Distributed Instrusion Prevention in Active and Extensible Networks

Todd Sproull and John Lockwood *

Applied Research Laboratory
Department of Computer Science and Engineering:
Washington University in Saint Louis
1 Brookings Drive, Campus Box 1045
St. Louis, MO 63130 USA

http://www.arl.wustl.edu/arl/projects/fpx/reconfig.htm

Abstract. The proliferation of computer viruses and Internet worms has had a major impact on the Internet Community. Cleanup and control of malicious software (malware) has become a key problem for network administrators. Effective techniques are now needed to protect networks against outbreaks of malware. Wire-speed firewalls have been widely deployed to limit the flow of traffic from untrusted domains. But these devices weakness resides in a limited ability to protect networks from infected machines on otherwise trusted networks.
Progressive network administrators have been using an Intrusion Prevention System (IPS) to actively block the flow of malicious traffic. New types of active and extensible network systems that use both microprocessors and reconfigurable logic can perform wire-speed services in order to protect networks against computer virus and Internet worm propagation. This paper discusses a scalable system that makes use of automated worm detection and intrusion prevention to stop the spread of computer viruses and Internet worms using extensible hardware components distributed throughout a network. The contribution of this work is to present how to manage and configure large numbers of distributed and extensible IPSs.

1 Introduction

Security has become a daunting task for network administrators. There are numerous vulnerabilities that affect the millions of computers attached to the Internet. Network administrators are overwhelmed by the task of securing their networks against operating system flaws, poorly written network applications, and end-system misconfigurations. Security devices integrated within the network have become a necessity for networks that need to be safe and reliable.

* This research was supported by a grant from Global Velocity. The authors of this paper have received equity from the license of technology to Global Velocity, and have served as consultants to the company.

Network administrators currently use several types of devices to secure their networks. The first line of defense is typically a firewall. Firewalls provide some protection by limiting how packets destined to and from machines on the Internet send traffic through a network node. While firewalls are useful, they lack the features needed to filter malicious content that passes between Internet hosts that have become infected with an Internet worm or computer virus. To detect a worm or virus activity, intrusion detection systems (IDSs) are needed. IDSs help administrators detect when exploits pass over a network and they log which machines were targeted. The most advanced type of network security device is called an Intrusion Prevention System (IPS). An IPS scans the content of traffic flowing through a network and actively drops the traffic flows which are detected to be malicious. Unfortunately, there are several problems with the way that firewall, IDS, and IPS devices are deployed throughout the Internet today.

In recent years, Internet worms generally entered a network only at the edge. Today, malware is *multi-modal* meaning that it uses multiple techniques to propagate and infect machines. Multi-modal malware can spread both over the network as a worm and via removable media as a virus. Multi-modal worms provide several mechanisms for an infected machine to infect other machines using other modes independent of the original mode of infection. With *Sasser*, for example, a laptop user could have their machine infected by network traffic while it was connected to a Digital Subscriber Line (DSL) at home. When that same user takes the machine to work, that laptop infects the rest of the hosts on the internal network by using a port scan. To be effective against this type of threat, network security devices need to distributed throughout the network, not just used at the edge.

A problem with network security devices is that they can be hard to manage. Many IPS and IDS devices lack the ability to automatically download patches that allow them to protect networks against new threats. As the number of network security devices increase, so does the time spent by an administrator to push out the latest rules and virus signatures to remote devices. Methods are needed to automatically distribute information regarding new virus signatures to all of the IPS devices on a network. Better security for entire networks can be achieved with a Distributed Intrusion Prevention System (DIPS). Figure 1 depicts an example network containing DIPS spread throughout a network. Hosts (H) attach to Subnets (S). Routers (R) forward traffic between subnets. DIPS nodes placed in-line with high-speed links actively measure and filter malicious traffic attempting to flow between subnets, routers, or virtual local area networks (V). We believe that active and extensible networks can be used as the foundation to implement highly scalable distributed intrusion prevention devices and that active network technology can be used to implement the control and configuration software for a network of DIPS.

