Towards an Early Warning System for Network Attacks Using Bayesian Inference

Kalutarage, H. K. , Lee, C. , Shaikh, S.A. and Sung, F. L. B.

Original citation & hyperlink:
Kalutarage, H. K. , Lee, C. , Shaikh, S.A. and Sung, F. L. B. (2016) Towards an Early Warning System for Network Attacks Using Bayesian Inference. 2015 IEEE 2nd International Conference on Cyber Security and Cloud Computing (CSCloud): 399 – 404
http://dx.doi.org/10.1109/CSCloud.2015.35

DOI 10.1109/CSCloud.2015.35

Publisher: IEEE

Towards an early warning system for network attacks using Bayesian inference

Harsha Kumara Kalutarage and Siraj Ahmed Shaikh
Faculty of Engineering, Environment & Computing
Coventry University, United Kingdom
Email: {ab4359,aa8135}@coventry.ac.uk

Chonho Lee and Francis Lee Bu Sung
School of Computer Engineering
Nanyang Technological University, Singapore
Email: {leechonho,ebslee}@ntu.edu.sg

***Abstract*—The Internet has become the most vulnerable part of critical civil infrastructures. Proactive measures such as early warnings are required to reduce the risk of disasters that can be created using it. With the continuous growth in scale, complexity and variety of networked systems the quality of data is continuously decreasing. This paper investigates the ability to employ Bayesian inference for network scenario analysis with low quality data to produce early warnings. Theoretical account of the approach and experimental results using a real world attack scenario and a real network traffic capture is presented.**

I. Introduction

The critical civil infrastructures depend heavily on the Internet. This is due to increasing shift of typical industrial IT systems to IP based infrastructures which can be easily compromised by an attacker to create a disaster. Proactive measures such as early warnings on computer networks are required to reduce such a risk. Early warning systems (EWSs) aim to detect unclassified but potentially harmful system behaviour based on preliminary indications. Ability to process uncertain and incomplete (low quality) data is a particular emphasis for EWSs [1]. Poor data quality increases due to various reasons such as volume, complex dynamics in communication networks [2] and privacy issues. The problem of learning from low quality data is a relatively new challenge.

This paper investigates the ability to employ Bayesian inference for security scenario analysis with low quality data to produce early warnings. The rest of the paper is organised as follows. Section II describes the methodology underlying our approach. Section III presents a case study followed by results in Section IV. Section V examines some related work. Section VI discusses results and concludes the paper.

II. Methodology

A simple, but a systematic, profile-based method is proposed. A node score is computed to explain the level of suspicious and that score is updated as time progresses. The proposed method combines information gathered from different parameters into a single score as follows.

Let H be the hypothesis that the particular node is an attacker and $I = \{i_1, i_2, i_3, ..., i_n\}$ be a set of mutually independent indicators (features) that can be used to describe an attack attempt. During a smaller time window w, using Bayesian inference, our hypothesis H can be tested as,

$$P(H/I) = \frac{\prod_k P(i_k/H) \cdot P(H)}{P(I)} \tag{1}$$

$$P(\neg H/I) = \frac{\prod_k P(i_k/\neg H) \cdot P(\neg H)}{P(I)} \tag{2}$$

Dividing equation (1) by (2) and taking logarithm,

$$ln\frac{P(H/I)}{P(\neg H/I)} = ln\frac{P(H)}{P(\neg H)} + \sum_k ln\frac{P(i_k/H)}{P(i_k/\neg H)} \tag{3}$$

If $ln\frac{P(H/I)}{P(\neg H/I)} > 0$ then H is true, else H is false. Equation 3 is the well known "Log likelihood ratio". $P(H), P(i_k/H)$ are prior and likelihoods terms while $P(H/I)$ is the posterior probability.

The critical challenge in EWSs is to keep information about node activities over extended period of times to support for continuous monitoring. Aggregating proportion terms, i.e. $ln\frac{P(H/I)}{P(\neg H/I)}$ when ≥ 0, in equation 3 over the time helps to accumulate relatively weak evidence for long periods. These accumulated terms, i.e. $\sum_t ln\frac{P(H/I)}{P(\neg H/I)}$ (t is time), known as node profiles can be used as a measurement of the level of suspicion for a given node at any given time.

