Accelerating Cyber-Breach Investigations through Novel use of Artificial Immune System Algorithms

Benjamin Donnachie[1][0000−0002−4771−5770], Jason Verrall[1][0000−0002−2905−2360], Adrian Hopgood[2,1], Patrick Wong[1], and Ian Kennedy[1]

[1] The Open University, Walton Hall, Milton Keynes, United Kingdom
[2] University of Portsmouth, Winston Churchill Ave, Portsmouth, United Kingdom
{benjamin.donnachie, jason.verrall, adrian.hopgood, patrick.wong, ian.m.kennedy} @open.ac.uk

Abstract. The use of artificial immune systems for investigation of cyber-security breaches is presented. Manual reviews of disk images are impractical because of the size of the dataset. Machine-learning algorithms for detection of misuse require labelled training data, which are generally unavailable. They are also necessarily retrospective, so they are unlikely to detect new forms of intrusion. For those reasons, this article proposes the use of artificial immune systems for unsupervised anomaly detection. Specifically, a deterministic dendritic cell algorithm (dDCA) has been implemented that has successfully detected automated SQL injection attacks from sample disk images. For comparison, it outperformed an unsupervised k-means clustering algorithm. However, many significant anomalies were not detected, so further work is required to refine the algorithm using more extensive datasets, and to encode complementary expert knowledge.

Keywords: Anomaly detection · Artificial Immune Systems · Cybersecurity · Dendritic Cell Algorithm · Unsupervised Learning.

1 Introduction

The hostile penetration of computer systems is an increasing security concern for organisations globally. Unauthorised access to a computer system ("cyberbreach") can either prevent it from doing something it should and/or cause it to do something it should not[12].

Following a cyber-breach, investigators are often under extreme pressure to deliver results quickly to meet regulatory or business timelines or to identify enhancements to bring systems back online. These investigations require the analysis of significant volumes of log information, in a process that is often manually intensive, inefficient, and liable to confirmation bias.

While more traditional log sources, such as firewall logs, intrusion detection system logs, and audit logs offer a partial view of any intrusion, a more detailed view can be obtained by examining the filesystems of affected computer systems.

One common method is assembly of a “super-timeline” of filesystem events with a specialist tool, such as Plaso[8], that typically includes metadata from:

1. Filesystem including file birth (created) times, access & change times
2. Extracted from known filetypes; e.g. system logs, office documents etc.
3. Additional plugins to detect potential or known malicious software.

Such output can be overwhelming with a simplified disk image for training human examiners[9] generating over 1.3m entries, consisting of all recognised log metadata over the lifetime of the system, compounding the analysis workload.

This paper presents a novel unsupervised anomaly detection method to prioritise cyber-breach investigations, using these complex super-timelines. Section 2 reviews selected previous work in this area; Section 3 describes the proposed approach; Section 4 details preliminary experimental results; and Section 5 concludes the paper with suggestions for future work.

2 Related Work

Existing investigation methods involving event reconstruction from log files, have disadvantages such as prior training requirements, or are significantly time-consuming[3]. Techniques also exist for identifying potentially malicious file content, such as fraudulent documents, but these do not identify anomalous behaviour[6]. Within the field of intrusion detection, two principal approaches to detecting potentially malicious behaviour from real-time log information are reported: misuse detection vs anomaly detection[4].

In addition to expert systems, Machine Learning and Artificial Intelligence (ML/AI) methods have been used to explore both detection approaches. However, similar to detecting malicious file content, misuse detection requires previous training; with models suffering from class imbalance with training data biased towards the non-attack activities[13]. Anomaly detection can demonstrate more flexibility by not requiring prior training for an application.

3 Proposed Approach

3.1 Artificial immune systems

Artificial Immune Systems (AIS) are a branch of ML/AI modelling inspired by the human immune system. AIS algorithms can encompass Danger Theory, whereby the immune system responds preferentially to signals (antigens) from tissue undergoing uncontrolled cell death (necrosis), typically arising from injury or infection; leading to development of a Deterministic Dendritic Cell Algorithm (dDCA), based on immunological antigen-presenting Dendritic Cells (DCs)[7].

dDCA requires two input signals:

Safe: Within the physiological analogue, an indicator of benign apoptosis. Digitally, an indicator of normal system behaviour, e.g. a (reasonably) constant rate of computer system activity.

