Anomaly detection in cyber security attacks on networks using MLP deep learning

T.T. Teoh
University of Technology and Design, Singapore
ateiktoe_teoh@sutd.edu.sg

Graeme Chiew
University of Technology and Design, Singapore
bgraeme_chiew @mymail.sutd.edu.sg

Edwin J Franco
University of Technology and Design, Singapore
cedwin_josephlal@mymail.sutd.edu.sg

P.C. Ng
University of Technology and Design, Singapore
d pockchee_ng@mymail.sutd.edu.sg

Benjamin M.P
University of Technology and Design, Singapore
epriscilla_benjamin@mymail.sutd.edu.sg

[d,e] Y.J. Goh
University of Technology and Design, Singapore
fyujin_goh@mymail.sutd.edu.sg

***Abstract*— Malicious traffic has garnered more attention in recent years, owing to the rapid growth of information technology in today's world. In 2007 alone, an estimated loss of 13 billion dollars was made from malware attacks . Malware data in today's context is massive. To understand such information using primitive methods would be a tedious task. In this publication we demonstrate some of the most advanced deep learning techniques available, multilayer perceptron (MLP) and J48 (also known as C4.5 or ID3) on our selected dataset, Advanced Security Network Metrics & Non-Payload-Based Obfuscations (ASNM-NPBO) to show that the answer to managing cyber security threats lie in the fore-mentioned methodologies.**

Keywords- Multilayer Perceptron (MLP), cyber security, information security, machine learning, deep learning, big data, decision trees, WEKA, J48, C4.5, ID3

I. INTRODUCTION

The age of information has been existential for almost three decades. The world has grown vastly alongside the inevitable advancement of technology. In recent years, the attention has been shifted to what we refer to as the fourth industrial revolution – ubiquitous mobile supercomputing, internet of things, networks. intelligent robots and self-driving cars. One important question we should ask ourselves: are we able to secure these assets faster than we are able to produce them?

Indeed, in today's context, securing the assets of businesses is a growing concern and a very real problem. The amount of information we need to protect, as well as the information exchanged is massive. A data log containing cyber-attack information can sit in the order of giga or even terra bytes.

It is inefficient and perhaps even impossible to manually study such voluminous amounts of data and fortunately, the advancement of technology in today's world gives us access to more powerful algorithms as well as hardware to fuel our understanding of cyber-attacks. In recent years, the development of various deep learning algorithms such as multilayer perceptron (MLP), Long-short-term memory (LSTM) and Hidden Markov Models (HMM) has demonstrated the learning power of machines.

In this paper, we are interested in applying MLP on the study of our selected dataset, ASNM-NPBO to demonstrate that the future of malware detection lies in deep learning.

II. BACKGROUND

In this section we will describe our model used to study this voluminous data. We begin first by reviewing some lore relevant to our investigation.

A cyber attack refers to a malicious attack that may damage a computer or computer system, that may arise from having unauthorized network access, data or code injection [16]. Cybersecurity refers to the protection against that said attack through controlling physical hardware or software. Malicious software, or *malware* refers to all kinds of intrusive software, capable of performing malicious attacks as defined [17].

Common types of malware include *viruses*, *trojan horses*, *backdoors* and *rootkits*. A virus is a seemingly harmless program that is capable of creating copies of itself, while injecting those copies into other programs or files to cause damage [18]. A trojan horse is a software disguised as benign code that awaits a victim to install it. The trojan contains some hidden function which is often destructive to the computer or computer system when the application runs [19]. A backdoor is a means to bypass some authentication procedure or system, often to gain unauthorized access to a network connection [20]. When a malicious code has found its way into a computer system, it is important for it to remain concealed to avoid detection. A rootkit is responsible for that by modifying a computer's firmware. This modification would cause that malicious code to appear as benign [21]. If this malicious code is to be detected, we would need some form of anomaly detection that does not reside within the computer system. To this end, diagnostic logs can be studied with machine learning to detect such anomalies even if the firmware of a system has been modified.

