

IN THE UNITED STATES DISTRICT COURT  
FOR THE DISTRICT OF DELAWARE

FINJAN, INC., )  
)  
Plaintiff, )  
)  
v. ) C.A. No. 18-1519 (MN)  
)  
RAPID7, INC., et al., )  
)  
Defendants. )

**MEMORANDUM ORDER**

At Wilmington this 5th day of February 2020:

IT IS HEREBY ORDERED that the claim terms of U.S. Patent Nos. 8,677,494 (“494 Patent”), 8,079,086 (“086 Patent”), 7,975,305 (“305 Patent”), and 7,613,918 (“918 Patent”) with agreed-upon constructions are construed as follows (*see* D.I. 116 at Exhibit 1):

1. “Downloadable[s]” means “an executable application program, which is downloaded from a source computer and run on a destination computer” (‘494 Patent, claim 10; ‘086 Patent, claims 1, 17, 41, & 42);
2. “Destination computer” means “separate computer receiving the [appended downloadable / Downloadable and a representation of the Downloadable security profile data]” (‘086 Patent, claims 1, 17, 24, 41, & 42);
3. “Database” means “a collection of interrelated data organized according to a database schema to serve one or more applications” (‘494 Patent, claim 10; ‘305 Patent, claims 1, 13, & 25; ‘086 Patent, claims 41 & 42);
4. “Parse tree” means “a hierarchical structure of interconnected nodes built from scanned content” (‘408 Patent, claims 1, 22, & 35);
5. “Code-A” means “potentially malicious executable code” (‘918 Patent, claims, 1, 12, 22, 28, & 33);
6. “Code-B” means “executable wrapper code” (‘918 Patent, claims 1, 22, 28, & 33);

7. “Code-C” means “combined code” (’918 Patent, claims 1, 12, 22, 28, & 33); and
8. “Security context” means “an environment in which a software application is run, which may limit resources that the application is permitted to access or operations that the application is permitted to perform” (’918 Patent, claims 1, 12, 22, 28, & 33).

Further, as announced at the hearing on January 22, 2020, IT IS HEREBY ORDERED that the disputed claim terms of the ’494 Patent, ’086 Patent, ’305 Patent, ’918 Patent, and U.S. Patent Nos. 8,141,154 (“’154 Patent”) and 8,225,408 (“’408 Patent”) are construed as follows:

1. “Downloadable scanner” means “software that decomposes code using conventional parsing techniques to identify suspicious patterns or suspicious computer operations” (’494 Patent, claim 10; ’086 Patent, claim 24);
2. “list of suspicious computer operations [that may be attempted by]” means “list of computer operations derived from a received Downloadable that are deemed hostile or potentially hostile” (’494 Patent, claim 10; ’086 Patent, claims 1, 17, 24, 41, & 42);
3. “appended Downloadable” means “downloadable with a representation of the Downloadable security profile data attached” (’086 Patent, claim 1);
4. “first function” / “second function” shall have its plain and ordinary meaning and no construction is necessary (’154 Patent, claims 1, 4, 6, & 10);
5. “content processor” means “a processor that processes content” (’154 Patent, claims 1 & 6);
6. “a call to a first function” means “a statement or instruction in the content, the execution of which causes the function to provide a service” (’154 Patent, claims 1, 4, 6, & 10);
7. “a rule-based content scanner” means “a scanner that is able to adapt itself dynamically to scan a specific type of content” (’305 Patent, claim 1);
8. “patterns of types of tokens” means “patterns of categories of tokens” (’305 Patent, claims 1, 13, & 25);
9. “scanner” means “a scanner that is able to adapt itself dynamically to scan a specific type of content” (’408 Patent, claims 1 & 22);

10. “parser rules” means “patterns of tokens that form syntactical constructs of program code” (’408 Patent, claims 1 & 22);
11. “dynamically building” shall have its plain and ordinary meaning, “non-static building” (’408 Patent, claims 1, 22, 29, & 35);
12. “dynamically detecting” shall have its plain and ordinary meaning, “non-static detecting” (’408 Patent, claims 1, 22, 29, & 35);
13. “incoming stream of program code” shall have its plain and ordinary meaning (’408 Patent, claims 1, 22, 29, & 35);
14. “instantiating, by the computer, a scanner for the specific programming language / instantiating a scanner for the specific programming language” means “generating or requesting a scanner that can scan the specific programming language” (’408 Patent, claims 1 & 22);
15. “executable wrapper code / wrapper executable code” means “executable code that wraps another executable” (’918 Patent, claims 1, 12, 22, 28, & 33);
16. “computer account(s)” shall have its plain and ordinary meaning, “computer account(s)” (’918 Patent, claims, 1, 12, 22, 28, & 33);
17. “combining” / “combine” shall have its plain and ordinary meaning (’918 Patent, claim 1).

The parties briefed the issues (*see* D.I. 76) and submitted an appendix that included briefing and orders from numerous related cases and proceedings (*see* D.I. 77). The parties also submitted a Joint Claim Construction Chart containing intrinsic evidence (*see* D.I. 55-58). The Court carefully reviewed all submissions in connection with the parties’ contentions regarding the disputed claim terms, heard oral argument (*see* D.I. 118, 121), and applied the following legal standards in reaching its decision.

