

# ISP-Operated Protection of Home Networks with FIDRAN

A. Hess, G. Schäfer

Telecommunication Networks Group, Technische Universität Berlin  
Einsteinufer 25, 10587 Berlin, Germany  
Email: [hess, schaefer]@tkn.tu-berlin.de

**Abstract**—In order to fight against the increasing number of network security incidents due to mal-protected home networks permanently connected to the Internet via DSL, TV cable or similar technologies, we propose that Internet Service Providers (ISP) operate and manage intrusion prevention systems (IPS) which are to a large extent executed on the consumer's gateway to the Internet (e.g. DSL router). This paper analyses the requirements of ISP-operated intrusion prevention systems and presents our approach of an IPS that runs on top of an active networking environment and is automatically configured by a vulnerability scanner. The system autonomously analyses the home network and correspondingly configures the IPS. Furthermore, our system detects and adjusts itself to changes in the home network (new service, new host, etc.). First performance comparisons show that our approach – while offering more flexibility and being able to support continuous updating by active networking principles – competes well with the performance of conventional intrusion prevention systems like Snort-Inline.

## KEY WORDS

Intrusion Detection and Response, Vulnerability Scanner, Home Networks, Active Networking

## I. INTRODUCTION

The number of users permanently connected to the Internet (via DSL, TV cable, etc.) continues to rise, as high bandwidth and permanent access currently represents one of the fastest growing markets in the telecommunications business. However, current figures given by the CERT [1] show that with this positive development, rather negative developments in terms of rising number of network attacks due to inadequate network security measures go hand in hand. The number of registered incidents for the year 2000 was 21,756 and for the year 2002 already 82,094. Furthermore, in the first half year of 2003, the CERT actually registered 76,404 incidents. Beyond this, also the diversity of software is increasing and still the quality of many software solutions is insufficient, especially in terms of security vulnerabilities resulting from programming errors. The CERT registered for the year 2000 1,090 vulnerabilities whereas for the year 2002 the CERT already counted 4,129 vulnerabilities.

Intrusion prevention systems, sometimes also called inline network intrusion detection systems, are an upcoming technology to protect communication networks. In contrast to a normal network intrusion detection system (NIDS) an IPS has capabilities that go beyond simply monitoring attacks as an IPS can actually block them. An NIDS sniffs the network and evaluates copies of the packets transmitted, whereas in the case of an IPS, all traffic is routed exclusively through

the IPS and thus, it has the possibility to drop malicious packets. However, these systems require continuous tuning and maintenance work. While companies may dispose of well skilled specialists in order to cope with this maintenance task, ordinary consumers generally do lack the skills required to ensure a secured network configuration that is up to date with recent network intrusion developments. In order to overcome this situation, we propose that Internet Service Providers (ISP) operate and manage intrusion prevention systems (IPS) that to a large extent run on the consumer's gateway to the Internet (e.g. DSL router).

The remainder of this paper is structured as follows. In the next section we discuss the requirements of ISP-operated intrusion prevention systems in home networks and then introduce FIDRAN, our flexible intrusion detection and response framework for active networks. We propose to deploy FIDRAN as an intrusion prevention system on an active networking node between home network and ISP-domain. Further on, we discuss a specific component of our system, the vulnerability scanner that gathers the knowledge about a home network required for network specific configuration of FIDRAN. In section III we describe a prototypical implementation and present first measurements in order to give a performance comparison with conventional intrusion prevention systems like Snort-Inline.

## II. INTRUSION DETECTION AND PREVENTION IN HOME NETWORKS

Figure 1 depicts an example configuration of a home network consisting of a gateway, three PCs and a notebook (for the moment, let us not consider the FIDRAN component running on the DSL router). Different networking technologies like Ethernet and Wireless LAN can be integrated into a home network and the systems of such networks will typically be running various operating systems (OS) and offering different services to the users of the home network (e.g. web server, video server, etc.). Furthermore, new aliasing services offered by companies like the Dynamic DNS Network Services company [2] that offers a service which allows to map static hostnames to dynamic IP addresses, enable public offering of services even from machines located inside a home network. It is clear that such configurations represent attractive targets for potential attackers and that numerous risks arise if such home networks are insufficiently protected.

