Public Review for

An Internet Routing Forensics Framework for Discovering Rules of Abnormal BGP Events

Jun Li, Dejing Dou, Zhen Wu, Shiwoong Kim,
and Vikash Agarwal

There is an emergent interest in using statistical and Machine Learning techniques to mine network data. Papers that use neural networks, Bayesian analysis, SVM, PCA, are increasingly common. They aim to move beyond direct measurements to more sophisticated tasks such as anomaly detection or root cause analysis. Though it might take some time before this area of research matures, the initial results are promising and deserve to be encouraged.

This paper uses BGP updates to detect Internet anomalies. The paper formalizes the problem as a multi-label classification, where the labels are: normal, blackout, worm, misconfiguration. The objective is to tag a BGP event (i.e., a series of BGP updates) with one of these labels. The paper discusses a set of relevant features such as the number of withdrawals, the number of announcements of a recently withdrawn prefix, etc. it shows that combinations of these features can be used effectively to distinguish worm and blackout events from normal events.

Though the paper is innovative and interesting to read, the value of the reported performance results is limited. The results in the paper focus on detecting normal BGP behavior from an abnormal behavior caused by either a major worm or a blackout. In practice, this task is fairly simple and usually does not require any advanced statistical techniques. The paper would have been much stronger if it focused on distinguishing various abnormalities form one another - i.e., can we identify whether an abnormal BGP event is caused by a worm, a blackout, or a misconfiguration? Yet, to answer this question, one needs a potentially large number of labeled examples of worm, blackout, and misconfiguration events. Unfortunately the number of known such events is relatively small making it hard to come up with a robust classifier.

Public review written by

Dina Katabi
MIT, Cambridge,
Massachusetts, USA

acm sigcomm

An Internet Routing Forensics Framework for Discovering Rules of Abnormal BGP Events

Jun Li, Dejing Dou, Zhen Wu, Shiwoong Kim, Vikash Agarwal
University of Oregon
{lijun, dou, zwu, shkim, vikash}@cs.uoregon.edu

ABSTRACT

Abnormal BGP events such as attacks, misconfigurations, electricity failures, can cause anomalous or pathological routing behavior at either global level or prefix level, and thus must be detected in their early stages. Instead of using ad hoc methods to analyze BGP data, in this paper we introduce an Internet Routing Forensics framework to systematically process BGP routing data, discover rules of abnormal BGP events, and apply these rules to detect the occurrences of these events. In particular, we leverage data mining techniques to train the framework to learn rules of abnormal BGP events, and our results from two case studies show that these rules are effective. In one case study, rules of worm events discovered from the BGP data during the outbreaks of the CodeRed and Nimda worms were able to successfully detect worm impact on BGP when an independent worm, the Slammer, subsequently occurred. Similarly, in another case study, rules of electricity blackout events obtained using BGP data from the 2003 East Coast blackout were able to detect the BGP impact from the Florida blackout caused by Hurricane Frances in 2004.

Categories and Subject Descriptors

C.2.2 [**Computer-Communication Networks**]: Network Protocols; C.2.3 [**Computer-Communication Networks**]: Network Operations; I.2.6 [**Artificial Intelligence**]: Learning; H.2.8 [**Database Management**]: Database applications—*Data mining*

General Terms

Security, Design, Reliability

Keywords

routing forensics, abnormal BGP events, data mining, Internet worms, blackout

1. INTRODUCTION

Abnormal Border Gateway Protocol (BGP) events, including attacks [31], misconfigurations [21], and large-scale power failures [32], can lead to anomalous or pathological routing behavior and affect the global routing infrastructure. In the past few years, several well-known BGP events have been reported. For example, in April 2001, a misconfiguration caused AS 3561 to propagate more than 5,000 invalid route announcements from one of its customers, causing connectivity problems throughout the entire Internet [22]. In January 2003, the Slammer worm caused a surge of BGP updates [16]. In August 2003, the East Coast electricity blackout affected 3,175 networks and many BGP routers were shut down [7]. Also, smaller scale anomalies, although probably unnoticeable, can happen even more frequently, further raising concerns for such events on a daily basis.

In this paper, we apply Internet Routing Forensics (IRF) methodology to study and detect abnormal BGP events, or **ABEs**. Forensic science was originally defined as the application of scientific methods and analysis to the search and examination of physical evidence in support of criminal investigation for the law enforcement community, and has also been used by computer scientists in recent years, typically to identify and respond to security breaches [27, 23]. With the advent of large-scale Internet routing data collection (such as RIPE [2] and RouteViews [30]), interestingly, we can also apply forensic science to study large-scale routing events in the Internet.

We build a reliable but flexible **IRF framework** that can continuously train itself using data mining techniques to learn detection rules from already-known ABEs, and further use these rules to detect unknown ABEs. Both the training and detection are based on a large volume of BGP data.

We describe challenges facing the IRF framework in Section 2, and present its design in Section 3. After describing our ground work from empirical studies of BGP in Section 4, we then focus on demonstrating the efficacy of the IRF framework through two case studies. Section 5 is a case study on Internet worm events. First using the BGP data from the CodeRed and Nimda period to train the system, we show that our IRF framework is accurate and fast in detecting the impact on BGP by the Slammer worm. Section 6 is another case study on electricity blackout events, also showing the effectiveness of our IRF framework. We describe related work in Section 8 and conclude the paper in Section 9.

2. CHALLENGES

An effective IRF framework should be able to process highly complex data to quickly and accurately detect ABEs according to the different needs of different users. We present the challenges for doing so from four aspects:

- **Accurate Detection.** An IRF framework simply fails if it cannot return accurate detection results. Unfortunately, an ABE does not always have clear-cut symptoms; worse, it may even appear legitimate in many aspects. Some ABEs may have never appeared before.

- **Fast Detection.** Many ABEs require immediate attention. Fast detection, therefore, is critical. Whereas it is easier to do offline analysis using archived BGP data, in many circumstances it is necessary to be able to do online analysis while the archive is collecting information in real time. It will be more useful, for example, if we can detect that a misconfigured router is causing a significant amount of BGP updates as it happens, rather than some time later.

- **Usability.** Users from different domains or contexts may define ABEs differently. An ABE that user A wants to pay attention to may not be a concern of user B. Different users may also want different levels of details regarding a newly detected ABE, in part because the users will have different levels of knowledge of how BGP works. All users will need a straightforward interface to express their requests and feedback to the system, or to receive alerts and rules of ABEs from the system. The IRF framework must support the needs of different users.

