# Internal Network Monitoring and Anomaly Detection through Host Clustering

W. J. B. Beukema[1], T. Attema[2] and H. A. Schotanus[2]
[1]*Design and Analysis of Communication Systems (DACS), University of Twente, Enschede, The Netherlands*
[2]*Cyber Security and Robustness, The Netherlands Organisation for Applied Scientific Research (TNO), The Hague, The Netherlands*

Keywords: Internal Network Traffic, Intrusion Detection, Host Clustering, Anomaly Detection, Advanced Persistent Threats.

Abstract: Internal network traffic is an undervalued source of information for detecting targeted attacks. Whereas most systems focus on the external border of the network, we observe that targeted attacks campaigns often involve internal network activity. To this end, we have developed techniques capable of detecting anomalous internal network behaviour. As a second contribution we propose an additional step in the model-based anomaly detection involving host clustering. Through host clustering, individual hosts are grouped together on the basis of their internal network behaviour. We argue that a behavioural model for each cluster, compared to a model for each host or a single model for all hosts, performs better in terms of detecting potentially malicious behaviour. We show that by applying this concept to internal network traffic, the detection performance for identifying malicious flows and hosts increases.



## 1 INTRODUCTION

In order to detect activity related to targeted attacks most network intrusion detection systems monitor the borders of network infrastructures, in an effort to make it harder for attackers to penetrate into the network or exfiltrate data from the network. By focusing on this approach, the internal network is largely left unmonitored (Shiravi et al., 2012). Both (Hutchins et al., 2011) and (Dell SecureWorks, 2013) indicate, however, that a significant part of the life cycle of a targeted attacks is execute *within* the targeted network. Targeted attacks that are in these stages cannot be detected by measures taken on the border of the network, but may be detected by monitoring internal network traffic. The attack phase *Expand access and obtain credentials* of the SecureWorks kill chain (Dell SecureWorks, 2013) might, for example, result in an unusual high volume of traffic from an infected host to authentication servers or other hosts.

Intrusion detection algorithms can generally be classified into two different categories; misuse-based approaches and anomaly-based approaches (Sabahi and Movaghar, 2008). Misuse-based approaches try to extract signatures of known threats such that they can be detected, whereas anomaly-based approaches try to detect substantial deviations from behaviour that is considered to be normal. Misuse-based approaches are very effective in finding known threats, missing however new threats that behave differently. Anomaly-based approaches are aimed to detect also new deviations from normal and benign network behaviour. The major challenge in applying anomaly detection to internal network intrusion detection lies in balancing the false positive and true positive rate.

In some traditional network-based anomaly detection approaches, a model is created for each individual host in a network. Such a model is a representation of normal behaviour for that host. Based on these models, the intrusion detection system analyses all network traffic for each host and tries to detect deviations. However, this approach suffers from several drawbacks. For instance, individual host models often result in high false positive rates, caused by the fact that minor changes in the host's behaviour are considered to be anomalous. Additionally, this method does not scale well to large networks, making this approach impractical for deployment in larger networks.

Another approach is creating a single model for a network as a whole. Although this solves scalability issues, the pitfall of this approach is that by taking too many hosts together, important details get lost. Con-



Beukema, W., Attema, T. and Schotanus, H.
Internal Network Monitoring and Anomaly Detection through Host Clustering.
DOI: 10.5220/0006288606940703
In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy (ICISSP 2017)*, pages 694-703
ISBN: 978-989-758-209-7

sider, for example, a workstation that starts behaving like a server. This behaviour is clearly anomalous, but if both workstations and servers are included in the same model the behaviour of this host is unlikely to be detected.

In other words, the problem with existing approaches is that despite currently being the best option to identify undiscovered threats, anomaly-based intrusion detection suffers from some serious limitations which may make its use impractical. The question therefore is how anomaly-based intrusion detection can be improved such that it is feasible to be used it in practice.

The research presented in this paper is conducted in the partial fulfilment of the requirements for the degree of Master of Science at the University of Twente. A more comprehensive description of the methods and results can be found in (Beukema, 2016). This research is aimed at developing techniques that contribute to the detection targeted attacks.

To this end, we propose a novel anomaly detection algorithm that is based on internal network traffic, i.e. network traffic between hosts on the internal network that does not cross the border with the (external) Internet. We argue that internal network traffic is a valuable source of information in detecting network intrusions and we have developed new techniques to make use of this source.

Secondly, we propose the use of clustering techniques to enhance our anomaly detection approach. These techniques allow us to group the hosts in a network that display similar behaviour together. As we will show, separately modelling the different clusters results in models that are on the one hand robust to minor behavioural changes of hosts and on the other hand capturing only a specific type of behaviour. Hence, for each cluster, a model is created, describing how entities within the cluster are expected to behave. We will specifically focus on the internal network behaviour that is displayed by hosts.