### **LEGAL STANDARDS**

“[T]he ultimate question of the proper construction of the patent [is] a question of law,” although subsidiary fact-finding is sometimes necessary. *Teva Pharms. USA, Inc. v. Sandoz, Inc.*, 574 U.S. 318, 325-27 (2015). “[T]he words of a claim are generally given their ordinary and

customary meaning [which is] the meaning that the term would have to a person of ordinary skill in the art in question at the time of the invention, i.e., as of the effective filing date of the patent application.” *Phillips v. AWH Corp.*, 415 F.3d 1303, 1312-13 (Fed. Cir. 2005) (en banc) (internal citations and quotation marks omitted). Although “the claims themselves provide substantial guidance as to the meaning of particular claim terms,” the context of the surrounding words of the claim also must be considered. *Id.* at 1314. “[T]he ordinary meaning of a claim term is its meaning to the ordinary artisan after reading the entire patent.” *Id.* at 1321 (internal quotation marks omitted).

The patent specification “is always highly relevant to the claim construction analysis . . . [as] it is the single best guide to the meaning of a disputed term.” *Vitronics Corp. v. Conceptronic, Inc.*, 90 F.3d 1576, 1582 (Fed. Cir. 1996). It is also possible that “the specification may reveal a special definition given to a claim term by the patentee that differs from the meaning it would otherwise possess. In such cases, the inventor’s lexicography governs.” *Phillips*, 415 F.3d at 1316. “Even when the specification describes only a single embodiment, [however,] the claims of the patent will not be read restrictively unless the patentee has demonstrated a clear intention to limit the claim scope using words or expressions of manifest exclusion or restriction.” *Hill-Rom Servs., Inc. v. Stryker Corp.*, 755 F.3d 1367, 1372 (Fed. Cir. 2014) (internal quotation marks omitted) (quoting *Liebel-Flarsheim Co. v. Medrad, Inc.*, 358 F.3d 898, 906 (Fed. Cir. 2004)).

In addition to the specification, a court “should also consider the patent’s prosecution history, if it is in evidence.” *Markman v. Westview Instruments, Inc.*, 52 F.3d 967, 980 (Fed. Cir. 1995) (en banc), *aff’d*, 517 U.S. 370 (1996). The prosecution history, which is “intrinsic evidence, . . . consists of the complete record of the proceedings before the [Patent and Trademark Office] and includes the prior art cited during the examination of the patent.” *Phillips*, 415 F.3d at 1317.

“[T]he prosecution history can often inform the meaning of the claim language by demonstrating how the inventor understood the invention and whether the inventor limited the invention in the course of prosecution, making the claim scope narrower than it would otherwise be.” *Id.*

In some cases, courts “will need to look beyond the patent’s intrinsic evidence and to consult extrinsic evidence in order to understand, for example, the background science or the meaning of a term in the relevant art during the relevant time period.” *Teva*, 574 U.S. at 331. Extrinsic evidence “consists of all evidence external to the patent and prosecution history, including expert and inventor testimony, dictionaries, and learned treatises.” *Markman*, 52 F.3d at 980. Expert testimony can be useful “to ensure that the court’s understanding of the technical aspects of the patent is consistent with that of a person of skill in the art, or to establish that a particular term in the patent or the prior art has a particular meaning in the pertinent field.” *Phillips*, 415 F.3d at 1318. Nonetheless, courts must not lose sight of the fact that “expert reports and testimony [are] generated at the time of and for the purpose of litigation and thus can suffer from bias that is not present in intrinsic evidence.” *Id.* Overall, although extrinsic evidence “may be useful to the court,” it is “less reliable” than intrinsic evidence, and its consideration “is unlikely to result in a reliable interpretation of patent claim scope unless considered in the context of the intrinsic evidence.” *Id.* at 1318-19. Where the intrinsic record unambiguously describes the scope of the patented invention, reliance on any extrinsic evidence is improper. *See Pitney Bowes, Inc. v. Hewlett-Packard Co.*, 182 F.3d 1298, 1308 (Fed. Cir. 1999) (citing *Vitronics*, 90 F.3d at 1583).

## **I. THE COURT’S RULING**

The Court’s rulings regarding the disputed claim terms of the ’494, ’086, ’305, ’918, ’154, and ’408 Patents were announced from the bench at the hearing as follows:

Before we start[], I wanted to give you my rulings on terms that we argued last week. So in this case, we have six patents and nineteen

disputed terms. We heard argument on nine disputed terms in four of the patents last week [– U.S. Patent Nos. 8,677,494 (“’494 Patent”), 8,079,086 (“’086 Patent”), 8,141,154 (“’154 Patent”), and 7,613,918 (“’918 Patent”) –] and will hear argument on the remaining terms here today. I did see your letter this morning, that you were withdrawing the terms “function type” in the ’305 Patent and “analyzer rules” in the ’408 Patent. Thank you for that. I appreciate the efforts.