- **Complexity of data.** To detect various ABEs over the Internet, one has to look into the "conversation" records of BGP routers in addition to their routing tables. Such routing data, however, is complex to be processed due to the complexity of BGP. The RouteViews archive of BGP data provides an example [30]. Not only are the data of a huge size—currently more than 700 GB and growing 30 GB per month, but a high level of noise is also prevalent when extracting ABE-related information. Intermediary data that are more meaningful than the raw data often should be prepared, such as global-level statistical data and local-level data on routing paths toward particular Autonomous Systems (ASes) or prefixes. Note that the intermediary data can also be of huge size, sometimes even larger than the raw data. Here, one must carefully select meaningful parameters in generating the intermediary data; otherwise, an arbitrary set of parameters will lead to low accuracy, and a very large set of parameters (with many of them being irrelevant) will lead to low performance.

3. DESIGN

Figure 1 shows our IRF framework. It has four major components: the forensics input, data processor, anomaly resolver, and the user. It also has two running threads: the *training thread* and the *detection thread*. Both threads involve all four components. The forensics input data is from the archive of BGP data (such as [2] or [30]). The data processor processes the forensics input and provides data in a format that the anomaly resolver can handle. In the training thread, the anomaly resolver will discover the rules of particular ABEs by using data mining techniques. In the detection tread, the anomaly resolver will apply those rules to its input to determine whether the input is normal or associated with an ABE. (Note that the detection may be either online or offline.)

3.1 Data Processor

The data processor takes the forensics input, cleans noise when necessary, processes it to obtain values for a set of selected parameters over every time window, and provides a *database table* or a multi-dimensional data model (such as a data cube [12]), as the input for the anomaly resolver.

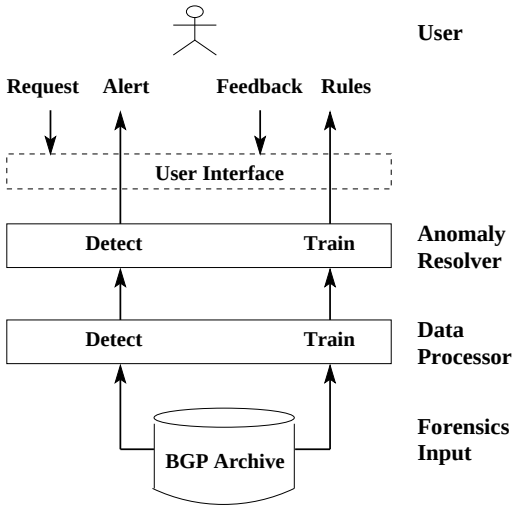


Figure 1: The architecture of the IRF framework

The input is used either by the training thread or by the detecting thread. Data used for training can be associated with already-known BGP events, such as those mentioned in Section 1. Each row of a training database table can have a *label* field to specify whether the data is associated with a particular ABE (e.g. a worm event) or should just be treated as normal. On the other hand, if used for detection, every row is instead unlabeled.

The data can be from both the global level and the IP prefix level. At the global level, familiar parameters include the number and type of updates, the inter-arrival time of updates, the percentage of updates related to the top x% of active prefixes, etc. At the prefix level, we create graphs describing the AS-paths taken to a given destination prefix from other AS sources in the Internet, and analyze parameters related to such graphs.

Parameters used in the data processor are selected according to the *information gain* of each parameter. Parameters with high information gain are highly discriminating parameters, and selecting them will make the data mining process more accurate and compact. Calculating the information gain is a widely accepted method for attribute relevance analysis. The detailed procedure can be found in [12].

3.2 Anomaly Resolver

The anomaly resolver, as shown in Figure 2, consists of a data miner and a rule processor that build optimized rules (both associated with the training thread) and a detector to use these rules to detect ABEs (associated with the detection thread).

In detail, the data miner processes the training data to discover rules between the input data and ABEs. Particularly, since ABE detection can be treated as a data classification problem of how to classify the input data into categories of normal events and various abnormal events, the data miner supports a classification functionality. It applies the widely used C4.5 algorithm [25], which we have found also works well in our context. This algorithm first builds a *decision tree*, where each internal node denotes a test on a parameter, each branch represents an outcome of a test, and each

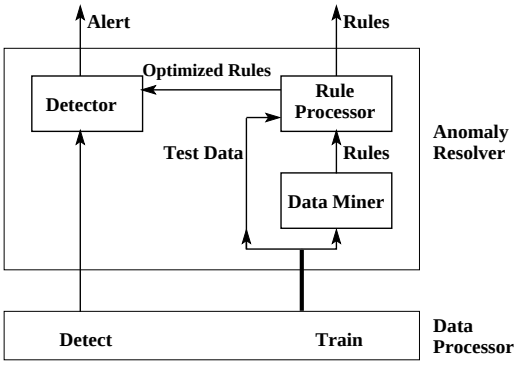


Figure 2: The architecture of the anomaly resolver

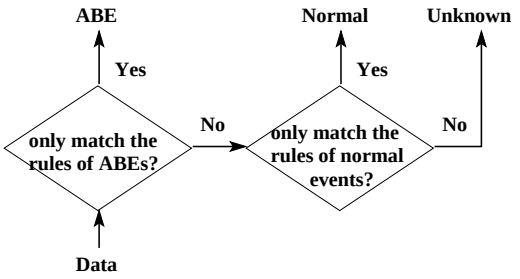


Figure 3: The decision process of the detector

leaf node represents a class (i.e. normal or a specific ABE). From the decision tree, C4.5 then can derive IF-THEN classification rules. It can also use training data to estimate the accuracy of each rule, or remove any conditions in the antecedent of a rule that do not improve the rule's accuracy.

Empirical evidences also show that some parameters exhibit correlations among each other. We are currently considering extending the Apriori algorithm [3] to discover those correlations in the form of association rules.

Upon the generation of classification and association rules from the data miner, the rule processor can select the rules which have high accuracy, and then optimize them by removing redundancies or inconsistencies. The detector then uses optimized rules to detect ABEs. It takes the steps shown in Figure 3: If the data matches and only matches rules of an ABE, the data is considered to have come from an ABE; on the other hand, if the data matches and only matches *normal* rules, the data is normal. Otherwise, if no match (as described above) is found, the detector will treat the data as "unknown."

