Blueprints for a Large-Scale Early Warning System

Conference Paper · October 2010

DOI: 10.1109/PCI.2010.27 · Source: IEEE Xplore

8 authors, including:

Paul Spirakis
University of Liverpool
415 PUBLICATIONS **3,520** CITATIONS
SEE PROFILE

Vasileios Vlachos
Technological Educational Institute of Thessaly
47 PUBLICATIONS **317** CITATIONS
SEE PROFILE

Vassilios Karakoidas
Athens University of Economics and Business
35 PUBLICATIONS **191** CITATIONS
SEE PROFILE

Dimitrios Liappis
1 PUBLICATION **10** CITATIONS
SEE PROFILE

Blueprints for a Large–Scale Early Warning System

Paul G. Spirakis[1], Vasileios Vlachos[2], Vassilios Karakoidas[3], Dimitrios Liappis[3], Dimitrios Kalaitzis[1],
Eftychios Valeontis[1], Spyros Kollias[2] and George Argyros[2]
Research Academic Computer Technology Institute (RACTI)
Patras, Greece
[1]{spirakis, kalaitzis, valeontis}@cti.gr
[2]{vsvlachos, spyridon.kollias, argyros.george}@gmail.com
[3]{vassilios.karakoidas, dimitrios.liappis}@wizhut.com

***Abstract*—Modern aggressive types of malcode demonstrate that existing security applications are not able to neutralise them efficiently. We present a Large–Scale Early Warning System named PROTOS, which is able to gather intelligence from a large number of personal computers, acting as sensors, utilising their default security mechanisms and applications, to collect and analyse locally intercepted malicious network traffic and generate an estimation of the global malware activity.**

I. Introduction

Recent trends indicate an escalation of cybersecurity incidents over the last years, as the number of threats is growing at an increasing rate [1]. Moreover, we observe a qualitative improvement in the malicious capabilities of most malware authors, as they are able to perform more sophisticated and skilful attacks. Numerous security applications are being utilised to hinder the rapid propagation of malcode. Home users rely on their pre-installed firewall and standard issue antivirus programs to protect their systems, while larger organisations defend their digital assets and infrastructure, using Intrusion Detection and Intrusion Prevention Systems (IDS, IPS). Theoretical analysis, as well as empirical evidence, suggests that modern security applications are beneficial, but not sufficient to eliminate the majority of recent threats [2], [3]. Self–propagating malcode, such as the Slammer worm [4], is able to infect the majority of susceptible systems on the Internet in less than ten minutes. Other forms of malicious software (Code Red, Code Red II [5], [6], Blaster [7], Witty [8], Nimda [9]) are using similar massively attacking methods or other covert tactics.

Computer system security is considered important; Europe alone spent 4.6 billion Euros for security applications in 2008 [10]. Even though a substantial amount of money and effort has been spent for improvement of the security of IT infrastructure, there are no indications of diminishing malicious activity. The reasons behind our inability to protect efficiently the critical technological infrastructures, as well as most of the interconnected computer systems, using the internet, are both technological and societal.

The short timeframe, during which modern malcode mutates, renders most of the traditional defensive mechanisms ineffective [11], [12]. The increasing rate that new malware appears, prohibits the timely generation and distribution of signatures for all new threats and their variants. Therefore, it seems of particular importance that the efforts of the research community ought to concentrate also on proactive measures.

This research work presents the blueprints and the first prototype of a collaborative Large–Scale Early Warning System, named PROTOS (PROactive Threats Observatory System). The first version of the system is going to be used at the Greek National School Network, thus becoming available to more than 100,000 students and teachers. The second planned version of the system will become publicly available for all internet users.

The basic concepts of our system lie in the hypothesis that it is possible to defend most of the users by utilising the heterogeneity of their computer systems. Most malware attacks, in order to propagate, utilise specific infection channels, such as vulnerable operating systems' services or applications. Diversity in operating systems, configuration settings, productivity suites and other applications, may provide important information, if monitored, regarding the activity of computer malware. Non–vulnerable systems are expected to deny and log suspicious connection attempts from unknown origins. The acquisition and analysis of a large number of such data may possibly reveal ongoing malicious activity at its early stages.