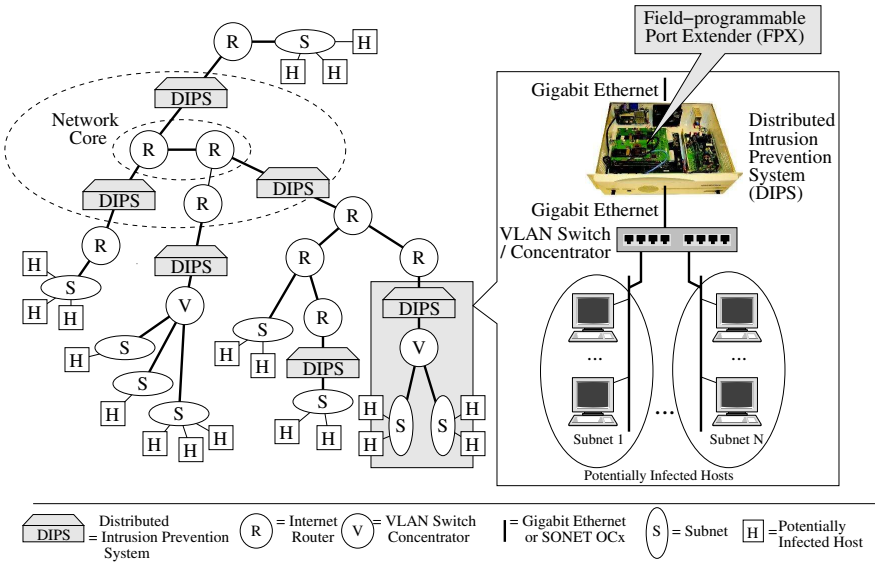

Fig. 1. High level view of potential threats

2 Intrusion Detection

One of the first widely used intrusion detection systems is called *SNORT* [1]. SNORT enabled network administrators to promiscuously scan a network link to see what type of exploits were passing over the network and being used to attack their hosts. HOGWASH [2] expanded upon SNORT to implement intrusion prevention functions. Traffic passing through a PC that ran the HOGWASH software would be sanitized to remove malware and malformed packets before the exploit could reach the machines on the other side of the network. A problem with HOGWASH was that the limited throughput of the PC that ran the HOGWASH software became a bottleneck to network throughput. Packets would be delayed or dropped as the software that executed on the node saturated the capacity of the processor.

2.1 Intrusion Prevention in Hardware

Intrusion prevention systems that use reconfigurable hardware can detect signatures at high speeds by scanning for signatures in traffic that contain malware and blocking certain data transmissions [3]. One system that scanned for signatures in packets payloads and blocked malware using Field Programmable Gate Array (FPGA) technology was described in [4]. Large numbers of parallel Finite State Machines (FSMs) were configured into FPGA hardware to implement the computationally intensive function of scanning for regular expressions. Another system used Bloom filters to scan for large numbers of signatures with FPGA

hardware. Bloom filters had allowed for fast incremental updates to the signature list with nearly no delay [5].

2.2 Distributed Firewalls and IDS Control

Distributed control is needed to manage large numbers of firewalls or IDSs. The distributed firewall described in [6] allows for a centralized access control policy that could be enforced at multiple remote locations. In [7], the implementation of a distributed firewall using the *KeyNote* trust management system was described in order to ensure secure transmission of credentials and distribution of network policies.

In other work, Huang introduced a framework for large scale intrusion detection using strategic decision making [8]. The model analyzes a sequence of events and uses global knowledge to make an informed decision regarding an intrusion. This approach relies on local agents to monitor and announce events, while a global agent predicts trends and makes strategic decisions. Here the sensor nodes do not actively block traffic until receiving an order from a global command node.

