Data farming for cyber security : an agent-based modelling approach

Pan, Jonathan

2016

Pan, J. (2016). Data farming for cyber security : an agent based modelling approach. International Journal of Information Privacy, Security and Integrity, 2(3), 197-215. doi:10.1504/IJIPSI.2016.078590

https://hdl.handle.net/10356/144712

https://doi.org/10.1504/IJIPSI.2016.078590

Data Farming for Cyber Security: An Agent Based Modelling Approach

Abstract— Increasingly organizations that deploy security defences are challenged by advanced Malware's persistent evasive intrusions. Malware are continually evolving to evade detection while detection technologies trail behind. Cyber security analytics provides promising possibilities to enable defenders to catch up. However there are challenges for cyber security analytics development. The unknown risks originating from ever evolving cyber attack patterns pose a significant challenge to cyber security analytics development to maintain the effectiveness of their analytics algorithms. My research proposition to deal with this challenge in cyber security analytics development is to use data farming techniques to produce data containing varied simulated conditions. This in turn will facilitate cyber security analytics development. Data Farming is used in military strategic planning to identify possible unknowns and in turn develop defensive countermeasures. The proposition entails using Agent Based Modelling to simulate the computing environment involving various actors including the Malware. The output of the model with simulated scenarios is generated data (or farmed data) that contains weblog network behaviour information. These farmed data can then be used to facilitate cyber security analytics development. In this paper, details of the proposed Agent Based Model simulator developed using Netlogo will be covered. The data generated by the model is verified using known anomaly detection statistical techniques as part of model verification.

Index Terms—Model Development, Invasive software (viruses, worms, Trojan horses)

———————————— ◆ ————————————

1 Introduction

The threat to cyber security is increasingly becoming more challenging for the solution developers of cyber defences. This is because the malicious products from the cyber attackers (or Malware) are becoming very effective in evading detection. Perhaps the biggest cause of the trailing effects of cyber defence technology advancement is because of the narrow focus and openness of the designs of cyber defences, hence giving the adversaries the asymmetric advantage. The latter will have a large field area to manoeuvre around the cyber defences and the surprise element advantage to apply targeted attacks against yet to be discovered defence vulnerabilities. Recently cyber security analytics has become a focal area of development by many security defence solution providers as the defensive detection mechanism to level off the fight. However the problem with cyber security analytics is that it has the same challenge as with all past cyber security defence development. Specifically, cyber security analytics typically is developed based on known behaviour records or data of cyber threats. For example, known behaviours gathered through Malware analysis of data exfiltration behaviour from the infected computers. What about the unknowns to facilitate the development of cyber security analytics ? (Meza et al., 2009) argued that to facilitate development of cyber security analytics, data is an important element. However how do we gather data for unknowns ?

Data Farming (Brainstein and Home, 1998) is a technique used by military strategists to identify unknown conditions or scenarios to facilitate the development or validate the effectiveness of countermeasures. This research proposition is to use an Agent Based Model to produce web surfing network traffic behaviour data through simulation scenario runs. The model simulates web surfing network traffic behaviour in networked computing environment with a configurable simulated Malware. The model also generates the legitimate and malicious web surfing log data induced by interaction between model's agents. In this paper, these agents are the computing nodes, humans using the computing nodes, operating system software running on computing nodes and the malicious Malware that has infected computing nodes. The Malware induces web bound traffic from infected computing nodes. The other non-malicious agents in the model will also generate legitimate web surfing activities. To verify the model and its output, the data generated will be tested using known cyber security statistical based anomaly detection analytics. The contribution of this research work is to have a cyber security analytics data farming tool that produces simulated data to facilitate the development and evaluation of cyber security analytics algorithms for known and unknown Malware behaviours. In this paper, only known Malware behaviours are simulated though the model can be extended to handle the unknowns.

In the next section of this paper, related research work is discussed. This is followed by the need for an appropriate cyber security analytics development environment and the need to

generate unknown scenario are discussed. The design of the proposed Agent Based Model to perform data farming for cyber security analytics is then covered. This is followed by a description of the analytical techniques used to verify the model. An analysis of the verification results will be covered subsequently. Finally future research options and conclusion end the paper.

2 Related Work

There is a number of related research work done thus far. However I believe that the proposed research work is novel to address the problem of acquiring the needed data for analytics development. The following are the related research work.

2.1 Cyber Security Analytics Development Environment

There is much research done to address the cyber security analytics problem. Examples include (Xie et al., 2010) proposition to use Bayesian networks for cyber security analytics and (Tsai and Chan, 2007) with their work in using probabilistic models to detect cyber security threats from weblogs. (Mayo et al., 2014) reported that their work to develop a tool to simulates the massive Internet and to subsequently develop theoretical models to develop and test hypotheses that includes the 'What-If' question to reboot the Internet if it was taken offline and handling of current and future cyber threats. Their work showed potential to address this research question, however their large platform of close to 5,000 computing nodes to simulate 106 virtual nodes may prohibit ease of development of new algorithms due to extensive investment into such facility. Hence my research proposition will still have relevance to early or primarily stages of cyber security analytics development.

2.2 Agent Based Modelling for Cyber Security Informatics

The Agent Based Modelling (ABM) simulation approach has been used in a varied manner to address different cyber security research problems. One of which was the use of Agent Based Modelling to simulate and evaluate Malware containment strategies (Pan and Fung, 2012). (Kotenko et al., 2010) developed a framework and model to simulate botnets to aid in the investigation of such malice and in the development of various cooperative distributed defence approaches to deal against such threats. (Asman et al., 2011) used ABM to simulate tactical computing networks deployed in military settings can be compromised. Their model simulated various network attacks, provide observations of network attack traffic with network security protocols, assess the resulting impact and quantify the damage of lost data. Their research had the same intent as this research except they did not use the ABM's output for subsequent analytics development. They argued that the advantage of ABM to simulate their research problem over discrete-event simulation model is that ABM focuses on the relationship between entities from the bottom-up that starts at the network packet level, rather than from a top-down approach. This is the same reason why ABM was chosen to address the research problem.

2.3 Data Farming

Data farming is a process of executing a simulation with many parameters that simulates the real world environment in a computer typically high performance computer. The simulation should facilitate 'What-If' scenarios through its parameter configuration. The result of the data farming is the data that can be analysed for trends, anomalies and insights from the multiple scenario runs. (Horne and Meyer, 2010) explored the application of data farming on defence application specifically in the Modelling of social networks to support the efforts in countering of improvised explosive devices. (Friman and Horne, 2005) argued that network centric military operations that involve extensive use of networking technologies are difficult to quantify and evaluate. This is further complicated with the need to consider possible scenarios and outcomes. They proposed solving this with Data Farming with Agent Based Modelling that allowed them to generate many scenarios with different settings configured with the model. They even advocated

that their work could be further extended to deal with scenarios in Information Age that is the intent of this research proposition.

