

FireFly

Embedded IP Packet-Filtering and Stateful Inspection Firewall



FireFly is a high performance, embedded IP packet-filtering firewall implementation. It enables filtering based on a wide variety of criteria such as source and destination IP address, TCP/UDP ports, protocol type, incoming and outgoing interfaces and many other packet fields. Its core engine permits or denies packets from passing through it based on pre-defined and easily configurable policies that may be specified using rules files, a command line interface or programmatically using its flexible APIs. **FireFly** includes hooks for dynamic firewalling and stateful inspection. Its small footprint, low latency and robustness make it the firewall of choice in embedded networking applications and an ideal perimeter security complement to network security technologies such as IPsec, SSH and SSL.

FireFly is a robust, lean, high performance, packet-filtering firewall implementation for embedded devices. Its core engine permits or denies packets from passing through it based on pre-defined and easily configurable policies, including dynamic addition of new rules to guard against any detected attacks or denial of service conditions such as unicast and broadcast ICMP flooding. **FireFly** can guard ports from external access or just monitor them based on the system's security policies. Sophisticated control of packet filtering may be based on the packet's source / destination IP addresses, source destination TCP or UDP ports, protocol type, network interface and much more! **FireFly's** unique, advanced features also include hooks for dynamic firewalling which when used in conjunction with a

Features

- ❖ Fully configurable IP packet filtering.
- ❖ Extremely small footprint.
- ❖ Low network latency.
- ❖ Dynamic firewall support in conjunction with NAT.
- ❖ Forwarding, logging, and hooks for stateful inspection.
- ❖ Easily controlled by webserver through string based CLI.
- ❖ Includes rule numbering support.
- ❖ Support for CPU types of either endian-ness including PowerPC, MIPS, X86, ARM/XScale.
- ❖ Royalty-free!

NAT implementation can implement convenient, yet highly secure policies for unrestricted access to traffic initiators from inside the firewall-protected network while blocking access to ports from the outside with just a single rule. Control for **FireFly** is in the form of string-based rules that can be implemented as a standard text file or embedded in XML or HTML for provisionable configurability. Fine-grained control of the firewall's datastructures is also possible using the supplied APIs. Application defined stateful inspection can also be added in easily with the provided hooks that let applications track the state of packets going through the engine.

FireFly may be used in standalone mode to provide perimeter or node security, or as an adjunct to network security protocols such as SSH, SSL and IPsec. When used in conjunction with network security protocols it can also be used to force all traffic into/out of the device to be encrypted by blocking off all application ports that may have allowed insecure / unencrypted access.

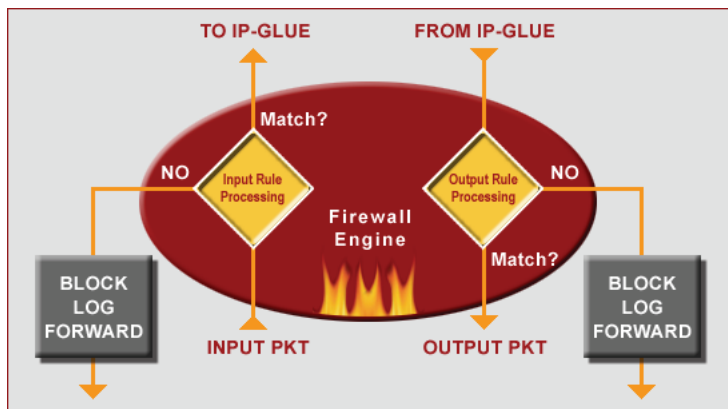


Fig 1: FireFly Operation

Special Features

- ❖ Supports dynamic NAT entry modification based on accesses initiated from inside the firewall to provide unrestricted internal initiated access with maximum external blocking
- ❖ Support of IP and network blacklisting
- ❖ Includes reference rules for Denial of Service (DoS) avoidance
- ❖ Ultra-small memory footprint and minimal dynamic memory usage
- ❖ Very low network latency

FireFly has been extensively validated on a variety of CPU architectures, and this minimizes development and integration efforts. FireFly transparently works with the native OS network stack or with TeamF1's NetF1 high-performance stack as well as other third-party stacks. FireFly's small footprint and robustness have been specifically designed for use in an embedded environment. FireFly's unprecedented flexibility and easy customization make it the firewall of choice in embedded networking applications.