3.3 User

Users, such as ISP operators, also play an important role in this IRF framework. First of all, users may choose training events that are of particular interest to them, whether they are on a very large or relatively small scale. Chosen events can be known to users only, or well known by the Internet community at large. This allows a flexible definition, created by the user, of what types of events are considered normal and abnormal.

Note that while major events do not occur on a regular basis, small-scale misconfigurations and other anomalies probably happen more frequently. By studying routing events at small scale, the IRF framework can provide insight into the more common cases, thereby improving the day-to-day operation of BGP routing. Recall that the data processor is equipped to prepare data not only from the global level, but also from the IP prefix level. Therefore, the IRF framework should not only be able to allow users to choose large well-known events for training that appear in aggregated global statistics, but also smaller events that appear only at the network level.

Users may interact with the IRF framework through a user interface. In the training tread, besides choosing ABEs for training and labeling the data rows as normal or not, users could also help the parameter selection process. For example, when all information gain values of candidate parameters fall into a narrow range, we can rely on users to decide which parameters are more interesting to them. In the detection thread, users can specify what particular ABEs to monitor, such as the type, domain, and period of those ABEs, how they want to be notified, and whether detailed information is also needed.

Users can receive two main types of output: alerts of possible ABEs from the detection thread, or ABE detection rules from the training thread that users will then tune or verify. The user can further use out-of-band knowledge to verify whether the output is accurate, and provide feedback to the framework.

4. GROUND WORK FROM EMPIRICAL STUDIES

Using the BGP updates archive from RouteViews [30] and RIPE [2], we have done several important ground works. We have analyzed independent well-known BGP events that network operators would usually regard as abnormal—such as the 2003 August East Coast blackout event and the 2003 January Slammer worm event. Furthermore, we have also tried to have an up-to-date understanding of BGP dynamics.

We obtained encouraging results in characterizing BGP in general and BGP during these well-known events in particular, strengthening our belief that more systematic analysis using data mining will likely result in even more accurate, comprehensive patterns and characterizations for identifying ABEs. In particular, before we apply information gain mechanisms to select what parameters to use from among candidate parameters, we are able to choose candidate parameters by analyzing various parameters throughout these empirical studies. Recall the data processor needs to collect values of only those useful parameters to send to the anomaly resolver.

4.1 The BGP Impact from the 2003 East Coast Blackout Event

In this work, we analyzed BGP behavior at the August 2003 East Coast blackout, during which 3,175 networks lost their connectivity [7], and identified a number of metrics for analyzing BGP behavior. Global metrics studied include the number of updates, the number of explicit withdrawals, inter-arrival time of BGP updates, etc. At prefix level, we studied per-prefix AS-path graphs that show AS-paths taken to a given destination prefix from other AS sources in the Internet, including characteristics related to graph changes such as the node degree and the number of nodes. Our re-

sults show that during the blackout event, there was an apparent increase in the number of explicit withdrawals (Figure 4), and at the prefix level, the AS-path graphs of some prefixes also show a sharp decrease in the number of edges and nodes, as well as changes in node degrees. Nonetheless, our results also show that BGP can recover from the blackout in a timely manner, and the negative impact of the power outage was limited only to the affected areas.

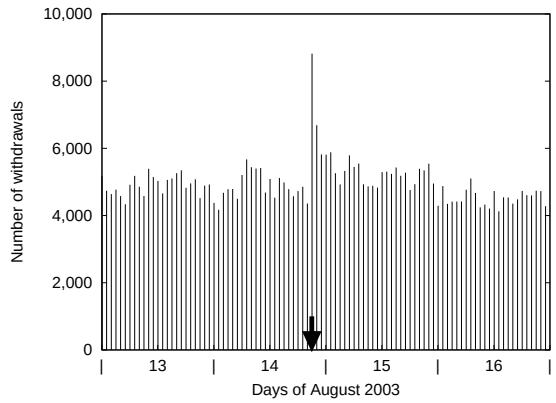


Figure 4: Hourly explicit withdrawals surrounding the East Coast blackout period. The short arrow marks the starting time of the blackout.

4.2 The BGP Impact from the 2003 Slammer Worm Event

Our work studying the January 2003 Slammer worm event is similar to the work on the blackout. Like the blackout event, the reachability of many networks was affected by the congestion produced by the worm traffic. We studied the same metrics as we did in the blackout, including both global-level and prefix-level graph-based metrics. The results show that the number of withdrawals during the worm event also increased dramatically (Figure 5), and the number of nodes in the AS-path graphs for certain prefixes oscillated rapidly.

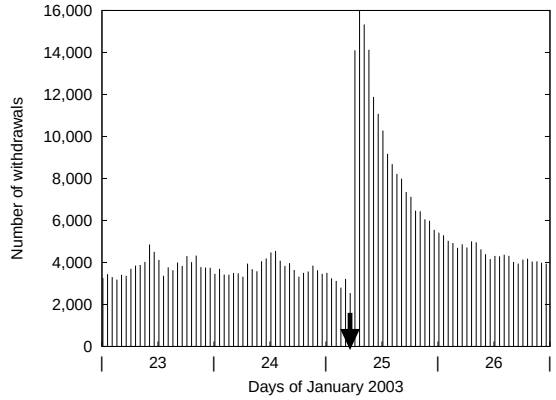


Figure 5: Hourly explicit withdrawals surrounding the Slammer period. The short arrow marks the starting time of the Slammer worm propagation.

4.3 BGP Dynamics Study

This study helped determine normal ranges of various BGP parameters related to BGP dynamics. These parameters are calculated by checking every pair of updates from a single BGP speaker about a particular prefix, and are classified based on the update fields. For example, *WADiff* is about announcing a new path after withdrawing an old path toward a prefix, *WADup* is about reannouncing a path after withdrawing the same path, *AADiff* is about announcing a new path (thus an implicit withdrawal), and *AW* is about withdrawing a path announced earlier.

Analyzing six months of BGP updates, we have found that the characteristics of BGP dynamics have changed significantly since the last study conducted eight years ago [15]. For example, the duplicate withdrawal messages that were dominant in 1997 now contribute the least of all dynamics types. Implicit withdrawals and policy fluctuations instead have become the first and second most dominant, respectively. We compared ASes of different sizes in terms of their contributions to the overall dynamics, and found that larger ASes generally experience more dynamics. We also studied time patterns of BGP dynamics, and found a less obvious weekly pattern but a strong daily pattern. Moreover, we found that throughout the six months of study, a parameter on BGP dynamics often has its own unique inter-arrival time distribution, as shown in Figure 6.