3 Need for Cyber Security Analytics

The benefits from the use of cyber security analytics have garnered much attention in recent times. However there are notable challenges to have the needed supporting development environment for cyber security analytics development.

3.1 Development Environment

There has been a significant amount of research work done to develop algorithms or mathematical models for cyber security analytics in order to improve the chances of detecting cyber security threats or intrusions. However to support the advancement of analytics development, a conducive development environment with tools and data is required. The National Institute for Science and Technology (NIST) recently published their roadmap for improving Critical Infrastructure Cybersecurity (NIST, 2014). Their report stated that in the area of data analytics, they cited the lack of "taxonomies of big data; mathematical and measurement foundations; analytic tools; measurement of integrity of tools; and correlation and causation". The report further adds that the most challenging problem with data analytics in cyber security is the privacy and public confidence concerns associated with the data used in the development. Hence analytics development will need to deal with the concerns of data privacy.

(Haas et al., 2011) argued that analytics that assess current situations can be done with current historical data repositories like database that provides the information to the known risks. However in order to perform comprehensive risk assessment and in turn the development of analytics to support such comprehensive coverage, there is a need to have a development environment with its data with the flexibility to vary configurable variables that supports 'What-If' analysis. An example of a 'What If' analysis is whether an organization's cyber security defences is able to handle more than one form of Malware infection concurrently. Hence these highlights the challenging need for a development environment with the data needed to address the complex analytics development while addressing the privacy concerns.

3.2 Data For Unknown

Today's advanced Malware incorporates many sophisticated techniques in order to exploit yet to be identified vulnerabilities, evade detection through a wide variety of obfuscation techniques and stealthily carry out its malicious acts without raising any alerts. These threats use custom developed cyber weapons to circumvent and exploit weaknesses against the target and its deployed defence (Pan and Fung, 2010). This poses a significant challenge for cyber security solution developers to identify or detect the evasive threat behaviours and yet to be discovered patterns originating from advanced cyber security adversaries.

The 9/11 terrorist attacks against USA surprised many military strategists, military research development is now focused on developing ways to identify all possible scenarios even scenarios that could not be conceived by military strategists. One such approach is Data Farming that involves generating variety of data containing varied scenarios that will subsequently be tested against defensive mechanisms. (Brandstein and Horne, 1998) argued that data farming is needed to deal with the limitation of historical records, while being rich in details, that lacks the technical means to identify as many possible outcomes. (Horne and Meyer, 2004) also argued data farming offers the opportunity to discover outliers, surprises and non-linearity in systems dynamics. Hence this paper's research proposition is to explore data farming for cyber security analytics.

4 Research Proposition

The research proposition is to use Agent Based Modelling (ABM) to perform data farming. ABM is a computation model that simulates the actions and interactions of autonomous agents. ABM supports 'What-If' analysis through configurable elements of the models. Hence

ABM suits well to simulate a cyber security environment involving multiple agents. The agents in a networked environment include legitimate software running on computing nodes as well as malicious ones. Aside from the software agents, humans will interact with the computing nodes and its corresponding software for web surfing. The malicious software, namely the Malware, will interact with infected host operating system and humans.

The Agent Based Model for this research work was developed using Netlogo (Wilensky, 2009). The desired objective for the model is to simulate web surfing activities in a typical computing network environment comprising of operating system web activities and users' surfing activities. Additionally, and more importantly that uniquely defines this model, is to model Malware behaviour when interacting in the network environment connected to the Internet. The model seeks to simulate the Malware bent on exfiltrating data from the infected computing nodes to the Internet. The model was designed with the intent to enable varied Malware characteristics to simulate the rudimentary Malware that generates high or 'loud' network traffic signature and the advanced evasive Malware behaviour that will obfuscate its network activities in order to evade conventional detection. The model generated logs with proxy like formatted web surfing entries. These output are the farmed data.

4.1 Construct of Simulation Model

The model has four simulation agents. The first represents the computing node. These agents hold the identity information about the computing nodes specifically the IP address and an unique identifier. The other three agents are dependent on the first set of computing node agents to exist. The relationship between the computing node agent to the other three agents is bi-directional one to one relationship. The following UML diagram illustrates the premise.

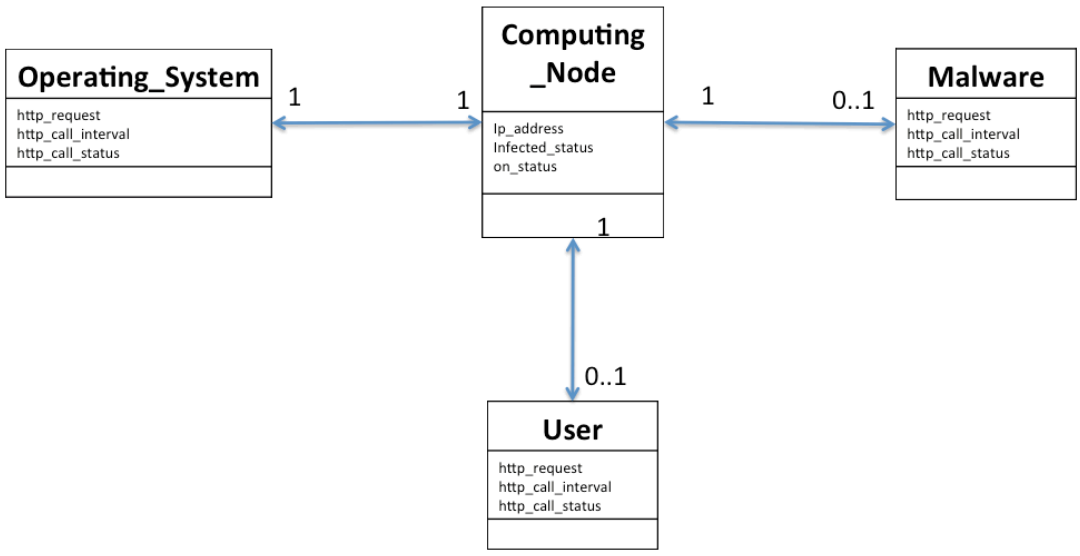

Fig. 1. UML Model of the Agent Based Model

In this model, each computing node agent is always associated with one operating system agent and one user agent. The operating system agent is always active as long as the computing node exists. The operating system agent initiates connection to the Internet to check for updates. In this model, it connects to a specific URL via HTTP to check with software updates like updates to operating system. Such connections happen periodically and in a random manner. Such activities continue as long as the computing node agents are active. For the user agent,

unlike the operating system agent, is not always active. This is to simulate typical user activities as a series of periods of web surfing activities and inactivity. However the user agent is only active when the computer node is active. The return payload from the user's web surfing is randomly assigned.

