Military and Aerospace Programmable Logic Device (MAPLD), Washington DC, 2003, Paper E10, Sep 9-11, 2003

Internet Worm and Virus Protection in Dynamically Reconfigurable Hardware

John W. Lockwood[1,2], James Moscola[1], Matthew Kulig[2], David Reddick[2], and Tim Brooks[2]

[1]Applied Research Laboratory, Washington University, Saint Louis, MO; http://www.arl.wustl.edu/arl/projects/fpx

[2]Global Velocity, Saint Louis, MO; http://www.globalvelocity.info/

Abstract

The security of the Internet can be improved using Programmable Logic Devices (PLDs). A platform has been implemented that actively scans and filters Internet traffic for Internet worms and viruses at multi-Gigabit/second rates using the Field-programmable Port Extender (FPX). Modular components implemented with Field Programmable Gate Array (FPGA) logic on the FPX process packet headers and scan for signatures of malicious software (malware) carried in packet payloads. FPGA logic is used to implement circuits that track the state of Internet flows and search for regular expressions and fixed-strings that appear in the content of packets. The FPX contains logic that allows modules to be dynamically reconfigured to scan for new signatures. Network-wide protection is achieved by the deployment of multiple systems throughout the Internet.

I. Introduction

Computer viruses and Internet worms cause billions of dollars in lost productivity. Well-known Internet worms, such as Nimda, Code Red, Slammer and most-recently MSBlast, contain strings of malicious code that can be detected as they flow through the network. By processing the content of Internet traffic in real-time, a system with programmable logic devices can detect data containing computer viruses or Internet worms, and prevent them from propagating. A complete system has been designed and implemented that scans the full payload of packets to route, block, and track the packets in the flow, based on their content. One challenge in implementing this system was that the location of a targeted signature in the packet payload could appear at any position within the traffic flow. Another challenge to implementing the system was that signatures could span multiple packets and be interleaved among multiple traffic flows. The paper will describe how these challenges were met and overcome. The result is an intelligent gateway that provides Internet worm and virus protection in both local and wide area networks.

On tomorrow's virtual battlefield, foreign agents could bait public networks with content containing malware specifically designed to damage crucial counterintelligence or military information systems. These foreign agents could introduce malignant worms or viruses disguised as benign data to attack information technology (IT) resources known to be located within secure networks. As of August 16, 2003, for example, the MSBlast worm infected more than 350,000 hosts worldwide, demonstrating once again the ineffectiveness of current protection mechanisms.

Today, most anti-virus solutions run in software on end systems. To ensure an entire network is secure from known attacks, it is required that every host within the network be running the latest version of an operating systems and virus-protection software. Should any machine in the network not be fully up-to-date, or should the software on the end systems contain any security flaws, the security of the overall network can be compromised.

By inserting data scanning and filtering devices throughout a network, rather than just at the end systems, Internet worms and computer viruses can be quarantined to just the segment of the network where they are introduced. Such a system of intelligent gateway devices recognizes and blocks malware at localized levels to dramatically limit the spread of the worm or virus. To provide a complete solution, there is a need for devices which can scan data quickly, reconfigure the scanning devices to search for new attack patterns, and take immediate action when attacks occur.

II. Related Work

A common prerequisite for network intrusion detection and prevention systems is the ability to search for predefined signatures in network traffic flows. A virus or Internet worm can be detected by the presence of a string of bytes (for the rest of the paper, a string is synonymous with a signature) in traffic that passes through a network link.

Software-based scanners are not fast enough to monitor all traffic passing through a high-speed network link. Due to the sequential nature of code execution, software-based systems can perform only a limited number of operations within the time period of a packet transmission. Hardware-based systems, on the other hand, can make use of parallelism to perform deep packet inspection with high throughput [1].

Programmable Logic Devices (PLDs) can be used to perform regular expression matching functions in hardware [2][3]. In previous work, a platform with Field Programmable Gate Array (FPGA) technology was implemented to process Asynchronous Transfer Mode (ATM) cells, Internet Protocol (IP) packets, and Transmission Control Protocol (TCP) flows at OC48 (2.4 Gigabit/second) rates [4][5][6][7]. Several mechanisms were developed to perform exact matching and longest prefix matching for header fields [8][9][10]. An automated design flow was created to scan the payload traffic for regular expressions [11][12]. In addition, a Bloom filter was developed to enable large numbers of fixed-length strings to be scanned in hardware [13]. Lastly, web-based tools were developed to enable easy remote monitoring, control, and configuration of the hardware [14].