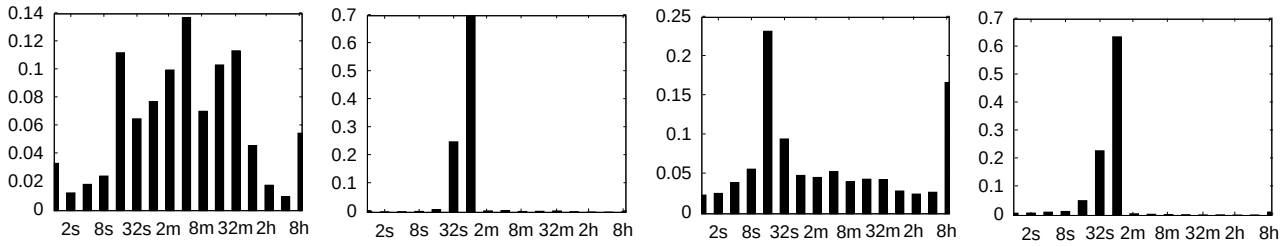
5. CASE STUDY I: WORM EVENTS

In this case study, we focus on routing anomalies during the propagation of Internet worms, which can cause severe congestion and session breaks of BGP routers at the edge of the Internet, as pointed out in [31, 16, 8, 9]. We analyzed BGP data during the outbreak of two different worms—CodeRed and Nimda—as well as BGP data during normal periods (when no ABEs are known to have occurred), and applied our data mining process to these data to discover rules for detecting occurrences of worms. We further verified the accuracy of these rules by applying them to data collected from the Slammer worm period, as well as data collected from different normal periods. From this case study, we found that in terms of both the speed and accuracy, our routing forensics methodology is successful. We describe the details step by step in the following.

5.1 Data Source and Data Cleaning

The BGP data archive that we used to prepare database tables was the RIPE archive. It is a huge archive of BGP updates and routing tables that are continuously collected by RIPE monitors around the world. We used the BGP update data from six randomly selected peers. (We did not use the Oregon RouteViews archive, as it does not contain BGP updates for the CodeRed and Nimda worm periods that we want to study.)

We need to be careful when considering the effects of the BGP *session reset* process: When two BGP routers reset their BGP sessions, one will send the other a full set of BGP announcements derived from its whole routing table. (Recall that every BGP session runs on top of TCP.) Because the BGP session between a RIPE monitor and its peering router crosses multiple hops, it may break more easily than in a typical single-hop session and thus reset more frequently, "polluting" the BGP archive with duplicate BGP announcements from repeated session resets. This effect is even more



(a) WADiff (b) WADup (c) AADiff (d) AW

Figure 6: Distribution of inter-arrival time for four types of BGP dynamics during March 2004.

obvious when a worm propagates and causes network link congestion, as observed in [31]. We borrowed the method from [31] to filter unnecessary announcements from BGP session resets. Basically, for each session reset period, the first announcement for a prefix will be treated as from a session reset and filtered, and the subsequent announcements for the prefix will be kept.

5.2 Data Processing

As discussed in Section 3.1, it is critical that the training and detection threads use a small optimized set of parameters. Based on our empirical studies (see Section 4), we first selected parameters that were likely to be useful. We then calculated the information gain for each parameter to obtain the most relevant ones.

Table 1 lists 35 parameters that are considered in this study. Numbers of BGP updates (parameters 1–3) are certainly indicators of the Internet routing dynamics. Because one BGP update message could speak for multiple prefixes, we also considered the number of updated prefixes (parameters 4–6). Labovitz *et al* [15] defined a method to classify BGP updates into different types; with our further refinement, this leads to nine parameters (parameters 7–15), each counting a different type of BGP updates. Furthermore, parameters that may capture temporal characteristics of BGP updates are also important, as demonstrated by Figure 6. Corresponding to every parameter from parameters 6–15, the inter-arrival time of a particular BGP update type can be studied; for example, related to parameter 10, *AADiff*, we can have the inter-arrival time of two announcements that declare two different paths for reaching a specific prefix. We introduce the mean and the standard deviation of every such inter-arrival time as two new parameters, leading to another twenty parameters (parameters 16–35).

The information gain values of these 35 parameters range from 0.005 to 0.2. We selected *nine* parameters whose information gain values are much greater than the rest parameters. Those nine parameters are underlined in Table 1.

To obtain rules regarding worm events that can affect BGP, we collected training data as follows. Using the CodeRed and Nimda worms as training ABEs, we prepared data from an eight-hour period immediately after each worm started to propagate. We also prepared data from *ten* randomly chosen "normal" days (dispersed within a two-year period from July 2001 to August 2003), in which no major events were known to have happened.

To test the rules obtained from the training, we further prepared data from the day when the Slammer worm was active (January 25, 2003). To provide a basis of comparison when testing the rules, data from another set of *ten* randomly chosen "normal" days were also collected.

ID	Parameter	Definition
1	Announce	# of BGP announcements
2	Withdrawal	# of BGP withdrawals
3	Update	# of BGP updates (= Announce + Withdrawal)
4	AnnouPrefix	# of announced prefixes
5	WithdwPrefix	# of withdrawn prefixes
6	UpdatedPrefix	# of updated prefixes (= AnnouPrefix + WithdwPrefix)
7	WWDup	# of duplicate withdrawals
8	AADupType1	# of duplicate announcements (all fields are the same)
9	AADupType2	# of duplicate announcements (only AS-PATH and NEXT-HOP fields are the same)
10	AADiff	# of new-path announcements (thus implicit withdrawals)
11	WADupType1	# of re-announcements after withdrawing the same path (all fields are the same)
12	WADupType2	# of re-announcements after withdrawing the same path (only AS-PATH and NEXT-HOP fields are the same)
13	WADup	WADupType1 + WADupType2
14	WADiff	# of new paths announced after withdrawing an old path
15	AW	# of withdrawals after announcing the same path
16 ... 35		the mean and the standard deviation of ten different types of inter-arrival time

Table 1: Parameter list

Each worm and normal period is further divided into 1-minute bins, with each bin represented by exactly one data row. Data from each bin is calculated in terms of the parameters listed in Table 1. As a result, for each bin, a new row is added to a corresponding database table used for training or testing. When used for training, a new row will also be labeled as either "worm" or "normal." Table 2 presents a sample database table for training. In total, the database table used for training contains 14,116 rows of normal data and 958 rows of (CodeRed and Nimda) worm data; The

Rules for the normal class		Rules for the worm class	
1.	IF Announce <= 236 THEN class = "normal" [99.8%]	1.	IF Announce > 236 AND Update_prefix > 720 AND WADiff > 47 AND AW > 84 AND WADup > 28 THEN class = "worm" [89.1%]
2.	IF Updates <= 358 AND Withdraw_prefix <= 106 THEN class = "normal" [99.6%]		
3.	IF WADiff <= 34 THEN class = "normal" [99.5%]		

Figure 7: Sample rules generated by data miner

database table used for testing uses data from 10 "normal" days and one Slammer day, with about 1440 rows for each day.

