Network Anomaly Confirmation, Diagnosis and Remediation

David Plonka[*] and Paul Barford[*†]
[*]University of Wisconsin - Madison
E-mail: {plonka,pb}@cs.wisc.edu
[†]Nemean Networks

Identifying and diagnosing network traffic anomalies, and rectifying their effects are standard, daily activities of network operators. While there is a large and growing literature on techniques for *detecting* network anomalies, there has been little or no treatment of what to do *after* a candidate anomaly has been identified. In this paper, we present a first step toward formalizing and automating the time-consuming and challenging tasks associated with network anomaly confirmation, diagnosis and remedy. Our work assumes that potential anomalies are identified either through visual analysis of key traffic measurements or from a Network Anomaly Detection System (NADS). We describe a flexible framework for network anomaly confirmation, diagnosis and remedy that is based on workflow concepts. The key features of this framework include data types/sources, analyses and decision points. We present an instantiation of our framework that includes a taxonomy of network traffic anomalies and detailed steps for confirmation of anomalies associated with malicious attacks. We demonstrate our framework by applying it to traffic in our university network. We propose that our framework is a starting point for streamlining operational tasks associated with traffic anomalies, and for the generation of annotated data sets that can be used in future NADS development.

I. Introduction

Addressing traffic anomalies is a core task for network operators who are charged with ensuring that their infrastructures operate reliably and at specified performance levels. Traffic anomalies present many challenges starting with the fact that they are often quite difficult to identify in complex, large scale networks where non-anomalous network traffic is highly variable.

These challenges have catalyzed a large body of research over the past several years. Well known examples include [2, 1, 6, 5, 7, 8]. The primary focus of those studies is to develop analysis techniques that enable network traffic anomalies to be detected *accurately* (*i.e.,* with low false positive and low false negative rates) and in a *timely fashion* (*i.e.,* such that an alert is generated shortly after the anomaly begins). While many of the detection techniques described in prior work are excellent foundations for Network Anomaly Detection Systems (NADS) that could be deployed in operational networks, there has been virtually no prior work on what to do *after* an anomaly alert has been generated.

In this paper, we begin the process of codifying the activities that take place after a candidate traffic anomaly has been identified. The starting assumption for our work is that a network operator has received an alert from a NADS or has identified a potential anomaly through visual analysis of traffic measurements. At this point, we posit that there are three primary activities that must take place: Confirmation, Diagnosis and Remediation (CDR). *Anomaly confirmation* is the process of verifying that an anomaly is authentic. If NADS were infallible, confirmation would not be necessary. When an anomaly is confirmed, then diagnosis is required, otherwise, the alarm is false. *Anomaly diagnosis* is the process of identifying the causes and effects of the confirmed anomaly. In some cases, an anomaly may be deemed to be transient or innocuous (*e.g.,* a small flash crowd) and to have had no lasting effect on the network in which case no further steps are required. In other cases, systems in the network may be damaged or compromised and further steps that mitigate the anomaly's effects are required. *Anomaly remediation* is the process of addressing the effects of the anomaly with the goal of returning the network infrastructure to a good state.

Each step in the CDR process can be complex, time consuming and dependent on details of particular network infrastructure. As could be expected, the lack of established methods or models for CDR processes has led to the proliferation of ad hoc approaches. The combination of these issues makes the codification of CDR processes similarly complicated. The high level challenge in our work is to create a framework for these processes that is general enough to be widely applicable and readily customizable for specific networks.

We present a framework for CDR processes based on workflow concepts. Workflow is the notion of documenting a pattern of activities associated with a higher level process, and has been widely used to improve efficiency in manufacturing and other business processes. A strength of a workflow-based approach is that it lends itself directly to several types of analysis (*e.g.,* critical path and queuing) that can expose inefficiencies in operational environments. Studies of workflows have a long history. The notion of documenting workflows and then using this as the basis for modeling and improving processes is a central focus of operations research. These methods have also been applied in the information technology domain, *e.g.,* in [3].