III. System Architecture

A. System Components

A complete system has been implemented to protect networks from Internet worm and virus attacks. The system is comprised of three interconnected components: a Data Enabling Device (DED), a Content Matching Server (CMS), and a Regional Transaction Processor (RTP). These systems work together to provide network wide protection.

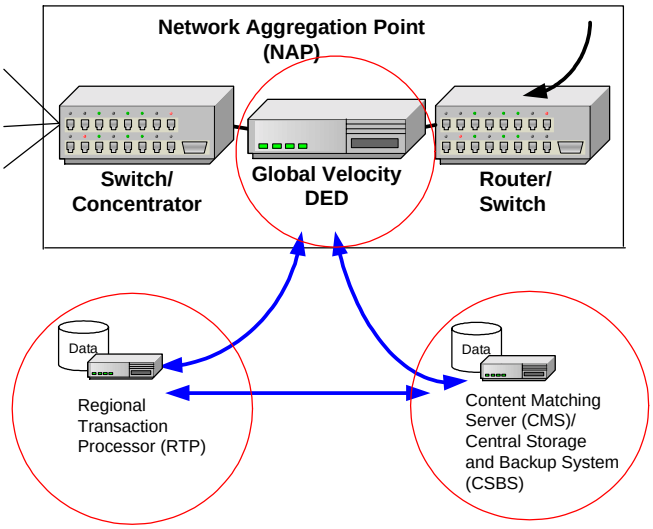

Figure 1: The system architecture includes a Data Enabling Device (DED), a Content Matching Server (CMS), and a Regional Transaction Processor (RTP)

Data Enabling Device (DED)

Packets in our system are scanned by the Data Enabling Device (DED). At the heart of the DED is the Field-programmable Port Extender (FPX). The FPX consists of a module implemented in FPGA hardware that scans the content of Internet packets at Gigabit per second rates. All of the packet processing operations are performed using reconfigurable hardware within a single Xilinx Virtex XCV2000E FPGA. A set of layered protocol wrappers parse the headers and payloads of packets using high-speed circuits implemented as combinatorial logic and state machines in the FPGA device. DEDs are installed at key traffic aggregation points of commercial, academic or governmental networks, as well as on the backbone.

Content Matching Server (CMS)

In order to reprogram the DEDs to search for new strings, a Content Matching Server (CMS) has been implemented. Custom circuits are compiled and synthesized on the CMS by an automated design flow. The CMS reads a table of Internet worm and virus signatures from a database, converts each into an optimized finite automata, instantiates parallel hardware to perform a data scanning function, embeds this hardware into logic that parses Internet protocol messages, synthesizes the entire circuit into logic gates, routes, places the circuit into a FPGA, and then reconfigures the hardware over the network.

Regional Transaction Processor (RTP)

The Regional Transaction Processor (RTP) is contacted by the DED when matching content is found to be passing through the network. The RTP consults a database to determine the action that the DED should take, such as to forward or block the traffic flow containing the sensitive data. The existing system maintains information about users, agents, properties, owners, and access rights in a MySQL database.

Common Gate Interface (CGI) scripts are used to provide a network administrator with an easy-to-use, web-based interface to both the CMS and RTP. A single RTP can be used to remotely coordinate the activities of up to 100 DEDs. RTPs can be co-located on the same site as the DEDs and managed by a local site administrator, or may be located across a network and administered by a centralized authority.

B. How the System Works

The system acts upon each packet of data as it moves in and out of the network, as shown in Figure 2. Whenever a new virus outbreak occurs, an administrator or an automated process adds the signature of the malware to the database table on the Content Matching Server (CMS). The CMS then programs the Data Enabling Device (DED) to scan Internet traffic for signatures that appear in the payload of messages The DED then scans the live Internet traffic for the targeted signature. Whenever matching content is found, the DED either blocks the traffic or allows it to pass. In either case, it simultaneously generates a warning message to the recipient. The option of whether to block or transmit data is determined by the policy of the network administrators. False positives are minimized by using long and distinct strings that are highly unlikely to appear in normal traffic content.