. . . [First,] I want to rule on the nine disputes addressed last week. I will not be issuing a written opinion, but I will issue an order stating my rulings. I want to emphasize before I announce my decisions that while I am not issuing a written opinion, we have followed a full and thorough process before making the decisions I am about to state. I have reviewed each of the patents in dispute. There was full briefing on each of the disputed terms. There was an extensive appendix that included briefing and orders from numerous related cases and proceedings and there was argument here last week. All of that has been carefully considered.

Now as to my rulings. I am not going to read into the record my understanding of claim construction law generally. I have a legal standard section that I have included in earlier opinions, including recently in *Waters Corporation v. Agilent Technologies, Inc.*, C.A. No. 18-1450. I incorporate that law and adopt it into my ruling today and will also set it out in the order that I issue.

As to the person of skill in the art, neither party has asserted that there are any relevant differences in their proposed definitions of a person of ordinary skill in the art that would be relevant to these proceedings.

Now the disputed terms:

The first term is “Downloadable scanner” in claim 10 of the ’494 Patent and claim 24 of the ’086 Patent. Plaintiff asserts that the term should have its ordinary meaning, which it states is “scanner for scanning Downloadables.” Defendants construe the term to mean “software that decomposes code using conventional parsing techniques to identify suspicious patterns or suspicion computer operations.”

There are two differences in their constructions. At argument last week, Plaintiff agreed that the claimed scanner scans

Downloadables to identify suspicion patterns or suspicious computer operations. In fact, language to that effect is included in claim 10 already and the concept is included in claim 24. And Plaintiff acquiesced to including that language in the term “Downloadable scanner” in the prior *Cisco* case. Thus, the dispute here appears to be over whether the scanner “decomposes code using conventional parsing techniques.”

Here, I will agree with Defendants and construe the term to mean “software that decomposes code using conventional parsing techniques to identify suspicious patterns or suspicious computer operations.” The Federal Circuit previously addressed this term, stating that “[t]he Downloadable scanner in claim 10 is a code scanner that generates the DSP data by decomposing the code using conventional parsing techniques. The code scanner identifies suspicious computer operations in the Downloadable code and lists them as DSP data. Finally, the DSP data for the Downloadable is stored in a database.” [*Palo Alto Networks, Inc. v. Finjan, Inc.*, 777 F. App’x 501, 503 (Fed. Cir. 2019).] It is the first sentence of that holding that Defendants rely on for their construction.

Plaintiff says that those statements are merely background and do not address the issue here. I disagree that those are merely background or are irrelevant. Although the Federal Circuit did not set forth its reasoning, it stated what the “Downloadable scanner” is and relied on that in upholding the PTAB decision that the claims at issue were not unpatentable over the prior art.

Moreover, Plaintiff’s position is that anything that meets the language following “Downloadable scanner” in claim 10 is a “scanner.” During the PTAB proceedings, the PTAB rejected that position. The Petitioner had argued that the prior art had an emulator that carried out the same function as the scanner in claim 10. The PTAB found that “the fact that [the prior art’s] emulator might serve the recited function of ‘deriving the security profile data for the Downloadable’ . . . does not establish that a person of ordinary skill in the art would understand it to teach or suggest a ‘scanner.’” The Federal Circuit affirmed this, effectively rejecting the position Plaintiff offers here. [*Id.* at 505-06.]

And the Federal Circuit’s statements are supported by the intrinsic evidence. The written descriptions of the ’494 Patent and the ’086 Patent do not include disclosure of a “Downloadable scanner,” but the ’194 Patent specification is incorporated by reference in the

'494 Patent and discloses a "code scanner." ['194 Patent at col. 7, l. 55.] The same is true of the '086 Patent. ['086 Patent at col. 2, ll. 5-11.] At column 5, lines 42 through 48, the '194 Patent explains "the code scanner 325 uses conventional parsing techniques to decompose the code (including all prefetched components) of the Downloadable into the DSP data 310." It then continues, stating that the code scanner may generate DSP data that includes suspicious computer operations and that a code scanner "may search the code for any pattern, which is undesirable or suggests that the code was written by a hacker." ['194 Patent at col. 5, ll. 50-57.] Similarly, the code scanner is referenced in column 9 of the '194 Patent.

Plaintiff says that these are simply embodiments, but they are the only embodiments disclosed in the patent.

Finjan asserts that the proposed construction cannot be correct because the '494 and '086 Patents have examples of Downloadables that are in "machine language," which is "not typically 'decompose[d] using conventional parsing techniques.'" For that, Finjan cites to the Abstract of the two patents as well as claims of those patents. Finjan, however, fails to explain or support that assertion - relying solely on one sentence of conclusory attorney argument in their brief that does not negate my conclusion that my construction is supported by the intrinsic evidence.