Our framework begins with a taxonomy of anomaly types that include routing/network failures, flash crowds, malicious attacks, measurement failures, and unknown events. We then describe our workflow model for traffic anomaly CDR. The model includes data types/sources, processes/transformations and decision points that are required for each of the CDR steps associated with each anomaly type. We posit that this approach satisfies the requirement of generality since we take a top down approach for specifying both the taxonomy and details of workflow associated with handling each anomaly type. We also argue that the approach satisfies the requirement of being customizable since the workflows can easily be augmented with tools, data streams and processes associated with a given network.

While we are not aware of prior work that defines CDR processes in detail, the operator-defined taxonomy of five broad anomaly types described in [1] is a starting point for our work. That study identifies four anomaly types (flash crowd, attack, measurement and network), which were identified by network operators in a multi-month set of flow traces. This labeled data set then formed the basis for testing a new anomaly detection scheme. Similar labeled data sets could be created if a consistent confirmation process is applied.

We demonstrate the efficacy of our framework by presenting a detailed instance of the confirmation process associated with malicious attacks. This instance is based on the confirmation methods used by operations staff in our university network. We then apply this workflow model to confirm and diagnose sample candidate anomalies observed in our university network. This effort was important for refining our workflow model.

II. A Confirmation Framework

We propose a framework for network anomaly CDR organized around workflows, describe its components, and apply it to the abuse anomaly confirmation process. Application of the framework to the diagnosis and remediation processes and to a broader set of anomalies is a future objective.

Our approach to developing the framework considered the perspective of practitioners. It quickly became clear that while some initial tasks in CDR are similar for a range of anomalies, the tasks become distinct as the process evolves. Thus, we made a design decision that the starting point for the framework was to generate a taxonomy of anomalies, and then develop workflow for addressing each major anomaly type. This highlights the importance of anomaly classification in our framework, and illustrates the many degrees of freedom that must be considered and that complicate this process.

A. Anomaly Taxonomy

Figure 1 is an initial taxonomy for network anomalies, and is based on what is used operationally at the University of Wisconsin when logging anomalies detected visually in IP traffic volume time-series plots [1]. The tree is rooted at the "Anomaly" type, expands to five general anomaly types, and then to specific causes and characteristics that further define and differentiate the anomalies. Rather than being all-encompassing, this initial taxonomy is defined pragmatically; only those types of anomalies that are pertinent to a particular operational anomaly detection procedure are defined here. We believe that the best way to further refine and expand the taxonomy is through community contributions.

B. Workflow Requirements

The goal for our framework is to establish a method for expressing the required steps for network anomaly CDR that can capture both the high level aspects and the details of the processes. Workflow diagrams are a natural solution. In general, a workflow documents a process that is repeatedly applied.

Many different diagramming methods can be used to express workflow. For anomaly CDR, there are two basic object types that must be explicit in any workflow representation. The first type is a *task* that is conducted by a network operator. A set of tasks collectively make up the confirmation, diagnosis and remediation processes. A task typically requires an operator to consider data, possibly apply some kind of transformation to the data, and eventually make a judgment based on experience or policy. Tasks can lead to other tasks, can require taking an action (*e.g.* generating a trouble ticket), or can simply result in the operator having more knowledge about the anomaly.

The second basic object type is *data* that is required for tasks. There can be many data sub-types. Data may also be collected from a large number of sources and over different time scales. An individual data stream can be associated with multiple tasks.

C. Anomaly Confirmation Workflow

The confirmation process should authoritatively determine when a given anomaly has occurred. Figure 2 is an initial confirmation workflow for network abuse anomalies that employ flooding *i.e.*, it might confirm an anomaly type specified by the "Flood" node in Figure 1. The representation we chose is a directed graph with the following elements:

- *Input nodes* are rectangles and are connected from the start node. Inputs in this process are the *required measurements* for anomaly confirmation.
- *Decision nodes* (questions) are rounded rectangles with sub-elements specifying the possible result (answer).
- *Directed edges* from inputs nodes express *prerequisites*; *i.e.*, the given input is required to make the connected decision. Directed edges from decision nodes represent *transitions* that follow from decisions connecting to the subsequent decision (if any) or confirmation output.
- *Dashed elements* are portions of the process that involve decisions based on intuition and operator experience. These elements are candidates for automation, in addition to the non-dashed elements that are already automated.