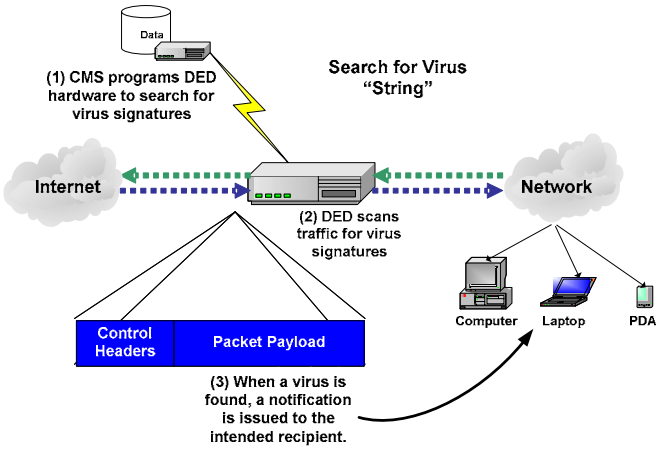

Figure 2: The CMS programs the DED to detect worm and virus signatures, then the DED notifies end users or administrators when the signature is found

C. Typical Network Architecture

In a typical installation, such as would be found in a large military network, Data Enabling Devices (DEDs) are installed at Network Aggregation Points (NAPs). Traffic flows from end-system networks (LANs, remote users, or wireless LAN base stations) are concentrated into a single high-speed link that is then fed into a router. The DED is inserted into the network at the point where traffic would otherwise simply be routed back to other networks or to the Internet. So long as at least one DED is positioned along the path between any two endpoints (shown as ovals in Figure 3), the virus signature will be detected.

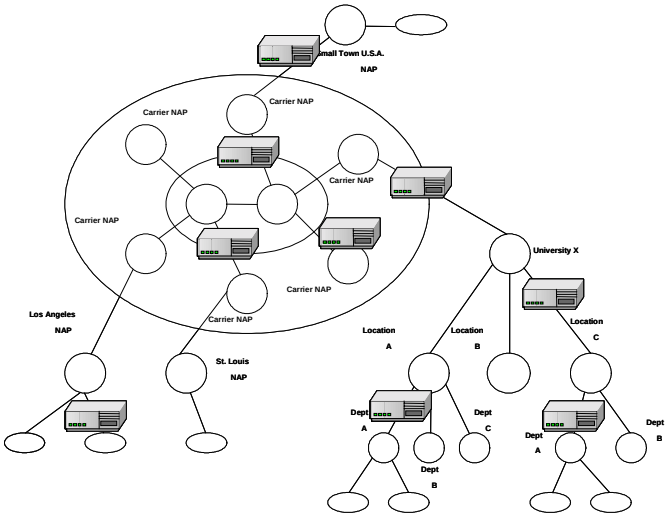

Figure 3: Data Enabling Devices (DEDs) which are installed at selected aggregation points provide protection against worms and viruses spreading over the Internet.

Internet protocol processing circuits, content matching modules, and automated design tools facilitate the implementation and timely updating of network security applications in reconfigurable hardware. The system allows for the immediate blocking of known viruses and may be rapidly reprogrammed to recognize and block new threats. These upgrades are system-driven, and are not dependant upon actions by the end users to assure that the protection remains up to date.

IV. Reprogrammable Logic

Programmable Logic Devices allow the system to achieve high performance. This section describes the components of the DED, details the implementation of the FPX, and describes how FPGA circuits are used to scan packets.

A. Configuration of a DED

A DED contains two network line cards (one for each side of the network being protected), a backplane, and two or more FPX cards. One FPX card is used to process Internet control packets that are sent over the network. Other cards are used to perform content scanning, and may be stacked between the line cards and the backplane. Physically, the DED is housed in a 19" rack-mount case. Up to four FPX cards can be installed in a 2U case and eight may be installed in a 3U case.

Figure 4: Rackmount case holds DED with FPX cards

B. FPX Platform

The Field-programmable Port Extender (FPX) card implements the core functionality of the DED. In order to provide sufficient space to store the state of multiple traffic flows, an FPX can be equipped with up to 1 Gigabyte of SDRAM and 6 Megabytes of pipelined Zero-Bus-Turnaround (ZBT) SRAM. The network interfaces connect to line cards, including Gigabit Ethernet and/or several types of ATM interfaces. A photo of the FPX is shown in Figure 5 and a diagram of the logical circuit is shown in Figure 6.

Figure 5: The FPX platform contains three SRAMs, two banks of SDRAM, two multi-Gigabit/second network interfaces, and two large FPGAs: The Reprogrammable Application Device (RAD) and the Network Interface Device (NID), implemented with a Xilinx Virtex XCV2000E and XCV600E, respectively.

Each FPX card contains two FPGAs, five banks of memory and two high-speed (OC-48 rate) network interfaces. On the FPX, one FPGA, called the Network Interface Device (NID), is used to route individual traffic flows through the device and process control packets, while the other FPGA, called the Reconfigurable Application Device (RAD), is dynamically reconfigured over the network to perform customized packet processing functions. The NID allows bitstreams to be received over the network and programmed into the RAD using the FPGAs [4].

