NOTE: This disposition is nonprecedential.

United States Court of Appeals for the Federal Circuit

PALO ALTO NETWORKS, INC.,
Appellant

v.

FINJAN, INC.,
Appellee

2017-2059

Appeal from the United States Patent and Trademark Office, Patent Trial and Appeal Board in Nos. IPR2015-02001, IPR2016-00157, IPR2016-00955, IPR2016-00956.

Decided: September 19, 2018

ORION ARMON, Cooley LLP, Broomfield, CO, argued for appellant.

PAUL J. ANDRE, Kramer Levin Naftalis & Frankel LLP, Menlo Park, CA, argued for appellee. Also represented by JAMES R. HANNAH.

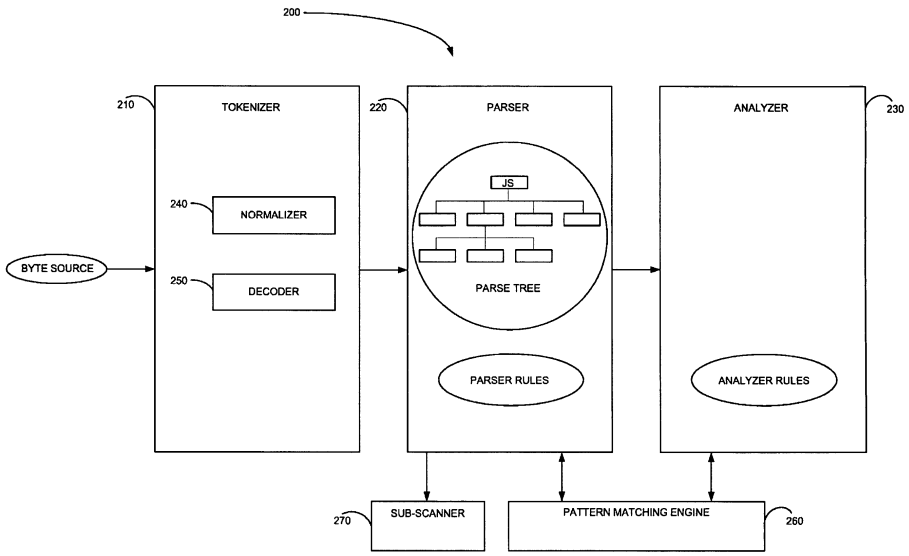
Before REYNA, SCHALL, and STOLL, *Circuit Judges*.

STOLL, *Circuit Judge*.

Appellant Palo Alto Networks, Inc. petitioned for two inter partes reviews of Appellee Finjan, Inc.'s U.S. Patent No. 8,225,408, alleging that certain claims were unpatentable as obvious. The Patent Trial and Appeal Board of the U.S. Patent and Trademark Office ("Board") found that there was insufficient evidence that Palo Alto Networks's proposed prior art combinations would have taught the "dynamically building" claim limitation. *Palo Alto Networks, Inc.*, No. IPR2015-02001, 2017 WL 1052502, at *4–10 (P.T.A.B. Mar. 17, 2017) ("*Board Decision*"). Therefore, the Board found that Palo Alto Networks failed to carry its burden of demonstrating, by a preponderance of the evidence, that any of the challenged claims would have been obvious. *Id.* Palo Alto Networks appeals. We affirm.

I

Finjan's '408 patent relates to methods and systems for detecting malware in data streamed from a network onto a computer. The patent relates to network security, including scanning code to determine whether there are potential viruses in the code. The patent describes a scanner system that preferably uses generic architecture, is language-independent, and is customized for a specific language by using a set of language-specific rules. The '408 patent explains that this adaptive rule-based scanner has three components (illustrated in Figure 2, below). Tokenizer 210 recognizes and identifies constructs (i.e., "tokens") within a byte source code. For example, code between {} or [] would become a token. Parser 220 controls the process of scanning incoming content, preferably by building a parse tree data structure that represents the incoming content. Finally, analyzer 230 checks for malware by searching for specific patterns of content that indicate malware.



'408 patent, Fig. 2.

Claims 1, 3–7, 9, 12–16, 18–23, 29, and 35 are at issue in this appeal, and independent claim 1 is illustrative:

> 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 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 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.

