NOTE: This disposition is nonprecedential.

United States Court of Appeals for the Federal Circuit

TRUSTED KNIGHT CORPORATION,
Plaintiff-Appellant

v.

INTERNATIONAL BUSINESS MACHINES CORPORATION, TRUSTEER INC.,
Defendants-Appellees

2016-1510

Appeal from the United States District Court for the District of Delaware in No. 1:14-cv-01063-LPS, Chief Judge Leonard P. Stark.

Decided: March 7, 2017

PAUL R. GUPTA, Reed Smith LLP, San Francisco, CA, argued for plaintiff-appellant. Also represented by GERARD M. DONOVAN, Washington, DC; R. ERIC HUTZ, RUDOLF EDWARD HUTZ, Wilmington, DE; JAMES CHRISTOPHER MARTIN, Pittsburgh, PA.

DAVID A. NELSON, Quinn Emanuel Urquhart & Sullivan, LLP, Chicago, IL, argued for defendants-appellees.

Also represented by JOHN THOMAS MCKEE, ALEXANDER RUDIS, New York, NY.

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

STOLL, *Circuit Judge*.

Trusted Knight Corporation appeals from a stipulated judgment of invalidity from the United States District Court for the District of Delaware following adverse indefiniteness rulings against its asserted patent, U.S. Patent No. 8,316,445. We affirm.

BACKGROUND

I.

Trusted Knight owns the '445 patent, which generally discloses "systems and methods for protection against the operation of malware commonly used in identity-theft and cyber-fraud." '445 patent col. 1 ll. 24–26. More specifically, the '445 patent purports to protect against a type of malware known as key logging.

According to the '445 patent, key logging "is a method of capturing keyboard input to a computer or computing device" and "is a common technique for obtaining passwords and sensitive information using unauthorized software." *Id.* at col. 1 ll. 57–60. There are many key-logging techniques, "including hooking various operating system Application Programming Interfaces (APIs) and system drivers, screen capture, and form grabbing and hook based keystroke logging." *Id.* at col. 2 ll. 1–4. The '445 patent describes in detail two types of key logging—hook-based key logging and form-grabbing key logging.

The '445 patent describes hook-based key logging as the insertion of a system API hook into an API stack, which allows the key logger to record all keystroke data

passing through an operating system driver. The logger saves this data to a text file, which can subsequently be sent to malefactors at a remote location. Because this method of key logging indiscriminately records all of the keystroke data, it often results in a large volume of data that is burdensome to store. Additionally, this voluminous data can be "difficult to search for the purpose of extracting the very small percentage of data that represents credential and password information." *Id.* at col. 2 ll. 24–26. "As a result, malefactors have fine-tuned their malware to meet these challenges and better reduce the large take of useless data stolen by their malware." *Id.* at col. 2 ll. 26–28.

One such fine-tuned version of key logging is form-grabbing key logging, which the '445 patent describes as the insertion of a hook that captures form data solely from form data inputs. "The form information being stolen is, essentially, those forms used for online banking and other online commerce that require users to enter personal information, card data, passwords, reminder questions, and mother's maiden names." *Id.* at col. 2 ll. 31–35. For example, "when a user submits data to a legitimate banking website using web forms, a form-grabbing key logger that is monitoring the web browser can grab the submitted data by injecting a hook and hooking API functions within the browser." *Id.* at col. 2 ll. 60–64. The patent further explains that sophisticated cyber criminals have come to prefer form-grabbing key loggers because: (1) they are resistant to detection and lack effective countermeasures; (2) they substantially reduce the volume of captured data; and (3) they capture the vast majority of credentials criminals want, since almost all credentials used for online transactions are inputted into a web form.