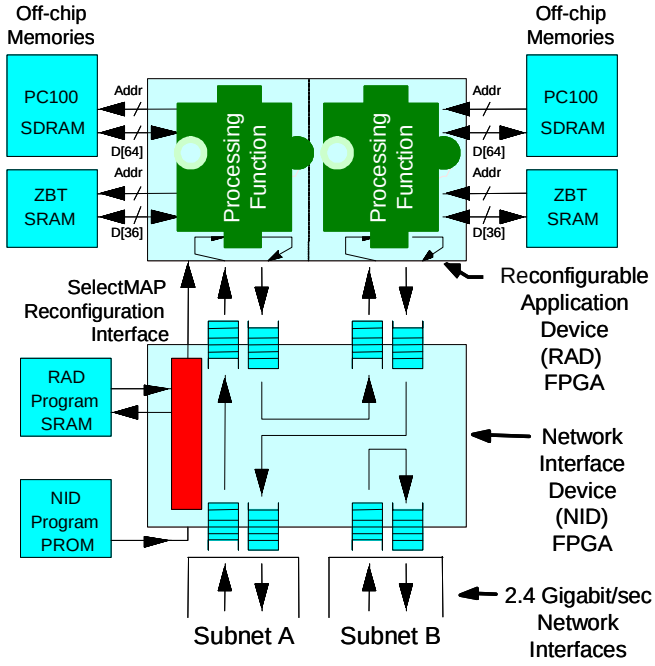

Figure 6: Block diagram of the FPX, showing how data processing functions are implemented as modules on the RAD and the traffic routing and reconfiguration functions are performed on the NID.

C. Protocol Processing Wrappers

The FPX can be used to process traffic on a wide variety of networks, including Ethernet and Asynchronous Transfer Mode (ATM). Line cards have been developed that interface the DED to both Gigabit Ethernet and ATM networks.

- For ATM networks, a Synchronous Optical NETwork (SONET) line card adapter interfaces to the physical network. Virtual paths and circuits can be specified that identify which traffic flows are to be scanned. Protocol wrappers implemented in hardware are used to reassemble individual ATM cells into complete Adaptation Layer 5 (AAL5) frames in hardware.
- For Gigabit Ethernet, the FPX has a GBIC to interface to fiber-based or copper-based network ports. This allows the network interface to use category 5 cable, short-haul optics, or long-haul optics. The Gigabit Ethernet line card extracts the data from MAC frames. It allows Ethernet packets to be searched that appear on LAN or on a specific 802.1q VLAN to be identified and passed through to the FPX.

Internet headers can be processed in many ways, such as with ternary content addressable memories [8], longest-prefix matching tries [9], or Bloom-based header-matching circuits [10]. Protocol wrappers have been developed to parse the Internet Protocol (IP) packets and Transmission Control Protocol (TCP) flows directly in hardware.

- Layered Internet Protocol (IP) wrappers break out the header fields that include the protocol field, source IP address, and destination IP address. The IP wrappers also compute the checksums over the header and process the Time-to-Live field [5].
- For User Datagram Protocol Internet Protocol (UDP/IP) traffic, UDP wrappers break out the fields for the UDP source and destination ports. The wrappers also perform checksums over the packet
- For Transmission Control Protocol Internet Protocol (TCP/IP) traffic, TCP wrappers reassemble flows that may consist of lost or re-ordered packets. They ensure that the higher-level protocols processing elements see a consecutively-ordered data stream [6]. The TCP wrapper, implemented in FPGA logic, reconstructs the flow of transmitted data by tracking sequence numbers of consecutive packets to provide a byte-ordered data stream to the content scanning engines. This means that even if a malware signature has been fragmented across multiple packets, it still will be detected and blocked. In order to maintain the state of multiple traffic flows, the system architecture has been designed to store the state of a TCP/IP flow in high capacity Synchronous Dynamic Random Access Memory (SDRAM). Given that each flow occupies 64 bytes of memory, one 512 Mbyte SDRAM (just under half of the memory available on the FPX) can track 8 million simultaneous traffic flows [7].

Higher-level protocol processing can be implemented at layers above the existing protocol wrappers.

- For web traffic, the payload processing wrapper will parse HTTP headers to perform filtering on URLs
- For email traffic, the payload processing wrapper will parse SMTP headers to block traffic to or from specific email addresses
- For peer-to-peer traffic, the payload processing wrapper will sit above the transport wrapper and scan content for signatures of specific files.

