Application of Hardware Accelerated Extensible Network Nodes for Internet Worm and Virus Protection

John W. Lockwood*,
James Moscola,
David Reddick,
Matthew Kulig, and
Tim Brooks

Applied Research Laboratory
Washington University in Saint Louis
1 Brookings Drive, Campus Box 1045
St. Louis, MO 63130 USA

Global Velocity
Bandwidth Exchange Building
210 North Tucker Blvd., Suite 315
St. Louis, MO 63101 USA

{lockwood, jmm5}@arl.wustl.edu,
{dreddick, mkulig, tbrooks}@globalvelocity.info

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

Abstract. Today's crucial information networks are vulnerable to fast-moving attacks by Internet worms and computer viruses. These attacks have the potential to cripple the Internet and compromise the integrity of the data on the end-user machines. Without new types of protection, the Internet remains susceptible to the assault of increasingly aggressive attacks. A platform has been implemented that actively detects and blocks worms and viruses at multi-Gigabit/second rates. It uses the Field-programmable Port Extender (FPX) to scan for signatures of malicious software (malware) carried in packet payloads. Dynamically reconfigurable Field Programmable Gate Array (FPGA) logic tracks the state of Internet flows and searches for regular expressions and fixed-strings that appear in the content of packets. Protection is achieved by the incremental deployment of systems throughout the Internet.

* This research was supported by a grant from Global Velocity. John Lockwood is a co-founder and consultant for Global Velocity and an Assistant Professor at Washington University in St. Louis. Washington University and John Lockwood may receive income based on a license of related technology by the University to Global Velocity.

1 Introduction

Computer virus and Internet worm attacks are pervasive, aggravating, and expensive, both in terms of lost productivity and consumption of network bandwidth. Attacks by Nimba, Code Red, Slammer, SoBig.F, and MSBlast have infected computers globally, clogged large computer networks, and degraded corporate productivity [1]. It can take weeks to months for Information Technology staff to sanitize infected computers throughout a network after an outbreak. The direct cost to recover from just the 'Code Red version two' worm alone was $2.6 billion.

In much the same way that a human virus spreads between people that come in contact, computer viruses and Internet worms spread when computers communicate electronically. Once a few systems are compromised, they proceed to infect other machines, which in turn quickly spread the infection throughout a network [2]. As is the case with the spread of a contagious disease like SARS, the number of infected computers will grow exponentially unless contained. Computer systems spread contagion much more quickly than humans because they can communicate nearly instantly over large geographical distances.

"The Blaster worm infected over 400,000 computers in less than five days. In fact, about one in three Internet users are infected with some type of virus or worm every year. The speed at which worms and viruses can spread is astonishing. What's equally astonishing is the lethargic pace at which people deploy the patches that can prevent infection in the first place", Congressman Adam Putnam said recently when he opened a congressional hearing [3].

1.1 Malware

Malicious software (malware) can propagate as a computer virus, an Internet worm or a hybrid that contains elements of both. Viruses spread when a computer user downloads unsafe software, opens a malicious attachment, or exchanges infected computer programs over a network. An Internet Worm spreads over the network automatically when malicious software exploits one or more vulnerabilities in an operating system, a Web server, a database application, or an email exchange system.

Malware can appear as a virus embedded in software that a user has downloaded. It also can take the form of a trojan that is embedded in what appears to be benign freeware. Alternatively, it can spread as content attached to an email message, as content downloadable from a website, or in files transferred over peer-to-peer systems. Modern attacks typically use multiple mechanisms to execute their attacks. Malware can spoof messages to lure users to submit personal financial information to cloaked servers. In the future, malware is likely to spread much faster and do much more damage [4].

1.2 Weakness of End-System Protection

Today, most anti-virus solutions run in software on end systems. To ensure that an entire network is secure from known attacks, some system administrators mandate that every host within the network: (1) run only trusted Operating Systems, system tools, and application software; (2) have a full suite of virus-protection software loaded; (3) have the latest updates and patches installed for all of the programs that might run on the machine; and (4) be carefully configured to guard against running unauthorized software. Should any machine in the network be missed, the security of the overall network can be compromised.

It is difficult for companies, universities, and government agencies to maintain network-wide security. Most Internet worms and viruses go undetected until they cause harm on an end-user's computer. Placing the burden of detection

on the end user is neither efficient nor trustworthy because individuals tend to ignore warnings about installing new protection software and the latest security updates. New vulnerabilities are discovered daily, but not all users take the time to download new patches the moment they are posted. It can take weeks for an IT department to eradicate old versions of vulnerable software running on end-system computers.