This paper is organised as follows; Section II presents the architecture and implementation details of our proposed system, PROTOS, while Section III analyses the prediction mechanisms implemented so far. Section IV summarises the related work. Section V discusses the future extensions of this work and concludes this paper.

II. Architecture & Prototype Implementation

Figure 1 illustrates the architecture of the system. The prototype follows a client/server approach. The client is composed of two modules, a *(1) system–level module*, which is running as a service and a *(2) front–end user–space application*. The system–level module is implemented in Python and it performs two basic operations: (1) it parses the Windows firewall log file every 30 seconds, calculates the rate of denied traffic, as well as the changes of that rate, which is known in epidemiology as epidemic curve and transmits them to the server and (2) it transmits four times a day a more comprehensive report including additional information such as the IP addresses of the 10 worst offenders, source and destination port and the packet type.

A thin client based on the Yahoo! widget framework is also available. The PROTOS widget is initiated during system startup and provides real–time information regarding the local malicious activity. It is implemented with simplicity in mind and does not require any user intervention or configuration, to help users to understand the general threat level.

All system data are encoded and transmitted using the JavaScript Object Notation[1] (JSON) format. A sample message is presented in Figure 2.

The prototype client has been developed for all modern versions of Microsoft Windows operating system, including Windows XP SP3, Windows Vista and Windows 7. The installation of the required software is available as an installer package for those platforms. Automatic updates functions are also supported to enable transparent updates without user intervention.

The server side of PROTOS is currently using the Apache Web Server and it is implemented in PHP. All incoming information from the clients, are stored in a MySQL database.

In the future, a Java application will also be available to the users, in case they want to monitor the intercepted local and global malicious activity in greater detail.

III. Security Prediction Mechanisms

A. Malware Activity and Epidemic Rate

Prototype PROTOS client, repeatedly checks during predefined short time intervals the log file of Windows' firewall and calculates the rate of the intercepted malicious activity, as shown in eq.(1) and the *Epidemic Rate*, eq. (2) for each host.

$$p_t^n = \frac{h_t^n - \frac{\sum_{i=t-k}^{t-1} h_i^n}{k}}{\frac{\sum_{i=t-k}^{t-1} h_i^n}{k}} \tag{1}$$

$$q_t^n = \frac{p_t^n - \frac{\sum_{i=t-k}^{t-1} p_i^n}{k}}{\frac{\sum_{i=t-k}^{t-1} p_i^n}{k}} \tag{2}$$

t is the ordinal number of a fixed time interval, n is the client identifier, h_t^n is the number of the security incidents n received in the interval t. $k, k \in (0, t-1)$ is the size of the "time window" used in the number of t time intervals. PROTOS clients send this local malicious activity rate to the central PROTOS server, which constantly listens for incoming rates and aggregates them in order to compute the global malicious activity rate by using the following equation:

$$p_{avg} = \frac{\sum_{i=1}^{n} p_t^i}{n} \tag{3}$$

The outcome of this equation is the *Malware Rate*, which has been visualised using empirical data in Figure 4.