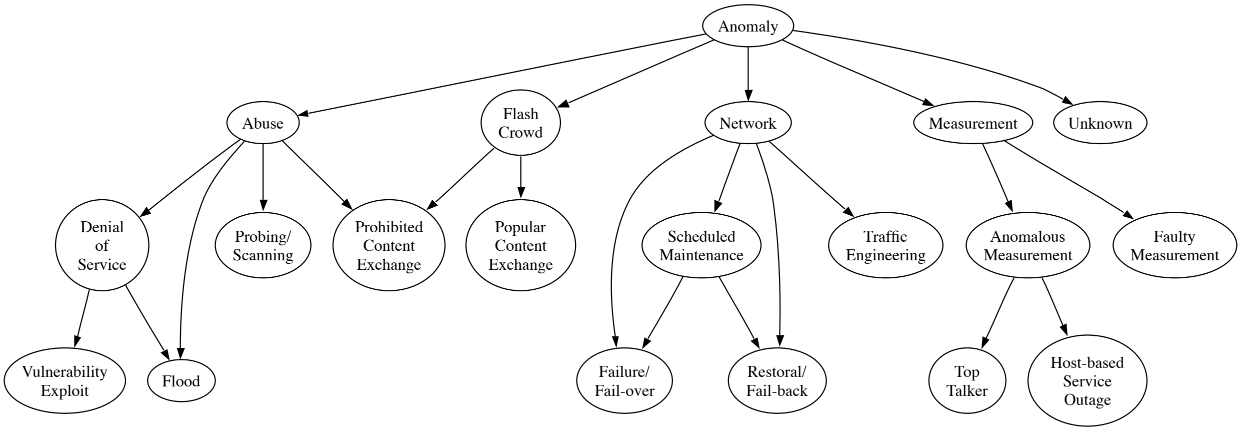

Fig. 1. A sample network anomaly taxonomy.

- *Join nodes* are ellipses and are labeled with functions such as "AND" and "OR."
- *Output nodes* are rectangles and are connected to the end node and represent the positive or negative *confirmation* for the given type of anomaly.

Note that there are multiple arcs from the start node. This indicates the presence of optional paths (based on available input measurements) and potential parallelism in the anomaly confirmation process when multiple inputs are available. Thus, this process is partially-ordered; not all the decisions are necessarily ordered.

The formal expression of a confirmation process as a graph (or perhaps some other diagrammatic reasoning representation) potentially enables automated confirmation. It also allows the possibility of automated analysis of the confirmation processes (as in partial-order planning) to arrive at alternate implementations or interleaving of confirmation processes (for simultaneous processing of multiple anomaly types) for efficiency or performance. [1]

Requisite Measurements: Figure 2, with rectangles representing input nodes, shows that this confirmation workflow contains decisions requiring three types of data: (i) bit and packet time series measurements, (ii) flow time series measurements, and (iii) flow or packet traces. The out-degree of these nodes is 2, 4, and 10, respectively. This indicates those inputs increasing utility in confirming or refuting potential anomalies, as we move from coarse (SNMP counters) to detailed measurements (flow or packet data). For instance, coarse measurements can expose inbound vs. outbound symmetry, but detailed measurements are required to discern multiplicity or out-degree of conversations or protocol-specific header values.

Note that while detailed measurements are ultimately required to reach the confirmation output nodes, coarse input measures are sufficient to begin the confirmation process. This suggests that it may be possible to use the ubiquitous, coarse measurements as leading indicators to guide the deployment, activation, or examination of more expensive, detailed measurements. Thus, the confirmation workflow mirrors what is observed in practice: *i.e.*, operators use easily digested visualizations of coarser measurements to direct their examination of detailed measurements during the forensic process of anomaly confirmation.

III. Applying the CDR Framework

In this section we demonstrate how the anomaly taxonomy and workflow framework are applied to confirm the presence of flood-

[1] One example of a system optimized for efficient automated response is described in [9].