A recent Gartner study predicts that by the year 2005, 90 percent of cyber attacks will attempt to exploit vulnerabilities for which a patch is available or a solution known. But systems are not always patched immediately, and anti-virus programs are not kept up to date. "System administrators are often times overwhelmed with simply maintaining all the systems they have responsibility for overseeing. Challenges that organizations face in maintaining their systems are significant: with an estimated 4,000 vulnerabilities being discovered each year, it is an enormous challenge for any but the best-resourced organizations to install all of the software patches that are released by the manufacturer. Not only is the sheer quantity of patches overwhelming for administrators to keep up with, but patches can be difficult to apply and also have potentially unexpected side effects on other system components that administrators must then evaluate and address. As a result, after a security patch is released, system administrators often take a long time to fix all their vulnerable computer systems. Obviously, small organizations and home users, who lack the skills of system administrators, are even less likely to be able to keep up with the flow of patches" said the congressman [3].

1.3 A Global Threat

Due to the global expansion of high speed networks and the proliferation of a dominant, monolithic operating system, a large portion of the Internet can be easily and quickly subverted. For SoBig.F, more than 1 million computers were infected within the first 24 hours, 100 million systems within the first five days, and an excess of 200 million computers within a week. The virus accounted for 70 percent of all email traffic on August 20, 2003.

Existing firewalls that examine only the packet headers do little to protect against many types of attack. Many new worms transport their malware over trusted services and cannot be detected without examining the payload. Intrusion Detection Systems (IDSs) that scan the payload can detect malware, but do nothing to impede the attack because they only operate passively. An Intrusion Prevention System (IPS), on the other hand, can intervene and stop malware from spreading. The problem is that current IPS devices cannot keep pace with the volume of traffic that transits high-speed networks. Existing systems that implement IPS functions in software limit the bandwidth of the network and delay the end-to-end connection.

1.4 A Proposed Solution with Intelligent Networking Nodes

There is a need for devices which can scan data quickly, reconfigure to search for new attack patterns, and take immediate action when attacks occur. By processing the content of Internet traffic in real-time within an extensible network, data containing computer viruses or Internet worms can be detected and prevented from propagating. By inserting a few data scanning and filtering devices at a few key Network Aggregation Points (NAPs) enables Internet worms and computer viruses to be quarantined to the subnetworks where they were introduced. Such a system of intelligent gateway devices recognizes and blocks malware locally to dramatically limit the spread of the worm or virus globally.

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 [5]. The result is an intelligent gateway that provides Internet worm and virus protection in both local and wide area networks.

According to the director of CERT, "It is critical to maintain a long-term view and invest in research toward systems and operational techniques that yield networks capable of surviving attacks while protecting sensitive data. In doing so, it is essential to seek fundamental technological solutions and to seek proactive, preventive approaches, not just reactive, curative approaches" [6].

2 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. Such signatures can be loaded into the system manually by an operator or automatically by a signature detection system.

2.1 Obtaining Signatures

There are many ways to obtain malware signatures. One way to learn about a new Internet worm or computer virus is to participate in group forums that share information about new attacks. When a new virus is encountered, information about it is collected, analyzed, and reported to the group. Agencies like the CERT Coordination Center provide alerts regarding potential threats and provide information about how to avoid, minimize, or recover from the damage [7]. Incident reports on vulnerabilities are collected and distributed once they are found. This data can be used by others to avoid and recover from an attack. New methods of detecting outbreaks can streamline the recognition and analysis of new threats and shorten the time needed to obtain a new signature.

Honeynets and Honeypots One new way that worms and computer viruses can be automatically detected is with a Honeypot or a Honeynet. Honeynets gather information by monitoring all traffic going to and from a non-production network. All captured activity is assumed to be unauthorized or malicious. Any connection initiated inbound to a Honeynet is most likely a probe, scan, or attack. Data that is captured is analyzed to gain insight into threatening activities. New tools can be discovered, attack patterns determined, and attacker motives studied by monitoring and logging all activities within the Honeynet [8].

Unlike a Honeypot, a Honeynet is an entire network of systems that runs real applications and services. An attacker can interact with operating systems and execute tools on what appears to be a legitimate production network, but is not. It is through this extensive interaction that information on threats is obtained and analyzed. Signatures are obtained from captured traffic that is determined to be malware. These signatures, in turn, can be used to program a data scanning device to block attacks into real systems.