The second term is "list of suspicious computer operations [that may be attempted by]" in claim 10 of the '494 Patent and claims 1, 17, 24, 41, and 42 of the '086 Patent. Plaintiff construe[s] this term to mean "list of computer operations that are deemed hostile or potentially hostile." Defendants construe it to mean "list of computer operations in a received Downloadable that are deemed hostile or potentially hostile."

The crux of this dispute is Defendants' proposed addition of the words "in a received Downloadable." In particular, the issue is whether the "suspicious computer operations" to be included on the list must be operations found within the Downloadable as opposed to derived from the Downloadable.

Here I agree with Plaintiff that the list must only be derived from data in the Downloadable.



So, I will construe it to mean “list of computer operations derived from a received Downloadable that are deemed hostile or potentially hostile.”

This [is] supported by the intrinsic evidence. First, the claim language itself. It requires deriving security data for the Downloadable, including the list that is at issue. The language requires that this list include operations that the Downloadable may perform, but it does not specify that those operations must all be found in the Downloadable rather than, for example, operations that are known to be used by a given source that has been identified for the received Downloadable.

Additionally, nothing in the specification of the '494 or '086 Patents or the '194 Patent requires the operations to be in the Downloadable. At argument, there was discussion of the two lists referenced in the '194 Patent. But there is no real dispute that the list referenced in the claims is the “short list.”

Additionally, the '194 specification supports this construction. For example, the Downloadable security profile data, which includes the list at issue, is derived specifically for the Downloadable received. The specification says that the Downloadable's derived security profile data can then be compared against “the access control list,” which “contains criteria indicating whether to pass or fail the Downloadable.” [See '194 Patent at col. 6, ll. 13-20.] While this pass-fail step is not itself recited or reached in claim 10, it illustrates that the “list of suspicious computer operations” generated is limited to a specific Downloadable - the one received. It doesn't, however, require that the list be operations found in the Downloadable – just that they be derived from it.

And there are additional statements in the specification, such as the discussion of “known Downloadables” in column 4[, lines 29-40,] of the '194 Patent, that suggest information about the operations a Downloadable may perform may come from other attributes, such as whether they were previously received or their source. [See also '194 Patent at col. 6, ll. 37-47.]

The third term is “appended Downloadable” in claim 1 of the '086 Patent. Plaintiff construes this term to have its “plain and ordinary meaning, which is ‘a Downloadable with data attached to it.’” Defendants construe it to mean “a Downloadable with a

representation of the Downloadable security profile data attached to the end of the Downloadable.”

There are two differences in language proposed, but the dispute here really is whether the “to the end” language proposed by Defendants is appropriate.

This dispute turns on the meaning of the word “append” and whether that requires attachment anywhere or attachment “to the end.” Here I construe this term to mean a “Downloadable with a representation of the Downloadable security profile data attached.” To be clear, the attachment need not be at the end.

Initially, I note that the patent does not use the word “append” in the written description. That word only appears in the claims.

The parties have each submitted proposed dictionary definitions of “append.” Some require attachment at the end and some do not. Given that at least one of these definitions defines “append” to mean “attach” or “affix” without requiring the attachment be at the end[, *see Append, Webster’s Pocket Computer Dictionary of the English Language* (New Revised ed. 1997) (“append: To add, as a supplement.”),] I will not read that limitation into the term absent clear intent by the patentee to do so or disclaimer.

Here, I do not find any such intent or clear disclaimer. Indeed, it appears that the specification of the ’086 Patent includes an embodiment in which the security profile data is sequenced before the Downloadable in the appended file - and thus it is not necessarily attached at the end. [’086 Patent at col. 11, ll. 16-26 (“by concatenating, within sandboxed package 340, MPC 341 for delivery to a Downloadable-destination first, followed by protection policies 342 and Downloadable 343”).]

The fourth term is “first function” / “second function” in claims 1, 4, 6, and 10 of the ’154 Patent. Plaintiff asserts that this term should have its plain and ordinary meaning and no construction is necessary. Defendants construe the term to mean “substitute function/original function, which is different than the first function.”

The parties’ dispute concerning this term is whether the “first function” and “second function” can be the same function.

This term has been the subject of more than one prior decision. And different courts have come out different ways. I, however, am going

to construe it according to its ordinary meaning. To be clear, I do not construe it to require that the first and second functions be different.

This is supported by the intrinsic evidence. There is nothing in the claims of the '154 Patent that requires the two functions to be different - or precludes the two from using the same function.

Defendants rely on the specification language that describes the “original” and “substitute” functions and says that “[t]o enable the client computer to pass function inputs to the security computer and suspend processing of content pending replies from the security computer, the present invention operates by replacing original function calls with substitute function calls within the content, at a gateway computer, prior to the content being received at the client computer.” [’154 Patent at col. 4, ll. 55-60.]

Defendants contend that the specification would not have described the functions as “original” and “substitute” functions if those functions were identical.

But other related patents use the terms “original” and “substitute” in the claims and the '154 Patent does not. And other places in the '154 Patent refer generally to the “first function” and “second function” without referencing “substitute” or “original” function. [See, e.g., '154 Patent at col. 7, ll. 32-43.]

