Angle: Detecting Anomalies and Emergent Behavior from Distributed Data in Near Real Time

Robert L. Grossman, Anushka Anand, Shirley Connelly,
Yunhong Gu, Matt Handley, Michal Sabala,
Rajmonda Sulo, Dave Turkington and Lee Wilkinson
National Center for Data Mining, University of Illinois at Chicago

Ian Foster, Ti Leggett, Mike Papka, Mike Wilde
University of Chicago and Argonne National Laboratory

Joe Mambretti
Northwestern University

Bob Lucas and John Tran
Information Sciences Institute, University of Southern California

Abstract

We describe the design of a system called Angle that detects emergent and anomalous behavior in distributed IP packet data. Currently, Angle sensors are collecting IP packet data at four locations, removing identifying information, and building IP-based profiles in temporal windows. These profiles are then clustered to provide high-level summary information across time and across different locations. We associate meaningful changes in these cluster models with emergent or anomalous behavior. Emergent clusters identified in this way are then used to score the collected data in near real time. The system has a visual analytics interface, which allows different emergent clusters to be visualized, selected, and used for scoring of current or historical data. Each Angle sensor is paired with a node on a distributed computing platform running the Sector middleware. Using Sector, data can be easily transported for analysis or reanalysis. Reanalysis is done using the Swift workflow system.

1 Introduction

Understanding data flows across the Internet is a challenging problem in high performance analytics. There are millions of computers connected to the Internet and billions of data flows that access them. Unfortunately not all these flows are benign, and an increasing number of them are associated with some type of anomalous behavior, such as probing for system vulnerabilities, phishing schemes, spam, components of bot networks, and related behavior. The problem is difficult in part because these behaviors change all the time. In this paper, we focus on detecting in near real time behavior that is unusual and different than has been seen before. We call this type of behavior *emergent* and give a precise definition in Section 2.

In this paper, we introduce a system called Angle for identifying emergent behavior. An advantage of this framework is that information from geographically distributed locations can be combined together easily in order to detect emergent behavior that may not be readily apparent simply by analyzing data from one location.

There were four primary challenges in this project: The first challenge was how to define emergent behavior in a meaningful way and to develop an algorithm to detect it. The second challenge was to deal with the high data volumes associated with this problem. The third was to develop a visual analytics interface that could be used effectively by analysts. The fourth was to deploy middleware that could effectively build statistical models from distributed data and to recompute profiles, clusters, and emergent clusters from large historical data sets when new features or new algorithms for detecting emergent behavior are introduced. We call this the re-analysis problem.

The majority of prior work in this area is what is usually called signature based. Signatures of specific attacks are created and IP data is screened using these signatures. Snort is one of the most widely deployed systems for analyzing IP packet data using signatures [3]. There are also a variety of statistical based techniques. See, for example, [1] and the references cited there. Angle is a statistical based system.

Angle is novel in two fundamental ways:

- It is the only system that we are aware of that is designed to detect emergent behavior in IP based data.
- It is a distributed system that is designed to work with data from multiple geographically distributed sensors and is designed in such a way that additional sources can be added without impacting the performance of the system. See Figure 1.

2 The Angle Data Analysis Methodology

2.1 Overview

Collecting and Processing Packets. The system consists of sensors, each of which is paired to a node on our distributed computing platform that is running a system we have developed called Sector as well as Globus. Sensors are currently located at the following sites: the University of Illinois at Chicago (UIC), University of Chicago (UofC), Argonne National Laboratory (ANL) and Information Sciences Institute (ISI).

Every ten minutes, in each location, the sensors collect real time IP packet data from the commodity internet, anonymize it, add metadata, and package it into a pcap file format for further processing.

Computing Profiles. The next step is that the pcap files are processed to produce profiles, which is summary data associated with a specific entity. In the examples, we compute profiles associated with Source IP addresses, but other types of profiles can be easily generated using the system. In the examples below, the profiles consist of eight features, but the Angle System can produce profiles containing more or fewer features.

Computing Models. Next each profile file is processed to determine clusters, which we view as a convenient way of summarizing the behavior at a particular location over a particular time period. Currently, clusters are computed using the k-means algorithm, but other algorithms can also be easily used.