III. A Case Study

The key contribution of this paper is our idea that Bayesian inference can be employed for security scenario analysis with low quality data to produce early warnings. A real world attack scenario and a real network traffic capture is used for this analysis.

A. Attack scenario description

Researchers found a catastrophic vulnerability, called heartbleed, in OpenSSL in March 2014 [3]. It allows attackers to read sensitive memory (e.g. cryptographic keys and credentials) from vulnerable servers. The vulnerability lies in the implementation of TLS heartbeat protocol extension. The heartbeat protocol consists of two message types: heartbeat request and heartbeat response which has the following structure [4].

```
struct {
 HeartbeatMessageType type;
 uint16 payload_length;
```

```
 opaque payload[HeartbeatMessage.payload_length];
 opaque padding[padding_length];
} HeartbeatMessage;
```

RFC6520 of Internet Engineering Task Force (IETF) describes the heartbeat protocol [4]. A very brief summary is given below since behaviour of the protocol is relevant to our work. According to the protocol description, it is reasonable to hypothesis that the parameters of heartbeat message population should have certain shapes of distributions, and deviation from the legitimate behaviour may result an outlier.

- A request message can arrive almost at any time during the lifetime of a connection. Whenever a request message is received it should be answered with a corresponding response message.
- A request message should not be sent during handshakes.
- There must not be more than one request message in flight at a time. A message is considered to be in flight until the corresponding response message is received or until the re-transmit timer expires.
- The size of type and payload_length fields are one and two bytes respectively. The sender must use a random padding of at least 16 bytes which must be ignored by the receiver. The total length of a message must not exceed 2^{14} (or max_fragment_length) as defined in RFC6066.
- If the payload_length of a received message is too large then it should be discarded silently.

If a request message is received and sending a response is not prohibited by the protocol, then the receiver must send a corresponding response message carrying an exact copy of the payload of the request by allocating a memory buffer as follows.

```
buffer = OPENSSL_malloc(1 + 2 + payload + padding);
```

The memory allocation is based on a user controlled value payload + padding. There was no length check for this particular allocation (applicable to OpenSSL 1.0.1 or prior) and an attacker can force the OpenSSL server to read arbitrary memory locations specifying higher payload values than the actual payload sending in the request (see figure 1). This vulnerability is called heartbleed. Since there is a maximum boundary for the total length of a heartbeat message, in a heartbleed attack attempt, a higher number of message frequency can be expected during the lifetime of a connection to leak as much as possible data from the server's memory. It should be noted that it is necessary to look at the TLS layer data to detect exploits attempts (exact heartbleed packets) which is not available in our dataset described in next section.

B. Dataset description

For the purpose of demonstration the above scenario is analysed using MAWI dataset [5], which focuses on long term traffic measurement analysis rather than short term security analysis of individual nodes. Traffic traces are collected by tcpdump and have made open to the public after removing privacy information. Hence traffic traces consist only protocol

```
No Time Source      Destination   Protocol Length D.Port
1 0.0  192.168.11.1 192.168.11.128 TLSv1.1  62     443

Frame 1: 62 bytes on wire (496 bits), 62 bytes ..........
Ethernet II, Src: Vmware_c0:00:08 (00:50:56:c0:00:08)....
Internet Protocol Version 4, Src: 192.168.11.1 ..........
Transmission Control Protocol, Src Port: 54848 (54848),
Dst Port: https (443), Seq:1, Ack:1, Len:8
Secure Sockets Layer
    TLSv1.1 Record Layer: Heartbeat Request
        Content Type: Heartbeat (24)
        Version: TLS 1.1 (0x0302)
        Length: 3
        Heartbeat Message
            Type: Request (1)
            Payload Length: 16384
[Malformed Packet: SSL]
```