2.3 Real-Time Anomaly and Worm Detection

A system which discovers worms on a network in real-time has been developed using reconfigurable hardware [9]. Network content is monitored to discover frequently occurring signatures that appear in packet payloads. The system uses FPGAs to scan packets for patterns of similar content at Gigabit per second link rates. This system can be used to automatically detect signatures of new Internet worms just as an outbreak begins. Another system has been developed that uses anomaly detection to contain a worm to a small subsection of the network [10]. This approach allows for cooperation among multiple containment devices to respond an attack more effectively.

2.4 Peer-to-Peer Control and Management

In order for a DIPS to be scalable, there needs to be a way to control and configure thousands to tens of thousands of remote devices. It is not necessary to implement a centralized control of all DIPS devices. Peer-to-peer strategies can and should be used to distribute information in large-scale networks.

A software system called *Scribe* [11] provides a scalable, self-organizing Peer-to-Peer (P2P) location and routing substrate. Scribe was built on top of *Pastry* [12] and added functionality to perform large-scale, decentralized, application-level multicast. In the Scribe model, nodes participate as equal participants in groups. These nodes are joined together using routes provided by the Pastry software to form a multicast tree. Scribe provides an API for nodes joining groups and takes advantage of the robustness and reliability provided by Pastry. The effectiveness of a coordinated approach as compared to other types of P2P communication models has been proven in [13].

Janakiraman [14] proposed a scalable IDS/IPS solution that distributed a firewall and placed IDS systems throughout a P2P network. In this work, nodes share information on network intrusion attacks that occur throughout the network. The prototype system classified intrusions such as failed login attempts or port scans. A framework called DShield provides a platform for firewalls to share intrusion information [15]. By sharing information about new exploits among multiple machines, better protection can be provided than if information was only collected locally. DShield interacts with network administrators by providing graphs in real-time that include the identification of the top attacker and most prevalent port being targeted. Other work in network management for security devices includes the model proposed by Hyland and Sandhu [16]. In this work, security devices on the network are described as managed objects that interact through SNMP. A new protocol is also introduced to propagate security information throughout a network similar to the mechanism used by Internet routing protocols.

2.5 Security

In order to protect the network of DIPS, the infrastructure that provides protection must be secure itself. The system must ensure that only trusted systems can control the operation of remote DIPS. Some work has been done to secure the control and configuration of reconfigurable hardware platforms [17]. But as noted there are challenges with the implementation of a public key exchange using hardware alone. Key generation functions can be better handled by a general purpose processor in software. There are now FPGAs, like the Xilinx Virtex II Pro, that embed a full-feature PowerPC core within the FPGA logic array to allow use of both hardware and software on a single integrated circuit [18].

Distributed security techniques have also been proposed in Centaurus2 [20] and SHOMAR [21]. These projects demonstrated how decentralized services throughout an enterprise could provide authentication, anti-replay prevention, and non-repudiation. The security model employed is based around a simplified public-key infrastructure (PKI) [22]. This allows nodes to communicate and authenticate themselves throughout an untrusted network. Centaurus2 describes the framework for supporting this secure infrastructure, while SHOMAR demonstrates a distributed intrusion detection system (DIDS) using the aforementioned security techniques.

3 Distributed Intrusion Prevention Design Framework

To protect entire networks from rapid outbreaks of worms, computer viruses, and other malware; next generation networks should actively scan data passing through the network and provide an automated response in a coordinated fashion to stop the spread of malware. We feel that the active networking community is well positioned to develop the technologies which can provide automated protection of networks. In fact, data security appears to be a killer application that will drive the use of active and extensible networks in the Global Internet.

Several issues must be considered in order to design effective distributed intrusion prevention systems. One goal is to detect and block large numbers of Internet worms and viruses. Another goal is to enable large numbers of DIPS to organize themselves into an overlay network and securely communicate with each other. To address these challenges, a framework has been developed that describes how multiple sensors and actuator nodes communicate to perform intrusion detection and prevention in a secure distributed system.

