Available online at www.sciencedirect.com

SciVerse ScienceDirect

Procedia Computer Science 19 (2013) 984 – 991

Procedia Computer Science

The 3rd International Symposium on Frontiers in Ambient and Mobile Systems (FAMS-2013)

Stability Visualizations as a Low-Complexity Descriptor of Network Host Behaviour

Matt MacDonald[a], Carrie Gates[b], Teryl Taylor[c], Diana Paterson[a], Stephen Brooks[a]

[a]*Dalhousie University, Halifax, NS, Canada*
[b]*CA Labs, New York, NY, USA*
[c]*University of North Carolina, Chapel Hill, NC, USA*

Abstract

Detecting anomalous or malicious behaviour from NetFlow data alone is a difficult task due mainly to the limited information available in a NetFlow record. In this paper we propose a "stability" metric based on only four elements of the NetFlow record (source address, destination address, port, time), which may be efficiently visualized. We show that despite not having access to packet payloads, visualizations of this stability metric display clear patterns in the case of certain anomalous network events. (scanning behaviour, peer-to-peer behaviour, etc.) We propose that these visualizations may be useful to the network analyst in detecting malicious behaviour or other deviations from typical network behaviour.

Keywords: network stability, net flows, visualization, security, flovis

1. Introduction

Several methods have been developed for detecting attacks based on an analysis of network traffic, and applications have been developed to assist system administrators in this task. Snort [1] and Bro [2] are two commonly-used and freely-available detectors; however, they both require access to full packets in order to perform a complete analysis of network traffic.

In many cases, full packets are not available for analysis, often due to privacy considerations. Additionally, even when full packets are available, traffic can be encrypted, thus requiring security detectors to restrict their analysis to packet header information (e.g., source and destination IP addresses, port information, etc.). Thus much security analysis has been done using packet header or even just flow data, rather than full packets.

The disadvantage of using summarized information such as network flow data rather than full packets is that much context is lost, and so one must often infer malicious activity from anomalous behaviour, rather than confirm via an analysis of actual traffic. We postulate that another approach to detecting malicious activity can be based on a social networking analysis of IP addresses. That is, we hypothesize that, similar to people, stable IP addresses exhibit a reasonably stable set of IP addresses to which they connect. We further hypothesize that deviations from this stability are indications of potential attacks or compromises.

doi:10.1016/j.procs.2013.06.137

In this paper we present an analysis of network traffic with the above hypotheses in mind. We further demonstrate that this approach can detect attacks that were not detected through other means. These results are presented via a visualization which provides for historical context.

1.1. Hypotheses

We address the following hypotheses:

Hypothesis I: That IP addresses exhibit a stable set of hosts with whom they connect.

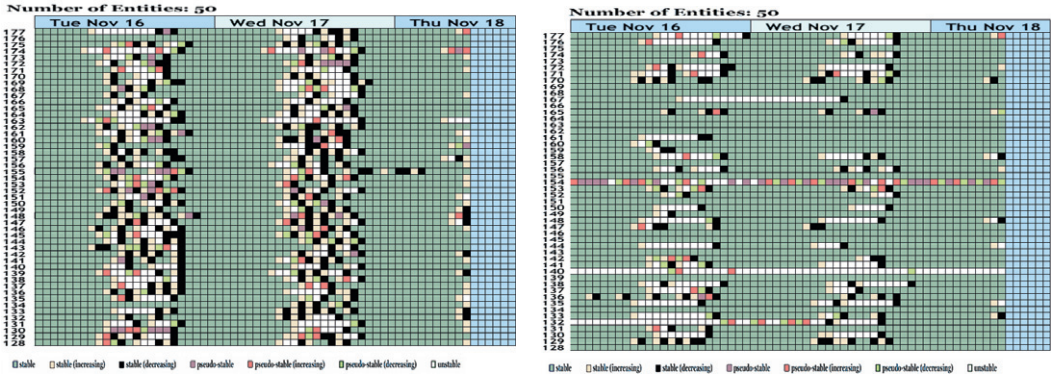
This hypothesis as stated is vague. In reality, we expect that there will be conditions under which this holds true, and conditions in which it does not. Such conditions might be based on such characteristics as the protocol in use (e.g., web-related versus non-web traffic), wireless versus wired networks, etc. We perform some preliminary analysis, presented in Section 4, to demonstrate that there exists conditions under which this hypothesis does hold; however, a full analysis of all the possible variables for this is outside the scope of this paper.