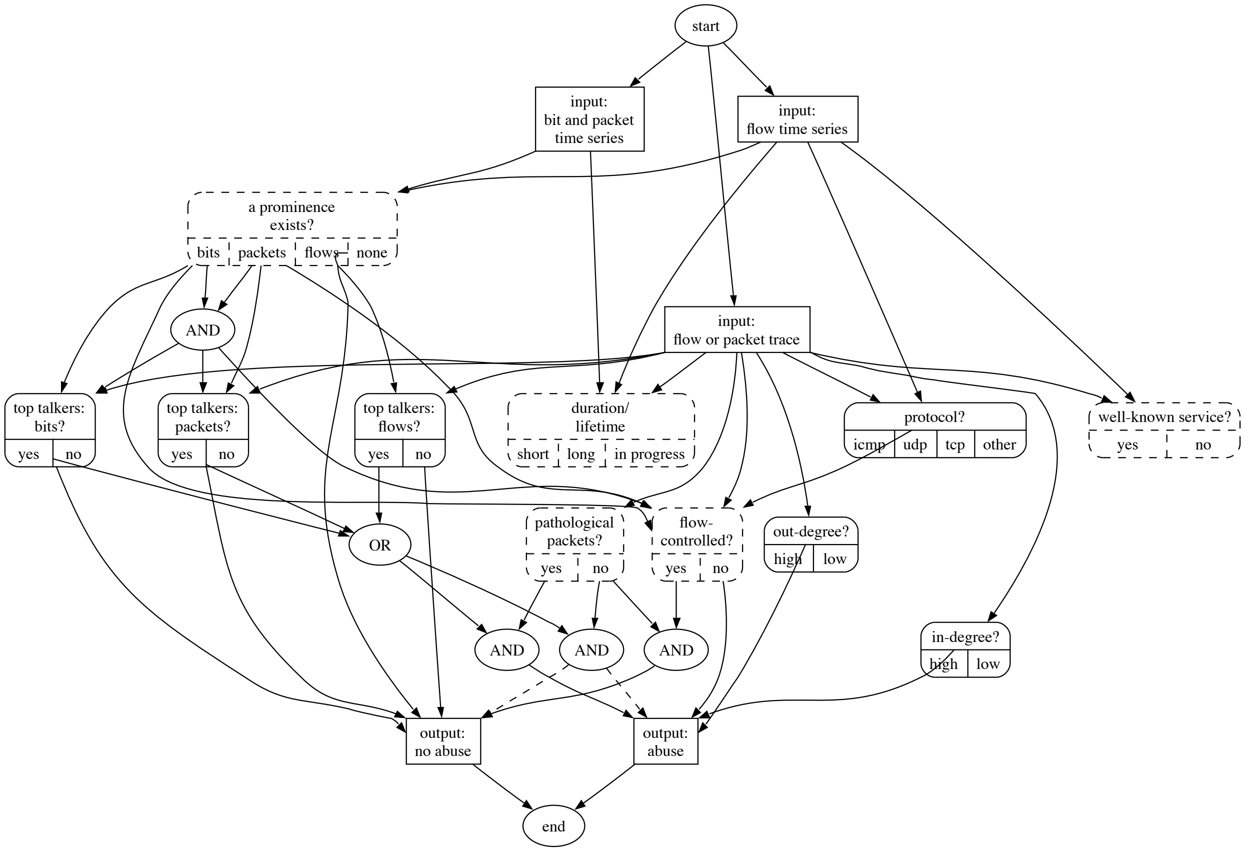

Fig. 2. A sample confirmation workflow for (outbound) flood-based network abuse anomalies.

based network abuse anomalies observed in our campus network. While we describe a single instance in order to highlight details, the framework has been used to confirm many other flood-based anomalies in our network.

Figure 3 is a time series plot of the unicast packet rate for the campus connection to its commodity Internet Service Provider. These measures are based on the MIB-II SNMP ifOutUcastPkts and ifInUcastPkts counters of a campus border router. (In this figure as well as Figures 4 through 6, outbound traffic is shown above the horizontal axis and inbound traffic below. All graphs show 48 hours, with the most recent time on the right.)