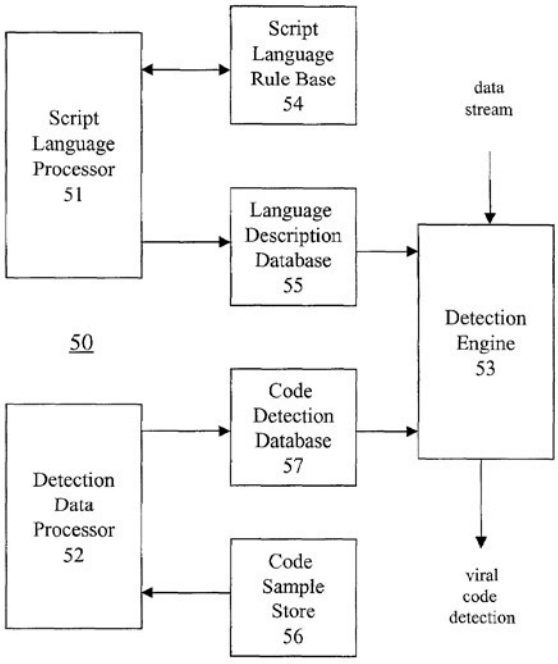
Id. claim 1 (emphasis added to highlight the disputed claim limitation). We focus on the claim limitation requiring "dynamically building" a parse tree, which is common to all the challenged claims. The Board construed "dynamically building" to mean: "requires that a time period for dynamically building overlap with a time period during which the incoming stream is being received." *Board Decision*, 2017 WL 1052502, at *3. This unopposed claim construction was proposed by Palo Alto Networks based on the plain claim language, which requires "dynamically building, by the computer while said receiving receives the incoming stream." '408 patent, claim 1.

II

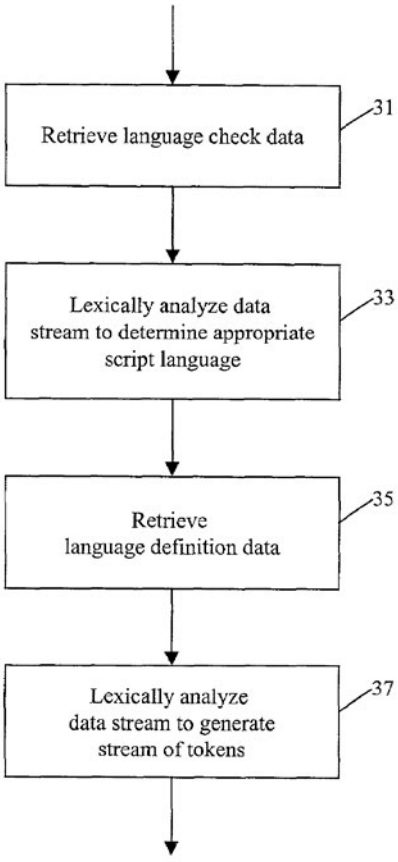
Palo Alto Networks asserted that claims 1, 3–5, 9, 12–16, 18, 19, 22, 23, 29, and 35 of the '408 patent would have been obvious over U.S. Patent No. 7,636,945 ("Chandnani") and U.S. Patent No. 5,860,011 ("Kolawa")

under 35 U.S.C. § 103. Palo Alto Networks also asserted that the same claims would have been obvious over Chandnani, Kolawa, and U.S. Patent No. 7,284,274 ("Walls").

Chandnani teaches a method of detecting malware in a data stream, including determining the programming language of the data stream and detecting viral code. Figure 2 from Chandnani (duplicated below) illustrates Chandnani's script language virus detection apparatus, including detection engine 53, one of the focal points of Chandnani's method:



Chandnani, Fig. 2, col. 8 ll. 5–7. Detection engine 53 tokenizes the incoming data stream by breaking it into smaller pieces known as tokens. As part of that process, it receives the language check data from the language description module 55, as indicated in step 31 of Figure 6:



Id. at Fig. 6, col. 7 ll. 61–63. The language check data is used to lexically analyze the data stream at step 33 to determine the appropriate script language. *Id.* at col. 7 ll. 63–65. At step 35, the language definition data for the script language determined in step 33 is retrieved from language description module 55. *Id.* at col. 7 ll. 65–67. Using the language definition data retrieved at step 35, the data stream is lexically analyzed for a second time to generate the stream of tokens at step 37. *Id.* at col. 7 l. 67–col. 8 l. 3 ("the data stream is *again* lexically analyzed to generate a stream of tokens" (emphasis added)).

Chandnani provides the following summary of the tokenizing procedure, explaining the function of its "lexical analyzer":

> To tokenize the data stream, a script language used in the data stream is determined using the language check data. *The data stream is analyzed* using the language check data to select the language definition data to use for the detection process. Next, the selected language definition data and the data stream are supplied to the lexical analyzer. *The data stream is lexically analyzed again*, this time using the language definition data, to generate a stream of tokens. As mentioned above, each generated token corresponds to a specific language construct, and may be a corresponding unique number or character.

Id. at col. 8 ll. 7–17 (emphases added).

