

Challenges in Building an Internet-Wide Monitoring Infrastructure for Network Security

Min Gyung Kang
Dept. of Computer Science
KAIST, Korea
mgkang@an.kaist.ac.kr

Sue Moon
Dept. of Computer Science
KAIST, Korea
sbmoon@cs.kaist.ac.kr

Abstract—As new types of network threats push the limit of the firewalls and Intrusion Detection Systems (IDSs), a new form of monitoring systems is emerging: voluntary monitoring systems. Unlike traditional monitoring systems, voluntary monitoring systems rely on a large number of volunteers as an information source for their measurement and monitoring purposes.

In this paper we first articulate the common challenges that voluntary monitoring systems face. These challenges outline how various ideas and features of today's voluntary monitoring systems can be combined and new features should be added for a new Internet-wide monitoring infrastructure. Then we offer a sketch for a new architecture for a voluntary monitoring system that we believe is more scalable and inclusive of available detection algorithms.

I. INTRODUCTION

Ever since the birth of the Internet, network security engineers and researchers have been in a ceaseless security arms race with hackers. The advent of firewalls and Intrusion Detection Systems (IDSs) in the 1990's has alleviated the burden of network operators and managers, but failed to obviate the worry for new attacks and threats. When Distributed Denial of Service (DDoS) attacks assaulted major web sites in early 2000 [1], and various Internet worms hit the Internet, the lack of proper defense mechanisms was felt sorely by researchers and network engineers alike.

These threats set up a new paradigm of network attacks and challenge the conventional systems for network security with novel attack techniques. These new types of threats share several common features. First, they use compromised end hosts, so-called *zombies*, to stage an attack, and make it hard to track down the real identity of attacks. Though the number of compromised zombies is hard to estimate, we fathom, from the frequency and intensity of port and address scanning activities [2], a fairly large number of end hosts compromised at any moment. Second, as many zombies are used in a DDoS attack, they are distributed all over the world. This distributed nature of attack sources renders a traceback of attack sources pointless and blocking suspicious network traffic at the originating source almost impossible. Third, the attacks not only debilitate their victims, but can cause second-hand impact on other users causing network congestion. A recent SQL slammer worm attack is one such example. Lastly, a time interval between a vulnerability

announcement and a code that exploits it is getting shorter. In a myriad of software patches and updates to keep up with, often users turn a blind eye to vulnerability announcements, while one malicious user can wreak havoc for all.

Existing solutions, such as firewalls and IDSs, monitor and control limited ranges of the Internet and can only handle problems within their managed space. Little exists in any form of corroboration or cooperation between existing solutions, and most of the time we do not know what is happening at the scale of the Internet. For example, even if the same new attack is affecting two neighboring Internet Service Providers (ISPs), there is no automated mechanism to share information one has with the other. However, due to administrative, legal, and business restrictions, sharing data may not be feasible between ISPs or any other entities of market competition.

As a response to the current void of systems that offer global views at the Internet scale, a new category of experiments is emerging: voluntary monitoring systems. Unlike traditional monitoring systems, voluntary monitoring systems rely on a large number of "volunteers" as an information source for their measurement and monitoring purposes [3], [4], [5], [6], [7]. A volunteer can be any end host willing to run a piece of code and report to share the results with other or a firewall to export logs. Just as victims and zombies of today's Internet security attacks are widely distributed, these volunteers can be as widely distributed. Thus voluntary monitoring systems can expand the coverage from a single administrative domain to the entire Internet and perform analysis on suspicious network activities emerging on a global scale to warn Internet users in an early stage of the threats.

As voluntary monitoring systems are in an early stage of development and deployment, they have much room for growth both architecturally and in numbers. In this paper we first articulate the common challenges that voluntary monitoring systems face. These challenges outline how various ideas and features of today's voluntary monitoring systems can be combined and new features should be added for a new Internet-wide monitoring infrastructure. Then we offer a sketch for a new architecture for a voluntary

monitoring system that we believe is more scalable and inclusive of available detection algorithms.

In the next section, we review recent work on voluntary monitoring systems and compare features. Today's voluntary monitoring systems, some implemented and others only in design and partial implementations, all face common challenges. In Section III, we outline those challenges in building an Internet-wide monitoring infrastructure for network security, and study how existing systems address them. Motivated on the existing body of work, we design a new monitoring architecture and provide a sketch of it in Section IV. In Section V, we summarize the paper with plans for future work.

