International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
Article Received: *25 May 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

Identifying Malicious Hosts Involved in Periodic Communications Using Machine Learning

Prof. Priyanka R. Raval[1], Prof. Sheetal J. Nagar[2]
[1]Computer Engineering Department, Government Engineering College, Rajkot, Gujarat, India.
priyankaraval.gec@gmail.com
[2]Computer Engineering Department, Government Engineering College, Rajkot, Gujarat, India
nagar.sheetal@gmail.com

Abstract— Network intrusion detection systems still have a lot of space for improvement after years of research. This work presents a novel approach to the automatic and timely analysis of traffic produced by big networks, which can detect malicious external nodes even when their actions trigger no alarms in the defence mechanisms now in place. Since our experimental evaluation indicates that periodic communications are more closely associated with harmful actions and may be readily incorporated with other detection systems, that is the focus of our suggestion. We point out that intermittent network activity can happen over a wide range of times, from seconds to hours. As a result, it can be difficult to analyse large time rooms of traffic produced by big businesses in a timely manner. While the approach presented in this research tries to discover external nodes that are likely implicated due to malicious interaction, existing work focuses exclusively on botnet identification. The output of the proposed method is a manageable Suspected List of external sources that are distinguished by a significantly higher probability of being malicious in comparison of the entire set of external nodes contacted by the analysed network, given that network actions linked to malware can be viewed as uncommon occurrences in the overall traffic. Our proposal's usefulness is demonstrated by a comprehensive evaluation on real huge network traffic. It can automatically choose only a few dozen suspect hosts out of more than thousands, allowing network administrator operators to focus the analysis on a small number of likely hostile targets.

Keywords: *Cyber security, Intrusion Detection System (IDS), Alert verification, Alert prioritization, Alert Fusion, Machine learning (ML).*

I. INTRODUCTION

There are two main issues with large-scale information system defence. Security analysts are overburdened by the massive number of logs flooded daily by network interactions [2], while attackers are able to evade detection by executing assaults over extended periods of time and utilizing sophisticated strategies [1]. Moreover, most Network Intrusion Detection Systems (NIDS) tend to generate multiple false alarms [3] or fail to identify new types of assaults [1]. The goal of proposals to boost NIDS efficacy is to either make it easier for them to identify threats [4] or to give the safety analysts clear information regarding assaults that are currently in progress [5, 8]. Some remedies rely on internal nodes that are probably infected for prioritizing [9].

The goal of this study is to identify external nodes that are attacking the network under observation, even if their actions do not trigger a Network Intrusion Detection System (NIDS) warning. This will enable automatic security evaluations. By analysing network flows, the suggested method can automatically produce a Suspected List of a small number of external nodes that have a far greater probability of being harmless than all the external nodes that the network is monitoring. Finding hosts engaged in periodic communications, commonly known as beaconing, after various time periods is the aim. Harmful beaconing activity detection is yet an unsolved research issue [10]–[12], which is made more challenging in big networks by the challenge of accurately and quickly analysing massive amounts of network traffic. Furthermore, we have conducted experiments to confirm that harmful actions are more common on external nodes with periodic connections than on hosts with irregular communication patterns. Our new program examines network flows to identify periodic behaviours. It can identify potential evasion attempts by classifying as periodic even signals that do not follow a rigid periodic schedule. A comprehensive set of trials are conducted on a sizable, actual network without the generation of artificial traffic are used to assess our idea.

We describe a novel approach that, given the whole collection of external nodes being conversed by the monitored

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
Article Received: *25 May 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

organization, automatically generates a suspected list of nodes that are highly to be involved in harmful beaconing operations. We propose to utilize network flow clustering techniques. The two primary domains of associated research are harmful beaconing activity detection and NIDS alarm optimization.

