# Leveraging Organizational Etiquette to Improve Internet Security

Niklas Carlsson
University of Calgary
niklas.carlsson@ucalgary.ca

Martin Arlitt
HP Labs and University of Calgary
martin.arlitt@hp.com

***Abstract*—As more and more organizations rely on the Internet for their daily operation, Internet security becomes increasingly critical. Unfortunately, the vast resources available on the Internet are attracting many malicious users and organizations, including organized crime syndicates. With such organizations disguising their activity by operating from the machines owned and operated by legitimate organizations, we argue that responsible organizations could improve overall Internet security by strengthening their own. By improving their *organizational etiquette*, legitimate organizations will make it more difficult for malicious users and organizations to hide. Towards this goal, we propose a system to identify and eliminate malicious activity on edge networks. We use a year-long trace of activity from an edge network to characterize the malicious activity at an edge network and demonstrate the potential effectiveness of our system.**

## I. Introduction

Improving systems and network security has been an important topic since the Internet was created. Organizations (such as Enterprises, Internet Service Providers, Universities, etc.) connected to the Internet face a continuous battle for control of their IT assets. As organizations increasingly rely on the Internet for their daily operation, Internet security will become even more critical. Improved security could also lower the carbon footprint of the Internet, by removing non-productive or malicious workloads [12].

It is a challenge for organizations to protect their IT assets. Internet crime is becoming more prevalent and better organized. Malicious users and organizations ("miscreants") use increasingly sophisticated techniques, and leverage geographical and legal boundaries to their advantage. A weakness in current Internet security practices is the primary focus on what others are doing to your resources, rather than giving equal consideration to what your resources are doing to others. We argue that responsible organizations must strive to improve their *organizational etiquette*; i.e., reduce the negative impact the machines (and users) on its domain(s) have on other organizations. As part of this goal, organizations should also help other (trusted) organizations achieve the same goal, primarily through systematic sharing of useful information.

In this paper, we propose a system, called the OE (after "Organizational Etiquette"), for reducing the negative impact an organization and its machines may have on others. We discuss how OE can systematically identify and eliminate malicious activity on edge networks, and exchange non-sensitive information to enable other organizations running OE to achieve the same goal. A key argument of our design is that organizations need to take greater responsibility for the traffic that leaves their edge network(s) for other destinations on the Internet. The design is based on the premise that "security rests on host accountability" [18] and includes a novel solution which allows machines to be moved between different security planes, with different levels of Internet accessibility.

Our system builds upon the key insights that improving organizational etiquette will make the Internet more secure, and that non-negligible improvements could be obtained by following five simple rules: don't attack, don't scan, don't intrude, don't infect, and don't spam. Using a year-long trace of an edge network's traffic, we demonstrate how such a system could reduce the volume of malicious or non-productive traffic that leaves the edge network, as well as improve the security of the edge network itself. We characterize different types of undesirable activity and introduce specific solutions to these activities, and quantify the potential effectiveness of our proposed solution. Our characterization also considers how miscreants have achieved their current levels of success, and use those insights to make it more difficult for miscreants to achieve their various goals in the future.

While the qualitative benefits of improving local security and information sharing are intuitive, little progress has been made on designing a solution. A contribution in this paper is we quantify the benefits of our proposed solution using extensive measurements from a network of a large organization. Although we evaluate a single edge network, Metcalfe's Law ("the value of the system grows at approximately the square of the number of users of the system"), indicates that improved etiquette and sharing of information across a set of organizations would have a much greater positive effect on overall Internet security.

The remainder of the paper is organized as follows. Section II describes the system design of OE. Section III provides a motivating case study. Section IV provides a characterization of the malicious (or non-productive) traffic and demonstrates the effectiveness of OE. Section V discusses related work. Lastly, Section VI summarizes our contributions.