Machine learning refers to a field in computer science that gives computer systems the ability to progressively improve performance on a specific task with some input dataset, as elegantly defined by Arthur Samuel in [3]. The nature of such data can be trivial to some extent as the prowess and scope of machine learning has advanced tremendously in the last decade. Machine learning algorithms can be generally categorized as supervised or unsupervised and classifying or generative. *Perceptron* is a simple and fundamental implementation of a linear classifier. Perceptron is mistake-

driven and can alters parameters per mistakes. This condition can be expressed as [6]

$$if\ y^{(t)} != h\ (x^{(t)};\ \theta^{(k)})$$

$$then\ \theta^{(k+1)} = \theta^{(k)} + y^{(t)}\ x^{(t)} \quad (1)$$

where $y^{(t)}(x)$ is some binary classifier. We also have some training dataset $D = (d_1 \ldots \ldots d_j)$, and some initialized weights $\boldsymbol{\theta}(t) = \theta_0 + \theta_1 + \ldots + \theta_j$. We would then calculate

$$y_i(t) = (h(\boldsymbol{\theta}(t).\ \mathbf{x}) \quad (2)$$

and update the weights we

$$\theta_i\ (t+1) = \theta_i\ (t) + (d_j - y_j(t))\ x_{i,j} \quad (3)$$

This algorithm is repeated for

$$s^{-1} \sum_j^s |d_j - y_i(t)| < \Upsilon \quad (4)$$

where Υ is the user specified error threshold. However, as this primitive form of machine learning is a form linear classification, the nature of problems it can be applied to in today's context is indeed limited.

In deep learning, neural networks form a powerful cluster of algorithms that can solve a variety of problems such as facial and voice recognition [2], natural language processing (NLP), bioinformatics, healthcare and anomaly detection. The fundamental idea stems from the simplest of feedforward networks, where we have some data (inputs), $x_1 \ldots \ldots x_i$, apply it to some mathematical operation $H(x)$, and eventually represents some output, which is usually some classify label in binary. The above can be loosely represented by Fig. 1.

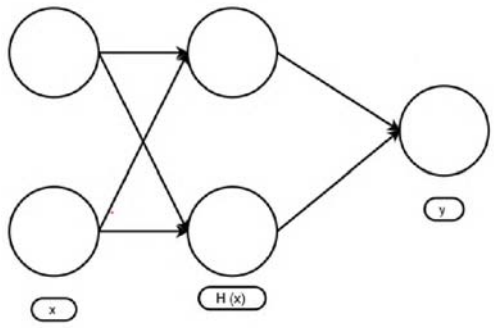

Fig. 1 – Simple neural network with one hidden layer

For a recurrent network, the hidden layer considers previously perceived inputs [4]. Mathematically, this can be written as

$$H_t = \phi\ (w\ x_t + U\ x_{t-1}) \quad (5)$$

where H_t some hidden state at a time t, w is a real value weight which can be modified, and U is some hidden-state-to-hidden-state matrix. ϕ can be represented by a *sigmoid* function or a *tanh* function.

For multilayer perceptron (MLP), it is a class of feedforward neural networks that consists of at least 3 layers of nodes [4]. Each node uses a nonlinear *activation* function [5] for the classification algorithm. The activation function can be written as

$$y(v_i) = \tanh (v_i) \quad (6)$$

$$y(v_i) = (1 + e^{-vi})^{-1} \quad (7)$$

For training, it uses a *supervised learning* technique known as backpropagation [8]. Assuming some target value we wish to achieve, d_j, and the output produced by the more primitive perceptron algorithm, y_j, we can assume the difference to be the error of one neuron,

$$e_j\ (n) = d_j\ (n) - y_j\ (n) \quad (8)$$

Our error function is then

$$\varepsilon\ (n) = \tfrac{1}{2} \sum e_j^2\ (n) \quad (9)$$

Applying gradient descent, the change in weights each iteration can be written as

$$\Delta\ w_{ij}\ (n) = -\ \eta\ (d\ \varepsilon\ /\ dv_j)\ \ y_i\ (n) \quad (10)$$

III. Model and Methodology

In this paper, we attempt to investigate our dataset, ASNM-NPBO. In the dataset, we have *~9000* rows of data and *~900* columns of attributes. The dataset has *2* ways of labelling the traffic as malicious or benign. Under *label_2*, the values are either *TRUE* or *FALSE*, indicating if traffic is malicious or benign respectively. Under *label_3*, we have values *1, 2* or *3*. A benign traffic is labeled as *3*. Malicious attacks can be

1 – Direct attack

2 – Obfuscated attack

The purpose of our investigation is to show that for such large datasets, the power of deep learning will greatly aid us in understanding and predicting malicious attacks and that this endeavor is scalable. For our purposes, we will simply be concerned if the traffic is malicious or not. Our methodology will systematically interpret and analyze voluminous datasets with rigor as well as haste. This model that we are using can be achieved by applying the following:

A. With some intuitive sense, select a reasonable number of attributes. For our dataset, we selected *15* attributes that appear to behave well with our binary classifier, *label_2*.

