OLIFF
ATTORNEYS AT LAW

CLAIM CONSTRUCTION
(PRECEDENTIAL)

FINJAN LLC v. SONICWALL, INC., Appeal No. 2022-1048 (Fed. Cir. October 13, 2023). Before Cunningham, Reyna, and Bryson. Appealed from N.D. Cal. (Judge Freeman).

Background:

Finjan sued Sonicwall for infringement of two groups of patents, which included a group of "Downloadable" patents, and a group of "ARB" patents.

The "Downloadable" patents describe techniques for protecting electronic devices from malware, and the ARB patents describe rule-based content scanners. The "Downloadable" patents each recite the term "Downloadable" in the claims, and at trial the parties agreed that this term means "an executable application *program*, which is downloaded from a source computer and run on the destination computer." The allegedly infringing product used by Sonicwall inspects payloads of data *packets* (not programs) and sends the packets to a destination without reassembling the packets into an executable program. Packets are building blocks of data transmission that are combined at the destination to produce the file requested by the user. The trial court held that Sonicwall did not infringe the "Downloadable" claims because data packets are distinguishable from executable programs.

The claims in the ARB patents refer to "a computer" and refer several times subsequently to "*the* computer" performing various functions related to scanning network content. The allegedly infringing products used by Sonicwall are essentially separate computers that separately perform the claimed functions. The trial court held that Sonicwall did not infringe the ARB patents because the claims in the ARB patents, by referring to "*the* computer," required at least one computer to perform all of the claimed functions.

Issues/Holdings:

(1) Did the trial court err in holding that the "Downloadable" claims are not infringed by data packet inspection and transmission? No, affirmed.

(2) Did the trial court err in holding that the ARB patent claims are not infringed by two or more separate computers that collectively perform the claimed functions? No, affirmed.

Discussion:

Regarding the "Downloadable" patents, the parties agreed to a claim construction at trial, and such an agreement cannot be challenged on appeal. Accordingly, Finjan was procedurally bound to the claim construction it agreed to. As for the substance of this claim construction, which required the "Downloadable" to be "an executable application *program*," the Federal Circuit held that "a device that merely receives and forwards packets without reassembling their contents does not receive a downloadable, under the parties' agreed-upon construction..." Therefore, Sonicwall did not infringe the patents in suit.

Regarding the ARB patents, the Federal Circuit held that even if "a computer" is construed to mean one or more computers, the subsequent references to "the computer" in the claims require the same single computer to perform the claimed functions. Therefore, Sonicwall's products which use two or more computers do not infringe the patents in suit.

Exemplary Claim of "Downloadable" Patent

10. A system for managing *Downloadables*, comprising:

a receiver for receiving an incoming *Downloadable*;

a *Downloadable* scanner coupled with said receiver, for deriving security profile data for the *Downloadable*, including a list of suspicious computer operations that may be attempted by the *Downloadable*; and

a database manager coupled with said *Downloadable* scanner, for storing the *Downloadable* security profile data in a database.

Exemplary Claim of "ARB" Patent

1. A computer processor-based multi-lingual method for scanning incoming program code, comprising:

receiving, *by a computer*, an incoming stream of program code;

determining, by the computer, any specific one of a plurality of programming languages in which the incoming stream is written;

instantiating, *by the computer*, a scanner for the specific programming language, in response to said determining, the scanner comprising parser rules and analyzer rules wherein the parser rules define certain patterns in terms of tokens, tokens being lexical constructs for the specific programming language, and wherein the analyzer rules identify certain combinations of tokens and patterns as being indicators of potential exploits, exploits being portions of program code that are malicious;

identifying, *by the computer*, individual tokens within the incoming stream;

dynamically building, *by the computer* while said receiving receives the incoming stream, a parse tree whose nodes represent tokens and patterns in accordance with the parser rules;

dynamically detecting, *by the computer* while said dynamically building builds the parse tree, combinations of nodes in the parse tree which are indicators of potential exploits, based on the analyzer rules; and

indicating, *by the computer*, the presence of potential exploits within the incoming stream, based on said dynamically detecting.