EXHIBIT 9

IN THE UNITED STATES DISTRICT COURT

FOR THE NORTHERN DISTRICT OF CALIFORNIA

OAKLAND DIVISION

FINJAN, INC.,

Plaintiff,

v.

QUALYS INC.,

Defendant.

CASE NO.: 4:18-cv-07229-YGR

OPENING EXPERT REPORT OF DR. AVI RUBIN

Dated: December 1, 2020

Dr. Aviel D. Rubin

c) repeating said instantiating, said identifying, said dynamically building, said dynamically detecting and said indicating for the embedded program code, based on the parser rules and the analyzer rules for the other programming language.

259. Finjan's infringement contentions for this step merely refer to its analysis of Claim 1 and states that "the Accused Products are capable of scanning for multiple programming languages within a stream of program code." I will therefore adopt Finjan's apparent construction that this term requires showing only that Li's scanner is capable of scanning for multiple programming languages within a stream of program code.

260. Li discloses the ability to detect and scan embedded program code (such as JavaScript or VBScript) within HTML:

> The Threadizor 208 simulates execution of the extracted key actions to generate linearized key action sequences termed executing threads 210 for each entry point into the code. Scripting programs have at least one main entry point. Some scripting programs may have more than one entry point. For example, JavaScript and VBScript programs are typically embedded in HTML, and, thus, some HTML-related event handlers are written in the script and may be considered entry points as well. Starting from each entry point, the Threadizor 208 simulates the execution of the program and records all key actions into an execution line or executing thread.

Li 6:50-61; *see also* 5:30-41:

> Although the two code pieces are written in different languages, they are very similar in that they create the same object and invoke the same methods. The differences are in the variable names and declarations. As Scripting viruses are propagated via Source code, the same virus could be represented in different forms, using different variables, control flows, and functions, or by adding/removing space characters. Thus, the present invention, differently from the prior art, identifies a virus by key actions that are scattered in the Source code, rather than the source code itself, and thus, provides an important advantage.

ix. Claim 22

a) A non-transitory computer-readable storage medium storing program code for causing a computer to perform the steps of:

261. I understand that a claim's preamble is generally not limiting. Nevertheless, I note that Finjan's infringement contentions for this preamble simply refer to the preamble of Claim 1 without noting or discussing any differences. I therefore will adopt Finjan's apparent construction

that this preamble is materially the same as Claim 1's preamble, which I discuss in Paragraphs 196-199 and incorporate here by reference.

b) receiving an incoming stream of program code;

262. I discussed this step, which is found in Claim 1, in Paragraphs 200-207. I incorporate those Paragraphs here by reference.

263. I note that Finjan's infringement contentions for this step simply refer to step of Claim 1 which I discussed in paragraphs identified above. Finjan makes no reference to any differences in the specific language between these steps as they appear here as compared to Claim 1. I therefore will adopt Finjan's apparent construction that this step is materially the same as the corresponding step in Claim 1.

c) determining any specific one of a plurality of programming languages in which the incoming stream is written;

264. I discussed this step, which is found in Claim 1, in Paragraphs 208-211. I incorporate those Paragraphs here by reference.

265. I note that Finjan's infringement contentions for this step simply refer to step of Claim 1 which I discussed in paragraphs identified above. Finjan makes no reference to any differences in the specific language between these steps as they appear here as compared to Claim 1. I therefore will adopt Finjan's apparent construction that this step is materially the same as the corresponding step in Claim 1.

d) instantiating a scanner for the specific programming language, in response to said determining, the scanner comprising parser rules and analyzer rules for the specific programming language, 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 corresponding exploits, exploits being portions of program code that are malicious;

266. I discussed this step, which is found in Claim 1, in Paragraphs 212-222. I incorporate those Paragraphs here by reference.