The novelty of this study is twofold. First of all, the presented approach models internal network behaviour and detects deviations from this behaviour. Secondly, host clustering is used to improve the intrusion detection algorithms that are used.

More generally, the proposed clustering techniques may be applied in other domains as well. Even though this research focuses on internal network traffic, the proposed method may be applied to solutions such as user behaviour analysis and host behaviour analysis. In this sense, a more general contribution of this research is providing a new way of enhancing anomaly detection when dealing with vast amounts of data.

## 1.1 Related Work

Research in the field of targeted attack detection is progressing rapidly. More and more intrusion detection systems (IDSs) are becoming available, aiming to mitigate cyber-security risks. In (Sabahi and Movaghar, 2008) a taxonomy of different characteristics of existing IDSs is given. This taxonomy indicates that IDSs can be differentiated based on the data source they use. Host-based IDSs, such as virus scanners and other end-point protection mechanisms, are installed on individual hosts and monitor in much detail the behaviour of the host (Scarfone and Mell, 2007). In contrast, network-based IDSs monitor the network traffic generated by hosts.

In (Ehrlich et al., 2010) network traffic is analysed in order to detect spam bots and their controllers. This article present a Bayesian classification algorithm to distinguish between legitimate mail servers and spammers. In (Xiao et al., 2015) a method for detecting temporal inconsistencies in network traffic is presented. Many aspects of network behaviour are consistent over time and inconsistencies can be a sign of malicious behaviour. A more exhaustive overview of network traffic detection techniques is given in (Roy and Chaki, 2014).

As was mentioned before, one can also differentiate between misuse- and anomaly-based IDSs. A survey of commonly used techniques is given by (Lazarevic et al., 2005). Despite the progress made in the field of anomaly detection, available IDSs still tend to focus on misuse-based techniques that require knowledge about cyberattacks and system vulnerabilities (Research and Markets, 2015).

Clustering techniques have been studied even more extensively. Overviews of various network- or graph-based clustering techniques are given in (Xu et al., 2005) and (Akoglu et al., 2015). More recent the concept of host clustering has also been studied (Xu et al., 2011; Wei et al., 2006). These studies focus on clustering hosts that are active on the Internet in order to detect, for example, botnets. The idea of making use of clustering techniques to improve anomaly-based approaches has been applied before. (Hutchins et al., 2011) apply clustering techniques to the results of their anomaly detection approach in order to detect false positives. To the knowledge of the authors, the use of clustering techniques to separately model hosts with similar behaviour has not been proposed before.

# 2 INTERNAL NETWORK TRAFFIC

With existing, common IDS technologies, networks are often protected according to the eggshell principle (Shiravi et al., 2012): strong on the outside, soft on the inside. In other words, the border between the internal network and the (external) Internet is well-monitored, while the activity of hosts on the internal network is hardly taken into account. This implies that once an attacker has gained foothold in the target environment, it can relatively easily move around the network in order to expand its access, making it more likely the attacker will be able achieve its goal. In this section we argue that internal network traffic, i.e. traffic inside the internal network, is another valuable source in detecting targeted attacks. Depending on the amount and geographical dispersion of hosts that are monitored, the internal network be anything from a Local Area Network to a Wide Area Network. The attack surface of common day networks is ever increasing and the first line of defence may not be able to mitigate all attacks. Making use of this additional source of information provides monitoring beyond the first line of defence to provide a more complete overview of possible attackers activity.

## 2.1 Attack Campaigns

The life cycle of targeted attacks, such as presented in (Dell SecureWorks, 2013) model the typical behaviour of targeted attacks. These life cycles describes the different phases of attack campaigns, starting with the phase *Define Target* and ending with the phase *Cover Tracks*.

As can be inferred from this life cycle, a number of the attack phases involve Internet traffic. However, there are also some attack phases that take place between the borders of the targeted network. The phases *Expand access and obtain credentials*, *Strengthen foothold* and in some cases the phase *Cover tracks* hardly result in any external network traffic and therefore go undetected by most existing IDSs.

As intrusion detection ought not to depend solely on intercepting attacks in the phases that generate external network traffic (Byrne, 2013), it is vital to develop methods to detect intrusions in the other stages as well. As the aforementioned phases have a strong component of internal network traffic, it is worthwhile to investigate this area further. In the next paragraphs, some bigger, well-described attack-campaigns are evaluated for their internal network aspect.

**Stuxnet** (Langner, 2011) was an advanced attack employing worms, targeted at an Iranian nuclear facility. This attack successfully changed the rotor speed of the facility's centrifuges causing them to be destroyed. Stuxnet made use of several zero-day vulnerabilities to automatically spread itself within the network in order to find and infect the systems controlling the centrifuges. Moreover, a P2P network structure was set up between the infected hosts part of the same (internal) network. This P2P network enabled the infected hosts to distribute updates among each other. Since the targeted network was not connected to the Internet this attack could not have been detected by monitoring external network traffic.