Traffic Analysis Another automated method for detecting new worms is based on traffic characteristics. The method tracks traffic to detect highly repetitive packet content sent from an increasing population of sources to an increasing number of destinations. The method generates content signatures for the worm without any human intervention. The method can quickly identify the signatures of new worms [9]. As with a Honeynet, these signatures can be used by the system we propose to block worms from attacking other real systems.

2.2 Signature Scanning Systems

Once a signature is found, an IDS or IDP can use this signature to block traffic containing infected data from spreading throughout a network. To perform this operation on a high-speed network, the signature scanning and data blocking must operate quickly.

Software Scanners Software-based scanners are not capable of monitoring all traffic passing through fast network links. 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.

A comparative analysis of a variety of systems running the Snort [10] rule-based NIDS sensor reveals that most general-purpose computer systems are inadequate as NIDS sensor platforms even for moderate-speed networks. The analysis shows that factors that include the microprocessor, operating system, and main memory bandwidth and latency all limit the performance achievable by a NIDS sensor platform. General-purpose computers, including the Intel Pentium-III and Pentium-4, were found to be inadequate to act as sensors even on a 100 Mbit per second network link. The best-performing system could support only a maximum of 720 header rules without losing packets. For larger numbers of rules, a significant percentage of packets are dropped, thus degrading the NIDS effectiveness in detecting security breaches [11].

Hardware Scanners Hardware-based systems can make use of parallelism to perform deep packet inspection with high throughput. Tradeoffs can be made between hardware, software, network processor, and FPGA technologies. [12]. There is a limited but growing number of commercial systems that offer high-speed virus blocking capabilities that use hardware to process packets. FortiNet, for example, uses Application Specific Integrated Circuits (ASICs) to provide virus protection. It is not clear from its documentation, however, how much flexibility in rule processing is possible [13]. TippingPoint Technologies also offers a range of hardware devices that it calls UnityOne systems. At the core of UnityOne is a security-specific processor called the Threat Suppression Engine (TSE) that enables intrusion prevention at multi-gigabit speeds. The TSE is a blend of ASICs and network processors, and detects known threats and anomalies in network traffic [14]. Another company, Packeteer, monitors traffic flows on slower networks. It primarily is used to shape bandwidths to streamline traffic. The current maximum throughput is listed at 200 Mbps [15]. IntuVert Networks, which Network Associates purchased in April 2003, lists its throughput speed at up to 2 Gbps [16].

Reconfigurable Scanners Programmable Logic Devices (PLDs) provide the flexibility and performance to scan for regular expressions within a high-speed network [17] [18]. 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 [19] [20] [21] [22]. Several mechanisms were developed to perform exact matching and longest prefix matching for header fields [23] [24] [25]. An automated design flow was created to scan the payload traffic for regular expressions [26] [27]. In addition, a Bloom filter was developed to enable large numbers of fixed-length strings to be scanned in hardware [28]. Lastly, web-based tools were developed to enable easy remote monitoring, control, and configuration of the hardware [29].

3 System Architecture

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). A diagram of the DED, CMS, and RTP appears at the right side of Figure 1. These systems work together to provide network-wide protection.

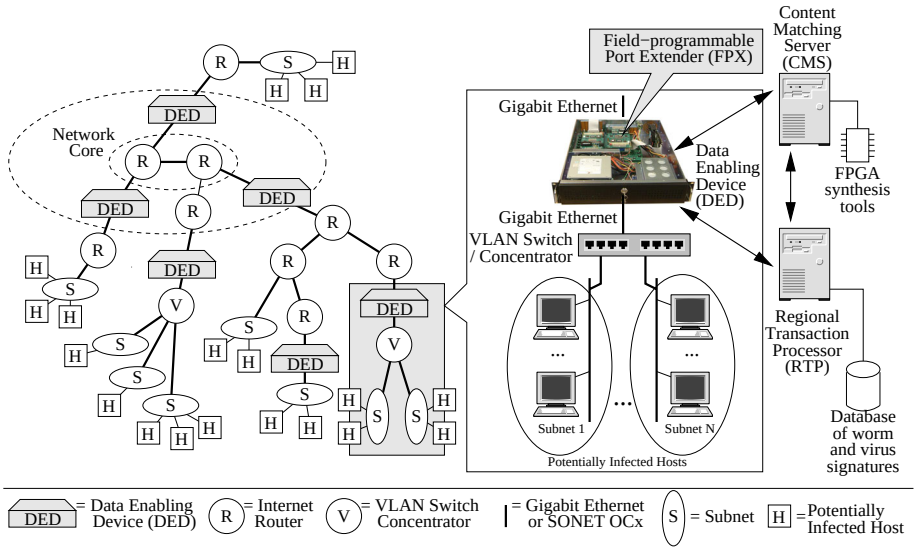

Fig. 1. Example topology of a Network Aggregation Point (NAP) with DEDs added to provide worm and virus protection

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 can be installed incrementally at key traffic aggregation points of commercial, academic or governmental networks, as well as on the network core. The more DEDs that are deployed, the better is the granularity of protection between different subnetworks.