II. RELATED WORK

From the information gathered from volunteers scattered widely in the Internet, a voluntary monitoring system can expand its scope to the entire Internet. Several systems exploiting this paradigm of volunteers have been proposed. We first review five systems that recruit volunteers and report on some aspect of Internet-wide network activities. One common feature of the five systems is the dichotomy of volunteers and data collectors. These data collectors play the role of a repository and a monitoring center, analyzing information sent by volunteers, and publishing results. Not all results of the analysis are made public or shared with the volunteers.

DIMES

DIMES (Distributed Internet Measurement & Simulation) [3] is a research project, aimed to study the structure and topology of the Internet. A DIMES server collects measurement data from the agents deployed on volunteers' personal computers (PCs). The agent program monitors the volunteer's Internet connections and tries to discover the path from the volunteer to a destination host using *ping* and *traceroute*. The open-source agent program consumes very low bandwidth and provides the volunteer with graphical presentation of topology of end-to-end connections which fan out from its PC. However, aggregated results, possibly global topological views of the Internet, are not open to public, and it is often not possible for DIMES agents to draw complete paths due to AS border routers blocking outgoing ICMP packets from *ping* and *traceroute*.

DShield

DShield.org [5] is an attempt to collect data about hacker activities from all over the Internet. Anyone can volunteer his firewall or IDS logs and then look up on the web to check if one's IP address is on the blacklist. A volunteer can learn if one's machine has been compromised and may turn into a zombie in a later attack.

All logs from volunteers are gathered at a centralized location and basic statistics, such as top 10 offenders and top 10 most probed ports, are posted on the web at the central location. DShield also provides a suite of conversion tools that

transform local firewall logs into the DShield report message format. The report message can be submitted to the server as an E-mail or through a web reporting form.

Currently, DShield offers simple firewall rules for volunteers to download and use on the system. It expects to set up more sophisticated policies for application level firewalls in the future.

DOMINO

DOMINO (Distributed Overlay for Monitoring InterNet Outbreaks) [4] is an architecture for a distributed intrusion detection system. It is composed of a peer-to-peer overlay network of axis nodes that collect intrusion data from NIDSs, firewalls, and active sinks that monitor inbound traffic sent to a block of unused IP addresses and provide highly reliable data for blacklisting. Axis nodes create blacklists and share them to generate a global view on attackers.

DOMINO takes several security issues into its design consideration. As a proactive measure against threats, authors analyze threat vulnerabilities such as DoS attacks, infiltration, and obfuscation. However, since DOMINO overlay network is not yet deployed, authors present the simulation results based on DShield [5] data which show the effectiveness of blacklists at different sizes and granularities and results of retrospective analysis on several Internet worms.

NETI@home

NETI@home [6] of Georgia Institute of Technology is an open-source software package that collects network performance statistics from end hosts. These performance statistics are sent to a server and publicly released. The main focus of these statistics is various metrics of four transport layer protocols: TCP, UDP, ICMP, and IGMP.

NETI@home project solicits volunteers for Internet-wide deployment of this agent program and its volunteer code supports diverse platforms, including MS Windows, Linux and Solaris. Moreover, users can choose from the three levels of privacy: high, medium, and low. In accordance with this degree setting, some or all part of source and destination IP addresses in the data sent to a NETI server is concealed.

NetBait

Netbait [8] is a PlanetLab [9] service that provides distributed detection of machines infected with Internet worms. It constructs a distributed query tree of NetBait nodes using Tapestry DOLR [10], and a user can obtain information on host infection data by querying on this structure. Each Netbait node detects probe attempts of remote machines infected by Internet worms and can tell which worm infected those machines by comparing their packet payload data with signatures of well-known security vulnerabilities. These signatures also can be disseminated over all the NetBait nodes through distributed query processing.

A prototype of NetBait was implemented based on simplified version of its architecture and now running on PlanetLab test beds. However, distributed query processing

using Tapestry DOLR is not realized on NetBait yet.

The above five systems all have some form of data collectors, different from volunteering end users. There have been other efforts to make the monitoring effort decentralized [11], [7]. Instead of having dedicated systems for data aggregation, the responsibility of building a global view falls on volunteers themselves. Through querying each other's state, volunteers obtain enough information to satisfy their needs. The jury on either of the voluntary monitoring system is still out, as none has reached maturity. In our opinion, they are complimentary, as the more data, simply the better.