D. Signature Detection

Two types of modules have been developed to search for signatures: those that use finite automata to scan for regular expressions and those that use Bloom filters to scan for fixed-length strings. The number of regular expressions that can be

searched grows approximately linearly with the amount of the FPGA logic on the device [11][12].

The number of fixed-length strings that can be searched expands with the size of on-chip RAM that is available to the system. A Bloom filter implemented on a single FPX card allows a content scanning module to identify up to 10,000 different, fixed-length strings [13].

E. Performance

Both the finite automata and the Bloom filter scan traffic at speeds of up to 600 Mbps. By implementing four modules in parallel, the FPX can process data at a rate of 2.4 Gigabits per second using a single Xilinx Virtex 2000E FPGA. Figure 7 shows how four parallel sets of six Regular Expression (RE) automata are instantiated within the protocol wrappers.

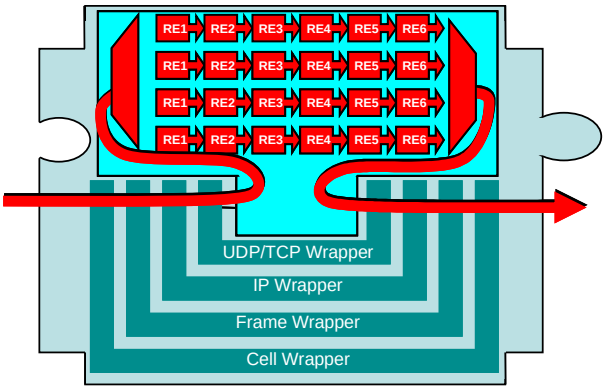

Figure 7: A complete network module contains the layered protocol wrappers, several Regular Expression (RE) finite automata (one for each string), and four parallel sets of REs to enable high throughput.

The FPX uses parallel hardware to maintain full-speed processing of packets. Data throughput is unaffected by the number of terms that are subject of the search. So long as the working set of signatures fits into the resources on the FPGA and the circuit synthesizes to meet the necessary timing constraints, the throughput remains constant. This is significantly different than software-based solutions, which slow down as the CPU is required to search for more terms.

The DED can achieve full throughput for both minimum length IP packets (40 bytes), maximum length Ethernet packets (1500 bytes), and all sizes in between.

The advantages of hardware over software are a result of the inherent parallel capabilities of hardware, which is capable of conducting multiple content matches simultaneously. Systems that process packets in software generally cannot keep up with the full throughput of the data link under high throughput. Once the processor becomes fully utilized, software-based systems become unable to process all of the traffic that passes through the network node. The result is that they process only a fraction of the packets, and thus the probability that a packet is matched decreases with higher throughput, as shown in Figure 8.

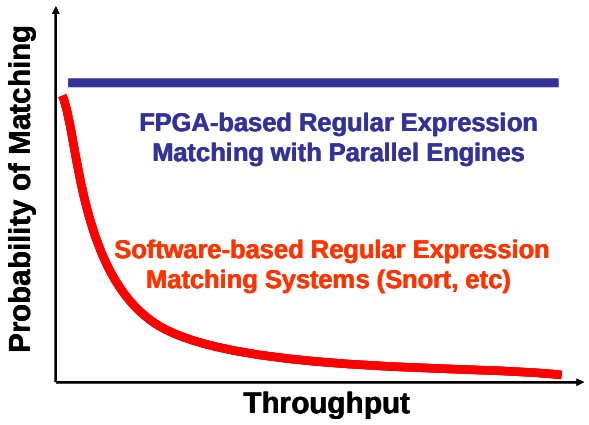

Figure 8: By performing the network scanning with parallel hardware, all packets can be examined even at high throughput. For software, the probability of matching all packets decreases because the CPU becomes overloaded with higher network throughput.

F. Automated Design Flow

To enable rapid deployment of regular expression-matching circuits for the DED, a fully automated design flow was developed. The complete design flow is detailed in Figure 9. The process begins when a new signature is added to the database on the front end of the CMS. Next, the CMS reads the signatures from the database table, creates an optimized finite automaton for each signature, then instantiates parallel scanning circuits that fit inside the layered protocol wrappers, as was shown in Figure 7. Next, the dynamically created VHDL is synthesizing into logic using the Synplicity CAD tool. Next, constraints for inputs and outputs (I/O) pins are given to map the circuit into the RAD. This circuit is then placed and routed with Xilinx FPGA tools and a bitstream is generated. The resulting module is then deployed to remote hardware using the NCHARGE tools [14]. Most of the time required by the CMS is consumed by the tools that route and place the FPGA. Using an AMD Athlon 2400MP to perform all of the steps shown in Figure 9, a new, 2-Million gate-equivalent packet scanning module can be created and deployed in 9 minutes.