3.1 Sensor and Actuator Nodes

We envision that each active network node in the intrusion prevention system contains six primary modules in order to detect and block worms and viruses. The first module in each node reconstructs Transmission Control Protocol (TCP) flows passing through the network node [27]. The second module processes headers and payloads to match rules that are specified using a syntax like the one used by SNORT [26]. The third module drops packets or flows containing known virus signatures. The fourth module performs anomaly detection. Unusual network activities cause the node to generate an alert, unusual activity includes port scans or a particular host opening a large number of TCP connections in a small period of time [10]. The fifth module monitors network traffic looking for a large increase in commonly occurring content. The sixth module decides what traffic flows to filter based on clues from the content scanning and anomaly detector modules.

3.2 Management Nodes

System administrators do not have the ability to monitor all of the IPSs distributed throughout a network, nor can they react quickly enough to stop an outbreak the moment that a new virus is discovered. Active intrusion prevention systems are needed that automatically reprogram IPS devices to stop rapid worm outbreaks. To be effective, entire networks of DIPS should be reconfigured within seconds of a new worm outbreak.

Scalable mechanisms are needed to control and configure large numbers (thousands to tens of thousands) of distributed intrusion prevention systems, in large scale, self-organizing networks. We propose use of a P2P solution based on the Scribe model [11].

To deploy active protection in the Internet, we propose that nodes be managed as small and large groups. Small groups consist of hundreds to thousands of hosts attached to tens to hundreds of active IPS nodes. Large groups encompass multiple small groups, and are managed by individual network providers with different levels of trust established between them.

3.3 Security for Group Membership

Care must be taken to decide whether or not to trust a node when it attempts to join a group. Access Control Lists (ACLs) restrict communication among a

group of nodes. In order for a new DIPS to join the group, the DIPS must be authenticated by a node already trusted on the network. If a node lies outside of the trust domain, other techniques are needed to verify its credentials.

Communication among DIPS nodes in a secure manner is critical. Public Key Infrastructure (PKI) uses digital certificates, public-key cryptography, and certificate authorities to implement trust relationships and secure communication between network nodes [23]. To secure the entire distributed network of intrusion prevention systems, we propose using a secure communication model based on SHOMAR [21].

In this model, communication occurs between a DIPS, the Certificate Authority (CA), and the DIPS Manager (DM). The CA generates and signs x.509 certificates [24] for each DIPS in the network. The CA also verifies certificate queries from DIPS. The DM holds an ACL of all the nodes and their group membership capabilities.

As with [21], certificates are initially generated for each DIPS. That information is placed into the DIPS through an out-of-band mechanism. Certificates are stored on each DIPS in a secure manner, using a mechanism such as a PKCS#11 container [25].

4 Distributed Intrusion Prevention System Model

Distributed intrusion prevention can be implemented in a way that both provides high performance and is cost effective. The model uses both extensible hardware to process large volumes of data and active network software to manage and control the distributed system.

4.1 Extensible hardware

Extensible hardware enables network traffic to be processed at the full line rate of Gigabit/second networks. As described in [3], an IPS was built using the Field Programmable Port Extender (FPX) platform. The FPX is equipped with a Virtex 2000E FPGA that can be dynamically reconfigured over a network to perform data processing functions. Several functions have been implemented on the FPX that perform IPS functions as modules. A TCP processor was implemented that can reconstruct traffic in 8 million active flows at 2.5 Gigabits/second [27]. A Bloom Filter was implemented on the FPX to scan for 10000 virus signatures at a data rate of 2.4Gbits/sec [5]. An Internet Security module was implemented that performs a subset of the SNORT functionality by processing headers and performing full packet scanning in hardware [26]. A worm detection module was also prototyped on the FPX platform [9].

The FPX platform has been integrated into a chassis that allows multiple FPX cards to be stacked and includes an embedded Single Board Computer (SBC). This SBC contains an Intel Celeron Processor that runs Linux from a flash memory device. The Celeron processor is only used to perform control functions. All of the core packet processing is done on one or more FPX cards.