We considered the left-most midnight local time, labeled “00:00”, in the graphs and follow the confirmation workflow:

1) From the start node, we visit the “bit and packet time series” input node because we have this coarse, SNMP-based measurement available. We then consider the left-most arc to “a prominence exists?” decision node. We can see in Figure 3 that an outbound prominence exists at time 00:00; this is amidst a period of increased outbound traffic (increased by approximately 40,000 packets per second.)
2) Since we also have flow time series available, we can additionally visit the “flow time series” input node. We then consider the left-most arc to “a prominence exists?” decision node. In Figure 5 we also note an outbound UDP prominence.
3) We now follow the arc from “packets” to the “top talkers: packets?’ decision node. This decision node requires “input: flow

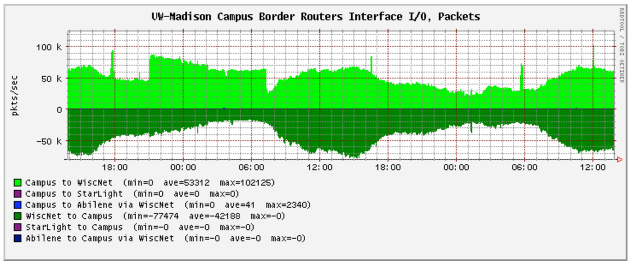

Fig. 3. SNMP time-series: **Interface packet rate.**

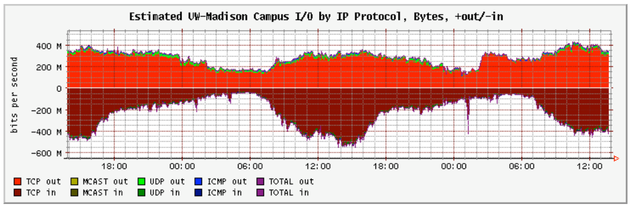

Fig. 4. Flow time-series: **Bit rate by protocol.**

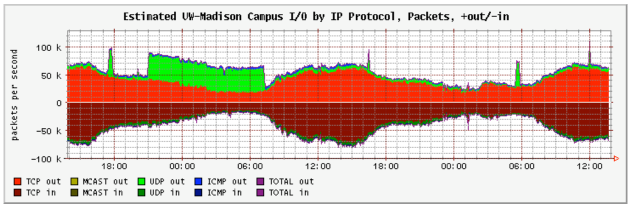

Fig. 5. Flow time-series: **Packet rate by protocol.**

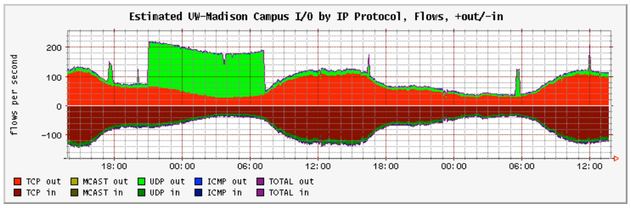

Fig. 6. Flow time-series: **Flow rate by protocol.**

or packet trace." We have a trace in the form of flow-export records in log files. We run a standard "top talker" report for the flow records including the time 00:00 and find that there is a single host with outbound packet rate commensurate with the magnitude of the prominence.

4) We now visit the "OR" join node and consider the subsequent "AND" join nodes. These follow the "pathological packets?" and "flow-controlled?" decisions nodes. By examining a trace of flow records, the operator can determine whether or not the UDP packets appear to be flow-controlled. In this case they are not.
5) We follow the arc from "flow-controlled? no" to the "abuse" output node, thus confirming that an outbound flood-based network abuse anomaly exists at time 00:00. We have now reached the "end" node and the traversal of this anomaly confirmation workflow is complete.

IV. Discussion

While our CDR framework, anomaly taxonomy and flood abuse workflow are first steps toward formalizing what takes place *after* and anomaly is detected, a number of open issues and questions remain, including:

- Framework Completeness: While our framework captures basic characteristics, it represents only a small portion of the CDR process: (i) the relationship between anomaly confirmation and diagnosis is complicated; *i.e.*, one can often confirm an anomaly without having completely diagnosed its causes and effects; (ii) we haven't specified remedies. Remedies are likely to be highly network dependent and may be numerous even for a single type of anomaly.
- Anomaly Lifetime: It is sometimes easier to classify an anomaly having observed both its inception and termination. Introducing lifetime into a workflow might include decisions based on observations at arbitrary points in the past for future.
- Anomaly Correlation: Multiple anomalies might be correlated, whether they occur in series or in parallel. For instance, an inbound probe might be followed by an inbound vulnerability exploit compromising a local host that is subsequently enslaved as the source of an outbound denial-of-service flood. Considering correlations between anomalies can aid the CDR processes.