B. Using *Microsoft Excel's Data Analysis*, we select our cell range and apply them into the in-built *covariance* function. Dividing this covariance by respective standard deviations, we obtain the *correlation*, which some may prefer as we can consider this statistically *normalized*.

C. Repeat (ii) if we obtain a poor correlation. If the correlation is reasonably good, we use the Waikato Environment for Knowledge Analysis (WEKA) to further visualize the selected attributes. This visualization should give us yet another perspective of our attributes. For our dataset, we have selected *MeanPktLenOut* and *SessDuration*.

D. With our selected attributes, we proceed to run a *J48* decision tree, giving us a remarkable prediction rate if *~0.993*.

E. We then implement the MLP algorithm using *python*.

F. Finally, we can compare our implemented algorithm with WEKA's J48.

The above steps can be summarized in the following figure.

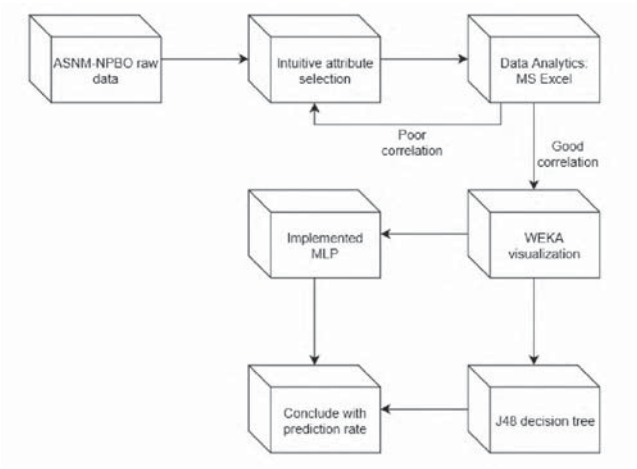

Fig. 2 – Our systematic model used in investigating the ASNM-NPBO. With the voluminous raw dataset, we select 15 attributes which are confirmed with MS Excel correlation; attributes are further thinned down using WEKA visualization. Finally, we use J48 decision tree and an implemented MLP to evaluate our prediction rate.

IV. Experiment and Results

We now discuss our approach with more rigor and experimental data. Covariance between X and Y can be written mathematically as [10]

$$COV(X, Y) = \sum |(X_i - \bar{X})(Y_i - \bar{Y}) / n| \qquad (11)$$

The correlation between X and Y can be written as [12]

$$COR(X, Y) = \frac{\sum [(Xi - X)(Yi - \bar{Y})}{\sigma x \, \sigma y} \qquad (12)$$

Using *Microsoft (MS) Excel Data Analysis*, we can obtain the covariance and correlation as shown in the following.

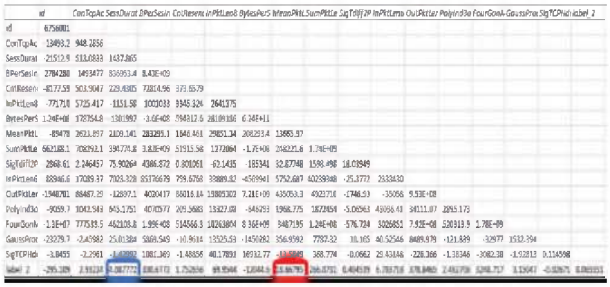

Fig. 3 – Covariance table generated from MS Excel, with the top two covariances highlighted in red and blue respectively.

Similarly, we obtain the correlation table with MS Excel as shown below.

Fig. 4 – Table of correlations with the labeling classifier. Attributes with the highest 2 correlations highlighted in red and blue. The two attributes are *MeanPktLenOut* and *SessDuration* in that order.

