

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PALO ALTO NETWORKS, INC.,
Petitioner,

v.

FINJAN, INC.,
Patent Owner.

Case IPR2015-01974
Patent 7,647,633 B2

Before, THOMAS L. GIANNETTI, MIRIAM L. QUINN, and
PATRICK M. BOUCHER, *Administrative Patent Judges*.

QUINN, *Administrative Patent Judge*.

DECISION
Denying Petitioner's Request for Rehearing
37 C.F.R. § 42.71(d)

Palo Alto Networks, Inc. (“Petitioner”) requests rehearing of our decision regarding institution of *inter partes* review of claims 14 and 19 of U.S. Patent No. 7,647,633 B2 (“the ’633 patent”). Paper 9 (“Req. Reh’g”). In our Decision on Partial Institution (Paper 7), we exercised our discretion, under 35 U.S.C. § 325(d), not to institute *inter partes* review of claims 1–4, 6–8, 13, 28, and 34. Dec. 12–13. Petitioner requests rehearing of our determination that the asserted grounds based, at least in part, on Shin, present the same or substantially the same prior art or arguments that were presented previously to the Office. *Id.* at 9–11. In particular, Petitioner asserts that our Decision was based on a clearly erroneous conclusion that the “teachings of [Shin] are substantially the same as the teachings of [Ji].” Req. Reh’g 1. We deny the request for rehearing for the reasons that follow.

The party challenging the decision of the Board has the burden of showing that the decision should be modified. 37 C.F.R. § 42.71(d). “The request must specifically identify all matters the party believes the Board misapprehended or overlooked, and the place where each matter was previously addressed in a motion, an opposition, or a reply.” *Id.*

In our Decision, we stated that upon review of Shin and comparison of its disclosure to that which the Office considered in the reexamination proceeding, we were persuaded that the Shin technology relied upon in the Petition is substantially the same as that which was considered relevant in Ji during the reexamination proceeding and during appeal at the Board. Dec. 10. Petitioner asserts that Shin discloses looking for a “magic byte sequence,” in addition to looking for <applet> tags. Req. Reh’g 5–6 (citing Pet. 26, 29). Petitioner argues that, regarding the “determining whether”

limitation, Dr. Rubin, Petitioner’s declarant, “explained that Shin discloses precisely what was found to be *missing* in Ji.” Req. Reh’g 5 (citing Pet. 29–30, 48; Ex. 1002 ¶¶ 109–14, 125, 127). We do not agree with Petitioner that we have overlooked or misapprehended Shin’s disclosure with respect to this issue.

First, neither the Petition nor Dr. Rubin’s Declaration, at the pages cited, states what Petitioner now argues—that Shin is different than Ji in any material respect, including how each reference detects applets. The argument that Shin is distinguishable from Ji was not overlooked because it was not presented. Moreover, even if there were differences between Ji and Shin’s implementations of applet detection, the statute does not require a complete overlap of these implementations for us to deem their disclosures (or arguments presented regarding those disclosures) substantially the same. *See* 35 U.S.C. § 325(d) (“the same or substantially the same prior art or arguments”).

Second, we did not overlook or misapprehend that Shin discloses searching a “magic byte sequence” to detect applets. As we stated in the Decision, “‘Applet’ detection techniques have been given full consideration during reexamination.” Dec. 10. During the reexamination proceeding (Control No. 90/013,016), the examiner addressed whether applet detection meets the “determining” limitation, and supported its assertion with two techniques disclosed in Ji: (1) discriminating between Java applets from non-applets (*see* Ex. 1003, at 25–26 (“Ji specifically teaches determining whether downloadable-information includes executable code by discriminating Java applets from non-applets”)); and (2) scanning

downloaded information for particular executable instructions (*see id.*).
Petitioner’s position regarding Shin is that it detects applets. Pet. 29–30.
This position is no different than the examiner’s position in reexamination that Ji detects Java applets. Indeed, the issue is whether applet detection, alone, meets the “determining whether” limitation—an issue that, we are persuaded, has been fully considered in reexamination and on appeal to the Board. Therefore, notwithstanding that Shin states another way in which an applet may be detected, we are persuaded that applet detection, as a whole, has been given full consideration.

Third, we are not persuaded that we overlooked or misapprehended the disclosure in the ’633 patent. Petitioner argues that the ’633 patent identifies Shin’s “magic byte sequence” method as an “example” of the “determining” limitation recited in the claims. Req. Reh’g 7. Petitioner states that Dr. Rubin and the Petition rely on the ’633 patent disclosure to assert that detecting a “magic byte sequence” was a known method of detecting Java applets. *Id.* (citing Pet. 29–30 and Ex. 1002 ¶ 114). But Shin, again, detects only Java applets, as stated above. Furthermore, Shin detects only uncompressed Java class files, (Ex. 1009, 18),¹ which is not co-extensive with the so-called “example” provided in the ’633 patent. The ’633 patent describes “analyz[ing] a received potential-Downloadable for a file header . . . [for] portable executable or standard ‘.exe’ file format for Windows OS application programs, a Java class header for Java applets, and

¹ Shin states that “this [magic byte sequence] technique can not detect class files which . . . are a part of compressed archive (Jar or Zip).”

so on for other applications, distributed components, etc. ‘Zipped’, meta or other compressed files.” Ex. 1001, col. 14, ll. 60–67. That is, even if we were persuaded to rely on the ’633 patent specification for examples of “determining whether the downloadable-information includes executable code,” an exercise in hindsight, the ’633 patent embodiment detects many types of file formats. This approach is consistent with an objective of the invention to “enable[] more reliable protection” against “Trojan horses and program code groupings, as well as software ‘components’, such as Java™ applets, ActiveX™ controls, JavaScript™/Visual Basic scripts, add-ins, etc., among others.” Ex. 1001, col. 2, ll. 25–33. We, therefore, do not agree with the characterization of Shin as disclosing an “example” of the “determining whether” limitation as disclosed in the ’633 patent. Accordingly, we are not persuaded that we overlooked or misapprehended Petitioner’s argument regarding Shin disclosing an alleged “example” described in the ’633 patent.

Finally, we take notice of the reexamination proceeding Control No. 90/013,652 (“the ’652 reexamination”), which asserts Shin as a primary reference with respect to claims 8 and 12 of the ’633 patent. Claim 8 has been challenged in this Petition, and it recites the “determining whether” limitation that Petitioner contends Shin discloses. The argument that Shin discloses a “magic byte sequence” was presented previously in the ’652 reexamination. Moreover, the Office has issued (5/10/2016) a Notice of Intent to Issue Reexamination Certificate for failure to make a *prima facie* case of obviousness, a case which relied on Shin. This fact confirms our determination that the instrumented applet technology disclosed in Shin and the corresponding arguments concerning the “determining whether”

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.