**Duqu** (Bencsáth et al., 2012) was an attack that is supposedly related to Stuxnet. Duqu seemed to be aimed at acquiring sensitive information by collecting, for example, keystrokes, screenshots, browser history and system logs from the infected hosts. Similar to Stuxnet, Duqu made use of a P2P network aiming to connect protected systems to the Internet via less protected hosts that were connected to the Internet.

**Flame** (Virvilis and Gritzalis, 2013) (also known as Flamer or Skywriter) was an attack with the main objective of espionage, for example making screenshots and microphone recordings. An advanced way in which Flame propagated itself through the network was by impersonating the Windows Update Server. Moreover if Flame would infect a domain controller, it would infect the controlled hosts as well by creating backdoor accounts.

**Carbanak** (Kaspersky, 2015) was an attack campaign targeting financial institutes. Carbanak enabled cyber criminals to steal large sums of money (estimations vary between 300 million USD and 1 billion USD). By taking its time to learn typical user behaviour, the attackers were able to infect the right systems and stay under the radar at the same time.

Each of the studied attack campaigns show that there is a strong internal network traffic aspect present. Particularly in the *Expand access and obtain credentials*, *Strengthen foothold* and *Perform attack* stages of the attack life cycle of (Dell SecureWorks, 2013), it can be shown that an targeted attack might lead to abnormal internal network traffic patterns once the attacker has completed the initial intrusion. Supported by this, we argue that it is beneficial to take internal network traffic into account when looking for advanced network intrusions.

# 3 ANOMALY DETECTION

Another important aspect of targeted attacks is that they are typically highly sophisticated and targeted at specific organisations or sectors making use of new techniques and vulnerabilities. For this reason, we aim to develop an anomaly-based approach in detecting malicious internal network patterns that have not been seen before. In particular, we are interested in detecting unexpected and deviating communication behaviour of hosts on the internal network.

The challenge in developing this anomaly-based solution lies in creating an accurate model that describes normal communication behaviour of hosts. Not all hosts will display similar behaviour. A network consists of many different types of hosts such as workstations, file servers, domain controllers and printers. Behaviour that is typical for one type of host might be anomalous for another type of host. Therefore creating a single model for all hosts in a network might result in an inability to detect relevant anomalies. On the other hand, creating a different model for each of the hosts might result in very specific models that will detect even minor behavioural changes. As reasoned before, this is likely to significantly increase the false positive rate, decreasing the overall accuracy of the intrusion detection. For this reason, we propose to enhance the anomaly detection approach by making use of host clustering. Host that exhibit similar behaviour are clustered together and for each of the resulting clusters a behavioural model will be created. The behavioural model indicates which behaviour that is expected to be observed. Network activity not adhering to this model is marked as anomalous.

# 4 CLUSTERING

In many domains it is useful to identify patterns in a given set of data for analysis of other (similar) data. In other words, given training data, one wants to come up with a pattern that helps predicting the behaviour of (unseen) test data, or one wants to determine whether (unseen) test data is part of the same category as the training set. The process of recognising a pattern, called learning, is divided into *supervised* learning and *unsupervised* learning, also known as respectively *classification* and *clustering* (Jain, 2010).

The main goal of data clustering is to "discover natural grouping(s) of a set of patterns, points, or objects" (Jain, 2010). In literature, there are many ways to determine such natural groupings. There are numerous clustering methods available, all using different algorithms that may result in different clustering for the same input data. Research by Jain et al. (Jain et al., 2004) compared 35 different clustering algorithm and they were able to show that these algorithms can be divided into three groups (heuristic-based, density-based and model-based) giving similar results within these groups.

By means of these clustering techniques, it is possible to create groupings of hosts that are in some respect similar. This aspect of similarity might be a valuable contribution in creating a wider understanding of user behaviour.

The behaviour of hosts in a network follows to some degree regular patterns, with many small and harmless deviations, however. These deviations might cause high false positive rates in the case of modelling a single host. Clustering might overcome this phenomenon by taking the 'average' behaviour of similar hosts together. Since the hosts in a single cluster are in some respect alike, it can be argued that the differences in behaviour between the hosts are acceptable, and hence fall within the bounds of normal behaviour. Using this 'common' or 'average' behaviour of the clustered hosts, it might be possible to create a more accurate definition of what normal behaviour is for the hosts concerned.

Following this line of reasoning, this approach will reduce the number of false positives, as what might be seen as an anomaly for an individual host might not be an anomaly for hosts of the same cluster, hence not marking such events as anomalous. Yet, it is expected that it will not reduce the true positive rate, as true abnormalities are expected to be anomalous compared to the cluster's common behaviour as well.