Figure 1: A heartbleed packet: requested length is higher than the payload.

```
No Time Source        Destination    Protocol Length D.Port
1 0.0  192.168.11.130 192.168.11.128 TCP      74     443
Frame 1: 74 bytes on wire (592 bits), 74 bytes captured
(592 bits)
Ethernet II, Src: Vmware_1e:01:8f (00:0c:29:1e:01:8f),
Dst: Vmware_bf:11:91 (00:0c:29:bf:11:91)
Internet Protocol Version 4, Src: 192.168.11.130
(192.168.11.130), Dst: 192.168.11.128 (192.168.11.128)
Transmission Control Protocol, Src Port: 57534 (57534),
Dst Port: https (443), Seq: 0, Len: 0
    Source port: 57534 (57534)
    Destination port: https (443)
    [Stream index: 0]
    Sequence number: 0    (relative sequence number)
    Header length: 40 bytes
    Flags: 0x002 (SYN)
    Window size value: 29200
    [Calculated window size: 29200]
    Checksum: 0xcb1d [validation disabled]
    Options: (20 bytes), Maximum segment size, SACK permitted,
    Timestamps, No-Operation (NOP), Window scale
```

Figure 2: A packet from MAWI dataset: information available upto TCP headers only.

headers (upto the TCP header information, see figure 2). Protocol payload that contains user data has been removed from the traces. Traffic traces have been collected at several sampling points within a backbone during a 15 minutes time period (from 2.00pm - 2.15pm) on each day.

Readers are invited to notice the limitations of details in the MAWI dataset with respect to heartbleed detection. By nature MAWI dataset is low quality for heartbleed detection due to:

- TLS data is not available, hence exact matching for heartbleed packets is not possible.
- Data is captured during only 15 minutes period on each day, which is a very smaller fraction of the whole traffic exchange.
- IP addresses are anonymous, i.e. each IP maps to an actual address by a particular hash function so there is no background information about nodes available to support for the analysis.

Since our aim is to use low quality data for producing early warnings, MAWI dataset fits with the purpose of this paper.

C. Variable selection

A systematic selection of correct variables is essential and often difficult without proper training to know which

variables are relevant to a given task and which are effectively noise [6], [7]. Many such algorithms look at the problem as "given a variable set of size M, finding the optimal set of size n (n $\ll$ M)" and use variable ranking as the principal of selection [6]. This assumption rarely holds in most of practical data analysing problems on the Internet. Number of available variables relevant to the given scenario is far less than the required number of variables for the optimal selection. Hence an optimal variable selection solution cannot be guaranteed in our case.

In order to apply equation 3 in MAWI dataset, H is defined as the hypothesis that given node (or IP address) is a heartbleed attacker and I is defined using four variables as follows. Within a 443 session from client to server:

i_1 - number of TCP segments
i_2 - upload during a session
i_3 - downward during a session
i_4 - time gap between two consecutive packets

The rationality behind this selection is that these variables are weakly connected (i.e. uncertain and incomplete) to the behaviour of the heartbeat protocol mentioned earlier. Our idea is to compare posterior distributions of these variables as shown in equation 3.

D. Parameter estimation

If a node has performed a scan activity prior to commence a 443 session then that node is considered as dubious. Initial prior belief $p(H)$ for such node is assigned as follows.

$$p(H) = \begin{cases} 0.6, & \text{if performed a scan activity} \\ 0.4, & \text{otherwise} \end{cases} \quad (4)$$

After the initial point, posterior probability at time $(t-1)$ is assigned as the prior belief of time point t.

The likelihood distributions $P(i_k/H)$ and $P(i_k/\neg H)$, for i=1,2,3,4, in equation 3 were estimated using a "malicious" and "clean" dataset respectively. A dataset prior to December 2011 (i.e. before the bug was introduced in Open SSL) was chosen as the "clean" set. The "malicious" set was chosen based on our assumption that there is a higher chance for heartbleed attack attempt during the heartbleed public announcement period. This is due to practical constrains accessing for a sufficiently large known heartbleed dataset.

IV. Results

This section presents the experimental results. Note that a particular emphasis of EWSs is to establish hypotheses and predictions as well as to generate advises in still not completely understood situations. To demonstrate this we will analyse behaviour of network nodes during the timeline of heartbleed public announcement. Due to the space limitation the main stream of heartbleed timeline [8], [9] is listed.