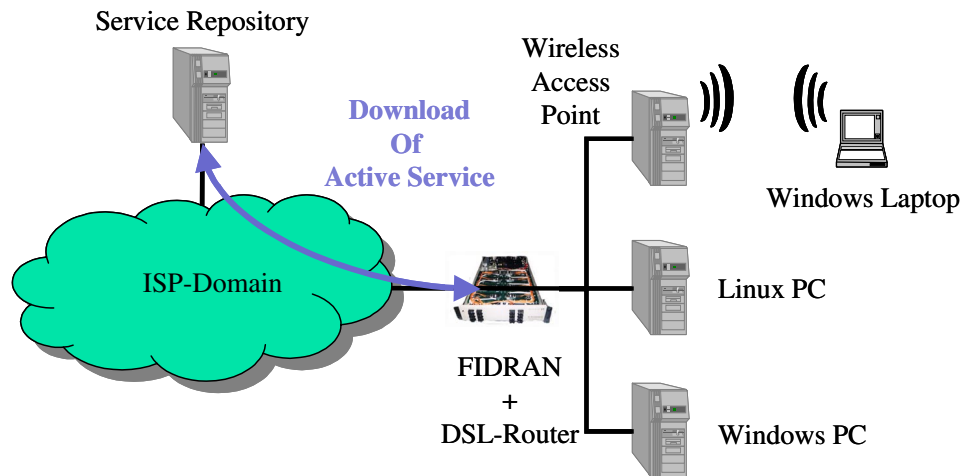


Fig. 1. A Typical Home Network Configuration

#### A. Reasons for and Requirements of ISP-operated IPS

While one could argue that the security of home networks are the sole responsibility of the consumers operating such networks, offering an intrusion prevention service running on part of the infrastructure components of a home network can in fact also result in a couple of benefits for the Internet Service Provider via whom the home network is connected to the Internet:

- As mal-configured computers are an easy target for attackers that aim to use those systems to launch widespread denial of service (DoS) attacks or to conceal the true source of other attacks, an ISP has a certain interest that consumer computers and other devices do not get hacked.
- Performing intrusion prevention functions directly on site at the consumer's premises leads to a favourable load distribution, as many problems can be directly detected and treated at the point where they occur and central bottlenecks are avoided.

Thus, the integration of a self-configuring IPS promises to be highly beneficial for private users, as most consumers are not interested in security internals but in having a well running and protected system. Furthermore, an Internet service provider (ISP) is able to offer a certain surplus value to his customers by operating intrusion prevention systems on behalf of its customers and additionally he is able to reduce problems like denial-of-service attacks in his network.

However, a couple of pitfalls have to be avoided when planning for an ISP-operated intrusion prevention systems for home networks, that lead to a series of requirements for the design of such systems:

- *No special hardware or operating systems requirements:* Ideally, the intrusion prevention system will directly be located on the gateway between the home network and the ISP, e.g. the DSL router, as all traffic coming in and out of the home network is going through this gateway.

However, as these components are to be offered by various independent manufacturers, it has to be avoided that special hardware requirements are dictated by the intrusion prevention system.

- *Ability to evolve over time:* As the field of intrusion prevention systems is rapidly evolving and new attacking techniques will continue to be crafted out by malicious hackers, it is not possible to standardize one fixed intrusion prevention functionality and expect this to be able to cope with attacks coming up in the future. This calls for an extensible approach that allows for flexible programmability of the intrusion prevention system.
- *Performance:* The performance costs of intrusion detection and prevention systems usually increase with growing number of attacking patterns to be defended against. First, it is clear that the IPS should work efficiently and not degrade the performance of the user's Internet connection.

Second, in order to avoid unnecessary performance degradation and at the same time allow for an economic realization of the intrusion prevention system, an ideal IPS should only look for those attacking patterns that actually threaten a specific home network configuration. However, as home network configurations are expected to evolve over time, an ideal solution should include a functionality to detect which vulnerabilities actually exist in a given network, e.g. by scanning which operating systems, services, etc. are operated in the network, in order to keep the configuration of the IPS in line with the protection needs of the actual network configuration.