## II. Our Proposed Method

There is an adage that *you cannot manage what you cannot measure*. Unfortunately, this reflects the state of many edge networks today. While the Internet has seen significant changes

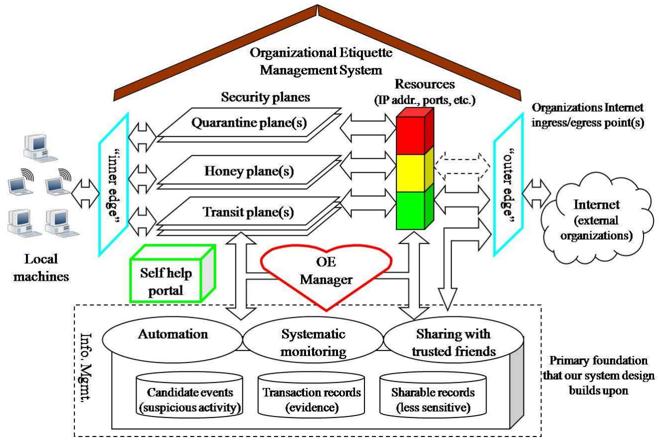


Fig. 1. Organizational Etiquette (OE): A system to help organizations become more responsible Internet citizens.

over time, the management of edge networks has transformed very slowly and conservatively. Many tasks are still done manually, which limits the number of events that can be acted upon. Miscreants are aware of this, and use this to their advantage. In particular, miscreants effectively leverage automation to achieve their goals, while network managers typically do not; this needs to change.

We propose a conceptually simple and intuitive system, the OE, that could significantly improve the security of an edge network, and if adopted more broadly, of the Internet as a whole. Our system consists of three primary components:

- **Information management:** Actionable information is critical for improving security. To gather reliable information, our system relies on systematic monitoring, automatic information processing, and sharing with trusted organizations.
- **Security planes:** Our solution relies on machines easily being moved between different security planes, potentially with different Internet accessibility and/or security restrictions. We envision the security planes being implemented as isolated virtual networks.
- **OE manager:** The OE manager is responsible for taking the information provided through the information management process and applying policies to determine which plane (or security restrictions) each machine on the network should be assigned, at each point in time.

An overarching goal of our design is to automate as much of the system operation as possible, including data gathering, processing, and system management. In the following we describe each of these components separately. An overview of our system is provided in Figure 1.

### A. Information management

The information management component of our system maintains event-based activity records. The information is gathered through systematic monitoring and information sharing, and is automatically analyzed for immediate threats, misbehaving machines, and suspicious events. The system also retains data for evidence reports and/or post-processing that may help identify machines under control (and/or under attack) by stealthier miscreants. The information management component leverage three design components.

**Systematic monitoring:** As with intrusion detection systems, systematic monitoring of traffic at strategic points in the network is essential. In our work we assume that the organization, in our case a university, can monitor the traffic at the "outer edge", just before traffic heads to the Internet, and at the "inner edge", just as it is sent from the local hosts.

**Automation:** With large volumes of traffic, automation of mundane tasks is needed to deal with events in a timely manner. We note that the monitoring system is essential in facilitating automated management, and quickly identifying and resolving problems. For this purpose, policies to identify suspect traffic must be sufficiently simple that they do not cause too much overhead on the monitoring system.

**Sharing with friends:** Organizations can help each other by sharing non-sensitive but useful knowledge with specific, trusted organizations ("friends"). For example, an organization $X$ can inform trusted organization $Y$ that at time $T$, a machine with IP address $Y_i$ sent a spam message to the mail server of $X$. Organization $Y$ need not believe organization $X$, as it can corroborate that "hint" with its own logs, to verify/refute if there was any suspicious message sent from $Y_i$ at time $T$. This information can help other "ethical" organizations reduce their negative impact on the Internet.

### B. Security Planes

Since some organizations will have "mission critical" services on their network, the OE system would not require all systems to be part of the automated management. However, we believe that most Internet security issues today occur with non-critical systems, which account for many more, often self-administered, systems. These systems would be subject to automatic quarantining. Multiple "planes" in the network are required, each implemented as an isolated virtual network, either using VLANs [5] (supported by most modern networking equipment), or via other mechanisms which may offer greater functionality or scalability (e.g., [4]). At a minimum, we believe such a system should have: a *transit* plane for normal traffic, a *management* plane for dynamically reconfiguring the IT environment, a *quarantine* plane to enable affected users to repair affected hosts, and a *honeynet* plane, to allow certain suspicious activities to continue without affecting other hosts until the problem(s) can be identified (at which time the machine may be moved to the quarantine plane).

### C. OE Manager

Various policies can be implemented to determine which plane to (re)assign each machine to. The OE manager makes these policy decisions based on the information provided through the information management component of the system. To help individual clients improve their security so that they can be moved to planes with greater accessibility without requiring increased manual efforts, the OE manager can leverage additional services, such as our "self help" component.