A. Heartbleed timeline

- December 2011 - The bug was introduced to OpenSSL.
- 14th of March 2012 - The bug has been out in the wild since OpenSSL released version 1.0.1.
- 21st of March 2014 or before - Google Security discovers heartbleed vulnerability in OpenSSL.
- 01st of April 2014 - Google Security notifies "OpenSSL team members" about the flaw, OpenSSL says it on social network Google Plus.
- 02nd of April 2014 - Finnish IT security testing firm Codenomicon separately discovers the same bug in OpenSSL.
- 04th of Friday April 2014 - Rumours begin to swirl in open source community about a bug existing in OpenSSL.
- 05th of April 2014 - Codenomicon purchases the heartbleed.com domain name, where it later publishes information about the security flaw. OpenSSL (not public at this point) publishes this (since taken offline) to its Git repository.
- 07th of April 2014 - The National cybersecurity Centre Finland reports Codenomicon's OpenSSL "heartbleed" bug to OpenSSL core team. A new OpenSSL version is uploaded to OpenSSL's web server. Most of the world finds out about the issue through heartbleed.com.
- 08th of April 2014 - NCSCF issues a security advisory on its website in English.

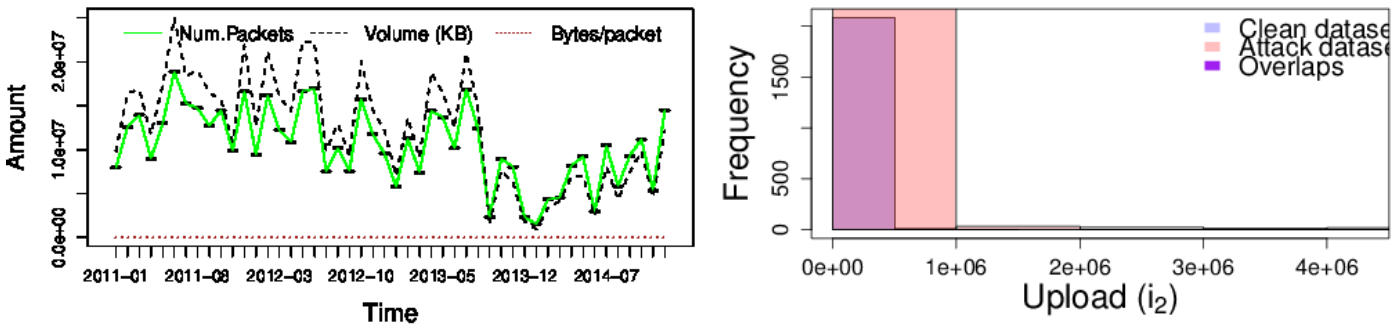
The typical increasing trend of SSL traffic does not affect on individual variables selected for this analysis as they describe properties within a session. But it is interesting to see fluctuations of overall SSL traffic against the heartbleed timeline to get an idea about user's reaction against the announcement. Figure 3a presents the demand for SSL traffic over the time. In fact, we observe slight decrease in demand for the SSL traffic during the heartbleed public announcement days (March/April 2014). This may due to the fear of using SSL during those days[1].

Likelihood distributions of each parameter (i.e. $i_k, k = 1, 2, 3, 4$) is estimated by visual inspection of histograms. The histograms was plotted for each parameter in attack dataset as well as for clean dataset. Figure 3b presents such a histogram plotted for variable i_2.

B. Suspicious nodes

Fifteen minutes duration traces from each day was split into 90 segments (10 seconds smaller windows). Within a window, nodes are profiled using equation (3) and results are presented in figure 4. If a node obtained negative (-) scores throughout the observation period (30 days) then that node is defined as "innocent". If a node obtained at least one positive (+) score during the observation period then that node is defined as a "suspicious" node. Among suspicious nodes, if any node stands out from their peer nodes (i.e. beyond three z-scores) then she is identified as "most suspicious". Zero (0) means the target node has not produced any event that are of

[1] e.g. Tor project advised to stay away from the Internet entirely for the next few days until things settle [10].



(a) Demand for SSL traffic.

(b) Histograms of variable i_2.

Figure 3: Overall traffic trend and estimating likelihood distributions.

interest to this analysis during the observation window. Note that 11 and 322 nodes (out of 9087 nodes) were selected as most suspicious and suspicious nodes respectively during this observation window. Hence the proposed analysis has reduced the search space by 96% in this case which is very welcome. Also it helps to prioritise some nodes than others for the analysis which is essential in security to capture the real attacker as early as possible before doing further damage.