Conveniently, we are also able to generate the *Regression*s of our attributes to further affirm our selection as shown in the following.

SUMMARY OUTPUT

Regression Statistics	
Multiple F	0.623718
R Square	0.389024
Adjusted	0.388956
Standard	0.200147
Observati	9004

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regressio	1	229.6088	229.6088	5731.809	0
Residual	9002	360.5084	0.040059		
Total	9003	590.2171			

	Coefficien	*Standard*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	0.941138	0.002715	346.6809	0	0.935817	0.94646	0.935817	0.94646
X Variable	0.001366	1.8E-05	75.70871	0	0.001331	0.001401	0.001331	0.001401

Fig. 5(a)

SUMMARY OUTPUT

Regression Statistics	
Multiple F	0.421056
R Square	0.177288
Adjusted	0.177196
Standard	0.232252
Observati	9004

ANOVA

	df	*SS*	*MS*	*F*	*Significance F*
Regressio	1	104.6383	104.6383	1939.858	0
Residual	9002	485.5788	0.053941		
Total	9003	590.2171			

	Coefficien	*Standard*	*t Stat*	*P-value*	*Lower 95%*	*Upper 95%*	*Lower 95.0%*	*Upper 95.0%*
Intercept	1.05289	0.00248	424.5279	0	1.048028	1.057752	1.048028	1.057752
X Variable	0.002843	6.45E-05	44.04381	0	0.002716	0.002969	0.002716	0.002969

Fig. 5(b)

Fig. 5(a) and (b) – Regression table for *MeanPktLenOut* (a) and *SessDuration* (b). Our output shows a very small *Significance F*, *~0* for both figures, and a satisfactorily large *Multiple R* for (a).

In Fig. 5, we are particularly concerned with the *Multiple Regression* and the *Significance F* values. For linear regression, it is typically applied to some scattered data points, and that the goal is to fit a best line such that the sum of the squared deviations of all the points from the line are kept to a minimum. In doing this, we can also say that the *error function* is minimized. For the case of multiple regression, for k dimensions, it is simply a dot product of the slope of all the dimensions, given by

$$y = m_1x_1 + m_2x_2 + \ldots \ldots m_kx_k + c \quad (13)$$

For both Fig. 5(a) and Fig. 5(b), we have multiple regression values of *0.624* and *0.421* respectively. We can infer that these values are typically healthy for our attributes in terms of the error function [7]. The Signficance F values are very small, *~0* suggesting a near-perfect correlation of the selected attributes and the classifying labels.

Proceeding next, we reduce our data to the 15 attributes earlier selected. We first visualize all the attributes against the classifying labels, marked in blue (malicious) and red (benign).

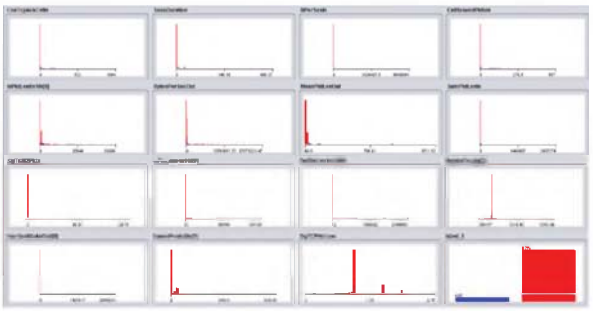

Fig. 6 – An overall visualization of our selected 15 attributes; the data clearly represents that the number of non-attacks far exceed the number of attacks.

From the visualizer, we observe that the number of benign traffic far exceeds the number of malicious traffic. This set of data can be said to be suitable in assessing if the learning method has high *false-positive* predictions, which we were not able to discern from layman tools such as MS Excel. Using WEKA to run the J48 decision tree, we obtain the following figure.