Below, we discuss how the OE manager can identify suspicious traffic in the transit plane. The machines responsible for this traffic can then be moved to either the quarantine or honeynet plane, depending on the expected risks. With false positives being unavoidable, we note that the usage of planes can reduce the cost associated with the affected machines. Systems that would have "expensive" false positives (e.g., mission critical applications) would be addressed in the traditional manual fashion.

**Self-help service:** To avert a substantial increase in support desk workloads, the affected systems would not lose network access, but would have reduced capability. Specifically, they could access the self-help system, but no other system or network; the organization may have to host a repository of patches to enable this. A self-help service would inform users why they have been moved off the transit plane, and enable them to quickly repair (and verify to the management system) their system to get back on the transit plane.

**Management of essential resources:** Current network environments tend to be very static; e.g., they block ports to prevent certain types of activity from happening. However, static policies can be worked around, and can actually make things easier for miscreants. Blanket policies can penalize all users on an edge network, rather than just the offending parties. To ensure that only users (or hosts) responsible for the malicious activity are affected, organizations need to manage essential resources, such as IP addresses, more closely.

**Threshold-based policies:** While OE is not dependent on any specific set of policies to automatically detect and identify local machines from which malicious traffic is originating, we show that relatively simple threshold-based policies (that leverage knowledge of which essential resources should be in use, for example) are able to detect the machines responsible for much of the malicious traffic.

## III. Understanding Miscreants

### A. Our Data Set

We collected a year-long trace of network activity at the University of Calgary. The university has a 400 Mbps full-duplex link to the Internet. We use the `conn` feature of the open-source Intrusion Detection System `bro`[1] to collect summary information on each connection that traverses the link. These summaries include information such as the IP addresses of the source and destination of the connection, the port numbers, and the end-state of the connection.[2]

For this study we analyzed data for the period April 1, 2008-March 31, 2009. Our trace contains 39.3 billion connection summaries. 52% of these connections were *outbound*. i.e., initiated from a host on campus and intended for a host elsewhere on the Internet. 48% of connections were *inbound*.

### B. A Case Study

This section provides a case study that illustrates the operation of a miscreant (or a set of miscreants) as observed from the campus network. This real-world example shows how malicious users target and use other organizations machines for extended periods of time, how seamlessly and systematically malicious systems are operated, and how big an impact a few compromised machines can have on overall Internet security.

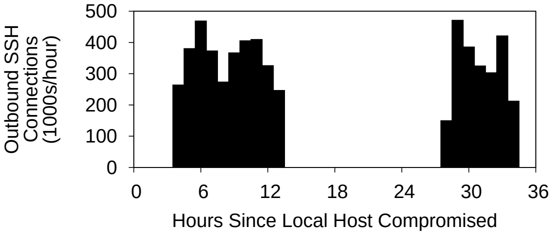


Fig. 2. Scans by a compromised machine on campus.

Our case study starts with a host on the university network being scanned at 10:23pm one Sunday night. From this single contact, the scanner (a host in South Korea) was able to determine that the local host was running an SSH server. Several hours later, the same external host returned, and began a password attack on the local host. Only 6 minutes and 258 connections later, the miscreant had successfully accessed an account on the local system.

About four hours later, a host from Germany logged into the compromised host. It uploaded some data to the local host, which subsequently downloaded data (via `ftp`) from a server in Virginia, followed by data (via `http`) from a server in the Netherlands. The local host then began scanning external IP addresses, searching for other SSH servers. Figure 2 shows the network behavior of the local host following its compromise. After 10 hours of scanning, the local host completed its assigned workload. It then pinged a host in California. Less than one hour later, the original scanner returned, to test more passwords on the compromised host. Monday evening, the local host pinged several additional hosts in the U.S. At 5:30am Tuesday, the host from Germany logged in once more, and the local host resumed scanning external hosts.

On Tuesday afternoon, the scanning activity by the local host was noticed by campus IT, and the host was disconnected from the network. The local host remained offline for two months, until it was repaired and reattached to the network. Interestingly, three hosts, one in Romania, one in Japan, and a third in South Korea, continued to probe for the compromised local host periodically during this two month period. Within two weeks of the local host coming back online, these machines noticed, and the host from Germany soon attempted to login again. When it determined that it could no longer login, the connection attempts stopped.