Meta-analysis of Models. Next collections of models are analyzed to determine emergent clusters. Emergent clusters can be defined in various ways. One way is to define an emergent cluster as a new cluster that

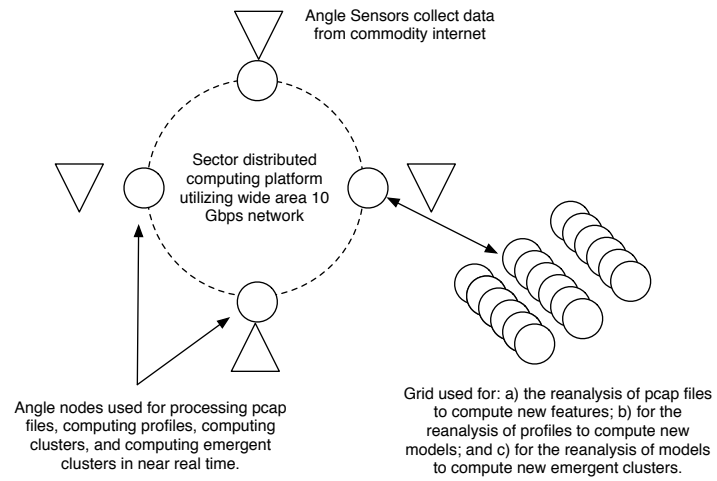

Figure 1: This is an overview of the Angle architecture. Data is collected in near real time from sensors located on the commodity internet. Each sensor is paired to a node on a distributed computing platform running Angle and Sector middleware. Data analysis is done using the Angle and Sector nodes. Reanalysis is done using a grid and the Swift workflow software.

appears when there has not been one before. Another way is to define an emergent or anomalous cluster as a cluster that is present in one site or just a few sites. We describe how we compute emergent clusters in more detail below.

Real time scoring of profiles. The Angle system allows models to be browsed, visualized and selected. Once a model is selected, emergent clusters in the model are scored using the Angle scoring functions. This process is described below.

2.2 Computing Features

Using the pcap data captured from all sensor locations at ten minute intervals, statistical summaries for each distinct IP address are extracted. Currently, the features extracted over the ten minute period are: number of ports touched by the IP, number of destination IPs touched by the IP, number of packets sent, average packet size, average data size, maximum packets per destination, maximum packets per port and maximum inter packet interval. These features are meant to expose important aspects of IP traffic but as this behavior is dynamic and complex, the application has been designed to be easily extended. The analyst can add or change features, therefore making the analysis process flexible and robust.

To allow for meaningful comparisons globally across all sensor locations, features are normalized using constants extracted from the data of all sensors.

2.3 Computing Stable and Emergent Clusters

After the feature extraction step, large volumes of data are clustered to allow for fast meta-analysis. A set of clusters for a selected combination of sensor locations and time represent a "model" in the Angle application.

The clustering can be seeded with centroids from the previous run to identify and relate similar clusters over time as well as to achieve faster computation.

Both the features and cluster profiles are stored in a database as they are computed every ten minutes. This gives Angle the option to build or compute models from any time and sensor location over the past year.

Models are analyzed to detect emergent behavior. As there is no well-established definition of emergent behavior, our approach is to analyze stable clusters over time and use the emergence of a new cluster as an indication of a new behavior in the network data. The multivariate F-statistic is used to determine if there is

a significant difference between two models (sets of clusters). If there is no difference the clusters are stable over time, otherwise they are emergent.

2.4 The Angle Score

In this section, we describe how the Angle score is computed. First, we describe how Angle models are computed. Then, we describe how an Angle model is used for scoring.

1. Given an event, we retrieve the corresponding feature vector and update it using the event data.
2. We first choose a model. A model contains zero or more emergent or red clusters. We let μ_k denote the center of each red cluster and σ_k denote the variance of each red cluster in the model.
3. Each model has weights θ_k and λ_k associated to it. Here the weights θ_k sum to 1, while the weights λ_k control the influence of the particular cluster in the score.
4. Given the updated feature vector x, we assign a score ρ to it using the emergent or red clusters using the formulas:

$$\rho(x) = \max_{k \in R} \rho_k(x), \qquad \rho_k(x) = \theta_k \exp\left(\frac{-\lambda_k^2 ||x - \mu_K||^2}{2\sigma_k^2}\right)$$

2.5 Scale Up

All the components of Angle are implemented to allow for real-time computation and analysis. The first reduction in data is the result of working with fixed length feature-based profiles rather than event-based IP packets. The second reduction in data is the result of working with clusters of profiles instead of the time varying profiles themselves.

As currently designed, nearly every component of the analysis pipeline is linear in the size of its input. The only step that does not have linear scale up is the finding of emergent behavior, where Angle compares two sets of clusters. However, since the number of clusters at any time is relatively small compared to the size of the input data, the computational time still allows for near real time analysis.

3 Data Set

We have been collecting network packets at the following four locations using a sensor that we have developed: University of Illinois at Chicago, University of Chicago, Argonne National Laboratory and the ISI/University of Southern California.

Network data is captured by independently managed network monitoring servers running IP packet capture software we have developed. Typically these are fast Opteron or Xeon servers monitoring a port-mirror of an output port of a switch or router on the edge of a network.

