Identifying suspicious users in corporate networks

Tomáš Pevný, Martin Rehák, Martin Grill

***Abstract*—This paper focuses on the identification of suspicious users in corporate networks, which can be so due to many reasons: e.g. being victims of an attack or performing it, ex-filtering sensitive information, downloading files through p2p network, etc. The proposed detection method models users' incoming and outgoing traffic without inspecting the actual content of network packets. Since it simultaneously has a very low computational complexity, the detection is almost instant, which is important for limiting the potential damages. The additional benefit is that the method works on encrypted networks as well.**

The proposed method uses entropies of distributions of IP addresses and ports to build two complementary models of user's traffic. These two models are coupled with two orthogonal anomaly measures based on the principal component transformation, which gives four different detectors.

The proposed detectors are experimentally evaluated and compared to adapted prior art on one week long capture of traffic on university network. The experiments reveals that no single detector can detect all types of anomalies, but together they can detect most of them. This result is expected and stresses the importance of ensemble approach towards intrusion detection.

I. Introduction

The recent years observe a rapid grow of cyber-crime, e.g. corporate servers and systems being hacked into and the sensitive data being stolen, web servers becoming victims of denial of service, or other types of attack launched against them. The danger can also comes from within the network, as users try to exfiltrate sensitive information, or participate in other undesirable behavior such as downloading content from p2p networks. This emphasizes the need to improve the not only the network intrusion detection systems (NIDS) (nowadays a standard component of security measures), but also the monitoring of internal network.

This paper describes family of detectors aimed to identify suspicious users, which can be so due to different reasons listed in the previous paragraph. It is important to identify them soon, such that their behavior can be quickly inspected in more detail and potential damage limited. Our detectors do not inspect the contents of the packets, as they rely solely on statistics in the NetFlow [2] data, which is exactly on par with this requirement. In enables them to operate at high speed and scrutinize many users simultaneously.

The models used by the detectors are based on aggregated statistics of user's traffic, namely on entropies of distributions source and destination ports and IP addresses. These quantities are usually correlated between users, which, together with the assumption that users' aggregated traffic varies slowly, allow us to use principal component analysis to build model of the benign behavior.

We build two separate models, (i) one for users' incoming traffic and (ii) one for users' outgoing traffic. This, together with two anomaly measures gives four different anomaly detectors. The low complexity of the detectors makes them well suited (i) for real-time intrusion detection systems (IDS), and (ii) enables to model all users within the network.

Our detectors were inspired by the work of Lakhina et al. [6], which aimed to detect anomalies on peering links between different sub-networks. His model captured state of the whole backbone network. Contrary, we focus on modeling users, because we believe that for our application scenario, it is important to know who caused the anomaly (and who is therefore suspicious) instead of knowing that there is an anomaly. The other differences to this work include very low computational complexity and short update time. This improves the capability to quickly adapt the models to changes in the traffic.

We have estimated the detection accuracy of the presented detectors on one week long capture of the traffic on university network manually labeled by an experienced operator. It shows that the detectors can well detect different types of malicious and unwanted user's behavior. The comparison to the prior art adapted to our application scenario also shows the superiority of our work. We believe that if the detectors are applied in enterprise networks, the detection accuracy would be even better, because they are usually better maintained then university networks, which are very wild.

The paper is organized as follows. The next section reviews the prior art, namely the Lakhina's detector [6] and discusses, why straightforward adaptation to our application scenario is difficult. Section III describes our models of user's traffic together with the anomaly detection algorithms based on principal component analysis. We also explicitly differentiate our work against the prior art on the conceptual level. The experimental evaluation and empirical comparison to the selected prior art is in Section IV.

Throughout this paper, we use several terms which we clarify here. The *flow* is an unidirectional connection between two users (computers) fully determined by the following tuple: (source IP address, source IP port, destination IP address, destination port, procol). One flow usually comprises of many packets, and thus it is associated with additional statistics, such as number of packets,

Agent Technology group, Czech Technical University in Prague

number of bytes, starting and ending time of the flow, etc. For a full list, we refer the reader to the definition of Cisco's NetFlow [2] format. We emphasize that in this work, we do not use these additional statistics. By *aggregation* of flows (or traffic), we mean set of flows sharing some property. For example, flows aggregated by source IP address means set of flows with the same source address. On aggregated flows, we can measure an empirical distribution of ports or and IP addresses. Since the support of these distributions is discrete and finite, they are simple histograms. Finally, from these histograms we can easily calculate the entropy by the usual formula $\mathrm{H}(x) = -\sum_{i=1}^{k} p_k \log p_k$, where p_k denotes the probability of k^{th} bin.