The '445 patent describes various prior art methods used to counteract key logging malware. Many of these methods "are available to detect and/or disable hook-

based key loggers." *Id.* at col. 3 ll. 15–16. For example, "[o]ne method used is the unhooking of API's that insert themselves into the API stack." *Id.* at col. 3 ll. 17–19. The '445 patent warns, however, that this method does not protect the user when the malware inserts a hook at the first instance in the API stack and it is also ineffective against form-grabbing key loggers.

Another method works by launching a new process when it detects a hook-based key logger, whereby the keystroke data is passed through the new process and bypasses the keystroke-logger hook. The '445 patent warns, however, that this method can cause system instability and can be counteracted by key loggers.

The invention, as described in the '445 patent specification, allegedly improves upon the prior art by preventing the actions of form-grabbing and hook-based key loggers in a way that "does not depend on the detection of malware at all." *Id.* at col. 3 ll. 60–61. One embodiment of the invention prevents form-grabbing key logging. Specifically, the software: (1) identifies forms on a called web page; (2) connects to each form submission event; (3) clears all form inputs marked with INPUT or PASSWORD; (4) provides the user-inputted data to the designated receiving party, such as a bank; and (5) ensures that all password form fields are cleared from the API chain.

Another embodiment of the invention prevents the actions of both hook-based and form-grabbing key loggers. The software hooks the kernel keyboard driver where it intercepts and encrypts the keystroke data received from the keyboard. This encrypted data is then sent to the intended application, such as a web browser, where the keystrokes are decrypted and presented to the web form for submission to the designated receiving entity.

The '445 patent has three independent claims: claims 1, 22, and 23. Claim 1 of the '445 patent recites:

> 1. A software program embedded in a non-transitory microprocessor-readable storage medium and executable by a microprocessor to prevent software key logging comprising:
>
> > a software module that inserts and executes predetermined software processes at a zero-ring level in an application programming interface ("API") stack of a browser, said software processes including:
> >
> > > a process of detecting a browser form submission initiation call event at the zero-ring level, wherein the form submission initiation call event takes a form of an on Submit call or a BeforeNavigate call;
> > >
> > > a process of intercepting data inputs keyed in by a user at the zero-ring level; and
> > >
> > > a process of (1) submitting the keyed-in data to a designated entity through the API stack while (2) clearing confidential data from intercepted data at the zero-ring level prior to a subsequent transmission, which does not contain said confidential data, *in response to the software key logging through the API stack to an internet communication port*.

Id. at col. 11 ll. 33–53 (disputed claim term italicized). Claim 22 recites:

> 22. A software program embedded in a non-transitory microprocessor-readable storage medi-

um and executable by a microprocessor to prevent software key logging comprising:

> a software module that inserts and executes predetermined software processes at a zero-ring level in an application programming interface ("API") stack of a browser, said software processes including:
>
> > a process of inserting an initial hook which works within the 0-Ring level and prevents any other hooks from inserting at the 0-Ring level;
> >
> > a process of detecting a browser form submission initiation call event at the zero-ring level, wherein the form submission initiation call event takes a form of an onSubmit call or a BeforeNavigate call;
>
> a process of intercepting and encrypting data inputs keyed in by a user at the zero-ring level;
>
> > a process of passing the encrypted data to a 3-ring level where a hook inserted by a hook-based key logger;
> >
> > a process of decrypting data which passed via the 3-ring level; and
> >
> > a process of submitting the decrypted data to a designated entity through the API stack to an internet communication port.

Id. at col. 12 l. 57 – col. 13 l. 13 (disputed claim term italicized).[1]

II.

Trusted Knight sued International Business Machines Corporation and Trusteer, Inc. for infringement of the '445 patent. The district court conducted a *Markman* hearing and issued a claim construction order construing four disputed claim terms. *Trusted Knight Corp. v. Int'l Bus. Machs. Corp.*, No. 14-1063-LPS-CJB, 2015 WL 7307134 (D. Del. Nov. 19, 2015). Relevant here, the district found the following disputed claim terms indefinite: (1) "in response to the software key logging through the API stack to an internet communication port," recited in independent claims 1 and 23; and (2) "a process of passing the encrypted data to a 3-ring level where a hook inserted by a hook-based key logger," recited in independent claim 22. *Id.* at *4–7.