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 reconfigures the hardware.

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 the database to provide detailed information about the reason that a certain data flow was blocked. The existing RTP maintains information about users, agents, properties, owners, and access rights in a MySQL database. Common Gateway Interface programs provide a network administrator with an easy-to-use, web-based interface to modify the database tables and to run the scripts that build new hardware. A single RTP can remotely coordinate the activities of up to 100 DEDs. RTPs can be co-located on the same site as the DEDs and be managed by a local site administrator, or be located remotely and be administered by a trusted authority.

4 System Operation

The system acts upon each packet of data as it moves in and out of the network. 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 synthesizes a new FPGA circuit and reconfigures the FPGA on the Data Enabling Device (DED) to scan Internet traffic for the updated signatures. A 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. The DED uses parallel hardware circuits to scan all bytes of the traffic for the targeted signature. Whenever matching content is found, the DED generates a warning message that is forward to the intended recipient of the message. The system operates in either the active or passive mode, depending upon the policy set by the network administrator. In active mode, it blocks the infected traffic from passing; while in passive mode, it allows the data to go through.

To detect an Internet worm, the system would be programmed to look for a signature that contained a portion of the binary executable program that performs a core function of the worm. Inadvertent matches can be nearly eliminated by using long and distinct strings that are highly unlikely to appear in normal traffic content. It is important to select a signature which does not contain content commonly found in Internet transmissions. For random data with a signature length of 16 bytes ($16 * 8 = 128$ bits), the odds that a signature will find a match in random traffic is only one in $2^{128} = 3.4 * 10^{38}$. On a 1 Gigabit/second link, this corresponds to finding a match only once in $(3.4 * 10^{38}$ bytes $) * (8$ bits/byte $)/(1 * 10^{9}$ bits/sec $) = 2.72 * 10^{30}$ seconds . This is equivalent to finding a stray match just once in $2.72 * 10^{30}$ sec /(60 sec/min $*$ 60 min/hr $*$ 24 hr/day $*$ 365 day/year) = 86 billion billion years, which is an amount of time likely to be far greater than the lifetime of the Internet.

In a typical installation, as shown at the left of Figure 1, Data Enabling Devices (DEDs) are installed at Network Aggregation Points (NAPs). Traffic flows from end-system hosts attached to subnets are concentrated into high-speed links that are 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, the virus signature will be detected.

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.

5 Reprogrammable Logic on the FPX Platform

Programmable Logic Devices allow the system to achieve high performance. A DED contains two or more FPX cards, two network line cards (one for each side of the network being protected), and a backplane. 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.

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. Each FPX card contains two FPGAs, five banks of memory and two high-speed (OC-48 rate) network interfaces. On the FPX, one FPGA, the Network Interface Device (NID), is used to route individual traffic flows through the device and process control packets, while the other FPGA, 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 [12]. A diagram of the FPX is shown in Figure 2.