Defendants also point to the Federal Circuit’s decision in *Palo Alto Networks*, but I do not read that decision as being inconsistent with Plaintiff’s proposal. The Federal Circuit noted that “each [claim] recit[es] a system or software program that executes a substitute function. The substitute function inspects the input to an original function to determine if executing the original function with the input violates a security policy.” The “‘first function’ is the inspection step in which the content is assessed for safety, and the ‘second function’ is when, having been deemed safe, the content is actually run.” [*Palo Alto Networks, Inc. v. Finjan, Inc.*, 752 F. App’x 1017, 1018 (Fed. Cir. 2018).]

Although it does use the terms “substitute” and “original,” I do not read those statements as requiring that the first and second function[s] be different.

Accordingly, I conclude that the specification does not support Defendants' argument that the first and second functions must be different. The term's plain and ordinary meaning governs, and no construction is necessary. [*See Phillips*, 415 F.3d at 1312 (“[T]he words of a claim are generally given their ordinary and customary meaning.”).]

Next, the fifth term, is “content processor” in claims 1 and 6 of the '154 Patent. Plaintiff asserts the term should have its plain and ordinary meaning, i.e., “a processor that processes content.” Defendants construe the term to mean “a processor that processes modified content.”

The dispute is over the addition of the [word] “modified” in Defendants' construction. Initially, I note that multiple courts have already construed this term to be governed by its plain and ordinary meaning – though none of those courts addressed the addition of the word “modified” that Defendants argue for here. [*See e.g., Finjan, Inc. v. Proofpoint, Inc.*, No. 4:13-cv-05808-HSG, Dkt. No. 267, at 18 (N.D. Cal. Dec. 3, 2015) (“[T]he term [‘content processor’] does not require any construction beyond its plain and ordinary meaning.”); *Finjan, Inc. v. Symantec Corp.*, No. 4:14-cv-02998-HSG, Dkt. No. 170, at 18 (N.D. Cal. Feb. 10, 2017) (“Accordingly, the Court finds that ‘content processor’ should be given its plain and ordinary meaning . . . .”); *Finjan, Inc. v. Bitdefender Inc.*, No. 4:17-cv-04790-HSG, Dkt. No. 101, at 21 (N.D. Cal. Feb. 14, 2019) (“In turn, the Court again finds that no construction is necessary for this term and that ‘content processor’ has sufficiently specific structure based on the claim language and specification.”).]

Defendants' argument is based largely on the same arguments it made in connection with the “first function” and “second function” terms. They argue that the Federal Circuit stated that asserted claims of the '154 Patent “recit[e] a system or software program that executes a substitute function,” which is the claimed “first function.” [*Palo Alto Networks*, 752 F. App'x at 2018.] They assert that this call to a substitute function only exists after the original content is modified at a gateway device to include it. ['154 Patent at col. 9, ll. 13-28 (“Content modifier 265 preferably modifies original content received by gateway computer 205, and produces modified content, which includes a layer of protection to combat dynamically generated malicious code.”).] Thus, Defendants argue that the claimed “content” received by the claimed “content

processor” is necessarily modified content since the claims require it to “includ[e] a call to a first function.”

[Defendants’] argument boils down to stating that the content must be modified because the first and second function[s] must be different. I have, however, already rejected that position.

In addition, I note that near the end of prosecution, Finjan [amended] claim 2 to remove the reference to “modified” content, because there was no antecedent basis for “modified content.” [October 5, 2011 Amendment and Response to Office Action at 138.] And claims of the ’289 Patent include an express requirement that the content is modified. [See, e.g., ’289 Patent at cl. 2 (“modifying the content”); *id.* at cl. 10 (“transmitting the modified content”).] The inclusion of this limitation in ’289 Patent claims confirms it should not be read into the claims of the ’154 Patent that do not include (and, in fact, removed) this limitation. [See *Andersen Corp. v. Fiber Composites, LLC*, 474 F.3d 1361, 1369 (Fed. Cir. 2007).]

The sixth term is “a call to a first function” in claims 1, 4, 6, and 10 of the ’154 Patent. Plaintiff originally asserted that it has its plain and ordinary meaning, which is “a call to a first function.” Defendants construe it to mean “a statement or instruction in the content, the execution of which causes the function to provide a service.” During the argument last week, Plaintiff agreed to Defendants’ proposed construction. I will thus adopt that construction.

The seventh term is “executable wrapper code” and “wrapper executable code” in claims 1, 12, 22, 28, and 33 of the ’918 Patent. Plaintiff asserts this term should have its plain and ordinary meaning, that is, “code that wraps an executable.” Defendants assert the term means “standalone executable file including appropriate computer account instructions and the byte address of potentially malicious executable code to be embedded therein.”

I will construe it to mean “executable code that wraps another executable.”

This construction is supported by the claim language itself. The parties agree that the two iterations of this term, i.e., “executable wrapper code” and “wrapper executable code,” mean the same thing. The plain words of “wrapper executable code” refer to executable code that serves as a wrapper.