Danger: Physiologically, an indicator of necrosis. Digitally a measure of potential abnormality, e.g. anti-virus alerts or known malicious activity.

While there is no need to define, or train, a normal pattern of activity, dDCA requires domain expert knowledge about the application to define these signals.

3.2 Deterministic Dendritic Cell Algorithm

As illustrated in Fig. 1, digital DCs begin in an immature, or initialised, state. Each super-timeline time window is presented as a separate antigen, acting as the vector of calculated safe and danger signals to locate any relevant activity. dDCA then fuses the input data across multiple time windows, individual DCs become either 'semi-mature' under the influence of safe signals and hence not provoking an immune response (normal behaviour); or 'mature' under the influence of danger signals and invoking a response (abnormal/malicious behaviour).

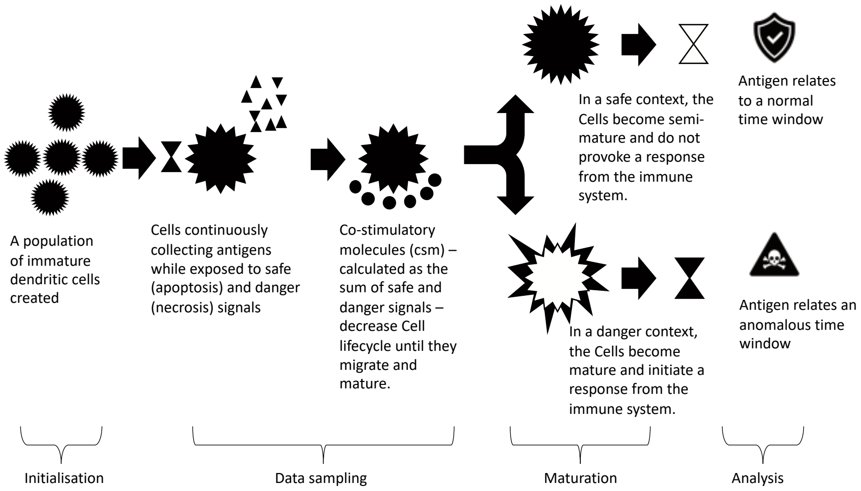
A population of immature dendritic cells created
Cells continuously collecting antigens while exposed to safe (apoptosis) and danger (necrosis) signals
Co-stimulatory molecules (csm) – calculated as the sum of safe and danger signals – decrease Cell lifecycle until they migrate and mature.
In a safe context, the Cells become semi-mature and do not provoke a response from the immune system.
Antigen relates to a normal time window
In a danger context, the Cells become mature and initiate a response from the immune system.
Antigen relates an anomalous time window
Initialisation
Data sampling
Maturation
Analysis

Fig. 1. Illustration of dDCA showing phases and data fusion (based on [5]).

3.3 Methodology

Inspired by work using k-means clustering algorithm with minimal features for incident response triage[11], this feasibility study calculates the safe and danger signals derived from the super-timeline data sampled into fixed-time windows. Log activity levels are used as a proxy for system activity, and thus rate of change, to calculate the safe signal as shown in equation (1). A yara rule repository[1] used by cybersecurity researchers to share malicious software patterns was used as an indicator of potential malicious activity, and derive the danger signal (2).

$$\text{For each time window } i:$$

$$Safe_i = 100 - \left[\frac{100 \log \left(|activityCount_{i+1} - activityCount_i| \right)}{\log \left(\max activityCount \right)} \right] \tag{1}$$

$$Danger_i = \frac{100 \log \left(yaraCount_i \right)}{\log \left(\max yaraCount \right)} \tag{2}$$

Plaso[8] was used to extract all recognised metadata from the training image discussed above, and derive safe and danger signals as outlined in Fig. 2.