In this work, *k*-means (MacQueen, 1967), Mean shift (Comaniciu and Meer, 2002), Louvain Community Clustering (Fortunato, 2010) and Stochastic Blockmodel Clustering (SBM) (Holland et al., 1983) are compared in the context of host clustering-based anomaly detection. The main differences between the algorithms are provided in table 1.

Table 1: Overview of the reviewed clustering methods, with their classes and type of input.

| Clustering method | Class | Type |
|---|---|---|
| *k*-means | Heuristic-based→ Pattern-based | Vector quantisation |
| Mean shift | Density-based → Mode seeking | Vector quantisation |
| Louvain | Density-based | Graph-based |
| Stochastic Blockmodel | Model-based | Graph-based |

*k*-means and Mean shift are both based on vector quantisation, meaning that the data subject to the clustering method has to be expressed as points in a multi-dimensional vector space. In this work, we investigate vector-based clustering methods based on host-

features such as the number of outgoing/incoming connections, the number of bytes sent and received, and the communication protocols used.

Graph-based clustering techniques focus on the communication partners of hosts, whereas vector-based approaches look at how (e.g. the number of bytes and the protocols used) a host is communicating. Louvain and SBM, on the other hand, are graph-based. Algorithms of this type of clustering require a graph, of which the connectivity (i.e. the adjacency matrix) is used to derive a clustering. As internal host networks data is relational, this type of clustering can be applied.

# 5 EVALUATION

An overview of the proposed anomaly detection method is provided in figure 1. See also (Beukema, 2016) for a more detailed description of the proposed approach.

**Training Phase.** Training data is the data used to create the model, which will serve as 'ground truth'. The behaviour of this data is assumed to be the 'normal behaviour' of the network under investigation. The data used in both training and test stage is flow data (Claise et al., 2015). Flow data is a common way of summarising network activity. Capturing flow data requires the network environment to have deployed internal network traffic flow probes, which are aggregated at a single system so they can be analysed. Table 2 shows the relevant flow information that was captured and that is used for clustering and anomaly detection.

**Normalisation.** After collection, the data is normalised first, which involves parsing the data into a common data format. Table 2 contains both numerical and categorical features. The categorical features are transformed into numerical ones by applying one-hot encoding (Harris and Harris, 2012). Moreover, any noise present in the data is removed in this phase. Typically lots of traffic is communicated over the network making it challenging to extract the relevant information. Network scans by other security products might, for example, obfuscate the relevant host behaviour. Removing noise from the data set is therefore an important step in order to reveal the relevant information. This is done by consulting experts with good knowledge of the network under consideration. These experts enable us to filter out network traffic, such as network scans, which is obfuscating relevant information.

**Host Clustering.** After the normalisation, it is possible to obtain a clustering of the hosts in the network by applying one of the aforementioned clustering techniques. The result of this stage is a mapping for each IP address to a single cluster. The two vector quantisation clustering methods (Table 1) group the hosts based on the numerical features obtained in the previous normalisation phase. The graph-based clustering methods group only make use of the (weighted) graph structure of the network. In this graph the nodes represent the hosts and the weight of the edges is determined by the number of flows from the source to the destination host.

**Modelling.** The obtained clustering is used as input for the modelling stage, in which for each cluster a definition of common behaviour is determined. This definition consists of several *features* that describe certain aspects of the behaviour of hosts within the cluster.

Typically, histogram-based approaches have been used in the past for this type of anomaly detection (Denning, 1987; Eskin, 2000; Eskin et al., 2001). Sometimes referred to as frequency or counting-based approaches, it is a simple non-parametric statistical technique in which histograms are used to maintain a profile of the 'normal' data.

There are some strong drawbacks to the histogram-based approach. Most importantly, considering combinations of features is inefficient using this approach. For $n$ features, one could just consider $n$ histograms. However, with this approach, combinations of values are not taken into account (e.g. bytes vs. duration, destination vs. port number). In the end, $n!$ histograms are needed to cover all possible combinations of features. To overcome this, we use Support Vector Machines (SVM) (Boser et al., 1992; Schölkopf et al., 2001) to model the normal behaviour of the different clusters. SVMs have been as a network-based anomaly classifier before in other works (Zhang et al., 2015), (Eskin et al., 2002), (Li et al., 2009). The big advantage of choosing SVM over an histogram-based approach is that all combinations of features are automatically taken into account. In other words, each entry is evaluated in terms of how anomalous the combination of all features taken together is.