Hypothesis II: That malicious activity can be detected by observing changes in the stability of the identified connection sets.

Once a ground state of stable behaviour can be calculated, analysis of deviations from this will produce a set of events which exhibit some instability or in other words anomalous events. This hypothesis states that malicious events can be characterized as anomalous, and therefore may be discovered by analysis of the set of anomalous events.

1.2. Method Overview

Before proceeding, we must refine our loose definition of set-of-hosts used in the abstract statement of our hypothesis. From this point on, we refer to the connection set as all hosts which interact with our host of interest. If not specified explicitly, this set is assumed to include all connections from the host and to the host. To each connection set, there is an associated interval of time over which the connection set is collected. Once each host of interest has an associated connection set, an analysis of how this connection set changes over time is performed to produce a measure of stability. The change in stability for that host can then be categorized and visualized as will be discussed below. In Section 2, we introduce the method of Connection Set Differentials (CSDs), which from our connection sets will produce a stability metric.



(a) Data Set 1, Wireless network with $\mu = 3$, $\Delta t = 1$ hour (b) Data Set 1, Wired network with $\mu = 3$, $\Delta t = 1$ hour

Fig. 1. Stability Graphs generated from Data Set 1

2. Approach

2.1. Definitions

To describe the method of CSD's formally we must introduce some notation before demonstrating the machinery of the categorization. Throughout the description we will reference the graphs in Figure 1 (a) and (b) as an aid to understanding. First, we define the sets of Active Hosts.

Sets of **Active Hosts**: $H_i \subseteq \mathbb{Z}_{2^{32}-1}$ (IPv4) or $H_i \subseteq \mathbb{Z}_{2^{128}-1}$ (IPv6)

These are our hosts of interest, they will ultimately be the rows in the CSD graph. Each indexed set may be a single host (H_1 = {192.168.1.100}, H_2 = {192.168.1.101}, etc.) or sets of hosts (H_1 = {172.1.1.100, ..., 255}, H_2 = {172.1.2.100, ..., 255}, etc.). Note that there need not be any sequential relationship between successive indexed sets of hosts, meaning that the rows in the CSD graph may be fully independent of one another.

Roll-up function $r(s, e, H)$ returns a set of hosts which interacted with the host set H over the time interval: [s,e), where s is the start time and e is the end time. The set of hosts which r returns could be defined to include all hosts to which any host in H initiated a connection to during the interval, all hosts which initiated a connection to any host in H during the interval, both of these and which could be limited to http connections, etc.

To simplify the notation, the **timestep constant** Δt defines the elapsed time between indexed times t_j and t_{j+1}.

Set of **Current Active Connections** rolled-up over the interval $(t_{j-1}, t_j]$:

$$A^i_j = r(t_{j-1}, t_j, H_i)$$

This is the set of all hosts returned by the r function over the last timestep Δt before the current t_i for the hosts in H_i. For example, this may be all of the destination hosts for all outgoing connections from any host in H_i over the previous hour if Δt = 1 hour.

Set of **Prior Active Connections** rolled-up over the interval $(t_{j-2}, t_{j-1}]$:

$$B^i_j = r(t_{j-2}, t_{j-1}, H_i)$$

This is the set of all hosts returned by the r function over the timestep Δt before the time t_{j-1} for the hosts in H_i. For example, this may be all of the destination hosts for all outgoing connections from any host in H_i over the hour before last. (Δt = 1 hour)