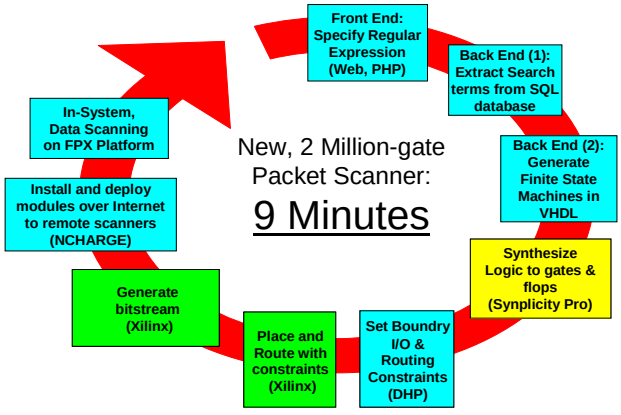

Figure 9: An automated design flow creates FPGA circuits for the Data Enabling Device (DED) in minutes

G. Web-based Control and Configuration Interface

A graphical user interface allows new search strings to be entered with a Graphical User Interface (GUI) from a web client. A database is used on the backend server to track the tables of search strings, their corresponding description, the owner associated with each of the content items, and a risk value assigned to each virus. When the 'Build' button of Figure 10 is pressed, the new design flow is run and the circuit is deployed on the remote DED.

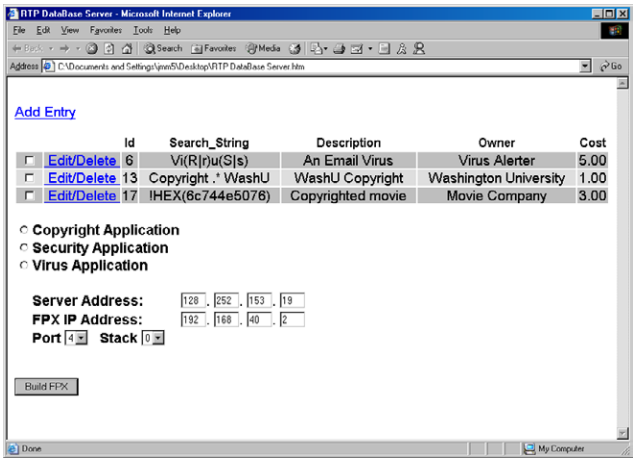

Figure 10: A Graphical User Interface (GUI) allows new strings to be entered from a web client and can start the process of building a new FPX circuit

V. End-System Applications

A. Passive Virus Protection

The system is designed to provide virus protection in either the passive or active mode. In both modes, the DED uses the FPGA hardware to scan the packets for the signature of specific malware.

In the passive mode, the DED will detect a virus signature embedded in traffic and immediately generate a warning prompt, shown in Figure 11, to the recipient, alerting them to the presence of the infected traffic. Similar prompts also may be generated to system and security administrators to alert them of the potential infection.

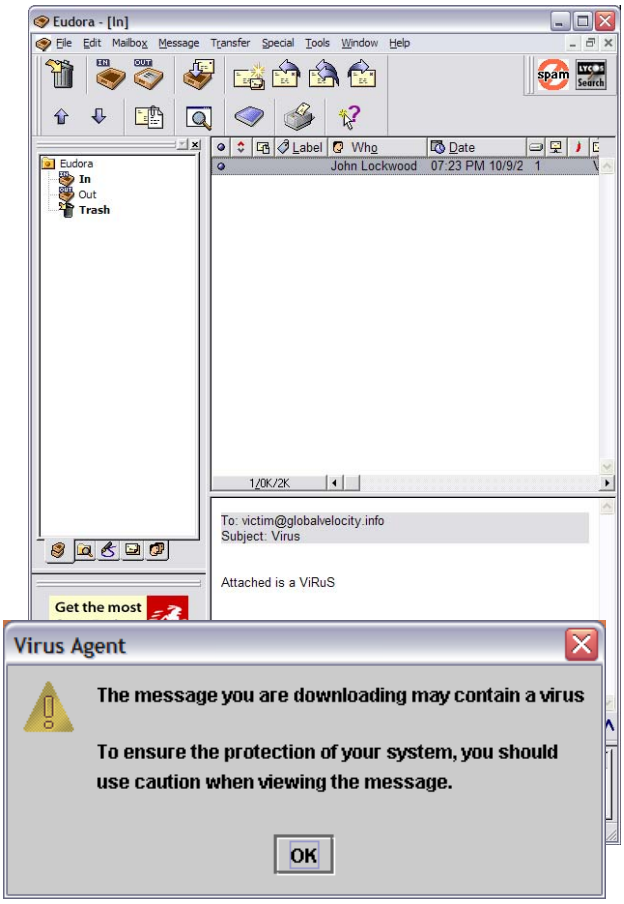

Figure 11: For passive virus protection, the DED generates a warning that the content being transferred over the network may contain a virus. The traffic itself is still allowed to be sent over the network.