II. Lakhina entropy detector

This section reviews Lakhina's entropy detector [6], which has been designed to detect anomalies on links between multiple sub-networks. The detector builds model of traffic on links, and identify those links that are significantly different from all other links. Because of different scope of the prior art, we later discuss issues related to the straightforward adaptation of the detector to the identification of suspicious users, which is the focus of this paper.

A. Description of the detector

This subsection assumes that the information about the flows is acquired from points of presence (PoP), which are routers connecting sub-networks to the backbone. The flows are aggregated at the level of Origin-Destination flows (OD flows), which comprises of all flows between selected pair of PoPs. In Ref. [6], authors propose two detectors, one using volume statistics of OD flows, the second using entropy based statistics. Here, we focus on the entropy based model, but the same mechanism is used for volume model, albeit with different quantities.

Denoting $\boldsymbol{\Upsilon}$ the set of all pairs of PoP, the detector capture the state of the traffic on the backbone at the time window t in the feature vector

$$x^t = \left(\mathrm{H}^t_{\mathrm{sPr}}(\upsilon), \mathrm{H}^t_{\mathrm{dPr}}(\upsilon), \mathrm{H}^t_{\mathrm{sIP}}(\upsilon), \mathrm{H}^t_{\mathrm{dIP}}(\upsilon) | \upsilon \in \boldsymbol{\Upsilon}\right) \in \mathbb{R}^{4\times|\boldsymbol{\Upsilon}|}, \tag{1}$$

where $\mathrm{H}^t_{\mathrm{sPr}}(\upsilon)/\mathrm{H}^t_{\mathrm{dPr}}(\upsilon)$ is entropy of distributions of source/destination ports and $\mathrm{H}^t_{\mathrm{sIP}}(\upsilon)/\mathrm{H}^t_{\mathrm{dIP}}(\upsilon)$ is entropy of distributions of source/destination IP addresses of all OD flows between υ^{th}-pair of PoPs. All entropies are calculated from traffic observed during five-minute long time windows.

The model of the state of the traffic on the network is built by means of principal component transformation (PCT). Feature vectors from previous τ time windows $x^{t-\tau}, \ldots, x^{t-1}$ are arranged as rows in the matrix

$$\mathbf{X} = \begin{pmatrix} x^{t-1} \\ \vdots \\ x^{t-\tau} \end{pmatrix} \in \mathbb{R}^{\tau,N}. \tag{2}$$

PCT applied on $\mathbf{X}$ returns set of principle components $\{y_j \in \mathbb{R}^{4\times|\boldsymbol{\Upsilon}|}\}_{j=1}^{4\times|\boldsymbol{\Upsilon}|}$ ordered according to their variance. This means that first components have highest variance, and consequently contain the most information about x^t. $\{y_j\}_{j=1}^{4\times|\boldsymbol{\Upsilon}|}$ can be calculated as eigenvectors of the covariance matrix $\mathbf{C} = \mathbb{E}\left[(\mathbf{X}-\mu)^T(\mathbf{X}-\mu)\right]$, where μ is a row-vector containing means of columns of $\mathbf{X}$.

Lakhina's detectors rely on the assumption that the subspace spanned by the first k principal components $\mathbf{Y}_{1:k} = (y_1, \ldots, y_k)$ corresponds to the normal traffic subspace. This subspace has projection matrix $\mathbf{P}_{1:k} = \mathbf{Y}_{1:k}\mathbf{Y}^{\mathrm{T}}_{1:k}$. Consequently, the residual subspace spanned by components $\mathbf{Y}_{k+1:4\times|\boldsymbol{\Upsilon}|} = (y_{k+1}, \ldots, y_{4\times|\boldsymbol{\Upsilon}|})$ has the projection matrix $\mathbf{P}_{k+1:4\times|\boldsymbol{\Upsilon}|} = \mathbf{I} - \mathbf{P}_{1:k}$. By virtue of the assumption, the residual space contains the anomalous traffic.