Figure 2 shows a photo of this new system, called the GVS 1500, with the cover open. As can be seen in the figure, FPX cards are stacked in the front of the chassis below two Gigabit Ethernet line cards. The SBC can be seen in the back of the chassis. When the system is powered on, the SBC boots into Linux and programs FPX cards using a program called NCHARGE [28].

4.2 Active Network Management

We propose to execute management and control services of the distributed system using the Scribe communication protocol. Each IPS would automatically discover other IPS in the distributed network using a communication protocol defined by Pastry. The entire DIPS would then self-organize into a tree structure as a single group.

The FreePastry tool provides an open-source Java implementation of Pastry including Scribe [29]. This software serves as framework for the P2P substrate with security extensions to allow for encrypted communication.

4.3 Detection and Reaction to New Malware

There are many characteristics of computer activity that indicate an end host has been infected with malware. Each IPS sensor has a local view of the traffic passing to and from hosts on a local subnet. The activity may be observed as a

Fig. 2. Distributed Intrusion Prevent System (DIPS)

port scan, worm propagation, high volumes of traffic, or other types of anomalous behavior. Observed behavior might be malware, or it could instead be a false positive triggered by a valid use of the host.

By fusing data collected by multiple sensors, then coordinating the efforts of multiple IPS, effective security against worm attacks can be implemented. We envision that three phases are needed in order to block malware and avoid programming the network to block legitimate traffic, as shown in figure 3. To be effective against a worst case worm, all of these activities must be performed within a few seconds to a minute.

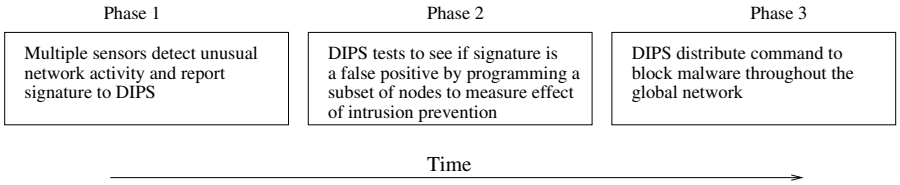

Fig. 3. Phases of the DIPS during a worst-case worm outbreak

4.4 Adding a new IPS to the trusted DIPS

To build a large network of trusted IPS nodes, nodes must assemble in a secure and scalable way. Several steps are required for an IPS to join an DIPS group, as shown in figure 4.

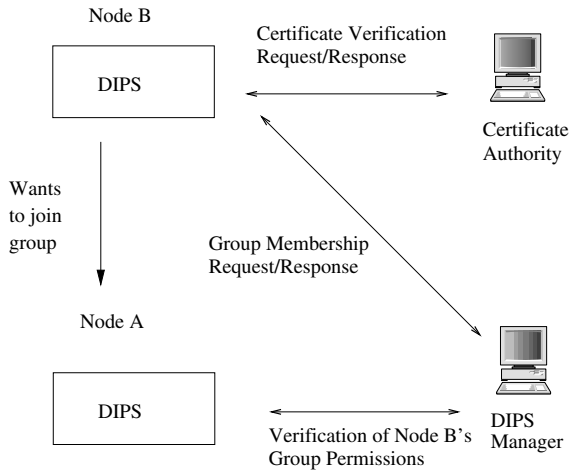

Fig. 4. DIPS joining a group

The example shows how DIPS B would join the group for which A is already a member. B first verifies the digital certificate it holds for the CA by issuing a certificate request. The CA then issues a signed response for the request, assuming it is valid. B next sends a message to the DM for a certificate request and to register itself to join the group. The DM responds with an acknowledgment indicating that it is eligible to join the groups predetermined for B by a network administrator. B then sends a request to A indicating it wishes to join the group. A issues a request to the DM to verify B's membership criteria, once validated B is allowed to become a part of the group and is able to communicate with A.

This verification process is only needed when nodes join a group. The number of requests received by the CA and the DM will be fairly small in comparison to the communication between nodes implementing the SCRIBE multicast overlay.