The **Categorization Function** c is defined as:

$$c(A, B, \mu) = \begin{cases} 1 & : A = B \\ 2 & : B \subsetneq A \\ 3 & : A \subsetneq B \\ 4 & : A \simeq B \\ 5 & : B \sqsubset A \\ 6 & : A \sqsubset B \\ 7 & : A \nsim B \end{cases}$$

where = is set equivalence, $\subsetneq$ is proper subset, $\simeq$ is pseudo-equivalence, $\sqsubset$ is pseudo-subset, and $\nsim$ indicates no relation (or the trivially true relation). Pseudo-equivalence $\simeq$ and pseudo-subset $\sqsubset$ are defined to be relations constrained by a tolerance constant μ. This value specifies the number of IP addresses which

are allowed to be "leftover" after the set difference operation is performed, and yet still have the difference categorized as pseudo-stable. In other words, μ specifies the error tolerance when comparing connection sets where higher values mean a higher tolerance and lower values mean a higher sensitivity to changes in the connection set. Note that the stable category can be considered a pseudo-stable category with a μ value of 0. An example of categorization is shown on some example sets below:

A = {1,5,3,7,8,9,10,15}, B = {1,7,8,14,19}, C = {10,7,8,14}

All of the following relations are true:

$B \neq C$
$B \simeq C$ (with $\mu \geq 2$)
$C \not\subseteq A$
$C \sqsubset A$ (with $\mu \geq 1$)
$B \sqsubset A$ (with $\mu \geq 2$)
$A \nsim B \nsim C$

Below we explicitly define the implementation of the **Categorization Function** c:

$$c(A, B, \mu) = \begin{cases} 1 & : if\ |A - B| = 0,\ |B - A| = 0 \\ 2 & : if\ |A - B| = 0,\ |B - A| > 0 \\ 3 & : if\ |A - B| > 0,\ |B - A| = 0 \\ 4 & : if\ |A - B| \leq \mu,\ |B - A| \leq \mu \\ 5 & : if\ |A - B| \leq \mu,\ |B - A| > \mu \\ 6 & : if\ |A - B| > \mu,\ |B - A| \leq \mu \\ 7 & : A \nsim B \end{cases}$$

Intuitively, these categories can be thought of as:

1. **stable**, 2. **stable** ***and decreasing***, 3. **stable** ***and increasing***, 4. **pseudo-stable**
5. **pseudo-stable** ***and decreasing***, 6. **pseudo-stable** ***and increasing***, 7. **unstable**

As these categories are not mutually exclusive, the relations are tested in order from the most constrained (stable) to the least constrained (unstable) with the first evaluation to true being the category chosen.
By defining the constants Δt, μ and the indexed sets of hosts H_i we can apply the r and c functions over some time interval. The resulting category information can be inserted into a **Categorization Matrix** M by using the host set index i and timestep index j as row and column indices respectively.

Each element of the categorization matrix M is defined as:

$$m_{i,j} = c(A^i_j, B^i_j, \mu)$$

An example categorization matrix would be (with hosts indexed as rows, time as columns):

$$\begin{bmatrix} 1 & 7 & 3 & 4 & ... & 2 \\ 5 & 7 & 7 & 1 & ... & 7 \\ 1 & 2 & 4 & 3 & ... & 3 \\ ... & ... & ... & ... & ... & 4 \\ 1 & 2 & 7 & 5 & 1 & 2 \end{bmatrix}$$

If we assign a colour to each of these category values then we can create a two-dimensional graph of this simple model of connection differentials with the Activity Viewer as seen in Figure 1 (a) and (b).

The Activity Viewer [3] data format supports a row-oriented organization of the input data which is well-suited to input the categorization matrix.

2.2. Determining μ

Selecting inappropriate μ values will result in visualizations which convey little insight. For example, a μ value which is too small will result in visualizations displaying entirely unstable behaviour. Alternatively, a μ which is too large results in visualizations displaying entirely stable/pseudo-stable behaviour. We have determined that low values of the tolerance constant μ, between 3 and 5 inclusive, appear to produce the most meaningful results when using a timestep value of 1 hour. However it is clear that the choice of this constant value is dependent on network activity and, in the case of an aggregate analysis, the number of hosts contained in each host set of interest. Therefore, to make our method more robust the tolerance constant may be tuned based on properties of the data-set. [1]

2.3. Visualization

The Connection Stability Graph is an extension of the work done by Paterson which is described as a general Activity Plot [3]:

In the activity plot, the activities of individual hosts are plotted against time in a simple two dimensional grid. For the time periods in which activity occurs, for a given host, the corresponding square is filled with a color that indicates the nature of the activity. The absence of activity is represented by an empty square. The nature of the represented activity is arbitrary. It can be host relative activity allowing the temporal variability of each host to be placed in a historical perspective for that host. It can be time relative activity in which the activity of each host for a given interval is compared to the activities of its peers.