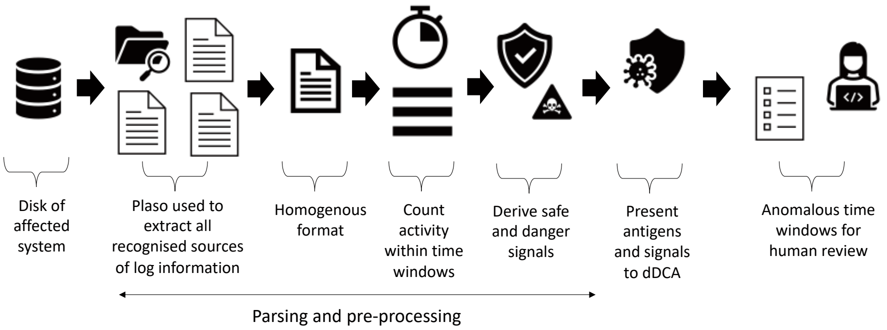
Disk of affected system
Plaso used to extract all recognised sources of log information
Homogenous format
Count activity within time windows
Derive safe and danger signals
Present antigens and signals to dDCA
Anomalous time windows for human review
Parsing and pre-processing

Fig. 2. dDCA algorithm implementation steps

The calculated safe and danger signals were presented sequentially to the dDCA algorithm (based on Dr Greensmith's C implementation - with thanks) with the time window pointer (i) presented as the reference antigen. The calculated anomaly score k_α was collated with values > 0 indicating the likelihood of an anomaly or malicious event[7].

For benchmark comparison, the same dataset of activity and yara counts was clustered using the unsupervised k-means clustering algorithm.

4 Experiment Results and Application

Significant events which human examiners are expected to uncover were collated[14] and manually confirmed, as detailed in Table 1. Times indicate when malicious activity began; detections may be lagged to reach a threshold. Table 2 shows an extract of all anomalous detections from running dDCA, as detailed above, together with the results from the k-means algorithm.

Comparison between the tables show this experiment successfully detected the automated SQL injection attacks that began at 11.15 on 2015/09/02 in the 11:19 time window. However, many significant items were not detected; this is likely to be due to the lack of necessary expert knowledge encoded into this experiment. This implementation also out-performed the k-means algorithm, which only detected software installs and did not cluster any malicious activity.

Table 1. Start of significant events contained within the training image.

Date	Time	Event Description	Comments
2015-09-02	07:10	Webserver reconnaissance activity begins	**Correctly** not detected
2015-09-02	09:04	Begin command injection attacks; create users and add to remote desktop group	Not detected
2015-09-02	09:31	Local files exploited through webserver	Not detected
2015-09-02	10:49	SQLi attacks begin	Not detected
2015-09-02	11:15	Begin SQLmap to automate SQLi attacks	**dDCA detection during activity**
2015-09-02	11:25	Malicious webshells created and executed	Not detected
2015-09-03	07:14	Further malicious webshells dropped	Not detected
2015-09-03	07:21	Use of webshell to execute commands	Not detected

Table 2. Filtered potentially anomalous time windows identified by dDCA and *k*-means.

	Summary values		Input		Output		
Time window (i)	Activity count	Yara count	Danger	Safe	dDCA (k_α>0)	k-means cluster	Comments
2015-07-20 12:55	3404	1731	64	29	7	-	Software install
2015-08-23 21:25	8191	7043	77	21	35	-	Software install
2015-08-23 21:26	3688	2859	69	26	51	-	Software install
2015-08-23 21:41	39920	32802	90	7	76	**Abnormal**	Software install
2015-08-23 21:42	75072	61673	95	8	155	**Abnormal**	Software install
2015-08-23 21:43	90163	69738	96	16	65	**Abnormal**	Software install
2015-08-23 21:44	1098	852	58	0	16	-	Software install
2015-08-24 06:52	8577	7537	77	22	34	-	Software install
2015-08-24 06:57	3172	2650	68	29	9	-	Software install
2015-08-24 07:43	66032	64979	96	3	91	**Abnormal**	Software install
2015-08-24 07:44	38167	34035	90	10	160	**Abnormal**	Software install
2015-08-24 07:45	16248	15855	84	12	59	-	Software install
2015-08-24 07:46	28446	27138	88	17	112	-	Software install
2015-08-24 07:47	41360	36091	91	17	57	**Abnormal**	Software install
2015-08-24 07:48	113678	105019	100	2	96	**Abnormal**	Software install
2015-08-24 07:49	38707	37593	91	2	88	**Abnormal**	Software install
2015-08-24 07:50	1113	897	59	8	132	-	Software install
2015-09-02 11:19	13481	9530	79	29	22	-	**Malicious SQLmap**