[1] http://www.json.org/

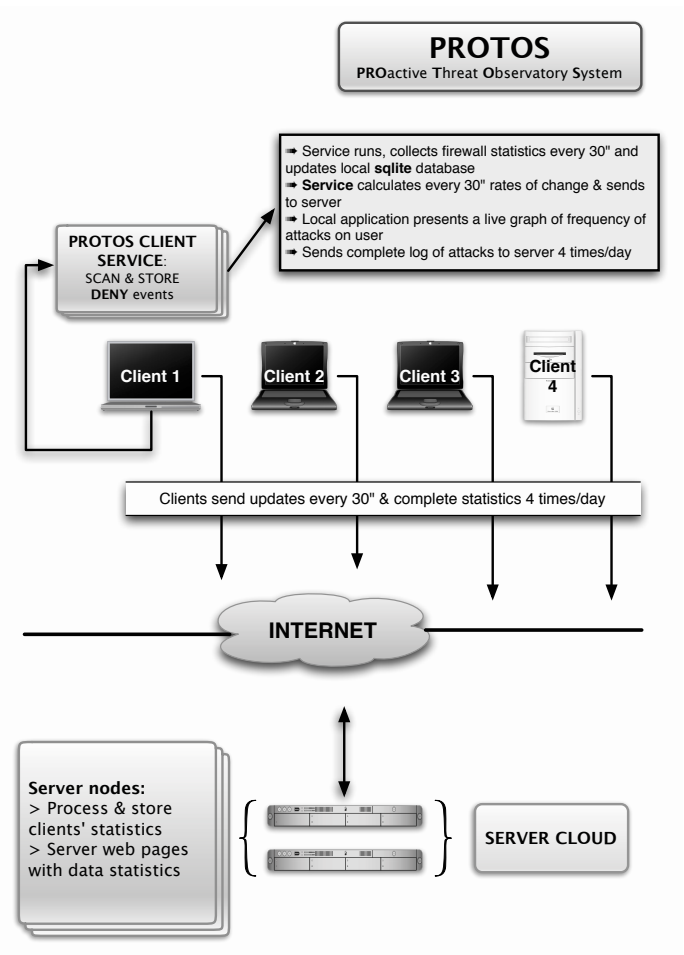

Fig. 1. Architecture of the PROTOS System

```
{  "clientid": "<md5 checksum>",
   "rate1": 1.52,
   "rate2": 2.47,
   "tcount": 50,
   "localip": "192.168.1.1" }
```