Since each NIDS creates a large number of alerts that are frequently too numerous for human operators to manually review, a number of solutions try to enhance the information provided to security analysts by providing more thorough, shorter recordings. The authors of [6] go over a method that clusters alerts made by similar harmful acts in order to lower the number of alarms generated by numerous NIDSs. Some studies, like [5], suggest clustering alarms to identify their underlying causes. Techniques for internal nodes prioritization are suggested in more recent research. Multistep attacks are the main topic of [13] and [8]'s authors. The plan in [14] takes advantage of the warnings given by the most important resources. A suggested architecture in [9] gives internal nodes a higher priority based on how likely it is that they may be participating in certain destructive cyberattacks. The overarching objective of all these publications is to assist security analysts by enabling them to concentrate on the most pertinent alerts that NIDS has found. On the other hand, even if an intrusion alarm is not raised by an external node's behaviour, our solution uses a combination of intrusion warnings, network flow analysis, and clustering algorithms to rectify the most suspect external nodes.

In the subject of botnet detection, one well-known issue is the identification of harmful beaconing activity. Assuming that bots inside the same botnet exhibit comparable network behaviours, Gu et al. [15] develop a framework for identifying internal nodes that are part of botnets by clustering network traffic. The authors of [16] intend to use supervised machine learning methods on network flows to find hosts that are infected with botnets by analysing the essential components of command-and-control exchanges. A comparable method is put out in [12], however its primary objective is to identify Command and Control servers rather than bots.

Our approach can identify any potential external danger that is engaging in beaconing operations, not just malware connected to botnets. In contrast to recommendations pertaining to botnets, we refrain from assuming any attributes about the traffic that is being studied. While the approach in [11] depends on the examination of both DNS and web-proxy logs, related work, like [10], examines DNS related logs to find harmful beaconing operations carried out by internal nodes. However, our goal is to identify hostile external nodes, which is a more challenging task given that a major business could interact with lacs of external nodes on a daily basis.

Unlike [12], [16], our solution uses an unsupervised machine learning technique and is based on the study of network flows, which is easily collected and stored [17]. Its execution time on a large network is also suitable with online traffic studies.

II. IDENTIFICATION OF MALICIOUS EXTERNAL NODES

The first subsection of this section provides a high-level overview of the suggested methodology, while the remaining subsections provide specifics on each processing module.

A. Overview

Providing a suspected list of external nodes engaged in recurring interactions with a high probability of malicious activity is the primary goal. The fundamental premise is that, despite the likelihood that new attack variants may avoid network intrusion detection [18], certain characteristics of infected network behaviour endure and possibly employed to detect potentially harmful activity.

The suggested approach utilizes two readily available inputs in contemporary infrastructures: security alarms produced by a signature-based NIDS and network flows associated with interactions between internal and external nodes. The three modules process these inputs. Network communications that take place on a regular basis between external and internal nodes are detected by the Periodicity Detector. Periodic connections are grouped by the Aggregator based on how their networks behave. The final suspected list of questionable external nodes is produced by the Suspected list Builder.

Given that periodic events can occur at granularities ranging from a few seconds to hours, detecting them in enterprise networks with an ever-increasing number of linked devices is becoming an increasingly difficult issue. Rather than examining patterns in unprocessed data, we take into account network flows that provide combined metadata that summarizes pertinent aspects of network traffic. A flow record is a set of unidirectional packets sharing certain network features, like the information regarding the sender and receiver’s ports numbers, the transport layer protocol type, and the IP addresss of the sender and receiver. In the field of cybersecurity, using network flows as input sources is common practice [17], since it reduces storage space requirements, speeds up analysis, and lessens privacy issues because it does not require packet-specific payloads.

While NIDSs are an invaluable tool for identifying malicious

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
Article Received: *25 May 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

activity, they are not able to identify new malware variants without a known signature. However, other aspects of malware behavior, like beaconing, remain consistent across a large range of malware variants that are generated automatically, leading to communication patterns that are identical. Different malware versions are likely to be clustered together since our technique groups network messages that exhibit similar periodic characteristics. Our method simply needs one malware variation to cause an NIDS alert in order to flag all of the cluster's periodic interactions having that variant as suspicious.

