US006182228B1

(12) **United States Patent**
Boden et al.

(10) **Patent No.: US 6,182,228 B1**
(45) **Date of Patent: Jan. 30, 2001**

(54) **SYSTEM AND METHOD FOR VERY FAST IP PACKET FILTERING**

(75) Inventors: **Edward B. Boden**, Vestal; **Wesley A. Brzozowski**, Endicott; **Paul A. Gebler, Jr.**, Vestal, all of NY (US)

(73) Assignee: **International Business Machines Corporation**, Armonk, NY (US)

(*) Notice: Under 35 U.S.C. 154(b), the term of this patent shall be extended for 0 days.

(21) Appl. No.: **09/135,102**

(22) Filed: **Aug. 17, 1998**

(51) **Int. Cl.**[7] **H04L 9/00**

(52) **U.S. Cl.** **713/201**; 713/154; 713/164; 709/227

(58) **Field of Search** 713/201, 151, 713/153, 154, 160, 164; 709/227, 228, 229, 237; 707/9, 101, 102

(56) **References Cited**

U.S. PATENT DOCUMENTS

5,517,622	5/1996	Ivanoff et al.	395/200.13
5,517,628	5/1996	Morrison et al.	395/375
5,557,798	9/1996	Skeen et al.	395/650
5,606,668 *	2/1997	Shwed	395/200.11
5,634,015	5/1997	Chang et al.	395/309
5,701,316	12/1997	Alferness et al.	371/53
6,092,110 *	7/2000	Maria et al.	709/225

FOREIGN PATENT DOCUMENTS

0854621 *	7/1998	(EP)	H04L/29/06

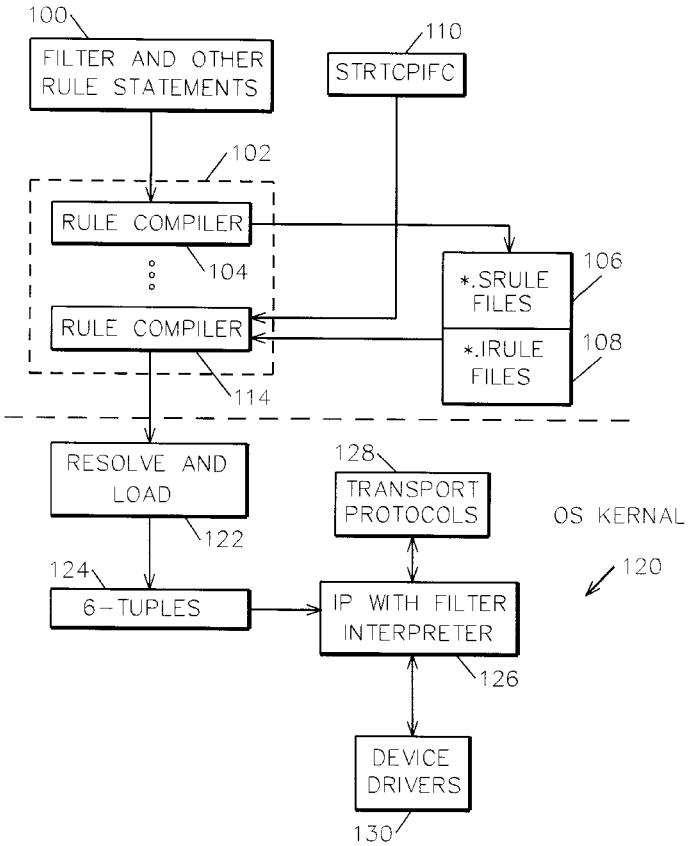
* cited by examiner

Primary Examiner—Robert W. Beausoliel, Jr.
Assistant Examiner—Scott T. Baderman
(74) *Attorney, Agent, or Firm*—Shelley M. Beckstrand

(57) **ABSTRACT**

Small, optimized sequences of binary 6-tuples representing filter rules achieve very fast IP packet filtering. Filtering IP packets received from a caller at the physical interface to an operating system kernel is accomplished by processing FILTER rule statements entered by a user in a rules file to generate 6-tuple filtering rules, each of the 6-tuple filtering rules including an operator index; resolving relative and symbolic indexes in these 6-tuples filtering rules to form resolved filtering rules and loading the resolved filtering rules to the operating system kernal; and interpreting the resolved filtering rules for each IP packet received at the physical interface.

