



UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

Table with 5 columns: APPLICATION NO., FILING DATE, FIRST NAMED INVENTOR, ATTORNEY DOCKET NO., CONFIRMATION NO.
14/067.032 10/30/2013 Omer Tripp IL920120078US2_8150-0476 7430

73109 7590 12/16/2016
Cuenot, Forsythe & Kim, LLC
20283 State Road 7
Ste. 300
Boca Raton, FL 33498

EXAMINER

RASHID, HARUNUR

ART UNIT PAPER NUMBER

2497

NOTIFICATION DATE DELIVERY MODE

12/16/2016

ELECTRONIC

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

Notice of the Office communication was sent electronically on above-indicated "Notification Date" to the following e-mail address(es):

ibmptomail@iplawpro.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte OMER TRIPP and OMRI WEISMAN

Appeal 2016-000346
Application 14/067,032
Technology Center 2400

Before JEAN R. HOMERE, JEREMY J. CURCURI, and
JON M. JURGOVAN, *Administrative Patent Judges*.

JURGOVAN, *Administrative Patent Judge*.

DECISION ON APPEAL

Appellants¹ seek review under 35 U.S.C. § 134(a) from a Final Rejection of claims 1–13. We have jurisdiction under 35 U.S.C. § 6(b).

We affirm.²

¹ Appellants identify IBM Corporation as the real party in interest. (App. Br. 1.)

² Our Decision refers to the Specification filed Oct. 30, 2013 (“Spec.”), the Final Office Action mailed Dec. 26, 2014 (“Final Act.”), the Appeal Brief filed May 26, 2015 (“App. Br.”), the Examiner’s Answer mailed Aug. 11, 2015 (“Ans.”), and the Reply Brief filed Oct. 8, 2015 (“Reply Br.”).

CLAIMED INVENTION

The claims are directed to identifying stored security vulnerabilities in computer software applications. (Spec. Title.) Claim 1, reproduced below, is illustrative of the claimed subject matter:

1. A method for identifying stored security vulnerabilities in computer software applications, the method comprising:
 - providing via a first interface of a computer software application during execution of the computer software application and using a processor, test data having a characteristic of a malicious payload;
 - wherein an interaction performed with the first interface results in data written to a location within a persistent data store; and
 - wherein an interaction performed with a second interface of the computer software application results in data read from the location within the persistent data store; and
 - identifying a stored security vulnerability associated with the computer software application if the test data are written to the persistent data store at the location.

(Claims App'x.)

REJECTIONS

Claims 1–13 are provisionally rejected on the ground of nonstatutory obviousness-type double patenting as being unpatentable over claims 14–37 of co-pending US Application No. 13/743,474 filed Jan. 17, 2013. (Final Act. 5.) Because this rejection is provisional, we do not address it further in this appeal.³

³ The Board has the flexibility to reach or not reach provisional obviousness-type double patenting rejections. *See Ex parte Jerg*, Appeal 2011-000044, 2012 WL 1375142 at *3 (BPAI 2012) (informative); *Ex parte Moncla*, 95 USPQ2d 1884 (BPAI 2010) (precedential).

Appeal 2016-000346
Application 14/067,032

Claims 1, 3–8, and 10–13 stand rejected under 35 U.S.C. § 103(a) based on Maor et al. (US 2012/0255023 A1; Oct. 4, 2012) and IBM Application Security Insider: Research <http://blog.watchfire.com/wfblog/research/page/2/>, June 10, 2009 (“IBM Security.”) (Final Act. 15–18.)

Claims 2 and 9 stand rejected under 35 U.S.C. § 103(a) based on Maor, IBM Security, and Williams et al. (US 2011/0231936 A1; Sep. 22, 2011). (App. Br. 18–19.)

ANALYSIS

Claims 1, 3–8, and 10–13

A. Argument concerning interactions of the first and second interfaces of the software application being tested.

Claim 1 recites “an interaction performed with the first interface [of the computer software application] results in data written to a location within a persistent data store,” and “an interaction performed with a second interface of the computer software application results in data read from the location within the persistent data store.” (Claims App’x.) Appellants contend the recited interactions with the first and second interfaces are not taught or suggested by Maor. (App. Br. 10–11.) To the contrary, the Examiner finds the claimed features are taught by Maor. (Ans. 3–6 citing Maor, Figure 1, ¶ 49.)

Maor is directed to detecting and analyzing correlated operations in common storage, particularly in the context of cross-site scripting (XSS) attacks. (Maor Title, ¶ 9.) Maor’s objective is thus similar to Appellant’s. (Spec. ¶ 3.)