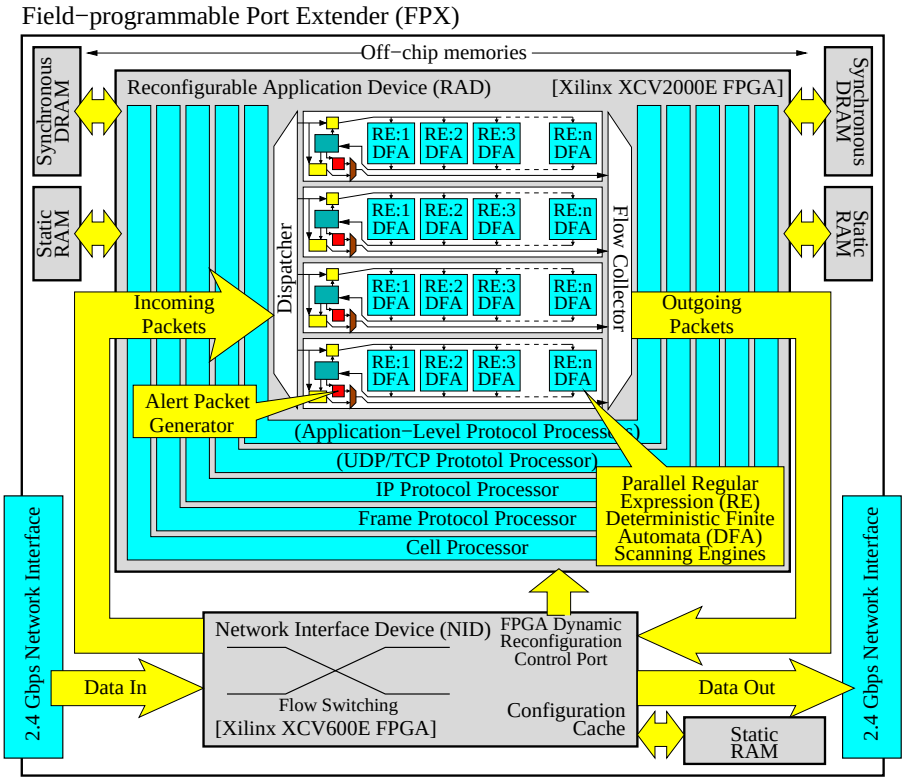

Fig. 2. Field-programmable Port Extender (FPX) with Protocol Wrappers and Regular Expression (RE) Deterministic Finite Automata (DFA) scanning engines

5.1 Protocol Wrappers

The FPX can be used to process traffic on a wide variety of networks. Line cards have been developed that interface the DED to both Gigabit Ethernet and Asynchronous Transfer Mode (ATM) networks. For Gigabit Ethernet, a GBIC is used to interface with either fiber or copper network ports. The Gigabit Ethernet line card extracts the data from MAC frames. VLANs can be used to identify which traffic flows packets should be processed. For ATM networks, a Synchronous Optical NETwork (SONET) line card interfaces to the physical network. Virtual paths and circuits determine which traffic flows are to be scanned. Protocol wrappers implemented in hardware reassemble individual ATM cells into complete Adaptation Layer 5 (AAL5) frames.

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 [20]. Internet headers can be processed in many ways, such as with ternary content addressable memories [23], longest-prefix matching tries [24], or with Bloom-based header-matching circuits [25].

For transport protocols, including the User Datagram Protocol (UDP) and Transmission Control Protocol (TCP), the wrappers track the source and destination port of the packet. The wrappers also compute checksums over the entire packet. 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 [30]. This means that even if a malware signature has been fragmented across multiple packets, it can still be detected and blocked. Synchronous Dynamic Random Access Memory (SDRAM) allows the state of multiple traffic flows to be tracked. By allocating 64 bytes of memory for each flow, one 512 Mbyte SDRAM can track 8 million simultaneous TCP/IP flows [22].

Higher-level protocol processing can be implemented above the transport-layer wrappers. For web traffic, a payload processing wrapper could parse HTTP headers to perform filtering on URLs. For email traffic, the payload processing wrapper could parse SMTP headers to block traffic to or from specific email addresses. For peer-to-peer traffic, the payload processing wrapper could scan for signatures of specific content.

5.2 Signature Detection

Many types of Internet traffic can only be classified by deep content inspection [31]. Dynamically reconfigurable hardware can perform content classification functions effectively. Unlike an ASIC, reconfigurable hardware can be optimized to search for new signatures. Two types of scanning modules have been developed to search for signatures on the FPX: those that use finite automata to scan for regular expressions [26] [27] and those that use Bloom filters to scan for fixed-length strings [28]. With a Bloom filter, a single FPX card can scan for up to 10,000 different, fixed-length strings.

In order to scan packet payloads for regular expressions, a matching circuit was implemented that dynamically generates finite automata. Regular expressions provide a shorthand means to specify the value of a fixed string, alternative substrings, and/or wildcards.

The architecture of a payload scanner with four parallel content scanners searching for n Regular Expressions, *RE:1-RE:n*, is illustrated in Figure 2. A match is detected when a sequence of incoming data causes a state machine to reach an accepting state. For example, to detect the presence of the SoBig.F Internet worm as it appears when encapsulated in a Mime64-encoded email, the expression: "**!HEX(683063423739)**" would be specified. In order to achieve

high throughput, parallel machines are instantiated. When data arrives, a *flow dispatcher* sends a packet to an available buffer which then streams data though the sequence of regular expression search engines. A *flow collector* combines the traffic into a single outgoing stream. The result of the circuit is the identification of which regular expressions were present in each data flow.

