasoning process deployed within ReasONets is the understanding that no security event will ever be identical to previously experienced incidents in absolute terms, but should show enough similarities to be qualified as an event of a certain type. The understanding of anomalous events is not gathered from rules or general statistics, but by the analysis of cases, where each case represents a specific type of event which is already analyzed. Furthermore, we effectively control and model the uncertain and inaccurate information collected in real-time, by exploiting the Fuzzy Logic Theory [5]. Adaptation is also included in our system: if no previous case matches an observed event, we adopt a rule-based system to identify when new cases are to be adapted and, possibly, merge existing previous cases. Adaptation is supervised by the network administrator who assigns a semantic meaning to the various cases, and configures system's parameters based on his experience and system knowledge.

The detection of a machine accessing malicious domains is a simple yet effective example demonstrating how ReasONets differs from common SIM/SEM/SOC systems. Common SIM/SEM/SOC systems usually rely on Black Lists and White Lists, through which it is impossible to infer if a "new" domain is a good one or not [1]. ReasONets uses a set of metrics that allow infer the nature of a domain on-the-fly. In details, our system measures the content type (i.e. pornographic, political, sport), and the geographical distance of the domain respect to the closest known malicious domain. Further, we check if the registrant is an organization that owns other malicious domains, and if the domain is in the same network of a well-known malicious domain. The added value of our system is the capability to do inference on completely unknown domains, allowing us to obtain early identification of malicious domains, and their relationship (if any) with existing ones.

2. THE REASONETS ARCHITECTURE

The design of the ReasONets system is presented in Figure 1. A three-layer multivariate analysis model is adopted. By

CCS'12, October 16–18, 2012, Raleigh, North Carolina, USA.
ACM 978-1-4503-1651-4/12/10.

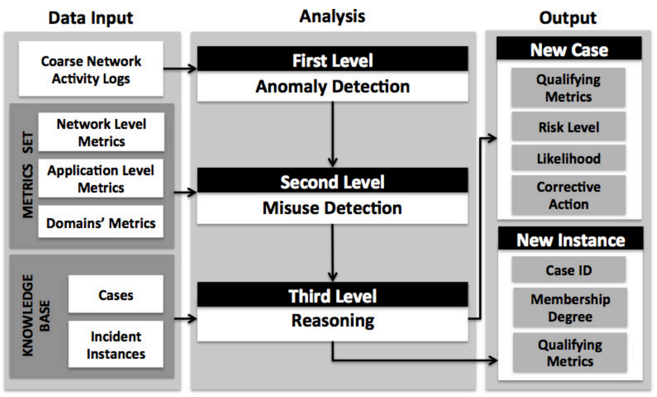


Figure 1: Overall Approach

exploiting coarse network activity logs, the system first analyzes machines' behavior in order to detect potential anomalies, by means of the *Anomaly Detection* component. In presence of anomalies in the network, a second layer of analysis is activated to assess if a misuse is ongoing. For this purpose, the *Misuse Detector* initially identifies and records the "normal behavior" of each monitored machine, in absence of anomalies. Then, in order to detect anomalies it periodically compares the current machine behavior with the normal one. As soon as a machine or network misuse is detected, the *Reasoner* is activated. The Reasoner uses the information collected through the first two layers augmented with additional data, to perform reasoning.

2.1 The ReasONets Components

Anomaly Detector. The Anomaly Detector (AD) is designed to be computationally light and extremely fast. Its goal is to identify potential anomalies and trigger alerts in presence of suspicious network traffic, rather than detect incidents and intrusions. The AD implements a two-variate model to check whether the reported suspicious traffic is in fact representative of anomalies. Two type of anomalies are detected: (1) single machines that open connection toward unknown domains/machines, or toward known domains/machines but using unusual protocols or port numbers; (2) subset of machines with same sequence of connections toward specific unknown domains/machines, not necessarily in the same order. Here, as unknown domain/machine we mean a web domain or machine external to the monitored network, never visited/contacted before by any internal machine. In order to improve the efficiency of our AD, we filter out ordinary traffic by training the model. New modules may be added to address other anomalies or patterns, by exploiting the modularity of our system.