To assess the level of anomalousness of OD flows at time t, the feature vector x^t is decomposed into the part modeled by the normal subspace, $\bar{x}^t$, and by the anomalous subspace, $\tilde{x}^t$. It holds that $x^t = \bar{x}^t + \tilde{x}^t$. If any component of $\tilde{x}^t$ corresponding to the traffic between two points of presence exceeds the design threshold, all traffic is deemed to be anomalous.

The original publication [6] claims that first four components determines the subspace where the legitimate traffic lies. But it has been shown by Ringberg et al. [9] that the accuracy of the algorithm is very sensitive to the settings of this parameter, which makes the application in practice dubious.

B. Adaptation to user-level detection

The detector, as described above, detects anomalies on the level of links between the pairs of points of presence. We believe that in reality this is not sufficient. We want to know who caused the anomaly and can be a potential attacker and who is the victim, not just that there is an anomaly (although this is important as well). Thus we need to detect anomalies at finer level, which we do by identifying suspicious users.

To detect anomalies at this level of detail, the scope of the original model described above would need to be changed from links between pairs of PoPs to links between pairs of users (represented by IP addresses). To implement this change, the feature vector (1) would have to accommodate all links between all pairs of users used during training and detection phases of the detector. Since the dimension of the feature vector x^t scales quadratically with the number of users (PoPs), the dimension would increase so much that the detector will be practically unusable. For example the traffic on network with 1000 users will be described by the feature vector of dimension $4 \cdot 10^6$. This huge dimension has two immediate consequences:

- Number of samples (previously observed feature vectors), τ, in (2) needed to build the model will increase beyond the point, where the samples are (i) no longer relevant to the present situation and (ii) are almost impossible to acquire. Notice that the number of samples, τ, has to be greater than the dimension of the feature vector. If we have network with 1000 users,

	$t-4$	$t-3$	$t-2$	$t-1$	t
H_{dIP}	0.74	0.76	0.76	0.80	1
H_{dPr}	0.70	0.71	0.72	0.78	1
H_{sPr}	0.41	0.44	0.49	0.56	1

(a) Time correlations

	H_{dIP}	H_{dPr}	H_{sPr}
H_{dIP}	1	0.70	-0.85
H_{dPr}	0.70	1	-0.95
H_{sPr}	-0.85	-0.95	1

(b) Space correclation

Table I: Average correlation of entropies of calculated with respect to the time (Figure (a)), and average correlation of entropies within the same time window (Figure (b)).

we need more than $4 \cdot 10^6$ samples. If we assume 5-minute time window for each sample, then the time span of samples would be approximately 190 years.

- The complexity of PCT depends cubically on the dimension of the feature vector x^t. Consequently, the calculation of the PCT transformation might be impossible, or at the time it will be calculated, it will be no longer relevant.

To address these issues, the [8] simplified Lakhina's detectors as follows: (a) it uses statistics aggregated only by source IP addresses, and (b) it uses only latest five time windows to build the model [1]. The feature vector in [8] equals to

$$x^t = \left(\mathrm{H}^t_{\mathrm{sPr}}(\iota), \mathrm{H}^t_{\mathrm{dPr}}(\iota), \mathrm{H}^t_{\mathrm{dIP}}(\iota) | \iota \in \mathbf{I}\right), \tag{3}$$

where $\mathbf{I}$ is the set of active IP addresses in the given time window. Feature vectors $x^{t-5}, \ldots, x^{t-1}$ are used to build the model for detection at the time window t. In order to decrease the computational time, only addresses with more than 100 flows are modeled, and their number is limited to 1500 (consequently the maximum dimension of feature vector (3) is 4500). Otherwise, the quantity determining the level of anomalousness is the same.

The work [1] also adapts Lakhina's detector, but the exact details of the model are not clearly described.

III. Proposed anomaly detection

This section describes the proposed models of users' behavior based on their network traffic. To be exact, two complementary models are proposed — one based on its incoming traffic and one based on its outgoing traffic.

The model based on user's outgoing traffic (flows aggregated by the source IP) uses entropies of distributions of destination IP addresses, source ports, and destination ports of all its outgoing flows. Similarly, the model based on user's incoming traffic (flows aggregated by the destination IP) uses entropies of distributions of source IP addresses, source ports, and destination ports of all incoming flows. All entropies are calculated from all flows observed during 5-minute long time windows.