Following the district court's claim construction order, the parties filed a stipulated final judgment of invalidity. Trusted Knight appeals the district court's indefiniteness rulings, and we have jurisdiction under 28 U.S.C. § 1295(a)(1).

DISCUSSION

On appeal, Trusted Knight argues that the district court erred in ruling that the claim limitation, "in response to the software key logging through the API stack to an internet communication port," recited in claims 1 and 23, is indefinite. Trusted Knight also argues that the district court erred in ruling that the claim limitation, "a process of passing the encrypted data to a 3-ring level

[1] We do not reproduce claim 23 because the parties agree that it "recites a method counterpart to claim 1." Appellant Br. 16; Appellee Br. 6.

where a hook inserted by a hook-based key logger," recited in claim 22, is indefinite. We address these arguments in turn.

I.

Pursuant to 35 U.S.C. § 112, ¶ 2, a patent specification must "conclude with one or more claims particularly pointing out and distinctly claiming the subject matter which the applicant regards as his invention."[2] The Supreme Court has held this definiteness provision "to require that a patent's claims, viewed in light of the specification and prosecution history, inform those skilled in the art about the scope of the invention with reasonable certainty." *Nautilus, Inc. v. Biosig Instruments, Inc.*, 134 S. Ct. 2120, 2129 (2014).

As the Supreme Court explained, the definiteness requirement "entails a 'delicate balance.'" *Id.* at 2128 (quoting *Festo Corp. v. Shoketsu Kinzoku Kogyo Kabushiki Co.*, 535 U.S. 722, 731 (2002)). On one hand, "the inherent limitations of language," *id.* (citing *Festo*, 535 U.S. at 731), must be taken into account, recognizing that "[s]ome modicum of uncertainty . . . is the 'price of ensuring the appropriate incentives for innovation,'" *id.* (quoting *Festo*, 535 U.S. at 732). On the other hand, "a patent must be precise enough to afford clear notice of what is claimed, thereby 'appris[ing] the public of what is still open to them.'" *Id.* at 2129 (alteration in original) (quoting *Markman v. Westview Instruments, Inc.*, 517 U.S. 370, 373 (1996)). "Otherwise there would be '[a] zone of

---

[2] Because the '445 patent was filed before the adoption of the Leahy–Smith America Invents Act, Pub. L. No. 112–29, § 4(e), 125 Stat. 284, 296-97 (2011), the previous version of § 112 governs. *See AbbVie Deutschland GmbH & Co. KG v. Janssen Biotech, Inc.*, 759 F.3d 1285, 1290 n.3 (Fed. Cir. 2014).

uncertainty which enterprise and experimentation may enter only at the risk of infringement claims.'" *Id.* (alteration in original) (quoting *United Carbon Co. v. Binney & Smith Co.*, 317 U.S. 228, 236 (1942)).

Here, the district court held that the claim term "in response to the software key logging through the API stack to an internet communication port" is indefinite because it is unclear what "in response to the software key logging" requires. We agree.

Trusted Knight argues that the claim limitation does not require responding to or detecting actual instances of malware on a user's computer. This argument finds support in the specification, which emphasizes that "[t]he solution of the present invention does not depend on detection of malware at all." '445 patent at col. 3 ll. 59–61. As the district court asked, however, "if the invention is not responding to malware, then what *is* happening '*[i]n response to* the software key logging?'" *Trusted Knight*, 2015 WL 7307134, at *5. Trusted Knight answers that the claimed invention responds to the threat of malware: "The specification clarifies that the 'in response to' term should be read as being in response to the threat of key logging malware, whether detected or not, and whether present or not." Appellant Br. 32. This position, however, is undermined by the claim language itself, which as the district court noted, "suggests an event ('response') triggered by another event ('logging')." *Trusted Knight*, 2015 WL 7307134, at *5. The claim limitation does not even refer to the "threat" or "potential presence" of key logging; rather it refers to key logging.

