

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

PALO ALTO NETWORKS, INC.,
Petitioner

v.

FINJAN, INC.,
Patent Owner

Patent No. 8,141,154

Inter Partes Review No. IPR2016-00151

**PETITIONER'S OPENING BRIEF
ON REMAND**

va-532786

TABLE OF CONTENTS

	Page
I. INTRODUCTION	1
II. ARGUMENT	1
A. The asserted prior art discloses the “input variable[s] includ[ing] a call to an additional function” limitation of claims 9 and 12 of the ’154 patent.	1
B. The Board misapprehended Calder’s disclosure relating to the interception of DLLs as failing to teach or suggest “wherein the modified input variable includes a call to a modified additional function instead of a call to the additional function,” as recited in claims 9 and 12.	5
C. The Board improperly required that the teachings of Calder be bodily incorporated into the teaching of Ross to establish a <i>prima facie</i> case of obviousness.	7
III. CONCLUSION.....	9

I. INTRODUCTION

Petitioner Palo Alto Networks, Inc. submits this brief, per the Board’s January 23, 2019 order, to address the patentability of claims 9 and 12 of the ’154 patent in light of the prior art, arguments presented in its original petition (“Petition”), and the Board’s discussion of those arguments in its original institution decision (“Institution Decision”). For the reasons discussed below, the evidence and arguments presented in the Petition show that claims 9 and 12 are invalid over Ross in view of Calder.

II. ARGUMENT

The Board’s Institution Decision concluded that Petitioner had not sufficiently shown that the combination of Ross and Calder teach the following limitations of claims 9 and 12 : (1) “an input that itself includes a call to an additional function as in the ’154 patent” (Institution Decision at 15); and (2) “modified input variable includes a call to a modified additional function instead of a call to the additional function” (*id.* at 17). The Board also found that Petitioner made an inadequate showing of “what modification of Ross’s hook script generator (or injector) would be needed . . . to achieve the recursive function alleged to be the result of Calder” (*id.* at 16-17). None of these issues should defeat institution.

- A. The asserted prior art discloses the “input variable[s] includ[ing] a call to an additional function” limitation of claims 9 and 12 of the ’154 patent.**

As Patent Owner discussed in its preliminary response (*See* Preliminary Response at 23-24), the '154 patent applicants explained in the specification what they meant by the “input variable[s] includ[ing] a call to an additional function” limitation. (*Id.*) This explanation is identical to what the Petition and the supporting declaration of Dr. Rubin (“Rubin Declaration”) set forth, explaining how Ross discloses this limitation in reference to Java code. The Petition additionally addresses Calder’s disclosure of this limitation. (Petition at 38-40.) The Board overlooked these arguments in denying institution. (Institution Decision at 15.)

In its Preliminary Response, the Patent Owner cited to 12:28-42 of the '154 patent to explain what the patent applicants meant by the “input variable[s] includ[ing] a call to an additional function” limitation. (Preliminary Response at 15.) That portion of the '154 patent discloses:

Malicious code may be generated within further recursive levels of function calls. For example, instead of the function call (3), ***which invokes a single function to dynamically generate JavaScript, two levels of function calls may be used.*** Consider for example, the recursive function call

```
Document.write("<h1>Document.write("<h1><SCRIPT>
Some JavaScript</SCRIPT></h1")</h1>")
```

Such a function call first calls Document.write() to generate the function call (3), and then calls Document.write() again to generate the JavaScript. If the inputs to each of the Document.write() invocations in (5) are themselves

dynamically generated at run-time, then one pass through input inspector may not detect the JavaScript.

(’154 patent at 12:28-42 (emphasis added).) In other words, the cited specification of the ’154 patent corroborates the point made by the Petition and Dr. Rubin that JavaScript code incorporates the ability to include a function as an input variable of another function.

The ’154 patent confirms that recursive function calls was a known feature of Java, as stated in the Petition and by Dr. Rubin. The *background section* of the ’154 patent, in discussing the context of the claimed invention, provides a nearly identical code example as the above specification disclosure. (’154 patent at 1:43-53.) The ’154 patent explains in the background section that “[i]n the example above, the function document.write() is used to generate HTML header text, with a text string that is generated at run-time. *If the text string generated at run-time is of the form <SCRIPTS-malicious JavaScript-/SCRIPTs then the document, write() function will insert malicious JavaScript into the HTML page that is currently being rendered by a web browser.*” This admission of the capabilities of malicious code in the background section confirms that an “input variable [that] includes a call to an additional function,” was a well-known feature and not a point of novelty as contended by the Petition and the supporting declaration of Dr. Rubin.

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.