



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

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
14/564,062	12/08/2014	Azeem FERAZ	C204	3682
152569	7590	02/04/2020	EXAMINER	
Patterson + Sheridan, LLP - VMware 24 Greenway Plaza Suite 1600 Houston, TX 77046			PAN, YONGJIA	
			ART UNIT	PAPER NUMBER
			2145	
			NOTIFICATION DATE	DELIVERY MODE
			02/04/2020	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):

ipadmin@vmware.com
psdocketing@pattersonsheridan.com
vmware_admin@pattersonsheridan.com

UNITED STATES PATENT AND TRADEMARK OFFICE

BEFORE THE PATENT TRIAL AND APPEAL BOARD

Ex parte AZEEM FEROZ and BINYUAN CHEN

Appeal 2018-008340
Application 14/564,062
Technology Center 2100

BEFORE DENISE M. POTHIER, JENNIFER L. McKEOWN, and
JAMES W. DEJMEK, *Administrative Patent Judges*.

POTHIER, *Administrative Patent Judge*.

DECISION ON APPEAL

STATEMENT OF THE CASE

Pursuant to 35 U.S.C. § 134(a), Appellant^{1,2} appeals from the Examiner's decision to reject claims 1–22. We have jurisdiction over the pending claims under 35 U.S.C. § 6(b). We affirm.

¹ We use the word Appellant to refer to “applicant” as defined in 37 C.F.R. § 1.42(a) (2017). Appellant identifies the real party in interest as VMware, Inc. Appeal Br. 3.

² Throughout this opinion, we refer to the Final Action (“Final Act.”) mailed September 5, 2017; the Appeal Brief (“Appeal Br.”) filed January 23, 2018; the Examiner's Answer (“Ans.”) mailed June 26, 2018; and the Reply Brief (“Reply Br.”) filed August 21, 2018.

CLAIMED SUBJECT MATTER

The claimed subject matter relates to protecting against cyber attacks, such as malware. Spec. ¶ 1. Defense mechanisms (e.g., anti-malware solutions) may employ blacklisting strategies or whitelisting strategies. *Id.* ¶¶ 1–2. In whitelisting strategies, “only well-known applications are given execution privileges.” *Id.* ¶ 2.

The main challenge with adopting a whitelisting strategy is that as new applications emerge, many of their runtime behavior remain unknown. Before such unknown applications, which are referred to as “gray” applications, can be added to the whitelist, their runtime behavior needs to be first examined and classified as being safe by the system administrator.

Id. ¶ 2. The claims are directed to “gray applications that are selected to be executed in a first virtual machine are executed and monitored in a second virtual machine that is a clone of the first virtual machine, and classified according to their monitored behavior.” *Id.* ¶ 7.

Claim 1 is reproduced below:

1. A method of performing admission control of an application that has been selected for execution in a first virtual computing instance on a host machine, comprising:
 - creating a second virtual computing instance on the host machine that is a clone of the first virtual computing instance;
 - executing the application in the second virtual computing instance;
 - during execution of the application in the second virtual computing instance, monitoring execution behavior of the application;
 - determining from the monitored execution behavior of the application whether or not the application is to be approved for execution in the first virtual computing instance;

if the application is approved for execution in the first virtual computing instance, executing the application in the first virtual computing instance; and

if the application is not approved for execution in the first virtual computing instance, transmitting an error message to the first virtual computing instance and not executing the application in the first virtual computing instance.

Appeal Br. 15 (Claims App.).

We have reviewed the Examiner's rejection in light of Appellant's arguments presented in this appeal. Arguments which Appellant could have made, but did not make in the Brief are deemed to be waived. *See* 37 C.F.R. § 41.37(c)(1)(iv). On the record before us, we are unpersuaded the Examiner has erred and adopt as our own the findings and reasons set forth in the rejections from which the appeal is taken and in the Examiner's Answer.

REFERENCES

The prior art relied upon by the Examiner is:

Name	Reference	Date
Ghosh	US 2010/0122343 A1	May 13, 2010
Kanno	US 2011/0252278 A1	Oct. 13, 2011
Teddy	US 2014/0283066 A1	Sept. 18, 2014
Huang	US 8,966,632 B1	Feb. 24, 2015 (filed Feb. 17, 2012)
Lukacs	US 2015/0178497 A1	June 25, 2015 (filed Dec. 20, 2013)

OBVIOUSNESS REJECTION OVER TEDDY, KANNO, AND HUANG

Because Appellant argues claims 1, 2, 4, 5, 8–10, 12, 13, 16–18, and 21 as a group (Appeal Br. 8–12), we select claim 1 as representative. *See* 37 C.F.R. § 41.37(c)(1)(iv).