Thus, an SVM is created for each cluster and trained with flows from the training data. Specifically, each flow is represented in the SVM using the following seven features:

| | | |
|---|---|---|
| Bytes_A→ B | Bytes_A←B | Relative_Start |
| Packets_A→B | Packets_A←B | Duration |
| Destination_Cluster | | |

Except for the last attribute, all are numeric attributes and can therefore be directly processed by an SVM. The latter is a categorical attribute, and hence

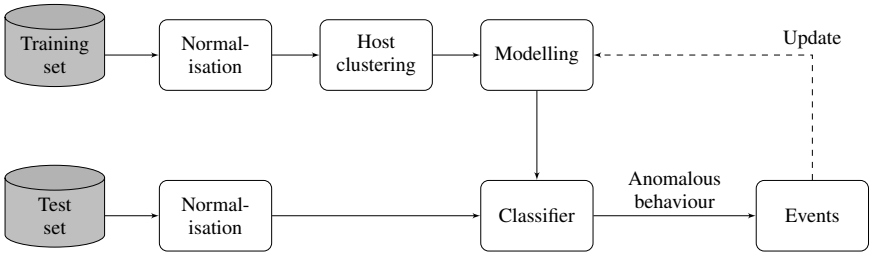


Figure 1: Flow diagram of the proposed host clustering-based anomaly detection algorithm.

Table 2: Overview of used IPFIX fields for anomaly detection.

| Name | ID | Description |
|---|---|---|
| octetDeltaCount | 1 | The number of octets since the previous report (if any) in incoming packets for this Flow. |
| packetDeltaCount | 2 | The number of incoming packets since the previous report (if any) for this Flow. |
| deltaFlowCount | 3 | The conservative count of Original Flows contributing to this Aggregated Flow. |
| sourceTransportPort | 7 | The source port identifier in the transport header. |
| sourceIPv4Address | 8 | The ipv4 source address in the IP packet header. |
| destinationTransportPort | 11 | The destination port identifier in the transport header. |
| destinationIPv4Address | 12 | The ipv4 destination address in the IP packet header. |
| flowEndSysUpTime | 21 | The relative timestamp of the last packet of this Flow. |
| flowStartSysUpTime | 22 | The relative timestamp of the first packet of this Flow. |
| postOctetDeltaCount | 23 | The number of octets since the previous report (if any) in outgoing packets for this Flow. |
| postPacketDeltaCount | 24 | The number of outgoing packets since the previous report (if any) for this Flow. |
| sourceIPv6Address | 27 | The ipv6 source address in the IP packet header. |
| destinationIPv6Address | 28 | The ipv6 destination address in the IP packet header. |
| ipVersion | 60 | The IP version field in the IP packet header. |

is converted to a numerical one by applying one-hot encoding. Hence, the number of attributes per flow is equal to $a = 6 + c$ for $c$ the total number of clusters.

After the training is complete, each SVM describes the boundaries in which the training data operates. These boundaries are different for each cluster and will be used in the next phase to identify anomalies.

**Test Phase.** Test data (sometimes '*validation data*') refers to the data being tested, i.e. the data in which the algorithm will look for anomalous behaviour. It should be captured in similar circumstances as the training set in order to be able to make a valid comparison with the training data. The test data is normalised in the same way the training data is. In this phase, the actual anomaly detection takes place.

**Classifier.** The classifier takes the model derived at the training phase and the normalised data from the test phase and compares them. It is assessed whether the characteristics of the network traffic as observed in the test data matches the behaviour as defined in the model. In other words, for each flow, it is tested whether it adheres to the behaviour as defined in the model corresponding to the cluster the host initiating the flow belongs to.

Each flow in the test data is compared to its corresponding SVM, which was created during the training phase. The SVM will return the distance to the derived boundary for each test data flow, which is an indication of the extent to which the flow is similar to the training data. If behaviour of a host is anomalous to such an extend that it exceeds a set threshold, the behaviour will be marked as anomalous.

**Events.** After the classifier has completed its work, it will return a set of events that should be investigated further. The resulting events have to be evaluated by the operator of the proposed detection system, upon which it might be decided to take further action. These evaluations may be used for additional training of the model. The advantage of this is twofold: the system gets new, labelled input that will increase its ability to distinguish between anomalous and non-anomalous behaviour, and secondly it allows the system to learn and adapt to changing behaviour.

# 6 DISCUSSION

Using the UNB ICSX 2012 set (Shiravi et al., 2012), the methodology as set out in section 5 is carried out. The set comprises seven consecutive days (11th of June - 17th of June 2012) of IPFIX flow data (Table 2). The data included in the set is simulated data based on characteristics of real network data. For each day of the data set, there are up to 145 different hosts generating up to 74,370 internal network flows. Most interestingly, three of the seven days contain malicious internal network flows (Sunday 13 June - Tuesday 15 June). Since most data is labelled (albeit only the last six days), this set is suited for our purpose of evaluating the proposed methodology. Data from Saturday 12 June is used for model creation, while the proposed intrusion detection system is evaluated against the labeled data of Sunday, Monday and Tuesday.

The flow-based detection aims to detect anomalous network flows. Hence each network flow is compared to the SVM that is created in the modelling phase. Receiver Operating Characteristic (ROC) curves for flow-based anomaly detection are given in Figure 2 (a-c). These ROC curves plot the ratio of true positives against the ratio of false positives. ROC curves enable us to compare the performance of different detectors. The four clustering methods are compared to single model and per-host model benchmark approaches.