This construction is also supported by the specification. The only reference to what this wrapper code is refers to it as an executable. The specification states: “CODE-B is preferably a standalone executable file, which includes appropriate computer instructions to initiate a new process using a specified user/group computer account name.” [’918 Patent at col. 7, ll. 63-66.] I recognize that that is in a preferred embodiment, but that embodiment refers to standalone executable code. I am not reading in standalone from the embodiment. But in light of the clear claim language, and the lack of embodiments in which the wrapper code is not executable, I will construe the term to mean “executable code that wraps another executable.”

The eighth term is “computer account(s)” in claims 1, 12, 22, 28, and 33 of the ’918 Patent. Here again, Plaintiff asserts “plain and ordinary meaning, which is ‘computer account(s).’” Defendants assert that this term means “user accounts for logging in to a specific client computer.”

Plaintiff asserts that the term is “a well-known term that people use every day” and, thus, there is no reason to depart from this plain and ordinary meaning in the absence of lexicography or a clear disclaimer. [See D.I. 76 at 68 (citing *Thorner v. Sony Comput. Entm’t America LLC*, 669 F.3d 1362, 1365-66 (Fed. Cir. 2012)).]

Defendants assert that during prosecution, in the December 23, 2008 Amendment, Plaintiff gave an express definition for this term in overcoming a prior art rejection. Specifically, the patentee stated that “the computer accounts of the claimed invention are user accounts for logging in to a specific client computer, such as the “system,” “restricted user” and “YBITZHAK” accounts. They then distinguished accounts in the prior art that were set up for “logging in to server computers.” [*Id.* at 22.]

I do not read this language to be a definition that restricts the computer accounts to only one computer. And I do not find it to be a clear and unambiguous disavowal of accounts that may be available on different computers. Thus, I will give this term its plain and ordinary meaning, which is “computer account(s).”

The final term from last week’s hearing is “combining” / “combine” in claim 1 of the ’918 Patent. Yet again, Plaintiff offers the plain and ordinary meaning, which it asserts is “combining” / “combine.”

Defendants assert that these terms mean “embed[ding] within CODE-B.”

I will give that term its plain and ordinary meaning. The patent uses the term in accordance with its ordinary meaning. Although there are embodiments in the specification that note that the way of combining is by embedding CODE-A within CODE-B using a file embedder, [*e.g.*, ’918 Patent at col. 8, ll. 8-11,] there are other references in the specification that are not so limited. For example, in the Abstract and in column 4, it simply refers to “combining” without specifying embedding. [*E.g.*, *id.* at col. 4, ll. 10-25 (“There is thus provided in accordance with a preferred embodiment of the present invention a method for computer security, including receiving content including potentially malicious executable code (‘CODE-A’), intended for downloading at a client computer, scanning CODE-A to derive a profile thereof, determining, based on the derived profile of CODE-A, an appropriate computer account from among a plurality of computer accounts, under which CODE-A may be processed by the client computer, wherein each computer account of the plurality of computer accounts has associated therewith a security context within which an executable run under such account is processed, combining (i) information about the determined computer account name and (ii) CODE-A, with (iii) executable wrapper code (‘CODE-B’) into combined code (‘CODE-C’), and forwarding CODE-C to the client computer for processing.”); *id.* at col. 4, ll. 26-46.]

And the claims at issue here do not refer to embedding or require an embedder. So, I will not read that into them.

\* \* \*

Thank you again for the arguments today on the remaining terms. As with the prior terms I have construed, I will not be issuing a written opinion, but I will include my rulings in the order construing the terms I ruled on earlier today. I have again carefully considered all of the materials submitted on these disputed terms as well as the argument here today.

The first term we have from today’s argument is “ruled-based content scanner” in claim 1 of the ’305 Patent. Plaintiff originally asserted plain and ordinary meaning. During the hearing, Plaintiff offered the construction “a scanner that scans and analyzes content to detect the presence of computer exploits using exploit rules.”

Defendants assert that it means “adaptive rule-based scanner that dynamically adapts itself to scan a specific type of content.”<sup>[1]</sup>

Here, I agree with Defendants and will construe the term to mean “a scanner that is able to adapt itself dynamically to scan a specific type of content.”

This construction is supported by the intrinsic evidence. In column 2, lines 10 through 27, the '305 Patent states that “[t]he content scanners of the present invention are referred to as adaptive rule-based (ARB) scanners. An ARB scanner is able to adapt itself dynamically to scan a specific type of content, such as inter alia JavaScript, VBScript, URI, URL and HTML.” The specification then goes on to distinguish the ARB scanners of the present invention from those in the prior art and explains that “ARB scanners are data-driven, and can be enabled to scan any specific type of content by providing appropriate rule files, without the need to modify source code.” The “rules files . . . serve as adaptors, to adapt an ARB content scanner to a specific type of content.” [See also '305 Patent at col. 8, ll. 61-66 (“[T]he present invention provides a flexible content scanning method and system, which can be adapted to any language syntax by means of a set of rules that serve to train the content scanner how to interpret that language. Such a scanning system is referred to herein as an adaptive rule-based (ARB) scanner.”).] Thus, the rules are used to “adapt” the scanner.