One Malware agent may be associated with a computing node agent. The association is randomly assigned from the set of computing node agents. The total number of Malware agent is configurable in the model's GUI interface. The Malware agents surfing activities are configurable in this model in order to simulate the variety of Malware characteristics. In the current version of the model, a fully configured Malware is capable of mimicking its network behaviour like that of known advanced Malware's characteristics (Ferguson, 2012), (Embleton et al., 2008), (Chong, 2013) with the intent to exfiltrate data to the Internet. Hence the egress traffic volume is modeled while the ingress is randomly assigned. Details of the Malware configuration options will be covered later. The following table illustrates the configuration options developed in this model to simulate the web surfing network behaviour environment.

TABLE 1
Agent Based Model Configuration

Configuration	Type	Description
Node_Count	Range Slider	This defines the number of computing nodes to be simulated.
Infected_Nodes	Range Slider	This defines the number of randomly selected computing nodes to be infected.
Start_Time	Input (Edit Box)	This is for the start time in Unix Time or POSIX time.
Duration	Input (Edit Box)	This is for the duration of the simulation run in Netlogo's time ticks.
Evade	Switch (Boolean Radio Button)	This defines the Malware agent's behaviour to use evasive URLs and protocol.
Hide	Switch (Boolean Radio Button)	This defines the Malware agent's behaviour to hide its activities users' activities.
Throttle	Switch (Boolean Radio Button)	This defines the Malware agent's behaviour to throttle its activities in terms of egress traffic.

Each agent intending to connect to the Internet uses one specific URL however it can be easily enhanced to support a variety of URLs. At each time tick of the model simulation, all agents are called to post their pre-defined log entry into the log file as part of the data farming function. In Netlogo, the agents are called as a collective and in a random manner, hence the agents' log postings will differ in sequence in un-deterministic manner. The following Pseudo Code describes the output function that invokes each and every agent to generate its traffic.

```
Output_Log_Entries()
{
        Select One Agent from the collective {
                For that Agent {
                        Add random time interval to Global Clock;
                        Output Internet bound traffic payload to Log;
                        }
}
```

Fig. 2. Pseudo Code for Log Output

The model generates a proxy like formatted web access log files. The format is derivative of the SQUID (open source caching proxy for web surfing) access log format with an extension to include amount of egress data for each connection attempt. The format is in CSV format that enables ingestion of the data into analytical tools. The log file is generated at each run of the simulation model. The following is a sample of the generated log contents with its field header.

```
UTC_Time,Elapsed,Duration,Src_IP,Cache_Status,Ingress,Egress,HTTP_Method,URL,Status1,HTTP_Status,St
atus2
1393374621.548,468,192.168.0.15,TCP_MISS,678,621,GET,http:\\www.safesite.com,100,200,TEST
1393374621.627,11,192.168.0.13,TCP_MISS,2710,469,GET,http:\\www.safesite.com,100,200,TEST
1393374621.647,432,192.168.0.31,TCP_MISS,4558,895,GET,http:\\www.safesite.com,100,200,TEST
1393374621.683,136,192.168.0.26,TCP_MISS,5054,89,GET,http:\\www.safesite.com,100,200,TEST
1393374621.684,422,192.168.0.26,TCP_MISS,238,728,GET,http:\\updates.operatingsystem.com,100,200,TEST
1393374621.703,272,192.168.0.10,TCP_MISS,248,316,GET,http:\\www.safesite.com,100,200,TEST
1393374621.767,195,192.168.0.9,TCP_MISS,8188,572,GET,http:\\www.safesite.com,100,200,TEST
1393374621.832,290,192.168.0.20,TCP_MISS,812,485,GET,http:\\www.safesite.com,100,200,TEST
```

Fig. 3. Sample Contents of Generated Log File

4.2 Qualification of Simulation Model

There are a few qualifications and assumptions included in this model. First, the model allows only one leaf agent to be associated with the computing node like one Malware or user to one computing node. However the model can be easily modified to allow for multiplicity in associations. For example, associating two or more Malware agents with one computing node agent. The model does not simulate the propagation of Malware. The Malware operates within its assigned computing node. In this model, all agents has unrestricted network connectivity to the internet.

The current limitation of the model is that it only simulates simple web surfing activities and not streaming traffic hence activities like video or audio streaming is not included. However such websites like Youtube website would generate different form of traffic patterns that are different from surfing static websites. Additionally other HTTP based non-browser application initiated traffic is not considered aside from the operating system checks for updates. The model uses a pseudo random generator built in Netlogo that in turns affects the degree of randomizeness of its generated log. Hence there will be some notable repeatable patterns.

5 Methodology

The following details how the model was evaluated to verify the model's farmed data.

5.1 Evaluation Approach

The model was made to simulate a networked environment and in turn produced the data farmed log files. The verification of data farmed log files was done by applying statistics based anomaly detection analytics. The objective of the verification is to assess the developed model whether it generated the intended data. In the first verification scenario, the ABM simulated Internet surfing activities generated by user and operating system agents on computing nodes that had Internet access. There was no Malware activity simulated in this scenario. This scenario shall be called '*No Infection*' scenario. The second scenario was about having the model simulate an unsophisticated Malware typically found in the web. The Internet bound traffic characteristics generated by the Malware include continuous attempts to access known blacklist URLs using non HTTP ports to exfiltrate data from the infected computing nodes. This scenario shall be called '*Rudimentary Infection*'. The third and final scenario was to emulate the evasive behaviour of an Advanced Persistent Threat (APT) Malware that obfuscated its Internet bound activities from detection or remediation (Virvilis et al., 2013). This simulated advanced Malware hid its activities behind user's web surfing activities, used commonly used protocols and URLs, and throttled its egress traffic to limit the chances of detection. This scenario shall be called '*Advanced Infection*'. The advanced Malware characteristics were codified into the model's Malware agent as configuration switches to regulate the intended Malware behaviour for Malware agents in every run of the model. The model ran three times to produce the three mentioned scenarios. In each run, the model completed its run on standard Mac Air configuration in less than a minute. The logs generated had about 13,000, 20,000 and 15,000 log entries for the '*No Infection*', '*Rudimentary Infection*' and '*Advanced Infection*' scenarios respectively. The difference in the log entry size was due to the log contribution by the Malware and the characteristics of the Malware.

In order to verify the model, statistical anomaly detection techniques were used. Machine Learning is frequently used in the research of anomaly detection in network traffic patterns however for this research, such techniques would not be suitable in the pretext of verifying the data farming model. There are two commonly used forms of statistical anomaly detection techniques in cyber security analytics. They are volume based detection and feature based detection. For each form of anomaly detection, two techniques were used.