### C. Discussion of Organizational "Don'ts"

The malicious activity is (typically) not the intention of the local organization or its users, but rather of miscreants using compromised machines. Thus, looking for evidence of malicious activity can help identify the compromised machines, enabling them to be removed from the network, thus improving Internet security. Overall, our system is designed to help organizations become better Internet citizens.

[1] http://www.bro-ids.org/

[2] The connection states are described at http://tinyurl.com/bro-conns.

As discussed in Section I, the Internet would be more secure if edge networks applied the following intuitive rules: don't attack, don't scan, don't intrude, don't infect, and don't spam. The above case study provides examples of how "local" machines (either at the university, or within some other organization) attack, scan, and control/intrude on other organizations.

In our case study, the local organization acted responsibly by identifying and removing a compromised machine from the network. However, the process was largely manual, and over 5.6 million SSH scans still occurred. By automatically searching for "brute force" activities and quarantining the responsible local hosts, over 99% of the SSH scans in the year long data set could have been removed from the network. Few false positives would be possible, as most of the scans were conducted by only a handful of local machines.

## IV. Characterization and Results

In this section we provide additional empirical examples of miscreant behavior, and most importantly, we leverage these to demonstrate how effectively our proposed system could work, without limiting the actions of the majority of users.

### *A. DDoS Attacks*

A commonly cited malicious activity is a Distributed Denial of Service (DDoS) attack [15]. These are commonly launched via a *botnet* - a set of compromised hosts (termed *bots*) remotely managed by a single *controller*, often via a set of intermediaries to improve the scalability of the botnet management. To make it difficult for the target to understand where the attack originated from, the bots typically spoof the source IP addresses. This works because the bots do not intend to establish the connection, just prevent the target from establishing valid connections with other hosts.

A best practice for mitigating DDoS attacks is for edge networks to use *egress* filtering [15]. With this technique, the edge network can drop outbound packets that do not have a source IP address associated with the local organization. As this best practice is used by the university, we originally did not anticipate observing any DDoS attacks. Furthermore, we expected that all source IP addresses on outbound connections would belong to an active host. However, these proved to be incorrect assumptions, and serve as motivation for several attributes of our proposed system (e.g., continuous monitoring, systematic control of resources like IP addresses).

Figure 3(a) provides initial evidence of DDoS attacks. It shows that at least 1,000 (outbound) connections originating from every possible IP address in the university's /16 network (65,536 possible addresses). We see that that slightly less than $2^{15}$ hosts (or half the IP addresses assigned to the university) had successfully established connections over the course of the year. This indicates that the remaining (36,855) IP addresses were used either inadvertently or inappropriately.

Figure 3(b) provides more compelling evidence of DDoS incidents. This shows the six incidents involving the most local IP addresses, all targeted at single external IP addresses. Two of the events happen almost simultaneously in July, and quickly use the majority of the 65,536 possible local IP addresses. In August, three more significant DDoS events occurred almost simultaneously. The sixth event is somewhat different from the others, in that it takes a much longer duration of time to use most of the local IP addresses. The most intense of these attacks generated over 5 million "connections" (in this case, only involving single packets).

**Threshold-based policy:** A straightforward policy for detecting such events is to monitor unused (local) IP address space. This is a similar approach to backscatter analysis [13], but for traffic originating locally. Since the DDoS attack selects the source address randomly from the local IP address range, the probability of selecting an address from a given range can be calculated. For a /24 prefix within a /16 IPv4 address space, the probability is 1/256. Thus, we could potentially mitigate the local participation in a DDoS attack within the first few hundred packets, as without using knowledge of which local addresses are in use, the bot(s) will quickly wander into "darkness". Using this method, we observed 48,294 potential DDoS attacks that local addresses participated in, over the one year period. These attack where distributed across 6,808 external IP addresses (or 3,724 external /24 prefixes). Figure 3(c) shows the number of connections from local machines associated with each of these potential DDoS events. Clearly, there were some external machines that had to endure repeated attacks.

Figure 3(d) shows the fraction of connections that could have been avoided if the remainder of the local DDoS traffic was eliminated once the first unused local addresses had been observed, as well as the number of local IP addresses observed before detection. Note that almost all connections in DDoS events could be eliminated using this simple policy.