B. Detection of Regularity

In order to identify periodic communications from network flows, the Periodicity Detector module first creates time series from the infected records. Next, it uses an autocorrelation technique to analyse these time series and determine whether or not they are periodic. The employed methods are resilient and can withstand disturbances brought about by noise or intentionally presented by an intruder to evade detecting of the threats.

Two hosts' network flow sequences show an irregularly spaced time series that is not immediately usable to identify periodic conversations. Therefore, for every pair of internal and external nodes that exchange packets within a time frame W, we first compute one equally spaced time series. This time series has an aggregate of W/P values for a sampling period P. Every component is constructed by summing up all of the network flows that take place during the same sampling time between the involved hosts. Because beaconing activities necessitate recurrent data exchanges, we compute each time series element by totalling the bytes transferred through the participating nodes over the relevant sampling period in order to capture these data interactions. We can more effectively distinguish between beaconing operations that exchange varying amounts of data thanks to this design decision. Following this stage, one time series is connected to every two of internal and external nodes.

Next, as autocorrelation may identify time series with many periods, we use it to identify periodicities in each time series [19]. An autocorrelation function (ACF) with W/P values, each of which represents the similarity of the time series with a delayed replica of itself, is obtained by calculating the auto-correlation on a time series. It is possible to ascertain whether or not a time series exhibits periodicities by analysing the local maxima of the ACF. Finding the positions of local maxima is especially important when searching for periodicities in the ACF, as precisely periodic time series typically have high-amplitude local maxima at the start of the associated ACF.

Only precisely periodic time series can be identified by previous research (e.g., [20]) that use this technique to find periodicities in time series. The problem associated with the proficient attackers might provide some disruptions to evade detection, for as by predicting or delaying the conversations or by arbitrarily altering the volume of data exchanged at each interaction. In addition, network traffic can be affected by noise that arises from inactivity intervals, temporary disconnections, retransmissions of various packets due to slow connections or other causes, and other network-based irregularities.

In order to overcome these problems, we suggest a novel algorithm that can identify as periodic even time series that don't follow a compulsory periodic pattern. The prominent idea behind noisy periodic time series cannot be identified by conventional methods because their limited amplitude of local maxima makes them difficult to identify. This is because noisy periodic time series are defined by limited amplitude of local maxima. In contrast, it exhibits several local maxima with comparable amplitudes for aperiodic time series, and large amplitudes in between a local maximum and its subsequent local minimum. We include two criteria in the ACF in order to produce a more adaptable approach to identify periodicities with noise.

C. Aggregator

The Aggregator groups together recurring messages that follow comparable trends. To our knowledge, this is the first work that suggests using clustering methods to detect communications with similar periodic behaviour, even though various clustering approaches have already been used in the information related security domain. This work is divided into two stages: first, we compute the spectrogram of each periodic time series by utilizing the Discrete Fourier Transform (DFT); subsequently, we use the spectrograms as input for a hierarchical based clustering technique.

A spectrogram can be produced by using the DFT on a periodic time series. Periodic time series can appear radically different, even though their spectrograms show the same profile. This makes the representation helpful in explaining the way network communications behave. The issue is that each spectrogram's form is also influenced by the quantity of data transferred between the participating hosts. For instance, even though the frequency components of two hosts that communicate 1MB of data on a regular basis are the same, their spectrograms will have smaller amplitudes than those of another pair of hosts that frequently exchange 10MB. In order to resolve this problem, we normalize each spectrogram's amplitudes between 0 and 1.

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 7s
Article Received: *25 May 2023* ***Revised:*** *12 June 2023* ***Accepted:*** *30 July 2023*

D. Suspected List Builder

The Suspected List Builder module generates the final Suspected List of dangerous external nodes. By mapping NIDS warnings into clusters of related periodic messages, it first finds malicious clusters of periodic communications. To be more precise, clusters that have at least one transmission that triggered network intrusion alert are classified as harmful; this procedure enables to identify malicious hosts even in cases where the NIDS has not detected them. The final Suspected List is then filled with all of the external nodes that are extracted by this module and belong to malicious clusters. A dendrogram is the result of the hierarchical clustering process. Objects that resemble one other can be grouped together by severing the dendrogram at a specific height h. In order to decrease intra-cluster variance and enhance inter-cluster variance, we adjust the parameter h. After completing this stage, we are left with a variable number of periodic communication clusters that exhibit comparable patterns. These clusters are then fed into the Suspected List Builder module.