After review of the relevant intrinsic evidence and the parties' positions, we agree with the district court that the meaning of this claim limitation is not reasonably certain. The "in response to" claim term does not "appris[e] the public of what is still open to them," *Nautilus*, 134 S. Ct. at 2129 (quoting *Markman*, 517 U.S. at 373), and creates

"[a] zone of uncertainty which enterprise and experimentation may enter only at the risk of infringement claims," *id.* (quoting *United Carbon*, 317 U.S. at 236). Accordingly, we hold that claims 21 and 23 of the '445 patent are invalid for indefiniteness because, when read in light of the specification, Trusted Knight has failed to inform with reasonable certainty those skilled in the art about the scope of its invention.

II.

Trusted Knight also argues that the disputed claim term, "a process of passing the encrypted data to a 3-ring level where a hook inserted by a hook-based key logger," recited in claim 22, is not indefinite. Both parties concede that this claim term contains a typographical error. *See* Appellant Br. 6; Appellee Br. 2. Specifically, the claim term is missing a verb between "hook" and "inserted."

"It is well-settled law that, in a patent infringement suit, a district court may correct an obvious error in a patent claim." *CBT Flint Partners, LLC v. Return Path, Inc.*, 654 F.3d 1353, 1358 (Fed. Cir. 2011) (citing *I.T.S. Rubber Co. v. Essex Rubber Co.*, 272 U.S. 429, 442 (1926)). A district court can only correct a patent, however, if: "(1) the correction is not subject to reasonable debate based on consideration of the claim language and the specification and (2) the prosecution history does not suggest a different interpretation of the claims." *Novo Indus. v. Micro Molds Corp.*, 350 F.3d 1348, 1357 (Fed. Cir. 2003).

The district court held that this claim limitation was not amenable to correction because the correction was subject to reasonable debate based on consideration of the claim language and the specification. We agree.

Trusted Knight argues that the claim limitation can be corrected by inserting "is" in between "hook" and "inserted," so the claim limitation would read "a process of passing the encrypted data to a 3-ring level where a hook

[is] inserted by a hook-based key logger." Such a correction, however, would suggest that the "process of passing the encrypted data to a 3-ring level" occurs only when a hook is actually inserted. But as noted above in the discussion of the "in response to" limitation, the specification emphasizes that the invention operates regardless of the presence or detection of malware. This ambiguity, noted by the district court, demonstrates that Trusted Knight's correction to the claim language is subject to reasonable debate.

Indeed, other possible corrections appear to be feasible. The district court, for example, explained that it could correct the claim limitation by placing "could be" between "hook inserted," such that the claim limitation would read "a process of passing the encrypted data to a 3-ring level where a hook [could be] inserted by a hook-based key logger." While this correction would seem to be consistent with the specification and Trusted Knight's position that the claimed invention operates even in the absence of malware, it would create a different claim scope than Trusted Knight's proposed correction of adding "is." Thus, because the proposed construction is subject to reasonable debate, the disputed claim limitation is not amenable to correction.

Additionally, as the claim limitation stands uncorrected, it does not inform those skilled in the art about the scope of the invention with reasonable certainty. We accordingly hold that claim 22 of the '445 patent is invalid for indefiniteness because, when read in light of the specification, Trusted Knight has failed to inform with reasonable certainty those skilled in the art about the scope of its invention.

CONCLUSION

We have considered Trusted Knight's remaining arguments and determined that they lack merit. Because claims 1, 22, and 23 do not reasonably inform those

skilled in the art about the scope of the invention with reasonable certainty, they are indefinite, and we affirm the district court's judgment of indefiniteness.

AFFIRMED

COSTS

Costs to Appellees.