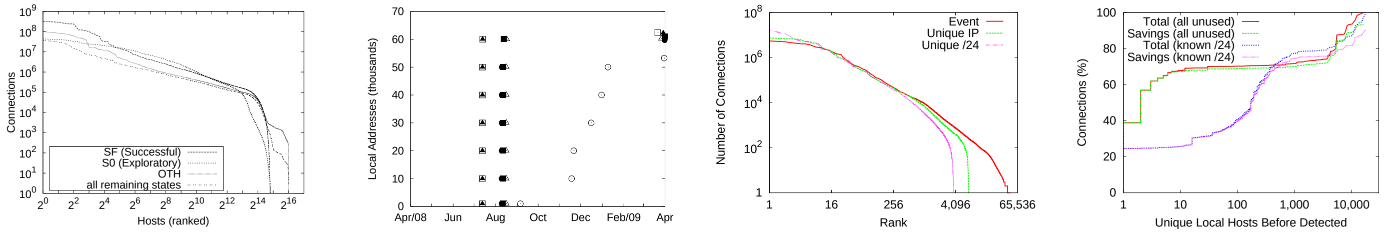
**Management of essential resources:** Relying on a static darknet to detect such events is not a desirable strategy, as miscreants likely would quickly adapt to it, as they have with the egress filtering. A more sophisticated solution would involve continuously monitoring the pool of inactive IP addresses, and filtering based on that. With this approach, the probability of quickly observing local participation in a DDoS event increases substantially.

A risk with automatically filtering all traffic to "targets" identified in this manner is that miscreants could use it to deny local users access to external services. Thus, the preferred solution is to have the network devices at the "inner edge" eliminate packets that were not sent from an authenticated host. An additional reason to do this is removing the extraneous traffic from the network will make it easier to identify other anomalous activities on the network.

### *B. Scans and Intrusions*

As seen in Section III-B, scanning and intrusions often go hand-in-hand. First, a machine scans targets at one or more locations. The scan may search for specific vulnerabilities or weaknesses. When a vulnerability is found, the miscreant may attempt to exploit it directly, or hand it to another miscreant.

During the one year measurement period, local hosts initiated nearly 20 million SSH connections to external hosts.



(a) Connection states (b) Example events (c) Rank plot (d) Potential savings

Fig. 3. Evidence of DDoS activity.

About 3% of these were from "inactive" local addresses. The bulk of the attempts came from a handful of hosts, the compromised host described earlier being one (5.6M). By monitoring/searching for automated activity and "brute force" behavior, it is possible to eliminate the bulk of this activity without any significant side effects. A more effective policy may be an opt-in policy, so that users needing SSH or other network services must register to gain access. As part of the registration process, best-practices could be enforced, to minimize the chances of the host being compromised.

Figure 4 shows the number of SSH connections originating from local machines. We note that 16 machines are accountable for close to 90% of all the SSH connections; each of these local hosts are responsible for at least 100,000 SSH connections. As motivated by the 3% of connections generated from "inactive" local addresses, we note that the flat tail of the rank distribution likely is due to spoofed local addresses.

### C. Infections

Infectious software like worms have strong similarities to the scanning and intrusions/infiltrations described in Section IV-B. A worm typically tries to quickly propagate itself, and may combine the scanning and infiltration steps into one. An underlying trait of most worm propagations is ignorance of the local traffic communication patterns. As such, they will likely "discover" a lot of new IP addresses (or /24 prefixes).

Figure 5(a) shows the cumulative number of distinct /24 prefixes observed over the measurement period. After an initial "warm up" period (e.g., the first week of the trace), the growth rate remains relatively stable through the next five months across the inbound, outbound, and combined counts. When students returned to campus in September, the volume of traffic increases, which results in an increase in unique /24 prefixes observed in the outbound direction. Figure 5(a) also shows two periods of anomalous behavior. In mid-to-late November, and again in late January, there are rapid increases in the number of distinct /24 prefixes seen on outbound connections. Both anomalies are due to worms attempting to propagate.

Figure 5(b) shows rank plots of the number of external IP addresses discovered by local hosts, over the full year and during the "busy" period in mid-November, respectively. 16 local hosts made 40% of the total discoveries over the full year, and 90% of the "busy period" discoveries. This skewed behavior can be used to quickly identify infected hosts.