There are a number of conclusions that can be drawn from these results. First of all, it shows that all six different approaches are able to detect the malicious network traffic at the cost of some false positives. Hence by monitoring internal network traffic we are indeed able to detect malicious network traffic. Moreover, we see that in most situations clustering performs better in terms of detection compared to using no clustering or per-host modelling. We see that Stochastic Blockmodel clustering in all cases is better than applying no clustering. Both Mean-shift and Louvain compare in some cases better, in some cases worse than no clustering. $k$-means performs in almost all cases worse than applying no clustering.

Within the scope of detection of targeted attacks, it is more relevant to look at the classification of hosts rather than the classification of flows. For instance, an infected machine might initiate flows that are in itself barely anomalous, but the number of these flows together makes the machine's behaviour anomalous altogether. One could therefore accumulate the scores of each flow initiated by a host, and assign that score to the host. This enables us to identify anomalous hosts rather than just anomalous flows.

Therefore, it is also useful to consider ROC graphs for the detection rate of anomalous hosts. Figure 2 (d-f) shows the ROC curves per clustering method for Sunday the 13th of June till Tuesday the 15th of June, using Saturday the 12th of June as a training data. Within this dataset hosts are labelled as malicious or infected when they initiate at least one labeled connection. Note that in the dataset used, the number of hosts is rather small. Therefore these ROC curves are based on significantly less evaluations, hence the low resolution of the ROC curves.

For the 13th of June, it does not matter whether or not clustering is applied - this is probably due to the fact that the attack carried out on that day is so obvious that traditional methods are able to detect it as well. For the 14th of June, all clustering methods perform better on almost all false-positive ranges than both not applying clustering and clustering per host. The same conclusion applies to the 15th of June, although the difference is slightly smaller.

The results lead to the conclusion that internal network traffic is a useful source of information that can enable detection of malicious activity inside a network. Important phases of targeted attacks can be detected through analysis of internal network traffic, which remains unnoticed by most IDSs that focus on external network traffic.

Moreover, the clustering approach we propose, improves the performance of anomaly-based detection mechanisms. We have shown that there is always at least one clustering approach that performs better than the two benchmark approaches. Also, we showed that all clustering methods perform better when the method is applied to identifying malicious hosts. Specifically, SBM clustering offers a consistently better detection performance for detecting malicious flows. Hence, it is important to carefully consider the clustering method depending on the type of anomaly that must be detected.

If we compare the the trade-off between the processing burden and the performance, the proposed approach is the best option as well. The computational complexity of SVM is $O(n^3)$ for $n$ training samples (Bordes et al., 2005); therefore it is more efficient to have multiple SVM instances with some training samples than a single SVM with all the training samples. As such, detection applying a single model is computationally demanding, whereas per-host modelling has a relatively low computational cost. In this respect, our approach is in between of these two. Hence, the proposed approach improves the detection accuracy and diminishes the processing burden in comparison to single-model detection.