The model works well with the Scribe infrastructure, as the only nodes allowed to join require the proper credentials to participate in the overlay network. In this example, each DIPS maintains a table of the nodes that it has authenticated as belonging to the group.

Since the time scale in which a DIPS joins or leaves a group is slow, the amount of overhead associated with introducing a new DIPS to the network is relatively small. Software can perform the authentication task and establish a secure connection via a digital certificate.

Trust between the networks can be established using properties similar to that of the Web-of-Trust model used for PGP communication [30]. For networks not under the control of a single authority, an administrator of one domain can choose to receive updates from other domains. In order for this model to scale with larger networks, a decentralized CA model can be implemented. An example of one such system was described by Koga [31].

5 Conclusion

The spread of worms and viruses throughout the Internet has had a devastating impact on end users who suffer when their computers become infected with malware and on system administrators who deal with the burden of protecting entire networks of hosts. Active and extensible networks can be used to implement a distributed intrusion prevention system that decreases the rate at which worms and viruses spread. By stopping or slowing a worm outbreak, data can be saved and machines can be patched before they would otherwise become infected. Passive systems for intrusion detection have been used in the past to alert when a machine is compromised or a network is under attack. Active systems can be used to stop an attack and prevent a worm from spreading. By using extensible hardware, this type of protection can be provided with minimal impact on overall network performance.

Distributed network intrusion prevention systems can be used to protect large numbers of system globally. This paper described how active network management software and extensible hardware can work together in order to protect high speed networks from fast outbreaks of new Internet worms and viruses.

A prototype implementation of the system is being developed at Washington University in Saint Louis and being deployed by Global Velocity.

6 Future Work

Large test beds should be built in order to evaluate the effectiveness of the distributed system. The circuits that implement the hardware functionality of the system are already in place. Anomaly detection modules can be developed as reconfigurable modules then deployed using active network technology. Time and effort is needed to port the Pastry/Scribe architecture to the DIPS platform. Measurements of the system should be performed to determine how quickly the system can deploy protection against new virus signatures. We plan to deploy this infrastructure on large scale networks to determine how quickly it can quarantine a network from the spread of viruses as the system reacts to various changes in the network.

References

1. Roesch, M.: SNORT - lightweight intrusion detection for networks. In: LISA '99: USENIX 13th Systems Administration Conference, Seattle, Washington (1999)
2. Hogwash Homepage `http://hogwash.sourceforge.net/docs/overview.html` (1999)
3. Lockwood, J.W., Moscola, J., Reddick, D., Kulig, M., Brooks, T.: Application of hardware accelerated extensible network nodes for internet worm and virus protection. In: International Working Conference on Active Networks (IWAN), Kyoto, Japan (2003)
4. Lockwood, J.W., Moscola, J., Kulig, M., Reddick, D., Brooks, T.: Internet worm and virus protection in dynamically reconfigurable hardware. In: Military and Aerospace Programmable Logic Device (MAPLD), Washington DC (2003) E10
5. Dharmapurikar, S., Krishnamurthy, P., Sproull, T., Lockwood, J.W.: Deep packet inspection using parallel Bloom filters. In: Hot Interconnects, Stanford, CA (2003) 44–51
6. Bellovin, S.M.: Distributed firewalls. ;login: magazine, special issue on security (1999) 37–39
7. Ioannidis, S., Keromytis, A.D., Bellovin, S.M., Smith, J.M.: Implementing a distributed firewall. In: ACM Conference on Computer and Communications Security. (2000) 190–199
8. Huang, M.Y., Wicks, T.M.: A large-scale distributed intrusion detection framework based on attack strategy analysis. In: Proceedings of the First International Symposium on Recent Advances in Intrusion Detection, Louvain-la-Neuve, Belgium (1998)
9. Madhusudan, B., Lockwood, J.: Design of a system for real-time worm detection. In: Hot Interconnects, Stanford, CA (2004) 77–83
10. Weaver, N., Staniford, S., Paxson, V.: Very fast containment of scanning worms. In: Proceedings of the 13th Usenix Security Symposium. (2004)
11. Castro, M., Druschel, P., Kermarrec, A., Rowstron, A.: SCRIBE: A large-scale and decentralized application-level multicast infrastructure. IEEE Journal on Selected Areas in communications (JSAC) (2002) To appear.