- Anomaly Atomicity: Since anomalies can overlap in time and may only be distinguishable in fine-grained measurements, the atomicity of anomalies is important since it has implications for remediation.
- Detail and Time Scales: The level of detail and the time scale of data streams determine whether or not some anomalies can be distinguished from each other or even discerned at all.
- Probabilistic Reasoning: We represented the flood confirmation workflow as a directed graph with only simple join nodes whose output is a simple function of inputs, such as "AND" and "OR." We believe that probabilistic reasoning, such as that embodied by a Bayesian network, could be employed to enhance our basic framework.

We have explored both time scales and the use of Bayes networks in [4].

V. Conclusion

While the notion of a self-healing or autonomic network that can adjust to anomalies automatically is compelling, we are a long way from that vision. The current state-of-the-art in dealing with network traffic anomalies are automated detection systems coupled with ad hoc confirmation, diagnosis and remediation processes. In this paper, we begin the process of formalizing network traffic anomaly CDR toward the goal of increasing operational efficiency. We describe a workflow framework for CDR that begins with a taxonomy of anomaly types and an approach for specifying the data sets, process steps and decision points. We present an example confirmation model associated with flooding attacks apply it to candidate anomalies observed in our university network. We plan to expand our model set to include workflows for other anomaly types and to refine these models by interacting with other network operators.

Acknowledgments

This work was supported in part by NSF grants CNS-0347252, CNS-0716460, CNS-0831427 and CNS-0905186. Any opinions, findings, conclusions or other recommendations expressed in this material are those of the authors and do not necessarily reflect the view of the NSF.

References

[1] P. Barford, J. Kline, D. Plonka, and A. Ron. A Signal Analysis of Network Traffic Anomalies. In *Proceedings of ACM SIGCOMM Internet Measurement Workshop*, Marseilles, France, November 2002.

[2] J. Brutlag. Aberrant Behavior Detection in Time Series for Network Monitoring. In *Proceedings of the USENIX Fourteenth System Administration Conference LISA XIV*, New Orleans, LA, December 2000.

[3] E. Eide, L. Stoller, T. Stack, J. Freire, and J. Lepreau. Integrated Scientific Workflow Management for the Emulab Network Testbed. In *Proceedings of USENIX Technical Conference*, Boston, MA, June 2006.

[4] J. Kline, S. Nam, P. Barford, D. Plonka, and A. Ron. Traffic Anomaly Detection at Fine Time Scales with Bayes Nets. In *Proceedings of the IEEE Third International Conference on Internet Monitoring and Protection*, Bucharest, Romania, June/July 2008.

[5] A. Lakhina, M. Crovella, and C. Diot. Characterization of Network-wide Anomalies in Traffic Flows. In *Proceedings of ACM SIGCOMM Internet Measurement Conference '04*, Taormina, Italy, October 2004.

[6] A. Lakhina, M. Crovella, and C. Diot. Diagnosing Network-wide Traffic Anomalies. In *Proceedings of ACM SIGCOMM '04*, Portland, OR, August 2004.

[7] A. Lakhina, M. Crovella, and C. Diot. Mining Anomalies Using Traffic Feature Distributions. In *Proceedings of ACM*

SIGCOMM '05, Philadelphia, PA, August 2005.

[8] A. Soule, K. Salamatian, and N. Taft. Combining Filtering and Statistical Methods for Anomaly Detection. In *Proceedings of ACM SIGCOMM Internet Measurement Conference '05*, Brekeley, CA, October 2005.

[9] Y.-S. Wu, B. Foo, G. Modelo-Howard, S. Bagchi, and E. H. Spafford. The Search for Efficiency in Automated Intrusion Response for Distributed Applications. In *Proceedings of the 27th IEEE Symposium on Reliable and Distributed Systems*, Napoli, Italy, October 2008.