We focus on these anomalies since malicious activities are likely to exhibit some specific communication pattern that is different from the norm, e.g. misusing protocols, or using uncommon port number or destination addresses that internal host have never used before [4]. We group the machines' behavior in clusters, to account for situations wherein a group of hosts is infected or compromised. With the first type of anomaly-based detection approach we address any other case in which only one machine is involved, for example attacks to an internal machine, or an internal machine running forbidden software like torrent clients.

Misuse Detector. The Misuse Detector (MD)[1] is in charge of delineating the "normal behavior" of monitored machines, and compare it with the registered machine's behavior during monitoring. Specifically, the MD collects a number of critical network and application metrics. For each metric, the average value, and the range of variability recorded is collected, and stored in records identified by means of the Machine's MAC address or Device ID. The metrics set can be divided in two groups: **Network Level Metrics**, aiming to measure network traffic activities; and **Application Level Metrics**, aiming approximate the HTTP traffic generated by each monitored machine, during its interaction with external machines and the Internet.

Reasoner. The Reasoner represents the core of our architecture, and it performs Case-Based Reasoning (CBR) [2] on the events detected by the higher system layers. A case represents a known incident or security event, which has been experienced in the system and addressed by administrators. Each case is represented by a vector of significant features, each denoting a metric and the corresponding value range.

We adopt a knowledge base (KB) which currently collects and models two classes of cases. First is the set of cases, which represent the most common network spread malware (e.g. Virus, Botnet, Worms, Keylogger). Second is the set of non-malware related incidents, that can be observed through network analysis. For example, access to a domain with adult content, unauthorized access to a monitored machine from a remote one, or SQL Injections. The cases are obtained using a hybrid approach that combines empirical evaluations and analysis of well-known security incidents affecting small enterprise networks. Cases are also dynamically added to the ReasONets as they are experienced, according to a set of adaptation rules within the model.

Threshold-based approaches have been previously used in order to detect incidents, with and without the aid of a CBR system. However, using simple thresholds may not provide sufficient knowledge about the event, and would fail in case of hybrid events, that appear similar to more than one incident [1, 4]. To cope with these issues, we have designed our reasoner by adopting a multi-layer approach that builds on *Fuzzy Logic* and on *ad hoc-case Fuzzy Ranking.* Fuzzy-logic handles situations where no-crisp answers can be found, for example by determining to what degree an incident is related to a known case. In addition, it allows us to reason on the overall similarity of events labeled with a same case. Beside fuzzy-based analysis, we compare selected and weighted features of relevant cases for any potential new input, so as to check for information that is most indicative of a case, and therefore discern the nature of the incident.

1. Rank candidate cases. We calculate the membership degree m of the current incident inc for each case modeled in the KB, by measuring absolute distances among each feature f characterizing inc and the ones profiling a case c, or previously registered instance $inst$ of c. The membership degree of inc for c is computed as follows:

$$m_{inc}(c) = \frac{\sum^{inst \in c}\left[m_{inst}(c) \times \sum^{|f|}\left(\beta - \left\| \frac{inc(f)}{\left(\frac{\alpha(f)}{\beta}\right)} - \frac{inst(f)}{\left(\frac{\alpha(f)}{\beta}\right)} \right\| \right) \right]}{\sum^{|f|}\left(\beta - \left\| \frac{inc(f)}{\left(\frac{\alpha(f)}{\beta}\right)} - \frac{inst(f)}{\left(\frac{\alpha(f)}{\beta}\right)} \right\| \right)}$$

[1]The term misuse detector is used with a slightly different meaning than the classic IDS terminology

Where $\alpha(f)$ represents the maximum value ever registered for the feature *f*, and β is the maximum possible distance between the measured values. Both are parameters used for normalization. $inc(f)$ and $inst(f)$ represent respectively the measured value of f for the current incident inc and the instance $inst \in c$. Finally, $m_{inst}(c)$ is the membership degree of the instance $inst$ for the case c.

2. Verify the presence of a dominant case. We select the first k Near Neighbor cases from the set of ordered cases and check how cohesive the k-NN set is, in search of a clear dominant case. If a dominant case is found, we model the incident as an instance of it.