Fig. 2. JSON–encoded client message

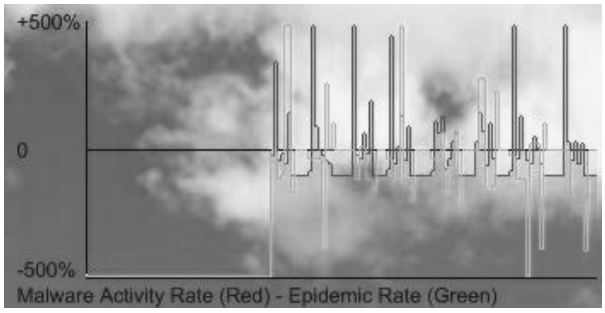

Fig. 3. Yahoo! Widgets Real-Time Malware Activity

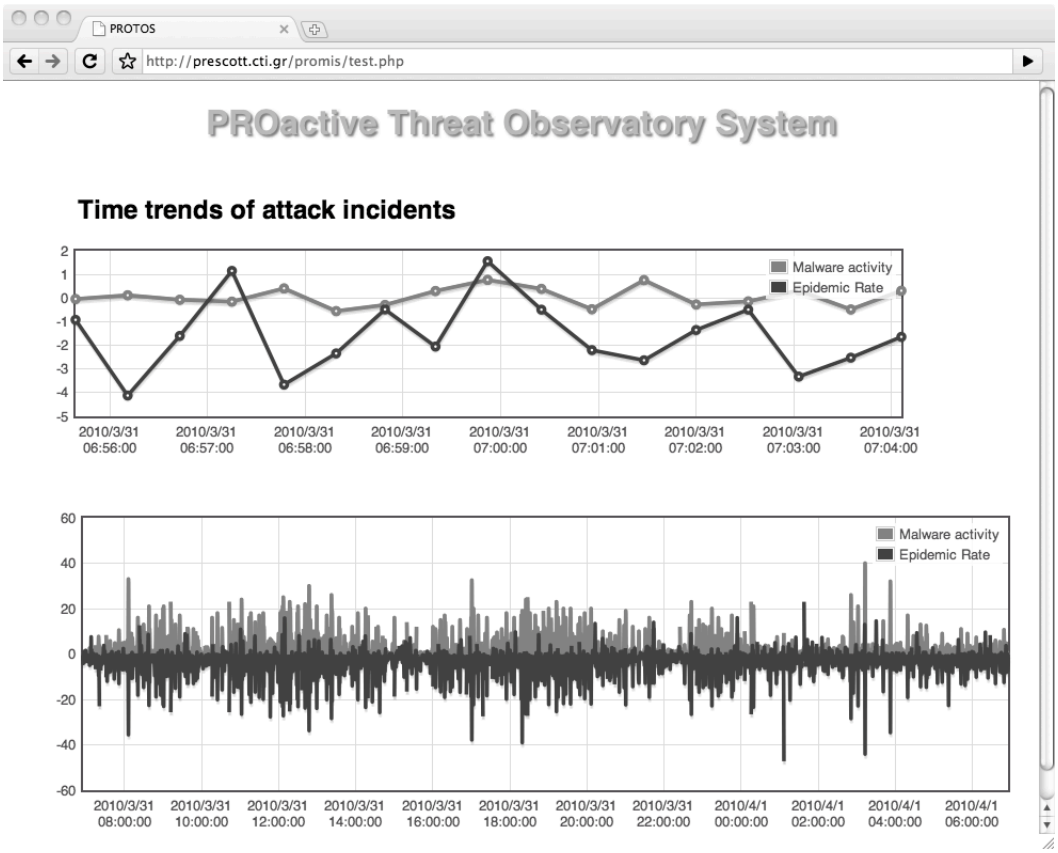

Fig. 4. The PROTOS Web Server

B. Intrusion Detection System

In order to enhance our system's effectiveness and threat-detection accuracy we will be implementing a detection mechanism based on the framework proposed by [13]. This mechanism will work side by side with the core system, constantly evaluating the security status in an effort to minimise our systems false-positive alarm rates.

This layer consists of two modules: (a) the network traffic sensor, called Agent and (b) the decision engine, called Juror. Through extensive experimentation with, both new and old, threats and cyber-attacks we are trying to isolate specific characteristics that most of them have in common. Such characteristics provide probabilistic distributions both for normal and abnormal network traffic that are being feed to the Agent. Using these distributions and a statistical toolkit the Agent produces probability reports on the network status, stating the probability of an attack along side with the probability of having a false alarm. These reports are sent to the Juror module, where they are processed. When the system reaches a predefined certainty threshold it issues an attack decision, informing the core system.

The current prototype focuses on intrusive behaviour detection at very beginning of an attack, when hackers and worms are trying to discover potential victims, mainly hosts with vulnerable services. We have devised two classes containing all the TCP/IP protocol packets. The first class, called NON-ACK, contains all of the packets that relate to connection opening, closing and handling, where the second class, ACK, contains packets that are sent and received during a normal connection. Experiments yielded two probability distributions for the NON-ACK class, one for normal traffic and one for intrusive behaviour. These distributions are given as input to the sensor engine described at [13] and along with some predefined sensitivity parameters, they compose the sensor module.

The decision engine module that handles the sensors reports can be installed at the same host as the sensor, making decisions only for the hosts traffic, or at a central point of a network, where it can handle data from the entire network. The choice of setup depends mostly on the host. Organisations with large networks are encouraged to use the distributed architecture with one Agent per host and one central Juror, as opposed to home users or servers where the standalone setup has better results.

Network traffic monitoring demands much resources, thus both modules have been implemented in C++ to ensure the best possible performance from native code compilation. The system processes only incoming packets as an additional performance tweak without compromising the quality of the results, due to TCPs' symmetric nature.