[1]The use of time-windows cannot be easily avoided, since sufficient statistics is need to estimate entropy of distribution.

The rationale behind models is following. Most of the time, users' behavior changes slowly with the respect to the chosen five minute time interval. This can be observed by a strong correlation of witnessed statistics from different time windows. We have exemplified this phenomenon in Table I(a), where time correlations are estimated from 50 consecutive captures of five minute long traffic windows (4 hours and 10 minutes) on university network with approx. 15 000 flows per minute. This time correlation was already exploited for example in [13], [12], [5], [6], but in the different framework of anomaly detection on the backbone traffic.

The behavior of users of the same network is also similar to each other, which means that witnessed statistics of different users in the same time window are correlated. This phenomenon is showed in Table I(b) for the aggregation over the source IP address (outgoing traffic). The similar holds for the aggregation over destination IP address.

A. Model of the network

Our models simultaneously exploit both types of aforementioned correlations — correlations with respect to time and correlations between the users within the same time period. We construct two separate models, one with aggregation over *source* IPs (modeling outgoing traffic), and one with aggregation over *destination* IPs (modeling incoming traffic). Since the models differ only in the aggregation, they are explained below on the aggregation with respect to the source IP address.

Denoting quantities calculated at the time step t by the same superscript, the behavior of one user $\iota \in \mathbf{I}$ (ι corresponds to one source IP address, $\mathbf{I}$ denotes set of all modeled IP addresses) is described by the following vector of dimension 15:

$$x^t(\iota) = \left(\mathrm{H}^\tau_{\mathrm{sPr}}(\iota), \mathrm{H}^\tau_{\mathrm{dPr}}(\iota), \mathrm{H}^\tau_{\mathrm{dIP}}(\iota) | \tau \in \{t-4, \ldots, t\}\right).$$

The vector $x^t(\iota)$ effectively describes user's traffic at five consecutive time windows $t-4, \ldots, t$. The entropies are calculated iff the number of flows associated with the user ι is higher than 1 (if there is only one flow originating at given source IP, or ending at given destination IP, than all entropies are equal to zero). Consequently, for successful detection, at least two flows per five minutes are needed.

As has been explained above, individual items of the feature vector x^t are correlated. It is highly likely that anomaly behaving users express different type of correlation then benign ones. By virtue of the assumption that most users are benign, PCT is used to build the model of traffic of benign users. This model is used to identify anomalies among users by using variance on major and minor components [11]. The rest of this section describes the technical details.

The model used to detect anomalies at time t is built from data acquired at time windows $t-5, \ldots, t-1$. The feature vectors, $\left\{x^{t-1}(\iota_i)\right\}_{i=1}^{|\mathbf{I}|}$, corresponding to all active IP address at these time windows, are arranged in the data

matrix

$$\mathbf{X} = \begin{pmatrix} x^{t-1}(\iota_1) \\ \vdots \\ x^{t-1}(\iota_{|\mathbf{I}|}) \end{pmatrix} \in \mathbb{R}^{|\mathbf{I}|,15}. \quad (4)$$

This data-matrix is used by PCT to calculate set of orthogonal vectors $\{y_j \in \mathbb{R}^{15}\}_{j=1}^{r}$ together with eigenvalues $\lambda_1 > \lambda_2 > \ldots > \lambda_r$. For numerical stability, all vectors y_j corresponding to eigenvalues λ_j smaller than 10^{-6} are discarded (consequently $r \leq 15$).

To assess anomaly level of user ι (feature vector $x^t(\iota)$) at time window t, following two quantities are used:

$$\begin{aligned} f(x^t(\iota)) &= \sum_{j=1}^{k} \frac{(y_j^T x^t(\iota))^2}{\lambda_j^2}, \\ f^{\perp}(x^t(\iota)) &= \sum_{j=k+1}^{r} \frac{(y_j^T x^t(\iota))^2}{\lambda_j^2}. \end{aligned} \quad (5)$$

$f(x^t(\iota))$ captures variance on major components, while $f^{\perp}(x^t(\iota))$ captures variance on minor components. If one quantity exceeds corresponding threshold $\delta_\alpha/\delta_\alpha^{\perp}$, user ι is deemed as being anomalous. Based on the preliminary experiments, the split between major and minor components is set to $k = 1$.