These statements are not embodiments. They are describing “the features of the ‘present invention’ as a whole” and thus “this description limits the scope of the invention.” *Regents of University of Minnesota v. AGA Med. Corp.*, 717 F.3d 929, 936 (Fed. Cir. 2013).

And my construction is also consistent with the prosecution history. Indeed, it is the construction Finjan itself previously represented to

---

<sup>1</sup> As noted *infra*, the parties agreed that “rule-based content scanner” in claim 1 of the '305 Patent and “scanner” in claims 1 and 22 of the '408 Patent “rise and fall” together. (D.I. 116). Although there was some confusion during the hearing as to the exact import of that agreement, Defendants submitted interchangeable constructions for the two terms that are distinguishable only by the order of the words employed – “a scanner that is able to adapt itself dynamically to scan a specific type of content” for “rule-based content scanner,” and “adaptive rule-based scanner that dynamically adapts itself to scan a specific type of content” for “scanner.” Thus, the Court discussed Defendants’ proposed constructions interchangeably.



the Patent Office, stating in the Patent Owner Preliminary Response[, at page 3,] that “[t]he ’305 Patent is directed to adaptive rule-based (ARB) content scanners, which adapt themselves dynamically to scan specific types of content.”

Today, Plaintiff argued that there are preferred embodiments of the invention in the specification that do not require the scanner to adapt itself to scan particular content. Those examples, however, do not alter the definition provided by patentee in column 2 [of the ’305 Patent]. For example, in column 19, in the embodiment on which Plaintiff relies for its new proposed construction, the scanner is an ARB scanner – and, thus, even though it does not repeat the definition in column 2, that definition still applies.

Similarly, the “sandbox scanner” in column 22 does not change the analysis. It is not an ARB scanner, and the way it works is inconsistent with the requirements of the claims.

Finally, the statements about enabling “behavioral analysis” that Plaintiff points to as being added in the ’305 Patent may describe additional functionality of an ARB scanner, but they do not change the definition. Indeed, for all of the changes that Plaintiff asserts patentees made from the ’408 to the ’305 Patent, they did not change the statement [in column 2] that the content scanners of the invention are ARB scanners as defined.

The parties submitted a letter to me stating that they agreed that the term “scanner” in claims 1 and 22 of the ’408 Patent would “rise or fall” with the definition of “a rule-based content scanner” [in claim 1 of the ’305 Patent,] and stating that they would not separately argue the term “scanner” here. There appears to have been some misunderstanding as to that agreement, however, as much of Plaintiff’s argument today was based on distinguishing the ’305 Patent from the ’408 Patent. The ’408 Patent contains the same definitions that I have already referred to in the ’305 Patent. [See ’408 Patent at col. 1, l. 65 – col. 2, l. 1.] Plaintiff has agreed that the “scanner” in the ’408 Patent claims is a “content scanner.” And thus, for the same reasons I have already stated with respect to “a rule-based content scanner,” I will construe “scanner” to mean “a scanner that is able to adapt itself dynamically to scan a specific type of content.”

The next term is “patterns of types of tokens” in claims 1, 13, and 25 of the ’305 Patent. Plaintiff offers the “plain and ordinary

meaning, which is ‘patterns of types of tokens.’” Defendants assert that it means “sequences of categories of tokens to match against.”

I will construe the term to mean “patterns of categories of tokens.”

This is supported by the intrinsic evidence. During the prosecution, in the September 15, 2010 Amendment at page 9, applicants distinguished between “tokens” and “type of tokens.” As to the latter, applicants noted that “‘types of tokens’ refers to a categorization of tokens into types.” [See also *id.* (“Applicants respectfully submit that the prior art does not relate to categorization of tokens into types, i.e., categories of tokens . . .”).]

There is no real dispute, however, as to what types of tokens mean. Instead, the crux of the dispute is Defendants’ attempt to insert “sequences” into the construction of “pattern.” Defendants assert that “sequences” is appropriate because the example that applicants gave during prosecution [(see *id.*)] was a sequence. I do not, however, read that example as limiting or as a clear disclaimer of other types of patterns that may exist.

In any event, I am not convinced that there is a real or ripe dispute here that requires further construction. If it turns out that these claims remain in the case and a dispute crystallizes such that further construction may be appropriate, I will address that after fact and expert discovery.

The next term is “parser rules” in claims 1 and 22 of the ’408 Patent. Plaintiff asserts “plain and ordinary meaning, which is ‘rules for parsing.’” Defendants assert “patterns of tokens that form syntactical constructs of program code.”