5 Conclusions and further work

This study demonstrates the potential of dDCA to triage vast volumes of log data for human review, identifying 18 potential anomalies from a dataset of 1,381,976 log entries with non-malicious anomalous detections potentially discounted using expert domain knowledge; and performing better than unsupervised *k*-means clustering algorithm.

The anomalous indications correlate with peaks in activity across the disk images; due partly to the derivation of safe and danger signals from overall activity, and also because the images are intended to train human breach investigators rather than accurately simulate malicious activity.

To address the limitations of deriving the input signals, further research is needed to determine a more effective way to encode domain expertise, potentially combining other types of ML within hybrid AI systems[10]. Training sets exist for network traffic, albeit with limitations[2], whilst datasets from real-world intrusions are difficult to obtain. Consequently, generating datasets which reflect real-world cyber-attack patterns are essential for the evaluation of future work.

References

1. Repository of yara rules (Mar 2022), https://github.com/Yara-Rules/rules
2. Al-Daweri, M.S., Zainol Ariffin, K.A., Abdullah, S., Md. Senan, M.F.E.: An Analysis of the KDD99 and UNSW-NB15 Datasets for the Intrusion Detection System. Symmetry **12**(10), 1666 (2020)
3. Bhandari, S.: Research and implementation of timeline analysis method for digital forensics evidence. Ph.D. thesis, Kaunas University of Technology (2022)
4. Çavuşoğlu, Ü.: A new hybrid approach for intrusion detection using machine learning methods. Applied Intelligence **49**(7), 2735–2761 (2019)
5. Costa Silva, G., Palhares, R.M., Caminhas, W.M.: A Transitional View of Immune Inspired Techniques for Anomaly Detection. In: Intelligent Data Engineering and Automated Learning, vol. 7435, pp. 568–577. Springer, Berlin, Heidelberg (2012)
6. Du, X., Le, Q., Scanlon, M.: Automated Artefact Relevancy Determination from Artefact Metadata and Associated Timeline Events. International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2020 (2020)
7. Gu, F., Greensmith, J., Aicklein, U.: The dendritic cell algorithm for intrusion detection. In: Lio, P., Verma, D. (eds.) Biologically Inspired Networking and Sensing. IGI Global (2012)
8. Guðjónsson, K.: Mastering the Super Timeline (2010), https://bit.ly/3DjJpSf
9. Hadi, A.: Web Server Case (Sep 2015), https://bit.ly/3LbYR4z
10. Hopgood, A.A.: Intelligent systems for engineers and scientists, fourth edition. CRC Press, Oxon, UK (2022)
11. Nila, C., Patriciu, V.: Taking advantage of unsupervised learning in incident response. In: 12th International Conference on Electronics, Computers and Artificial Intelligence. IEEE, Bucharest, Romania (2020)
12. Price, B., Tuer, J.: Digital Forensics. In: White, P. (ed.) Crime Scene to Court 4th ed: The Essentials of Forensic Science, chap. 12. R. Soc. Chem., London (2016)
13. Singhal, A., Maan, A., Chaudhary, D., Vishwakarma, D.: A Hybrid Machine Learning and Data Mining Based Approach to Network Intrusion Detection. In: 2021 International Conference on Artificial Intelligence and Smart Systems. pp. 312–318. IEEE, Coimbatore, India (Mar 2021)
14. Swartwood, A.: Web Server Case Write-up (Mar 2017), https://bit.ly/3eLWY2g