Filtering Support

FireFly supports a variety of filtering options, including:

- ❖ Source and destination IP addresses.
- ❖ Source and destination port numbers.
- ❖ IP/TCP/UDP/ICMP Protocol based filtering.
- ❖ TCP flags such as FIN, SYN, RST, PUSH, ACK & URG.
- ❖ All ICMP types.
- ❖ IP options such as strict source route, loose source route, record route, and time stamp.
- ❖ Fragment flag in the IP header.

Hooks for Stateful Inspection

Stateful inspection provides the ability to track and control the flow of communication passing through the firewall filter. The ability to keep track of state and context information about a session simplifies rules and tries to interpret higher-level protocols. FireFly does not force any specific implementation of such inspection but enables custom versions of circuit-level filtering and application-level filtering to be easily added with the hooks provided.

Management Support

FireFly supports a customizable management interface presented through a string-based command layer, which can be easily controlled through a web-server, with structured data files such as XML or via a command line interface (CLI). Support for rule numbering provides ease of overriding at any level. Customizable hooks for logging and forwarding enable specific actions to be taken when accepting or rejecting packets.

Complements Network Security

Securing a connected embedded device requires security in different dimensions. FireFly's system security typically

Also Available

- ❖ **SSLimSecure**
Secure Socket Layer
- ❖ **ClassHopper**
Alternate Queuing Disciplines
- ❖ **SSHield**
Secure Shell
- ❖ **V-IPSecure IPsec & IKE**
Network Layer Security

Custom Solutions

TeamF1's professional services can provide the resources and expertise to build customized implementations of FireFly and support for specific provisioning configurations for the firewall rules.

Customization Flexibility

- ❖ Available in full-source format
- ❖ Interface, port, and direction specific rules
- ❖ Support for adding custom filtering options
- ❖ Customization hooks and callouts
- ❖ Unwanted components can be scaled out

involves keeping an embedded device protected from external access on specific ports. This perimeter or system security acts as a powerful complement to network security which protects data in transit, when it is used with security solutions such as TeamF1's SSHield Secure Shell (SSH) or V-IPSecure IPsec/IKE. For example, a combination of SSHield's tunneling and FireFly's restricted external access enables sophisticated security policy settings by allowing only a single or few secure points of entrance through the network to the embedded device. Fine grained control over the accessibility of application ports from the public network can be gained while at the same time allowing full access from within the tunneling capabilities of a protocol such as SSH or IPsec.

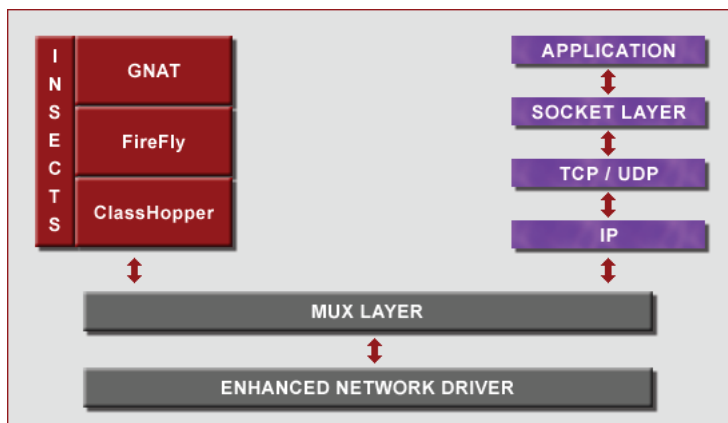


Fig 2: INSECTS "BUMP-ON-THE-STACK" MODEL

Email: sales@TeamF1.com
 Web: www.TeamF1.com
 Ph: 510-505-9931 ext. 5
 Fax: (510) 505-9941



© 2011 TeamF1, Inc.