As worms typically attempt to propagate very quickly, it is necessary to act as soon as infected machines are identified [16]. About 750 local hosts discovered 80% of the external IP addresses seen over the year. A threshold-based policy that flags machines that discover "new" subnets (or IP addresses) faster than some threshold rate could quarantine the worm propagations rather quickly, before the worms can spread broadly. Whitelisting could be used as well, to prevent important services (e.g., email) from being automatically quarantined if they trigger a threshold.

### D. Spam

Spam is a miscreant activity that relies on low cost, automation, and brute force. The earlier in the delivery lifecycle [8] we can eliminate spam, the more we can hinder its effectiveness. Strategies such as blocklists and taking botnet controllers out of service [12] are reasonably effective, but are overcome through automation and brute force. Making it more difficult for bots to send spam or dramatically shortening the lifespan of bots would be more effective. At a minimum, these approaches would complement existing techniques.

As with the other types of miscreant prevention discussed in this paper, our approach is to prevent illegitimate email clients from delivering any messages. A part of the solution is to only allow outbound SMTP connections for registered email servers. Other attempts should not be blocked though, as the communication may help identify compromised machines. Thus, with OE, the suspected bots (based on any attempt to communicate with a recognized external SMTP server) should be automatically moved to the honeynet plane, so that their messages could be collected as evidence.

Our analysis suggests that there are numerous infected computers on the wireless and residential prefixes, which are self-administered machines. For example, of the 3,000 local IP addresses for which we observed successful transactions with known SMTP servers at Google, Microsoft and Yahoo!, 71% of these local addresses were from the wireless and residence prefixes, and are not legitimate university mail servers.

## V. Related Work

Intrusion detection and prevention systems aid organizations in identifying miscreant activity. Research on automated diagnosis tools for performance, fault or security incidents are also relevant [7], [11], [2]. While we leverage insights from such works, our work is novel in that we focus on locally initiated activities, and our proposed system allow machines to be automatically moved between planes with different security levels. With carefully defined policies, our system

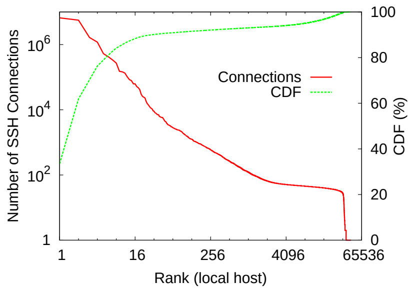
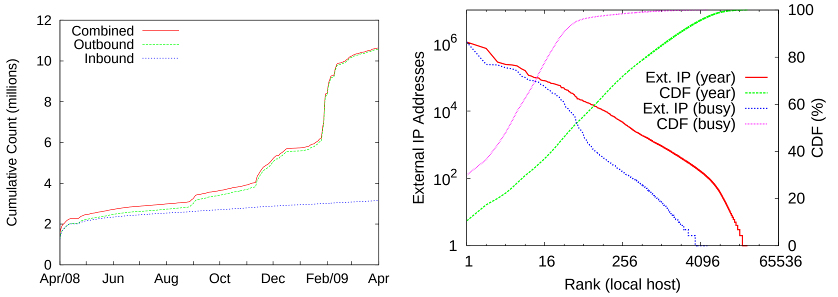


Fig. 4. Rank plot of SSH connections.



(a) New /24 discoveries

(b) Breakdown per event

Fig. 5. Discovery of new IP addresses and /24 prefixes.

can respond to all known types of miscreant activity without affecting critical systems. Our system would use existing virtual network implementations (e.g., [5] or [4]).

While the qualitative benefits of improving local security and increasing information sharing are intuitive and have been proposed by others (e.g.,[10]), little progress has been made on designing a solution. Our approach attempts to rectify this.

The need to better control unused IP space is motivated by characterizations of non-productive traffic. Pang *et al.* [14] monitored unused IP address space to characterize "Internet background radiation", and Jin *et al.* [6] examined *gray space* on a local network to identify and characterize scanners.

Other researchers have measured and characterized various aspects of the threats on the Internet. Allman *et al.* [1] conducted a longitudinal study of third-party scanning, investigating the onset of scanning, scanning frequency, and scanned services over a 12-year period. Zou *et al.* [19] and Weaver *et al.* [17] examine methods for detecting and containing worms. Barford and Yegneswaran [3] and Karasaridis *et al.* [9] explore properties of large-scale botnets. None of these works presents a solution that systematically identifies and automatically eliminates malicious activity on edge networks.