Here, I agree with Defendants. In column 2, lines 10 through 12, the ’408 Patent states that “patterns of tokens that form syntactical construct[s] of program code” are “referred to as ‘parsing rules.’” This statement is sufficiently clear to put one reasonably skilled in the art on notice that the inventor intended to redefine the claim term. [See *Elekta Instrument S.A. v. O.U.R. Sci. Int’l, Inc.*, 214 F.3d 1302, 1307 (Fed.Cir.2000) (“Absent an express intent to impart a novel meaning, claim terms take on their ordinary meaning.”); *Renishaw PLC v. Marposs Societa’ per Azioni*, 158 F.3d 1243, 1249 (Fed. Cir. 1998) (“The patentee’s lexicography must, of course, appear ‘with reasonable clarity, deliberateness, and precision’ before it can affect the claim.” (quoting *In re Paulsen*, 30 F.3d 1475,

1480 (Fed. Cir. 1994)); *Union Carbide Chems. & Plastics Tech. Corp. v. Shell Oil Co.*, 308 F.3d 1167, 1177-78 (Fed. Cir. 2002) (stating that the “heavy presumption in favor of the claim term’s ordinary meaning is overcome, however, if a different meaning is clearly and deliberately set forth in the intrinsic evidence”).] I will thus adopt this definition.

And I also note that Plaintiff at argument today did not say that the definition in the specification was incorrect. It said that it is redundant because the concept is already included in the claims. Although I am sensitive to reading in extraneous language, here I cannot conclude that it is wholly redundant, and there is a definition provided in the specification and I see no need to diverge from it.

The next two terms are “dynamically building” and “dynamically detecting.” Both of these terms are found in claims 1, 22, 29, and 35 of the ’408 Patent. Plaintiff offers “plain and ordinary meaning,” which for “dynamically building” is “non-static building” and for “dynamically detecting” is “non-static detecting.” Defendants don’t define dynamically building or dynamically detecting. They originally used those terms in their proposed constructions of each. For dynamically building, they asserted that it “requires that a time period for dynamically building overlap with a time period during which the incoming stream is being received.” Similarly, for dynamically detecting, Defendants asserts that it requires “that a time period for dynamically detecting overlap with a time period during which the parse tree is built.” Defendants have now amended their proposals to remove the word “dynamically” from each.

Here, again, it is not clear what the dispute is. The language in the claim limitation containing each of these terms includes the language that Defendants propose. I will not read that language into the term again and will instead give the terms their ordinary meanings as defined by Plaintiff. I will note, however, that both parties today agreed that the use of the word “while” in the claims requires that the “dynamic building” must overlap with the time the incoming stream is received and the “dynamic detecting” must overlap with the time during which the parse tree is built. [*See also* February 13, 2012 Amendment and Response to Office Action at 22.]

The next term is “incoming stream of program code” in claims 1, 22, 29, and 35 of the ’408 Patent. Plaintiff asserts that it should have its plain and ordinary meaning, “which is ‘incoming stream of

program code.” Defendants assert that it means “a continuous flow of program code that can be processed on-the-fly as it is being received over a network.”

Here, I agree with Plaintiff and give the term its plain and ordinary meaning. To be clear, however, as Plaintiff agreed today, inspection [of the incoming program code] has to occur before the entire content is downloaded.

As to “stream” versus “continuous flow,” Defendants cite to a dictionary definition. I do not, however, see anything ambiguous about the term “stream,” that requires us to change that word and the definition. As to the “processed on the fly” language in Defendants’ proposed construction, that is based on statements made during the prosecution in which applicants stated “processing of incoming data on-the-fly, as the data is being received over a network, is different than processing incoming data after all the incoming data has been received.” [August 26, 2010 Response to Office Action at 14.] I do not find that that statement clearly is referring to the term at issue here. To the contrary, it appears to be referring to other elements in the claim. And indeed, in proposing its construction in the briefing, Defendants relied on those other elements of the claim to explain what the claim as a whole requires.


The final constructions are for “instantiating, by the computer, a scanner for the specific programming language” and “instantiating a scanner for the specific programming language,” which are found in claims 1 and 22 of the ’408 Patent. Plaintiff asserts that these terms have a plain and ordinary meaning, which is “creating an instance of a scanner that can scan the specific programming language.” And Plaintiff agreed here today that “generating” could be substituted for “creating an instance” in its construction. Defendants assert that these terms mean “substituting specific data, instructions, or both into a scanner to make it usable for scanning the specific programming language.”

I will construe the “instantiating” terms to mean “generating or requesting a scanner that can scan the specific programming language.” This is supported by the intrinsic evidence. For example, column 15, lines 41 through 43 of the ’408 Patent states that “[a]n advantage of the present invention is the ability to generate . . . a multitude of content scanners within a unified framework.” [See also *id.* at col. 15, ll. 56-58.]

The word “substituting,” which Defendants propose, is not in the ’408 Patent. Defendants took that word from the PTAB. But the PTAB used a different standard and its statements are not binding on this Court.

Defendants also argue that “substituting” is consistent with the use of generic components described in the patent. The reference in the ’408 Patent to the ARB scanner being designed as a “generic architecture,” however, is an embodiment. [*See id.* at col. 6, ll. 14-20.]

Additionally, I find that the concept of requiring substitution appears to be inconsistent with the description of a scanner repository as referenced in the ’408 Patent. [*See id.* at col. 15, ll. 31-32.]

  
The Honorable Maryellen Noreika  
United States District Judge