The choice of these two measures is not ad-hoc. It has been showed [11] that if data (rows of matrix $\mathbf{X}$) follows a multivariate normal distribution, then distributions of $f(y)$ and $f^{\perp}(y)$ follow F-distributions (this theoretically enables to set thresholds δ_α, $\delta_\alpha^{\perp}$ analytically). However in practice, the condition of multivariate normality is likely to be violated and thresholds δ_α and $\delta_\alpha^{\perp}$ have to be determined empirically from the observed data. In this paper, we do not solve this problem. To compare the detection algorithms, we use ROC curves which are agnostic to the choice of the threshold.

In practice, we of course need to set the detection threshold. We suggest to use adaptive techniques described in [7], which determine the threshold adaptivelly according to the present situation.

B. Difference to the prior art

Although our model might resemble the model proposed by Lakhina et al. [6] (see Section II), there are substantial differences which we clarify below.

Whereas Lakhina's approach models traffic on peering links between two PoPs, we model typical behavior of users. Consequently, the our model enables identification of anomalies at finer level.

The computational complexity of our method is significantly lower than Lakhina's. The most complex operation in both algorithms is the principal component transformation with complexity $O(d^3)$, where d is the dimension of the model. Since the proposed model has fixed dimension 15, the computational complexity of the proposed method is $O(15^3)$ irrespectively to the number of users. On the other hand, the computational complexity of Lakhina's method grows in the order of $O(|\Upsilon|^3)$, where $|\Upsilon|$ is the number of peering links (remember that the number of peering links grows quadratically with number of subnetworks).

The smaller dimension of the proposed model also propagates to reduction of the number of samples needed to build the model. Rubinstein et al. [10] report using one week of data to build the model of the network based on (2), whereas our model (4) needs only five consecutive time windows (25 minutes).

The last two features of our approach (small computational complexity and the requirement on smaller set of observations for building the model) allows a frequent update of underlying models, which helps to keep the model up to date to reflect the present situation.

Last but not least, Lakhina's work used square prediction error [4] as an anomaly measure, whereas our approach uses variance on major and minor components. The square prediction error measure ignores the information carried on minor components, which, according to our experimental results, is useful for identification of users connected in p2p networks.

IV. Experimental evaluation

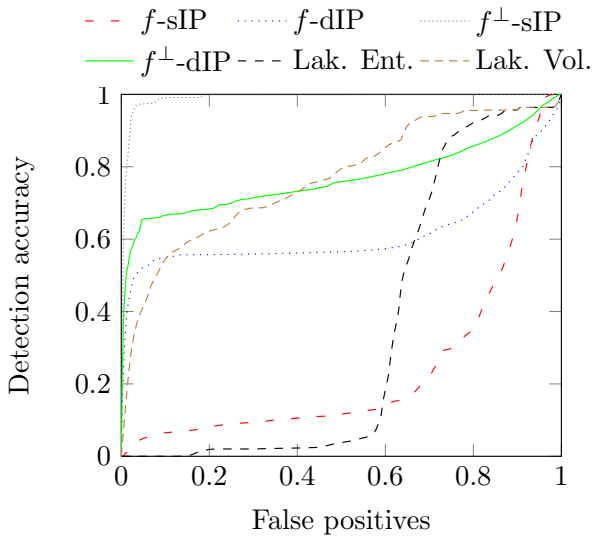
This section presents the comparison of our model(s) equipped with anomaly measures f and $f^{\perp}$ (5), and with aggregation over source and destination IP addresses (4 detectors in total) to the entropy and volume version of Lakhina's detector adapted to the scenario of the interest (see Section II-B).

The comparison is made on the traffic captured at the university network during the first week of December, 2010. The dataset continuously spans 6 days and 14 hours. It contains approximately 42 millions of flows with 19000 IP addresses. An experienced network operator has identified 10% of flows as malicious and 11% of flows as benign. Since the university network is wild, almost all the time under attacks, we consider this dataset as a good testbed.