10 Claims, 3 Drawing Sheets



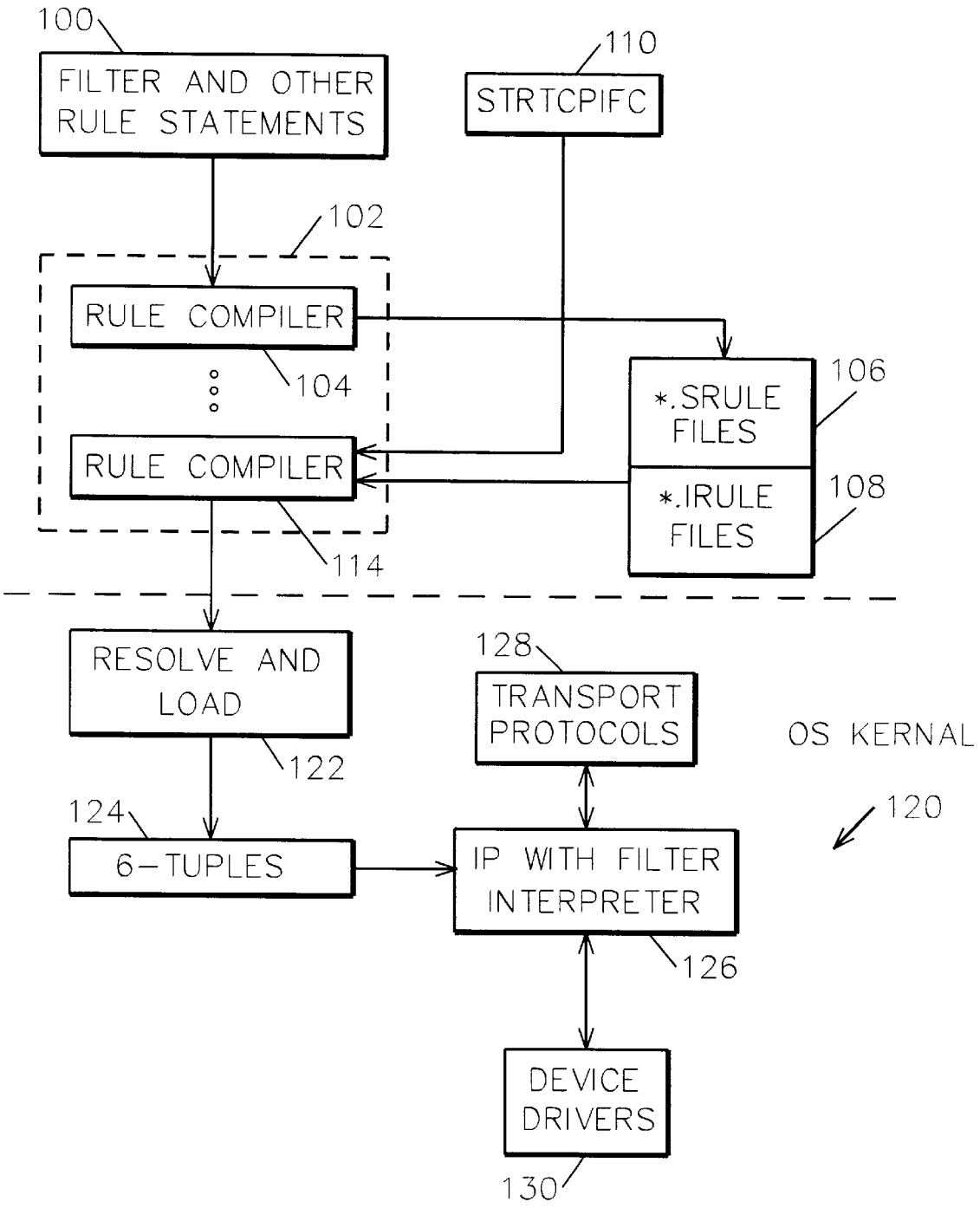


FIG. 1

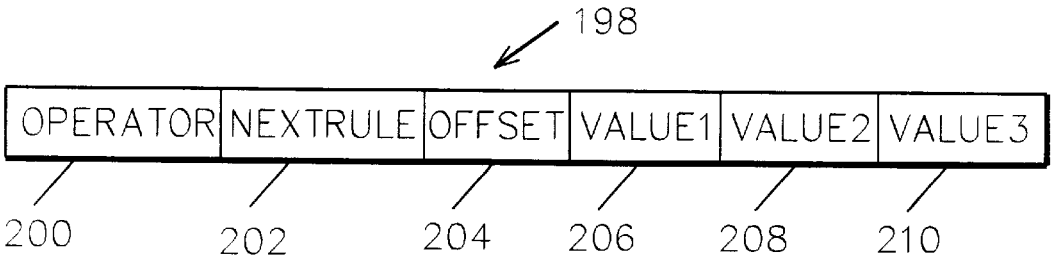


FIG. 3

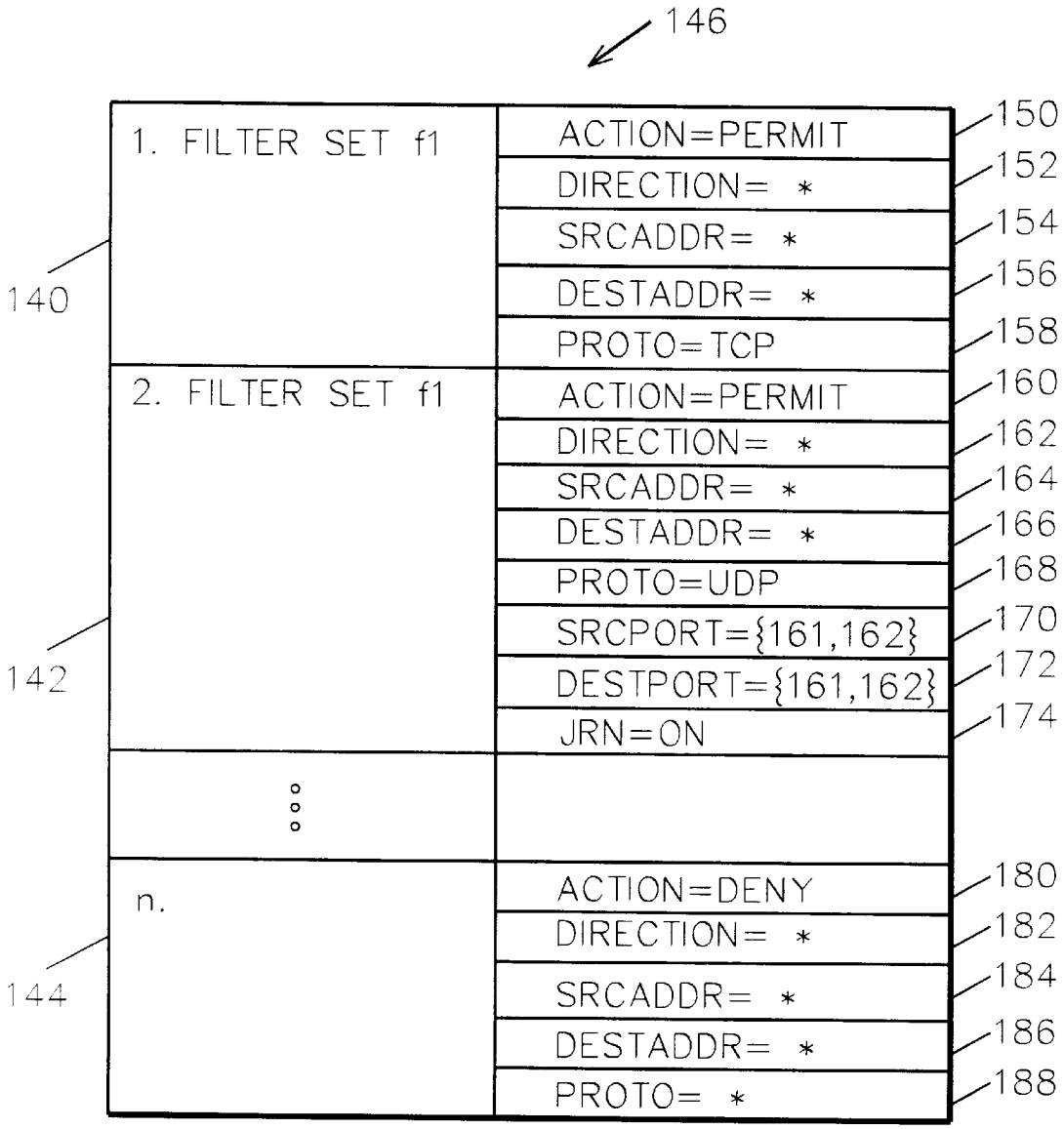


FIG. 2

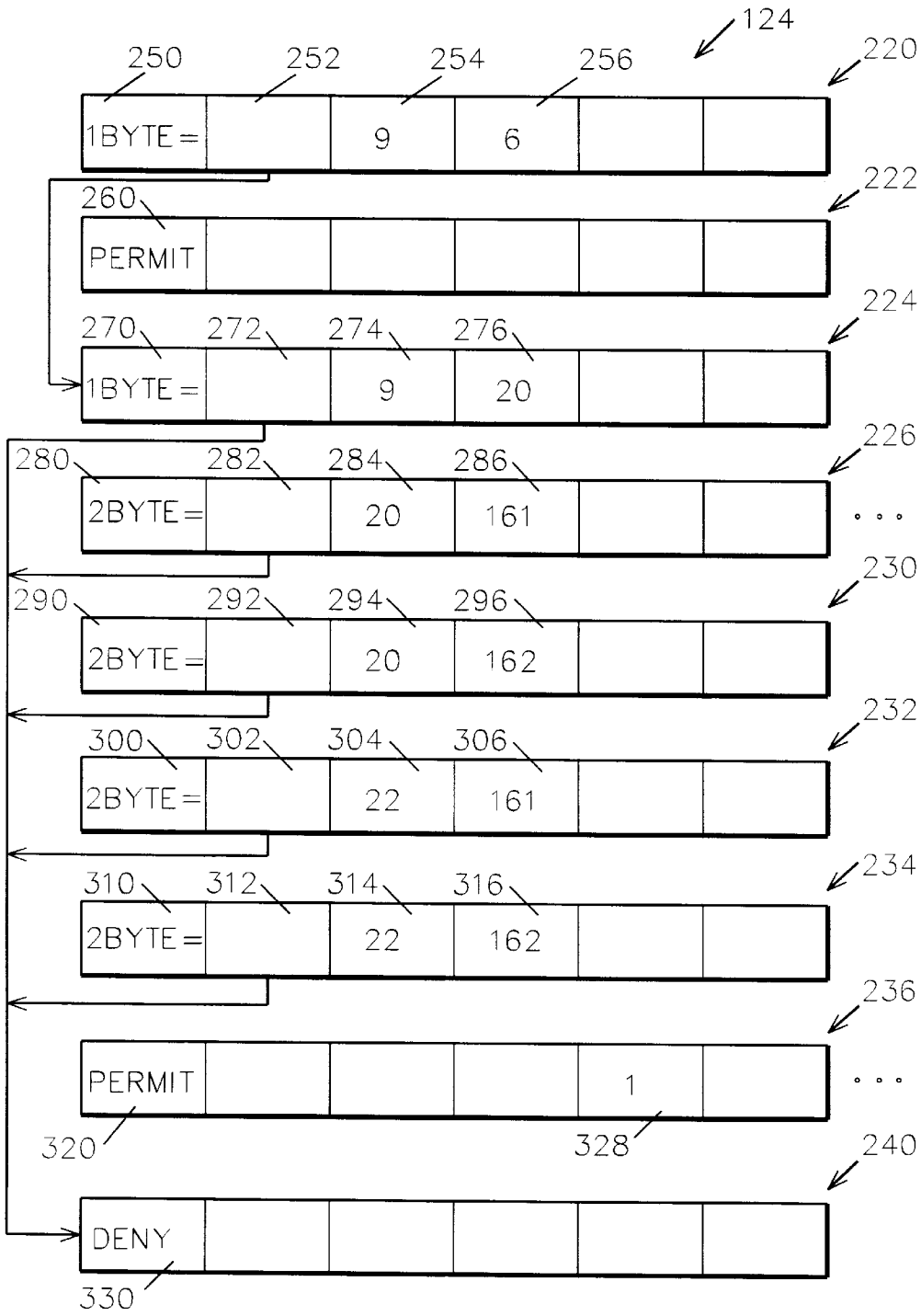


FIG. 4

SYSTEM AND METHOD FOR VERY FAST IP PACKET FILTERING

CROSS REFERENCE TO RELATED APPLICATIONS

U.S. patent application Ser. No. 09/135,148, filed Aug. 17, 1998, entitled "SYSTEM AND METHOD FOR IP NETWORK ADDRESS TRANSLATION AND IP FILTERING WITH DYNAMIC ADDRESS RESOLUTION", assignee docket number EN998067, filed concurrently herewith is assigned to the same assignee hereof and contains subject matter related, in certain respect, to the subject matter of the present application. The above-identified patent application is incorporated herein by reference.

BACKGROUND OF THE INVENTION

1. Technical Field of the Invention

This invention pertains to IP packet filtering. More specifically, it relates to a use of small, optimized sequences of binary 6-tuples representing filter rules to achieve very fast IP packet filtering.

2. Background Art

Internet protocol (IP) network address translation (NAT) and IP filtering are functions which provide firewall-type capability to an Internet gateway system. In one specific system, this is accomplished by providing means for the system administrator to specify specific NAT and filtering rules via an operational navigator graphical user interface (GUI).