Announce	Update	...	WADiff	AW	Label
107	116	...	26	15	normal
140	146	...	16	15	normal
...	...	...	...	...	...
523	537	...	100	27	worm
884	897	...	289	29	worm

Table 2: A sample database table. Each row corresponds to a 1-minute bin.

5.3 Data mining for classification rules

Using the database table for training as the input, we apply the procedure described in Section 3.2 to discover rules that classify data as being associated with a worm event or being normal. Note that we do not differentiate between data from different types of worms. On the contrary, we sought to obtain rules for just two classes—normal and worm.

When we did not calculate the information gain of parameters for attribute relevance analysis, we used all 35 parameters to obtain classification rules. The data miner generated 18 rules for the normal class and 9 rules for the worm class. After applying information gain and selecting the top 9 parameters, our data miner generated 7 rules for the normal class and 5 for the worm class.

All rules obtained by the data miner are in IF-THEN form. Figure 7 shows a subset of such rules. The antecedent of each rule is a conjunction of conditions, where each condition is used to check the value of a parameter. The consequent is a label that identifies the class of the data (i.e. "normal" or "worm"). In general, the normal rules consist of conditions that the value of a parameter is less than some threshold, while the worm rules take the forms of the value of a parameter is larger than some threshold. This makes sense considering that worm propagation may bring congestion and break down BGP sessions, increasing the number of BGP updates and other BGP types. When applying these rules for detection, we also found that the rules with higher accuracy are more effective and matched more data rows.

5.4 Applying rules for detection

In order to test whether the rules generated by our data miner can successfully detect abnormal BGP events, we feed these rules to the framework's detector to process the database table that we prepared for testing (see Section 5.2 for the details of the table).

To simulate the behavior of an online detector, we used a sliding window with a size of 60 minutes, thus using 60 rows of data from the testing database table, to apply the detection process. (Recall that every row from the testing database table corresponds to one minute of data.) The sliding window moves forward 10 minutes (or 10 data rows) each iteration. The detector will check each row against all normal and worm rules. Every row will either match only normal rules, only worm rules, no rules, or both worm and normal rules. We classify the latter two cases as belonging to the "unknown" class. Corresponding to the sliding window at a particular time (with a total or 60 rows), the detector reports the percentage of rows matching normal rules, the percentage matching worm rules, and the percentage of unknown data (which is 100% minus the first two percentage values).

As the sliding window moves, the process is repeated until it is the end of a testing day. As a result, for every testing day the detection results will include the values of each percentage type over the whole day. After processing the whole testing database table that contains ten "normal" days and one Slammer day, an output file containing detection results is then generated for each testing day.

5.5 Results and analysis

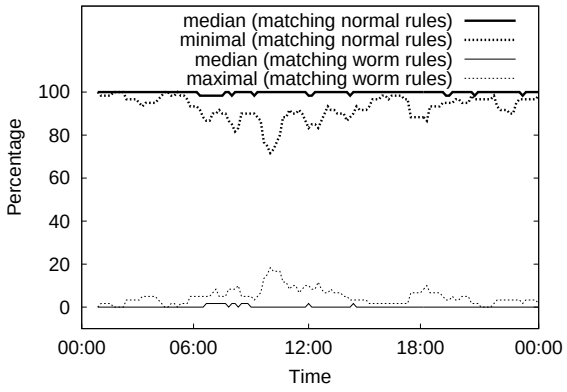
Suppose that α is the percentage of data matching only worm rules, and β is the percentage of data matching only normal rules. If our training phase were successful in deriving accurate classification rules, we would expect that during a *normal* period, α would be close to 0% and β would be close to 100%. Conversely, during the Slammer worm period, α would be much higher than 0% and β would be much lower than 100%.

We generated detection results using both 35 parameters (without calculating information gain) and 9 parameters (after calculating information gain). As can be seen in Figure 8(a) and 8(c), when applying rules to data from ten different "normal" days, a very high percentage of data match normal rules, while a minuscule amount match worm rules. On the other hand, when applying rules to the Slammer day (Figure 8(d)), the percentage of data matching normal rules is high until the time when the Slammer worm is known to have started heavy propagation (5:30 AM UTC). At that point, the percentage of data matching worm rules quickly skyrocketed, close to 100%. The trend continues until later in the day, when (presumably) the effects of the worm on BGP diminished. Note that when using all 35 parameters together, Figure 8(b) suggests that the worm impact is visible for only about five hours, while Figure 8(d) indicates that when considering only the nine parameters with highest information gain, the worm impact on BGP actually lasted longer (about twelve hours).

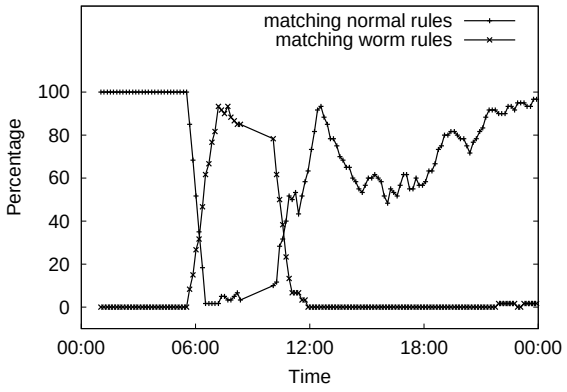
The accuracies of rules can also be considered in the detection process. For example, data that match worm rules with γ accuracy can be treated as worm data with a probability of γ. We obtained similar results that clearly distinguished the Slammer worm period from a normal period.