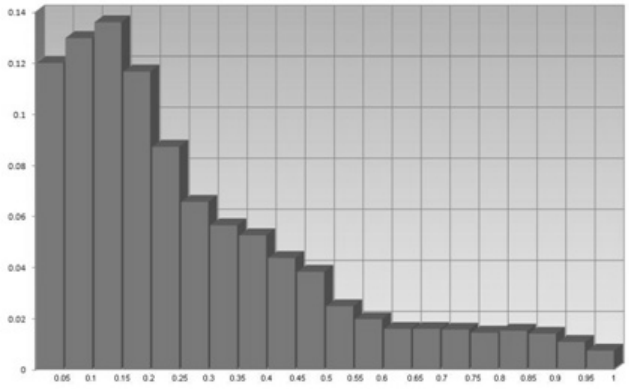

Fig. 5. The NON-ACK normal traffic distribution

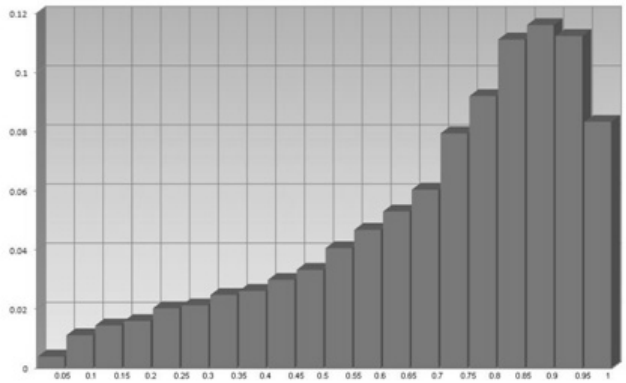

Fig. 6. The NON-ACK intrusive behavior distribution

In the course of experimenting, the prototype has proven efficient at detecting port-scans and OS fingerprinting attempts. In the case of distributed setup, the system was able to identify horizontal port-scans, attempts to find specific vulnerable services at a specific private network.

IV. Related Work

A large–scale Early Warning System requires a sufficient number of sensors in order to provide accurate results and timely warnings. Various academic frameworks have been proposed for distributed malware detection [14], [15], [16], [17], [18], but none according to the best of our knowledge has been deployed at large.

Interesting alternative systems, based on peer–to–peer or other decentralised technologies are available [19], [20]. Existing systems that are actively collecting security related data and provide early notifications to their users are operated by large security vendors. The oldest and most known commercial early warning system is DeepSight [21], operated by the Symantec corporation. Another commercial system is Cisco's IronPort [22], which combines a number of criteria in order to decide whether a network location is secure or not. Both of these systems deliver a healthy volume of information to their clients, but they are not available for public use. Symantec's DeepSight requires paid subscriptions at the cost of several hundreds of US Dollars per year, while IronPort requires specialised hardware from Cisco. Therefore both these frameworks are not widely adopted.

Another widespread collaborative effort to correlate the security incidents is the Dshield [23] system, which covers more than 500,000 IP addresses from over 50 countries. All those systems present their estimations of the general malware activity in their web portals and issue warnings via e-mails and RSS feeds.

V. Discussion and Future Work

The system is a prototype and only its basic functions are operational. The service modules and the PROTOS widget are working on a 24–hour basis without any problems. We have installed the PROTOS client in a small number of workstations and gathered some initial data. So far, we have been interested in the development of update mechanisms for the clients and compatibility with Microsoft's security applications, which are in use by the majority of our potential users.

The system has been tested in a lab environment. The system modules of PROTOS and the widget have shown that they are not inducing any significant overhead to the overall performance, but so far we have not tested it thoroughly in all the netbook models that the base users have. Most Greek students have been funded by the Greek Ministry of Education to purchase computer equipment during the past years. Most of those systems have low-end hardware specifications, varying from netbooks to cheap laptops, thus the target runtime platform for PROTOS is an entry level level system with limited processing capabilities, running the Windows XP operating system. PROTOS in its final form will support non–Microsoft operating systems as well, such the Mac OS X and Linux. Our next steps include a full evaluation of the prototype system, both server and client, in terms of scalability and actual overhead to the clients.

In the future, we plan to support dynamic adjustment of the system security level, to act and protect clients from a malware epidemic or other serious incidents. Each client can have different predefined thresholds, which correspond to specific security profiles. During normal activity PROTOS clients operate as usual, however, when the PROTOS server estimates increased general malware activity, the client can voluntarily decide to strengthen its security settings by disabling non critical services and heighten the security level of the web applications. The thresholds are being set according to false positive/negative rates of the local system based on the subsequent directives:

- if $p_{avg} > r_{high}$, then increase the security policy by disabling non essential services, for example HTML preview in mail clients or by increasing the web browser's security settings, where r_{high} is the predefined threshold to increase the security settings of the PROTOS.
- if $p_{avg} < r_{low}$, then decrease the security policy by reactivating the above–mentioned services, where r_{low} is

the predefined threshold to decrease the security settings of the PROTOS.