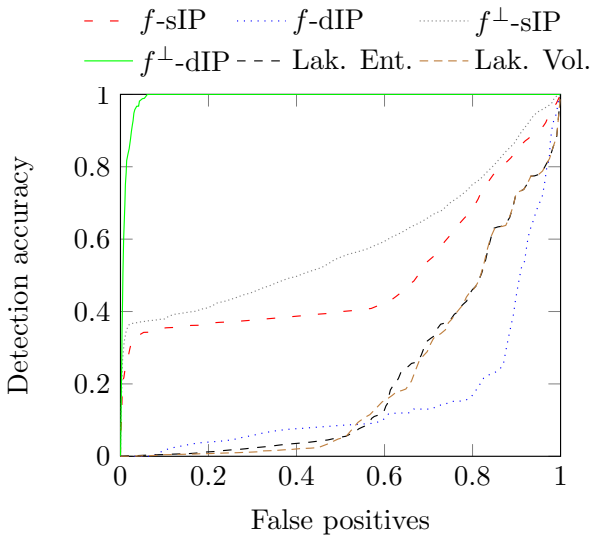
The performance is compared by receiver operating characteristics (ROCs) and by modified area under ROC curve (AUC). The modification consists from calculating it only in the interval of false positive rate $[0, 0.01]$, and normalizing it such that its value is 1.0 for the perfect detection. This limit on false positive rate up to 0.01 is used to highlight the quality of the detection on low false positive rates, which is important for practice.

A. Experimental results

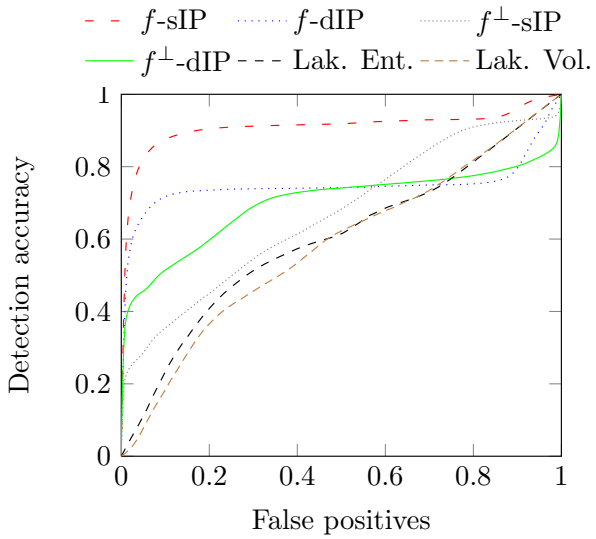
Table II shows values of modified AUCs of the compared detectors on different types of suspicious traffic. Notice, that no single detector is the best in detecting all kinds of anomalies. This is not surprising, since detectors *utilize different models* of the network and consequently they detect *different anomalies*. We can see that the presented family of four detectors can detect almost all kinds of suspicious traffic better than the adapted prior art [8],