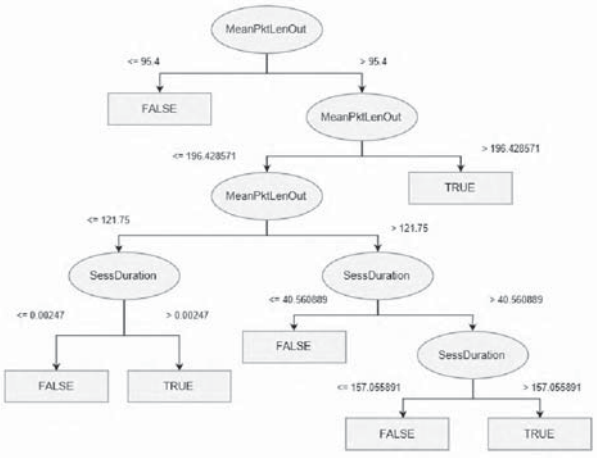

Fig. 7 – J48 decision tree. Using 1000 epochs with a learning rate of 0.2, the decision tree has learnt accordingly as shown for classifying attacks based on *MeanPktLenOut* and *SessDuration*.

The J48 decision's learning outcome is as follows –

i. If *MeanPktLenOut* ≤ 95.4, traffic is benign.

ii. If *MeanPktLenOut* >196.4, traffic is benign.

iii. If *MeanPktLenOut* ≤ 121.75, and if *SessDuration* ≤ 0.00247, traffic is benign.

iv. If *MeanPktLenOut* > 121.75, and if *SessDuration* ≤ 40.56, traffic is benign.

The summary output for the J48 decision tree is as below.

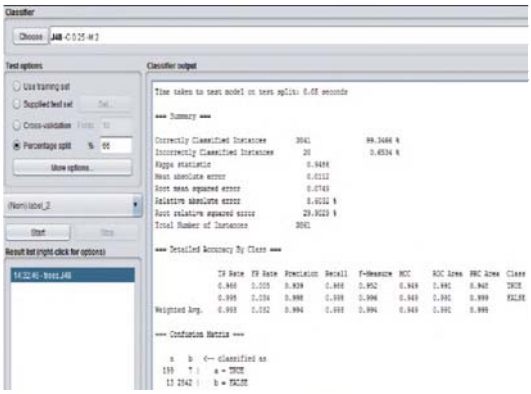

Fig. 8 – J48 decision tree summary with a prediction rate of *~0.9935*.

For a quick demonstration, we have manually entered our data into our MLP implementation instead of reading the *.csv* file into our script. This is also to demonstrate how easily this implementation can be done even with limited resources and time. We took an extract of the original dataset, hard coding that extracted data into a *python* script.

The labels are

```
Labels=np.array([[1],[1],[1],[1],[1],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0],[0]])
```

Running our python script, we get the predicted outputs

```
[[  9.99999957e-01]
 [  9.99999530e-01]
 [  9.99999981e-01]
 [  9.80533264e-01]
 [  9.98270708e-01]
 [  1.97733468e-06]
 [  1.48477704e-06]
 [  1.86613535e-06]
 [  1.98710941e-02]
 [  1.12571379e-04]
 [  1.48494462e-06]
 [  1.48891587e-06]
 [  1.87498796e-06]
 [  1.87498636e-06]
 [  1.41308133e-06]
 [  2.45320507e-08]
 [  2.45330081e-08]
 [  2.45320507e-08]
 [  2.45325826e-08]
 [  2.45320507e-08]]
```

Fig. 9 – MLP output predictions. The first *5* printed outputs are *~1* and the remaining 15 printed outputs correspond to *~0*. This reflects exactly the manually coded labels earlier.

From the output, we observe that the first 5 printed outputs approximate to *1* whilst the remaining 15 printed outputs approximate to *0*, reflecting exactly the binary labels earlier hard-coded.

V. Conclusion & Future work

In today's context, mustering information and being able to interpret what they represent is a crucial aspect in society. The domain of cybersecurity is by no means different. Whilst information is often voluminous and impossible to fully study with human eyes and minds, we now have the boon of deep learning.

Applying multilayer perceptron to a large dataset containing information of malicious and benign attacks, we were able to demonstrate the efficiency and accuracy of our implemented MLP script. With WEKA's J48, we were able to achieve a prediction rate of *0.9935* on attributes that have near-perfect correlation with the classifying labels.

The domain of deep learning has a multitude of powerful algorithms that may be applied to almost any problem in data analytics. In view of the rising demand for responses and solutions to cyber risks, applying other deep learning techniques onto cybersecurity problems holds an extremely large prospect for future work.