Angle capture software was designed to preserve privacy while capturing sufficient packet information to allow behavioral data mining. Source and destination IP addresses are hashed using a randomly generated salt, which is changed automatically by the software every time it is restarted or when the previous salt is one week old. Payload checksum is computed and stored, and the payload itself is nulled. MAC address fields along with checksum are nulled. Geo-location information is looked up based on IP addresses prior to their hashing and includes country, state, city and zip code (as available). The captured data is stored in a standard pcap format [2] to allow processing with standard tools and at no time are non-anonymized packets stored on disk. Furthermore, salts are non-recoverable.

The software is provided in source code form and is started and monitored by staff at each of the edge nodes. Uploads of pcap files are handled automatically by a robust upload tool that manages an upload queue on disk, spools files and uploads them to Angle System. The upload tool will upload pending queue files in-order even after system reboot. Currently a compressed pcap file is sent by each monitoring location every ten minutes.

The pcap queuing server receives pcap files and queues them for processing by Angle components that compute profiles and cluster models. The results are stored in files and are also tracked in a SQL database.

Sensors	
number of sensor locations	4 distributed locations
Events	
daily event data - events	97 million packets
daily event data - files	576 pcap files
daily event data - GB	7.68 GB pcap data
event data by SC 07	3.5 TB pcap data
event data by SC 07	43.7 billion events
Profiles	
daily profile data - profiles	128,300
daily profile files - files	576 profile files
daily profile data - GB	25 MB
profile data by SC 07	77,000 profile files
profile data by SC 07	2.6 GB
Models	
daily number of models	576 models
number of models currently	10,000 models
number of models by SC 07	53,000 models

Table 1: This table summarizes the amount of event and profile data produced daily and the amount that will be available by SC 07. The number of models is also summarized. Emergent behavior is computed by a meta-analysis of the models.

The database stores information about each pcap file; site, time interval, number of packets, number of dropped packets, number of hosts; and stores URLs to said pcap file along with URL to the computed profile file. Every ten minutes, the cluster models for each site, as well as global cluster models for all the sites, are stored in the SQL database to allow easy access and search capability.

Each day the pcap gateway server receives 576 pcap files totaling on average 7.6GB and 97 million packets. By Supercomputing 2007, the Angle project will contain 3.4TB of pcap files, 43.7e9 packets and 53,000 K means clustering results.

4 Visual Analytics

The Angle visualization was developed to monitor anomalous flows or events across the system. Flows and events that are likely to be related to unusual behaviors are identified visually. Our visual analytics system represents statistical models in data viewers to allow easy recognition of both regularities and anomalies in data. The visual analytic components in Angle are the Map View, the Model View and the Inspector View. The Map View shows the spatial locations of IP hashes identified as anomalies.

To get a concise view of models and emergent behavior in the Model View, we utilize Multidimensional Scaling (MDS). MDS is a technique that allows the projection of points in a multidimensional space into a lower dimensional space (in our case, 2D space). The benefit of MDS is that it places similar clusters close together and dissimilar clusters far apart. The user can observe the spatial arrangement of clusters to infer relations between clusters within a model and across models. The axes of this plot are arbitrary (because the MDS model is invariant with respect to rotation and translation), so we omit other annotations in the plot. The goal of this plot is simply to identify clusters of similar clusters and to recognize similarities among emergent clusters.

While the emergent behavior analysis and visualization helps identify anomalies, the time series Inspector View helps characterize the anomaly. In some cases certain features are more helpful for explaining a given type of behavior.

The baseline can be constructed by aggregating the information we capture from all our monitored locations or any single one of them. This helps the analysis of both global and local structure and will expose suspicious behavior that would not otherwise be obvious.

Figure 2: This is a view of one of the Angle models. Squares represent emergent clusters, while circles represent regular clusters. The Angle system currently has over 10,000 separate models that can be selected and used to look for anomalies and emergent behavior in near real time or from historical data.

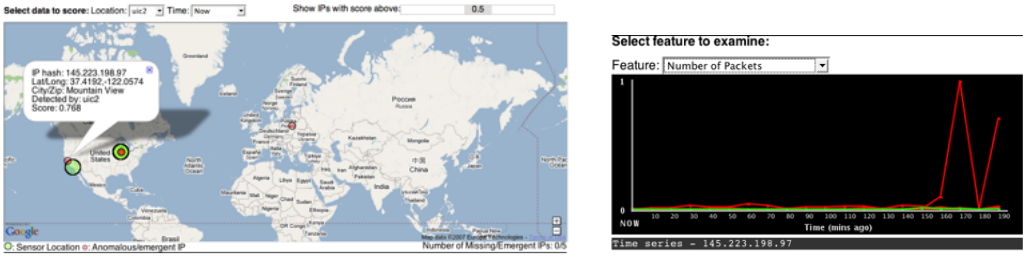

Figure 3: In the left panel is the map view of emergent behavior. Once a model is selected, emergent or anomalous IPs are identified in pink. In the right panel is the inspector view. Each anomalous or emergent IP also appears as a time series in red for the prior 200 minutes. The mean for the model appears in green. This is the inspector view of emergent behavior.