B. Active Virus Protection

In the active mode, the DED will detect a virus signature within network traffic and block its transmission. The system will generate a warning message to the recipient, shown in Figure 12, explaining why the transmission was blocked, and alerting them to potential danger the message represents. Unless and until the intended recipient responds and deletes the incoming message, no further traffic will be allowed through the DED. A similar warning message can be generated to the system and security administrators.

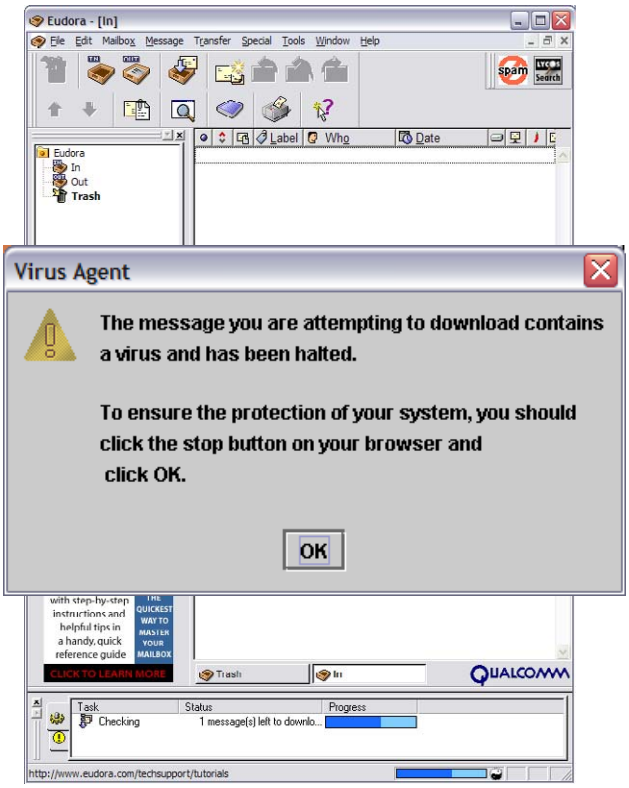

Figure 12: For active virus protection, the DED blocks traffic passing through the network that contains virus-infected messages. As shown in the lower status bar of the Eudora mail client, the end system never receives the infected message.

C. ADDITIONAL APPLICATIONS

The system described here is an effective tool against the spread of computer viruses and worms The system's component devices are also, capable of accomplishing a far wider range of security-related applications, including data security, copyright-protection and transaction documentation and accounting.

For example, because the DED can scan the content of traffic moving both directions – into and out of networks – it can easily be configured to detect the unauthorized release of confidential, classified or otherwise sensitive data, and block the release before it occurs. Military organizations could use the system to scan for classified documents passing through a network, and block them before they are transmitted out of a secure network; healthcare providers could use the system to assure compliance with privacy regulations such as the Health Insurance Portability and Accountability Act; manufacturers could utilize the system to protect against the release of proprietary product designs or strategic plans. Corporations may use the systems to assure that employees are not misusing their networks to download unapproved information, including pornography or other inappropriate data.

Colleges and universities will find the system useful in regulating the abuse of their high-speed networks for unapproved peer-to-peer transmission of copyright-protected music, motion pictures or software.

Companies and governmental agencies will find value in the system's ability to track, document and manage financial transactions, when tied to a specialized accounting system. For example, governmental agencies required to administer trust funds for specific individuals or groups could use the Intelligent Gateway to assure up-to-the-minute accounting for a wide variety of receipts and dispersals.

When viewed as a utility, the system can serve as a valuable stand-alone asset, since only a single DED is required to block viruses, protect against the release of sensitive data, and halt the unauthorized use of specific networks for the hosting and downloading of copyright-protected works.

As more of the systems are installed, the cumulative ability of the comprehensive network of systems is enhanced, in several ways. These include facilitating the streamlining and simplification of e-commerce transactions, by moving the point of purchase for goods and services to local Internet Service Providers and other network aggregation points, in essence, bringing the retailer to a computer user's home or office. That rethinking of the e-commerce model, the relocation of the point of purchase into local communities, may provide an opportunity for nexus.