The Board determined that the dispositive issue was whether Palo Alto Networks demonstrated, by a preponderance of the evidence, that the prior art teaches or suggests "dynamically building" a parse tree "while" receiving an incoming stream of program code. Specifically, the parties disputed whether Chandnani discloses that the time period for generating the token stream overlaps with the time period for receiving the incoming stream, as required by the Board's construction of "dynamically building."

The Board found that Chandnani does not teach the "dynamically building" limitation of the '408 patent because it does not demand or even imply that the data stream is being received while being tokenized. In support of its conclusion, the Board cited testimony of Finjan's expert, Dr. Nenad Medvidovic. Ultimately, the Board concluded that Palo Alto Networks had not demonstrated, by a preponderance of the evidence, that the combination of Chandnani and Kolawa would have taught

or suggested "dynamically building" a parse tree "while" receiving an incoming stream of program code. The Board likewise concluded that Palo Alto Networks had not demonstrated, by a preponderance of the evidence, that the combination of Chandnani, Kolawa, and Walls teaches or suggests "dynamically building" a parse tree "while" receiving an incoming stream of program code. The Board found that Finjan had failed to carry its burden of showing that the instituted prior art disclosed the "dynamically building" limitation. Accordingly, it did not consider evidence of secondary considerations of nonobviousness, including expert testimony from Finjan's expert, Dr. Harry Bims, on that issue.

Palo Alto Networks appeals. We have jurisdiction pursuant to 28 U.S.C. § 1295(a)(4)(A).

III

Obviousness under 35 U.S.C. § 103 is a mixed question of law and fact.[1] We review the Board's ultimate obviousness determination de novo and its underlying fact-findings for substantial evidence. *Harmonic Inc. v. Avid Tech., Inc.*, 815 F.3d 1356, 1363 (Fed. Cir. 2016). "Substantial evidence is something less than the weight of the evidence but more than a mere scintilla of evidence," meaning that "[i]t is 'such relevant evidence as a reasonable mind might accept as adequate to support a conclu-

---

[1] Congress amended § 103 when it enacted the Leahy–Smith America Invents Act ("AIA"). Pub. L. No. 112-29, § 3(c), 125 Stat. 284, 287 (2011). Because the application that led to the '408 patent has never contained (1) a claim having an effective filing date on or after March 16, 2013, or (2) a reference under 35 U.S.C. §§ 120, 121, or 365(c) to any patent or application that ever contained such a claim, the pre-AIA § 103 applies. *See id.* § 3(n)(1), 125 Stat. at 293.

sion.'" *In re NuVasive, Inc.*, 842 F.3d 1376, 1379–80 (Fed. Cir. 2016) (quoting *In re Applied Materials, Inc.*, 692 F.3d 1289, 1294 (Fed. Cir. 2012)).

As a preliminary matter, the parties do not dispute the Board's definition of a person of ordinary skill in the art as having a bachelor's degree or equivalent experience in computer science or related academic fields, and three to four years of additional experience in the field of computer security, or equivalent work experience.

A

On appeal, Palo Alto Networks challenges the Board's reading of Chandnani. In particular, Palo Alto Networks asserts that the Board erred in finding that Chandnani does not teach "dynamically building" a parse tree while receiving an incoming stream of program code.

We hold that the Board's finding is supported by substantial evidence because the reference itself, by using the word "again," indicates that the data stream is lexically analyzed more than once and not simultaneously. Chandnani teaches using "language check data to lexically analyze the data stream to determine the appropriate script language" and that "[u]sing the language definition data . . . the data stream is *again* lexically analyzed to generate a stream of tokens." Chandnani, col. 7 l. 60–col. 8 l. 3 (emphasis added). The Board was also entitled to credit the testimony of Dr. Medvidovic, who opined that "simply because Chandnani's tokenizer operates on a data stream does not demand or even imply that the data stream is being received while being tokenized." *Board Decision*, 2017 WL 1052502, at *8 (quoting J.A. 3091–92, ¶ 74 (citing Chandnani, col. 9 ll. 12–16, col. 7 l. 60–col. 8 l. 17, Fig. 6)). Dr. Medvidovic further testified that "Chandnani would still temporarily store the entire data stream in memory at least between the first and second lexical analyses." J.A. 3092 ¶ 74. His testimony supports the Board's conclusion that a person of ordinary skill,

reviewing Chandnani, would not have understood it to teach "dynamically building" a parse tree.