Angle takes advantage of new web technologies to provide a responsive and an intuitive interface for navigating large amounts of network packet data. The user browsing the web interface can select many options that regulate how data should be analyzed. These options are sent to the Angle server using AJAX (Asynchronous Javascript And XML). This technology lets a web page make a request for information in the background without disrupting the user's browsing experience. The server receiving these requests performs an analysis using the user's parameters. Once the analysis is complete, the results are returned to client's web browser and appropriate views are updated.

To use the interface, the first task for the user is to select a model to determine emergent behavior. This is accomplished using the lower-right Model View panel of the interface. Next, the user selects the type of comparison to occur (Prior clusters, Common clusters or Daily average) and the data capture location for Angle to use for signaling emergent behavior. Data sets captured at any interval may be specified to search for emergent clusters.

Once emergent behavior is located, a user can determine which IP address hashes (IPhash) are involved in the anomalous activity. The upper Map View panel, showing a map of the world, is utilized to place anomalous IPhashes onto the world map. The angle score slider at the upper-right corner of the panel controls the threshold of Angle scores to be visualized. The user can experiment with this slider, starting it near 0, and moving it up to filter visualization of low-scoring IPhashes. Clicking on red circular icons reveals information about the selected IPhash.

Angle uses the selected time interval and site of anomalous IPhashes to construct a features graph against time in the lower-left Inspector View panel. The user can select one of the 8 features to be graphed. The time line represents features in the last 200 minutes. IPhashes that stand out within a given feature are easy to notice, such as in the case shown below, where an extremely large number of packets was transmitted.

To summarize, all interface components are linked, color-coordinated and updated in real-time. This highly interactive display allows the user to filter and focus her analysis. The views are complementary. The Model View provides a visually concise representation of emergent behavior. The Map View provides the geo-spatial context of the anomalies, while the Inspector View provides the quantitative description of the anomalous behavior over time.

File Size	1GB	10GB	100GB
UIC to ISI			
UDT	83.7	86.0	85.1
TCP	8.8	8.8	8.8
UIC to UC			
UDT	641.7	559.1	551.2
TCP	204.8	205.6	202.4

Table 2: The tables shows the transfer throughput in Mb/s for transfering pcap files from UIC to UC and from UIC to ISI using Sector/UDT and using TCP. Notice that Sector is about 10x faster to ISI, which limited to 100 Mb/s and about 2x faster to UC.

5 Re-Analysis

The Angle System is designed to support re-analysis of the profiles, cluster models, and meta-analysis of the cluster models used to identify emergent behavior.

When re-analysis is required, the desired sets of pcap files can be retrieved from the edge nodes using the Sector distributed computing platform. Some experimental results using Sector to transport large collections of pcap files for re-analysis are Table 2. Once the required pcap files are available, the re-analysis per se is accomplished using the Swift workflow system [4].

6 Experimental Results

To illustrate the strengths of Angle, below are three scenarios of defining emergent behavior to expose interesting anomalous behaviors.

The first scenario uses data from the University of Chicago from July 18th at 1:50pm as the model. Angle flagged a computer in Chicago as being anomalous. Looking at time series for "Number of IPs" feature shows a large spike during the selected time period. The feature profile data confirmed this visual result. This Chicago computer's address touched 655 distinct IPs and 54 ports with small packets (408 bytes) within a 10 minute interval. This anomalous behavior suggests that a malicious user was scanning computers on the network.

In the second example, we detected a computer in Dallas that sent data to 15 different ports on two computers in UIC within a 10 minute interval. The malicious user might have been attempting to exploit services running on carefully chosen open ports on these two machines.

In the third example we used data from all locations on July 23rd at 3pm to flag a computer in Springfield, Illinois that sent a large number of packets (about 126,000 per 10 minutes) from port 80. This unusual behavior stayed consistent over a 30 minute time period and could be explained by a user in UIC either streaming video or downloading a large file. Notice that when we used only UIC to build a model, this anomalous behavior was not detected.

References

[1] A. Lazarevic, L. Ertoz, A. Ozgur, J. Srivastava, V. Kumar, Evaluation of Outlier Detection Schemes for Detecting Network Intrusions, Proc. Third SIAM International Conference on Data Mining, San Francisco, CA, May 2003.

[2] Programming with pcap, www.tcpdump.org/pcap.htm.

[3] Snort, www.snort.com.

[4] I. Raicu, Y. Zhao, C. Dumitrescu, I Foster, and M. Wilde, Falkon: a Fast and Light-weight tasK executiON framework Supercomputing Conference 2007.