While we have explored MLP and J48, there are many other algorithms that can be used for different problem statements.

References

[1] Malware Report: The Economic Impact of Viruses, Spyware, Adware, Botnets, and Other Malicious Code

[2] NIPS Workshop: Deep Learning for Speech Recognition and Related Applications, Whistler, BC, Canada, Dec. 2009

[3] Samuel, Arthur (1959). "Some Studies in Machine Learning Using the Game of Checkers". IBM Journal of Research and Development. 3 (3). doi:10.1147/rd.33.0210

[4] Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961Rosenblatt, Frank. x. Principles of Neurodynamics: Perceptrons and the Theory of Brain Mechanisms. Spartan Books, Washington DC, 1961.

[5] R. Collobert and S. Bengio (2004). Links between Perceptrons, MLPs and SVMs. Proc. Int'l Conf. on Machine Learning (ICML).

[6] Haim Sompolinsky, "MIT notes", October 2013

[7] Rencher, Alvin C.; Christensen, William F. (2012), "Chapter 10, Multivariate regression – Section 10.1, Introduction", Methods of Multivariate Analysis, Wiley Series in Probability and Statistics, 709 (3rd ed.), John Wiley & Sons, p. 19, ISBN 9781118391679

[8] Rumelhart, David E., Geoffrey E. Hinton, and R. J. Williams. "Learning Internal Representations by Error Propagation". David E. Rumelhart, James L. McClelland, and the PDP research group. (editors), Parallel distributed processing: Explorations in the microstructure of cognition, Volume 1: Foundation. MIT Press, 1986.

[9] HOMOLIAK Ivan, BARABAS Maros, CHMELAR Petr, DROZD Michal a HANACEK Petr.: ASNM: Advanced Security Network Metrics for Attack Vector Description. In: Proceedings of the 2013 International Conference on Security & Management. Las Vegas: Computer Science Research, Education, and Applications Press, 2013, s. 350-358. ISBN 1-60132-259-3.

[10] Rice, John (2007). Mathematical Statistics and Data Analysis. Belmont, CA: Brooks/Cole Cengage Learning. p. 138. ISBN 978-0534-39942-9

[11] J. R. Quinlan. Improved use of continuous attributes in c4.5. Journal of Artificial Intelligence Research, 4:77-90, 1996

[12] Rodgers, J. L.; Nicewander, W. A. (1988). "Thirteen ways to look at the correlation coefficient". The American Statistician. 42 (1): 59–66. doi:10.1080/00031305.1988.10475524. JSTOR 2685263

[13] Ian H. Witten; Eibe Frank; Mark A. Hall (2011). "Data Mining: Practical machine learning tools and techniques, 3rd Edition". Morgan Kaufmann, San Francisco. p. 191

[14] Ian H. Witten; Eibe Frank; Len Trigg; Mark Hall; Geoffrey Holmes; Sally Jo Cunningham (1999). "Weka: Practical Machine Learning Tools and Techniques with Java Implementations" (PDF). Proceedings of the ICONIP/ANZIIS/ANNES'99 Workshop on Emerging Knowledge Engineering and Connectionist-Based Information Systems. pp. 192–196. Retrieved 2007-06-26

[15] May 2006 Diarmuid Pigott's Encyclopedia of Computer Languages Archived 2011-02-20 at the Wayback Machine. hosted at Murdoch University, Australia lists 8512 computer languages

[16] Daniel, Schatz,; Rabih, Bashroush,; Julie, Wall, (2017). "Towards a More Representative Definition of Cyber Security". Journal of Digital Forensics, Security and Law. 12 (2). ISSN 1558-7215. Archived from the original on 28 December 2017

[17] "Defining Malware: FAQ". technet.microsoft.com. Retrieved 10 September 2009.

[18] Peter Szor (3 February 2005). The Art of Computer Virus Research and Defense. Pearson Education. p. 204. ISBN 978-0-672-33390-3.

[19] "Non-Windows Malware". Betanews.

[20] Vincentas (11 July 2013). "Malware in SpyWareLoop.com". Spyware Loop. Retrieved 28 July 2013.

[21] McDowell, Mindi. "Understanding Hidden Threats: Rootkits and Botnets". US-CERT. Retrieved 6 February 2013.