- *Privacy:* The preceding requirement of scanning the vulnerabilities of a home network might lead to a potential privacy violation, if the vulnerability scanning and processing functionality is not properly designed. Ideally, vulnerability scans should be executed directly inside the home network and as little information as possible from their results should be made known to the ISP.

## B. Related Work

The paper Active Network Based DDoS Defense [6] describes how active networking technology can be used for DDoS protection. The presented approach consists of a sensor which remarks a rapid increase of network traffic and a mobile traffic rate limiter which clones itself. The rate limiter migrates upstream along the attack path in order to stem the attack. This approach only focuses on the detection and prevention of DDoS-attacks which base on the creation of a high network traffic volume.

The Intrusion Blocker based on Active Networks - IBAN [7] consists of a management station, mobile vulnerabilities scanners, and mobile intrusion blockers. A mobile scanner is an application designed to detect one particular vulnerability by looking at system fingerprints. If the scanner has found a vulnerable service an intrusion blocker is placed close to the corresponding system which inspects the traffic for the vulnerable service and blocks the traffic if it detects an attack attempt. IBAN focuses on the detection of automated known attacks. A scanner and a blocker are designed for one particular vulnerability. A mobile application is designed for a particular vulnerability. Consequently, numerous mobile applications could exist in an average network. Further on, each application observes the traffic for a specific traffic pattern, thus each mobile application performs a big set of identical operations. As the authors of the paper write themselves, often it is more difficult to write a detection tool for a specific vulnerability than to provide an adequate defense mechanism. Consequently, IBAN would not deploy a defense mechanism close to a vulnerable host as long as it is not able to detect it. FIDRAN would deploy a new defense mechanism on any FIDRAN which could probably be attacked according to the specification of OS, protocol and service (if given).

Furthermore, a couple of NIDS/IPS systems (Snort-Inline [5], Bro [12], etc.) and vulnerability scanner solutions (Nmap [9], Nessus [4], etc.) exist, but they have generally in common that a human operator is still required as an interface between these components.

Summarizing, we can state that even though a few approaches exist which either exploit the possibilities provided by an active networking environment for intrusion detection, or which realize intrusion detection and prevention following more conventional deployment strategies, none of those systems meets the requirements identified in section II-A.

## C. Realizing ISP-operated IPS for Home Networks with FIDRAN

In this section we describe our FIDRAN-architecture and how to realize ISP-operated intrusion prevention systems for home networks based on this architecture.

Figure 2 depicts the FIDRAN-architecture, which consists of a management module, a vulnerability scanner, a control module, a security policy and a varying set of operational modules (IP-Signatures, TCP-state machine, etc.), which are designed as active services. We skip a detailed discussion of the components, as they are intensively discussed in [10]

and [11], except for the vulnerability scanner which will be explained in the next section.

The fundamental design objectives of our approach are the realization of a demand-driven intrusion prevention system that in principle allows to detect intrusions for a wide variety of operating systems and services while at the same time being able to keep up with the typical traffic volume being exchanged between home networks and the Internet. Therefore, the vulnerability scanner of FIDRAN is placed at the entry point to a home network. First, the traffic volume is generally lower at the network's edge, and second, the scanner must not deal with problems like packet loss caused through packet filters or overloaded routers. Finally, the variance of some parameters (e.g. TTL) which are evaluated by the scanner is small at this point, leading to more reliable interpretation.

The management module is the interface between active node software and FIDRAN, e.g. the management module is able to trigger the active node to download a new operational (op) module. Before a new op-module can be integrated into the FIDRAN-system the management module performs an initial check on it, which consists of the verification of the digital signature and a security policy lookup, whether the installation of the op-module is authorized. Finally, the management module is responsible for the invocation of countermeasures in user-space (e.g. email notification, activation of another service, etc.).