C. Recurrent analysis

To understand the recurrent of the target scenario by the same or different nodes, above analysis is repeated periodically (every two months) since July 2011 to July 2014. Due to the space constraints only results generated during July 2013 to July 2014 is presented in figure 5. Comparing graphs in figures 4 and 5 would give a good picture about node behaviour against the time. For example, during the heartbleed announcement period the number of suspicious nodes is very high (i.e. 333 see figure 4) than other periods (see figure 5).

If there are any general systematic patterns (e.g. trend and seasonality) in the time series of variables in section III-C, such patterns need to be removed from the data before analysing. If not they might increase the false positive rate of the analysis counting such systematic variations in the data to the heartbleed account. For example, an increasing trend was observed in variable i_3 over the time in our dataset and that trend was removed using differencing. This may be due to clients use https for general purposes such as social media and video streams as many http servers switching into https over the time. Techniques like auto-correlation and differencing help to remove such kind of general dependencies in the data to make other hidden patterns more apparent and relevant.

V. Related work

A malware warning centre is proposed in [11] which uses a Kalman filter to detect a worm's propagation at its early stage in real-time. An architecture of an automatic EWS is discussed in [12]. Authors aim to provide predictions and advice regarding security threats without incorporation of cognitive abilities of humans. Optimal sensor placement strategies for EWS is discussed in [13] which studies correlation between attack patterns of different locations (national and international) and explores how sensors should be located accordingly. The Internet Malware Analysis System (InMAS) aims for distributed, large-scale monitoring of malware on the Internet [14]. InMAS integrates diverse tools for malware collection analysis. All collected information is accessible through a web interface. An infrastructure and organisational framework for a situation awareness and early warning system for the Internet is presented in [15]. It aims for sharing, correlating and cooperatively analysing sensor data collected from number of organisations located in different geographical locations. eDare (Early Detection, Alert and Response system) [16] and the Agent-based EWS [17] also focus on early warnings. The Internet Motion Sensor, a globally scoped Internet monitoring system aims to measure, characterise, and track threats [18]. It statistically analyses dark net traffic that needs to be interpreted by humans. DeepSight intelligence collects, analyses and delivers cyber-threat information through a editable portal and data feeds, enabling proactive defensive actions and improved incident response [19].

An extensive survey of collaborative intrusion detection proposals can be found in [20]. Existing systems support for information aggregation, visualisation and statistical analysis. But predictions and advice generation is left to the human user. Disregarding the attack type, its characteristics and analysis method, they all depend on a common feature, information sharing and central collection. We look at this dependency as a major constraint for advances in EWSs on the Internet. The motivation behind choosing a probabilistic approach for profiling in our work is that a network event is not always easy to judge for malicious nature. Some suspicious events can appear as part of an attack signature as well as originate from benign network activity. Such uncertainty needs to be acknowledged [21]. Using Bayesian technique and its variants for intrusions detection can be found in [22], [23]. The relevance of information fusion for network security monitoring has been widely discussed [24]. Learning from imbalanced data [25] is a similar line of research supports the idea of this paper. Note that our approach maintains only number of profile scores equals to the number of nodes in the network. Hence our approach is truly scalable in terms of storage and can be deployed in high-speed networks for live analysis.