Regarding representative claim 1, the Examiner finds that Teddy teaches many of its limitations. Final Act. 2–4 (Teddy ¶¶ 22, 36, 40, 79, 81). The rejection turns to Kanno in combination with Teddy to teach the recited “application” and creating a computing instance that is “a clone” of a computing instance on a host machine as claim 1 requires. *Id.* at 4–5 (citing Kanno ¶¶ 74–75). The rejection also turns to Huang in combination with Teddy and Kanno to teach the “transmitting an error message” step in the claims. *Id.* at 5 (citing Kanno ¶ 110 and Huang 6:28–31), 7.

Appellants argues Kanno alone or in combination with Teddy does not teach or suggest “a second virtual computing instance . . . that is a clone of the first virtual computing instance” and “determining from the monitored execution behavior of the application whether or not the application is to be approved for execution in the first virtual computing instance” as claim 1 recites. Appeal Br. 10–11; Reply Br. 3–4. Appellant further argues the Examiner relies on impermissible hindsight when proposing to combine Kanno with Teddy. Appeal Br. 9, 11; Reply Br. 4–5.

ISSUES

(1) Under § 103, has the Examiner erred in rejecting claim 1 by finding that Teddy, Kanno, and Huang collectively would have taught or suggested: (a) “creating a second virtual computing instance . . . that is a clone of the first virtual computing instance” and (b) “determining from the

monitored execution behavior of the application whether or not the application is to be approved for execution in the first virtual computing instance”?

(2) Is the Examiner’s reason to combine Teddy and Kanno supported by articulated reasoning with some rational underpinning to justify the Examiner’s obviousness conclusion?

ANALYSIS

We begin by noting the Specification states “every executable” is “referred to herein as [an] ‘application.’” Spec. ¶ 1. Teddy teaches its files, including those loaded and executed in the sandbox or another virtual environment (e.g., anti-malware system), can be executable. Teddy ¶¶ 32, 43–44, 50, 72. Although the Examiner stated “Teddy fails to teach the file is an application,” we determine Teddy teaches and suggests its file, as mapped by the Examiner (Final Act. 3), is an “executable” based on the disclosure. Thus, in light of the Specification, Teddy teaches the recited “application” as claimed. *See In re Am. Acad. of Sci. Tech Ctr.*, 367 F.3d 1359, 1364 (Fed. Cir. 2004) (stating a claim is given its broadest reasonable construction “in light of the specification as it would be interpreted by one of ordinary skill in the art.”).

Turning to Appellant’s contentions, Appellant argues neither Kanno nor the combination of Teddy and Kanno teaches or suggests “a second virtual computing instance . . . that is a clone of the first virtual computing instance” in claim 1. Appeal Br. 10–11. We disagree. Kanno explicitly teaches generating “a clone of” a guest OS (e.g., guest OS 33 (#2) is a clone of guest OS 33 (#1) in Fig. 6) and that a guest OS executing an application program can be a virtual machine (e.g., a virtual computing instance).

Kanno ¶¶ 3, 74–75, Fig. 6, *cited in part in* Final Act. 4. Appellant does not address these cited passages in Kanno specifically or the Examiner’s findings and conclusions in this regard. *See* Final Act. 4; Appeal Br. 9–10 (citing Kanno ¶¶ 23, 26–28, Abs.).

Appellant further asserts neither Kanno nor the combination of Teddy and Kanno teaches or suggests “determining from the monitored execution behavior of the application whether or not the application is to be approved for execution in the first virtual computing instance” in claim 1. Appeal Br. 10–11. We are not persuaded. Appellant’s individual attack on Kanno concerning the “determining” step in claim 1 (*see* Appeal Br. 9–11) is not persuasive because the rejection relies on Teddy *and* Kanno collectively to teach this disputed step. *See* Final Act. 3–5 (citing Teddy ¶¶ 79, 81 and Kanno ¶¶ 74–75). In particular, the rejection relies on Teddy mainly for the “monitoring” and “determining” steps (*see* Final Act. 3) and turns to Kanno in combination with Teddy to teach the recited “application” within the “monitoring” and “determining” steps (*see* Final Act. 4).

To elaborate, Teddy teaches a scanner that uses virtual machines (e.g., a second virtual computing instance) to implement the anti-malware support system in a virtualized environment, to scan files³ from host devices to observe how the file operates when various simulated events take place in the virtualized environment, and to obtain information related the file behavior (Teddy ¶ 79, *cited in* Final Act. 3) (e.g., “monitoring execution behavior of” a file during its execution “in the second virtual computing instance” as recited). Teddy also teaches scan results can be pushed to the

³ As previously noted, Teddy’s file is an “application” consistent with how the Specification describes an executable/application. *See* Spec. ¶ 3.

host device where the host device can complete the action on the file based on the scanning (e.g., “determining from the monitored execution behavior” whether or not the file is to be “approved for execution in the first virtual computing device” as recited). *Id.* ¶ 81, *cited in* Final Act. 3). We additionally note that Teddy teaches a host device can include a computing device that may be adapted to execute virtual machines for virtualize execution of a particular operating system. Teddy ¶ 22, *cited in* Final Act. 3. Appellant does not address these cited passages in Teddy specifically or the Examiner’s findings and conclusions in this regard (*see* Final Act. 3). Appeal Br. 11 (citing only Teddy ¶ 43, Abs., Title).