III.EXPERIMENTAL EVALUATION

A. Experimental Test Set

The suggested technique is verified using actual traffic produced over the course of a week by a sizable network with close to 10,000 hosts, or over 500 billion network flows. Security operators used and configured the most recent rulesets [23] on NIDS equipped with Suricata [22] to monitor outgoing traffic. The most significant testbed metrics for each day of the week under consideration are shown in Table I.

Table I Traffic Analysis of Each Day of the Dataset.

Day taken	Unique external nodes	Network incidents
1	105 884	53 500 389
2	89 283	47 789 977
3	298 241	101 314 287
4	314 313	110 875 503
5	249 768	99 359 716
6	258 439	106 304 916

B. Experimental Results

Every day, the identification framework is put into action. The objective is to show that it can generate a comprised list of external nodes those are Suspected and have a significantly greater probability of being harmful than the original set of external nodes that were contacted. Furthermore, we demonstrate that compared to aperiodic communications, the frequency of malevolent external nodes engaging in periodic interactions is significantly higher. Lastly, we show that even external nodes that did not trigger network intrusion alert are included in the Suspected List and that our method's execution duration is consistent with internet traffic assessments. First, we evaluate the proportion of malicious external nodes among all the external nodes that the network under observation has made contact with. Subsequently, we allow the Periodicity Detecting module to produce time series and identify those that exhibit periodicity. We report the findings of the validation procedure carried out on these two sets of nodes in Table III to show that the rate of harmful external nodes participating in periodic interactions is significantly greater than the rate of malicious hosts involved in aperiodic interactions. The number of external nodes engaged in periodic and aperiodic interaction is reported for each field. We find that the average percentage of harmful external nodes that communicate on a regular basis is 2.7%, whereas the average percentage of hosts that communicate irregularly is 0.51%. Our choice to concentrate on this group of hosts is supported by these findings, which indicate that periodic interactions exhibit a higher rate of maliciousness than aperiodic interactions. These findings also suggest that harmful external communications may be regarded as uncommon occurrences in the total traffic, which encourages us to work toward creating a manageable Suspected List where there is a greater chance of discovering a malicious host.

Table II Verification of External Nodes Involved Aperiodic Communications.

Day taken	External nodes	Harmful external nodes
1	2284	59 (2.58%)
2	2123	53 (2.49%)
3	3194	74 (2.31%)
4	3288	91 (2.77%)
5	3044	80 (2.63%)
6	3034	90 (2.97%)

Both time series show periodic behaviour, despite considerable noise, as we can see. More precisely, the nodes linked to the first time series interchange roughly 1KB of data every 32 minutes, while the nodes linked to the second time series exhibit three periodic behaviours, as shown by the exchange of roughly 5.2KB of data every 5 hours and approximately 3.2KB and 4.3KB of data every 32 minutes.

These findings show that even periodic communications impacted by certain perturbations can be detected by our

method. Furthermore, we note that the spectrograms are extremely similar even though they have separate data exchanges, which accounts for their clustering together. This result suggests that our normalized DFT-based method is resilient to changes in the volume of exchanged data and offers a good approximation of a time series' periodic characteristic.

IV. CONCLUSIONS

An automatic technique for detecting malicious periodic contacts with remote hosts is presented in this paper. Security analysts can concentrate on a select few targets as the output is attainable Suspected List of external nodes that have a far greater probability of being harmful than the total set of interacted hosts. Our proposal's effectiveness is demonstrated by a thorough study using actual traffic data from a big enterprise, verified by other sources. It can identify malicious hosts even when no NIDS alarm is raised. The suggested approach is easily integrable into any detection system, deployable even on very large networks, and compatible with a wide range of detection methods.