5.2 Evaluation – Volume Based Ratio

Ratio analysis is a simple volume based analysis. Ratio has been proposed previously to aid in the detection of network traffic anomaly (Kim, Na, Jang, 2006), (Zhang et al., 2012). In this research, ratio was computed using the total egress amount of data transmitted from the computing node over the total ingress amount of data received into the computing node. The ratio value is a real number. If the ratio value is below one, it indicates that the computing node received more data as compared to what it sent out. A high ratio value that is greater than the numerical value of one is a good probability that the computing node is infected. The Malware would be exfiltrating voluminous amount of data through the proxy.

5.3 Evaluation – Volume Based Cumulative Control

The CUSUM (or cumulative sum control chart) is a statistical sequential analysis technique developed by E. S. Page of the University of Cambridge (Page, 1954). This volume based technique is used as a statistical non-parametric quality control analysis. It is typically used for monitoring change detection. CUSUM involves the calculation of a cumulative sum. CUSUM has been used by (Wang et al., 2002) to detect for DDoS attack patterns. (Lu and Tong, 2009) applied CUSUM to detect network anomalies where Intrusion Detection Systems failed to detect. The CUSUM equation used in this research is as follows:

$$S_m = \sum_{i=1}^{m} (\overline{x}_i - \hat{\mu}_0) \tag{1}$$

The sample number m in this research is the entire population log entries generated by the model. $\hat{\mu}_0$ is the estimated in-control mean. $\overline{x}_i$ in this research is each and every egress traffic value.

In this research, CUSUM analysis was done observing the characteristics of the egress traffic specifically the rate of change and extent of egress traffic traversing through the proxy. A comparative analysis on the collective computing nodes was done to observe any anomalies from the cumulative egress traffic. Hence a probable anomaly would be a notable larger variance in cumulative curve from the collective.

5.4 Evaluation – Feature Based Signature

Signature analysis is a commonly used feature based technique. It is used by intrusion detection or prevention solutions 0. For example, it is used with the Security Information Event Management system (SIEM) or Intrusion Prevention System (or IPS). It is used extensively to detect web based attacks (Kruegel and Vigna, 2003). In this research, the feature that observed was the type of protocol used specifically whether it is non-HTTP protocol. Another feature used here was the detection of access to malicious URLs through the blacklist signature of URLs. Hence when an attempt to access a blacklisted URL will result in a detection made.

5.5 Evaluation – Feature Based Entropy

Entropy measurement is a more advanced feature based analysis. This involves the measure of the extent of randomness in the variable under consideration. Entropy feature analysis has been used by various researches to demonstrate its ability to detect traffic anomalies (Nychis et al., 2008), (Wagner and Plattner, 2005) . Here entropy measurement is

applied to the egress traffic volume to each URL site. The entropy equation used in this research for the finite sample size is as such.

$$H(X) = \sum_{i} P(x_i) \log_{10} P(x_i) \tag{2}$$

$P(x_i)$ represents the probability mass function. The assumption here is that if a person surfs online, the egress traffic volume to each URL site would vary significantly resulting in a high entropy value. However a coded software, in this case a Malware that programmatically throttles its egress traffic, will exhibit a lower entropy value, hence leading to the high possibility that a malicious software may be transmitting data out.

5.6 Model Configuration For Scenarios

In order to simulate these three scenarios, the following configurations were used.

TABLE 2
Model Configuration

Configuration	Value	Description
Node_Count	36	This was constant throughout this research.
Infected_Nodes	0 or 6	Value 0 was used with 'No Infection' scenario and 6 for other two scenarios.
Start_Time	1393375301.000	This value equates to 24 Feb 2014 00:41:41 GMT. This value was an arbitrary value but was the initial value. It changed after simulation run as overall duration was added to the start time.
Duration	2000	2000 ticks constant duration. This will amount to about 10 mins simulated time with a simulation run time of 1 min.
Evade	Off or On (Depending on the scenario)	'On' for 'Advanced Infection Scenario' and 'Off' for the other two scenarios.
Hide	Off or On (Depending on the scenario)	'On' for 'Advanced Infection Scenario' and 'Off' for the other two scenarios.
Throttle	Off or On (Depending on the scenario)	'On' for 'Advanced Infection Scenario' and 'Off' for the other two scenarios.

In this research, there were one constant and varying variables. The constant was assigned to the number of computing nodes variable [Node_Count]. The duration of each simulation run is the number of ticks defined in the variable [Duration]. The simulated duration after each simulation run is the number of ticks and the randomly assigned duration for each Internet bound traffic by each agent. This in turn will alter the [Start_Time] variable is the start time of simulation run in real world time. This value is represented in POSIX time value. In the three scenarios, the simulation run lasting 2000 ticks took less than one min to finish. The simulated duration registered in the logs was about 10 minutes.

Four configurations were varied to generate the three scenario outputs. The following summarizes the simulation configuration settings for the three scenarios.

TABLE 3
Summary Configuration For Scenarios

Scenario	Infected Host	Evade	Hide	Throttle
No Infection	0	Off	Off	Off
Rudimentary Infection	6	Off	Off	Off
Advanced Infection	6	On	On	On

The first configuration [Infected_Node] represents the number of infected nodes. For the first scenario of '*No Infection*', this configuration was set to zero to simulate the condition that none of the computing nodes were infected. For the other two scenarios, this configuration was set to the value of 6 to simulate six infected computing nodes. The random selection of 6 computing nodes was done once and applied to the two scenarios. This was done to ensure consistency when applying model verification. The 6 computing nodes had the following IP addresses.

- 192.168.0.10
- 192.168.0.11
- 192.168.0.15
- 192.168.0.18
- 192.168.0.30
- 192.168.0.33

When [Infected_Node] is set to zero, there will be no infected computing nodes and the generated proxy log file will only contain non-malicious web surfing activities by the user (accessing 'htttp://www.safesite.com') and operating system (accessing 'http://updates.operatingsystem.com'). If this configuration is set to greater or equal to one (for the number of infected nodes), there will be malicious activities generated from these Malware. The characteristics of the malicious activities will depend the next three configurations for Malware behaviours.

The second configuration [Evade] is whether the Malware will hide its attempt to access the Internet by using HTTP protocol (or port 80) which will likely to be left unblock and least monitored by security appliance like the Firewall as this protocol is used by users and the operating systems. The Malware will also use the short URL to circumvent the use of blacklisting of URLs to block access attempts. Recent advanced Malware uses such techniques to hide its activities (Ferguson, 2012), (Chong, 2014).

The third configuration [Hide] is to instruct the Malware agents to hide all its activities behind users' activities. The Malware agents remain dormant until users' activities are noted. This is exemplified through a recent analysis of a Malware (Chong, 2013) that only invokes itself from 'slumber' when the user's presence is noted on the infected computer through the multiple clicks of the mouse.