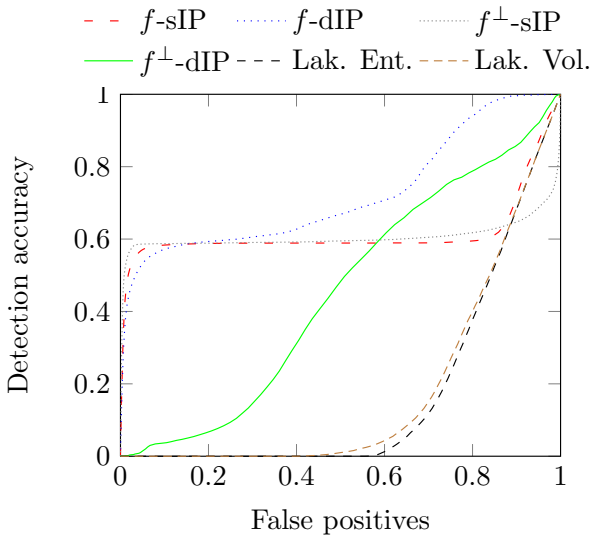
(a) Horizontal scan



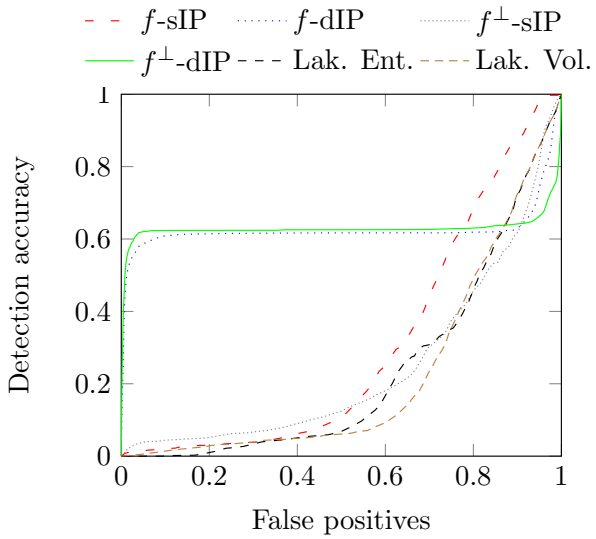
(b) Responses to horizontal scan



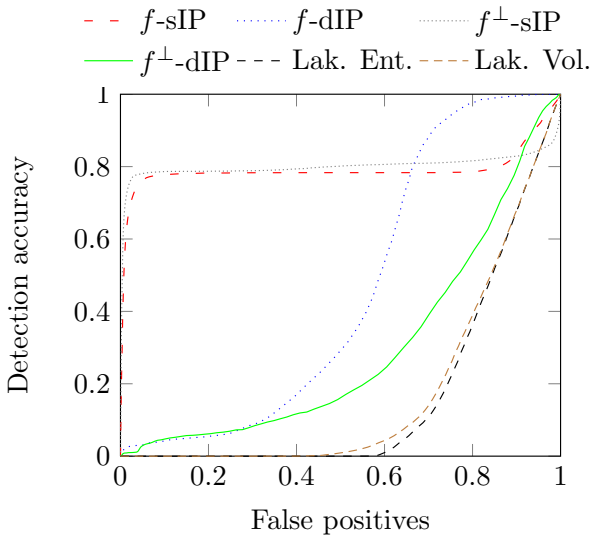
(c) P2P



(d) Responses to P2P



(e) Skype supernode



(f) Responses of skyper supernode

Figure 1: ROC curves on selected types of malicious traffic.

	f-sIP	f-dIP	$f^{\perp}$-sIP	$f^{\perp}$-dIP	Lak. Ent.	Lak. Vol.
anomalous FTP download	0.00	0.00	**0.58**	**0.57**	0.00	0.48
horizontal scan request	0.00	0.22	**0.52**	0.32	0.00	0.07
horizontal scan response	0.16	0.00	0.22	**0.41**	0.00	0.00
scan sql	0.00	0.00	0.41	0.30	0.00	**1.00**
ssh cracking request	0.02	0.01	**0.05**	0.02	0.00	0.00
ssh cracking response	0.00	0.00	0.00	**0.02**	0.00	0.00
vertical scan	0.00	0.72	**0.79**	0.77	0.00	0.72
p2p request	**0.33**	0.26	0.15	0.25	0.01	0.00
p2p response	0.25	0.21	**0.36**	0.00	0.00	0.00
p2p like request	0.01	0.37	0.07	**0.49**	0.00	0.00
p2p like response	0.25	0.11	**0.34**	0.15	0.00	0.00
skype supernode request	0.00	0.28	0.00	**0.36**	0.00	0.00
skype supernode response	0.34	0.00	**0.48**	0.00	0.00	0.00

Table II: The area under ROC curve in the interval $[0, 0.01]$ normalized such that perfect detection has the AUC equal to one. Higher value is better and the best performance is bold faced.

except the sql scan. The only case, where all detectors have failed is the ssh cracking.

We highlight that p2p and skype traffic is to some extent detected by our detectors, whereas the prior art has failed completely. Even though the detection is not perfect, it is important as it can be used as a seed in graph based detectors to identify other connected users [3].

It is also interesting to observe, how detectors with different aggregations complement each other. Detectors aggregating over source IPs detect anomalous outgoing traffic (they identify attackers), while detectors aggregating over destination IPs detects the victims. These results suggest that the improvement can be gained by clever fusion of the outputs, which is postponed to the future work.