- if $r_{low} \leq p_{avg} \leq r_{high}$ do nothing.

More elegant approaches than the above simplified threshold method are currently considered and we expect to have them added in the final version of the PROTOS.

We are in the process of developing a forecasting system inspired by the financial sector using common econometric models and more sophisticated tools, such as technical analysis on charts. It will be capable of predicting imminent epidemics and threats in contrast to the above method, which acts only when a threat gets characterised as a possible epidemic. The prediction's precision will be more accurate when we have a quite large set of traffic data. We are planning to use AutoRegressive Integrated Moving Average model (ARIMA) [24] for predicting malicious activity rate, General AutoRegressive Conditional Heteroskedasticity model (GARCH) [24] for predicting volatility clusters [25] and Autoregressive Conditional Duration model (ACD) [26] for predicting duration between the attacks or duration until a specific amount of attacks reached or even the duration until a malware mutates. Moreover, chart patterns will be analysed from visualised data by automated software to find specific correlation of common patterns such as "Cup and Holder", "Head and Shoulders" and others that are presented in [27].

The prototype has served us admirably so far in identifying many security pitfalls. The first prototypes have been built from scratch using free and open source software. We are confident that the final system, which will be supported by enterprise–grade software and hardware components, will give us the opportunity to maximise the number of PROTOS clients that will participate in our systems and provide them with timely warnings regarding malicious activity in the cyberspace.

ACKNOWLEDGMENTS

This research work is being implemented under the framework "Integrated Services to Reinforce Digital Trust", funded by the Operational Programme "Digital Convergence". The framework is co-funded by the European Regional Development Fund (ERDF). We would like to thank Paris Bayias, Andry Toska, Valia Toska and Kostas Gratsias for their comments and ideas during the compilation of this paper.

REFERENCES

[1] D. Turner, J. Blackbird, M. K. Low, T. Adams, D. McKinney, S. Entwisle, M. L. C. Wueest, P. Wood, D. Bleaken, G. Ahmad, D. Kemp, and A. Samnani, "Symantec global internet security threat report. trends for 2008," Symantec, Tech. Rep., April 2009.

[2] S. Staniford, V. Paxson, and N. Weaver, "How to 0wn the internet in your spare time," in *Proceedings of the 11th USENIX Security Symposium*, August 2002, pp. 149–167. [Online]. Available: http://www.icir.org/vern/papers/cdc-usenix-sec02/

[3] S. Staniford, D. Moore, V. Paxson, and N. Weaver, "The top speed of flash worms," in *WORM '04: Proceedings of the 2004 ACM workshop on Rapid malcode*. New York, NY, USA: ACM Press, 2004, pp. 33–42.

[4] D. Moore, V. Paxson, S. Savage, C. Shannon, S. Staniford, and N. Weaver, "Inside the slammer worm," *IEEE Security & Privacy*, pp. 33–39, July 2003.

[5] C. Zou, W. Gong, and D. Towsley, "Code red worm propagation modeling and analysis," in *Proceedings of the 9th ACM Conference on Computer and Communication Security (CCS)*, Washington DC, USA, November 2002.

[6] D. Moore, C. Shannon, and k. claffy, "Code-red: a case study on the spread and victims of an internet worm," in *IMW '02: Proceedings of the 2nd ACM SIGCOMM Workshop on Internet measurment*. New York, NY, USA: ACM, 2002, pp. 273–284.

[7] M. Bailey, E. Cooke, F. Jahanian, D. Watson, and J. Nazario, "The blaster worm: Then and now," *IEEE Security & Privacy*, vol. 3, no. 4, pp. 26–31, July 2005.