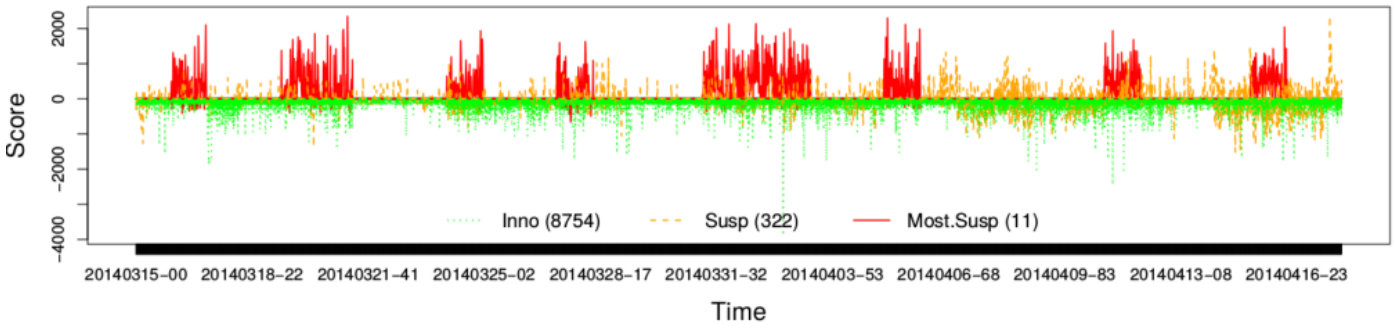


Figure 4: Monitoring from 15.03.2014 to 16.04.2014: the graph presents the node score against the timeline.