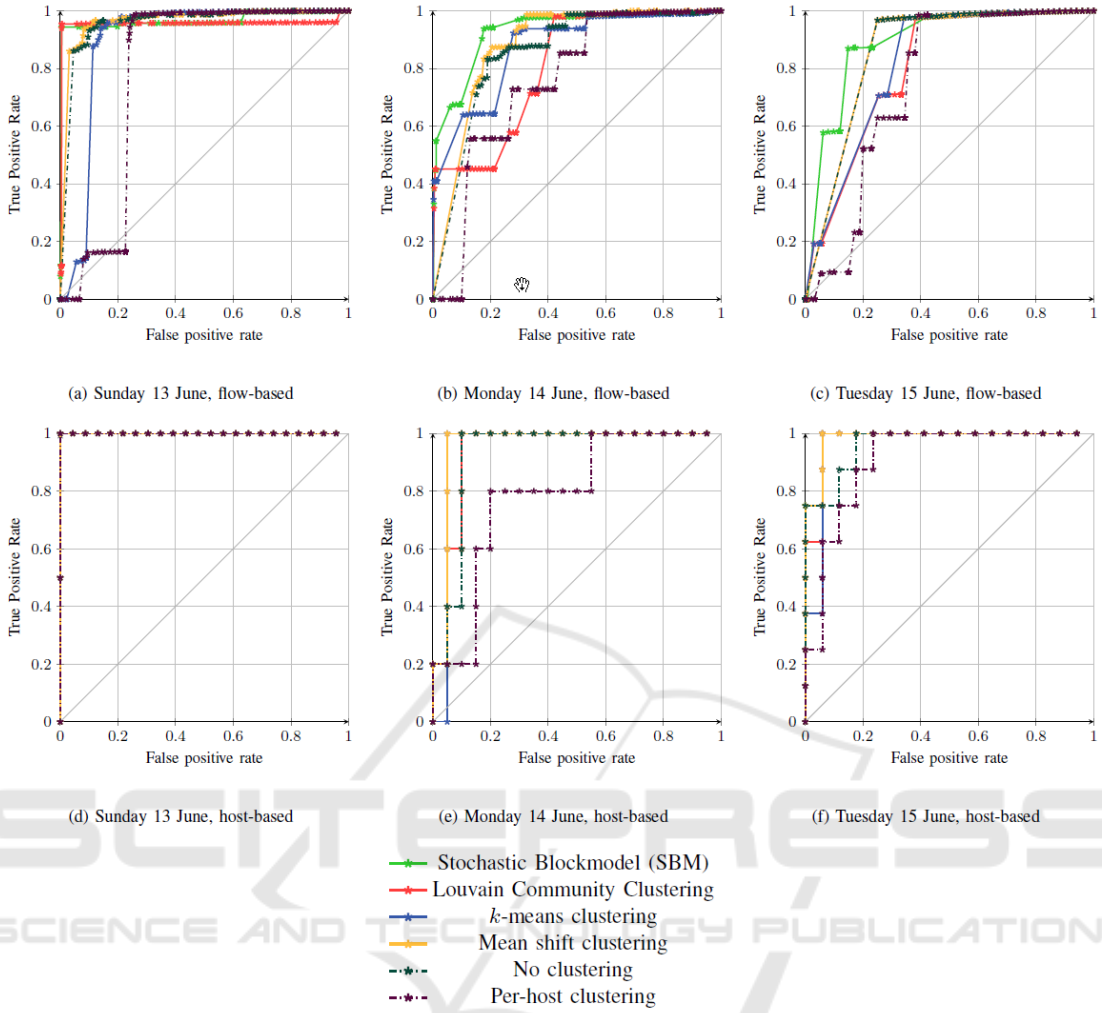


Figure 2: ROC curves for the UNB ICSX 2012 data set.

## 7 CONCLUSION

The increasing complexity of targeted attacks was primary motivation for this research. New and unseen attacks are frequently not detected by misuse-based approaches whereas high false positives rates may plague anomaly-based detection. Moreover, internal network traffic is often overlooked by many intrusion detection systems yet still is a valuable information source. We investigated how well anomaly detection applying host clustering based on internal network traffic performs, in order to improve current detection mechanisms, focusing on detection of targeted attacks.

We have developed an anomaly detection system based on internal flow data as an initial demonstration of the usefulness of internal network traffic analysis. Other information sources, for instance DNS-queries, Identity and Access Management logs and proxy logs, may also also suitable for revealing internal communication patterns and finding anomalous behaviour therein. Since the developed framework is generic further research could focus on the application of the proposed methodology on other information sources.

For furture work, the authors suggest evaluating more clustering techniques and anomaly detection algorithms. There are many more techniques available. Furthermore, the choice for the optimal clustering technique highly depends on the detection algorithm that is used. Hence, further research is necessary to apply application of clustering to be used for anomaly detection.

# REFERENCES


Akoglu, L., Tong, H., and Koutra, D. (2015). Graph based anomaly detection and description: a survey. *Data Mining and Knowledge Discovery*, 29(3):626–688.

Bencsáth, B., Pék, G., Buttyán, L., and Félegyházi, M. (2012). Duqu: Analysis, detection, and lessons learned. In *ACM European Workshop on System Security (EuroSec)*, volume 2012.

Beukema, W. J. B. (2016). Enhancing network intrusion detection through host clustering. Master's thesis, University of Twente.

Bordes, A., Ertekin, S., Weston, J., and Bottou, L. (2005). Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6(Sep):1579–1619.

Boser, B. E., Guyon, I. M., and Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, pages 144–152. ACM.

Byrne, M. D. (2013). How many times should a stochastic model be run - An approach based on confidence intervals. In *Proceedings of the 12th International conference on cognitive modeling*, Ottawa.

Claise, B., Quittek, J., Meyer, J., Bryant, S., and Aitken, P. (2015). Information Model for IP Flow Information Export. doi:http://dx.doi.org/10.17487/rfc510210.17487/rfc5102.

Comaniciu, D. and Meer, P. (2002). Mean shift: A robust approach toward feature space analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(5):603–619.

Dell SecureWorks (2013). Advanced persistent threat analysis. Accessed on 21/01/2016.

Denning, D. E. (1987). An intrusion-detection model. *IEEE Transactions on Software Engineering*, 13(2):222–232.

Ehrlich, W. K., Karasaridis, A., Hoeflin, D. A., and Liu, D. (2010). Detection of spam hosts and spam bots using network flow traffic modeling. In *LEET*.

Eskin, E. (2000). Anomaly detection over noisy data using learned probability distributions. In *In Proceedings of the International Conference on Machine Learning*. Citeseer.