6. CASE STUDY II: BLACKOUT EVENTS

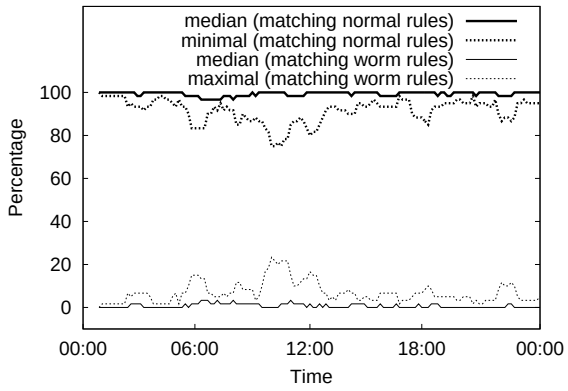
In addition to worm events, we also applied our IRF frame-



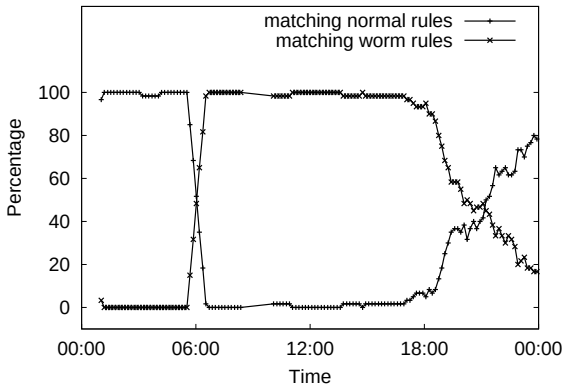
(a) Results for normal periods (35 parameters)



(b) Results for the Slammer worm period (35 parameters)



(c) Results for normal periods (9 parameters)



(d) Results for the Slammer worm period (9 parameters)

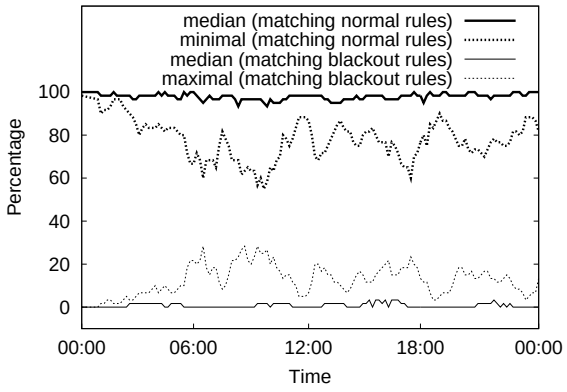
Figure 8: Applying rules to detect a worm event. For graphs about the detection results over normal periods, the solid lines represent the median values of the percentages from ten days, and the dashed lines represent the worst-case values (minimal percentages for matching normal rules and maximal percentages for matching worm rules).

work to another type of event, the large-scale electricity blackout. On August 14, 2003 (21:10 UTC), a major power outage paralyzed dozens of cities in the eastern United States and Canada. Although the outage spread within just three minutes, it took about 22.5 hours, from August 15, 4:00 UTC to August 16, 2:30 UTC, to fully restore the electricity, The blackout affected the connectivity of 3,175 networks according to the Renesys report [7]. Also, another large-scale power outage caused by Hurricane Frances, which landed on Florida on September 3, 2004, caused 2.8 million customers to lose power [1].

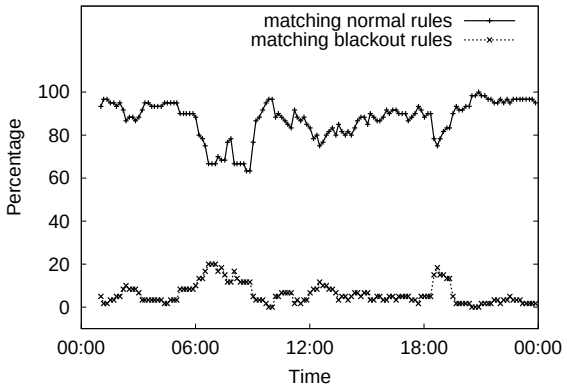
In this second case study, we applied the same process described in Section 5. More specifically, we used the six-hour period of BGP updates corresponding to the East Coast blackout period (August 14, 21:10 UTC – August 15, 4:00 UTC) as the training data, and then applied the derived rules to the 24-hour period (September 3, 2004) during which the Florida blackout happened. The rules distinguish data belonging to a normal period from data belonging to a blackout period. We considered results using all 35 parameters, as well as results gained using the one parameter with the highest information gain (its value is much greater than the rest of the parameters).

With the rules we obtained using all 35 parameters, the Florida blackout period does *not* look very different from normal periods, as shown in Figure 9(a) and 9(b). However, when using just the single parameter with the highest information gain, *Withdrawal*, a clear distinction appears. During normal periods the percentage of data rows matching blackout rules is usually below 40% (Figure 9(c)), while during the blackout period it climbs to 60–80% (Figure 9(d)).

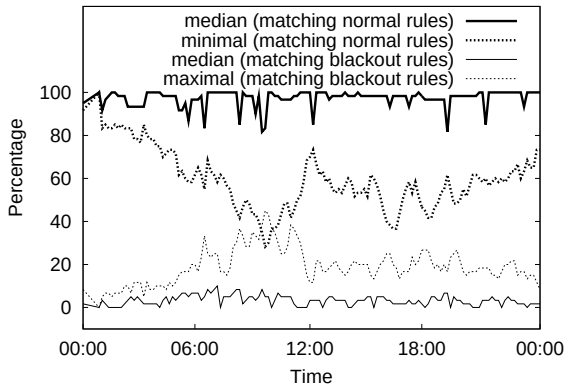
In a normal day, a rare worst case may occur in which the percentage of data that match blackout rules can be as high as 45%, even greater than data matching normal rules (28%) (Figure 9(c)). But notice that when a blackout happens, the percentage of data matching blackout rules will be at



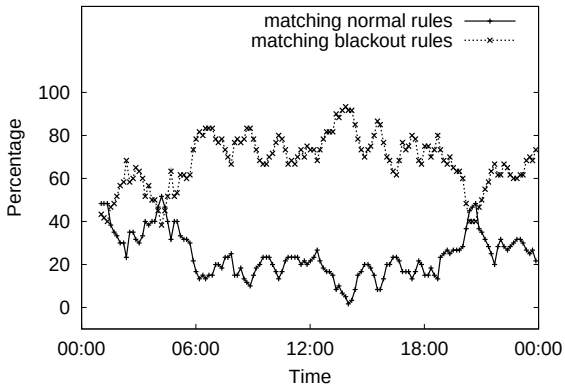
(a) Results for normal periods (35 parameters)



(b) Results for Florida blackout period (35 parameters)



(c) Results for normal periods (1 parameter)



(d) Results for Florida blackout period (1 parameter)

Figure 9: Applying rules to detect a blackout event. For graphs about normal periods, the solid lines represent the median values of the percentages from ten days. The dashed lines represent the worst-case values (minimal for normal and maximal for worm).

least 65% (see Figure 9(d)), meaning that the ABE detector should use a high percentage value as a threshold in order to determine whether or not there is a blackout event.