5.3 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. The parallel hardware enables the FPX 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 processor becomes fully utilized, meaning that 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.

5.4 Automated Design Flow and Web-based Control Interface

To rapidly generate hardware that searches for new regular expressions on the DED, a fully automated design flow was developed. The design flow begins when the CMS reads a set of signatures from its database. Finite automaton are dynamically created for each regular expression and are optimized to have a minimal number of states. Very-High-Speed Integrated Circuit (VHSIC) Hardware Description Language (VHDL) code is generated to instantiate parallel automaton that scan the packets passing through the layered Internet protocol wrappers. The VHDL code is synthesized into logic using the Synplicity computer aided design tool. Input and Output (I/O) pins of the circuit are mapped to I/O pins of the RAD. This circuit is then placed and routed with Xilinx tools and a FPGA bitstream is generated. The resulting bitstream is then deployed into a remote FPX platform using the NCHARGE tools [29]. An AMD Athlon 2400MP can create and deploy a completely new hardware circuit in 9 minutes. Most of that time is consumed by the tools that route and place the entire FPGA, which can be greatly reduced through use of partial reconfiguration [32].

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 to track the tables of search strings, their corresponding description, and a risk value assigned to each virus. When a single 'Build' button is pressed, the new design flow is run and the circuit is deployed on the remote DED [5].

6 End-System Applications

The virus protection works for many types of live Internet traffic. Network-wide protection is provided, so long as there is at least one DED in the path from the content sender to the recipient. The DED scans all packets going through the network link and recognizes those that contain a virus signature and acts on the transmission. This content can appear in web traffic, email messages, instant messages, or peer-to-peer transmissions. The system provides virus protection in passive or active modes. In both modes, the DED uses the FPGA hardware to scan the packets for malware signatures and act on their transmission.

In the passive mode, the DED will detect a virus signature embedded within network traffic and generate a UDP/IP packet that is forwarded to the recipient's computer. An application running on that computer, in turn, is directed to query the RTP to retrieve a full-text description that alerts them to the potential danger that the content represents. This warning message is currently presented to the user via a popup window from a Java application.

In the active mode, the DED not only detects a virus signature, but also blocks the flow of traffic. When a match is found, the DED hardware generates a UDP/IP packet that is sent to the intended recipient's network address. The machine that receives that message can query the RTP server using data in that message to retrieve a full-text description that explains why the content was blocked.

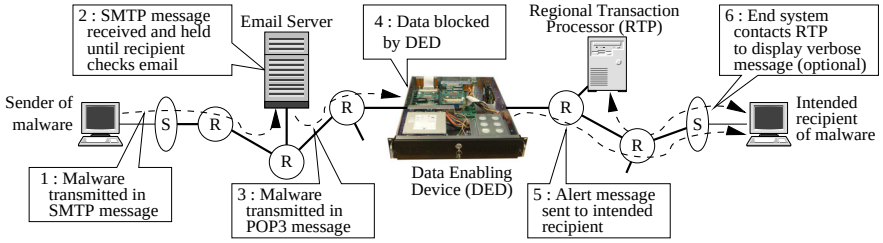

Fig. 3. Example that shows how malware can be blocked by a DED as it attempts to transit from an email server to an end-system

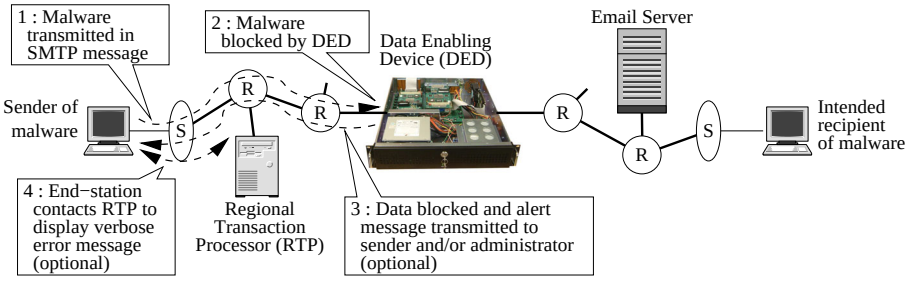

Fig. 4. Example that shows how malware can be blocked by a DED as it attempts to transit from the sender to an email server

If at least one DED is located along the path between an email server and the intended recipient of the malware, then an infected message can be blocked before it has a chance to reach that end-system. A system protected by a DED will receive an immediate notice whenever a virus-infected message targets the machine. For example, if a DED recognizes that a malware signature is being carried within a POP3 email message, then that inbound data is blocked and a notification is forwarded to the intended recipient's host. A diagram that shows these message exchanges is shown in Figure 3.