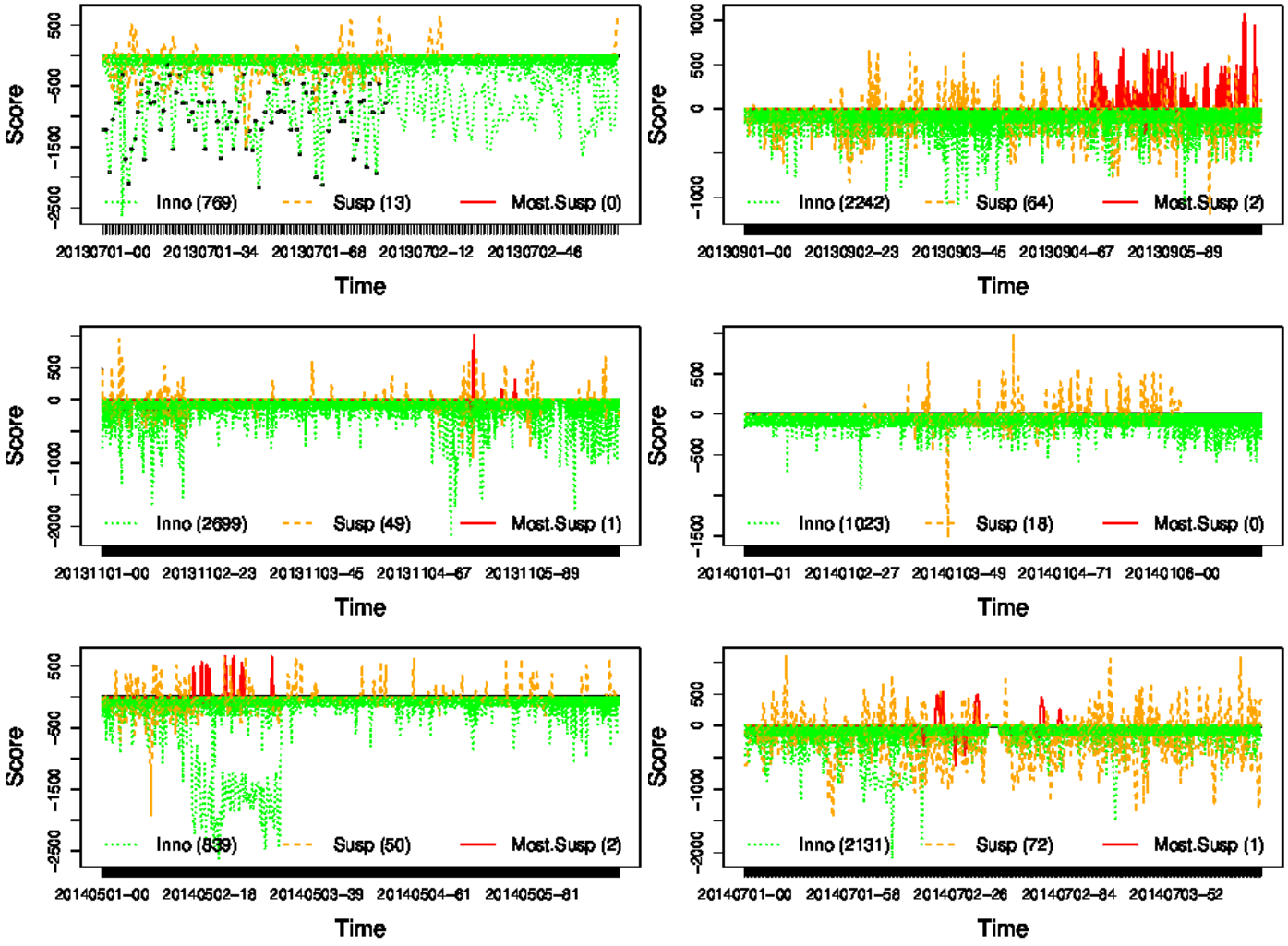


Figure 5: Monitoring from July 2013 to July 2014.

VI. Discussion and conclusion

Most traditional security analysing methods seldom use any of the quantitative information that could be gathered from alternative sources which may have direct or indirect connection (e.g. scan activities in this work) to the target hypothesis. Using a Bayesian approach in this work facilitates it. The proposed method deviates from the idea of "information sharing and central collection", but provides automated analysis necessary to support for the analysts. Sharing and central collection is a major barrier for advances in EWSs. In the context of security, data and information sharing is difficult between different organisations and nations. Our method does not entirely depend on "partners" (i.e. different organisations in different geographical locations) as in [15]. However we encourage sharing between partners as such information can be incorporated in the profiling algorithm to improve its precision. But it has the ability to analyse the situation using "whatever data in hand" and provide alerts subject to different confidence levels.

Profiling strategy in this work allows raising alarms on anomalous behaviours that are not by themselves anomalous in any single indicator. This helps to achieve a high level of sensitivity and specificity in our work. It reduces false positive rates, and hence reducing investigation costs and distracting attention from the actual malicious activities. Since the data amounts to millions of data objects, a few percent of false alarms can make analysis overwhelming for an analyst. Therefore, even a smaller amount of reduction to false alarm rates is welcome. In particular, reduction to the search space by 96% should be acknowledged.

The evaluation of the proposed method depends on speculative reasoning rather than an empirical evaluation. One of the weaknesses of above analysis is choosing the attack dataset without proven attack attempt to estimate the distribution of $P(i_k/H)$. It is based on our assumption that there is a higher chance for attack attempts during public announcement period. The quality of the entire analysis depends on this assumption though it is a reasonable one. However it is not a methodological weakness of this approach, and only due to a practical constraint accessing a sufficiently large known malicious (heartbleed) dataset. In such a situation, if the historical rate of occurrences of certain attacks is known, it can be used to estimate the likelihood that certain events derive from such attacks or it may be sufficient to quantify these frequencies by an expert in a similar way to estimating risk likelihoods to an accuracy of an order of magnitude [22].

References

[1] J. Biskup, B. Hämmerli, M. Meier, S. Schmerl, J. Tölle, and M. Vogel, "2. 08102 working group – early warning systems," in *Perspectives Workshop: Network Attack Detection and Defense*, ser. Dagstuhl Seminar Proceedings, G. Carle, F. Dressler, R. A. Kemmerer, H. König, and C. Kruegel, Eds., no. 08102. Dagstuhl, Germany: Schloss Dagstuhl - Leibniz-Zentrum fuer Informatik, Germany, 2008. [Online]. Available: http://drops.dagstuhl.de/opus/volltexte/2008/1493

[2] D. K. Arrowsmith, R. Mondrag, and M. Woolf, "Data traffic, topology and congestion," in *Complex Dynamics in Communication Networks*. Springer, 2005, pp. 127–157.

[3] C. Lee, L. Yi, L.-H. Tan, W. Goh, B.-S. Lee, and C.-K. Yeo, "A wavelet entropy-based change point detection on network traffic: A case study of heartbleed vulnerability," in *Cloud Computing Technology and Science (CloudCom), 2014 IEEE 6th International Conference on*. IEEE, 2014, pp. 995–1000.

[4] R. Seggelmann, M. Tuexen, and M. Williams, "Transport layer security (tls) and datagram transport layer security (dtls) heartbeat extension," *IETF draftietf-tls-dtls-heartbeat-00 (June 2010)*, 2012.