We appreciate Palo Alto Networks's argument and its expert's testimony that speed is critical to malware detection and that "delaying analysis of a file obtained from a network until the entire file is received would have been viewed as creating unnecessary delay for the user and subjecting the receiving computer to risk of damage due to execution of stored malware files." Appellant's Br. at 40; *see also id.* at 8–11. The issue before us, however, is whether the Board's reading of Chandnani is supported by "such relevant evidence as a reasonable mind might accept as adequate to support" the Board's conclusion. *NuVasive*, 842 F.3d at 1379–80. Based on plain language quoted above from Chandnani's specification, we conclude that it is supported by such substantial evidence. *See id.*

Palo Alto Networks next challenges the Board's treatment of Walls, a prior art reference Palo Alto relied on in the alternative for disclosure of the "dynamically building" limitation. Palo Alto argues that the Board erred by analogizing Walls to Chandnani and by not meaningfully reviewing Walls as a separate reference that discloses the dynamically building limitation. In its Final Written Decision, the Board noted that Palo Alto Networks's "challenges based on the combination of Walls with Chandnani and Kolawa suffer from the same deficiencies as its challenges based on Chandnani and Kolawa alone, in that [it] does not sufficiently establish that the prior art it relies on discloses the temporal interleaving required by our construction of 'dynamically building.'" *Board Decision*, 2017 WL 1052502, at *8. In so holding, the Board considered the disclosure of Walls and the expert testimony regarding Walls from both parties. It also performed its own review of Walls and ultimately concluded that Palo Alto Networks had not demonstrated, by a preponderance of the evidence, that the combination of Chandnani, Kolawa, and Walls teaches or suggests

"dynamically building" a parse tree "while" receiving an incoming stream of program code. Accordingly, we conclude that, contrary to Palo Alto Network's assertion, the Board did not fail to meaningfully consider the teachings of Walls.

B

Finally, Palo Alto Networks asserts that the Board erred by not considering particular cross-examination testimony from Finjan's expert witnesses when analyzing whether the prior art taught "dynamically building." Before trial, Finjan had moved to exclude this particular cross-examination testimony of its experts, Dr. Medvidovic and Dr. Bims. The Board denied Finjan's motion. Instead, the Board explained that it would consider Finjan's arguments "as going to the weight that should be given to the cross-examination testimony," not its admissibility. *Board Decision*, 2017 WL 1052502, at *10. In its Final Written Decision, the Board stated that it had "considered and weighed the testimony provided by Dr. Medvidovic," but that it had "not relied on the testimony of Dr. Bims in reaching [its] decision." *Id.* Palo Alto Networks argues that the Board failed to meaningfully consider the testimony from both witnesses.

First, Palo Alto Networks protests that Dr. Medvidovic's cross-examination testimony was not substantively discussed in the Board's decision. Palo Alto Networks cites *Google Inc. v. Intellectual Ventures II LLC*, 701 F. App'x 946 (Fed. Cir. 2017), where this court held in a nonprecedential opinion that it could not review the Board's findings because it could not discern, from the Board's opinion, the scope of "all evidence and arguments" considered by the Board. *Google*, 701 F. App'x at 954. Here, however, it was clear why Dr. Medvidovic's testimony would not have been convincing. More elaboration was not required. The Board acknowledged Dr. Medvidovic's cross-examination testimony and ex-

plained its reasoning for why Chandnani does not disclose the required claim limitation, even having considered Palo Alto Networks's counterarguments:

> [T]o the extent that Chandnani discloses different embodiments directed to the underlying identification of the file to be scanned—whether stored on a hard or floppy disk, or received via a network—Chandnani still requires multiple passes through the file, first to determine the appropriate script language and then to lexically analyze the data stream to generate the stream of tokens.

Board Decision, 2017 WL 1052502, at *8 (Mar. 17, 2017) (citing Chandnani Fig. 6; J.A. 3090–92). Indeed, as we held in *PGS Geophysical AS v. Iancu*, even while "we may not supply a reasoned basis for the agency's action that the agency itself has not given, we will uphold a decision of less than ideal clarity if the agency's path may reasonably be discerned." 891 F.3d 1354, 1365 (Fed. Cir. 2018) (citing *Bowman Transp., Inc. v. Arkansas–Best Freight Sys., Inc.*, 419 U.S. 281, 286 (1974); *NuVasive*, 842 F.3d at 1383). As in *PGS*, we think that the Board did not fail to address the question at hand. We therefore affirm.

Second, Palo Alto Networks argues that the Board gave no rationale for not relying on Dr. Bims's cross-examination testimony. The Board, having found that Finjan had failed to carry its burden of showing that the instituted prior art disclosed the "dynamically building" limitation, did not reach the issue of secondary considerations of nonobviousness. Therefore, it was not necessary for the Board to consider Dr. Bims's testimony, which was limited to the issue of secondary considerations of nonobviousness. Accordingly, we find no error in the Board's decision not to consider Dr. Bims's testimony, and we conclude that the Board sufficiently explained its rationale for declining to do so.

IV

We have considered the parties' remaining arguments, including Palo Alto Networks's arguments regarding "temporal interleaving," and find them unpersuasive. We affirm the Board's decision.

AFFIRMED

COSTS

Costs to Appellee.