V. Conclusion

This paper presented a family of real-time detectors identifying suspicious users of corporate networks. The detectors acquire the statistics of users traffic at one point, which is usually the connection point to the internet. The detectors uses principal component transformation to model dependencies between entropies of distributions of ports and IP addresses. By using different aggregations (source and destination IPs) and different anomaly measures, four different anomaly detectors were created, each detecting different type of attacks or anomalies.

The detectors were experimentally compared to adapted prior art on one week long capture of traffic on university network labeled by an experienced operator. The experiments showed that our methods are most of the time better that the prior art, as they well detect network scans, anomalous downloads, p2p traffic, and skype supernodes.

The important feature of our detectors is their low complexity, which makes them well suited for real-time intrusion detection systems based on an ensemble of simple detectors. Because of this, we intentionally did not deal with problems of setting of anomaly thresholds and the problem of fusing the outputs of detectors. We leave these important problems to our future work, as we believe that both problems have to be approached generally with as less assumptions on detectors, as possible.

References

[1] Daniela Brauckhoff, Kave Salamatian, and Martin May. Applying pca for traffic anomaly detection: Problems and solutions. In *Proceedings of IEEE INFOCOM 2009*, 2009.

[2] Cisco Systems. Cisco IOS NetFlow. http://www.cisco.com/go/netflow, 2007.

[3] Baris Coskun, Sven Dietrich, and Nasir Memon. Friends of an enemy: identifying local members of peer-to-peer botnets using mutual contacts. In *Proceedings of the 26th Annual Computer Security Applications Conference*, ACSAC '10, pages 131–140, New York, NY, USA, 2010. ACM.

[4] J. E. Jackson and G. S. Mudholkar. Control procedures for residuals associated with principal component analysis. *Technometrics*, 21(3):341–349, 1979.

[5] Anukool Lakhina, Mark Crovella, and Christophe Diot. Characterization of Network-Wide Anomalies in Traffic Flows. In *ACM SIGCOMM conference on Internet measurement IMC '04*, pages 201–206, New York, NY, USA, 2004. ACM Press.

[6] Anukool Lakhina, Mark Crovella, and Christophe Diot. Mining Anomalies using Traffic Feature Distributions. In *ACM SIGCOMM, Philadelphia, PA, August 2005*, pages 217–228, New York, NY, USA, 2005. ACM Press.

[7] M. Rehák, M. Pěchouček, M. Grill, J. Stiborek, K. Bartoš, and P. Čeleda. Adaptive multiagent system for network traffic monitoring. *IEEE Intelligent Systems*, 24(3):16–25, 2009.

[8] Martin Rehak, Michal Pechoucek, Karel Bartos, Martin Grill, and Pavel Celeda. Network intrusion detection by means of community of trusting agents. In *IEEE/WIC/ACM International Conference on Intelligent Agent Technology (IAT 2007 Main Conference Proceedings) (IAT'07)*, Los Alamitos, CA, USA, 2007. IEEE Computer Society.

[9] Haakon Ringberg, Augustin Soule, Jennifer Rexford, and Christophe Diot. Sensitivity of pca for traffic anomaly detection. *SIGMETRICS Perform. Eval. Rev.*, 35:109–120, June 2007.

[10] Benjamin I.P. Rubinstein, Blaine Nelson, Ling Huang, Anthony D. Joseph, Shing-hon Lau, Satish Rao, Nina Taft, and J. D. Tygar. ANTIDOTE: understanding and defending against poisoning of anomaly detectors. *Internet Measurement Conference*, pages 1–14, 2009.

[11] M. Shyu, S. Chen, K. Sarinnapakorn, and L. Chang. A novel anomaly detection scheme based on principal component classifier. In *in Proceedings of the IEEE Foundations and New Directions of Data Mining Workshop, in conjunction with the Third IEEE International Conference on Data Mining (ICDM'03*, pages 172–179, 2003.

[12] F. Silveira, C. Diot, N. Taft, and R. Govindan. Detecting traffic anomalies using an equilibrium property. In *Proceedings of the ACM SIGMETRICS international conference on Measurement and modeling of computer systems*, SIGMETRICS '10, pages 377–378, 2010.

[13] Y. Zhang, Z. Ge, A. Greenberg, and M. Roughan. Network anomography. In *Proceedings of the 5th ACM SIGCOMM conference on Internet Measurement*, IMC '05, page 30, Berkeley, CA, USA, 2005. USENIX Association.