## VI. Conclusion

In this paper we described a system for addressing undesirable network activity occurring within a local organization. By identifying miscreant activity and quickly eliminating compromised machines, overall Internet security could be improved.

Our solution relies on systematically detecting anomalous behavior of local machines and quickly revoking their ability to transit traffic to other hosts. We use a one-year long data set to characterize five types of undesirable activity and introduce specific policy solutions to these activities. While these (and other) policy solutions may address current miscreant activity, more advanced policies will be needed as miscreants are forced to be a lot stealthier. Of course, OE will work with more complex policies too.

By using different planes (with different access levels), we are able to reduce the cost of false positives. While false positives associated with automation often cause concerns for network managers, we note that not all false positives are "expensive". In addition, for any system that would have expensive false positives (e.g., mission critical applications), our solution still allows manual actions to be taken.

A second concern about increasing automation on edge networks is that if done incorrectly, it could make it easier for miscreants to gain control of large numbers of resources on a particular edge network. However, steps can be taken to minimize such opportunities. This is left for future work.

## Acknowledgment

This work was supported by the Informatics Circle of Research Excellence (iCORE) in the Province of Alberta.

## References

[1] M. Allman, V. Paxson and J. Terrell. A brief history of scanning. *IMC*, 2007.

[2] M. Allman, C. Kreibich, V. Paxson, R. Sommer and N. Weaver. Principles for developing comprehensive network visibility. *HotSec*, 2008.

[3] P. Barford and V. Yegneswaran. An inside look at botnets. *Workshop on Malware Det., Adv. in Info. Sec.*, 2006.

[4] A. Edwards, A. Fischer, and A. Lain. Diverter: A New Approach to Networking Within Virtualized Infrastructures. *WREN*, 2009.

[5] IEEE. Virtual Bridged Local Area Networks. *Technical Report ISBN 0-7381-3662-X*, IEEE, 2003.

[6] Y. Jin, G. Simon, K. Xu, Z. Zhang, and V. Kumar. Gray's anatomy: Dissecting scanning activities using IP gray space analysis. *SysML*, 2007.

[7] S Kandula, R. Mahajan, P. Verkaik, S. Agarwal, J. Padhye, and P. Bahl. Detailed Diagnosis in Enterprise Networks *ACM SIGCOMM*, 2009.

[8] C. Kanich, C. Kreibich, K. Levchenko, B. Enright, G. Voelker, V. Paxson and S. Savage, Spamalytics: An Empirical Analysis of Spam Marketing Conversion, *ACM CCS*, 2008.

[9] A. Karasaridis, B. Rexroad, and D. Hoeflin. Wide-scale botnet detection and characterization. *HotBots*, 2007.

[10] S. Katti, B. Krishnamurthy, and D. Katabi. Collaborating against common enemies. *IMC*, 2005.

[11] A. Mahimkar, Z. Ge, A. Shaikh, J. Wang, J. Yates, Y. Zhang, and Q. Zhao. Towards Automated Performance Diagnosis in a Large IPTV Network, *ACM SIGCOMM*, 2009.

[12] McAfee and ICF International, "The Carbon Footprint of Email Spam Report", 2009.

[13] D. Moore, G. Voelker and S. Savage, Inferring Internet Denial-of-Service Activity, *USENIX Security*, 2001.

[14] R. Pang, V. Yegneswaran, P. Barford, V. Paxson, and L. Peterson. Characteristics of Internet background radiation. *IMC*, 2004.

[15] S. Specht and R. Lee. Distributed denial of service: Taxonomies of attacks, tools and countermeasures. *PDCS*, 2004.

[16] S. Staniford, V. Paxson and N. Weaver. How to own the Internet in your spare time. *USENIX-SS*, 2002.

[17] N. Weaver, S. Staniford, and V. Paxson. Very fast containment of scanning worms. *USENIX-SS*, 2004.

[18] Y. Xie, F. Yu and M. Abadi, De-anonymizing the Internet Using Unreliable IDs, *ACM SIGCOMM*, 2009.

[19] C. Zou, W. Gong, D. Towsley, and L. Gao. The monitoring and early detection of internet worms. *IEEE/ACM ToN*, 2005.