12. Rowstron, A., Druschel, P.: Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. Lecture Notes in Computer Science **2218** (2001) 329–350
13. Castro, M., Jones, M.B., Kermarrec, A., Rowstron, A., Theimer, Wang, H., Wolman, A.: An evaluation of scalable application-level multicast built using peer-to-peer overlays. In: INFOCOM, San Francisco, CA (2003)
14. Janakiraman, R., Waldvogel, M., Zhang, Q.: Indra: A peer-to-peer approach to network intrusion detection and prevention. In: Proceedings of IEEE WETICE 2003. (2003)
15. Dshield homepage. `http://www.dshield.org` (1995)
16. Philip C. Hyland et al.: Management of network security applications. In: Proceedings of the 21st NIST-NCSC National Information Systems Security Conference, Arlington, Virginia (1998)
17. Song, H., Lu, J., Lockwood, J., Moscola, J.: Secure remote control of field-programmable network devices. In: FCCM, Napa, CA (2004)
18. Xilinx Virtex 2 Pro product webpage. `http://www.xilinx.com/virtex2pro` (2004)
19. Bagnulo, M., Alarcos, B., Caldern, M., Sedano, M.: Rosa: Realistic open security architecture for active networks. In: International Working Conference on Active Networks (IWAN), Zurich, Switzerland (2002)
20. Cedilnik, A., Kagal, L., Perich, F., Undercoffer, J.L., Joshi, A.: A Secure Infrastructure for Service Discovery and Access in Pervasive Computing. Technical report, University of Maryland, Baltimore County (2001)
21. Undercoffer, J.L., Perich, F., Nicholas, C.: SHOMAR: An Open Architecture for Distributed Intrusion Detection Services. Technical report, University of Maryland, Baltimore County (2002)
22. IETF Simple public key infrastructure (spki) charter `http://www.ietf.org/-html.charters/spki-charter.html` (1994)
23. Maurer, U.: Modelling a public-key infrastructure. In Bertino, E., ed.: Proceedings of 1996 European Symposium on Research in Computer Security (ESORICS' 96). Volume 1146 of Lecture Notes in Computer Science., Springer-Verlag (1996) 325–350
24. Housley, R., Ford, W., Solo, D.: RFC 2459 Internet X.509 Public Key Infrastructure Certificate and CRL Profile (1999)
25. RSA Laboratories PKCS 11 Cryptograhic Token Interface Standard. `http://www.-rsasecurity.com/rsalabs/pkcs/pkcs-11`
26. Attig, M., Dharmapurikar, S., Lockwood, J.: Implementation results of bloom filters for string matchings. In: FCCM, Napa, CA (2004)
27. Schuehler, D.V., Lockwood, J.: A modular system for fpga-based tcp flow processing in high-speed networks. In: Field Programmable Logic and Applications (FPL), Antwerp, Belgium (2004) 301–310
28. Sproull, T., Lockwood, J.W., Taylor, D.E.: Control and configuration software for a reconfigurable networking hardware platform. In: IEEE Symposium on Field-Programmable Custom Computing Machines, (FCCM), Napa, CA (2002)
29. Freepastry webpage. `http://www.cs.rice.edu/CS/Systems/Pastry/FreePastry` (2004)
30. Zimmermann, P.: The official PGP user's guide. MIT Press, Cambridge, MA (1995)
31. Koga, S., Sakurai, K.: Decentralized Methods of Certification Authority Using the Digital Signature Schemes. In: 2nd Anual PKI Research Workshop Proceedings. (2003)