The fourth configuration [Throttle] that instructs the Malware agents to control the amount of data being exfiltrated and to refrain themselves from generating excessive burst or high voluminous of egress activities. Malware had been noted to exhibit such capabilities in order to hide their network activity signatures (Embleton et al., 2008).

6 Analysis

The following is the analysis of the experiments of the evaluation techniques applied to all three scenarios.

6.1 Evaluation – Volume Based Ratio

For the first test scenario whereby there were no infected nodes or '*No Infection*', as expected no ratio values were greater than the numerical value of one. Hence no malicious activities or no anomalies were noted.

For the second scenario with '*Rudimentary Infection*', six selected computing nodes from the collective set of computing nodes were infected. These generated significantly more proxy log entries from their associated Malware agents. When the volume based ratio analysis was applied, the infected computing nodes notably had more egress traffic volume with their ratio values significantly greater than one.

For the final scenario with '*Advanced Infection*', the ratio values were all well below suspicious level (that is with ratio values of less than one) and reflecting that the egress traffic was low for all computing nodes. Hence the computing nodes infected with advanced Malware

did not show any indication of possible infection. It had the same noted condition as that of the first scenario's analysis results.

The following charts below showed that the ratio values for all computing nodes along with six infected nodes for the three scenarios.

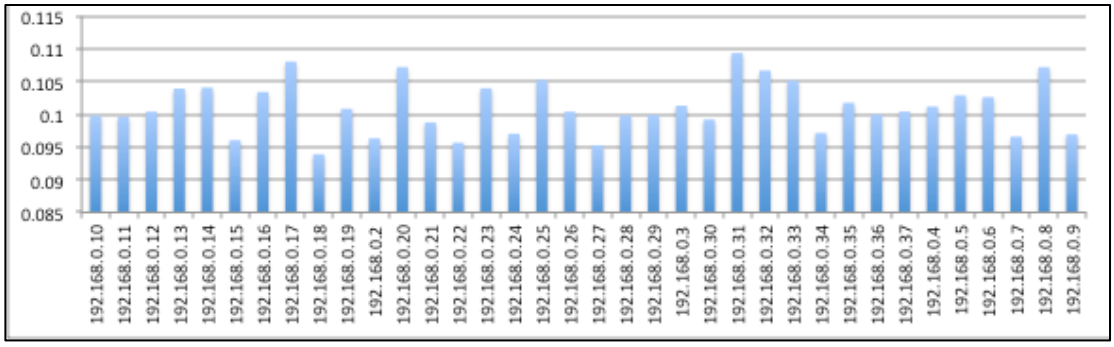

Fig. 4. Volume Based Ratio Analysis for '*No Infection*' Scenario

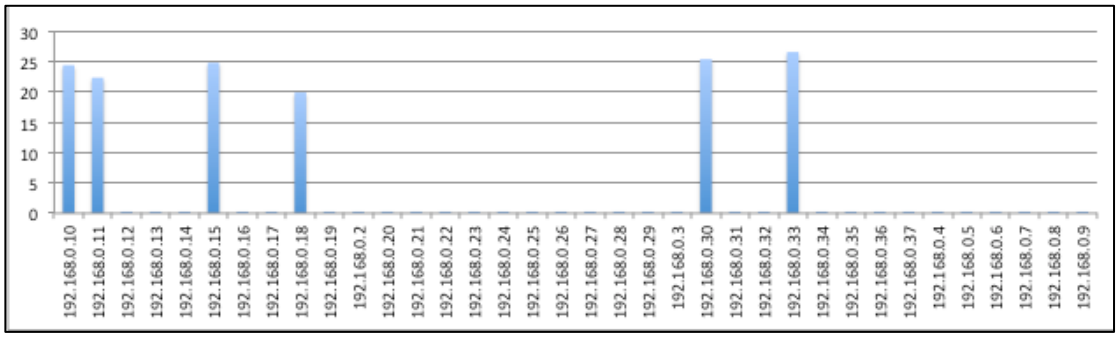

Fig. 5. Volume Based Ratio Analysis for '*Rudimentary Infection*' Scenario

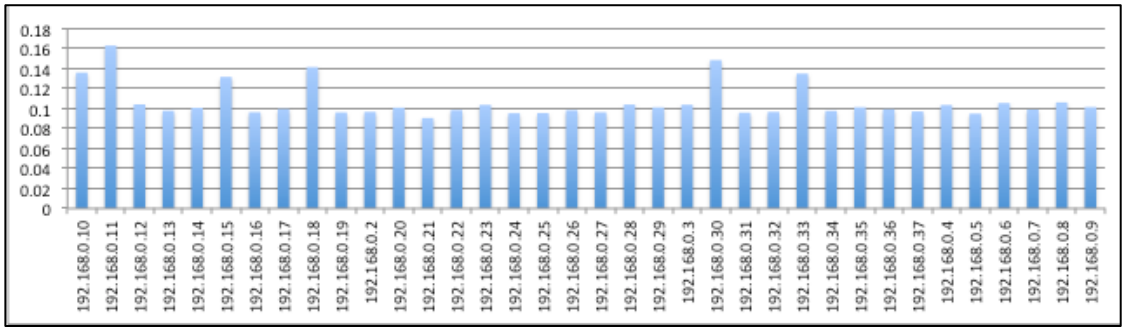

Fig. 6. Volume Based Ratio Analysis for '*Advanced Infection*' Scenario

The chart showed that the infected computing nodes in the '*Rudimentary Infection*' scenario generated a high ratio value variance as compared to the infected computing nodes with '*Advanced Infection*' scenario.

6.2 Evaluation – Volume Based Cumulative Control

The following boxplot charts show the Volume Based Cumulative Control (CUSUM) analysis for each of the three scenarios.