3. Apply fuzzy ranking. If no dominant case is found, we compute an additional ranking for each case in the previous k-NN set. The fuzzy ranking algorithm calculates $m_{inc}(c)$ as follows:

$$r_c \times p_c \times \left(\sum^{|f|} w_c(f) \times Defuzzy\left[\beta - ||Fuzzy(inc(f)) - Fuzzy(c(f))||\right] \right)$$

This ranking algorithm assigns a different weight $w_c(f)$ to each specific feature. A risk parameter r_c and a likelihood parameter p_c, are also taken into account to classify the incident with a greater accuracy.

Fuzzification and defuzzification steps of the measured features - represented in the equation by mean of $Fuzzy$ and $Defuzzy$ functions - are employed in accordance with the Fuzzy Theory [5].

4. Create a new case. If after step 3, it is still not possible to find a dominant case, we classify the incident as new case by starting the case profiling process and storing it.

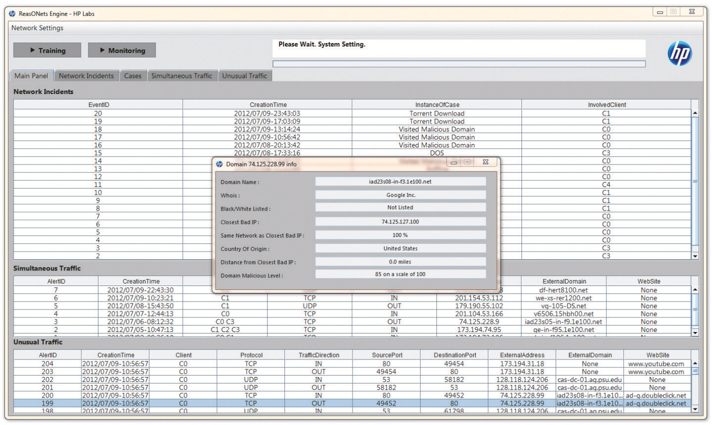
5. Merge cases. The model is periodically optimized: if several incidents with same features are mapped on the same subset of cases, we merge them in a single case.

2.2 Architectural Deployment

ReasONets is deployed on two machines: the *Network Logger*, in charge of intercepting all the traffic generated within the network, logging the network activity; and the *Engine*, in charge of performing analysis and reasoning, basing on collected data. Since single devices can change IP addresses frequently, a mapping of the IP address back to MAC addresses or Device ID is performed through DHCP logs.

Inputs to the system are represented by data obtained from monitoring devices connected to the network, whereas outputs are detailed indications about the suspected nature of incidents, their severity, and the confidence associated with these assessments. Feedback and possible actions to complete in order to address the incident are also provided to the network administrator, as shown by the admin's control panel presented in Figure 2.

We have conducted a preliminary study of scalability for the presented solution. We have used, as Engine and as Network Logger, two machines equipped with an Intel(R) Core(TM) i7-2720QM CPU @ 2.20Ghz and 4 GB of RAM. They were running Windows 7 Ultimate OS (32 bit). Seventy clients (both laptops and mobile devices) were connected to the system. A script generating random network traffic was running on each client during the evaluation. This allowed to simulate a high network rate activity. Clients had a mean log rate of 10,000 lines per minute. The Engine had an incident rate up to 50 incidents per minute. Our findings suggest that the Engine is able to handle a network of 35 devices, on high rate activity, without any performance degeneration or delay.



Please Wait. System Setting.

Network Incidents

EventID	CreationTime	InstanceOfCase	InvolvedClient
20	2012/07/09-23:43:03	Torrent Download	C1
19	2012/07/09-17:03:09	Torrent Download	C1
18	2012/07/09-13:14:24	Visited Malicious Domain	C0
17	2012/07/09-10:56:42	Visited Malicious Domain	C0
16	2012/07/08-20:13:42	Visited Malicious Domain	C0
15	2012/07/08-17:33:16	DOS	C3
14			C0
13			C0
12			C0
11			C4
10			C1
9			C1
8			C1
7			C0
6			C0
5			C0
4			C0
3			C3
2			C3

Domain 74.125.228.99 info

Domain Name :	iad23s08-in-f3.1e100.net
Whois :	Google Inc.
Black/White Listed :	Not Listed
Closest Bad IP :	74.125.127.100
Same Network as Closest Bad IP :	100 %
Country Of Origin :	United States
Distance from Closest Bad IP :	0.0 miles
Domain Malicious Level :	85 on a scale of 100