Eskin, E., Arnold, A., Prerau, M., Portnoy, L., and Stolfo, S. (2002). A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pages 77–101. Springer.

Eskin, E., Lee, W., and Stolfo, S. J. (2001). Modeling system calls for intrusion detection with dynamic window sizes. In *DARPA Information Survivability Conference & Exposition II, 2001. DISCEX'01. Proceedings*, volume 1, pages 165–175. IEEE.

Fortunato, S. (2010). Community detection in graphs. *Physics Reports*, 486(3):75–174.

Harris, D. and Harris, S. (2012). *Digital design and computer architecture*. Elsevier.

Holland, P. W., Laskey, K. B., and Leinhardt, S. (1983). Stochastic blockmodels: First steps. *Social Networks*, 5(2):109 – 137.

Hutchins, E. M., Cloppert, M. J., and Amin, R. M. (2011). Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. *Leading Issues in Information Warfare & Security Research*, 1:80.

Jain, A. K. (2010). Data clustering: 50 years beyond K-means. *Pattern Recognition Letters*, 31(8):651–666.

Jain, A. K., Topchy, A., Law, M. H. C., and Buhmann, J. M. (2004). Landscape of Clustering Algorithms. In *Proceedings of the Pattern Recognition, 17th International Conference on (ICPR'04) Volume 1 - Volume 01*, ICPR '04, pages 260–263, Washington, DC, USA. IEEE Computer Society.

Kaspersky (2015). Carbanak APT: The Great Bank Robbery. Accessed on 18/2/2016.

Langner, R. (2011). Stuxnet: Dissecting a cyberwarfare weapon. *IEEE Security and Privacy*, 9(3):49–51.

Lazarevic, A., Kumar, V., and Srivastava, J. (2005). Intrusion detection: A survey. In Kumar, V., Srivastava, J., and Lazarevic, A., editors, *Managing Cyber Threats: Issues, Approaches, and Challenges*, pages 19–78. Springer US, Boston, MA.

Li, Y., Wang, J.-L., Tian, Z.-H., Lu, T.-B., and Young, C. (2009). Building lightweight intrusion detection system using wrapper-based feature selection mechanisms. *Computers & Security*, 28(6):466 – 475.

MacQueen, J. (1967). Some methods for classification and analysis of multivariate observations. In *Proceedings of the fifth Berkeley symposium on mathematical statistics and probability*, volume 1(14), pages 281–297. Oakland, CA, USA.

Research and Markets (2015). Advanced persistent threat protection market - global forecast to 2020.

Roy, D. B. and Chaki, R. (2014). State of the art analysis of network traffic anomaly detection. In *Applications and Innovations in Mobile Computing (AIMoC), 2014*, pages 186–192. IEEE.

Sabahi, F. and Movaghar, A. (2008). Intrusion Detection: A Survey. In *Systems and Networks Communications, 2008. ICSNC '08. 3rd International Conference on*, pages 23–26.

Scarfone, K. A. and Mell, P. M. (2007). SP 800-94. Guide to Intrusion Detection and Prevention Systems (IDPS). Technical report, National Institute of Standards & Technology, Gaithersburg, MD, United States.

Schölkopf, B., Platt, J. C., Shawe-Taylor, J., Smola, A. J., and Williamson, R. C. (2001). Estimating the support of a high-dimensional distribution. *Neural computation*, 13(7):1443–1471.

Shiravi, A., Shiravi, H., Tavallaee, M., and Ghorbani, A. A. (2012). Toward developing a systematic approach to generate benchmark datasets for intrusion detection. *Computers & Security*, 31(3):357 – 374.

Virvilis, N. and Gritzalis, D. (2013). The Big Four - What We Did Wrong in Advanced Persistent Threat Detection? In *Availability, Reliability and Security (ARES), 2013 Eighth International Conference on*, pages 248–254.

Wei, S., Mirkovic, J., and Kissel, E. (2006). Profiling and clustering internet hosts. *DMIN*, 6:269–75.

Xiao, H., Gao, J., Turaga, D. S., Vu, L. H., and Biem, A. (2015). Temporal multi-view inconsistency detection for network traffic analysis. In *Proceedings of the 24th International Conference on World Wide Web*, pages 455–465. ACM.

Xu, K., Wang, F., and Gu, L. (2011). Network-aware behavior clustering of internet end hosts. In *INFOCOM, 2011 Proceedings IEEE*, pages 2078–2086. IEEE.

Xu, R., Wunsch, D., et al. (2005). Survey of clustering algorithms. *Neural Networks, IEEE Transactions on*, 16(3):645–678.

Zhang, M., Xu, B., and Gong, J. (2015). An Anomaly Detection Model Based on One-Class SVM to Detect Network Intrusions. In *2015 11th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, pages 102–107.