The control module is the central unit of FIDRAN in the kernel and primarily it coordinates the integration of op-modules. Furthermore, it captures the traffic from the network and distributes it to the op-modules. An op-module may contain intrusion detection functionality for specific operating systems, protocols or even specific services (e.g. Linux Apache server). Therefore, an op-module contains a short description including the protocol (IP, UDP, TCP, HTTP, etc.) it operates on, the operating systems which can be protected by it (Windows 9x, Linux, etc.), the service (optional) it is designed for and a priority. The control module evaluates these descriptions and creates a set of corresponding linked lists such that there is a linked list for each tuple (OS/protocol/service) and the priority defines the position of the op-module in the corresponding list.

The security policy specifies the types of modules that can be integrated into the FIDRAN-system. The specification can be given as a set of concrete names or tuples (OS/protocol/service). Inside the security policy it is also defined how to respond to attacks. The spectrum of countermeasures ranges from dropping packets to starting an active service on a remote node.

As already mentioned each operational module performs an individual set of operations on a packet or flow and returns the result to the FIDRAN control module. Op-modules are realized as active networking services, such that each op-module can dynamically be downloaded and integrated into the running FIDRAN system.

In our previous work, it was the FIDRAN-administrator - the user himself or a person charged with this task - who con-

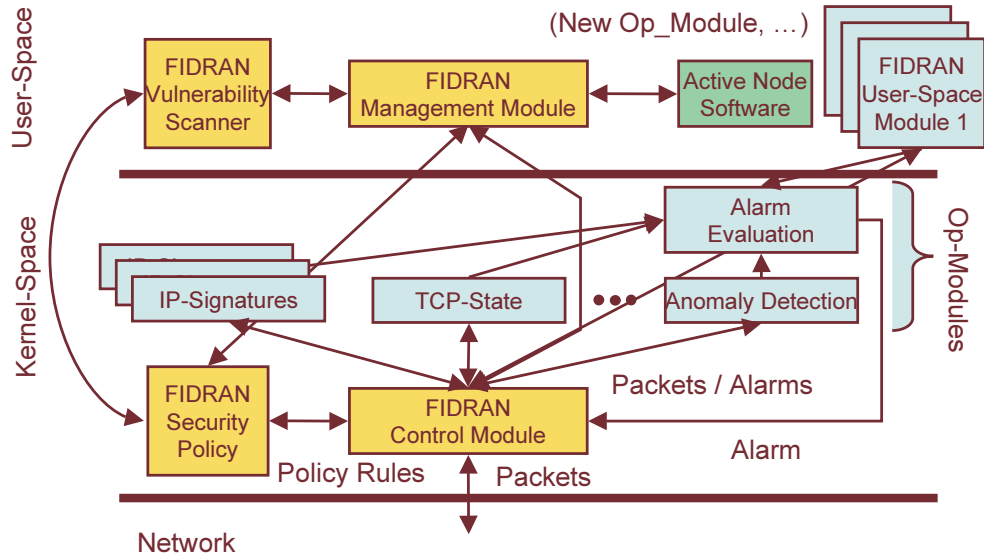


Fig. 2. The FIDRAN Architecture

figured the security policy requiring both an broad knowledge and a continuously effort. In order to ease and automate the protection of home networks we added a vulnerability scanner to the FIDRAN system. The scanner analyses the network to be protected in terms of machines, OSES, services, and accordingly it configures the security policy. Afterwards the management module is triggered to adjust the set of running op-modules to the current specification given in the security policy.

All op-modules are stored on the service repository, that is realized with an LDAP-server with a hierarchical directory structure. The current LDAP directory structure is pretty simple. The root-node is pointing to all op-modules. Then there is a sub-category for each possible tuple (OS/protocol) and one special sub-category containing further FIDRAN active services as e.g. the vulnerability scanner. Further on, each sub-category (OS/protocol) contains a further sub-category for each service (OS/protocol/service). Now, the management module compares the running op-modules, with the op-modules that are available in the recommended categories on the LDAP-server. Afterwards the management module triggers the active node software to download the missing op-modules and to remove the running op-modules which are not further required. For example the vulnerability scanner detects a newly connected host running an Apache server on Linux 2.4 inside a Windows home-LAN. Then the management module triggers the active node software to download any op-module of the LDAP category (Linux-2.4/HTTP/Apache).