IP packet filtering is the process of checking each Internet protocol (IP) packet that is going to be sent from or has just arrived at a gateway system, or node, in a communications network, and based upon that check of making a decision. The decision is (typically, and insofar as it relates to the preferred embodiment of this invention) whether the packet should be discarded or allowed to continue. These are termed the 'deny' and 'permit' actions. IP filtering is widely used in Internet firewall systems, by independent service providers (ISPs) and organizations connected to the Internet.

Filter rules are most commonly an ordered list of rules, processed sequentially from top to bottom (order is specified by the system administrator). Each rule permits a certain kind of IP traffic. Processing for an IP packet continues until the packet is permitted, explicitly denied, or there are no more rules, in which case it is denied. Usually a number of filter rules must be written for each protocol to be permitted.

It is important the IP filtering actions be particularly efficient and very fast because of the huge volume of IP packets a typical gateway system will handle each day, and because of the fairly large number of filter rules that might have to be processed for each IP packet. Typically, each IP packet that flows through the system must be processed by all the filter rules. A moderately busy system can easily be expected to process 10**6 packets per day. Hence, any unnecessary overhead might cause throughput problems.

It is an object of the invention to provide an improved IP packet filtering system and method.

It is a further object of the invention to provide a very fast IP packet filtering system and method.

SUMMARY OF THE INVENTION

In accordance with the invention, a system and method for filtering IP packets received from a caller at the physical interface to an operating system kernel is provided. Filtering is accomplished by processing FILTER rule statements entered by a user in a rules file to generate 6-tuple filtering rules, each of the 6-tuple filtering rules including an operator index; resolving relative and symbolic indexes in these 6-tuples filtering rules to form resolved filtering rules and loading the resolved filtering rules to the operating system kernel; and interpreting the resolved filtering rules for each IP packet received at the physical interface.

Other features and advantages of this invention will become apparent from the following detailed description of the presently preferred embodiment of the invention, taken in conjunction with the accompanying drawings.

BRIEF DESCRIPTION OF THE DRAWINGS

FIG. **1** illustrates the data flow of a preferred embodiment of the invention.

FIG. **2** illustrates an example set of IP filter rules.

FIG. **3** illustrates the format of a 6-tuple in accordance with the preferred embodiment of the invention.

FIG. **4** illustrates the logical structure of 6-tuples, for the example set of FIG. **2**, following the load and resolution step of FIG. **1** in accordance with the preferred embodiment of the invention.

BEST MODE FOR CARRYING OUT THE INVENTION

The problem solved by this invention is: how to generate filtering code that executes in the operating system (OS) kernel from customer-entered rules, that will function correctly and perform very well.

Referring to FIG. **1**, the key elements of the invention and the logical relationships and data flow among them, are illustrated.

This invention is concerned with the translation of FILTER statements **100** to a 6-tuple representation **124**, and the interpretation **126** of the 6-tuples **124** as IP datagrams flow through the OS kernel **120**. FILTER (and other rule) statements **100** are processed by rule compiler **102**. An output of a first invocation **104** of the rule compiler **102** is two sets of files, s-rule files **106** and i-rule files **108**. These files **106**, **108** contain the binary form of the rules in image format (in i-rule files **108**) or retain some symbolic information (in s-rule files **106**). An 'i' or 's' rule file **106**, **108** is generated for each physical interface for which there are rules. Later, when the interface is started in response to start TCP interface (STRTCPIFC) command processing **110**, a second invocation **114** of rule compiler **102** completes resolution of s-rule files **106**. As is represented by step **122**, the resolved rules are loaded to OS kernel **120** in the form of 6-tuples. A key part of loading in the kernel is to resolve the various relative and symbolic addresses in 6-tuples to absolute addresses. Thereupon, 6-tuples **124** are ready to be used by filter interpreter **126** as IP datagrams enter and leave the system via device drivers **130** to input/output processor (IOP), not shown. In a specific embodiment, IOPs provide the actual physical interface to a system, and a network cable of some type plugs into an IOP. Each IOP is represented within the OS kernel by a physical interface control block. Filter interpreter **126** communicates with such IOPs through device driver code **130** residing within kernel **120**. Transport protocols **128** (such as TCP, UDP) are accessed by filter interpreter **126** is processing 6-tuples **124**.

Both image rules (irules) **108** and symbolic rules (srules) **106** are in 6-tuple form, which is the output of rule compiler invocation **104**, which is further described hereafter in

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS
Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS
Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS
Sync your system to PACER to automate legal marketing.