As you can see from the above listed projects and systems, many forms of volunteers are persuaded to join the Internet-wide monitoring effort. Heterogeneity of the volunteers and their reports should not hamper this ongoing and growing effort. Already some solutions exist in IDMEF [12] and DShield conversion tools for syntactic compatibility. As various voluntary monitoring systems will be eventually deployed and mature, we hope to see more work to exploit the diversity of volunteers and bridge the semantic gap.

At the moment, systems like those listed above have been only simulated or in early development phase. As they have so much in common, we could envision a more general architecture that encompasses most features of the existing systems.

III. COMMON CHALLENGES OF VOLUNTARY MONITORING SYSTEMS

Expanding the scope of monitoring from a single domain to the entire Internet is an exciting challenge, as well as a major technical undertaking. In this section, we articulate challenges that are common to voluntary monitoring systems, and review how existing systems address them. As the five systems employ the dichotomy of volunteers and data collectors, we call the code that a volunteer runs *agent* and the data collector, *alarm center*, hereafter.

A. Luring Participants

The most important goal of an Internet-wide monitoring infrastructure is to acquire global views of attackers and victims, and the number of contributing volunteers is a key issue. Most of all, it is important to make volunteering beneficial to those who participate. Giving feedback about their systems and news about ongoing attacks and threats is one reward, we think, any voluntary monitoring system should provide. However, even with a reward, today's average users of the Internet are not likely to hear about opportunities for volunteering. Having an order of a few thousand users may be feasible with a stand-alone volunteer code [5], but to reach beyond the circle of technology-savvy users to the general public, incorporating the volunteer code into a popular application is one solution. To increase the pool of volunteers, one may consider turning popular programs, such as mail servers, into volunteers [11]. For example, if we can turn personal firewalls into volunteers,

we immediately gain footage in all those end hosts that use the personal firewalls.

Agents of NETI@home and DIMES present part of collected data in graphical illustration for this purpose, and DShield supports a simple and efficient search interface of blacklists on its web page.

In addition, agent programs should support as diverse platforms as possible. Developing and maintaining agents for multiple platforms will be a major endeavor.

B. Timeliness of Feedback

To take proper countermeasures against malicious network activities, timely feedback on the network activities is necessary. For this purpose, the monitoring infrastructure should run continuously and have feedback in real time. Recent worm and virus spreads have been strikingly fast and most countermeasures would arrive too late for many users. Even some bandwidth-limited worms can infect most vulnerable hosts in a day [13]; it is estimated that a hypothetical Warhol worm can do that in less than 15 minutes [14]. It is impossible to contend with these threats through sluggish analyses and reports of security incidents. The monitoring infrastructure can bring instant feedback about ongoing threats and possible future attacks back to a volunteer's system, one can be more agile in managing security threats which otherwise would already have been damaged one's system. As one form of the feedback, a countermeasure can be distributed along, which would make the typical chores of applying patches and updates more like a push technology. Thus a volunteer can make a more informed decision, as one knows about the possible candidate attacks that inflicted its end system.

C. Scalability

To handle ever-growing number of volunteers and their inputs, the monitoring infrastructure should distribute its load amongst a set of alarm centers. For alarm centers to deal with an overload of volunteers, mechanisms and policies of alarm centers for managing their agent population are important to resolve load balancing issues. When a certain alarm center is overloaded, it should guide agents trying to connect to it to another alarm center which has room to accommodate more agents. We also would like to point out that just as we luring volunteers, we need a strategy to increase the number of alarm centers in a scalable fashion.

When some analysis results are stored on an alarm center and frequently requested by many agents, the alarm center is bound to be a bottleneck of agent queries (NetBait also indicates this possibility). For this reason, alarm centers need to share copies of frequently requested data and keep track of modification of it for data consistency. This archiving option is also as important as agent population management. To remedy this problem, DOMINO defines compact report summaries and shares them among axis nodes which are components directly analogous to alarm centers in this paper.

D. Robustness of Architecture

Like any other public infrastructure, the monitoring infrastructure should operate continuously in a stable manner as an essential security infrastructure. To achieve robustness of the architecture, we need to consider threats from external attackers and system downtime caused by a crash or maintenance work.

As mentioned in DOMINO, DoS and DDoS attacks targeting alarm centers are possible. Moreover, in case the alarm centers have a security vulnerability, it is possible for the whole infrastructure to serve as a hotbed of attacks. In the worst case, the alarm centers must have a shutdown mechanism to mitigate damage on participants caused by these attacks.