If at least one DED is located along the path between the sender and the targeted email server, then an infected message can be blocked before that message even has a chance to reach that email server. Further, the DED can generate a message that alerts the operator on the sending machine that some process on their machine attempted to transmit an infected message. A diagram that shows these message exchanges is shown in Figure 4.

Additional Applications The system described here is not only effective against the spread of computer viruses and worms, but can also perform many other functions. Because the system operates at Gigabit speeds and has near-zero latency, it can process live web traffic and scan data sent over peer-to-peer networks. For example, the DED can be configured to detect the unauthorized release of confidential, classified or otherwise sensitive data, and block its distribution. Corporations could scan for proprietary documents passing through a network, and block them before they leave a secure network; and healthcare providers could assure compliance with privacy regulations such as the Health Insurance Portability and Accountability Act. The flexibility of the system to be quickly and remotely reconfigured enables it to meet new needs and perform new tasks. Additional modules can be loaded into the FPX to encrypt and decrypt data, filter packets, and schedule transmission of data over a link to preserve Quality of Service [23].

7 Conclusions

An extensible networking system has been developed that not only blocks the spread of Internet worms and computer viruses, but also can be used to support a wide range of active networking applications. 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, with targeted data acted upon before it has an opportunity to damage networks. An automated design flow allows new FPGA circuits to be rapidly deployed to protect the network against new attacks.

References

1. E. Skoudis and L. Ziltser, *Malware: Fighting Malicious Code*. New Jersey: Prentice Hall, first ed., 2003.
2. D. Moore, C. Shannon, G. Voelker, and S. Savage, "Internet quarantine: Requirements for containing self-propagating code," in *IEEE INFOCOM*, (San Francisco, CA), Mar. 2003.
3. US Congressman Adam Putnam, Chairman, Subcommittee on Technology, Information Policy, Intergovernmental Relations and the Census, "Oversight Hearing: Opening statement, Worm and Virus Defense: How Can We Protect the Nations Computers from These Serious Threats." http://reform.house.gov/TIPRC/-Hearings/EventSingle.aspx?EventID=526, Sept. 2003.
4. V. Paxson, S. Staniford, and N. Weaver, "How to 0wn the internet in your spare time," in *Proceedings of the 11th Usenix Security Symposium*, Aug. 2002.
5. J. W. Lockwood, J. Moscola, M. Kulig, D. Reddick, and T. Brooks, "Internet worm and virus protection in dynamically reconfigurable hardware," in *Military and Aerospace Programmable Logic Device (MAPLD)*, (Washington DC), p. E10, Sept. 2003.
6. Richard D. Pethia, Director of CERT Coordination Center, "Subcommittee on Technology, Information Policy, Intergovernmental Relations and the Census: Oversight Hearing, Worm and Virus Defense: Viruses and Worms: What can we do about them." http://reform.house.gov/UploadedFiles/-Pethia_testimony_Sept2003-v7.pdf, Sept. 2003.
7. "CERT coordination center." `http://www.cert.org/`, 2003.
8. "Know your enemy: Honeynets." `http://www.honeynet.org/papers/honeynet`, Nov. 2003.
9. S. Singh, C. Estan, G. Varghese, and S. Savage, "The EarlyBird system for real-time detection of unknown worms." UCSD Tech Report CS2003-0761, Aug. 2003.
10. M. Roesch, "SNORT - lightweight intrusion detection for networks," in *LISA '99: USENIX 13th Systems Administration Conference*, (Seattle, Washington), Nov. 1999.