3. Experiment

3.1. Data Set 1

Data set 1 was collected [2] from a network set up for one week for a large computing-related conference. A /17 address block was assigned to the conference, however only approximately 25% of these (8135 addresses) were active throughout the week. The network was subnetted into multiple /24 and smaller subnets, the majority of which were wired networks using DHCP with one wireless network also in place. There were several ingress/egress points for the network; however, we only monitored one point — the "commodity" link. All traffic to or from commercial sites (e.g., sites other than those using high-speed research networks such as NLR) traversed this point; however, we note that assymetric routing was in place and could affect some of the observed traffic. IP addresses that were identified as either attacking the network or performing particularly egregious scanning were black-holed at the router. Thus we do not have a complete picture of all of the traffic to/from this network. We did, however, have full packet information available to us in order to confirm the ground truth of any traffic suspected as being malicious. We used traffic collected over a 94 hour period, consisting of approximately 76.2 million network flows and 1.8 trillion bytes.

3.2. Hypothesis I Methodology

To test this hypothesis we must provide evidence that the application of the stability metric to sets of network data produces a coherent digest of this data. That is to say, intuitive notions of host stability are expressed by the metric as stable when the activity of the network host is regular, and unstable when there are deviations from regular host behaviour. The most basic test of coherency would be demonstrating that the stability metric shows stable behaviour when applied to a host which is experiencing no traffic. Secondly, we should also be able to show that when the host is communicating with the same set of hosts over some time period in a regular manner, that stability is also observed. Thirdly, we should be able to show that when the host traffic is made up of a varying set of hosts over time, then the metric should be unstable.

[1] In a longer technical report available on http://flovis.net, we present a method of tuning the μ parameter based on properties of the network data.

[2] In this short paper version, the analysis of Data Set 2 is not presented.

3.3. Hypothesis II Methodology

Having established that the stability metric produces a reasonable digest of the network data, the first step in providing evidence of this second hypothesis being true is showing that significant patterns emerge within wider-scope examinations of the network data. In other words, we must be able to use the stability metric time-series to identify more complex behaviour within the network data. Once we have established that the metric is capable of describing more complex behaviour, we must then provide evidence of an ability to discern malicious behaviour from strictly anomalous behaviour.

4. Results

4.1. Hypothesis I

First, to examine evidence of the coherency of the stability metric, a set of digests was created from Data Set 1 in the manner specified in Section 2.1. These digests were then visualized by assigning a colour to each stability category as also specified previously. A representative visualization of 50 hours of network data for 50 network hosts is shown in Figure 1 (a). The first feature which is apparent in the image is that changes in stability are only occurring during the period of approximately 8 hours into the start of the day, to within 3 hours of the end of the day. This matches the time period during which the conference network would be active, as over these days conference attendants would only be present in the early morning with network staff remaining present until the late evening. Therefore, this evidence appears to show that inactive hosts remain stable.

When viewing this visualization, it is important to recall that all categories displayed as colours other than white are considered to be stable. With a pseudo-stability tolerance of $\mu = 3$, and $\Delta t = 1$ hour, a host would be categorized as unstable if it connects to more than 3 new hosts in a given hour, that it has not connected to in the previous hour. Meaning, this analysis is very sensitive to changes in stability. Despite this, it is clear a large proportion of the categorizations are non-red indicating stable network behaviour. A closer examination of the raw flow data over this time period indicates regular network behaviour made up of almost exclusively web activity occurring for the host indices and time period shown. This demonstrates that web activity is usually categorized as stable. Therefore, this evidence appears to show that regular active hosts behave in a stable manner as we would expect.