#### D. The Vulnerability Scanner

The task of the vulnerability scanner is to analyze the home-LAN in terms of hosts, OSES and services, thereby two operation modes are differentiated: active scanning and passive fingerprinting. Initially launching FIDRAN on the

gateway, the security policy is empty and no op-modules are loaded. Thus, the management module triggers the active node software to download the vulnerability scanner, which is also designed as an active service. Beginning with an empty security policy the vulnerability scanner starts an active scan of the home network in order to specify an initial security policy. After having accomplished the active scan the vulnerability scanner changes its operational mode from active to passive. The passive fingerprinting phase is used to detect changes in the home-LAN and to refine the existing security policy. According to the configuration of FIDRAN, active and passive phases alternate regularly.

During the active scan phase the vulnerability scanner sends specifically crafted packets to well defined addresses and evaluates the replies. In contrast the passive fingerprinting phase uses a network interface card (NIC) in promiscuous mode to sniff the network. Further on, the vulnerability scanner evaluates the sniffed packets and accordingly it adjusts the security policy. In the following we explain the integrated methods for host detection, OS specification and service discovery.

1) *Host Detection:* The gateway host is equipped with two network interface cards, the outer interface to the Internet and the inner interface to the home-LAN. In a first step the vulnerability scanner takes the network address of the inner interface (private IP-address, smaller traffic volume), then it starts to perform a Ping Sweep which is still the most efficient host detection method. Normally, we would have to take into account that IP is connectionless and unreliable but due to the placement of the vulnerability scanner on the gateway, packet loss is negligible.

The passive fingerprinting method is easy. The vulnerability scanner observes the traffic and registers all source and target IP addresses of the home-LAN.

2) *Operating System Detection*: After the host detection phase we know the hosts of the home network, but more information is required about the operating systems running on the hosts. Therefore, the vulnerability scanner performs further tests using the ICMP protocol. While a response to an ICMP echo request is mandatory, other requests like a Timestamp request or an Address Mask request may stay unanswered. Furthermore, the behavior of an OS sometimes vary for unicast and broadcast requests. It is the same with fragmented and not-fragmented requests. For example the fragment ID could either be a constant value or it is increased by a constant value. Linux 2.4 systems use by default a value of 0, while most Microsoft OSes use an increment of 256. Another example is the DF bit. In the case that this bit is set, some OSes echos it back some do not. Finally a third group of OSes sets the bit by default (Linux 2.4, HP-UX, AIX, etc.). Another hint is the Time-to-Live (TTL) field. Normally it is difficult to get the accurate value, as the TTL value is decremented by each router passed, but as the vulnerability scanner is placed on the gateway this method is accurate. Finally, it is also possible to use the Type-of-Service (TOS) field, the Precedence field and the Must-be-Zero (MB) bit for OS-specification.

Generally, the ICMP protocol does not only contain requests and replies but also error messages. The provocation of error-message gives us another tool to collect hints for OS-specification. Normally, an error message quotes the header and at least 8 bytes of the offending packet. But again, some operating systems quote more than 8 bytes and others quote inaccurately (e.g. Linux adds 20 bytes). Some OSes even quote the IP header incorrectly, e.g. the checksum and the TTL field are two candidates for this.

Taken these effects together, with up to five tests we are able to specify groups of OSes. As an example we show in the following how the operating systems Windows 95, Windows 98 and Windows NT 4.0 can be identified. In a first step we would send an ICMP ECHO request to the target host and further on, we would wait for the corresponding ICMP ECHO reply. A TTL value of 32 clearly identifies a Windows 95 system. Other Windows operating systems set by default a TTL value of 128. In a next step we would send an Address Mask request message to the target host. A host which is running Windows 98 or Windows NT 4.0 under Service Pack 4 would reply to the request. To distinguish between these two Windows operating systems we need to send a Timestamp request message. If the host is running Windows NT, it would not respond to the request.