7. DISCUSSIONS

The success of detecting a completely unseen worm (i.e. Slammer) or a blackout event shows that our IRF approach to discovering rules of ABEs is promising. In particular, the two case studies show that it successfully addresses the challenges raised in Section 2. Not only can the IRF framework detect the BGP impact from a new subsequent event, such as the Slammer worm or the Florida blackout event, when it occurs, but also quickly once it occurs. The IRF framework also handles data complexity by choosing most informative parameters, and raise the usability by allowing users to provide various input (such as specific training events, feedback on new rules, or their requests) and receive various output (such as alerts or new rules).

Note that our case studies are mainly used to showcase the efficacy of the framework, not to show that data mining is the only way to distinguish a couple abnormal BGP events. IRF needs data mining not just because it works in our case studies above, but because of its flexibility and generality in detecting a variety of abnormal BGP events, some of which will be prefix-level abnormal BGP events that has yet to be investigated, and some of which we cannot foresee today.

It may appear that abnormal BGP events only occur rarely and training data is less than sufficient, thus the IRF framework is probably both unnecessary and insignificant. However, abnormal BGP events can be at both global level (such as those from our case studies) and prefix level. The latter is also important to study; typically at small scale, they can be occurring at a daily or even hourly basis. Furthermore, for those global-level events, even though there were just a few of them, they are extreme events that are important to address even if we do not seem to have enough training data, especially similar ones could occur in the future. Ac-

tually, only a few such events does not mean we do not have enough training data —the amount of BGP data associated with each event is still large.

Also, compared with other possible approaches to analyzing patterns of ABEs, such as manual statistical methods, our approach provides several main functionalities:

- *Automating the rule discovery and detection process.* Most steps we have presented so far are automated. User input is needed only at certain points, such as determining the parameters to use or checking rules produced from the anomaly resolver. Compared to manual data analysis, this implies a faster, more systematic detection methodology and makes online detection possible.

- *Selecting or filtering parameters.* By combining empirical studies and information gain and other mechanisms, parameters that have a significant impact on determining an ABE can be discovered, while parameters with minimal impact can be filtered out.

- *Improving accuracy.* It is likely that sometimes the rules generated are not sensible, especially when the training data is insufficient or not representative. Outliers from normal periods may raise false alarms, too. While user intervention is necessary in certain circumstances, the IRF system can be trained continuously with more data or more parameters, thus further improving the accuracy.

8. RELATED WORK

Research has been conducted to study BGP behavior during unexpected events. Several studies have found that some Internet worms, such as CodeRed II, Nimda, or Slammer, can cause BGP update storms, even though the worms did not directly target BGP routers [8, 9, 16, 31]. Cowie *et al* [7] analyzed the number of affected networks and changes in routing table sizes during the 2003 East Coast electricity blackout. Research also found that minor BGP misconfigurations can add significant overhead to BGP routers [21], and accidental configuration errors could result in large-scale connectivity outages [10, 22].

Researchers has also studied the security of BGP. Researchers have not only surveyed BGP vulnerabilities [24, 6], but also attempted to secure BGP by proposing new protocols [13], developing incremental solutions [11], or addressing specific security issues [4]. In addition, researchers have proposed new functions to enhance the general capabilities of routers, such as [20, 28, 26]. Unfortunately, these improvements all face deployment issues, while the current BGP protocol continues to play a fundamental role in the functioning of the Internet. It is critical to understand, detect and handle ABEs under the *current* environment.

Numerous works have analyzed traffic and events to detect intrusions or other anomalies from the Internet data plane. However, only a few anomaly detection studies have been performed on BGP, and they often just focused on specific anomalies. For example, the study in [34] proposed to enhance BGP protocol to detect IP address ownership violation . Researchers also applied visualization [29] and topology-based [14] techniques to attack the above problem. The more related work to IRF is [33], which used signature-based and statistics-base methods. In particular, the authors used the NIDES statistical algorithm to detect anomalous BGP updates by measuring the statistical deviation of current behavior from a long-term profile. Differently, IRF applies data mining to generate *rules* for detection. The framework offers the flexibility of employing other data mining techniques and the power of detecting a wide range of BGP anomalies.

Data mining has been applied to network intrusion detection. Research in [19, 5] described misuse detection by using standard data mining algorithms to label data as "normal" or "intrusive." Instance-based machine learning [17] and outlier analysis [18] have been used to build models of normal data and detect deviations as anomalies. While these approaches strengthen our confidence that data mining and related techniques can be useful for IRF, they are not directly applicable to IRF. Data for IRF are different not only in their semantics but also in their complexity.

9. CONCLUSIONS

Whereas there are numerous studies on detecting intrusions or other anomalies by investigating traffic from the Internet data plane, the Internet Routing Forensics (IRF) framework presented in this paper provides a new, systematic approach to detecting abnormal BGP events from the control plane, a major concern for the reliability of the Internet routing infrastructure.

Detection of abnormal BGP events must be accurate, applicable to different user needs, and fast if online anomaly detection is required. The IRF framework supports a reliable but flexible process that continuously trains itself to learn detection rules from already known abnormal BGP events, and then uses these rules to detect unknown abnormal BGP events. Albeit a difficult problem, our research shows that a well-designed routing forensics framework can be promising.

In one case study, we found that effective worm-specific classification rules can be obtained by training the system using BGP data from the CodeRed and Nimda periods as well as those from the normal periods. With these rules, unseen BGP data from the Slammer worm period and BGP data from the normal periods can be clearly distinguished from each other. A separate case study on blackout events showed similar results on the efficacy of the IRF framework.

Open issues of this research include a deeper understanding of the implication of rules discovered, and more accurate and efficient designs of the training and detection processes. Also important is the investigation of a broader range of abnormal BGP events with more comprehensive parameters for both global and local level routing information. In particular, we are investigating how to distinguish different abnormal BGP events in addition to distinguishing the abnormal from the normal. We are also studying prefix-level abnormal BGP events that are probably more abundant but at smaller scale than those from our case studies.