Appellant further asserts Teddy and Kanno, when combined, would at best result in an anti-malware support system that executes the file/application in multiple cloned virtual environments but that is not for “a second virtual computing instance ... that is a clone of the first virtual computing instance”, “determining from the monitored execution behavior of the application [(during execution of the application in the second virtual computing instance)] whether or not the application is to be approved for execution in the first virtual computing instance,” as recited in claim 1.

Appeal Br. 11; Reply Br. 4–5. For the above reasons, we disagree. In particular, Appellant has not address the above-cited passages in Teddy and Kanno and, when combined, how they would result in the disputed claim limitations.

Appellant also argues that an ordinarily skilled artisan would not have combined Teddy and Kanno (Appeal Br. 9) and that the Examiner relies on impermissible hindsight in proposing the combination (*id.* at 9, 11). We are not persuaded. As noted above and in the rejection, Teddy teaches scanning

how the file operates when *various simulated events* takes place within a virtual environment (Teddy ¶ 79, *cited in* Final Act. 3) and Kanno teaches cloning virtual computing instances (e.g., a guest OS) (Kanno ¶¶ 74–75, *cited in* Final Act. 4) for testing an application under *various expected test conditions* (Kanno ¶ 23, Abs.). As the Examiner also discusses, Teddy teaches monitoring execution behavior of a file/application in a second virtual computing instance under various simulated events (e.g., various test condition), and Kanno teaches one technique for performing those various simulated events/tests (e.g., using cloning of a first virtual computing instance). *See* Teddy ¶ 79; Kanno ¶¶ 23, 74–75. That is, as the Examiner explains, both Teddy and Kanno concern testing a file/application in simulated environments. *See* Final Act. 4–5 (stating “Teddy and Kanno teach a system and method for testing”).

We thus disagree with Appellant that the proposed rejection combines Kanno with Teddy “for a whole different purpose.” Reply Br. 3. By cloning the first virtual computing instance as a second virtual computing instance as taught by Kanno and as discussed above, one skilled in the art would have recognized the second virtual computing instance creates a simulated environment (e.g., test environment) identical or similar to that of the first virtual computing instance and thus “provid[es] an accurate simulation of a computing instance” (Final Act. 5) to determine whether or not to approve for execution the application in the first virtual computing instance. *See id.* at 4–5. Additionally, as Appellant acknowledges, Kanno clones first virtual computing device to test the application under different testing conditions in a more efficient and quick manner (*see* Appeal Br. 10; *see* Kanno ¶¶ 118–119), providing yet another reason one skilled in art would have recognized

for combining Kanno’ teaching with Teddy to test the various simulated environments (e.g., to improve testing efficiency and speed). We therefore disagree with Appellant that the Examiner gleans knowledge from the Specification when proposing the combination of Teddy and Kanno or that the proposed combination does not achieve providing an accurate simulation of a computing instance. *See* Appeal Br. 9, 11; Reply Br. 3–4.

Accordingly, the record establishes a reason with some rational underpinning for combining Kanno with Teddy to teach the disputed “creating” and “determining” steps in claim 1 and to support a legal conclusion of obviousness for representative claim 1.

For the first time in the Reply Brief, Appellant argues *Teddy* does not teach “creating a second virtual computing instance on the host machine that is a clone of the first virtual computing instance,” “executing the application in the second virtual computing instance;” and “determining from the monitored execution behavior of the application whether or not the application is to be approved for execution in the first virtual computing instance” as recited in claim 1. Reply Br. 3. Appellant further argues “claim 1 provides certain advantages over Teddy” and that Teddy does not discuss cloning the first virtual computing instance and monitoring application’s execution behavior in a clone to determine whether or not to approve the application’s execution in the first virtual computing device as recited. *Id.*

In the Appeal Brief, Appellant did not dispute Teddy alone failed to the “executing” step or the “monitoring” step and did not argue Teddy *alone* failed to teach the “creating a second virtual computing instance . . . that is a clone of the first virtual computing instance” step or “determining” step. *See* Appeal Br. 9–11. These arguments in the Reply Brief are, therefore, waived.