E. Accuracy of Feedback

Even if the monitoring system could detect all the network attacks and incidents, they could be worthless if it generated too many false alarms. The same problem has plagued IDS developers from the very beginning, and they have devised various mechanisms from stateful analyses to coordination with other security tools to reduce false alarms. In our case of the monitoring infrastructure, false alarms are more lethal due to the scope of coverage. Therefore, accurate analysis should be a top priority from the very design phase, and it might be better to err on the side of false positives. However, false positives are a deterrent on attracting volunteers, as they are a disappointment to volunteers afflicted by attacks.

IV. SKETCH OF A NEW INTERNET-WIDE MONITORING INFRASTRUCTURE

In this section we lay out design decisions and options in building an infrastructure that addresses the challenging issues by combining features of current related work and other new features. We envision an infrastructure made up of volunteers and aggregation points of data from the volunteers. We begin the discussion on the architecture design with the issue of cooperation amongst those aggregation points, and then we describe functions and requirements of volunteer's agent and the alarm center. At the end of this section, we introduce standards for the infrastructure to be interoperable with diverse security tools and products.

A. Cooperation amongst Alarm Centers

Cooperation amongst alarm centers lies at the core of the proposed Internet-wide monitoring infrastructure. Alarm centers should be able to locate other alarm centers easily, and share intrusion logs and analysis results. Depending on the number of participating alarm center, we can consider two models for the cooperation, a query structure based on Light-weight Directory Access Protocol (LDAP) or a peer-to-peer network.

Communication between alarm centers in the order of thousands may be dealt with LDAP, while if the number

reaches millions, a peer-to-peer network should be a better fit.

Additionally, query structure should be designed to shorten response time for volunteer queries since it is also vital to volunteers' reaction to threats in an early stage of an outbreak. Query and response should be light-weight, and it needs to be taken into design considerations that there is a possibility of centralized query load for only one node in case of distributed queries.

B. Agent

Since diversity of data helps enhancing accuracy of analysis results, the agent should have active mechanisms for collecting data. While collecting data from the volunteers, it needs to provide a conciliatory point, a privacy option, for those who do not want to disclose sensitive part of data. It is also necessary to affiliate with a security research group to provide detail account for incidents and feedback besides convenience of viewing results. Additionally, the agent should be robust and compatible with various OS platforms to lure more volunteers.

Collecting Data: Agent programs can gather end-to-end measurement data and host system usage and information as well as security event logs from security programs installed on volunteers' host machines. Ultimately, some features of *honeypot* can be implemented on agent programs to collect security logs in an active manner like the active sink of DOMINO will do in the future. To illustrate, an agent is able to extract signatures of malicious user's behavior by emulating simple services on host machine's unused ports. (Honeycomb [15] of Christian Kreibich and Jon Crowcroft proposes automated signature generation using this technique. When it is implemented in a 'distributed manner', it is possible to generate more accurate signatures by comparing signatures collected from numerous agents distributed over the infrastructure.)

Privacy Options: Volunteers would be reluctant to submit their security logs and measurement data if some private facts could be inferred from the data. The facts can tell a lot about volunteers: what web sites they frequently visit, what application programs, possibly pirate versions of software, they use, what security vulnerabilities they have, what network servers they are protecting, where they are located, and etc. Out of this private information, IP addresses of volunteers and those in event logs are most sensitive data in a sense of privacy. We can support user options to conceal them in submitted data like NETI@home does, but they are vital information for tracing malicious attackers' trails and correlating events. Thus, in case that volunteers do not want to reveal their IP addresses, some techniques are needed to hide real IP addresses of volunteers and preserve uniqueness of them at the same time. To satisfy this requirement, a configuration option to enable IP anonymization [16] should

be used.

Presenting Data: It is important for agents to support convenient methods for average end-user volunteers to browse data received from alarm centers. Graphical data presentation like graphs and charts is one way to achieve this goal. When reporting specific security incidents and feedback, the agent should also deliver background knowledge (e.g., in a form of help files or hypertext links to web pages) to help understand what is actually happening on the Internet and what preventive measures or countermeasures they can take. Since it is, however, an extremely laborious task to collect and categorize vulnerability information and advisories for it, we need to affiliate with security research groups.

Robustness and OS Compatibility: To recruit more participants, robustness and compatibility with diverse OS platforms are essential requirements of the agent. Inborn bugs of an agent hamper normal operation of collecting data from end-users' machines. Furthermore, repeated crashes or error messages would make the participants exasperated and eventually remove it from their machines. It is an arduous task to develop and maintain robust agents on multiple OS platforms, so the agent should be designed in a simple structure.