[5] K. Cho, K. Mitsuya, and A. Kato, "Traffic data repository at the wide project," in *Proceedings of the annual conference on USENIX Annual Technical Conference*. USENIX Association, 2000, pp. 51–51.

[6] I. Guyon and A. Elisseeff, "An introduction to variable and feature selection," *The Journal of Machine Learning Research*, vol. 3, pp. 1157–1182, 2003.

[7] J. Bins and B. A. Draper, "Feature selection from huge feature sets," in *Computer Vision, 2001. ICCV 2001. Proceedings. Eighth IEEE International Conference on*, vol. 2. IEEE, 2001, pp. 159–165.

[8] B. Grubb, "Heartbleed disclosure timeline: who knew what and when," *Online], Retrieved on 15th Sept*, 2014.

[9] "The heartbleed bug," http://heartbleed.com/, accessed: 2015-05-20.

[10] "Openssl bug cve-2014-0160," https://blog.torproject.org/blog/openssl-bug-cve-2014-0160, accessed: 2015-05-20.

[11] C. C. Zou, L. Gao, W. Gong, and D. Towsley, "Monitoring and early warning for internet worms," in *Proceedings of the 10th ACM conference on Computer and communications security*. ACM, 2003, pp. 190–199.

[12] M. Apel, J. Biskup, U. Flegel, and M. Meier, "Towards early warning systems–challenges, technologies and architecture," in *Critical Information Infrastructures Security*. Springer, 2010, pp. 151–164.

[13] J. Göbel and P. Trinius, "Towards optimal sensor placement strategies for early warning systems." in *Sicherheit*, 2010, pp. 191–204.

[14] M. Engelberth, F. C. Freiling, J. Göbel, C. Gorecki, T. Holz, R. Hund, P. Trinius, and C. Willems, "The inmas approach," 2010.

[15] B. Grobauer, J. I. Mehlau, and J. Sander, "Carmentis: A co-operative approach towards situation awareness and early warning for the internet." in *IMF*, 2006, pp. 55–66.

[16] Y. Elovici, A. Shabtai, R. Moskovitch, G. Tahan, and C. Glezer, "Applying machine learning techniques for detection of malicious code in network traffic," in *KI 2007: Advances in Artificial Intelligence*. Springer, 2007, pp. 44–50.

[17] K. Bsufka, O. Kroll-Peters, and S. Albayrak, "Intelligent network-based early warning systems," in *Critical Information Infrastructures Security*. Springer, 2006, pp. 103–111.

[18] M. Bailey, E. Cooke, F. Jahanian, J. Nazario, D. Watson *et al.*, "The internet motion sensor-a distributed blackhole monitoring system." in *NDSS*, 2005.

[19] Symantec, "Cyber security: Deepsight intelligence," http://www.symantec.com/deepsight-products/, June 2015.

[20] C. V. Zhou, C. Leckie, and S. Karunasekera, "A survey of coordinated attacks and collaborative intrusion detection," *Computers & Security*, vol. 29, no. 1, pp. 124–140, 2010.

[21] H. K. Kalutarage, S. A. Shaikh, I. P. Wickramasinghe, Q. Zhou, and A. E. James, "Detecting stealthy attacks: Efficient monitoring of suspicious activities on computer networks," *Computers Electrical Engineering*, pp. –, 2015. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0045790615002384

[22] H. Chivers, J. A. Clark, P. Nobles, S. A. Shaikh, and H. Chen, "Knowing who to watch: Identifying attackers whose actions are hidden within false alarms and background noise," *Information Systems Frontiers*, vol. 15, no. 1, pp. 17–34, 2013.

[23] K. Das and J. Schneider, "Detecting anomalous records in categorical datasets," in *Proceedings of the 13th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2007, pp. 220–229.

[24] V. Chandola, A. Banerjee, and V. Kumar, "Anomaly detection: A survey," *ACM Comput. Surv.*, vol. 41, no. 3, pp. 15:1–15:58, Jul. 2009. [Online]. Available: http://doi.acm.org/10.1145/1541880.1541882

[25] H. He and E. A. Garcia, "Learning from imbalanced data," *Knowledge and Data Engineering, IEEE Transactions on*, vol. 21, no. 9, pp. 1263–1284, 2009.