Simultaneous Traffic

AlertID	CreationTime					ExternalDomain	WebSite
7	2012/07/09-22:43:30					df-hert8100.net	None
6	2012/07/09-10:23:21	C1	TCP	IN	201.154.53.112	we-xs-rer1200.net	None
5	2012/07/08-15:43:50	C1	UDP	OUT	179.190.55.102	vq-105-DS.net	None
4	2012/07/07-12:44:13	C0	TCP	IN	201.104.53.166	v6506.15hbh00.net	None
3	2012/07/06-08:12:32	C0 C3	TCP	OUT	74.125.228.9	iad23s05-in-f9.1e100.net	None
2	2012/07/05-10:47:13	C1 C2 C3	TCP	IN	173.194.74.95	qe-in-f95.1e100.net	None

Unusual Traffic

AlertID	CreationTime	Client	Protocol	TrafficDirection	SourcePort	DestinationPort	ExternalAddress	ExternalDomain	WebSite
204	2012/07/09-10:56:57	C0	TCP	IN	80	49454	173.194.31.18	None	www.youtube.com
203	2012/07/09-10:56:57	C0	TCP	OUT	49454	80	173.194.31.18	None	www.youtube.com
202	2012/07/09-10:56:57	C0	UDP	IN	53	58182	128.118.124.206	cas-dc-01.aq.psu.edu	None
201	2012/07/09-10:56:57	C0	UDP	OUT	58182	53	128.118.124.206	cas-dc-01.aq.psu.edu	None
200	2012/07/09-10:56:57	C0	TCP	IN	80	49452	74.125.228.99	iad23s08-in-f3.1e10...	ad-q.doubleclick.net
199	2012/07/09-10:56:57	C0	TCP	OUT	49452	80	74.125.228.99	iad23s08-in-f3.1e10...	ad-q.doubleclick.net
198	2012/07/09-10:56:57	C0	UDP	IN	53	61798	128.118.124.206	cas-dc-01.aq.psu.edu	None

Figure 2: ReasONets Graphical User Interface

However, an increase of the number of devices monitored, from 35 to 50 causes a delay of roughly 1 minute (mean values) on data processing by the Engine. The results show a similar behavior for the Network Logger, which has a slightly higher capacity, being able to handle up to 70 machines without recording any performance degeneration. We believe that by using server machines, with more computational power, we will be able to handle a network with at least 100 devices. In large deployments, hardware-based appliances or load balancing approaches to scale to larger networks may be needed.

3. CONCLUSION

We presented ReasONets, a network monitoring system developed as part of an HP-funded research project bridging situational awareness with incident detection systems. The demo will demonstrate the behavior of our prototype in absence of incident first, and after in presence of incidents, both known and unknown.

Acknowledgment: This work was partly supported by Hewlett-Packard Innovation Award 2011, "Analytics for Situational Awareness".

4. REFERENCES

[1] J. M. Estévez-Tapiador, P. Garcia-Teodoro, and J. E. Díaz-Verdejo. Measuring normality in http traffic for anomaly-based intrusion detection. *Computer Networks*, 45(2):175–193, 2004.

[2] R. Guha, O. Kachirski, D. Schwartz, S. Stoecklin, and Y. Yilmaz. Case-based agents for packet-level intrusion detection in ad hoc networks. In *Proceedings of the 17th International Symposium on Computer and Information Sciences*, pages 315 – 320. CRC Press, October 2002.

[3] D. B. Leake. Case-based reasoning. *The Knowledge Engineering Review*, 9(01):61–64, 1994.

[4] R. Perdisci, W. Lee, and N. Feamster. Behavioral clustering of http-based malware and signature generation using malicious network traces. In *NSDI*, pages 391–404, 2010.

[5] T. J. Ross. *Fuzzy Logic*, pages i–xxi. John Wiley Sons, Ltd, 2010.

[6] M. West-Brown, D. Stikvoort, K.-P. Kossakowski, G. Killcrece, R. Ruefle, and M. Zajicek. Handbook for computer security incident response teams (csirts). Technical Report CMU/SEI-2003-HB-002, 2003.