[8] C. Shannon and D. Moore, "The spread of the witty worm," *IEEE Security & Privacy*, vol. 2, no. 4, pp. 46–50, July 2004.

[9] A. Mackie, J. Roculan, R. Russell, and M. VanVelzen, "Nimda worm analysis - incident analysis report version ii." Security Focus, Tech. Rep., September 2001. [Online]. Available: {C}urrenton-line({June}2003):http://aris.securityfocus.com/alerts/nimda/010919-Analysis-Nimda.pdf

[10] R. Anderson, R. Boehme, R. Clayton, and T. Moore, "Security economics and the internal market," European Network and information Security Agency (ENISA), Tech. Rep., Januar 2008.

[11] N. Weaver, V. Paxson, S. Staniford, and R. Cunningham, "Large scale malicious code: A research agenda," Current on-line (June 2005): http://www.cs.berkeley.edu/ nweaver/large scale malicious code.pdf, May 2003.

[12] N. Weaver, V. Paxson, and S. Staniford, "A worst-case worm," in *Proceedings of the Third Annual Workshop on Economics and Information Security (WEIS04)*, May 2004.

[13] T. Komninos, P.Spyrakis, and H. Tsaknankis, "Real time distributed detection of network attacks," *IJCSNS International Journal of Computer Science and Network Security*, vol. 6, no. 7, July 2006.

[14] C. Zou, L. Gao, W. Gong, and D. Towsley, "Monitoring and early warning for internet worms," in *Proceedings of the 10th ACM Conference on Computer and Communication Security*, Washington DC, USA, October 2003.

[15] S. Sna, J. Brentano, G. Dias, T. Goan, T. Heberlein, C. Ho, K. Levitt, B. Mukherjee, S. Smaha, T. Grance, D. Teal, and D. Mansur, "DIDS (distributed intrusion detection system) - motivation, architecture, and an early prototype," in *Proceedings of the 14th National Computer Security Conference*, Washington, DC, 1991, pp. 167–176. [Online]. Available: citeseer.nj.nec.com/sna91dids.html

[16] S. Singh, C. Estan, G. Varghese, and S. Savage, "The earlybird system for real-time detection of unknown worms," HOTNETS-II, 2003.

[17] J. Shoch and J. Hupp, "The "worm" programs–early experience with a distributed computation," *Computing Practices*, vol. 25, no. 3, pp. 172–180.

[18] G. Bakos and V. Berk, "Using sensor networks and data fusion for early detection of active worms," in *Proceedings of the SPIE Aerosense conference, Orlando Florida*, April 2003.

[19] V. Vlachos, S. Androutsellis-Theotokis, and D. Spinellis, "Security applications of peer-to-peer networks," *Comput. Networks*, vol. 45, no. 2, pp. 195–205, 2004.

[20] V. Vlachos and D. Spinellis, "A PRoactive malware identification system based on the computer hygiene principles," *Information Management and Computer Security*, vol. 15, no. 4, pp. 295–312, 2007. [Online]. Available: http://www.dmst.aueb.gr/dds/pubs/jrnl/2007-IMCS-Promise/html/VS07.html

[21] DeepSight, "Symantec DeepSight Threat Management System," available at (March 2010): http://tms.symantec.com/.

[22] Cisco, "Cisco Ironport Web Reputation Filters," available at (March 2010) http://www.ironport.com/pdf/ironport_wbrs_datasheet.pdf.

[23] DShield, "Distributed intrusion detection system," available at (March 2010) http://www.dshield.org/.

[24] T. Watsham and K. Parramore, *Quantitative methods in finance*. International Thomson Business Press, 1997.

[25] R. Cont, "Empirical properties of asset returns: stylized facts and statistical issues," October 2000.

[26] R. F. Engle and J. R. Russell, "Analysis of high frequency financial data," New York University and University of California, San Diego University of Chicago, Graduate School of Business, Tech. Rep., 2004.

[27] T. N. Bulkowski, *Encyclopedia of Chart Patterns*. John Wiley & Sons, Inc., 2005.