Maor, Figure 1, is shown as follows:

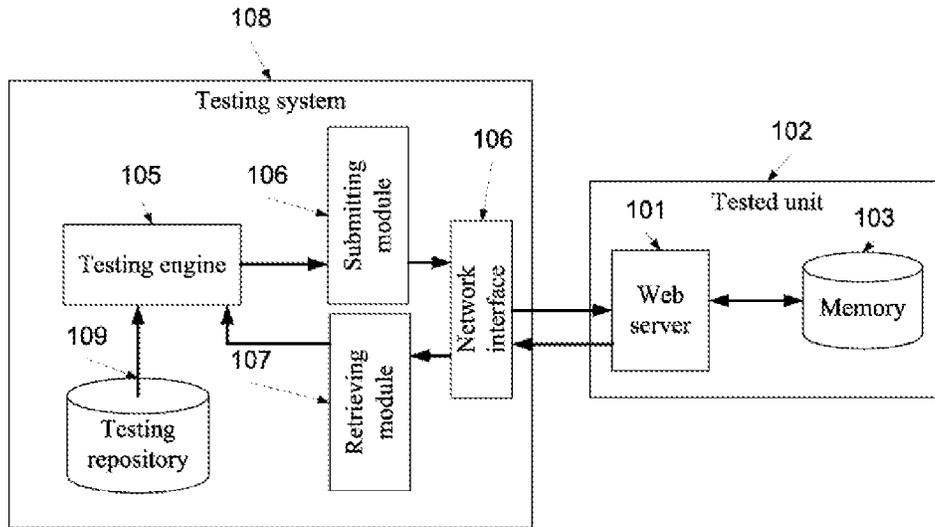


FIG. 1

Figure 1 of Maor shows a testing system **108** with submitting module **106** and receiving module **107** communicating via network interface **106** with a tested system **102** including web server **101** running applications using test data stored in a memory unit **103**.

The Examiner finds the first interface in Maor's teaching that "[t]he runtime testing system **108** further includes a submitting module **106** that is set to submit and/or monitor one or more test data inputs, in one or more input operations, via the network interface **106**." (Ans. 5 citing Maor ¶ 49.) The Examiner finds the second interface in Maor's disclosure that "[t]he runtime testing system **108** further includes a retrieving module **107** that receives or monitors one or more data outputs from the network applications executed by the tested unit **102** in one or more output operations." (*Id.*) The Examiner finds the claimed interactions in Maor's statement that "[t]he connections established by the network interface **106** allows the submitting module **106** to transmit test data inputs having uniquely identifiable data as messages to the network applications executed by the tested unit **102** and to

Appeal 2016-000346
Application 14/067,032

the retrieving module **107** to receive responses therefrom.” (*Id.*) We agree with the Examiner these findings teach the claimed interface interactions argued by Appellants.

B. Argument that Maor’s interfaces are within its testing system, not the computer software application being tested.

Appellants contend the Examiner erroneously relies on Maor’s submitting module and retrieving module that are located within the runtime testing system, not the computer software application being tested. (App. Br. 11–12.) Thus, Appellants contend that Maor does not teach or suggest the claimed first and second interfaces *of the computer software application*. (*Id.*)

Claims are given their broadest reasonable interpretation consistent with the specification. *In re Am. Acad. Of Sci. Tech. Ctr.*, 367 F.3d 1359, 1369 (Fed. Cir. 2004). At the same time, care must be exercised not to import limitations into the claims or to read a particular embodiment appearing in the written description into the claim if the claim language is broader than the embodiment. *In re Van Geuns*, 988 F.2d 1181, 1184 (Fed. Cir. 1993)(*citing In re Zletz*, 893 F.2d 319, 321 (Fed. Cir. 1989)).

Referring to the Specification, we find no special definition is given for what is meant by first and second interfaces *of the computer software application*. Although it is true that one interpretation of interface *of the computer software application* means that the interface is part of or belongs to the computer software application, the word *of* in this context can also mean “relating to.” (“Of.” def. 5a. Merriam-Webster.com. Merriam-Webster, n.d. Web. Dec. 1, 2016.) Considering the full breadth of the phrase *of the computer software application*, therefore, we find that the person of

ordinary skill would have considered the submitting module **106**, the retrieving module **107**, and the network interface **106**⁴ to be interfaces that interact with, and therefore relate to, the computer software application, and thus would have regarded each such interface as an interface *of the computer software application* as claimed. Thus, we sustain the Examiner's rejection.

We emphasize that the foregoing is not the only way a person of ordinary skill in the art would have construed the Maor reference. In Maor, from the perspective of the web server **101** in Figure 1, the incoming arrow represents an interface associated with the software application running on the web server **101**. This interface receives test data from the submitting module **106** via network interface **106** and stores it in the memory unit **103**. The outgoing arrow of the web server **101** also represents an interface associated with the software application, and this interface transmits responses to the test data from the memory unit **103** to the retrieving module **107**. Thus, for this additional reason, we are not persuaded the Examiner errs. It is well-known that computer software applications have interfaces to allow input and output of commands and data. For all of these reasons, we agree with the Examiner the claimed features are taught by Maor.

Claims 2 and 9

Appellants present the same arguments for claims 2 and 9 as previously discussed. For the stated reasons, we are not persuaded of Examiner error.

DECISION

We affirm the rejection of claims 1–13 under 35 U.S.C. § 103(a).

⁴ Both the submitting module and the network interface are labeled **106** in Maor, Figure 1, which appears to be a typographical error.

Appeal 2016-000346
Application 14/067,032

No time period for taking any subsequent action in connection with this appeal may be extended under 37 C.F.R. § 1.136(a)(1)(iv).

AFFIRMED