Despite network activity being generally stable, there are apparent periods of time where there is unstable behaviour occurring in the visualization. What is interesting is that this unstable behaviour is occurring around roughly the noon hour on each day. To investigate this trend, the raw flow data was examined near this hour. What we discovered was that this was the peak time for network activity during the conference. This dataset being pulled from a wireless network, shows rapid changes in the connection sets during this period. An examination of the wireless MACs used to request an address via DHCP show changes often during this period. That said, the unstable period is an expression of completely new connection sets being abruptly introduced as a new wireless client is introduced on a recently used IP address. It is reasonable to consider the changing of source machines as unstable behaviour, and this is reflected properly in the visualization of the stability metric in this case. We believe this is evidence showing that the connection set is stable in many circumstances and that analysis of how this stability evolves over time may provide some insight into network behaviour.

4.2. Hypothesis II

Figure 1 (b) demonstrates a visualization generated with the same parameters as above, $\mu = 3$, and $\Delta t = 1$ hour, but applied to a non-public wired network. This is important as the majority of hosts examined in this data set correspond to the same machine and network user over the time span of the analysis. What is interesting in this visualization are the long streaks of unstable behaviour. After examinations of raw flow data, these long streaks actually correspond to peer-to-peer traffic, and in the case of host index 140, BitTorrent traffic. Accepting that peer-to-peer traffic could be considered anomalous in this context, this observation confirms that at the very least the connection stability metric enables the user to identify certain forms of anomalous behaviour from general network behaviour.

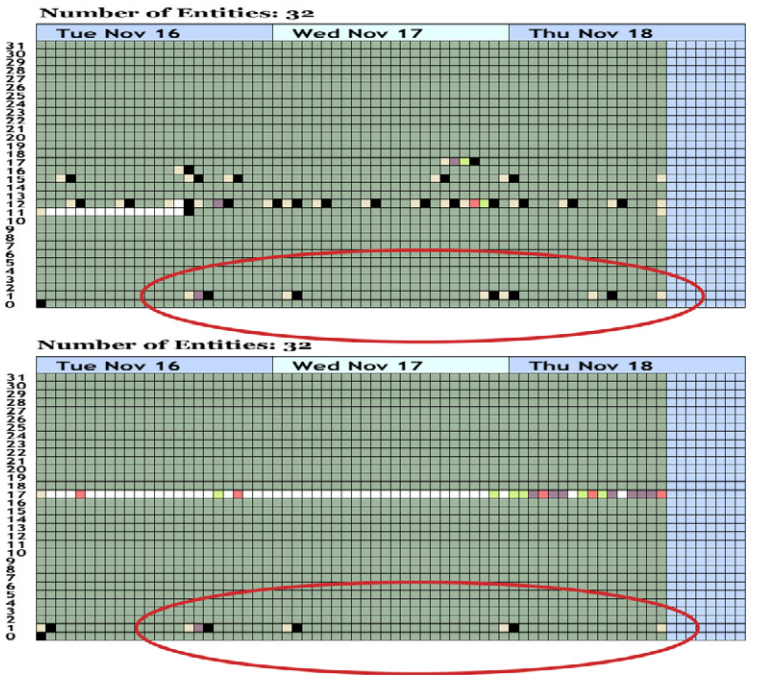


Fig. 2. Data Set 1, Wired network with $\mu = 3$, $\Delta t = 1$ hour

While examining visualizations generated at an aggregate resolution of /24, a visual pattern emerged on a number of subnets which prompted further investigation at host resolution as shown in Figure 3. To investigate, the relevant raw flow data was examined. The destination port of these connections was determined to be TCP 22, indicative of an SSH connection, but the source addresses for the connections were wildly varying. Further inspection of this pattern led us to speculate that this was a distributed SSH attack which was iterating over the internal network by attacking the start of every /16 CIDR block, aligned at 1. This is an interesting result as although we were unable to determine the exact nature of the anomaly from the connection stability metric, we were able to observe that in the visualization a pattern was occurring across unrelated hosts. At the very least this demonstrates that some forms of malicious behaviour are discoverable by visualizing the stability metric.