3) *Service Detection*: Finally, after the host detection phase including the OS-specification, we are interested in the offered services (Web-server, file-server, video-server, etc.) by each host. Again we differentiate between active and passive operational mode.

Most services are assigned to a distinct port number, especially the so-called well known ports below 1024. On Linux systems the file `/etc/services` contains information about the mapping of port number and service. Indeed, it is possible that services bind to other ports ( $\neq 1024$ ) than specified in

`/etc/services` and in this case it is difficult to specify the service. But this is mainly done for proxy services or to achieve "security by obscurity".

The active service detection will be done by the half open scan of nmap [9] for TCP-services and provocation of ICMP Error Messages for UDP services. Passive service detection will listen for SYN/ACK TCP packages. UDP services are much more difficult to detect, as there is no connection pattern to monitor for and an UDP packets does not contain any information whether it carries a request or an answer. In addition, we have the difficulty to decide who is the client and who is the server. But statistics can help here. The idea is to set up a table with source- (port/address) and the number of observed packages coming from that respective system. The higher the number of packets, the higher the probability that this system might have UDP services running. For example if we see a system transferring hundreds of packets to port 69 of a specified host, then this is a tftp client doing an upload.

### III. IMPLEMENTATION AND MEASUREMENTS

In this section we shortly describe the prototype which we used for the experiments described later on. The prototype was designed for the active networking environment AMnet [8] and consists of FIDRAN and the described vulnerability scanner. AMnet is an architecture for programmable networks, which provides the possibility to dynamically deploy services on chosen active nodes. These services are stored on a machine called the service repository.

The vulnerability scanner is realized as active networking service for AMnet. It uses the *Packet Capture Library PCAP* for the passive fingerprinting mode. The library allows to put network devices into promiscuous mode and to filter for certain types of traffic. Thus, also ARP traffic can be monitored which is used by the vulnerability scanner during the passive fingerprinting phase in order to detect of new hosts. During the active scanning phase the vulnerability scanner crafts special purpose IP-packets and sends these to the possible target addresses.

At startup the vulnerability scanner actively analyses the home network and stores the results in the security policy. In detail the vulnerability scanner creates an address-list of all hosts of the home network including the operating system and the running services. Afterwards the management module is triggered to download and integrate the corresponding op-modules from the service repository. An op-module could be realized for a specific OS and protocol or even for a specific service as for example an op-module for the protection of the Microsoft IIS. Consequently, several op-modules can be integrated into a FIDRAN system. This is a fact that we have to consider as the performance of FIDRAN depends on the amount of op-modules.

As earlier stated, an important requirement for an IPS is performance. It should work efficiently and not degrade the performance of the user's Internet connection. Therefore, we compared FIDRAN with Snort-Inline, which is an inline version of the well-know Snort. We used three Pentium III 800

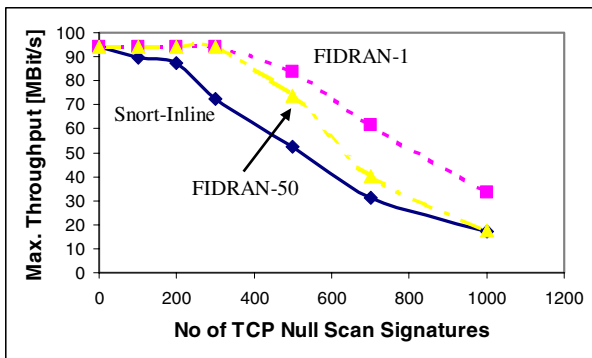


Fig. 3. Comparison: Snort-Inline and FIDRAN

machines running Linux 2.4.20 and connected these via 100 MBit/s. The middle host was running the intrusion prevention system and thus, it was equipped with two network interface cards. We ran the experiment once using Snort-Inline and once FIDRAN as IPS. Both systems were configured such that they were observing the traffic for so-called TCP null scan packets. An attacker uses the TCP null scan technique for OS specification. If a TCP probe packet with no flags set is sent to a closed port, a RST/ACK packet is received, whereas if the port is open no response is received. We chose the TCP null scan signature due to its simple nature. The observation of the network traffic for the TCP null scan signature does not require tricky algorithms.