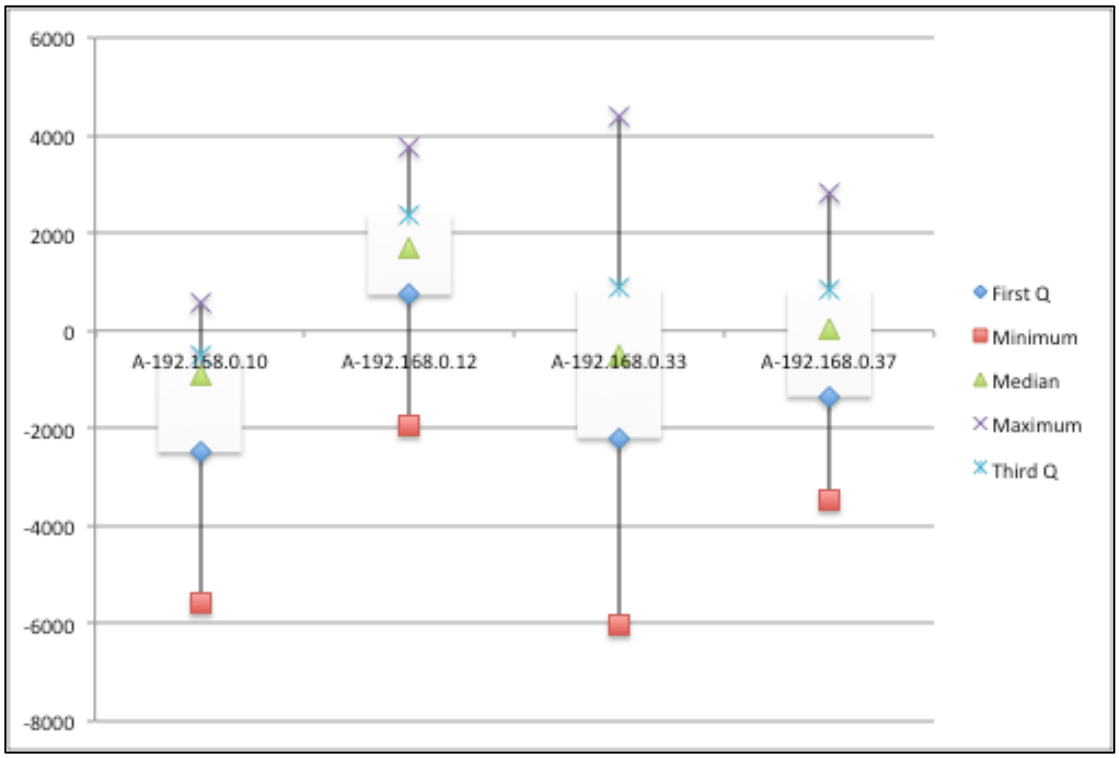

Fig. 7. CUSUM Analysis for '*No Infection*' Scenario

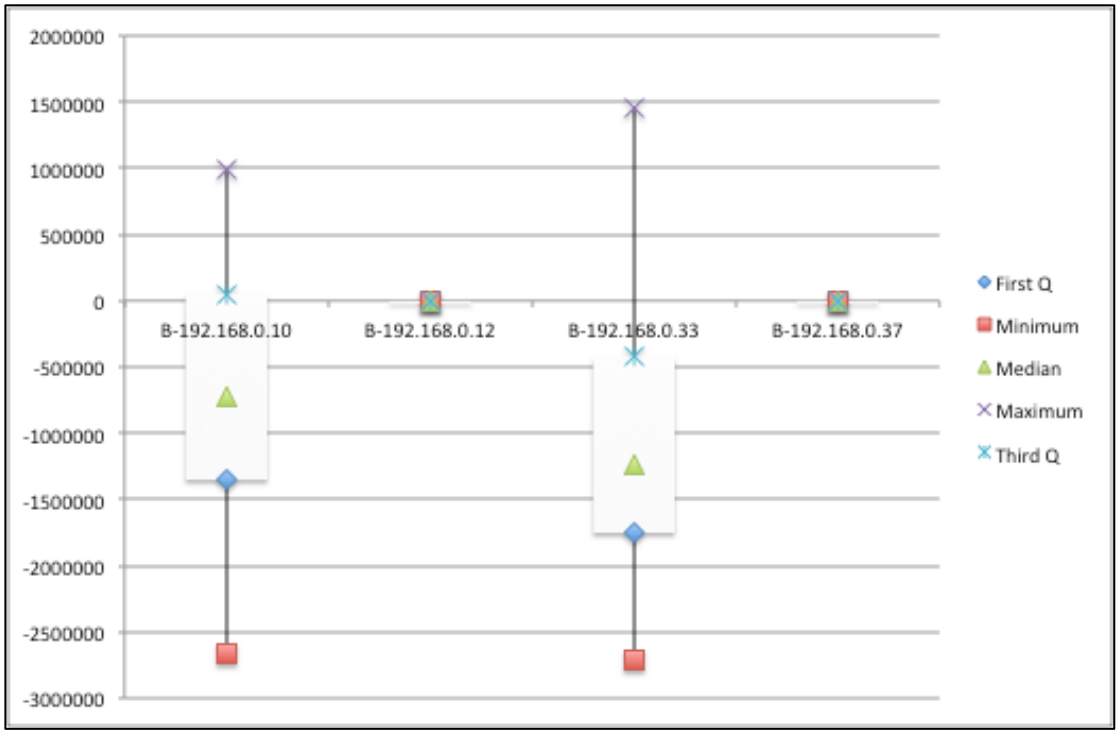

Fig. 8. CUSUM Analysis for '*Normal Infection*' Scenario

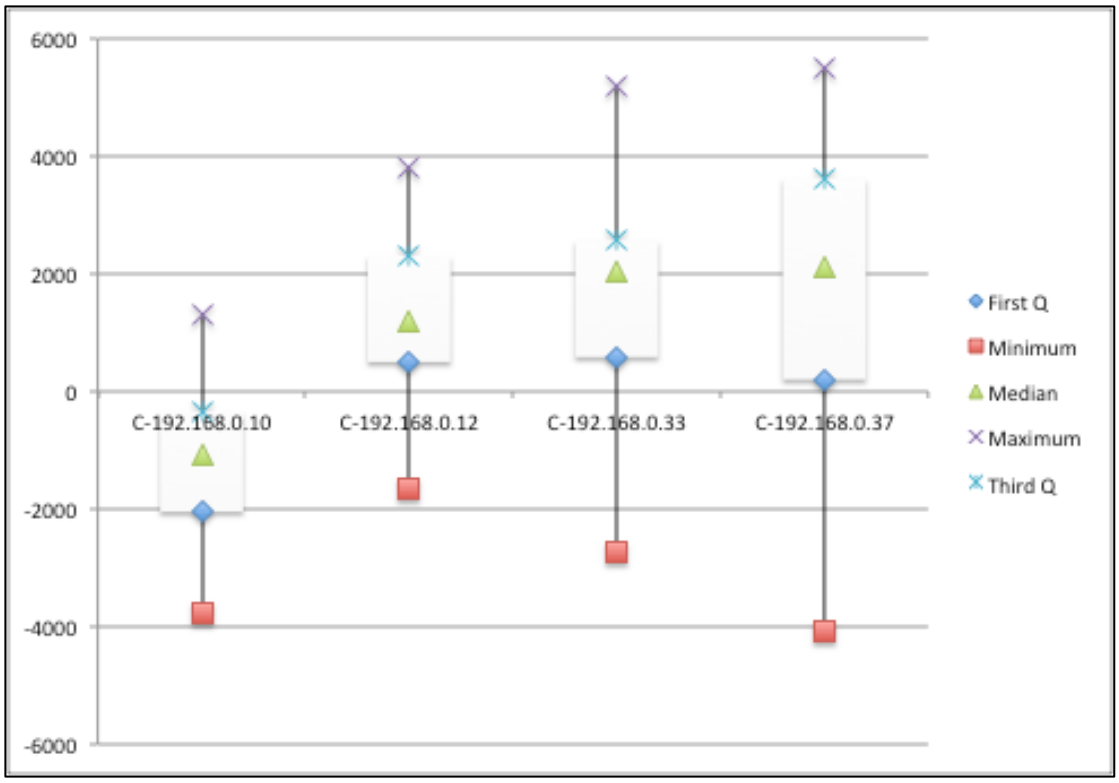

Fig. 9. CUSUM Analysis for '*Advanced Infection*' Scenario

The same four nodes (192.168.0.10, 192.168.0.33, 192.168.0.12 and 192.168.0.37) were chosen for this analysis of the three scenarios. The first two IP addresses belonged to infected computing nodes. The boxplot charts showed that the spread of cumulative sum control had a generally consistent behaviour noted for all computing nodes for the first scenario. For the second scenario, it was noted that the infected nodes had significantly cumulative spread characteristics specifically with the infected nodes as compared to the uninfected nodes. The advanced Malware infection demonstrated the simulation model's ability to obfuscate its egress traffic pattern from this form of analysis liken to the first scenario.

The infected computing nodes in the '*Rudimentary Infection*' generated 'loud' network behaviour pattern that is noticeable using CUSUM analysis however the Malware from '*Advanced Infection*' exhibited relatively silent behaviour.

6.3 Evaluation – Feature Based Signature

The following table shows the Feature Based Signature analysis over the three scenarios.

TABLE 4
Results of Signature Based Analysis

Scenario	Detection Status	Detection Details
No Infection	No	Nil
Rudimentary Infection	Yes	Malicious site : Attack.Malware.com, Non-white listed port : 4444
Advanced Infection	No	Nil

For the first scenario, the model was intentionally configured not to have any Malware included in this run hence the feature signature based analysis had no noted malicious behaviour pattern. For the second scenario, the use of port 4444 and the attempt to access an obvious

blacklisted URL was noted hence could be stopped by a variety of security defence solutions. For the third scenario, the advanced Malware used commonly used protocol (specifically HTTP protocol) to exfiltrate data. Additionally, it used shortly URLs to circumvent blacklisting detection. Hence no malicious network traffic behaviour was noted.

6.4 Evaluation – Feature Based Entropy

The following table shows the Feature Based Entropy analysis over the three scenarios.

TABLE 5
Results of Entropy Based Analysis

Scenario	Detection Status	Detection Details
No Infection	No	http:\\updates.operatingsystem.com 22.2895326 http:\\www.safesite.com 122.282491
Rudimentary Infection	No	http:\\www.safesite.com 131.489098 http:\\updates.operatingsystem.com 21.9456428 http:\\attack.malware.com:4444 13.0138381
Advanced Infection	Yes	http:\\short.ly\M434p3 0.0000 http:\\updates.operatingsystem.com 21.7493 http:\\www.safesite.com 127.6545

For the first scenario, the feature entropy analysis showed that the egress traffic to the variety of URLs reflected a good measure of randomness hence no suspicious activities was noted from this form of analysis. For the second scenario, the feature entropy analysis showed that the egress traffic to the variety of URLs reflected a good measure of randomness. In the third scenario with the feature entropy analysis of egress traffic to each and every destination URLs, it was noted that the egress traffic to the shortly URL was not random and raising the possibility that this could be malicious.

6.5 Evaluation – Summary

The following table showed the overall detection effectiveness for the three scenarios.

TABLE 6
Summary of Evaluation Results