C. Alarm Center

To fight ever-evolving and fast-propagating attack techniques, the alarm center should facilitate means to add up new modules for analysis and correlation engines and 'push' quickly urgent information to its volunteers. In addition to information push of the alarm center, query interfaces for both internal and external users are necessary. The alarm center must monitor its own system load to distribute load and execute shutdown mechanisms.

Analysis and Correlation: The most significant capability of alarm centers is to analyze and correlate accumulated event logs in a timely manner. Unlike misuse detection of IDSs and access control of firewalls, this analysis and correlation process of the alarm centers cannot be performed through comparing simple signatures or applying plain policies. In other words, to detect and address new types of network attacks and outbreaks, alarm centers should support more extensible architecture for their analysis and correlation engines. However, current systems carry out relatively simple fixed task such as blacklisting suspicious hosts that perform port probes. In our design options, plug-in architecture of the engines and a script language to describe correlation processes are keys to this issue, capable of adding a new plug-in module or a correlation routine written in the script language when a new network threat rises.

Disseminating Urgent Information: The infrastructure should be able to rapidly 'push' urgent information to volunteers. It is extremely important to distribute security advisories when a new security vulnerability is released,

or malwares start to spread. It is shown that it is effective to contain propagation of worms using content filtering in the early stage of an outbreak. [17] Provided that this content filtering information, a kind of IDS signature or a policy of firewall, is disseminated through the infrastructure promptly, propagation speed can be slowed down to give network administrators and end users more time to take countermeasures at least.

Query Interface: Not only should alarm centers provide query interfaces for the internal volunteers but also be equipped with external query interfaces for the public. Query power and resolution might not be as powerful as those for the volunteers, but it is an efficient method for advertising the infrastructure by giving people some rudimentary information and a cooperation channel with other network infrastructures like Network Oracle[18].

Assessing Load Metrics: To distribute processing load of alarm centers, the alarm center should be assessing system load metrics to monitor how much load they are taking and decide when they should 'split' volunteers and replicate most-wanted data to other alarm centers. Usage of CPU, memory, and network bandwidth and cumulative number of queries requesting specific data are such metrics. Interestingly, alarm center can detect anomalous symptoms through statistical analysis on accumulated metric data.

D. Standardization for Interoperability

To facilitate efficient exchange and processing of data from diverse security tools and products, it is necessary to establish standard data formats, common vulnerability IDs, and communication protocols. Current voluntary monitoring systems are mainly focusing on standard message formats such as IDMEF, but communication protocol to deliver the messages and semantic compatibility of message content also should be considered.

As regarding to the standard data formats, several Internet drafts are proposed for reporting Internet measurement and intrusion detection data. Preliminary Measurement Spec for Internet Routers [19] of CAIDA defines some measurement metrics for router traffic, and IDMEF [12] of IETF specifies an exchange format of intrusion detection logs in XML. Aggregated data and analysis results also should be easily accessible by other monitoring infrastructures like DOMINO or Network Telescope [20] as well as individual participants through standard data formats.

In case of reporting intrusion detection logs, it is useless effort to collect intrusion logs written in different 'languages'. Each vendor of information security products keeps its own vulnerability database, and logs produced by these products have vulnerability IDs (or names) from this database. However, to put logs from diverse security tools

into automated analysis and correlation process, it is essential to standardize these vulnerability IDs. Fortunately, MITRE provides CVE (Common Vulnerabilities & Exposures) [21] database for security product vendors and tools writers for this purpose and, hence, adopting this kind of common vulnerability ID will help reducing load of unifying intrusion event logs.

To exchange intrusion detection logs, mutual-authentication, integrity, and confidentiality over a connection-oriented protocol are essential requirements for a data exchange protocol of the infrastructure. In addition, a flexible feature of protocol is necessary to communicate with end-users' agents residing in networks protected by firewalls through tunneling. To meet these requirements, Intrusion Detection eXchange Protocol (IDXP) [22] is proposed by IETF using a Blocks Extensible Exchange Protocol (BEEP) [23] profile, and it will be one of the options for a data exchange protocol.

V. SUMMARY AND FUTURE WORK

We have outlined challenging problems in building an Internet-wide monitoring infrastructure for network security. To fight new types of network threats which are capable of inflicting extensive damage on the Internet in a short time, we should devise a monitoring infrastructure which are beneficial enough to lure more participants, prompt to provide accurate and timely reports, robust against internal and external threats, and highly scalable to manage numerous volunteers and data archives. To resolve these challenges, we also have presented design decisions and options including cooperation amongst alarm centers and standardization requirement.