Regarding FIDRAN we also varied the number of integrated op-modules in order to measure the network throughput in dependency of the number of running op-modules. In a first run we put all signatures in one op-module and in a second run we restricted the number of signatures per op-module to maximally 50. During the second run each packet had to traverse all op-modules, which would not be the case in real life. For example in real life only the packets destined to the Apache Server would traverse the Apache specific op-modules. We measured the maximum network throughput using the tool *Iperf* [3].

In figure 3 three graphs are depicted, one representing Snort-Inline and two FIDRAN. The graph labeled FIDRAN-1 shows the results achieved with FIDRAN in the case that all signatures are stored in one op-module. Restricting the number of signatures per op-module to 50 and thus, increasing the number of op-module results in a decreased maximum network throughput as depicted by the graph labeled FIDRAN-50. The integration of an op-module costs a certain overhead due to internal memory management functions of the kernel. Summarizing the graphs, we can state that while offering more flexibility and being able to support continuous updating by active networking principles – our approach competes well with conventional intrusion prevention systems like Snort-Inline.

## IV. CONCLUSIONS

In this paper we motivated and analysed the requirements for ISP-operated intrusion prevention systems for consumer home networks. Based on this analysis we propose to realize such IPS based on our Flexible Intrusion Detection and Response Framework for Active Networks FIDRAN. We described the interworking of our FIDRAN system and an integrated vulnerability scanner that allows to configure the IPS according to the current configuration of a given home network. The design of FIDRAN allows to dynamically add new functionality and to reconfigure the system at runtime according to the instructions given by the vulnerability scanner.

We mainly see two reasons for the employment of FIDRAN in home networks under control of an Internet Service Provider. First, it allows unskilled consumers or users who do not have enough time to take care about their home networks to benefit from FIDRAN's ability to adequately protect their networks without having to deal with cumbersome configuration tasks. Second, Internet Service Providers get the opportunity to offer a value-added service to their customers and at the same time benefit from protected customers hosts leading to an overall reduction of security problems like DDoS-attacks, etc.

First experiments with a prototype show that the performance of FIDRAN is sufficient for the protection of home networks and can compete with conventional approaches like Snort-Inline that are less flexible and need continuous maintenance and manual updating.

## REFERENCES

- [1] Cert/cc statistics 1988-2003. <http://www.cert.org/stats>.
- [2] Dynamic dns network services, llc. [www.dyndns.org](http://www.dyndns.org).
- [3] Iperf. <http://dast.nlanr.net/Projects/Iperf/>.
- [4] Nessus. <http://www.nessus.org>.
- [5] J. Beale, J. C. Foster, J. Posluns, R. Russell, and B. Caswell. *Snort 2.0 Intrusion Detection*. Syngress, 2003.
- [6] D. S. et al. Active network based ddos defense. In *Proc. of Dance 2002*, 2002.
- [7] W. L. C. et al. Iban: Intrusion blocker based on active networks. In *Proc. of Dance 2002*.
- [8] T. Fuhrmann, T. Harbaum, M. Schöller, and M. Zitterbart. Amnet 2.0: An improved architecture for programmable networks. In *Proceedings of the International Workshop on Active Networks IWAN2002*, December 2002.
- [9] Fyodor. The art of portscanning. *Phrack*, 7, 1997.
- [10] A. Hess, M. Jung, and G. Schäfer. Combining multiple intrusion detection and response technologies in an active networking based architecture. In *Proc. of 17th DFN-Arbeitsstagung über Kommunikationsnetze*, June 2003.
- [11] A. Hess, M. Jung, and G. Schäfer. Fidran: A flexible intrusion detection and response framework for active networks. In *Proc. of 8th IEEE Symposium on Computers and Communications (ISCC'2003)*, July 2003.
- [12] V. Paxson. Bro: a system for detecting network intruders in real-time. *Computer Networks (Amsterdam, Netherlands: 1999)*, 31(23-24):2435-2463, 1999.