Scenario	Volume Based Ratio	Volume Based CUSUM	Feature Based Signature	Feature Based Entropy
No Infection	Negative	Negative	Negative	Negative
Rudimentary Infection	Positive	Positive	Positive	Negative
Advanced Infection	Negative	Negative	Negative	Positive

There was no notable suspicion of any malicious activities for the first scenario of '*No Infection*' based on the four anomaly detection techniques used. The second scenario ('*Rudimentary Infection*') had positive detection noted from three out of the four analytics used. The third scenario ('*Advanced Infection*') had only one of the four analytics detecting possible suspicious proxy log activities. From these three scenario tests, the model demonstrated the ability to produce or data farmed proxy logs that varied in characteristics that could be used to facilitate the development and validation of cyber security analytics development.

7 Future Research

The model can first be improved by simulating more network environment behaviours. This may include simulating a more complex traffic patterns like network streaming and online gaming behaviours. As the model was intended for cyber security analytics, the model could be further enhanced to support more forms of malicious network behaviour pattern. The model could incorporate evolutionary advancement of Malware characteristics or forms of cyber attacks especially unknown ones as part of data farming liken to data farming used to generate scenarios for war planning. Other forms of model verification and validation could be applied to the proposed model. They include performing subject matter expert evaluation and the use of cyber security detection tools to ingest the generated log data to verify and validate the generated data. Also pre-trained Machine Learning models with real world data may be applied to these farmed data to further validate its suitability.

The farmed data format can be further enhanced to generate other forms of machine generated log data aside from proxy log files. This may include firewall alert logs, intrusion detection / prevention system logs and network sniffing PCAP files.

8 Conclusion

Cyber security analytics provides the potential for defenders to level up in its abilities to detect and prevent malicious intrusions. However there is a need for a suitable development environment to produce datasets containing variety of malicious activities to facilitate the development and evaluation of new cyber security analytics algorithms or models while addressing privacy concerns. The environment should support 'What-If' conditions to support the constantly varying environment. Hence this paper proposes an Agent Based Model that simulates Malware within a network environment to perform cyber security data farming to facilitate cyber security analytics development. This paper covers the design of the model to produce the network web traffic behaviour logs and verification of these data logs using statistical anomaly detection techniques. The farmed data generated from the model can incorporate known and yet to be discovered evolutionary behaviour of advanced Malware that will enable the development of cyber security analytics and in turn improve our fight against the continued advancing cyber security threats.

References

Asman B.C., Kim M.H., Moschitto R.A., Stauffer J.C., and Samuel H. Huddleston S.H. (2011), "Methodology for Analyzing the Compromise of a Deployed Tactical Network", 2011 IEEE Systems and Information Engineering Design Symposium (SIEDS), Pages 164 – 169, Charlottesville, VA, USA, 29 April.