Because the hardware is easily and remotely reconfigurable, the utility of the system is limited only by the vision and imagination of its users.

VI. CONCLUSIONS

A system has been developed that not only blocks the spread of Internet worms and computer viruses, but also has utility for a range of other applications, including data security, copyright protection and the documentation and management of digital transactions. This system uses programmable logic devices to scan Internet traffic for malware at high speeds. Malware is identified by signatures that may consist of either fixed strings or regular expressions. Through the use of layered protocol wrappers, application-level Internet traffic flows can be tracked, even for signatures that span multiple packets. An automated design flow allows new circuits to be rapidly deployed to protect the network against new attacks. The FPX platform allows these new circuits to be rapidly deployed into the Internet.

VII. . BIBLIOGRAPHIC REFERENCES

[1] J. W. Lockwood, "Evolvable Internet Hardware Platforms", *Evolvable Hardware Workshop*, Long Beach, CA, USA, July 12-14, 2001, pp. 271-279.

[2] R. Sidhu and V. K. Prasanna. "Fast Regular Expression Matching using FPGAs", *Field-Programmable Custom Computing Machines (FCCM)*, Rohnert Park, CA, USA, Apr. 2001.

[3] R. Fanklin, D. Caraver, and B. Hutchings. "Assisting network intrusion detection with reconfigurable hardware," *Field Programmable Custom Computing Machines (FCCM)*, Napa, CA, USA, Apr. 2002.

[4] J. W. Lockwood, N. Naufel, J. S. Turner, and D. E. Taylor, "Reprogrammable Network Packet Processing on the Field Programmable Port Extender (FPX)," *ACM International Symposium on Field Programmable Gate Arrays (FPGA)*, pages 87–93, Monterey, CA, USA, Feb. 2001.

[5] F. Braun, J. W. Lockwood, M. Waldvogel, "Layered Protocol Wrappers for Internet Packet Processing in Reconfigurable Hardware", *IEEE Micro*, Vol 22, pp. 66-74, Feb. 2002.

[6] D. V. Schuehler and J. W. Lockwood. TCP-Splitter: "A TCP/IP Flow Monitor in Reconfigurable Hardware", *Symposium on High Performance Interconnects (HotI)*, pages 127–131, Stanford, CA, USA, Aug. 2002.

[7] D. V. Schuehler, J. Moscola, and J. W. Lockwood, "Architecture for a Hardware Based, TCP/IP Content Scanning System", *Symposium on High Performance Interconnects (HotI)*, Stanford, CA, USA, pp. 89-94, Aug. 2003.

[8] J. W. Lockwood, C. Neely, C. Zuver, J. Moscola, S. Dharmapurikar, D. Lim, "An Extensible, System-On-Programmable-Chip, Content-Aware Internet Firewall", *Field Programmable Logic and Applications (FPL)*, Lisbon, Portugal, Sep. 2003.

[9] D. E. Taylor, J. S. Turner, J. W. Lockwood, T. S. Sproull, D. B. Parlour, Scalable IP Lookup for Internet Routers, IEEE Journal on Selected Areas in Communications (JSAC), Vol. 21, No. 4, May 2003, pp. 522-534

[10] S. Dharmapurikar P. Krishnamurthy D. E. Taylor, "Longest Prefix Matching Using Bloom Filters", SIGCOMM, Sep. 2003.

[11] J. Moscola, J. Lockwood, and R. P. Loui. "Implementation of a Content-Scanning Module for an Internet Firewall," *Field-Programmable Custom Computing Machines (FCCM)*, Napa, CA, USA, Apr. 2003.

[12] J. Moscola, M. Pachos, J. W. Lockwood, R. P. Loui, "Implementation of a Streaming Content Search-and-Replace Module for an Internet Firewall", *Symposium on High Performance Interconnects (HotI)*, Stanford, CA, USA, pp. 122-129, Aug. 2003.

[13] S. Dharmapurikar, P. Krishnamurthy, T. Sproull, J. W. Lockwood, "Deep Packet Inspection Using Parallel Bloom Filters", *Symposium on High Performance Interconnects (HotI)*, Stanford, CA, USA, pp. 44-51, Aug. 2003.

[14] T. Sproull, J. W. Lockwood, D. E. Taylor, "Control and Configuration Software for a Reconfigurable Networking Hardware Platform", *IEEE Symposium on Field-Programmable Custom Computing Machines, (FCCM)*, Napa, CA, USA, April 24, 2002