REFERENCES

[1] "Mandiant M-Trends 2015." https://www2.fireeye.com/rs/fireye/ images/rpt-m-trends-2015.pdf, visited in Jun. 2017.

[2] R. Zuech, T. M. Khoshgoftaar, and R. Wald, "Intrusion detection and big heterogeneous data: a survey," Journal of Big Data, 2015.

[3] H.-J. Liao, C.-H. R. Lin, Y.-C. Lin, and K.-Y. Tung, "Intrusion detection system: A comprehensive review," Journal of Network and Computer Applications, 2013.

[4] L. Portnoy, E. Eskin, and S. Stolfo, "Intrusion detection with unlabeled data using clustering," in DMSA, 2001.

[5] K. Julisch, "Clustering intrusion detection alarms to support root cause analysis," TISSEC, 2003.

[6] R. Perdisci, G. Giacinto, and F. Roli, "Alarm clustering for intrusion detection systems in computer networks," Engineering Applications of Artificial Intelligence, 2006.

[7] F. Valeur, G. Vigna, C. Kruegel, and R. A. Kemmerer, "Compre- hensive approach to intrusion detection alert correlation," IEEE TDSC, 2004.

[8] M. Marchetti, M. Colajanni, and F. Manganiello, "Framework and Models for Multistep Attack Detection," IJSIA, 2011.

[9] F. Pierazzi, G. Apruzzese, M. Colajanni, A. Guido, and M. Marchetti, "Scalable architecture for online prioritisation of cyber threats," in IEEE CyCon, 2017.

[10] A. Shalaginov, K. Franke, and X. Huang, "Malware beaconing detection by mining large-scale dns logs for targeted attack identification," in WASET ICCISIS, 2016.

[11] X. Hu, J. Jang, M. P. Stoecklin, T. Wang, D. L. Schales, D. Kirat, and J. R. Rao, "Baywatch: Robust beaconing detection to identify infected hosts in large-scale enterprise networks," in IEEE DSN, 2016.

[12] L. Bilge, D. Balzarotti, W. Robertson, E. Kirda, and C. Kruegel, "Disclosure: detecting botnet command and control servers through large-scale netflow analysis," in ACSAC, 2012.

[13] F. Manganiello, M. Marchetti, and M. Colajanni, "Multistep attack detection and alert correlation in intrusion detection sys- tems," Information Security and Assurance, pp. 101–110, 2011.

[14] S. Noel and S. Jajodia, "Optimal ids sensor placement and alert prioritization using attack graphs," Journal of Network and Systems Management, 2008.

[15] G. Gu, R. Perdisci, J. Zhang, W. Lee et al., "Botminer: Clustering analysis of network traffic for protocol-and structure-independent botnet detection." in USENIX Security Symposium, 2008.

[16] F. Tegeler, X. Fu, G. Vigna, and C. Kruegel, "Botfinder: Finding bots in network traffic without deep packet inspection," in ACM CoNEXT, 2012.

[17] T. Chakraborty, F. Pierazzi, and V. Subrahmanian, "Ec2: Ensem- ble clustering and classification for predicting android malware families," IEEE TDSC, 2017.

[18] P. J. Brockwell and R. A. Davis, Introduction to time series and forecasting. Taylor & Francis, 2002, vol. 1.

[19] G. Gu, J. Zhang, and W. Lee, "Botsniffer: Detecting botnet command and control channels in network traffic." in NDSS, 2008.

[20] J. Benesty, J. Chen, Y. Huang, and I. Cohen, "Pearson correlation coefficient," in Noise reduction in speech processing. Springer, 2009.

[21] "Suricata IDS," http://suricata-ids.org/, visited in August 2017.

[22] "Emerging Threats.net Open rulesets." https : / / rules . emergingthreats.net/, visited in Aug. 2017.

[23] "VirusTotal," https://www.virustotal.com, visited in Aug. 2017.