Brandstein, A. G., and Horne, G. E. (1998). Data Farming: A Meta-technique for Research in the 21st Century, Maneuver Warfare Science 1998. Quantico, VA: Marine Corps Combat Development Command.

Chong R.H. (2013), "Trojan.APT.BaneChant: In-Memory Trojan That Observes for Multiple Mouse Clicks", Fireeye, Apr. 1. [Online]. Available: http://www.fireeye.com/blog/technical/Malware-research/2013/04/trojan-apt-banechant-in-memory-trojan-that-observes-for-multiple-mouse-clicks.html. [Accessed Apr. 26, 2014].

Denning D.E. (1987), An Intrusion-Detection Model, IEEE Transactions on Software Engineering, v.13 n.2, p.222-232, February 1987

Embleton S., Sparks S., and Zou C. (2008), "SMM Rootkits: A New Breed of OS Independent Malware", SecureComm '08 Proceedings of the 4th international conference on Security and privacy in communication networks, Article No. 11, New York, USA.

Ferguson P. (2012), "Observations on Emerging Threats", USENIX LEET 2012. [Online]. Available: https://www.usenix.org/conference/leet12/workshop-program/presentation/ferguson. [Accessed Apr. 26, 2014].

Friman H. and Horne G.E. (2005), "Using agent models and data farming to explore network centric operations", Simulation Conference, 2005 Proceedings of the Winter, Orlando, FL, USA.

Haas P.J., Maglio P.P., Selinger P.G., and Tan W.C. (2011), "Data is Dead... Without What-If Models", Proceedings of the VLDB Endowment, Vol. 4, No. 12, Seattle, Washington, August 29th - September 3rd.

Horne G.E., and T.E. Meyer. (2010), "Data Farming and Defence Applications", Proceedings of the MODSIM World 2010 Conference, Hampton Roads, Virginia, USA.

Horne, G.E., and T.E. Meyer. (2004), "Data Farming: Discovering Surprise", Proceedings of the 2004 Winter Simulation Conference, ed. R. G. Ingalls, M. D. Ros- setti, J. S. Smith, and B. A. Peters, 807-813. Piscata- way, New Jersey: Institute of Electrical and Electron- ics Engineers, Inc.

Kim H.J., Na J.C., and Jang J.S. (2006), "Anomaly Detection of Excessive Network Traffic Based on Ratio and Volume Analysis", Intelligence and Security Informatics, Lecture Notes in Computer Science Volume 3975, 2006, Pages 726-727.

Kotenko I., Konovalov A., and Shorov A. (2010), "Agent-Based Modelling and Simulation of Botnets and Botnet Defence", in Proc. Conference on Cyber Conflict 2010, C. Czosseck, K. Podins (Eds.), CCD COE Publications, Tallinn, Estonia.

Kruegel C., and Vigna G. (2003), "Anomaly detection of web-based attacks", Proceedings of the 10th ACM conference on Computer and communications security, Pages 251 – 261.

Lu W., and Tong H. (2009), "Detecting Network Anomalies Using CUSUM and EM Clustering", Advances in Computation and Intelligence, 4th International Symposium, ISICA 2009, Lecture Notes in Computer Science Volume, 5821, Pages 297–308, China.

Mayo J.R., Minnich R.G., Rudish D.W., and Armstrong R.C. (2014), "Approaches for Scalable Modelling and Emulation of Cyber Systems: LDRD Final Report", SANDIA REPORT, SAND2009-6068, Sandia National Laboratories, Sep. 2009. [Online]. Available: http://prod.sandia.gov/techlib/access-control.cgi/2009/096068.pdf. [Accessed Apr. 12, 2014].

Meza J., Campbell S., and Bailey D. (2009), "Mathematical and Statistical Opportunities in Cyber Security", arVix.org, Apr. 9. [Online]. Available: http://arxiv.org/abs/0904.1616. [Accessed Apr. 26, 2014].

NIST (2014), "NIST Roadmap for Improving Critical Infrastructure Cybersecurity", NIST, Feb. 12, 2014. [Online]. Available: http://www.nist.gov/cyberframework/upload/roadmap-021214.pdf. [Accessed Apr. 12, 2014].

Nychis G., Sekar V., Andersen D.G., Kim H., and Zhang H. (2008), "An empirical evaluation of entropy-based traffic anomaly detection", Proceedings of the 8th ACM SIGCOMM conference on Internet measurement, Pages 151-156, ACM, New York, USA.

Page E.S. (1954), "Continuous Inspection Scheme", Biometrika 41 (1/2), Pages 100-115, JSTOR 2333009, Jun. 1954.

Pan J. and Fung C. C. (2010), Boutique Malware – Custom made for e-business. In: The 9th International Conference on e-Business iNCEB, Bangkok, Thailand.

Pan J. and Fung C.C. (2012), "An agent-based model to simulate coordinated response to Malware outbreak within an organisation", Journal International Journal of Information and Computer Security, Volume 5 Issue 2, Pages 115-131, Jan 2012.

Tsai F.S., and Chan K.L. (2007), "Detecting Cyber Security Threats in Weblogs Using Probabilistic Models", Intelligence and Security Informatics, Lecture Notes in Computer Science Volume 4430, Pages 46-57.

Virvilis N., Gritzalis D., and Apostolopoulos T. (2013), "Trusted Computing vs. Advanced Persistent Threats: Can a defender win this game?", 2013 IEEE 10th International Conference on Ubiquitous Intelligence and Computing and 10th International Conference on Autonomic and Trusted Computing (UIC/ATC), Pages 396-403, Vietri sul Mere, 18-21 Dec.

Wang H., Zhang D., and Shin K. G. (2002), "Detecting SYN flooding attacks," in Proceedings of the 21st Annual Joint Conference of the IEEE Computer and Communications Societies (INFO- COM '02), vol. 3, pp. 1530–1539, New York, NY, USA, June 2002.

Wagner A., and Plattner B. (2005), "Entropy based worm and anomaly detection in fast IP networks", 14th IEEE International Workshops on Enabling Technologies: Infrastructure for Collaborative Enterprise, Pages 172-177.

Xie P., Li J. H., Ou X., Liu P., and Levy R. (2010), "Using Bayesian networks for cyber security analysis", 2010 IEEE/IFIP International Conference on Dependable Systems and Networks (DSN), Pages 211-220, Chicago, IL, USA, June 28 -1 July.

Zhang H., Banick W., Yao D., and Ramakrishnan N. (2012), "User Intention-Based Traffic Dependence Analysis for Anomaly Detection", 2012 IEEE Symposium on Security and Privacy Workshops (SPW), Pages 104 – 112, San Francisco, CA, USA, 24-25 May.

Wilensky U. (2009), "NetLogo", Center for Connected Learning and Computer-Based Modelling, Northwestern University, Evanston, IL, 2009.