5. Limitations

5.1. Scaling

One limitation of this method becomes clear when we examine it in terms of scalability. That is both how the visualization scales, and the calculation of the connection sets necessary to create the visualization. The current builds of the visualization plug-in are restricted to approximately 150 hosts displayed simultaneously. However, this is an artificial restriction imposed by the scroll-less window used in the prototype, with larger visualizations available via a small update to this design. Additionally, the analyst is able to scale up the analysis by utilizing the plug-in feature of examining the "aggregate stability" of subnets or other groups of hosts as single entities. A second scalability issue is due to the computational difficulty involved in calculating the connection set for each host under examination for each relevant time-step. Naively, this operation is at least proportional to the number of flow records recorded during that time-step, for each host. [3]

[3] A more efficient implementation within the SiLK [4] framework will be made available on http://flovis.net.

5.2. Dynamic Hosts

A second limitation is analyzing flow records without knowledge of the Dynamic Host Pool. That is, we make the assumption that we are able to track the network behaviour of a specific machine over some period of time. If we are analyzing hosts on a network where the DHCP lease time is very short, then it is possible we would lose track of a machine after a simple reboot or stand-by operation when a new IP is assigned. In our analysis of the first dataset we addressed this problem by logging the DHCP lease table over the duration of our analysis.

6. Comparison to Related Work

The recent work by Collins et al in [5] bears some resemblance to our method. In their work, they are tracking the connection set for a group of addresses which exist but are not routed.

Our method was inspired by the prior work of Koukopoulos et al [6] on detecting universal stability and intrusion detection attacks by gauging changes in this stability. In terms of work focused on visualizations used to identify malicious behavior in network data, there are a significant number [7, 8, 9, 10]. As our visualization was constructed within the FloVis [3] framework, we are influenced by previous extensions of the framework [11].

7. Conclusions, Future Work

We have defined a connection set as derived from NetFlow records, constructed a method of analyzing changes in the connection set of a host, and shown that an analysis of this connection set results in a metric which can be treated as an abstract representation of the "stability" of a network host. In addition to this, we have demonstrated a method which can visualize this stability metric for a set of hosts. From this visualization, we were able to identify patterns representative of irregular network behaviour. A specific example of this irregular behaviour was identified as being malicious.

We believe we have provided evidence that the application of this simple metric to NetFlow data can produce interesting and often surprising results. We plan to expand these ideas further by creating visualizations where the user is able to manipulate the stability tolerance constant μ as well as modify the time-scale interactively, observing the results and tuning these parameters as required.

Acknowledgments

For their valued input we would like to thank Joel Glanfield and Scott Campbell.

References

[1] Snort: The open source network intrusion detection system [online] (2012). http://www.snort.org.
[2] V. Paxson, Bro: a system for detecting network intruders in real-time, Computer Networks 31 (23-24) (1999) 2435–2463.
[3] T. Taylor, et al., Flovis: Flow visualization system, Proceedings of the Conference For Homeland Security, CATCH 09 (2009) 186–198.
[4] Silk - documentation [online] (2012). http://tools.netsa.cert.org/silk/docs.html.
[5] J. Janes, M. Collins, Darkspace construction and maintenance, Proceedings of FloCon 2011.
[6] D. Koukopoulos, et al., Optimal algorithms for detecting network stability, Proceedings of WALCOM'08 (2008) 188–199.
[7] G. A. Fink, et al., Visual correlation of host processes and network traffic, VIZSEC 05: Proceedings of the IEEE Workshops on Visualization for Computer Security (2005) 11–19.
[8] Y. Hideshima, H. Koike, Starmine: A visualization system for cyber attacks, Proceedings of APVIS 2006 (2006) 131–138.
[9] K. Abdullah, et al., Ids rainstorm: Visualizing ids alarms, VIZSEC 05: Proceedings of the IEEE Workshops on Visualization for Computer Security (2005) 1–10.
[10] F. Mansmann, et al., Visual analysis of network trafc for resource planning, interactive monitoring, and interpretation of security threats, IEEE Transactions on Visualization and Computer Graphics 13 (6) (2007) 1105–1112.
[11] J. Glanfield, et al., Visualizing properties of network hierarchies within the flovis framework, Proceedings of VizSec 2009 (2009) 11–19.