Currently we are building a volunteer-based monitoring system based on architectural sketch presented in this paper, and we plan to deploy and run it to maturity. In the future, we also want to support interfaces to work in collaboration with more general network infrastructures such as Network Oracle [18].

ACKNOWLEDGMENT

We are grateful to Timothy Roscoe of Intel Research for his invaluable advice and feedback on this paper.

REFERENCES

- [1] R. Comerford, "No longer in denial," *IEEE Spectrum*, vol. 38, pp. 59–61, Jan. 2001.
- [2] V. Yegneswaran, P. Barford, and J. Ullrich, "Internet intrusions: Global characteristics and prevalence," In Proceedings of ACM SIGMETRICS, June 2003.
- [3] (2003) DIMES. [Online]. Available: <http://www.netdimes.org/>
- [4] V. Yegneswaran, P. Barford, and S. Jha, "Global intrusion detection in the domino overlay system," In Proceedings of NDSS 2004, 2004.
- [5] (2004) DShield - distributed intrusion detection system, the internet's early warning system and internet security community site. [Online]. Available: <http://www.dshield.org/>
- [6] (2004) NETI@home. [Online]. Available: <http://www.neti.gatech.edu>
- [7] S. Srinivasan and E. Zegura, "M-coop: a scalable infrastructure for network measurement," in *Proceedings the Third IEEE Workshop on Internet Applications. WIAPP 2003*, June 2003, pp. 35–39.
- [8] J. L. Brent N. Chun and H. Weatherspoon, *Netbait: a Distributed Worm Detection Service*. Technical Report IRB-TR-03-033: Intel Research Berkeley, 2003.
- [9] (2004) PlanetLab Home Page. [Online]. Available: <http://www.planetlab.org/>
- [10] B. Y. Zhao, J. D. Kubiatowicz, and A. D. Joseph, *Tapestry: An infrastructure for fault-tolerant wide-area location and routing*. Technical Report CSD-01-1141: University of California, Berkeley, Computer Science Division, 2000.
- [11] R. Huebsch, J. M. Hellerstein, N. L. Boon, T. Loo, S. Shenker, and I. Stoica, "Querying the internet with PIER," In Proceedings of 19th International Conference on Very Large Databases, 2003.
- [12] D. Curry and H. Debar. (2004, Feb.) The intrusion detection message exchange format. [Online]. Available: <http://www.ietf.org/internet-drafts/draft-ietf-idwg-idmef-xml-11.txt>
- [13] C. Shannon and D. Moore, "The spread of the Witty Worm," 2004. [Online]. Available: <http://www.caida.org/analysis/security/witty/>
- [14] V. Paxson, S. Staniford, and N. Weaver, "How to Own the internet in your spare time."
- [15] C. Kreibich and J. Crowcroft, "Honeycomb - creating intrusion detection signatures using honeypots," Proceedings of 2nd Workshop on Hot Topics in Networks (HotNets-II), 2003.
- [16] J. Xu, J. Fan, M. Ammar, and S. Moon, "Prefix-preserving IP address anonymization: Measurement-based security evaluation and a new cryptography-based scheme," Proceedings of ICNP, 2002.
- [17] D. Moore, C. Shannon, G. Voelker, and S. Savage, "Internet quarantine: requirements for containing self-propagating code," *Proceedings of INFOCOM 2003*, vol. 3, pp. 1901–1910, Mar. 2003.
- [18] J. M. Hellerstein, V. Paxson, L. Peterson, T. Roscoe, S. Shenker, and D. Wetherall, "Network Oracle," [In submission], 2004.
- [19] CAIDA, "Preliminary measurement spec for internet routers," June 2002. [Online]. Available: <http://www.caida.org/tools/measurement/measurementspec/>
- [20] D. Moore, "Network Telescopes: Observing small or distant security events," Oct. 2002. [Online]. Available: http://www.caida.org/outreach/presentations/2002/usenix_sec/
- [21] (2004) About CVE. [Online]. Available: <http://www.cve.mitre.org/about/>
- [22] B. S. Feinstein, G. A. Matthews, and J. C. C. White, "The intrusion detection exchange protocol (IDXP)," IETF Draft, 2002.
- [23] M. T. Rose, "The blocks extensible exchange protocol core," RFC 3080, Mar. 2001.