10. ACKNOWLEDGMENTS

The authors would like to thank the generous comments from the anonymous reviewers. Their comments have greatly helped improve this research and prepare the camera-ready version of this paper. This research is being funded by the National Science Foundation under Grant CNS-0520326.

11. REFERENCES

[1] Hurricane frances chronology. http://www.fpl.com/storm/contents/hurricane_frances_chronology.shtml.

[2] RIPE routing information service raw data. `http://data.ris.ripe.net/`.

[3] R. Agrawal and R. Skrikant. Fast algorithms for mining association rules. In *Proceedings of 1994 VLDB Conference*, pages 487–499, 1994.

[4] W. Aiello, J. Ioannidis, and P. McDaniel. Origin authentication in interdomain routing. In *Proceedings of the ACM Conference on Computer and Communication Security (CCS)*, 2003.

[5] D. Barbar, J. Couto, S. Jajodia, and N. Wu. Adam: A testbed for exploring the use of data mining in intrusion detection. *SIGMOD Record*, 2001.

[6] S. Convery, D. Cook, and M. Franz. An attack tree for the Border Gateway Protocol. Internet Draft, September 2003. Work in progress.

[7] J. Cowie, A. Ogielski, B. Premore, E. Smith, and T. Underwood. Impact of the 2003 blackouts on Internet communications. Technical report, Renesys, November 2003.

[8] J. Cowie, A. Ogielski, B. Premore, and Y. Yuan. Global routing instabilities during Code Red II and Nimda worm propagation. Technical report, Renesys, 2001.

[9] J. Cowie, A. Ogielski, B. Premore, and Y. Yuan. Internet worms and global routing instabilities. In *Proceedings of SPIE International symposium on Convergence of IT and Communication*, 2002.

[10] J. Fartar. C&W routing instability. http://www.merit.edu/mail.archives/nanog/2001-04/msg00209.html.

[11] G. Goodell, W. Aiello, T. Griffin, J. Ioannidis, P. McDaniel, and A. Rubin. Working around BGP: An incremental approach to improving security and accuracy of interdomain routing. In *Proceedings of Network and Distributed System Security Symposium (NDSS)*, February 2003.

[12] J. Han and M. Kamber. *Data Mining: Concepts and Techniques*. Morgan Kaufmann Publishers, 2001.

[13] S. Kent, C. Lynn, and K. Seo. Secure Border Gateway Protocol (S-BGP). In *Proceedings of Network and Distributed Systems Security Symposium*, 2000.

[14] C. Kruegel, D. Mutz, W. Robertson, and F. Valeur. Topology-based detection of anomalous BGP messages. In *Proceedings of the International Symposium on Recent Advances in Intrusion Detection (RAID)*, September 2003.

[15] C. Labovitz, G. Malan, and F. Jahanian. Internet routing instability. *IEEE/ACM Transactions on Networking*, 6(5):515–528, 1998.

[16] M. Lad, X. Zhao, B. Zhang, D. Massey, and L. Zhang. An analysis of BGP update surge during Slammer attack. In *Proceedings of the International Workshop on Distributed Computing (IWDC)*, 2003.

[17] T. Lane and C. Brodley. Temporal sequence learning and data reduction for anomaly detection. *ACM Transactions on Information and System Security*, 2(3):295–331, 1999.

[18] A. Lazarevic, L. Ertz, V. Kumar, A. Ozgur, and J. Srivastava. A comparative study of anomaly detection schemes in network intrusion detection. In *Proceedings of SDM*, 2003.

[19] W. Lee, S. Stolfo, and K. Mok. Mining in a data-flow environment: Experience in network intrusion detection. In *Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD)*, 1999.

[20] J. Li, J. Mirkovic, M. Wang, P. L. Reiher, and L. Zhang. SAVE: Source address validity enforcement protocol. In *Proceedings of IEEE INFOCOM*, 2002.

[21] R. Mahajan, D. Wetherall, and T. Anderson. Understanding BGP misconfiguration. In *Proceedings of ACM SIGCOMM*, August 2002.

[22] S. Misel. Wow, AS7007! http://www.merit.edu/mail.archives/nanog/1997-04/msg00340.html.

[23] G. Mohay, A. Anderson, B. Collie, and R. M. O. Vel. *Computer and Intrusion Forensics*. Computer Security Series. Artech House Publishers, 2003.

[24] S. Murphy. BGP security vulnerabilities analysis. Internet Draft, June 2003. Work in progress.

[25] J. Quinlan. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers, 1993.

[26] S. Savage, D. Wetherall, A. Karlin, and T. Anderson. Practical network support for IP traceback. In *Proceedings of ACM SIGCOMM*, 2000.

[27] E. Schultz and R. Shumway. *Incident Response: A Strategic Guide to Handling System and Network Security Breaches*. New Riders, 2002.

[28] A. Snoeren, C. Partridge, L. Sanchez, C. Jones, F. Tchakountio, S. Kent, and W. Strayer. Hash-based IP traceback. In *Proceedings of ACM SIGCOMM*, August 2002.

[29] S. Teoh, K. Ma, S. Wu, D. Massey, X. Zhao, D. Pei, L. Wang, L. Zhang, and R. Bush. Visual-based anomaly detection for BGP origin AS change (OASC) events. In *Proceedings of the Distributed Systems, Operations, and Management Workshop*, 2003.

[30] University of Oregon Route Views Project. `http://antc.uoregon.edu/route-views/`.

[31] L. Wang, X. Zhao, D. Pei, R. Bush, D. Massey, A. Mankin, S. Wu, and L. Zhang. Observation and analysis of BGP behavior under stress. In *Proceedings of Internet Measurement Workshop*, November 2002.

[32] Z. Wu, E. S. Purpus, and J. Li. BGP behavior analysis during the August 2003 blackout. In *International Symposium on Integrated Network Management*, 2005. extended abstract.

[33] K. Zhang, A. Yen, S. Wu, L. Zhang, X. Zhao, and D. Massey. On detection of anomalous routing dynamics in BGP. In *Proceedings of the International IFIP-TC6 Networking Conference*, 2004.

[34] X. Zhao, D. Pei, L. Wang, D. Massey, A. Mankin, S. Wu, and L. Zhang. Detection of invalid routing announcement in the Internet. In *Proceedings of International Conference on Dependable Systems and Networks (DSN'02)*, 2002.