11. L. Schaelicke, T. Slabach, B. Moore, and C. Freeland, "Characterizing the performance of network intrusion detection sensors," in *Proceedings of the Sixth International Symposium on Recent Advances in Intrusion Detection (RAID 2003)*, Lecture Notes in Computer Science, (Berlin–Heidelberg–New York), Springer-Verlag, September 2003.
12. J. W. Lockwood, "Evolvable Internet hardware platforms," in *The Third NASA/DoD Workshop on Evolvable Hardware (EH'2001)*, pp. 271–279, July 2001.
13. FortiNet, "Product overview." `http://www.fortinet.com/products/`, Nov. 2003.
14. TippingPoint, "UnityOne 2000." `http://www.tippingpoint.com-/resource_library/pdfs/2000_Data_Sheet.pdf`, Nov. 2003.
15. Packeteer, "Shaping your network for business." `http://www.packeteer.-com/resources/prod-sol/PacketeerBroFinal2.pdf`, Nov. 2003.
16. IntruVert, "Press release." `http://www.networkassociates.com/us/about/-press/corporate/2003/20030401_173857.htm`, Apr. 2003.
17. R. Sidhu and V. Prasanna, "Fast regular expression matching using FPGAs," in *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, Apr. 2001.
18. R. Franklin, D. Carver, and B. L. Hutchings, "Assisting network intrusion detection with reconfigurable hardware," in *IEEE Symposium on Field-Programmable Custom Computing Machines (FCCM)*, (Napa, CA), Apr. 2002.
19. J. W. Lockwood, N. Naufel, J. S. Turner, and D. E. Taylor, "Reprogrammable Network Packet Processing on the Field Programmable Port Extender (FPX)," in *ACM International Symposium on Field Programmable Gate Arrays (FPGA'2001)*, (Monterey, CA, USA), pp. 87–93, Feb. 2001.
20. F. Braun, J. Lockwood, and M. Waldvogel, "Protocol wrappers for layered network packet processing in reconfigurable hardware," *IEEE Micro*, vol. 22, pp. 66–74, Jan. 2002.
21. D. V. Schuehler and J. Lockwood, "TCP-Splitter: A TCP/IP flow monitor in reconfigurable hardware," in *Hot Interconnects*, (Stanford, CA), pp. 127–131, Aug. 2002.
22. D. V. Schuehler, J. Moscola, and J. W. Lockwood, "Architecture for a hardware based, tcp/ip content scanning system," in *Hot Interconnects*, (Stanford, CA), pp. 89–94, Aug. 2003.
23. J. W. Lockwood, C. Neely, C. Zuver, J. Moscola, S. Dharmapurikar, and D. Lim, "An extensible, system-on-programmable-chip, content-aware Internet firewall," in *Field Programmable Logic and Applications (FPL)*, (Lisbon, Portugal), p. 14B, Sept. 2003.
24. D. E. Taylor, J. S. Turner, J. W. Lockwood, T. S. Sproull, and D. B. Parlour, "Scalable IP lookup for Internet routers," *IEEE Journal on Selected Areas in Communications (JSAC)*, vol. 21, pp. 522–534, May 2003.
25. S. Dharmapurikar, P. Krishnamurthy, and D. E. Taylor, "Longest prefix matching using bloom filters," in *SIGCOMM*, (Karlsruhe, Germany), Aug. 2003.
26. J. Moscola, J. Lockwood, R. P. Loui, and M. Pachos, "Implementation of a content-scanning module for an Internet firewall," in *FCCM*, (Napa, CA), Apr. 2003.
27. J. Moscola, M. Pachos, J. W. Lockwood, and R. P. Loui, "Implementation of a streaming content search-and-replace module for an Internet firewall," in *Hot Interconnects*, (Stanford, CA, USA), pp. 122–129, Aug. 2003.
28. S. Dharmapurikar, P. Krishnamurthy, T. Sproull, and J. W. Lockwood, "Deep packet inspection using parallel bloom filters," in *Hot Interconnects*, (Stanford, CA), pp. 44–51, Aug. 2003.
29. T. Sproull, J. W. Lockwood, and D. E. Taylor, "Control and configuration software for a reconfigurable networking hardware platform," in *IEEE Symposium on Field-Programmable Custom Computing Machines, (FCCM)*, (Napa, CA), Apr. 2002.
30. D. V. Schuehler and J. W. Lockwood, "TCP splitter: A TCP/IP flow monitor in reconfigurable hardware," *IEEE Micro*, vol. 23, pp. 54–59, Jan. 2003.
31. Y. Cho, S. Nahab, and W. H. Mangione-Smith, "Specialized hardware for deep network packet filtering," in *Field Programmable Logic and Applications (FPL)*, (Montpellier, France), Sept. 2002.
32. E. L. Horta, J. W. Lockwood, D. E. Taylor, and D. Parlour, "Dynamic hardware plugins in an FPGA with partial run-time reconfiguration," in *Design Automation Conference (DAC)*, (New Orleans, LA), June 2002.