See 37 C.F.R. § 41.41(b)(2) (stating “[a]ny argument raised in the reply brief which was not raised in the appeal brief, or is not responsive to an argument raised in the examiner’s answer . . . will not be considered by the Board for purposes of the present appeal, unless good cause is shown.”). In any event, as previously discussed, the rejection turns to Teddy *and* Kanno to teach these limitations, including recitation to the “application” and “creating” a clone of the first virtual computing instance. Final Act. 3–4 (citing Teddy ¶ 79 and Kanno ¶¶ 74–75). Appellant has not persuaded us sufficiently that the proposed combination would not teach or suggest the disputed language.

For the foregoing reasons, Appellant has not persuaded us of error in the rejection of independent claim 1 and claims 2, 4, 5, 8–10, 12, 13, 16–18, and 21, which are not argued separately.

OBVIOUSNESS REJECTION OVER TEDDY, KANNO, HUANG, AND GHOSH

Claims 6, 7, 14, 15, 19, and 20 are rejected under 35 U.S.C. § 103 based on Teddy, Kanno, Huang, and Ghosh. Final Act. 8–10. For claim 6, the Examiner found that Teddy, Kanno, and Huang teach the recited “monitoring includes examining for compliance with security polic[ies]” but do not teach the remaining features in this claim, turning to Ghosh. *Id.* at 8–9 (citing Teddy ¶ 118 and Ghosh ¶ 79). Appellant argues Ghosh teaches virtualization sandboxing is undesirable and teaches away from creating a clone of a virtual computing instance in which to execute an application. Appeal Br. 13 (citing Ghosh ¶¶ 8, 22, 24); Reply Br. 6 (same). We are not persuaded.

Dependent claim 6 recites “wherein monitoring the execution behavior of the application includes intercepting file events that are generated during execution of the application in the second virtual computing instance, and the file events are examined for compliance with security policies.” Appeal Br. 16 (Claims App.). Notably, this limitation further limits the “monitoring” step, not the “creating a second virtual computing instance . . . that is a clone of the first virtual computing instance” in claim 1. *Id.* at 15 (Claims App.). As previously discussed, the Examiner relied on Teddy and Kanno to teach the “creating” a clone feature. Final Act. 3–5. Thus, Appellant’s arguments concerning claim 6 and whether Ghosh teaches cloning a virtual computing instance as recited do not address the rejection as presented. *See* Appeal Br. 13; *see* Reply Br. 6–7

We further do not see how cited paragraph 79 in Ghosh (Final Act. 8–9), which does not address cloning a virtual computing instance (*see* Ghosh ¶ 79), teaches away from the presented combination. Appellant does not address Examiner’s findings related to Ghosh and paragraph 79 (Appeal Br. 13), which we find reasonably teach or suggest the features in claim 6 when combined with Teddy, Kanno, and Huang. *See* Final Act. 8–9.

In any event, although Ghosh discusses hardware virtualization sandboxing may have some problems (*see* Ghosh ¶ 8), rendering a system inferior in some ways does not demonstrate Ghosh teaches away from claim 6 as presented. *See In re Gurley*, 27 F.3d 551, 553 (Fed. Circ. 1994). Moreover, Teddy’s virtualization of its computing instances are not limited to sandboxing. Teddy ¶ 43 (discussing “a sandbox or *other* virtual or protected environment”) (emphasis added); *see* Ans. 13–14 (citing Ghosh ¶¶ 30, 56, 67, 79 and discussing improving on the hardware sandboxing).

For the foregoing reasons, Appellant has not persuaded us of error in the rejection of claim 6 and claims 7, 14, 15, 19, and 20, which are not argued separately.

REMAINING OBVIOUSNESS REJECTION

Claims 3, 11, and 22 depend from claims 1, 9, and 17 respectively and are rejected under 35 U.S.C. § 103 based on Teddy, Kanno, Huang, and Lukacs. Final Act. 7–10. Appellant asserts this rejection has been overcome for the reasons set forth related to independent claims 1, 9, and 17, and Lukacs does not cure the alleged deficiencies. Appeal Br. 12.

Because we sustain the rejection of claims 1, 9, and 17, we sustain the rejection of claims 3, 11, and 22 for similar reasons.

DECISION SUMMARY

In summary:

Claims Rejected	35 U.S.C. §	References	Affirmed	Reversed
1, 2, 4, 5, 8–10, 12, 13, 16–18, 21	103	Teddy, Kanno, Huang	1, 2, 4, 5, 8–10, 12, 13, 16–18, 21	
3, 11, 22	103	Teddy, Kannon, Huang, Lukacs	3, 11, 22	
6, 7, 14, 15, 19, 20	103	Teddy, Kanno, Huang, Ghosh	6, 7, 14, 15, 19, 20	
Overall Outcome			1–22	

Appeal 2018-008340
Application 14/564,062

TIME PERIOD FOR RESPONSE

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

AFFIRMED